



HAL
open science

Synthèse de vues à partir d'images de microscopes photoniques pour la micromanipulation.

Julien Bert

► **To cite this version:**

Julien Bert. Synthèse de vues à partir d'images de microscopes photoniques pour la micromanipulation.. Automatique / Robotique. UFR des Sciences et Techniques de l'Université de Franche-Comté, 2007. Français. NNT: . tel-00195867

HAL Id: tel-00195867

<https://theses.hal.science/tel-00195867>

Submitted on 11 Dec 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Année : 2007

THÈSE

présentée à

**L'U.F.R. DES SCIENCES ET TECHNIQUES
DE L'UNIVERSITÉ DE FRANCHE-COMTÉ**

pour obtenir le

**GRADE DE DOCTEUR DE L'UNIVERSITÉ
DE FRANCHE-COMTÉ**

en **AUTOMATIQUE**

(Ecole Doctorale Sciences Physiques pour l'Ingénieur et Microtechniques)

**Synthèse de vues à partir d'images de microscopes
photoniques pour la micromanipulation**

par

Julien BERT

Soutenue le 17 Octobre 2007 devant la Commission d'Examen :

Rapporteurs :

M. André CROSNIER

Professeur, université de Montpellier

M. Jean-Marc LAVEST

Professeur, université d'Auvergne

Examineurs :

M. Jacques JACOT

Professeur, E.P.F.L. Lausanne

M. Michel PAINDAVOINE

Professeur, université de Bourgogne

Mme Nadine LEFORT-PIAT

Professeur, E.N.S.M.M. Franche-Comté

M. Soukalo DEMBÉLÉ

Maître de Conférences, université de Franche-Comté

Mesdames et messieurs..., je vous signale tout de suite que je vais parler pour ne rien dire.

Oh! Je sais!

Vous pensez :

« S'il n'a rien à dire... il ferait mieux de se taire! »

Evidemment! Mais c'est trop facile!... C'est trop facile!

Vous voudriez que je fasse comme tous ceux qui n'ont rien à dire et qui le gardent pour eux?

Eh bien, non! Mesdames et messieurs, moi, lorsque je n'ai rien à dire, je veux qu'on le sache!

Je veux en faire profiter les autres!

Et si, vous-mêmes, mesdames et messieurs, vous n'avez rien à dire, eh bien, on en parle, on en discute!

Je ne suis pas ennemi du colloque.

Mais, me direz-vous, si on parle pour ne rien dire, de quoi allons-nous parler?

Eh bien, de rien! De rien!

Car rien... ce n'est pas rien!

[...]

Raymond Devos, « *Parler pour ne rien dire* », Sens-dessus dessous, éditions Stock, 1976.

Remerciements

Chers tous,

si j'écris ces remerciements sous forme de lettre c'est bien parce que je veux qu'ils soient comme tels, une lettre ouverte de remerciements à l'issue de mes travaux de thèse parmi vous, amies et collègues. Je me souviens que quelqu'un (je ne sais plus qui), m'a demandé pourquoi il est préférable d'écrire les remerciements à la fin. Bien évidemment ce n'est pas pour dire du mal des méchant rapporteurs ou responsables après la soutenance de thèse. Mais tout simplement parce qu'ils doivent s'écrire à la fin, lorsque tout est terminé, lorsqu'on fait le bilan de ces trois années passées si vite, lorsqu'on est loin, emprunt à la nostalgie. Alors commençons par le plus classique qu'il soit, voulez-vous.

Les travaux qui ont été menés pendant ces trois années de thèses ont été réalisés au Laboratoire d'Automatique de Besançon, le LAB pour les intimes. Je tiens donc à remercier les deux personnes qui ont été directeurs pendant mon séjour au LAB. Monsieur Alain Bourjault, ancien directeur du laboratoire, qui était professeur à l'Ecole Nationale Supérieure de Mécanique et des Microtechniques (ENSMM, c'est plus court) et également Monsieur Nicolas Chaillet, actuel directeur du laboratoire, professeur à l'université de Franche-Comté, de m'avoir offert d'excellentes conditions de travaux.

Une très jolie phrase de Stephen King¹, dans un de ces romans, était : « Stephen King n'est pas la rivière, Susannah. Il n'est que le tuyau dans lequel elle s'écoule ». Si la rivière est le travail accompli par le thésard ils en sont naturellement le conduit. Je parle de nos... de mes responsables. Je tiens à remercier Madame Nadine Lefort-Piat, professeur à l'ENSMM et de Monsieur Soukalo Dembélé, maître de conférences à l'université de Franche-Comté, de m'avoir permis l'accomplissement de ces travaux. Connaissant d'autres thésards, d'autres labos, je sais très bien que tous les responsables ne se battent pas pour les petits besoins de leurs thésards. Merci, à vous deux.

Je tiens à remercier mes deux rapporteurs, Monsieur André Crosnier, professeur à l'université de Montpellier et de Monsieur Jean-Marc Lavest, professeur et vice président de l'université d'Auvergne, pour m'avoir fait l'honneur de rapporter mon mémoire, pour y avoir consacré du temps notamment pendant la période estivale d'été, pour que je puisse soutenir cette thèse à la rentrée universitaire. Je vous remercie pour vos remarques et questions qui m'ont permis un

¹La tour sombre, Tome 7 : la tour sombre, p-634.

regard différent sur mes travaux. Cela fut très enrichissant.

Je suis très honoré que Monsieur Michel Paindavoine, professeur à l'université de Bourgogne, ait accepté de présider ma soutenance de thèse et je l'en remercie. Je suis très reconnaissant à Monsieur Jacques Jacot, professeur à l'École Polytechnique Fédérale de Lausanne pour sa participation à ce jury, pour ces remarques et questions constructives.

Comme dans toute quête il y a les compagnons de route. Sa table ronde avec son noyau dure, je leur dédie tous ces moments de vie partagés pendant ma thèse. Commençons avec Arnaud H. (dit le moine Shaolin ou Kung fu fighting). Je sais je n'oublierai pas, je commence par en prendre cinq puis : humilité. À tous les traquenards et glissades que nous avons dû te faire subir à l'insu de ton plein gré. Cédric A. (dit Dada), être ou ne pas être, telle est sa question. À toutes les soirées où nous avons burner le dance floor. François M. (dit Toufou ou chouchou il y a des variantes selon les jours), sans oublier Anne-Christine S. (dit Nanou) et Clémentine M. (dit Arlette). François, il m'arrive de repenser à tous ces dimanches passés au bord du lac de Saint-Point ou de Neuchâtel, à toutes les bouteilles que nous avons vidées (de plongée naturellement) dans ces eaux glaciales. Benoit B. (dit Ben) sans oublier Marianne O. bientôt B. (dit Marianne). Ben qui a le célèbre regard de l'homme sortant d'un coma de 40 ans, à tous les lendemains comateux partagés. Mickael P. (dit Mica), mon vieux compagnon de route, le tonton flingueur du Pont, à ces trois années de colocation, et plus d'amitié. Je cuisine toujours pour deux mais maintenant ça me fait plusieurs jours. Nicolas P. (dit Nico) et Fernanda C. G. d. S. (dit Fé), les Brésiliens, à toutes les explorations qui ont été partagés par l'ensemble de mes compagnons de fortune. Il y a également Viatcheslav T. (dit Slava), Elena T. (dit Elena) et leur petit nouveau (dit petit nouveau, en russe), les Ruskovs. J'ai toujours été étonné par ton incroyable culture de la boutade Française, mon cher Slava, « Un whisky avant ? ». À tous les fous rires commun de cette table ronde.

Je parle du noyau dur mais il y a également tous les autres, tous les doctorants et collègues du labo, que je remercie pour leur franche camaraderie. Je remercie également tous mes stagiaires direct ou indirect, où à chaque fois qu'il venait frapper à mon bureau avec une question, ils me donnaient l'occasion de prendre du recul ou d'approfondir mon travail. Je tiens à remercier aussi mes étudiant(e)s pour m'avoir ventilé l'esprit de temps en temps avec un peu de n'importe quoi en cours. J'adresse maintenant un remerciement collectif, à l'ensemble du personnel du LAB, administratif, technique, etc.

Je remercie ma famille, mes parents Brigitte et Daniel ainsi que mon frère Thomas pour leur soutien pendant ma longue, longue période scolaire. Je n'aurais pas pu passer d'un C.A.P. d'Électrotechnique à un Doctorat, aujourd'hui, sans vous.

La thèse n'est pas l'histoire d'une seule personne vous l'aurez compris, ainsi je clôturerai définitivement les pages de ma thèse, sur ces quelques mots de... et oui, encore du Stephen King² :

« Nous sommes un seul en plusieurs, la multiplicité faite unité »

Julien Bert, Houston
Décembre 2007

²La tour sombre, Tome 7 : la tour sombre, p-396.

Table des matières

Glossaire des notations	xiii
1 Introduction	1
1.1 Contexte	1
1.2 Objectifs	2
1.3 Histoire de la vision	4
1.4 De la macroscopie à la microscopie	11
1.5 Conclusion	14
2 Systèmes de Vision en Micromanipulation, du Réel au Virtuel	15
2.1 Introduction	15
2.2 Les systèmes d'imagerie et de vision dans les stations de micromanipulation	16
2.3 Propriétés d'un microscope photonique	22
2.3.1 Encombrement important	23
2.3.2 Faible distance de mise au point	24
2.3.3 Faible Profondeur de champ	25
2.3.4 Faible champ de vision	26
2.3.5 Dépendance à l'illumination	28
2.3.6 Résumé	28
2.4 Perception distribuée	30
2.5 Synthèse de nouvelles vues	32
2.5.1 Rendu à partir de modèle	33
2.5.2 Modèle à partir d'images	33
2.5.3 Rendu à partir d'images	34
2.5.4 Rendu à partir d'images sans géométrie	35
2.5.4.1 Le <i>Light field</i> et le <i>Lumigraph</i>	36
2.5.4.2 Le <i>concentric mosaics</i>	37
2.5.4.3 Le <i>mosaicing</i>	39
2.5.5 Rendu à partir d'images avec géométrie implicite	40
2.5.5.1 Interpolation de vue	41

2.5.5.2	Morphage de vue	41
2.5.5.3	Vue par triangulation commune	42
2.5.5.4	Transfert par colinéation	43
2.5.5.5	Transfert épipolaire	44
2.5.5.6	Transfert trifocal	44
2.5.6	Rendu à partir d'images avec géométrie explicite	45
2.5.6.1	Nuage de panneaux	46
2.5.6.2	Déformation $3D$	47
2.5.6.3	Calque d'image par profondeur	48
2.5.6.4	Texture par dépendance de vue	49
2.6	Synthèse d'images en microscopie photonique	51
2.7	Conclusion	53
3	Auto-Calibrage Stéréoscopique Faible en Microscopie Photonique	55
3.1	Introduction	55
3.2	Géométrie Epipolaire	56
3.3	Mire de calibrage par objet naturel	59
3.4	Acquisition d'images par couche	61
3.5	Détecteur de points d'intérêt	62
3.5.1	Détecteur de Harris	63
3.5.2	Détecteur Harris Simplexe	64
3.6	Mise en correspondance	68
3.6.1	Mise en corrélation par ZNSSD et ZNCC	69
3.6.2	Mise en corrélation par III-ZNSSD et III-ZNCC	70
3.6.3	Mise en correspondance par corrélation avec une contrainte de voisinage	75
3.7	Calcul du Calibrage	78
3.7.1	Estimation de la matrice de colinéation	79
3.7.2	Estimation de la matrice fondamentale	82
3.8	Estimation robuste par RANSAC	84
3.9	Quantification de la qualité du calibrage	86
3.10	Conclusion	88
4	Synthèse de Nouvelles Vues par Construction de Mosaïques	91
4.1	Introduction	91
4.2	Construction de mosaïque à partir d'images à faible texture	93
4.2.1	Recalage par mouvement calibré	94
4.2.2	Interpolation vs Super-résolution	94
4.2.3	Étalonnage par détection de contour à partir d'un motif carré	100
4.2.4	Étalonnage par mesure de surface d'un motif circulaire	106
4.2.5	Étalonnage dynamique	107
4.2.6	Exemple d'application	108
4.2.7	Conclusion	109
4.3	Construction de mosaïques à partir d'images à forte texture	111

4.4	Recalage à partir de l'intensité	111
4.5	Recalage à partir des indices invariants	113
4.5.1	Détection des indices invariants	115
4.5.2	Mise en correspondances des indices	115
4.5.3	Estimation robuste de la transformation projective	115
4.5.4	Transformation de l'image	116
4.6	Fusion	119
4.7	Détourage automatique	126
4.8	Application en micromanipulation	129
4.8.1	Dispositif	129
4.8.2	Mosaïque matricielle	130
4.8.3	Autofocus	132
4.8.4	Incrustation de vidéos	134
4.8.4.1	Incrustation à partir d'imageurs statiques	134
4.8.4.2	Incrustation à partir d'imageurs mobiles	135
4.8.5	Supervision de station de micromanipulation	139
4.9	Conclusion	142
5	Synthèse de Nouvelles Vues par Transfert Trifocal	145
5.1	Introduction	145
5.2	Géométrie trifocale	146
5.2.1	Extension de la géométrie épipolaire	146
5.2.2	Équations trilinéaires	147
5.2.3	Tenseur trilinéaire	149
5.3	Construction de nouvelles vues par transfert trifocal	151
5.3.1	Tenseur trilinéaire à partir de deux vues	151
5.3.2	Nouveau tenseur	153
5.3.3	Placement de la vue virtuelle	154
5.3.3.1	Méthode par calibrage stéréoscopique fort	155
5.3.3.2	Méthode par calibrage stéréoscopique faible	156
5.3.3.3	Notre méthode avec calibrage stéréoscopique faible	158
5.3.3.4	Zone de validité du placement	159
5.3.3.5	Placement automatique de la vue	161
5.3.4	Rendu	166
5.3.4.1	Forte extrapolation	166
5.3.4.2	<i>Forward mapping</i>	171
5.3.4.3	<i>Backward mapping</i>	177
5.3.4.4	Algorithme du <i>painting bridge</i>	180
5.4	Application en Micromanipulation	184
5.4.1	Vue de côté	185
5.4.2	Mesure de distance	186
5.4.3	Asservissement en position	187
5.5	Conclusion	190

6 Conclusions et Perspectives	193
6.1 Conclusion Générale	193
6.2 Perspectives à court terme	196
6.2.1 Synthèse de nouvelles vues par EDF	196
6.2.2 Synthèse de nouvelle vue par super-résolution	196
6.2.3 Calibrage	198
6.2.4 Construction de mosaïques	198
6.2.5 Transfert trifocal	199
6.3 Perspectives à long terme	199
Bibliographie	203
Index	217

Table des figures

1.1	Vladimir Zworykin et l'icône, 1929.	4
1.2	La première caméra CCD, G. Smith et W. Boyle, Bell labs, 1969.	5
1.3	Le filtre de Bayer, 1976. Chaque filtre affecte une intensité à des cellules spécifiques.	5
1.4	Combien de jambes a l'éléphant? Quel est le fonctionnement de notre cerveau pour traiter l'information visuelle?	7
1.5	Le visage de Sylvie, avant et après lissage de Gouraud, 1971.	8
1.6	La théière de Newell, <i>Boston Computer Museum, USA</i>	9
1.7	Shakey, Stanford, 1966 – 1972.	9
1.8	Le microscope de Hooke, 1664.	11
1.9	Image provenant d'un MEB, observation de différents pollens de plantes. .	13
1.10	Arrangement hexagonal d'atomes d'or, vue par un microscope à effet tunnel.	13
2.1	De gauche à droite : fibroscope ($\phi 1\text{ mm}$) ; mini-microscope TIMM ($155\text{ mm} \times \phi 22\text{ mm}$) ; mini-caméra endoscopique ($20\text{ mm} \times \phi 5\text{ mm}$).	18
2.2	Vue éclatée d'un fibroscope (de gauche à droite) : lentille convergente ; ensemble de fibres en réception (collecte de l'image) ; les trois fibres en émission (éclairage) ; gainage.	19
2.3	Graphique comparatif de la résolution des différents systèmes de perception visuelle en fonction de leur zone d'utilisation en micromanipulation. .	21
2.4	Montage optique d'un microscope de base, constitué d'un objectif à gauche et d'un oculaire à droite.	22
2.5	Encombrement du système de vision sur la station de micromanipulation du groupe SAMMI.	24
2.6	Microscope photonique à faible distance de travail.	25
2.7	Illustration de la faible profondeur de champ d'un microscope photonique. .	26
2.8	Illustration du faible champ de vision d'un microscope photonique. A gauche la résolution est de $9\ \mu\text{m}/\text{pix}$, à droite elle est de $3\ \mu\text{m}/\text{pix}$	27
2.9	Assemblage de deux pièces ($400\ \mu\text{m}$) pris selon deux angles de vue différents. Les réflexions spéculaires ajoutent de la complexité à la scène. (<i>Images Benoît Ballarin, LAB</i>)	29

2.10	Tableau des caractéristiques des objectifs Mitutoyo Plan Apo longue distance de travail.	29
2.11	Représentation des différentes caractéristiques d'objectifs Mitutoyo (Plan Apo longue distance de travail) en fonction du grandissement : a. la distance de travail, b. la résolution, c. la profondeur de champ et d. le champ de vision.	30
2.12	Les trois classes de la synthèse de vues : rendu à partir de modèle (RPM), modèle à partir d'images (MPI), rendu à partir d'images (RPI).	32
2.13	Chaîne de reconstruction 3D, de l'acquisition au rendu.	34
2.14	La chaîne de reconstruction 2D par rapport à la chaîne de reconstruction 3D.	35
2.15	La fonction plénoptique $P_7(V_x, V_y, V_z, \theta, \phi, \lambda, t)$	36
2.16	Fonctionnement du Light field : (a) principe de l'acquisition de chaque rayon des images dans le plan (u, v) avec un exemple d'imagette obtenue, (b) principe du rendu d'une nouvelle vue pour un seul élément discret du plan (r, s) avec un exemple d'échantillon obtenu.	37
2.17	Processus d'acquisition par <i>concentric mosaics</i> pour différents rayons C_i , la caméra suit le cercle pour trajectoire avec un angle α selon deux méthodes : (a) la caméra est à la normale du cercle, (b) la caméra est sur la tangente du cercle.	38
2.18	Vidéo panoramique par <i>Dynamosaicing</i> . (<i>Image A. Rav-Acha et al. ©2005 IEEE</i>)	39
2.19	Illustration du recalage : l'image 1, l'image 2 et l'image 3 sont projetées dans la mosaïque avec pour référence l'image 2.	40
2.20	Principe du view morphing.	42
2.21	Résultat d'un maillage par triangulation de Delaunay sur les deux images dans la méthode du <i>Joint view triangulation</i> . (<i>Images M. Lhuillier et L. Quan ©1999 IEEE</i>)	43
2.22	Couple d'images stéréo à gauche et le rendu à droite de la nouvelle vue par transfert épipolaire et <i>backward mapping</i> . (<i>Images S. Laveau et O. Faugeras ©1994 INRIA</i>)	45
2.23	Couple d'images stéréo à gauche et le rendu à droite de la nouvelle vue par transfert trifocal. (<i>Images K. Connor et I. Reid ©2002 BMVC</i>) . . .	46
2.24	De gauche à droite : le modèle 3D original composé de 5138 polygones ; la localisation de chaque panneau ; une partie des 32 panneaux texturés mis à plat ; l'objet reconstitué par les différents panneaux. (<i>Images X. Decoret et al. ©2003 ACM</i>)	47
2.25	De gauche à droite : texture de l'image et carte de profondeur ; texture de l'image avant et après transformation. (<i>Images M. M. Oliveira 2000</i>) . . .	47
2.26	De gauche à droite : représentation de l'objet par 6 textures de relief ; exemple de pré-transformation de texture ; nouvelle vue texturée à partir de deux quadrilatères de texture transformée. (<i>Images M. M. Oliveira 2000</i>)	48

2.27	De gauche à droite : image originale ; définition des calques en fonction de leur profondeur ; représentation de la scène sous forme de calques texturés. (<i>Images S. Baker, R. Szeliski et P. Anandan ©1998 IEEE</i>)	49
2.28	De gauche à droite : photographie originale avec marquage des arêtes ; modèle reconstruit ; reprojction du modèles fil de fer sur la photographie ; rendu de l'image de synthèse. (<i>image P. E. Debevec, C. J. Taylor et J. Malik ©1996 ACM</i>)	50
2.29	De gauche à droite : les coordonnées de texture de l'image de l'environnement codées sous forme de couleur d'un théière ; rendu final de la théière ; rendu final d'un tore. (<i>Images W. Heidrich et al. 1999</i>)	51
3.1	Géométrie épipolaire	57
3.2	Exemple de mires de calibrage pour stéréo microscope.	61
3.3	Détection et mise en correspondance de l'ensemble des points p et p' séparément par couche de profondeur de champ.	61
3.4	Les trois images de notre mire de $300\mu m$ choisie pour le calibrage avec seulement 3 couches (il n'y a que les images de droite représentées).	62
3.5	Définition d'un sommet, d'une arête ou d'une zone plate en fonction de la réponse du détecteur et des valeurs propres.	64
3.6	Exemple de détection de points par le détecteur de Harris : (a) une des images de la mire de calibrage, (b) image résultat de la réponse du détecteur (en fausse couleur), (c) et résultat des 907 points détectés sur l'image originale ($s = 0,02$).	65
3.7	Le schéma bloc du détecteur Harris simplexe.	66
3.8	Exemples de résultat du détecteur Harris simplexe, pour chaque image, la consigne était de 500 points.	68
3.9	A gauche, la fenêtre de corrélation rectangulaire. A droite, la fenêtre de corrélation III avec un facteur $k = 2$	72
3.10	Images stéréoscopiques de Tsukuba (384×288 pixels), utilisées comme image de test.	72
3.11	Erreur moyenne de la mise en correspondance sur les images de Tsukuba, avec un test qui comprend la recherche de 2500 points selon différentes tailles utiles de fenêtre pour une fenêtre rectangulaire et une fenêtre III (critère de corrélation ZNSSD).	73
3.12	Images stéréoscopiques d'une mire de $1 mm$ de diamètre (768×576) pixels, utilisées comme images de test.	74
3.13	Erreur moyenne de la mise en correspondance sur les images d'une mire, avec un test qui comprend la recherche de 2500 points selon différentes tailles utiles de fenêtre, pour une fenêtre rectangulaire classique et une fenêtre III (critère de corrélation ZNSSD).	75
3.14	Procédé du choix des candidats pour la mise en correspondance par relaxation de la distance Euclidienne entre la proximité des points.	76

3.15	Mise en correspondance des images de la mire : (a), (b) et (c) respectivement les couples de couches 1, 2 et 3, (d) regroupement de tous les points correspondants des trois couches.	78
3.16	Estimation robuste par RANSAC : (a) représentation de tous les couples de points appariés, (b) les couples de points utilisés pour estimer G (après RANSAC), (c) les couples de points utilisés pour estimer F (après RANSAC).	86
3.17	Distribution des erreurs pour 20000 estimations différentes de F par RANSAC (erreur calculée avec 513 points).	87
3.18	Distribution des erreurs pour 20000 estimations différentes de A par RANSAC (erreur calculée avec 513 points).	88
4.1	Exemple d'interpolation 2D : (a) image d'un disque en résolution d'origine et sa représentation en altitude, (b) image du même disque par interpolation $\times 20$ et sa représentation en altitude.	95
4.2	Exemple de super-résolution : (a) une des images originales basse résolution d'une séquence vidéo de 300 images, (b) la restauration par super-résolution. (<i>Images S. Farsiu et al. ©2004 J. Wiley & Sons, Inc.</i>)	96
4.3	Méthode de super-résolution par <i>shift-and-add</i> : (a) information du pixel gauche et droite en fonction du chevauchement d'un élément déplacé sub-pixéliquement, (b) exemple de reconstruction de l'image (u, v) à partir de 4 images déplacées de façon sub-pixélique selon les deux directions u et v	98
4.4	Grossissement $10\times$ d'une partie d'un pignon de montre : (a) par super-résolution <i>Shift-and-Add</i> , (b) par interpolation bicubique.	100
4.5	Image d'un pignon et d'un axe de montre : (a) une des images originales de la séquence vidéo, (b) application d'un filtre passe bas temporel avec les 100 images de la séquence vidéo.	101
4.6	Premières étapes pour la mesure de la relation métrique/pixel : (a) interpolation $\times 10$ de l'image de la mire, (b) détection des points de contour selon l'axe y , (c) limitation des points pour obtenir seulement l'arête supérieure et inférieure.	102
4.7	Comparaison entre une droite de régression par moindre carré (trait pointillé) et par moindre carré médian (trait plein).	103
4.8	Les dernières étapes de la mesure de la relation métrique/pixel : (a) ensemble des points de l'arête supérieure et inférieure de la mire, (b) détermination des droites de régression, des arêtes, par moindre carré médian, (c) on utilise le segment perpendiculaire aux arêtes de la mire pour mesurer sa largeur.	104
4.9	Graphique représentant la taille de la mire en fonction des différentes mesures dans le temps.	105
4.10	Taille du pixel en fonction de la distance de travail de la source d'image.	108
4.11	La mini-caméra endoscopique embarquée sur les tables de positionnement montées en (x, y, z)	109

4.12	Construction d'une mosaïque par mouvement calibré d'une scène de $55\text{ mm} \times 55\text{ mm}$ avec un facteur d'échelle de $50\ \mu\text{m}/\text{pixel}$ soit une image de dimensions de 1100×1100 pixels.	110
4.13	Diagramme résumant la méthode directe pour le recalage d'images.	114
4.14	Paire d'images d'une puce électronique d'EPROM (facteur d'échelle $450\text{ nm}/\text{pixel}$, taille de chaque image 640×480 pixels).	115
4.15	Indices invariants détectés dans les images avec le détecteur Harris simplexe (400 points).	116
4.16	Représentation des points appariés : (a) avant RANSAC, (b) après RANSAC.	117
4.17	Méthode de transformation d'une image : (a) <i>forward mapping</i> , (b) <i>backward mapping</i>	118
4.18	Le couple d'images I et I' original, avec l'image \hat{I}'_r le résultat du recalage de I' en fonction I avec une interpolation de type bicubique.	119
4.19	Le profil de la fonction $f_h(x, y) + s_{x>\gamma}(f_h)$ selon l'axe horizontal.	121
4.20	Représentation 2D et 3D du claqué C'_p pour $\alpha = 0$, $\beta = 0,8$ et $\gamma = 1$	122
4.21	Représentation des différents calques de la méthode de fusion adaptative : C'_p le calque des coefficients de pondération du fondu, C_o le calque de recouvrement entre les deux images d'entrée, C'_t le calque de transfert de l'image recalée et C_b le calque de fusion finale.	123
4.22	A gauche le couple d'images originales, à droite l'image de l'espace de travail représentant la mosaïque obtenue par recalage et fusion.	124
4.23	Comparaison entre des approches simples de <i>blending</i> de deux images quelconques : à gauche, la méthode de recopie directe ; au centre, la méthode par moyennage ; et à droite la méthode adaptative.	124
4.24	Méthode du détournement automatique : (a) Les quatre quadrants définis par le centre de gravité avec les points de chaque quadrant qui maximisent la surface utile, (b) définition du rectangle maximal à partir des quatre points des quadrants.	126
4.25	Evolution de l'aire du rectangle de la surface utile dans le quadrant q_3 en fonction de la coordonnée y de la ligne de balayage dans le masque.	127
4.26	Mosaïque des deux images de l'EPROM après détournement.	129
4.27	Première étape d'assemblage du mosaïque matricielle, la constitution des bandes mosaïques.	130
4.28	Assemblage des bandes mosaïque.	131
4.29	Détournement final sur la mosaïque complète de l'espace de travail.	131
4.30	Mosaïque d'une puce d'EPROM constituée de 5×3 images (facteur d'échelle $2,51\ \mu\text{m}/\text{pixel}$, de dimension de 1865×977 pixels et d'un champ de vision de $4,6\text{ mm} \times 2,4\text{ mm}$).	133
4.31	Suivi de la micropince par la méthode Lucas-Kanade (masque de 51×51 pixels pour 2 itérations).	137
4.32	Suivi de la micropince en utilisant des indices invariants comparé à un modèle pré-enregistré (masque de corrélation de taille de 29×29 pixels).	138

4.33	Suivi de la micropince par sous-fenêtrage, en bas à gauche de chaque image, l'image de la sous-fenêtre ainsi que son seuillage sont rappelés (sous-fenêtre de taille de 51×51 pixels).	139
4.34	Image de la séquence vidéo de l'incrustation d'une vue statique dans une mosaïque.	140
4.35	Deux images de la séquence vidéo de l'incrustation d'une vue statique et dynamique dans une mosaïque.	141
4.36	Le logiciel <i>MosaicingRTI</i> (<i>mosaicing</i> et incrustation temps réel).	143
5.1	Couple d'images stéréoscopiques utilisé pour l'illustration du transfert trifocal : en haut, les mires pour le stéréo-calibrage faible, en bas, les images réelles.	147
5.2	Géométrie trifocal.	148
5.3	Indices de notation du tenseur trilinéaire \mathcal{T}	150
5.4	Principe de la synthèse de nouvelle vue à partir de deux vues.	152
5.5	Placement de la vue virtuelle en fonction de t_x	159
5.6	Placement de l'objet ou de la scène en fonction du plan π pour l'estimation de la matrice de colinéation.	160
5.7	Bande de placement où la validité est incohérente.	161
5.8	Configuration du modèle équivalent simplifié du placement de la vue virtuelle.	162
5.9	A gauche, la carte des calques de même disparité, à droite, l'histogramme des disparités.	164
5.10	A gauche, l'image de la vue droite originale, à droite, les points de contour avec les points p_δ en noir.	165
5.11	Création de l'objet factice : (a) séparation des points en deux sous-ensembles par rapport à son centre de gravité, (b) représentation des points clés du segment factice.	166
5.12	Mesures du segment factice : (a) taille du segment en fonction de t^x , (b) comparaison entre le modèle de placement et les mesures réelles.	167
5.13	A gauche, les points de contours pour une vue à 0° , à droite, les mêmes points de contours pour une vue à 75°	168
5.14	Mécanisme de la perte de cohésion en forte extrapolation : (a) image de la scène avec le facteur d'échelle λ_1 , (b) plan π_1 au facteur d'échelle λ_1 utilisé pour l'estimation du tenseur trilinéaire, (c) image de la même scène mais avec une position de la vue virtuelle très éloignée.	169
5.15	A gauche, les points de contours pour une vue à 75° sans facteur de correction, à droite, la même vue avec facteur de correction.	171
5.16	Couple d'images stéréoscopiques utilisé pour le test de rectification épipolaire.	174
5.17	Couples d'images stéréoscopiques après rectification épipolaire.	174
5.18	A gauche, la carte de disparité sans recherche sub-pixélique, à droite, la même carte mais avec une recherche sub-pixélique.	175

5.19	Rendu par <i>forward mapping</i> : (a) illustration du phénomène d'écrasement des pixels, (b) le même rendu avec une technique de <i>z-buffer</i>	176
5.20	Rendu par <i>forward mapping</i> avec à gauche une des images originales et à droite le rendu final de la vue virtuelle.	177
5.21	A gauche, machine à état de la détection des trous de la méthode du <i>painting bridge</i> , à droite, quelques exemples de profil de lignes de l'image contenant des trous.	181
5.22	A gauche, représentation des trous détectés pour une ligne, à droite, le grossissement d'une partie de cette même ligne.	182
5.23	A gauche, deux lignes quelconques avec deux trous de tailles différentes, à droite, l'application sur ces lignes du <i>painting bridge</i> pour une longueur de pont limitée à 6 pixels.	183
5.24	A gauche, le rendu par <i>forward mapping</i> , au centre, une seconde passe de rendu par <i>painting bridge</i> avec un seuil de pont de 15 pixels, à droite, le même rendu mais avec un seuil de pont de 100 pixels.	184
5.25	Configuration du système de vision d'une station de micromanipulation : (a) configuration classique, (b) configuration stéréo vision.	185
5.26	Un exemple de couple d'images obtenue à partir de notre système de vision.	185
5.27	Vue virtuelle de la micropince pour différents angles de vue entre la vue de face et la vue de côtés.	186
5.28	Vue de côté virtuelle de la micromanipulation, avec la position de la micropièce à gauche et la position de la micropince à droite.	187
5.29	Asservissement visuel par une vue virtuelle : (a) vue virtuelle avec la pince en position initiale et la vérification sur la vue réelle correspondante, (b) vue virtuelle avec la pince en position finale et la vérification sur la vue réelle correspondante.	188
5.30	Asservissement visuel par une vue virtuelle : (a) erreur de position dans la vue virtuelle en pixel pendant l'asservissement, (b) position de la micropince sur l'axe z en μm pendant l'asservissement.	189
5.31	Le logiciel TTR <i>Trifocal Transfer Rendering</i>	191
6.1	Extension de la profondeur de champ : (a) image originales nette sur sa parti gauche et représentation d'une partie des coefficients de la décomposition en ondelettes, (b) idem avec une image nette sur la partie droite, (c) reconstruction de la profondeur de champ, l'image est nette sur la partir droite et gauche.	197
6.2	A gauche, le système de vision d'une station de micromanipulation, à droite, le détail d'un de ces imageurs.	200
6.3	A gauche, un ensemble de stations de micromanipulation avec les flux de production représentés, à droite, le détail de la connectique du système de vision dans la micro-usine.	200

Glossaire des notations

Chapitre 2

f_{obj}	Distance focale de l'objectif
f_{oc}	Distance focale de l'oculaire
Γ_{mic}	Grossissement du microscope photonique
l	Longueur de tube, distance entre les foyers de l'objectif et de l'oculaire
a_0	Distance minimale de vision nette (25 cm pour l'oeil humain)
d_t	Distance de travail du microscope photonique
ON	Ouverture numérique de l'objectif
λ	Longueur d'onde
n	Indice de réfraction
α_o	Demi angle d'ouverture de l'objectif
PdC	Profondeur de champ du microscope photonique
e	Résolution de la caméra
CdV	Champ de vision du microscope photonique
$T_{capteur}$	Taille du capteur de la caméra
C_I	Contraste de l'image I
P_n	Fonction plénoptique de n -ième dimension
α, θ, ϕ	Angles de vue dans l'espace
(V_x, V_y, V_z)	Position de vue dans l'espace
t	Variable du temps
(u, v, r, s)	Repère de boîte englobante dans l'espace
R_i	Rayon du cercle d'indice i

H_i	Matrice de rectification d'indice i
I_i	Image d'indice i
\hat{I}_i	Image rectifiée d'indice i
O_i	Centre optique d'indice i

Chapitre 3

\mathcal{P}^2	Plan projectif
\mathcal{P}^3	Espace projectif
ψ, ψ'	Vues des caméras gauche et droite
O, O'	Centre optique des vues ψ, ψ'
I, I'	Images des vues ψ, ψ'
π	Plan physique dans l'espace projectif \mathcal{P}^3
P	Point dans l'espace projectif \mathcal{P}^3 de coordonnées $(X, Y, Z)^T$
p	Point dans le plan projectif \mathcal{P}^2 de l'image I de coordonnées $(x, y, 1)^T$
p'	Point dans le plan projectif \mathcal{P}^2 de l'image I' de coordonnées $(x', y', 1)^T$
F	Matrice fondamentale
l	Ligne épipolaire du point p' dans la vue ψ
l'	Ligne épipolaire du point p dans la vue ψ'
v	Épipole de la vue ψ projeté du centre optique O'
v'	Épipole de la vue ψ' projeté du centre optique O
G	Matrice de colinéation
$\delta v'$	Parallaxe projectif
δ	Parallaxe relatif
$(*)_{\times}$	Opérateur matriciel de symétrie croisée, identique au produit vectoriel, exemple : $[A]_{\times}B = A_{\wedge}B = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$
$E(u, v)$	Intensité lumineuse selon les variations spatiale (u, v)
W	Fenêtre de corrélation
M	Matrice de l'approximation de $E(u, v)$ appelée aussi matrice de mouvement optique (<i>Optical flow</i>)
I_x	Dérivée partielle selon l'axe x de l'image I équivalent à $\frac{\partial I}{\partial x}$
I_y	Dérivée partielle selon l'axe y de l'image I équivalent à $\frac{\partial I}{\partial y}$
k	Constante définie de façon empirique
R	Réponse du détecteur de Harris

$\lambda_1 \lambda_2$	Valeurs propres de la matrice M
s	Seuil déterminant le nombre de points du détecteur de Harris
Np_d	Nombre de points détectés
Np^*	Nombre de points désirés
s^*	Seuil déterminant le nombre de points désirés Nc^*
p_d	Ensemble de points détectés par le détecteur Harris simplexe
I_s	Image seuillée par le seuil s
\bar{I}, \bar{I}'	Valeur de la moyenne des intensités des images I et I'
$c_{x,y}$	Valeur du critère de corrélation d'une fonction à deux dimensions (x, y)
$f_w(x, y)$	Fonction continue d'une fenêtre de corrélation rectangulaire
W_{\square}	Fonction discrète d'une fenêtre de corrélation rectangulaire
$\Delta x, \Delta y$	Période d'échantillonnage sur l'axe x et sur l'axe y
$k\Delta x, k\Delta y$	Période d'échantillonnage ré-échantillonnée par le multiple de k sur l'axe x et sur l'axe y
$\delta(x)$	Fonction de Dirac
$\text{III}_{\Delta x}(x)$	Peigne de Dirac 1D de période Δx
$\text{III}_{\Delta y}(y)$	Peigne de Dirac 1D de période Δy
$\text{III}_{\Delta x, \Delta y}(x, y)$	Peigne de Dirac 2D de période Δx et Δy
W_{III}	Fonction discrète d'une fenêtre de corrélation ré-échantillonnée par le peigne de dirac III
$W_{k\square}^{\text{I}}$	Fonction discrète d'une fenêtre de corrélation rectangulaire de dimension multiple de k et rendue impaire
$\text{card}(*)$	Cardinal de l'ensemble $(*)$, c'est à dire le nombre d'éléments de l'ensemble $(*)$
$\text{mod}_n(*)$	Modulo n de la valeur $(*)$, c'est à dire le reste de la division entière de $(*)$ par n
$\text{III}_w(I, p)$	Fenêtre de corrélation de type III sur l'image I autour du point p
$\text{dist}_E(p; p')$	Distance Euclidienne entre le point p et p' , $\text{dist}_E = \sqrt{(y' - y)^2 + (x' - x)^2}$
$\text{max}(*)$	Détermine la valeur maximum de l'ensemble $*$
$a_i b^i$	Notation Einstein $a_i b^i = \sum_{i=1}^n a_i b_i$ équivalent à $a_1 b_1 + a_2 b_2 + \dots + a_n b_n$ en algèbre et à $\vec{a}^T \cdot \vec{b}$ en vectoriel
σ_i	i ème valeur singulière
S	Ensemble de points
S_{e_n}	Sous-ensemble de points sélectionné aléatoirement dans l'ensemble S
\hat{M}_n	Modèle estimé à l'itération n par l'estimateur robuste RANSAC
s_ϵ	Seuil de l'erreur d'estimation de RANSAC

\hat{F}, \hat{G}	Matrice fondamentale et matrice de colinéation estimées
\hat{p}'	Ensemble des points p' estimés à partir de \hat{G}
\hat{l}'	Ensemble des lignes épipolaires estimées à partir de \hat{F}
P_A	Probabilité de trouver le bon modèle par RANSAC
$\epsilon\%$	Le pourcentage d'erreur de mise en correspondance entre $p \leftrightarrow p'$
N	Le nombre de points nécessaire à la détermination du modèle utilisé par RANSAC
t_i	Nombre de tirages aléatoires nécessaire pour avoir une probabilité P_A de trouver le bon modèle par RANSAC

Chap 4.

$\underline{Y}(t)$	Ensemble d'images basse résolution
$M_d(t)$	Modèle de dégradation d'un système optique
$\underline{X}(t)$	Ensemble d'images haute résolution
$\underline{V}(t)$	Modèle du bruit sur une image
I_f	Image filtrée
$\vec{\nabla} I$	Gradient de l'image I
a_{ij}, b_{ij}	Paramètres des droites passant par les deux points (x_i, y_i) et (y_j, x_j)
e_k	Erreur d'écart au modèle d'une droite
Q	Probabilité d'erreur
$\epsilon\%$	Pourcentage estimé de données aberrantes
σ	Ecart type des écarts au modèle d'une droite
m	Valeur médiane la plus faible des écarts au modèle d'une droite
M_{mn}	Moment de l'image I d'ordre m et n
Sp	Relation du pixel/métrique pour une distance Z_m objet/imageur
Z_m	Distance de travail entre l'objet observé et l'imageur
(\bar{x}, \bar{y})	Coordonnées du centre de gravité d'un objet dans une image I
$(\epsilon_{\bar{x}}, \epsilon_{\bar{y}})$	Erreur de position de la mire par rapport à un centre de gravité (\bar{x}, \bar{y})
d_m	Déplacement métrique de l'imageur
d_p	Déplacement pixélique de l'imageur
G	Matrice de colinéation
D_{inc}	Matrice de colinéation incrémentale
\mathcal{I}	Matrice identité
$E(d)$	Erreur de la fonction de corrélation en fonction des éléments de la matrice D_{inc}

\tilde{I}'	Image de la transformation projective de l'image I'
\tilde{p}'	Points de l'image \tilde{I}'
g_i	Opérateur image du module du gradient
J_i	Opérateur du Jacobien
e_i	Erreur de l'intensité entre deux images
G_r	Matrice de colinéation utilisée dans le recalage
I'_r	Image recalée de l'image I' par rapport à l'image I
p'_r	Points de l'image recalée I'_r
\hat{I}'_r	Image interpolée de l'image recalée I'_r
\hat{p}'_r	Point de l'image \hat{I}'_r
$[*]$	Opérateur d'arrondi à la valeur entière
cc_i ou cc_{ij}	Coefficients des pixels connexes
C'_p	Calque de pondération de l'image I'
C'^r_p	Calque de pondération de l'image \hat{I}'_r
f_h et f_v	Fonctions des rampes respectivement définies sur l'axe horizontal et vertical
$s_{a>b}(*)$	Opérateur de symétrie dans l'intervalle $a > b$ par rapport à b de la fonction $*$
C_t	Calque de transfert de l'image I
C'^r_t	Calque de transfert de l'image \hat{I}'_r
C_o	Calque d' <i>overlapping</i> (recouvrement) entre l'image I' et \hat{I}'_r
C_b	Calque du <i>blending</i>
$*^G$	Opérateur complément à 1, dans l'image également appelé l'opérateur de négatif, allusion au négatif photographique
I_{rendu}	Image du rendu du <i>blending</i> entre l'image I et \hat{I}'_r
E_t	Image de l'environnement de travail du <i>mosaicing</i>
q_i	i ème quadrant de travail dans une image
A_{q_i}	Aire de la surface utile dans le quadrant q_i
I^{phi}	Image binaire de l'image I
$min(*)$	Valeur minimum de l'ensemble $*$
$(x'_\square, x''_\square)$	Abscisses d'une boîte englobante
$(y'_\square, y''_\square)$	Ordonnées d'une boîte englobante
I_{mos}	Image mosaïque rectangulaire
\mathcal{F}	Focus d'une image
$E_{rr}(u, v)$	Erreur de déplacement selon (u, v)

Chap 5.

\mathcal{R}^3	Espace euclidien
ψ, ψ' et ψ''	Les trois vues réelles d'un système trifocal
O, O' et O''	Les centres optiques des vues respectives ψ, ψ' et ψ''
p, p' et p''	Les points respectifs des vues ψ, ψ' et ψ''
x, x' et x''	Abscisses des points p, p' et p''
y, y' et y''	Ordonnées des points p, p' et p''
A et B	Les matrices de colinéation entre les vues respectives $\psi \rightarrow \psi'$ et $\psi \rightarrow \psi''$
a_i^j et b_i^j	Les éléments des matrices de colinéation A et B avec i l'indexe des colonnes et j l'indexe des lignes
v' et v''	Les épipoles provenant de la projection du centre optique O dans les vues respectives ψ' et ψ''
α_i et β_i	Les coefficients des équations trilineaires
\mathcal{T}	Tenseur trilineaire composé des coefficients \mathcal{T}_i^{jk} et calculé à partir du système des vues (ψ, ψ', ψ'')
\mathbf{l}_i et \mathbf{K}_i	Les sous matrices du tenseur \mathcal{T}_i^{jk} pour les indexes fixés respectifs i et k
$\tilde{\mathcal{T}}$	Tenseur trilineaire estimé à partir du système à deux vues $(\psi, \psi', \psi'' = \psi')$
ϵ^{ijk}	Opérateur Levi-Civita (produit croisé tensoriel)
$\hat{\psi}''$	Vue virtuelle désirée
\hat{I}''	Image virtuelle de la vue virtuelle $\hat{\psi}''$
$\hat{\mathcal{T}}$	Tenseur trilineaire désirée à partir du système des vues $(\psi, \psi', \hat{\psi}'')$
\hat{B}	Matrice de colinéation entre les vues ψ et $\hat{\psi}''$
\hat{b}_i^k	Les éléments de la matrice de colinéation \hat{B}
\hat{v}''	L'épipole provenant de la projection du centre optique O dans la vue virtuelle $\hat{\psi}''$
D	Matrice de colinéation qui transforme la vue ψ'' en la vue $\hat{\psi}''$
d_i^k	Les éléments de la matrice de colinéation D
t	Le vecteur de translation entre v'' et \hat{v}''
R	Matrice de rotation pour le placement de la vue $\hat{\psi}''$
Ω	Vecteur des angles de rotation $(\Omega_X, \Omega_Y, \Omega_Z)^T$
E	La matrice essentielle d'un système stéréoscopique
K	Matrice des paramètres intrinsèques d'une caméra
R_0	Matrice de rotation entre les vues ψ et ψ'
t_0	Vecteur de translation entre les vues ψ et ψ'

$\delta_{pp'}$	Disparité entre les points p et p'
$d_{OO'}$	Distance dans l'espace entre les centres optiques O et O'
f	La distance focale
dz	La distance entre l'objectif d'une caméra et l'objet observé
D_r	Matrice de colinéation contenant qu'une transformation par rotation entre la vue ψ'' et la vue $\hat{\psi}''$
\mathcal{I}	La matrice identité
θ_x	L'angle de rotation entre la vue réelle ψ' et la vue virtuelle $\hat{\psi}''$
θ_x^*	L'angle de rotation aproximé de l'angle réel θ_x
P	Point de l'espace projectif \mathcal{P}^3
P^*	Point approximé du point P dans \mathcal{P}^3
h_0	La hauteur du triangle rectangle $P^*v'\hat{v}''$ rectangle en v'
L	Longueur de l'objet factice en pixel dans l'espace \mathcal{P}^2 de la vue $\hat{\psi}''$
L_0	Longueur de l'objet factice dans l'espace \mathcal{P}^3
\tilde{f}	Fréquence des mêmes disparités $\delta_{pp'}$
\bar{c}	Centre de gravité de coordonnée $(\bar{x}, \bar{y}, 1)^T$ d'un nuage de point
Ω_g et Ω_d	Ensemble des points respectif de la partie gauche et droite de l'objet factice
p_g et p_d	Point de contrôle de l'objet factice de type segment horizontal
\hat{p}''	Point de coordonnée $(\hat{x}'', \hat{y}'', 1)^T$ de la vue $\hat{\psi}''$
λ_i	Facteur d'échelle
(λ_x, λ_y)	Facteur de correction d'échelle pendant la reprojection de type <i>forward mapping</i>
(Δ_x, Δ_y)	Facteur de correction de centrage pendant la reprojection de type <i>forward mapping</i>
$\max(*)$	Valeur maximum dans l'ensemble $*$
Ψ	Angle de rotation utilisé pendant la rectification épipolaire
R	Matrice de rotation utilisée pendant la rectification épipolaire
\mathcal{G}	Matrice de transformation utilisée pendant la rectification épipolaire
H et H'	Matrices de rectification épipolaire pour les images respectives I et I'
O_{23}	Opérateur qui permet de passer d'une configuration $\mathcal{T}_i^{jk}(1, 2, 3)$ à $\mathcal{T}_i^{jk}(1, 3, 2)$
O_{12}	Opérateur qui permet de passer d'une configuration $\mathcal{T}_i^{jk}(1, 2, 3)$ à $\mathcal{T}_i^{jk}(2, 1, 3)$
\mathcal{J}	Matrice de corrélation entre les vues ψ et ψ'
l_{0i} et l'_{0j}	Les lignes épipolaires provenant de la matrice \mathcal{J}
i_n	Intensité d'un pixel de l'image I_n
p_c	Les points d'un contour

Introduction

Ce chapitre est consacré à l'entrée en matière du sujet de cette thèse. Nous présentons le contexte de l'étude qui est celui de la microrobotique et qui permettra de comprendre son orientation. Nous présentons aussi les objectifs de chaque chapitre. Pour terminer, un bref historique nous plongera dans l'origine du domaine dont relève cette thèse à savoir la vision par ordinateur, et plus particulièrement la microvision.

1.1 Contexte

Besançon, pôle de compétitivité des microtechniques, ancien creuset de l'horlogerie, est le nouveau berceau de la microrobotique. Le groupe Systèmes Automatisés de Micromanipulation et Micro-assemblage (SAMMI), du Laboratoire d'Automatique de Besançon, s'inscrit pleinement dans l'innovation dans la microrobotique. Toutes les composantes du groupe SAMMI sont des éléments gravitant autour de cette thématique. La finalité choisie est d'aboutir à des systèmes automatisés de production de microproduits, de taille en rapport avec celle des produits. Appelées micro-usines, elles ont été introduites par les Japonais en 1990. Sur de solides bases déjà ancrées de la microrobotique, l'idée de la micro-usine est lancée au laboratoire en 2002.

La production de micro-produits nécessite de maîtriser les tâches de micromanipulation et de micro-assemblage. La micromanipulation est la manipulation de produits à l'échelle micrométrique, c'est-à-dire l'échelle comprise entre $1 \mu m$ et $1 mm$. Les objets manipulés peuvent être aussi bien biologiques (cristaux, cellules, poudres, ...) qu'artificiels (billes, lentilles, membranes, ...). Une des difficultés de telles tâches provient du fait qu'à ces dimensions, la gravité est négligeable par rapport aux forces électrostatiques, capillaires, de Van der Waals et de *pull-off*. Il en résulte, que l'interaction entre objet et préhenseur s'en trouve changée. Ces forces dépendent des paramètres de l'environnement tels que l'hygrométrie, la température, le niveau d'électricité statique, qu'il est difficile de contrôler. Lorsqu'une pince lâche un objet, celui-ci reste collé au doigt de la pince, un peu comme quand vous jouez avec du sucre sur une table, les grains restent

accrochés à votre doigt. Il est aussi très difficile de positionner des pièces de très petites dimensions, certaines prennent des postures acrobatiques en fonction de la répartition des charges électrostatiques dans l'espace de travail. Une autre difficulté des tâches de micromanipulation provient des moyens d'observation utilisés à savoir l'incompatibilité entre les capteurs classiques et la plage de travail. L'automatisation des tâches de micromanipulation et de micro-assemblage constitue donc un vrai challenge.

Chaque composante du groupe SAMMI étudie une partie différente de la micro-usine. Les problématiques suivantes sont abordées : conception de microsystèmes (préhension, positionnement, changement d'outils, convoyage, etc) ; modélisation (nouveaux matériaux actifs, forces du micro monde en milieu sec ou liquide) ; perception et commande (commande robuste, commande sans modèle, commande référencée capteur (vision et force), etc.) ; stratégie et architecture (assemblage, sûreté de fonctionnement, communication, supervision, etc).

Cette thèse est un satellite de plus gravitant autour de ces sujets. Elle est clairement identifiée dans l'axe perception et commande. Elle est rattachée à un projet ANR Principes et Outils Nouveaux pour le Micro-Assemblage Automatisé (PRONOMIA). Ce projet a pour objet la micromanipulation en milieu liquide, milieu qui permet d'atténuer l'impact de certains phénomènes physiques fortement désagréables en milieu sec. Elle s'intègre également dans le projet Européen *Ultra Precision Manufacturing Process Based on Positional and Self-Assembly for Complex Micro-Products* (HYDROMEL). Celui-ci porte sur le micro-assemblage de produits tels que des roulements, des diodes lasers, des étiquettes RFID, etc. Cette thèse a permis d'initialiser des réflexions et apporter des solutions sur l'automatisation des tâches micro-assemblage. Plus généralement, les travaux menés dans cette thèse sont nécessaires aux différentes stations de micromanipulation du groupe SAMMI. En effet, la vision par microscopie est incontournable quand les objets que vous devez manipuler se voient difficilement à l'oeil nu.

1.2 Objectifs

Les résultats de cette thèse vont contribuer à l'automatisation des tâches de micromanipulation. Notre point de vue se place sur la perception visuelle à l'échelle micrométrique. Cette thèse s'inscrit comme étant la première au laboratoire portant entièrement sur la vision pour la micromanipulation. Par conséquent, son but premier, est de souligner les problématiques et d'éclaircir les verrous scientifiques et technologiques de la vision pour la micromanipulation. La ligne directrice, consistera à observer les systèmes de vision actuels dans les stations de micromanipulation, extraire les spécificités liées aux matériels et aux structures de vision utilisées pour en identifier les contraintes, puis par la suite de les compenser ou de les atténuer. Elle peut se résumer en trois points :

- identification des problèmes liés aux systèmes de vision en micromanipulation,
- amélioration de la perception visuelle par des algorithmes de vision avancée,
- validation des résultats sur des scènes concrètes de micromanipulation.

Ce mémoire est rédigé pour un public le plus large possible, ce qui signifie que théoriquement les méthodes de vision employées seront peut être trop détaillées pour

des personnes du domaine de la vision, mais certainement pas suffisamment pour les collègues microroboticiens. Certains algorithmes très connus du domaine de la vision artificielle seront quand même développés pour que ces méthodes soient directement utilisables par quiconque veut les utiliser, sans avoir à courir à la recherche de l'information bien souvent enfuie dans des couches théoriques parsemées à droite et à gauche dans plusieurs livres aux notations les plus exotiques les unes que les autres. N'oublions pas que cette thèse figure comme la première thèse de vision artificielle au laboratoire et comme telle elle se doit de poser les bases mêmes si elles sont simples. La thèse s'articule autour de techniques de synthèse de vues pour améliorer les caractéristiques des sources d'images à base de microscope photonique. La thèse s'organise en six chapitres de la façon suivante.

Chapitre 2 : Dans ce chapitre nous identifions la composition des systèmes de vision utilisés en micromanipulation et nous notons les spécificités de la principale source d'images à savoir le vidéo microscope photonique. Par la suite nous présentons comment un thème de la vision, le rendu à partir d'images, nous permettra de résoudre certains de ces problèmes.

Chapitre 3 : Dans n'importe quel système de vision le calibrage est une nécessité pour connaître la géométrie de ce système et l'utiliser de manière performante. Les systèmes de microscopie photonique n'échappent pas à la règle. Le type de calibrage dépend du type de rendu recherché. Nous présentons de nouvelles méthodes et améliorations de la chaîne de calcul pour réussir à calibrer un stéréo microscope photonique.

Chapitre 4 : La première technique de synthèse de vues utilisée permet d'augmenter le champ de vision du vidéo microscope photonique. Cette technique appelée construction de mosaïques (*mosaicing*), fusionne les images locales obtenues par balayage du champ de travail, pour au final obtenir une image globale de grande résolution. Pour les besoins des tâches de micromanipulation nous incrustons en temps réel les flux vidéos locaux de l'espace de travail dans le fond ainsi créé.

Chapitre 5 : La deuxième technique de synthèse de vues permet de libérer l'espace de travail de la station de micromanipulation. Les microscopes photoniques trop encombrants, ne permettent pas d'avoir toutes les vues nécessaires au bon déroulement des tâches. Nous utilisons une méthode de reconstruction originale pour créer des vues nouvelles. Cette méthode, le transfert trifocal, donne la possibilité de synthétiser des microscopes virtuels. Nous présentons comment elle peut être utilisée dans des tâches en micromanipulation.

Chapitre 6 : Ici nous discutons des différents apports de la thèse. Nous concluons sur les différents résultats, et présentons quelques perspectives à court terme, dont une en particulier, complémentaire des deux méthodes présentées précédemment pour augmenter la profondeur de champ. Nous présentons également des perspectives à plus long

terme, sur l'avenir du système de vision dans la micro-usine.

La section suivante présente l'histoire de la vision, son évolution, en macroscopie comme en microscopie. Il est impensable, de contribuer à un domaine sans en connaître les origines.

1.3 Histoire de la vision

La vision est un des sens les plus importants chez l'être humain et l'animal. Comment ce petit organe, l'oeil, nous permet de collecter une multitude d'informations de notre environnement dans la vie de tous les jours? Mais percevoir n'est pas suffisant, faut-il encore comprendre ce que l'on voit. Dans l'étape capitale de la perception est inclu le traitement de l'information. Elle est assurée par l'organe le plus important et le plus inconnu du corps humain, le cerveau. C'est notre cerveau qui détermine si ce que nous voyons est une pomme ou une poire. C'est encore lui qui coordonne les actions à effectuer selon l'information perçue.

La chaîne sensori-psycho-moteur, percevoir, décider et agir utilise respectivement, les yeux, le cerveau et les muscles. Comment l'homme, de façon artificielle a réussi à transformer cette chaîne par acquérir, traiter et commander en utilisant respectivement les imageurs, les ordinateurs et les actionneurs?

Acquisition de l'information : Tout commence avec l'invention de la télévision, l'icôneoscope. Vladimir Zworykin ingénieur russo-américain née en 1889 à Murom en Russie invente la première télévision en 1929 appelée l'icôneoscope (Figure 1.1). Le tube électronique muni d'un canon à électrons et d'une plaque photosensible, fonctionne comme nos télévisions actuelles, les électrons viennent percuter la plaque sensible à la lumière. Ces points lumineux (pixels), constituent un ensemble d'éléments qui forme l'image. Une des propriétés importantes de l'icôneoscope est sa réversibilité. En effet il peut fonctionner comme récepteur. Si une image est projetée sur la plaque photosensible, celle-ci retient plus ou moins les charges électriques en fonction de la lumière au moment où le faisceau d'électrons vient balayer la surface. Les charges électriques sur la plaque sont récupérées sous forme de signal électrique. La première caméra à tube était née. Cet oeil électronique est le premier objet artificiel capable de convertir des informations lumineuses complexes en signaux électriques. Les caméras à tube produisent des signaux analogiques de faible résolution et de faible qualité, nous sommes encore très



FIG. 1.1 – Vladimir Zworykin et l'icôneoscope, 1929.

loin de nos caméras actuelles. Il faudra encore quarante ans pour assister à une vraie révolution dans le domaine de l'acquisition d'image.



FIG. 1.2 – La première caméra CCD, G. Smith et W. Boyle, Bell labs, 1969.

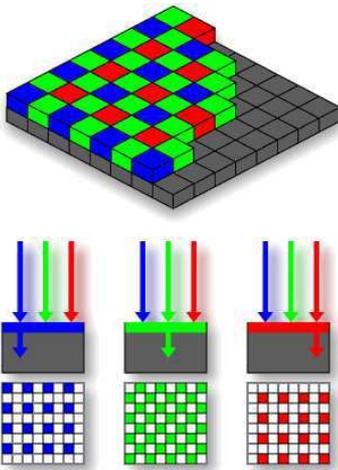


FIG. 1.3 – Le filtre de Bayer, 1976. Chaque filtre affecte une intensité à des cellules spécifiques.

En 1969, au matin du 17 octobre, dans un des bureaux du *Bell labs*, deux physiciens travaillent sur la conception de nouveaux types de mémoire à semi-conducteurs pour les ordinateurs. Leurs idées de base, manipuler de petites poches de charges électriques dans une matrice de silicium. Ce jour là, Smith et Boyle viennent d'échafauder, ce qui sera une des révolutions la plus importante dans le domaine de l'imagerie par ordinateur. George Smith et Willard Boyle inventent le premier capteur CCD (*Charge Coupled Device*) (Figure 1.2). Le CCD utilise l'effet photoélectrique en transformant l'énergie des photons incidents en énergie électrique. Un an plus tard les deux chercheurs proposent une caméra intégrant l'optique et le capteur CCD. Sa qualité, sa définition et sa capacité à travailler dans une large bande de longueur d'onde lumineuse, intéresse fortement les scientifiques puis la télévision. Il y aura sans cesse des améliorations, une importante est l'apparition des CCD couleurs. En plaçant un filtre de couleur rouge, vert et bleu devant certains pixels du CCD, il est possible de reconstituer une image polychromatique. Une des technologies les plus utilisées actuellement est le filtre de

Bayer inventé par Bryce E. Bayer en 1976 chez *Eastman Kodak*. Sa particularité est que le filtre permet un plus grand nombre de pixels vert par rapport à ceux de rouge et de bleu, pour une adaptation de la colorimétrie de l'image à la bande passante de l'oeil humain, plus prononcée sur les verts (Figure 1.3). Cette technique a un inconvénient, comme chaque pixel est affecté à une couleur, on diminue la résolution du capteur. Une des solutions que proposa *Philips* avec le « *Philips prism* » c'est de séparer les trois composantes rouge, vert et bleu par décomposition via un prisme pour les projeter sur trois CCD différents, permettant ainsi de conserver la totalité de la résolution de chaque capteur. La géométrie des cellules sensibles s'est aussi améliorée avec le Super-CCD annoncé par *Fujifilm* en 1999, permettant une plus grande exploitation de la surface sensible. L'intégration d'une plus grande densité d'éléments sur la surface permet d'avoir un capteur de plus grande résolution. Récemment *Kodak* (en 2005) propose le CCD progressif qui bénéficie là encore d'une plus grande surface utile. Pendant toutes ces années d'autres technologies ont vu le jour en parallèle, on ne peut pas ne pas mentionner les capteurs CMOS (*Complementary Metal Oxide Semi-conductor*) apparus dans les années 1980 plus complexes à fabriquer mais produits selon des techniques classiques de micro-électronique et de ce fait peuvent avoir des résolutions importantes (14 mégapixels en 2003) et permettant des traitements *in-situ*. Ces technologies se sont rapidement démocratisées, on les retrouve partout, allant du simple téléphone portable au plus grand télescope spatial tel Hubble. Smith et Boyle furent récompensés bien plus tard, en janvier 2006 par le prix *Charles Stark Draper Prize* de l'*United States National Academy of Engineering* (NAE) pour leurs travaux sur le CCD. Historiquement l'icône reste ancré dans nos caméras numériques actuelles puisque la taille des capteurs CCD, donnée en pouces par le fabricant, est fautive. Un capteur de 1/3 de pouce ne fera pas 8,4 mm de diagonale. En fait, nous faisons encore référence au diamètre externe des caméras à tube encore utilisées jusqu'à la fin des années 80. Le diamètre théorique du tube est de 18 mm, ainsi un capteur de 1/3 de pouce fera 6 mm de diagonale. Il s'agit d'une convention de langage qui perdure depuis trente ans.

Traitement de l'information : Percevoir n'est rien sans la compréhension de l'information collectée. La perception visuelle peut se séparer en deux branches qui dépendent de ce que l'on utilise dans l'image pour retrouver l'information et de ce que l'on en fait. La première communément appelée traitement de l'image est basée sur l'extraction et la transformation de l'information. Elle utilise l'information photométrique dans l'image c'est-à-dire l'intensité des pixels de l'image. Par exemple la détermination du centre de gravité d'un objet dans une image relève du traitement de l'image, car on extrait une information qui est la position de l'objet, par le biais de la photométrie de l'image, c'est-à-dire l'intensité des pixels de l'objet. La deuxième nommée de façon plus restrictive, vision artificielle ou vision par ordinateur, est basée sur l'interprétation et la représentation de l'information. Elle utilise généralement l'information géométrique extraite de l'image ou des images provenant du traitement de l'image. Cela signifie que dans la majorité des cas la vision par ordinateur est basée sur la position et la relation des pixels entre eux dans une ou plusieurs images. Dans l'exemple d'un objet dans une

scène, l'approche de la vision par ordinateur sera de calculer son déplacement, sa vitesse ou sa position dans l'espace à partir de points de cet objet pour différentes vues à différents moments. On utilise cette fois-ci seulement l'information géométrique des points provenant du traitement de l'image. C'est pourquoi traitement de l'image et vision par ordinateur devraient être indissociables car l'un précède toujours l'autre. Un processus de vision par ordinateur a toujours une première étape de traitement de l'image.

Le traitement de l'image est apparu dans les années 1950 avec un réel besoin scientifique. Au CERN les physiciens des particules produisaient 10000 à 100000 images par expérience dans les chambres à bulles pour étudier les trajectoires des particules produites par la collision d'atomes. Une explosion sans précédent du traitement de l'image apparaît durant les années 1960. Ce boom est en partie lié à une vague de technologies appareillées d'imageurs comme les satellites, les microscopes électronique à balayage (MEB) et les radiographies.

Les systèmes d'acquisition peu performants produisaient des images de faible qualité avec de fortes déformations optiques et chromatiques. Il a fallu développer rapidement des méthodes permettant le pré-traitement des images pour leurs corrections. Deux domaines font leurs apparitions, la compression d'images au début des années 60, et la reconnaissance de formes en milieu et fin des années 60. La compression d'images était plus que nécessaire, les images de l'époque posaient des problèmes de volume et de traitement très lent. Sans parler des séquences d'images animées pour la télévision qui nécessitait des moyens de stockage importants. La volonté d'automatiser l'interprétation des images a fait éclore la reconnaissance de formes. De la lecture optique de documents dactylographiés, en passant par l'imagerie médicale par radiographie ou encore la course effrénée entre américains et russes sur l'interprétation d'images satellitaires pour les informations tactiques militaires ont fait que la reconnaissance de formes a eu un essor impressionnant pendant cette période. De façon plus transcendante le formalisme de l'interprétation de l'information, et les bases des sciences cognitives ont été posées par David Marr. Né en 1945 à Essex en Angleterre, il est diplômé en mathématiques et en neurosciences. Il travailla pendant de nombreuses années au MIT. Il publie entre 1971 et 1972 le fondement de la théorie des fonctions du cerveau appliquée à la vision. Il cherche à comprendre comment le cerveau humain arrive à extraire de l'information et comment l'homme pourrait le formaliser de façon mathématique (Figure 1.4). Il définit les bases formelles du traitement de l'image.

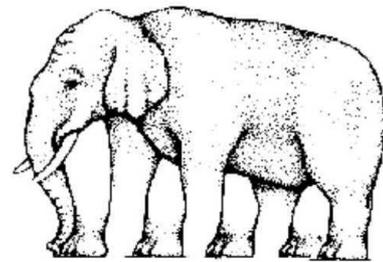


FIG. 1.4 – Combien de jambes a l'éléphant ? Quel est le fonctionnement de notre cerveau pour traiter l'information visuelle ?

De la même façon la vision, c'est-à-dire l'interprétation géométrique de l'image, a débuté dans les années 1960. Lawrence G. Roberts né en 1937 dans le Connecticut, est très connu pour avoir fait partie des quatre personnes associées à la naissance d'internet. Par contre il l'est beaucoup moins pour sa thèse soutenue au MIT en 1965. Pourtant il a

bel et bien inventé la discipline de la vision à ce moment là, il fut le premier à représenter en trois dimensions des polyèdres simples à partir d'une image en deux dimensions. Cette représentation a été la première synthèse d'image. La vision vue du côté géométrique a toujours été fortement liée à la représentation de l'information sous forme synthétique. Effectivement si la reconstruction de l'information dans l'espace commençait à faire sa réputation, il fallait aussi pouvoir se la représenter. La reconstruction d'image s'est ainsi d'abord développée à partir de formes simples et abstraites comme des cubes ou des polyèdres via leurs sommets et leurs arêtes. La représentation par l'image de synthèse était tout aussi simple que ses modèles, l'objet était sous forme filaire. Très vite le besoin d'un rendu plus complexe s'est nettement fait sentir. Il faut attendre l'année 1970 pour qu'un ingénieur de chez Renault, Pierre Bézier invente les courbes et les surfaces à point de contrôle qui portent son nom. La représentation d'objet de synthèse n'est plus simplement filaire mais à facettes. La visualisation d'objet en facettes n'était pas encore tout à fait très esthétique, pour obtenir un objet plus lissé, et donc plus réel, il fallait augmenter le nombre de facettes ce qui revenait à voir des objets trop complexes et trop longs à afficher.

Une des grandes découvertes, encore utilisée de nos jours, fut le lissage de Gouraud. Henri Gouraud jeune étudiant français, pose les bases de l'ombrage de Gouraud pendant sa thèse en 1971 à l'Université d'Utah, avec Ivan Sutherland, qui n'est autre qu'un des pères de l'internet avec Roberts. Son concept est simple, au lieu d'augmenter le nombre de facettes, on en garde un nombre limité, mais on peint tout simplement la surface comme si elle était courbe alors qu'elle est géométriquement constituée de facettes. Cette technique est tellement utilisée que le musée de l'informatique de Boston expose désormais le modèle de la tête de Sylvie (Figure 1.5), la femme de Gouraud, utilisée pour illustrer les ombrages. Si l'information géométrique est bien restituée par rapport à l'image source en revanche l'information photométrique ne l'est pas. Les représentations même lissées ne sont qu'en niveaux de gris. C'est Ed Catmull, actuel président de Pixar, en 1974 qui a l'idée d'appliquer la méthode du papier peint à ces modèles tridimensionnels, en leur associant l'image du modèle. Il est désormais possible d'extraire de l'information et se la représenter sous formes géométrique et photométrique.

C'est en 1975 que la vision donne lieu à un nouveau domaine à part entière, déjà présent en 1971, elle signe son indépendance avec Martin Newell. L'infographie, ou l'image de synthèse et non plus, la synthèse d'image, s'est émancipée dans le creuset du « *computer graphics* » de l'Université d'Utah. Newell est le premier à avoir modélisé manuellement un objet complexe, une théière, à partir de son modèle réel. Il créa une des plus célèbres images de l'infographie, la théière, devenue

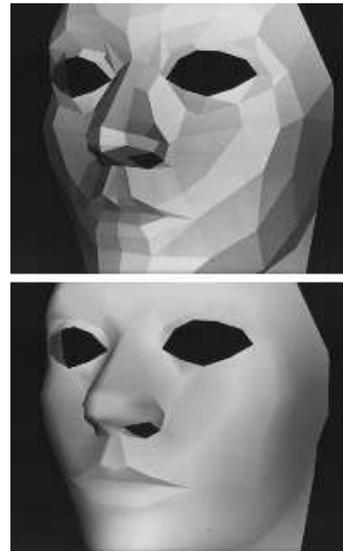


FIG. 1.5 – Le visage de Sylvie, avant et après lissage de Gouraud, 1971.



FIG. 1.6 – La théière de Newell, *Boston Computer Museum, USA*.

depuis un objet classique de test pour les applications 3D. La théière qui a servi pour modèle est exposée au musée de l'informatique de Boston (Figure. 1.6). Cet objet est devenu le symbole de l'infographie 3D, dans certains logiciels de modélisation 3D (Ex. : 3D Studio Max) on retrouve en plus des primitives classiques (sphère, cylindre, ...), la théière de Newell. Maintenant il apparaît clairement deux domaines, la vision, qui étudie l'extraction et l'interprétation de l'information visuelle, et l'infographie qui étudie l'amélioration de sa représentation, appelé le rendu.

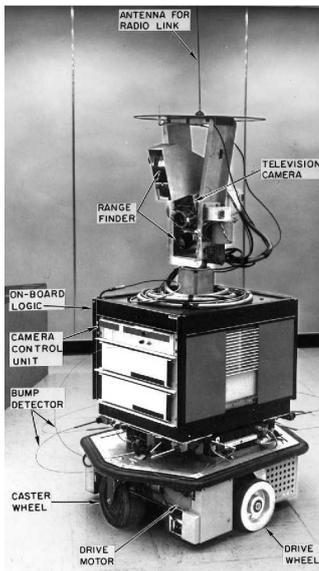


FIG. 1.7 – Shakey, Stanford, 1966 – 1972.

Action et perception : Au début, la plupart de l'information était analysée par l'homme, le traitement de l'image n'était là que pour l'améliorer et l'accentuer. Par la suite, les données sont analysées de façon automatique, la vision fait un diagnostic sur l'image analysée. Il faut attendre les années 1970 pour faire interagir perception et action. C'est dans cette période que des robots commencent à être équipés de caméra. On commence à comprendre que l'on est capable d'associer une action en fonction des perceptions visuelles, tout comme le fait l'être humain (observer, réfléchir et agir). Les différents programmes spatiaux ont apporté une dynamique dans le domaine de la vision en robotique mobile. Qu'elle soit soviétique pour l'exploration lunaire avec le robot mobile Lunokhod 1 en 1970, ou américaine avec le programme Viking de la NASA en 1976. Entre 1966 et 1972 l'Institut de recherche de Stanford développe le robot mobile Shakey (Figure 1.7). Il est équipé d'une caméra, de capteurs de proximité et d'une liaison radio. Il est le premier robot qui peut raisonner sur ses actions. La vision en robotique permet de construire de manière dynamique des modèles de leurs environnements. C'est pour cela, qu'elle a émergé et qu'elle est toujours très présente, dans les systèmes de navigation en robotique mobile. En 1977, Hans Moravec, chercheur à Stanford pour de la navigation visuelle

depuis 1973, conçoit avec l'aide de Victor Scheinman le premier robot capable d'évoluer de façon autonome en extérieur. Le chariot motorisé utilisait un système mécanique monté sur pivot qui permettait à une caméra de télévision d'obtenir plusieurs vues sans que le robot ait à se déplacer. La planification de la trajectoire était calculée à partir du premier détecteur de points d'intérêt invariants. Le robot se déplaçait d'un mètre toutes les 10 à 15 min, c'était le temps nécessaire au traitement de l'image pour planifier la trajectoire. C'est dans ce même élan que les chercheurs vont utiliser un système de vision binoculaire pour l'évitement d'obstacle. La vision par stéréoscopie (du grec *stéreo* : solide et *scopie* : vision), est l'ensemble des techniques mises en oeuvre pour reproduire une perception du relief à partir de plusieurs images planes. Habituellement utilisée à partir de seulement deux vues, la stéréoscopie binoculaire permet au robot de lui donner une perception visuelle en trois dimensions. La stéréoscopie est une technique qui remonte à l'invention de la photographie et du cinéma. Le premier système à vision croisée de Charles Wheatstone, le « stéréoscope », date de 1838. Il permettait à partir d'une photographie gauche et droite de voir la scène en relief, en affectant par un jeu de miroirs l'image correspondante à chaque oeil. Les premiers films en 3D anaglyphe ont été inventés par William Friese-Greene en 1890 pour le procédé et par Frederick Eugene pour la première caméra à deux lentilles en 1900. Bien des années après le « stéréoscope », la vision stéréoscopique binoculaire devient un outil important en robotique. Maintenant la perception visuelle et la commande fusionnent en une symbiose totale. Contrairement à ce que l'on pense la vision en robotique industrielle est apparue très tard. Effectivement la force de la vision est d'adapter le modèle d'environnement pour la commande de tâche. Dans le domaine industriel les tâches sont répétitives et l'environnement est très bien connu. La vision en milieu industriel se limite bien souvent à des systèmes de reconnaissance de formes, pour du triage de pièces en vrac, et de traitement de l'image, en contrôle qualité. C'est encore rare de trouver un asservissement visuel ou une reconstruction 3D, pour deux raisons. La première raison est que les méthodes ne sont pas encore assez fiables et relèvent seulement du domaine de la recherche. La deuxième raison est que, bien qu'il y ait une évolution de la puissance de calcul des ordinateurs, certains traitements sont encore trop lourds pour une cadence effrénée de production. Et pourtant plus que jamais, pour gagner en flexibilité, rapidité et en complexité des tâches, on ressent une demande importante de la vision dans l'industrie.

Depuis l'invention de la première caméra, la vision n'a eu de cesse de monter en puissance. Elle ne l'aurait jamais pu sans l'évolution en parallèle, technologique et scientifique, de la puissance de calcul numérique des ordinateurs. A l'époque des premiers traitements de l'image des années 60, le seul frein à l'avancement des chercheurs était la puissance de leurs calculateurs. Ce qui est surprenant, c'est qu'à notre époque, malgré l'avancée technologique, nous avons encore le même verrou. Notre premier frein reste la puissance de calcul. Certes, la complexité de nos algorithmes, la quantité des données à traiter ont augmenté par rapport au années 60 mais le problème reste le même. Nous sommes toujours tributaires du coût financier et des limites technologiques du calcul numérique. Chacun y va de sa solution, grappe de pc, développement sur DSP (*Digital Signal Processing*) et FPGA (*Field Programmable Gate Array*). En prenant du

recul, cette barrière technologique a apporté durant toutes ces années, une dynamique à la recherche. En effet elle nous a obligé, nous chercheurs à trouver sans cesse les algorithmes et les méthodes qui permettent d'avoir le meilleur résultat avec le moins de calcul possible. Cette limite n'est qu'une motivation pour trouver l'algorithme ultime.

1.4 De la macroscopie à la microscopie

L'homme est un explorateur par nature, de la découverte de l'Amérique par Christophe Colomb, aux premiers pas sur la lune de Neil Amstrong, il a toujours poussé ses limites pour aller poser le pied là où il ne s'était pas encore aventuré. De l'infiniment grand à l'infiniment petit, tout n'est que prétexte à la découverte de mondes nouveaux. L'infiniment petit, nouveau terrain de jeux, pour une conquête aventurière, ou comment l'homme veut mettre son nez là où il ne pourra jamais mettre le pied. Cette curiosité nous anime et nous pousse à découvrir ce qu'il y a, au plus profond de la matière, connaître le constituant de celle-ci. Cette réflexion, les savants de la Grèce antique se la posaient déjà. Ils en étaient arrivés à la conclusion qu'à un moment donné dans l'infiniment petit notre matière était constitué d'éléments indivisibles, qu'on ne peut plus couper. Ils avaient appelé cet élément, l'atome (du grec *a* : pas et *tome* : couper). Bien entendu, à cette époque, impossible de connaître la vérité, impossible d'aller voir ce qui se passe à cette échelle pour confirmer. Depuis ce temps, l'homme a cherché à se constituer des outils pour répondre à ces questions, de savoir, à quoi ressemble l'infiniment petit, l'atome peut-on le voir ?

L'ensemble des techniques et matériels pour produire une image visible d'un objet ou d'un détail trop petit pour l'œil humain, par des moyens d'agrandissement s'appelle la microscopie (du grec *micro* : petit et *scopie* : vision). L'outil le plus ancien est sans conteste le microscope photonique, plus couramment appelé microscope optique. C'est en 1590, alors âgé de 10 ans que Zacharias Janssen inventa le premier microscope photonique. Fortement aidé par son père Hans, il constitua un ensemble de lentilles optiques, destiné à un travail de classe. Tous deux s'aperçoivent que cet ensemble optique permettait de grossir les objets avec un agrandissement plus important qu'une simple lentille (loupe).

Il faut attendre 1664 pour que Robert Hooke, publie son livre « *Micrographia* », qui contient de nombreuses observations réalisées à partir de ce qui sera le premier microscope photonique moderne, constitué d'un oculaire, d'une lentille de champ et d'un objectif (Figure. 1.8). On lui at-

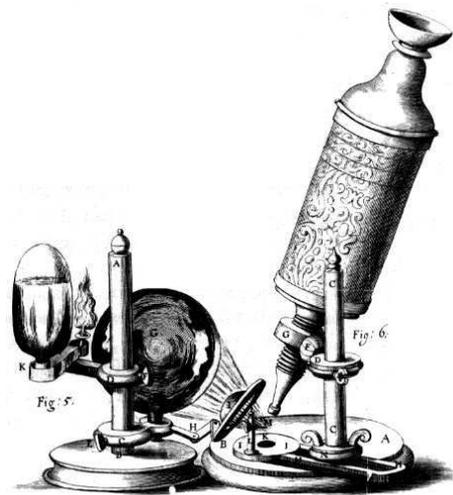


FIG. 1.8 – Le microscope de Hooke, 1664.

tribue aussi la première description d'une cellule biologique faite à partir de l'observation de végétaux. C'est le premier pas de l'homme dans le micro monde, nombreux scientifiques et biologistes vont utiliser cet instrument pour l'étudier. Le microscope photonique va engendrer deux nouveaux moyens photoniques d'observation. Le premier très proche de l'instrument actuel, est le microscope confocal, décrit par Marvin Minsky en 1953. Il connaîtra un succès plus tard dans les années 1980 par le biais de l'informatique. Le plan focal de l'objectif, qui a une profondeur de champ très faible, se déplace à différents niveaux de profondeur dans l'échantillon. La surface peut être reconstruite en 3D via les images prises. Le deuxième instrument émanant du microscope photonique est le microscope photonique en champ proche. Ce microscope proposé par Edward Hutchinson Synge en 1928 est appliqué pour la première fois par D. W. Pohl en 1984. Pour briser la limite du microscope photonique en partie liée à la diffraction, le microscope photonique à champ proche va collecter la lumière très proche de la surface de l'échantillon. Une fibre optique est utilisée pour envoyer et récupérer la lumière. Le détail de l'objet observé peut être plus petit que la longueur d'onde de la lumière car non diffractée.

Les dernières innovations sont apparues dans les années 1980, le microscope photonique ne suffisant pas, car limité par une résolution théorique de $0,2 \mu m$, en adéquation avec le critère de Rayleigh et de la plus petite longueur d'onde du spectre visible. Une solution dans les années 1960 consista à utiliser une source à rayons X qui a une longueur d'onde plus courte que le spectre visible, et donc une plus grande résolution qu'un microscope photonique classique. Il fallait aller encore de plus en plus loin dans la qualité et la résolution de l'imagerie microscopique. C'est en 1931 qu'une vraie révolution en imagerie fait son apparition avec Max Knoll et Ernst Ruska, d'origine allemande, récompensés par un prix Nobel de physique en 1986 : le microscope électronique en transmission (MET). Le MET a pour principe d'utiliser des électrons comme sources de rayonnement à la place des photons. Cette fois-ci la limite de résolution de ce microscope est régi par la longueur d'onde de De Broglie des électrons. Ce qui signifie que la résolution théorique idéale serait de l'ordre de grandeur du picomètre, en raison des fortes aberrations, elle n'est en réalité que de quelques ångströms ($1\text{Å} = 100$ picomètres).

Comme les premiers microscopes photoniques, l'échantillon est observé en transmission, c'est-à-dire par transparence, ce qui implique que l'épaisseur des échantillons doit être extrêmement fine. Le faisceau d'électrons est focalisé via des lentilles magnétiques et collecté pour obtenir les images (ou clichés) par film photographique. L'épaisseur très fine des échantillons limite les analyses du microscope électronique à transmission aux structures biologiques seulement. En 1935, Max Knoll, après avoir travaillé sur le MET, décrivait un principe de microscope électronique qui viendrait balayer, par un faisceau d'électrons plus fin, la surface de l'échantillon. Manfred von Ardenne, scientifique Allemand, créa en 1938 le premier microscope électronique à balayage par transmission (MEBT). Il donnait des images de 400 lignes par 400 lignes d'une résolution de 50 à 100 nm après une acquisition de 20 min. Son défaut reste toujours l'utilisation d'échantillons très fins. C'est Vladimir Zworykin, inventeur de la première caméra électronique, qui publia en 1942 les détails du premier microscope électronique à balayage (MEB) pouvant analyser une surface opaque. Le MEBT et le MEB ne suscitent

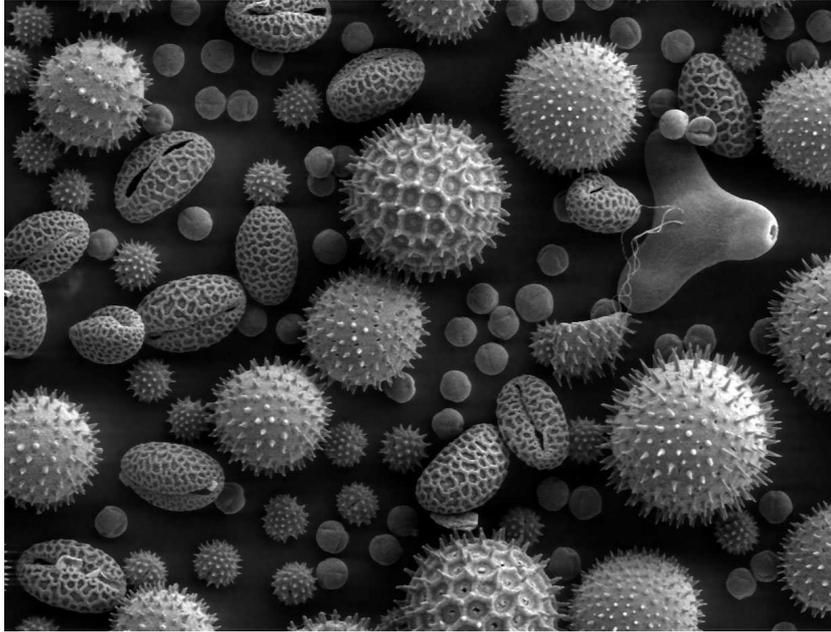


FIG. 1.9 – Image provenant d'un MEB, observation de différents pollens de plantes.

pas beaucoup d'intérêt par rapport à son grand frère le MET, qui affiche des résolutions exceptionnelles. Malgré un ralentissement dans le développement, Dennis McMullan et Charles Oatley construisirent le premier MEB moderne, et en 1952, l'instrument rendait des images de grande qualité et un effet « relief » qui caractérise nos MEB modernes (Figure. 1.9). Une nouvelle gamme de microscopes va apparaître dans les années 80, elle découle de l'idée du MEB, à savoir de scanner la surface de l'échantillon. Les microscopes à balayage par sonde (MBS), utilisent non plus un flux de particules mais une sonde. Le premier type de microscope à sonde est le microscope à effet tunnel. Inventé en 1981 par deux chercheurs d'IBM, Gerd Binnig et Heinrich Rohrer. Ils reçoivent le prix Nobel de physique pour cet instrument basé sur l'effet tunnel en 1986. Une pointe conductrice est placée au dessus de la surface de l'échantillon à balayer. La hauteur de la pointe est asservie en fonction du courant mesuré entre la pointe et l'échantillon. Le courant résulte du passage d'électrons par effet tunnel. Cet effet désigne la propriété qu'a une particule quantique de traverser une barrière de potentiel, qui est impossible en physique classique. Il existe une probabilité non nulle pour qu'un électron passe de l'autre côté de la barrière de potentiel. La hauteur de la pointe pendant le balayage est asservie pour que le courant du tunnel soit constant. On vient effleurer la

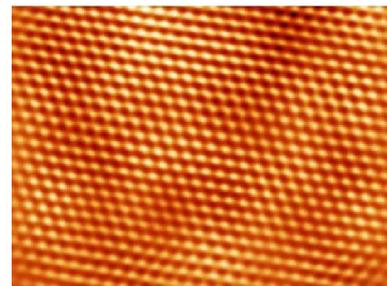


FIG. 1.10 – Arrangement hexagonal d'atomes d'or, vue par un microscope à effet tunnel.

surface suffisamment près pour détecter un courant sans pour autant la toucher. On peut alors déterminer le profil de la surface avec une précision inférieure aux distances interatomiques (inférieur à la taille d'un atome). Le microscope à effet tunnel permet également de manipuler par capture des atomes. En 1990, des chercheurs d'IBM ont pu écrire les premières lettres de l'histoire des nanotechnologies en disposant 35 atomes de xénon qui dessinaient les trois lettres IBM. De la Grèce antique à l'invention du microscope à effet tunnel, il a fallu attendre près de 24 siècles pour que l'homme inventa le premier instrument capable de rendre « visible » l'atome (Figure. 1.10). Une autre technologie fondée sur le même principe apparaît en 1986. De nouveau Gerd Binnig, Calvin Quate et Christoph Gerber proposent d'utiliser une poutre munie d'une pointe à son extrémité pour topographier la surface de l'échantillon à analyser. Un couple laser/photo-diode mesure la déflexion de la poutre. Cette fois-ci ce n'est pas le courant de l'effet tunnel qui est mesuré mais les forces atomiques à la surface de l'échantillon. Tout comme le microscope à effet tunnel, la poutre est asservie en hauteur pour avoir une force mesurée constante, d'où son nom, de microscope à force atomique (MFA). Il permet d'obtenir des résolutions pouvant atteindre la fraction d'ångström. Il est toutefois moins bon en résolution que le microscope à effet tunnel. Par contre il est beaucoup moins onéreux et permet de mesurer tout un panel de caractéristiques via différentes interactions de forces sur la surface de l'échantillon (Van der Waals, capillaire, liaison chimique, électrostatique, magnétique, Casimir, etc).

1.5 Conclusion

Quel que soit le microscope utilisé la vision a toujours su accompagner son évolution. Le traitement de l'information numérique provenant de ces instruments est une source intarissable pour les chercheurs. Elle intervient dans différents domaines tels que la médecine, la biologie, les microsystèmes, la micromanipulation, etc. Le domaine qui nous intéresse plus particulièrement est la micromanipulation. Les deux thèmes présentés dans l'historique, vision et microscopie peuvent fusionner en un seul : la microvision, c'est-à-dire la vision appliquée à des sources d'images permettant l'observation de scènes qui ne peuvent pas être vues à l'oeil nu. Dans le prochain chapitre nous allons nous focaliser sur les systèmes de vision dédiés à la micromanipulation.

Systemes de Vision en Micromanipulation, du Réel au Virtuel

Nous présentons dans ce chapitre les différentes solutions de systèmes de vision employés actuellement dans les stations de micromanipulation. Nous listerons leurs principales spécificités et contraintes. Nous présenterons également les méthodes de rendu de vues 2D qui relèvent de la vision et qui permettent de contrecarrer les limites précédemment citées. Nous concluons avec le choix de deux méthodes qui seront développées dans les prochains chapitres.

2.1 Introduction

La résolution limite de $0,2 \mu\text{m}$ d'un microscope photonique en fait un outil plus que nécessaire en micromanipulation et en microassemblage. Il est le seul dispositif optique passif associé à un capteur capable à cette échelle ($1 \mu\text{m}$ à 1mm) de percevoir de l'extérieur l'espace de travail avec précision, c'est-à-dire sans influencer l'environnement de travail. On le retrouve dans presque toutes les stations de micromanipulation. Nous en listerons quelques unes dans ce chapitre. Le microscope photonique est un outil à double tranchant. Sa spécificité si particulière d'agrandir la scène observée par un système de lentilles engendre cinq spécificités dues à l'optique : un encombrement important, un faible champ de vision, une faible profondeur de champ, une faible distance de travail et une forte dépendance à l'éclairage. Chaque spécificité sera développée et illustrée dans ce chapitre. La conséquence directe de ces spécificités est l'utilisation de systèmes de vision distribués avec des sources d'images dont les caractéristiques sont différentes et complémentaires. Par exemple pour pallier un champ de vision réduit, une méthode consiste à utiliser un microscope photonique pour une vue locale (grande résolution mais faible champ de vision) et une caméra avec un objectif classique pour une vue globale (faible résolution mais grand champ de vision). Seulement, la faible distance de travail et l'encombrement des microscopes photoniques ne permettent pas d'ajouter n sources d'images (microscopes photoniques, caméras, etc.) afin d'avoir les n vues in-

dispensables au fonctionnement de la station de micromanipulation. C'est pour cela que notre approche consiste à utiliser un système de vision distribué mais avec un nombre réduit de sources d'images aux caractéristiques identiques. Ce qui permet également de limiter la diversité du matériel pour une standardisation moins onéreuse. Certaines vues indispensables au contrôle de la station seront obtenues par synthèse de vues à partir d'images réelles de microscope photonique. La synthèse de ces nouvelles vues peut être assimilée à des microscopes photoniques virtuels dont on reconfigure les caractéristiques à notre avantage. Une grande partie de ce chapitre est consacré aux méthodes de synthèse de vues, afin d'en présenter le large panel. Nous discuterons pourquoi parmi toutes ces méthodes nous en avons sélectionné deux et comment elles pourront être utilisées en micromanipulation. Dans l'immédiat, nous allons présenter des exemples d'utilisation de sources d'images et de systèmes de vision dans les stations de micromanipulation. Cela nous permettra d'identifier la principale source d'images et son architecture.

2.2 Les systèmes d'imagerie et de vision dans les stations de micromanipulation

Dans cette section nous présentons quelques articles tirés de la littérature permettant de lister le matériel de perception visuelle dans les stations de micromanipulation en nous focalisant plus particulièrement sur les systèmes de micro-assemblage. La liste d'articles est non exhaustive et ne se prétend pas un état de l'art en micromanipulation. Nous avons sélectionné des papiers représentatifs qui faisaient référence directement ou indirectement à de la vision en contexte de micro-assemblage. Le but n'est pas de connaître tous les systèmes de vision en micromanipulation de la littérature, mais seulement de faire ressortir les types de source d'images et les méthodes de vision associées.

Aux USA, à l'Université du Minnesota, K. Yesin et B. Nelson (2002), utilisent deux microscopes photoniques montés en stéréoscopie. A partir des microscopes photoniques et du modèle CAO des objets, ils calculent la position dans l'espace des objets à assembler. L'équipe de cette université en collaboration avec Nelson actuellement à l'École Polytechnique Fédérale de Zurich (ETHZ) en Suisse, a développé une station de microassemblage très aboutie. Ge Yang, James Gaines et Bradley Nelson (2003), utilisent deux vues de côté et une vue de dessus par microscope photonique et une caméra avec un objectif standard placé au dessus de la station pour une vue d'ensemble de la station. Les auteurs présentent le fonctionnement de toutes les étapes de microassemblage de leur station. Quelques années plus tard, G. Yang, J. Gaines et B. Nelson (2005) montrent dans leur étude, l'impact de l'utilisation du microscope photonique. Ils utilisent une méthode d'autofocus basée sur les ondelettes pour toujours obtenir des images nettes. Ils se heurtent aux différents problèmes liés à la microscopie photonique et en dressent la liste. Benjamin Potsaid, Yves Bellouard et John Wen (2002) et B. Potsaid, Y. Bellouard et J. Wen (2005), de l'Institut Polytechnique Rensselaer de Troy, s'attaquent à un de ces problèmes. Ils proposent de piloter l'optique d'un microscope photonique pour augmenter son champ de vision. Cette modification leur permet de balayer la scène observée. Les im-

ages collectées sont mises bout à bout pour donner une image mosaïque de grande taille. Ce nouveau microscope photonique permet d'avoir un plus grand champ de vision que les microscopes conventionnels. A l'Université de Illinois à Chicago, B. Vikramaditya et B. Nelson (1997) présentent la première méthode d'asservissement visuel avec un microscope photonique. Le système ne comporte qu'une vue de dessus. Quelques années après, B. Vikramaditya et B. Nelson (1999), cette fois-ci à l'Université du Minnesota, présentent un asservissement visuel plus abouti. La vue de dessus est obtenue par un microscope photonique, le contrôle est assuré par une carte à base de DSP (*Digital Signal Processing*). Par la suite, Y. Zhou, B. Nelson et B. Vikramaditya (2000) enrichissent le système avec une vue de côté fournie par une autre source à base de microscope photonique. Dans l'article de S. Ralis, B. Nelson et B. Vikramaditya (2000), ils testent différentes vues de côté, dont une source à basse résolution composée d'un module de caméra bas prix, placé à 45° pour la gestion des occultations pendant l'asservissement en micromanipulation. Au Laboratoire National Sandia, J. Feddema et R. Simon (1998) proposent d'utiliser un asservissement visuel avec une seule vue sous microscope photonique. La consigne qui est l'image de référence de l'asservissement est récupérée par image de synthèse à partir du modèle CAO (Conception Assistée par Ordinateur). A l'Université de l'Arizona, E. Enikov, S. Clark et L. Minkov (2003) utilisent quand à eux une seule vue de dessus par une caméra avec un objectif zoom pour piloter l'asservissement visuel d'une station de microassemblage.

En Grandre Bretagne, à l'Université de Sheffield Hallam, B. Amavasai, S. Meikle et R. Yates (2000) travaillent à la fois avec des microscopes photoniques et un microscope électronique à balayage. Dans cet article, on retrouve différentes techniques de traitement de l'image, contours actifs pour de la reconnaissance de cellules biologiques et suivi d'objets mécaniques par détecteur d'arêtes du type Haynes et Jain (S. Haynes et R. Jain 1983). Dans la même Université, Jan Wedekind (2004), propose d'utiliser la spécificité de la faible profondeur de champ du microscope photonique. Par une méthode de reconstruction $2D$, il effectue une tomographie par le déplacement de la mise au point du microscope photonique. L'assemblage des images des différents plans lui permet de reconstruire la surface des objets observés. Cet technique de reconstruction s'appelle le *depth-from-focus*.

A la fondation Robotiker en Espagne, H. Hötendorfer et al. (2006) utilisent un microscope photonique en vue dessus et deux caméras avec objectifs montées en stéréoscopie. Ils utilisent ces dernières pour déterminer par triangulation stéréoscopique la position dans l'espace de l'effecteur d'un mini-robot qui sera utilisé pour la micromanipulation.

Au Laboratoire d'Automatique de Besançon en France, nos stations de micro-assemblage (SAMMI et PRONOMIA) sont équipées également de plusieurs sources d'images. La station SAMMI est destinée à la manipulation d'objets en milieu sec et est composée d'un stéréo microscope photonique (grandissement faible) pour la vue de dessus, une caméra avec un objectif de type zoom pour une vue latérale. La station PRONOMIA est destinée à la manipulation d'objets en milieu liquide et est composée d'un microscope

photonique (fort grandissement) pour la vue de dessus, un microscope photonique (fort grandissement) pour une vue latérale. En fonction des applications nous avons un panel de matériels divers et variés : microscope électronique à balayage, stéréo microscope photonique, microscope inversé, mini-microscope photonique (TIMM), caméra avec des objectifs de type zoom, caméra bas prix, fibroscope et mini-caméra endoscopique (Figure 2.1).



FIG. 2.1 – De gauche à droite : fibroscope ($\phi 1\text{ mm}$) ; mini-microscope TIMM ($155\text{ mm} \times \phi 22\text{ mm}$) ; mini-caméra endoscopique ($20\text{ mm} \times \phi 5\text{ mm}$).

A l'Université de Karlsruhe, en Allemagne, S. Fatikow, A. Buerkle et J. Seyfried (1999), utilisent une vue globale par une caméra avec un objectif standard, pour localiser et placer le robot de façon grossière. Pour placer le préhenseur de manière fine, ils utilisent une triangulation par laser observé par un microscope photonique placé au dessus. Un an plus tard, S. Fatikow et al. (2000), détaillent les différentes étapes qui composent la micromanipulation en incluant cette fois-ci un microscope électronique à balayage dans le système. T. Sievers et S. Fatikow (2005), quelques années plus tard à l'Université d'Oldenburg, proposent d'utiliser pleinement le microscope électronique à balayage (MEB). La seule vue du MEB n'est pas suffisante dans la tâche d'asservissement visuel, c'est pourquoi ils ajoutent deux caméras de faibles dimensions à bas coût. L'estimation de la position de l'effecteur est obtenue par une méthode de corrélation à partir de formes pré-enregistrées. J. Schilp et al. (2003), de l'Université Technologique de Munich, présentent un système de micro-assemblage à partir d'un système de vision composé de caméras avec des objectifs de type zoom. Le système fournit une vue de dessus et d'une vue de côté. On trouve également du matériel moins conventionnel, à l'Institut de Technologie de Production de Fraunhofer. M. Weck et C. Peschke (2004) utilisent un fibroscope monté entre les doigts d'une pince. Le fibroscope est un système de perception visuelle basé sur un faisceau de fibres optiques dont le diamètre varie de quelques centaines de microns à quelques millimètres (Figure 2.2). Chaque fibre, d'environ $5\ \mu\text{m}$ de diamètre retourne une information lumineuse équivalant à un pixel. L'ensemble des fibres permet de composer une image. Quelques fibres à l'intérieur sont prévues en émission pour l'éclairage. Le fibroscope par son aspect très maniable (rayon de courbure important) et son faible diamètre est très alléchant en terme de technologie car il permet de déporter la caméra dans une station de micromanipulation. Hélas, ce type de matériel a

une résolution qui n'excède pas les $50 \mu m/pixel$.

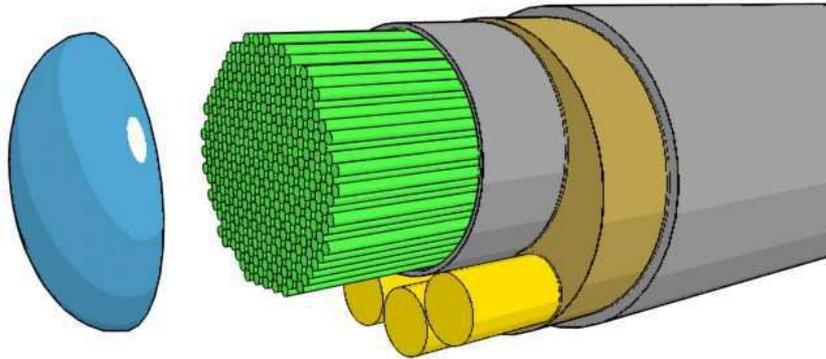


FIG. 2.2 – Vue éclatée d'un fibroscope (de gauche à droite) : lentille convergente ; ensemble de fibres en réception (collecte de l'image) ; les trois fibres en émission (éclairage) ; gainage.

Au Danemark, J. Glückstad et al. (2003), du Laboratoire National Riso, utilisent un microscope photonique inversé de forte puissance pour des applications de *laser trapping*.

A l'École Polytechnique Fédérale de Zurich (ETHZ) en Suisse, A. Codourey, M. Rodriguez et I. Pappas (1997) utilisent un stéréo microscope photonique. La stéréoscopie est utilisée pour la mesure de distance en profondeur de certains objets. Ces objets sont identifiés par corrélation de forme. Le système présenté n'est pas un système de micromanipulation automatisé mais télé-opéré.

Du côté asiatique, à l'Institut de Technologie d'Industrialisation de Singapour, Andrew Shacklock (2004) propose une navigation multi-échelle et multi-vue dans l'espace de travail d'un système de micromanipulation. Une méthode de suivi d'objet est utilisée dans les différentes vues pour contrôler l'ensemble des connecteurs d'un boîtier de puce électronique. Une méthode d'estimation de mouvement entre les vues est utilisée pour assembler ces différentes vues en une seule image, créant ainsi une mosaïque (A. Shacklock et W. Sun 2005).

En Chine à l'Institut de Technologie Harbin (HIT), H. Xie et al. (2005), proposent d'assembler un système mécanique de pignons d'une taille comprise entre $400 \mu m$ et $2 mm$. La station est automatisée en utilisant un contrôleur PID et un microscope photonique en vue de dessus. Le microscope photonique règle sa mise au point automatiquement avec un autofocus basé sur les ondelettes. Les auteurs proposent également de reconstruire la surface de travail avec le microscope photonique en utilisant une méthode de *depth-from-focus* (H. Xie, W. Rong et L. Sun 2006). L. Sun et al. (2005a), rajoute à la même station de micro-assemblage des fonctions de localisation 2D des pièces à

assembler, pour augmenter la flexibilité des tâches d'assemblage.

A l'Université de Tokyo, au Japon, T. Kasaya et al. (1999), utilisent un système de vision composé d'un microscope électronique à balayage pour la vue de dessus et d'un microscope photonique pour la vue de côté. Ils travaillent avec différents types de traitement d'images simples, détection d'arêtes, filtrage, transformée de Hough, etc. Le tout est exploité pour l'automatisation de tâches de prise et dépose (*pick-and-place*).

A l'Institut de Science et de Technologie de Corée (KIST), Seok Lee et al. (2001), et K. Kim et al. (2002) utilisent des méthodes de mise en correspondance de formes par corrélation et d'autofocus dans leur manipulation avec un asservissement visuel de type *look-and-move*. Ils utilisent un microscope photonique stéréoscopique Leica. Par contre, l'information stéréoscopique n'est pas exploitée, le traitement n'est fait que sur une des vues de dessus. Les méthodes qu'ils présentent sont développées pour des tâches simples de micromanipulation. Ce sont quelques années plus tard, que B. Kim et al. (2006) présentent la supervision des différentes étapes de microassemblage de composants optoélectroniques. Pour cela ils utilisent en plus du stéréo microscope photonique, une vue globale à faible grandissement et une vue de coté locale à fort grandissement par microscope photonique.

Dans toutes ces applications de micro-assemblage de pièces mécaniques nous pouvons citer également quelques systèmes dédiés à la micromanipulation d'objets biologiques. S. Yu et B. Nelson (2001), de l'Université du Minnesota aux USA, utilisent un microscope photonique inversé pour des applications d'asservissement visuel en biologie, plus particulièrement dans l'automatisation d'injection de matériel génétique dans des cellules. A l'Université d'Okayama au Japon, T. Sano et H. Yamamoto (2004) proposent d'utiliser un stéréo microscope photonique dans une application biomédicale. La pose de micro-électrodes dans une zone précise du cerveau d'un rat est calculée dans l'espace par le système de stéréovision. La détermination de la zone d'insertion est localisée par des méthodes de segmentation, de contour actif par algorithme génétique et de corrélation. Également au Japon, à l'Université d'Osaka, Y. Katoh, T. Arai et K. Inoue (2001), dans le cadre de la micromanipulation de cellules biologiques, utilisent du traitement de l'image simple pour faire du suivi d'objet et de l'autofocus. L'application biologique ne nécessite qu'une seule vue de dessus via un microscope photonique.

Notons que dans les papiers où une seule tâche est traitée le système de vision comporte généralement un seul microscope photonique. Par contre, lorsque la station augmente en complexité, c'est-à-dire que le nombre de tâches est plus élevé, la supervision est assurée par un système de vision plus étoffé. On retrouve des sources d'images fournissant plusieurs types de vue : des vues locales haute résolution, des vues globales basse résolution, des vues de dessus, des vues de côté, des vues à 45° , des vues en stéréoscopie, des vues de dessous. Les sources d'images correspondantes comportent différents type de matériels : microscope électronique à balayage, microscope photonique, fibroscope, caméra avec objectif standard ou de type zoom, caméra bas prix. Le choix du système

de vision dépend de deux critères, le premier est le niveau de complexité du micro-assemblage car il définira le nombre de vues nécessaire. Le deuxième critère définira le type de matériel nécessaire et dépendra directement de la taille des objets du micro-assemblage. Si un objet fait $800 \mu m$ ou $20 \mu m$, la résolution du matériel de vision sera très différente. T. Sato et al. (1995) propose un classement du matériel de perception visuelle en fonction de leur résolution, et le compare avec les possibilités dimensionnelles pour la fabrication et la manipulation de micro-objets. Nous présentons à la Figure 2.3, une modification d'une partie que présente T. Sato et al. (1995), en ajoutant l'ensemble du matériel présenté dans cette section. Les résolutions du matériel de la Figure 2.3 sont définies en fonction de la plage d'utilisation. Si un microscope photonique peut atteindre de très faibles résolutions théoriques (plusieurs centimètres par pixel), en expérimentation il ne sera jamais utilisé dans cette plage, une caméra avec un objectif standard fera l'affaire.

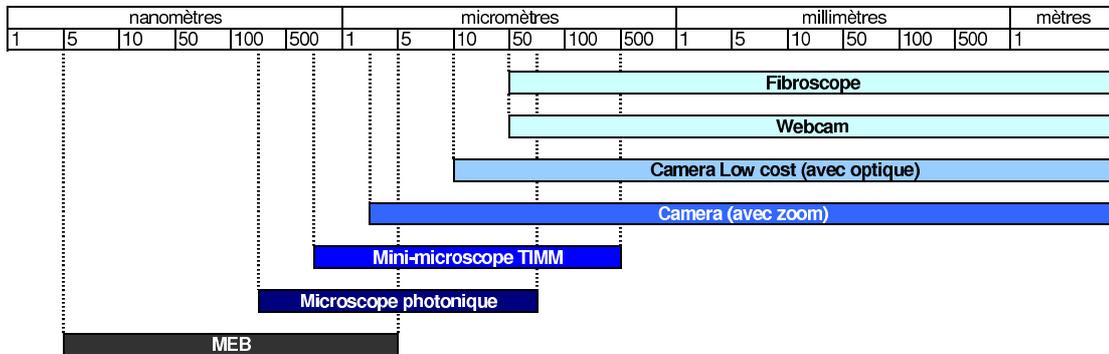


FIG. 2.3 – Graphique comparatif de la résolution des différents systèmes de perception visuelle en fonction de leur zone d'utilisation en micromanipulation.

Ces différents matériels, par leurs diversités, montrent l'importance de prévoir l'intégration, en terme d'espace du système de vision dans les stations de micromanipulation. Paradoxalement, le matériel qui a une résolution importante a un encombrement important et inversement. Par exemple, le fibroscope et la mini-caméra semblent être des systèmes de perception visuelle totalement adaptés à l'environnement du concept de micro-usine. Par contre leur résolution n'est pas en dessous des $50 \mu m/pixel$, ce qui n'est pas suffisant en micromanipulation. Ce type de matériel, n'est employé que pour de la supervision globale de l'espace de travail. Les différentes caractéristiques des matériels listés dans cette étude, nous permettent de mettre en évidence une particularité technique de la conception de système de vision dans les stations de micromanipulation : le dilemme encombrement/résolution, *pour une source d'images, l'ordre de grandeur de sa résolution varie inversement de l'ordre de grandeur de son encombrement*. Plus la résolution du système est grande, plus l'encombrement de ce système est important. Par exemple, le fibroscope à un encombrement très faible avec une utilisation très flexible, par contre le MEB (Microscope Électronique à Balayage) est un système imposant très contraignant dans son utilisation, mais avec une résolution exceptionnelle. Entre ces deux

extrêmes nous retrouvons un compromis, qui est le microscope photonique. Ce graphique souligne pourquoi le système de perception visuelle le plus utilisé dans les stations de micromanipulation est le microscope photonique.

2.3 Propriétés d'un microscope photonique

La section précédente a montré l'importance du microscope photonique conventionnel en micro-assemblage. Regardons de plus près son fonctionnement et surtout ses propriétés. Le microscope photonique classique de base, est formé d'un objectif et d'un oculaire assimilés à des lentilles minces convergentes de même axe (Figure 2.4).

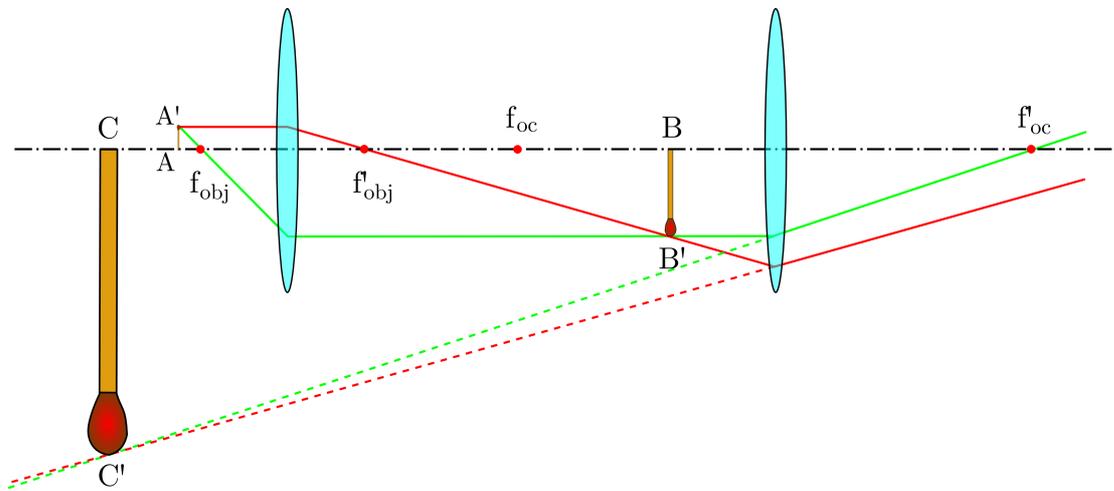


FIG. 2.4 – Montage optique d'un microscope de base, constitué d'un objectif à gauche et d'un oculaire à droite.

L'objectif a une distance focale très petite (f_{obj}), il est placé près de l'objet à observer (AA'), fournissant une image réelle renversée fortement agrandie (BB'). Cette dernière est encore plus agrandie par l'oculaire qui fonctionne comme une loupe, car l'image réelle formée est à l'intérieur de la distance focale de l'oculaire (f_{oc}). L'objet observé est en fait une image virtuelle renversée (CC'). Au final, si l'observateur se place à la distance minimale de vision nette a_0 , c'est-à-dire à une distance d'environ 25 cm (pour l'oeil humain) entre l'image virtuel (CC') et l'oeil de l'observateur, il perçoit l'objet avec un fort grossissement. Le montage optique présenté ici, est un montage de base, pour améliorer l'observation, les microscopes modernes ont tous un jeu de lentilles plus important qui permet de diminuer les déformations géométriques et les aberrations chromatiques. Dans le cas d'un vidéo microscope photonique le montage optique est différent. En effet une caméra ne peut pas percevoir l'image virtuelle CC' . Par conséquent l'oculaire n'est pas utilisé, permettant l'image réelle BB' de se former sur le capteur. Ces caractéristiques n'en restent pas moins inchangées par rapport à un microscope photonique de base. Toutes les caractéristiques du microscope photonique découlent de

la particularité du montage objectif et oculaire. Les propriétés de l'optique géométrique et physique façonnent les contraintes et les performances liées à la microscopie photonique.

La performance recherchée dans un microscope photonique provient de ses propriétés, un fort pouvoir de résolution et un fort grandissement. Toutes les autres propriétés, dans un contexte de micro-assemblage, sont apparentées à des contraintes, que ce soit l'encombrement, la faible distance de mise au point, la faible profondeur de champ, le faible champ de vision ou encore la forte dépendance à l'illumination. Toutes ces caractéristiques augmentent la difficulté de commander des tâches de micromanipulation à partir d'un microscope photonique. Pourtant cet outil est utilisé et maîtrisé depuis longtemps dans des applications de micromanipulation biologique. Prenons l'exemple des méthodes de fécondation *in-vitro*, la pipette d'injection et l'ovocyte sont dans le même plan. L'utilisateur règle et contraint la pipette sur l'axe Z , permettant au manipulateur de travailler seulement sur le plan xOy . Le succès de la micromanipulation d'objets biologiques est basé sur le fait que tous les éléments sont placés sur un même plan, que ce soient les objets ou les outils de micromanipulation. Ce qui est, dans le cas du micro-assemblage, totalement différent. Le but ici est de faire évoluer les objets afin de les assembler dans l'espace. Typiquement en micro-assemblage on ajoute une dimension à l'espace de travail sous le microscope photonique, on passe d'un plan à un volume. Est-ce que cela change vraiment la stratégie de micromanipulation? En quoi toutes les contraintes dues à ces propriétés vont intervenir dans la gestion d'une micromanipulation dans l'espace. Pour répondre à ces questions, nous allons présenter les différentes propriétés d'un microscope photonique. On trouve leurs descriptions dans plusieurs ouvrages d'optique notamment dans (E. Slayter et H. Slayter 1992), (L. Dettwiller 1997) et (C. Cibert 2005). Ceci nous permettra de comprendre les répercussions directes en micro-assemblage.

2.3.1 Encombrement important

En microscopie, on recherche toujours le grandissement le plus adapté. Il doit être suffisamment petit pour observer la totalité de l'échantillon et suffisamment grand pour obtenir le maximum de détails de cet échantillon. Le grandissement du microscope s'exprime de la façon suivante :

$$\Gamma_{mic} = \frac{l}{f_{obj}} \left(1 + \frac{a_0}{f_{oc}} \right) \quad (2.1)$$

avec Γ_{mic} le grandissement du microscope, l la longueur du tube qui correspond à la distance entre les foyers de l'objectif et de l'oculaire, f_{obj} la distance focale de l'objectif, f_{oc} la distance focale de l'oculaire et a_0 la distance minimale de vision nette. Cette expression permet de mettre en valeur une relation très importante : le grandissement du microscope est proportionnel à la longueur du tube. C'est pourquoi, il est impossible physiquement d'avoir un microscope avec un grandissement important et une taille réduite. Par sa définition un microscope à fort grandissement est forcément imposant.

Contrainte en micro-assemblage La première propriété, évoquée ici, est l'encombrement. Le microscope photonique est un matériel imposant, par rapport à la philoso-



FIG. 2.5 – Encombrement du système de vision sur la station de micromanipulation du groupe SAMMI.

phie de la micro-usine (système de production de taille convenable). Son encombrement représente environ 50% à 75% de l'espace d'une station de micromanipulation, comme dans le cas de la station du groupe SAMMI (Figure 2.5). Le système de vision de la station SAMMI est assuré par une binoculaire stéréoscopique LEICA pour la vue de dessus et un ensemble objectif/tube d'allonge/caméra pour la vue de côté. Le volume du système de vision est imposant par rapport au reste du matériel. La conséquence directe est l'explosion du volume des stations de micromanipulation. Combiné à une distance de travail très faible, le système de vision est une limitation à l'accès à l'espace de travail. Dans la plupart des stations la limitation de l'encombrement n'est pas une fin en soi, mais elle remet en cause la définition de la micro-usine sensé avoir une taille de station raisonnable.

2.3.2 Faible distance de mise au point

La distance de mise au point correspond à la distance maximale acceptable entre l'objet observé et la première lentille de l'objectif, appelée aussi la distance de travail du microscope photonique. Si cette distance n'est pas respectée, la mise au point du microscope photonique n'est pas garantie. Ce qui rend impossible une observation nette de la scène. Cette distance est directement liée à l'ouverture numérique du microscope :

$$d_t = \frac{\lambda}{ON \sin(\alpha_o)} = \frac{\lambda}{n \sin^2(\alpha_o)} \quad (2.2)$$

avec d_t la distance de travail, ON l'ouverture numérique, c'est-à-dire le cône d'acceptance de l'objectif (c'est elle qui régit la quantité de lumière entrant dans l'instrument), λ la longueur d'onde de l'illumination, n l'indice de réfraction de l'objectif et α_o le demi

angle d'ouverture de l'objectif. Pour avoir une résolution de qualité avec un grandissement maximal, il faut collecter le plus de lumière possible. Cela implique que plus le grandissement est grand et plus l'ouverture numérique doit être importante. La limite de l'ouverture numérique tend vers $\frac{\pi}{2}$, le seul paramètre restant pour augmenter la collecte de lumière, c'est une distance de travail très courte. La Figure 2.6 représente une illustration de cette contrainte, l'objectif du microscope photonique doit très près de l'échantillon.



FIG. 2.6 – Microscope photonique à faible distance de travail.

Contrainte en micro-assemblage La distance de mise au point pose un sérieux problème en micro-assemblage. Un micro-assemblage classique se fait généralement par l'espace de travail supérieur (par le dessus des objets). Si la distance objectif-objet est trop petite, le préhenseur ne pourra accéder correctement à l'espace de travail. De plus on ne pourra pas ajouter plusieurs vues à la station, notamment les vues de côté. Il est possible d'utiliser des objectifs spéciaux pour augmenter cette distance. Les objectifs longue distance de travail permettent de miniser la contrainte de la distance de mise au point par contre, c'est toujours au détriment d'une autre propriété du microscope photonique et en premier lieu celle de la profondeur de champ.

2.3.3 Faible Profondeur de champ

La profondeur de champ est la distance sur laquelle on peut déplacer un objet en gardant une image nette sur le capteur. Le microscope photonique a une profondeur de champ très courte et ne permet pas au système de vision de percevoir l'intégralité de la profondeur de la scène. La profondeur de champ peut s'écrire de la façon suivante :

$$PdC = \frac{n \cdot \lambda}{ON^2} + \frac{n \cdot e}{\Gamma_{mic} \cdot ON} \quad (2.3)$$

où PdC est la profondeur de champ et e la résolution de la caméra. L'équation montre que plus le grandissement Γ_{mic} augmente et plus la profondeur de champ diminue. Si Γ_{mic} tend vers l'infini, la deuxième partie de l'équation de la profondeur de champ va tendre vers 0. Par contre la première partie de l'équation reste constante, ce qui indique que même si le microscope photonique a un grandissement très important, la profondeur de champ tend vers une constante. Par abus de langage, nous entendons souvent parler de plan focal d'un microscope photonique. C'est incorrect, la profondeur de champ ne tend jamais vers 0, le plan est en fait un volume. Même pour le microscope photonique confocal, la profondeur de champ n'est pas un plan parfait car elle est d'environ 600 nm d'épaisseur.

Contrainte en micro-assemblage En micro-assemblage la faible profondeur de champ rend impossible l'observation nette d'objets à différentes profondeurs dans l'espace de travail en même temps. Par exemple dans l'illustration proposée à la Figure 2.7, il est impossible d'obtenir en même temps une image nette des objets sur le plan de travail et une image nette de la micropinçe à une profondeur différente.

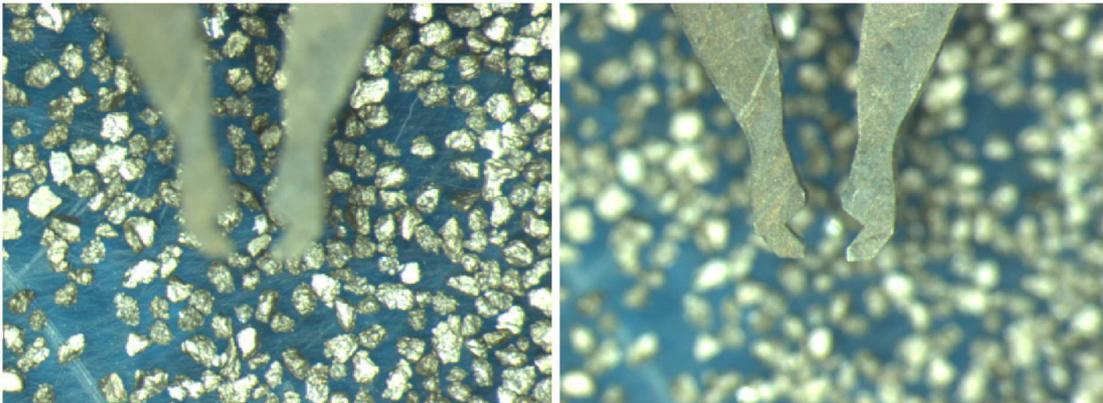


FIG. 2.7 – Illustration de la faible profondeur de champ d'un microscope photonique.

Pour observer tel ou tel objet dans l'espace de travail il faut modifier la mise au point. Dans certaines stations de micromanipulation on utilise une astuce qui consiste à basculer entre deux mises aux points différentes pré-réglées sur les objets que l'on désire observer, par exemple sur l'objet et sur le préhenseur. Par contre il faut être certain de la présence de ces objets dans les zones de mise au point pré-enregistrées. Si on rajoute la contrainte du faible champ de vision à celle-ci, on prend conscience de toute la difficulté de gérer un déplacement ou un suivi d'objet dans l'espace par microscopie.

2.3.4 Faible champ de vision

Le champ de vision est la partie de la scène vue par le système de perception. En microscopie, le champ de vision est également très faible et ne permet pas au système de vision de percevoir l'intégralité de l'espace de travail. Le champ de vision s'écrit :

$$CdV = \frac{T_{\text{capteur}}}{\Gamma_{\text{mic}}} \quad (2.4)$$

avec CdV le champ de vision et T_{capteur} la taille du capteur de la caméra. Plus le microscope a un grandissement important et plus le champ de vision est faible. Plus on désire une résolution importante de la scène en observation et plus on est obligé, en quelque sorte, de se rapprocher de cette scène. Mais comme l'ouverture numérique est constante, en se rapprochant le champ de vision diminue. Une illustration est présentée à la Figure 2.8, à gauche le champ de vision est important, la résolution est de $9 \mu\text{m}/\text{pix}$, par contre à droite, le champ de vision est plus faible, la résolution est de $3 \mu\text{m}/\text{pix}$.

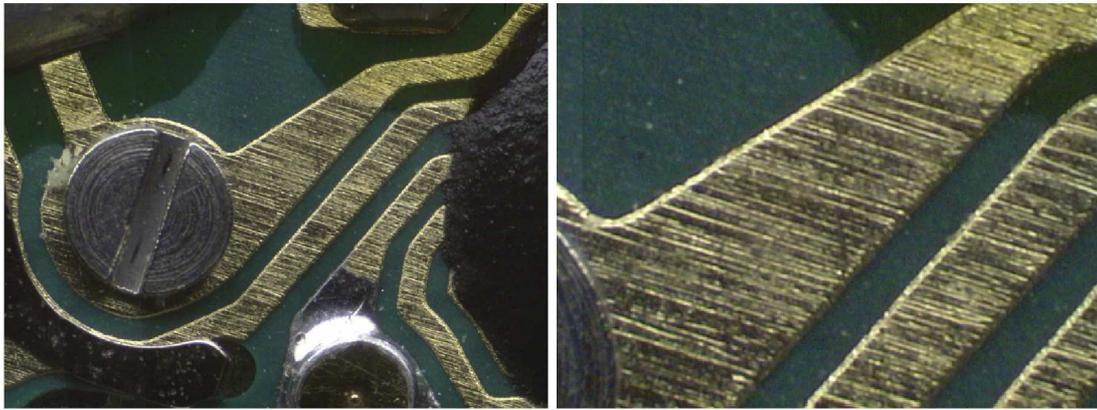


FIG. 2.8 – Illustration du faible champ de vision d'un microscope photonique. A gauche la résolution est de $9 \mu\text{m}/\text{pix}$, à droite elle est de $3 \mu\text{m}/\text{pix}$.

Contrainte en micro-assemblage Cette propriété du microscope photonique rend la tâche de contrôle et de surveillance d'un micro-assemblage très difficile. En effet si on désire une résolution maximale sur le micro-assemblage en cours, le champ de vision est localement centré sur le préhenseur. Ce champ de vision trop restreint ne permet pas à l'utilisateur de percevoir la totalité de l'espace de travail. Il ne connaît donc pas la position des zones importantes de la station, comme le magasin des éléments à assembler ou encore le magasin de sortie des pièces montées. Le besoin d'une vue locale de grande résolution, est tout aussi importante qu'une vue globale de faible résolution. Couramment pour pallier ce problème, en micro-assemblage deux solutions sont utilisées. La première consiste à modifier le grandissement de l'instrument pour jouer sur le champ de vision. Cette méthode s'applique seulement à certain matériel spécifique, car le grandissement n'est pas toujours modifiable. Pour certains microscopes photoniques à fort grandissement le seul moyen est de changer d'objectif par le biais d'une tourelle motorisée. Cette méthode ralentit considérablement le déroulement des tâches de micro-assemblage. La deuxième solution consiste à utiliser une autre caméra avec un objectif classique pour apporter une vue globale de la station. Cette solution rajoute un imageur

de plus dans le système de vision et par conséquent augmente la taille de la station de micromanipulation.

2.3.5 Dépendance à l'illumination

Rajouter des imageurs à différents angles de vue nécessite un contrôle de l'illumination de telle sorte à éclairer toutes les parties de la scène. Le réglage de l'illumination d'une scène est toujours une partie sensible dans un système de vision, encore plus quand il est composé de microscopes photoniques. La première difficulté dans l'illumination de la scène est liée au contraste de rendu du microscope photonique, qui s'écrit de la façon suivante :

$$C_I \propto \left(\frac{ON}{\Gamma_{mic}} \right) \quad (2.5)$$

où C_I est le contraste de l'image I obtenue par le microscope photonique. Le contraste est directement proportionnel à l'ouverture numérique et inversement proportionnel au grandissement du microscope. Le grandissement et l'ouverture numérique sont liés, par contre le grandissement évolue plus rapidement. Par conséquent, plus le microscope a un grandissement élevé et plus les images obtenues seront faiblement contrastées.

Contrainte en micro-assemblage Le faible contraste d'un microscope photonique n'est pas une limitation, par exemple, dans un système de microscopie photonique utilisé pour des scènes biologiques le mode d'éclairage par injection directe dans l'objectif est suffisant pour obtenir des images de qualité. Par contre dans une scène de micro-assemblage une bonne illumination n'est pas suffisante. Il existe un problème directement lié aux pièces mécaniques. En effet, la plupart des pièces mécaniques sont fabriquées en microtechnologie dans des wafers de silicium. Les plans des pièces placées dans des orientations différentes miroitent à cause de la réflexion due à la surface du silicium. Ces réflexions spéculaires peuvent compromettre la stratégie adoptée du traitement de l'image relatif à chaque pièce. Une illustration est représentée à la Figure 2.9, elle représente un assemblage de deux pièces réalisées sur la station SAMMI. En fonction de l'angle de prise de vue, les réflexions spéculaires n'interviennent pas sur les mêmes parties des pièces.

Pour contrôler l'illumination ambiante, le système d'éclairage d'une scène de micro-assemblage doit être tout aussi pensé que le système de vision. Pour obtenir une illumination globale de la scène, en plus des éclairages intégrés dans les microscopes photoniques, il faut des sources extérieures très diffuses pour éclairer chaque partie des pièces que l'on manipule. Une fois de plus on rajoute des éléments à la station de micromanipulation augmentant ainsi sa taille.

2.3.6 Résumé

Pour résumer la présentation des différentes propriétés du microscope photonique et leurs impacts sur la station de micromanipulation, nous proposons de les synthétiser



FIG. 2.9 – Assemblage de deux pièces ($400\ \mu\text{m}$) pris selon deux angles de vue différents. Les réflexions spéculaires ajoutent de la complexité à la scène. (Images Benoît Ballarin, LAB)

en chiffres et en images. La Figure 2.10 montre les caractéristiques d'objectifs que l'on peut trouver dans le commerce. Nous avons sélectionné une gamme d'objectifs Mitutoyo longue distance de travail, très répandue en microscopie pour leur qualité. Les cinq caractéristiques les plus importantes sont représentées : la distance de travail, la résolution, la profondeur de champ, le champ de vision et le poids.

	2x	5x	10x	20x	50x	100x
Distance de travail [mm]	34	34	33,5	20	13	6
Résolution [μm]	5	2	1	0,7	0,5	0,4
Profondeur de champ [μm]	91	14	3,5	1,6	0,9	0,6
Champ de vision* [mm]	9	3,6	1,8	0,9	0,36	0,18
Poids [g]	220	230	240	270	290	320

* diamètre du champ de vision pour un oculaire d'observation de 18 mm de diamètre.

FIG. 2.10 – Tableau des caractéristiques des objectifs Mitutoyo Plan Apo longue distance de travail.

La Figure 2.11 montre sous forme de graphique à bâton l'évolution de chaque caractéristique en fonction du type de grossissement. On remarque que la contrainte la plus touchée par l'augmentation du grossissement est la profondeur de champ, elle est de $91\ \mu\text{m}$ pour un 2x, et chute rapidement à $3,5\ \mu\text{m}$ pour un 10x.

Au début de cette section nous nous posons une question fondamentale entre une micromanipulation dans le plan (typique en biologie) et une micromanipulation dans l'espace (micro-assemblage) : est-ce que cela change vraiment la stratégie de micromanipulation ? Après avoir listé toutes les propriétés propres au microscope photonique il est assuré que cela change complètement la gestion de la micromanipulation. Il y a un

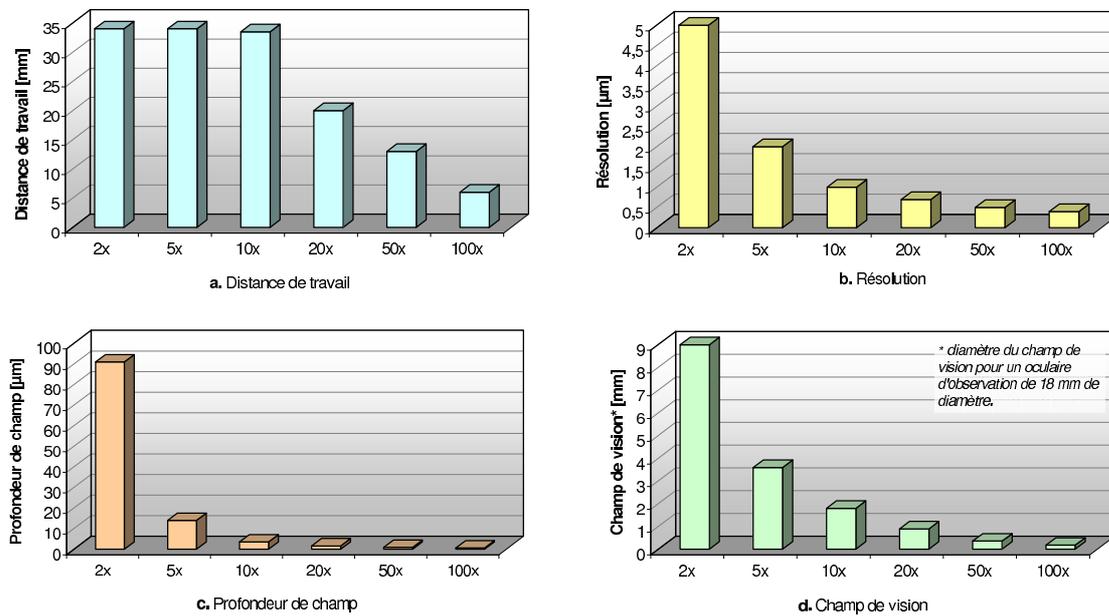


FIG. 2.11 – Représentation des différentes caractéristiques d’objectifs Mitutoyo (Plan Apo longue distance de travail) en fonction du grandissement : a. la distance de travail, b. la résolution, c. la profondeur de champ et d. le champ de vision.

besoin de changer de grandissement et de mise au point pendant le micro-assemblage, de contrôler un éclairage complexe et global, d’avoir des vues locales et globales, etc. La solution générale à toutes ces contraintes est d’ajouter des sources d’images au système de vision (vue de dessus, vue de côté, vue locale, vue globale, etc.). Le développement de système de vision distribué est-il une réelle solution ? Quel est son impact sur l’encombrement ?

2.4 Perception distribuée

Les sous-sections précédentes ont permis de mettre en évidence la nécessité d’avoir plusieurs vues dans le système de vision. Il n’y a pas qu’une source d’images mais plusieurs. Cette perception distribuée permet de collecter plus d’informations sur l’environnement observé. Ce sont soit des informations redondantes qui permettent de gagner en robustesse sur l’exactitude de l’information, soit des informations nouvelles complémentaires. Par exemple, pour contrôler l’évolution des outils sur l’axe Z , il faut disposer d’une vue de côté, cette nouvelle source d’images ajoute de la connaissance sur la scène. Un même microscope photonique, avec deux mises au point différentes, donne deux images de la scène, elles sont redondantes mais permettent d’accroître la robustesse de l’information observée. A noter que dans ce cas, c’est la même source d’images qui produit les deux vues distinctes.

Dans les systèmes de vision un système dit distribué consiste à répartir les imageurs à différentes positions et orientations distinctes dans l'espace, ainsi le système perçoit l'environnement avec n vues au même instant en temps réel comme par exemple deux caméras montées en montage stéréoscopique. Dans certain cas, on peut considérer un système composé d'une seule source d'images comme étant apparenté à un système pseudo distribué. Par exemple, une caméra qui balaie une surface, perçoit l'environnement avec des vues de positions différentes. Seulement la perception est réalisée à différents instants dans le temps, c'est un système à temps déporté. Un système de vision peut être considéré comme étant un système pseudo distribué si la scène est invariante dans le temps.

L'objectif d'avoir un système de perception distribué est d'accroître la connaissance de l'environnement le plus possible, en vue de piloter avec robustesse les différentes tâches de micromanipulation. Quel est le nombre minimum de vues nécessaire pour connaître parfaitement l'environnement ? En pratique, on se limite strictement aux vues nécessaires pour le pilotage de la station de micromanipulation, c'est-à-dire seulement connaître la partie de l'environnement qui est utilisée pour le contrôle des tâches. Même en se limitant au strict nécessaire, on ne compte pas moins de 2 à 4 sources d'images de tous types dans une station de micromanipulation. Le système de vision est un élément très contraignant, mais indispensable, il limite l'espace de travail et surcharge la station. L'idéal serait un système minimal de sources d'images pour un nombre de vues maximales. Physiquement, une source d'images ne donne qu'une vue à un instant donné, c'est pourquoi la solution réside dans la reconstruction de l'information à partir d'un minimum de connaissances de l'environnement. Cette reconstruction peut être assimilée à une vue nouvelle (obligatoirement virtuelle), d'une source d'images qui n'existe pas physiquement. Le nombre de vues virtuelles va dépendre du nombre de vues réelles utilisées pour la reconstruction :

- 1 vue réelle seule ne permet pas de reconstruire d'autres vues si on a aucune connaissance *a priori* du modèle d'environnement,
- 2 vues réelles permettent de reconstruire n vues nouvelles dans l'espace d'un modèle d'environnement limité,
- n vues réelles permettent de reconstruire m vues virtuelles dans l'espace d'un modèle d'environnement complet.

Si la reconstruction avec n vues réelles est très robuste du fait de la connaissance d'un modèle complet, elle est cependant très lourde, tant pour l'aspect physique que méthodologique. La solution serait un système distribué minimal de 2 vues réelles permettant la reconstruction de n vues virtuelles. L'inconvénient, c'est que la vue virtuelle reconstruite doit être dans un modèle d'environnement partiel, c'est-à-dire qu'il est impossible de reconstruire de l'information à laquelle on n'a pas accès. Par exemple, il est impossible, malgré la puissance cognitive de notre cerveau, de nous voir de dos sans utiliser des artifices optiques ou contorsionnistes. Ce mode de reconstruction limite dans l'espace la position des nouvelles vues par rapport aux positions de celles réelles. Il faut déterminer les n vues virtuelles avec des méthodes de reconstruction. Le domaine de la vision propre à la reconstruction de vues virtuelles, se nomme la synthèse de nouvelles vues (*Novel View Synthesis*). Nous allons lister les différentes méthodes de construc-

tion. Cela nous permettra de déterminer les méthodes les plus appropriées pour lever les contraintes des spécificités du microscope photonique.

2.5 Synthèse de nouvelles vues

La synthèse de nouvelles vues est une partie de la vision artificielle introduite par S. Chen et L. Williams (1993). Leurs réflexions portaient sur l'obtention d'un maximum de vues d'un environnement à partir d'un minimum de données réelles sur celui-ci. Leurs idées étaient d'interpoler des vues intermédiaires entre chaque image de synthèse d'une animation 3D. Cette méthode permet de diminuer le nombre de rendus à calculer dans une séquence d'images de synthèse. Ceci constitua le premier concept de la synthèse de nouvelles vues. Ce domaine est à l'interface entre la vision artificielle (*computer vision*) et l'infographie (*computer graphics*). On peut la définir de la façon suivante.

La synthèse de nouvelles vues est la reconstruction d'une nouvelle vue d'un environnement réel ou synthétique avec le moins d'informations sur celui-ci.

L'environnement peut être à la fois réel, comme dans le cas de la vision artificielle, mais également synthétique, dans le cas d'images de synthèse. C'est pourquoi la synthèse de nouvelles vues se classe en trois catégories : le rendu à partir de modèle, le modèle à partir d'images et le rendu à partir d'images.

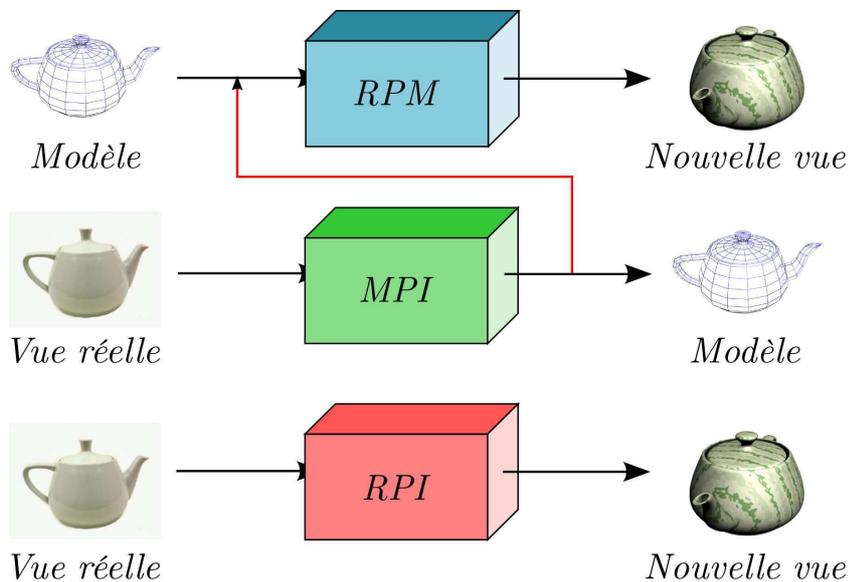


FIG. 2.12 – Les trois classes de la synthèse de vues : rendu à partir de modèle (RPM), modèle à partir d'images (MPI), rendu à partir d'images (RPI).

La Figure 2.12 illustre les entrées et les sorties des différentes catégories de la synthèse de nouvelles vues. Certaines classes sont généralement des systèmes de type MIMO

(*Multiple Input Multiple Output*), c'est-à-dire que pour avoir en sortie une ou plusieurs vues, il faut plusieurs vues en entrée, sauf dans le cas du rendu à partir de modèle (RPM). Le RPM génère une image de synthèse d'une nouvelle vue à partir d'un modèle pré-établi de la scène, donc sans aucune vue. Le modèle à partir d'images (MPI), construit un modèle mathématique de la scène via des vues réelles de celle-ci. On peut utiliser ce modèle directement ensuite avec un RPM. Ce processus MPI+RPM est la chaîne classique de la reconstruction 3D à partir d'images. La dernière catégorie, appelé rendu à partir d'images, calcule une nouvelle vue à partir des images de vues réelles sans passer par la connaissance d'un modèle.

2.5.1 Rendu à partir de modèle

Le RPM correspond au domaine de l'infographie c'est-à-dire de la modélisation et du rendu de l'image de synthèse. Il est à la limite du domaine de la vision artificielle mais reste toujours rattaché à l'infographie par le côté de la représentation de l'information synthétique, c'est-à-dire que tout le modèle de l'environnement est synthétique et modélisé par l'utilisateur. Pour cela, on utilise un logiciel appelé *modeleur*, qui permet de construire mathématiquement chaque objet de la scène, parfois simplement à partir d'éléments simples appelés primitives (cube, sphère, ...). Pour des éléments plus complexes on définit chaque point des sommets des objets dans l'espace. Connaissant le modèle mathématique de l'objet, une image peut être calculée pour différents angles de vue, c'est l'étape du rendu. Il existe sur le marché de nombreux logiciels de modélisation et de rendu : 3DS Max, SoftImage XSI, Houdini, Maya, Blender, Lightwave, Renderman, ... Cette classe du domaine de la synthèse de nouvelles vues est très populaire, par l'intermédiaire de son utilisation courante dans la modélisation et l'animation de personnages dans les jeux vidéos et dans les films.

2.5.2 Modèle à partir d'images

Le modèle à partir d'images (MPI) fait partie de la vision artificielle par ordinateur, sa particularité est de modéliser mathématiquement l'environnement via des images de vues réelles. Le modèle ainsi déterminé peut être directement utilisé par un RPM pour représenter l'environnement sur différentes vues. Cet enchaînement des deux catégories MPI+RPM caractérise la problématique de la synthèse de vues par reconstruction 3D suivie d'un rendu. Au final, la synthèse de vues via une reconstruction 3D emprunte aux deux domaines, la vision artificielle et l'infographie. La première partie utilise des méthodes propres à la vision artificielle pour reconstruire un modèle 3D, comme la correspondance des points (H. Saito, S. Baba et T. Kanade 2003) ou encore par la silhouette des objets (C. Hernandez et F. Schmitt 2004). Pour plus de détails on se référera à S. Seitz et al. (2006). La deuxième partie du processus permet d'avoir un rendu du modèle en images selon une vue pré-établie. La partie rendu d'images de synthèse est propre au domaine de l'infographie. La synthèse de vues par reconstruction 3D est un processus très contraignant et lourd, pourtant très à la mode par le biais de la puissance de calcul de nos machines de plus en plus performantes. La chaîne du processus de la

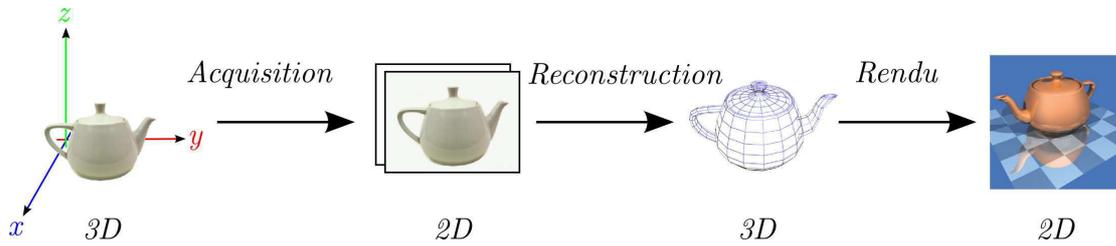


FIG. 2.13 – Chaîne de reconstruction $3D$, de l'acquisition au rendu.

synthèse de vues avec reconstruction $3D$ avec rendu est représentée à la Figure 2.13. Cette chaîne montre les différentes étapes : acquisition, reconstruction $3D$ et rendu. La première étape est l'acquisition visuelle de notre environnement. Nous vivons dans un monde aux 3 dimensions spatiales, donc après une acquisition par une source d'images, l'information est réduite à 2 dimensions. A partir de plusieurs images $2D$ on retrouve la troisième dimension manquante après reconstruction. Une fois le modèle retrouvé dans ces dimensions d'origine, le modèle est utilisé pour l'étape du rendu. Nos moyens de visualisation de l'information visuelle se résument à des simples écrans $2D$, par conséquent, le rendu va projeter le modèle $3D$ dans la vue désirée en une image de dimension 2. Au final, la chaîne, au niveau des dimensions spatiales de l'information, se résume par acquisition $2D$, reconstruction $3D$, rendu $2D$. Cette chaîne fait le yo-yo entre informations de dimension 2 et 3 pour au final la rendre sous un format $2D$. Dans cette méthode, intuitivement, on ressent que l'étape de reconstruction est de trop. Pourquoi ne pas reconstruire directement à partir d'un ensemble d'images. Cette catégorie spéciale est le rendu à partir d'images.

2.5.3 Rendu à partir d'images

Le rendu à partir d'images (RPI) également appelé reconstruction $2D$, utilise directement les informations $2D$ de l'image des vues pour reconstruire une image de la vue virtuelle désirée. Par rapport à la chaîne de synthèse de vues par reconstruction $3D$ les étapes de reconstruction et de rendu $3D$ sont remplacées par une seule étape (Figure 2.14). L'étape qui correspond au passage direct de $2D$ à $2D$ correspond également à une reconstruction mais directement dans l'image et non dans l'espace, car chaque pixel est directement transféré dans l'image résultat (rendu $2D$). Par contre cette reconstruction n'utilise pas de modèle $3D$, les nouvelles vues de synthèse seront donc limitées en extrapolation dans l'espace. Le RPI a une philosophie intéressante malgré les extrapolations limitées, car on ne reconstruit que l'information qui nous est nécessaire. Par exemple, nul besoin de reconstruire le détail de l'arrière d'un objet si la nouvelle vue désirée est orientée devant cet objet. Comme on choisit seulement l'information utile pour la reconstruction, en fonction du résultat que l'on attend, on va utiliser plus ou moins la connaissance que l'on a de l'environnement. Ce qui permet de les classer en trois catégories proposées par H.-Y. Shum et S. Kang (2000) : le rendu sans géométrie, le rendu avec géométrie implicite et le rendu avec géométrie explicite. Le classement est en

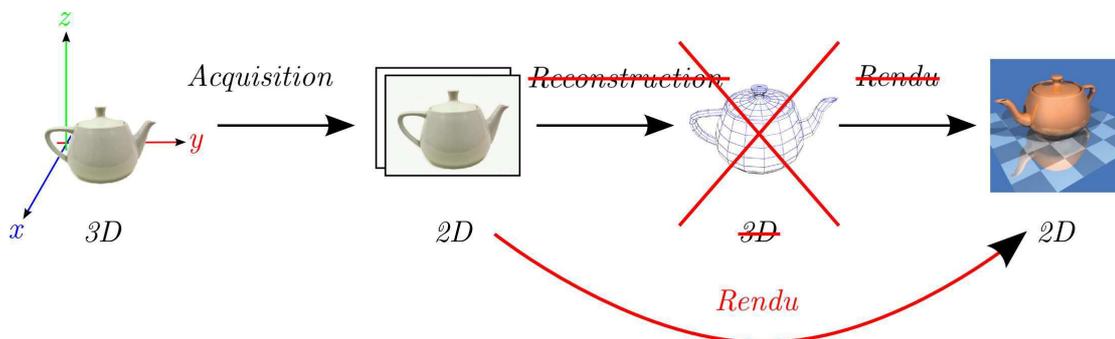


FIG. 2.14 – La chaîne de reconstruction 2D par rapport à la chaîne de reconstruction 3D.

fonction du niveau de la connaissance géométrique que l'on va utiliser dans la scène pour reconstruire la nouvelle vue. Les différentes techniques de chaque catégorie dépendent également du nombre de vues réelles nécessaire à la reconstruction, cela va d'une seule caméra (S. Boivin et A. Gagalowicz 2001) à une batterie de caméras en anneaux (J.-F. Evers-Senne et R. Koch 2005).

L'approche du RPI est très bien appropriée à nos attentes pour différentes raisons : elle nécessite très peu de vues réelles, c'est une méthode sans modèle qui permet de diminuer le nombre de calcul et de plus elle utilise différents niveaux de géométrie dans la scène ce qui limitera le niveau de complexité des calibrages des microscopes photoniques par la suite. Dans la littérature, on retrouve plusieurs documents qui donnent un aperçu de la méthode du RPI. Les différentes techniques, dans les sections qui suivent seront classées à partir de ces ouvrages (S. Kang 1997, C. Zhang et T. Chen 2003, D. Burschka et al. 2003, H.-Y. Shum, S.-C. Chan et S. Kang 2006).

2.5.4 Rendu à partir d'images sans géométrie

Cette sous-catégorie du domaine de la synthèse nouvelles de vues n'utilise à aucun moment l'information géométrique présente dans la scène pour faire la reconstruction. Pour cela, ces techniques de reconstruction utilisent un autre moyen de représentation de l'environnement. Ce moyen est basé sur l'apparence et plus particulièrement sur l'intensité des rayons lumineux. La description de l'environnement via des rayons lumineux est modélisée par une fonction appelée la fonction plénoptique. Cette fonction permet d'exprimer l'intensité d'un flux lumineux, qui passe au centre de la caméra, émis à partir de chaque point de l'espace et ce dans toutes les directions. La fonction plénoptique est proposée par E. Adelson et J. Bergen (1991), elle est représentée à la Figure 2.15.

La fonction plénoptique originale est définie selon 7 dimensions : pour tous les points de l'espace (V_x, V_y, V_z) , pour tous les angles possibles (θ, ϕ) , pour toutes les longueurs d'onde λ et à tout instant t . La fonction plénoptique est de la forme $P_7(V_x, V_y, V_z, \theta, \phi, \lambda, t)$. Dans la majorité des scènes étudiées en vision, l'environnement est stationnaire et l'éclairage constant. C'est pourquoi L. McMillan et G. Bishop (1995b) définissent non plus

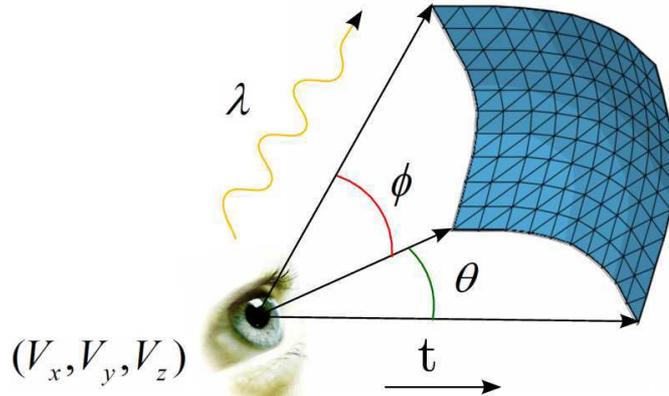


FIG. 2.15 – La fonction plénoptique $P_7(V_x, V_y, V_z, \theta, \phi, \lambda, t)$

une fonction plénoptique à 7 dimensions mais un modèle plénoptique à 5 dimensions de la forme $P_5(V_x, V_y, V_z, \theta, \phi)$ en supprimant la longueur d'onde et le temps. Toutes les techniques du rendu à partir d'images sans géométrie seront classées par ordre décroissant en fonction de la dimension de la fonction plénoptique, dans l'ordre, le *Light field* et le *Lumigraph* avec une fonction de la forme $P_4(u, v, r, s)$, le *Concentric mosaics* $P_3(\alpha, C_i, V_z)$, et le *mosaicing* $P_2(\theta, \phi)$.

2.5.4.1 Le *Light field* et le *Lumigraph*

Le *Light field* et le *Lumigraph* ont été présentés au même moment, respectivement par M. Levoy et P. Hanrahan (1996) et par S. Gortler et al. (1996). Les deux méthodes sont pratiquement identiques. Elles sont basées sur l'enregistrement des intersections des rayons lumineux de l'environnement avec deux plans. Ces deux plans définissent la boîte englobante de la perception de l'environnement. Ce qui contraint la fonction plénoptique à 4 dimensions, (u, v) pour le premier plan et (r, s) pour le deuxième (Figure 2.16).

Le premier plan est défini comme étant le plan de la caméra et le deuxième comme étant le plan focal. Chaque plan est discrétisé en un nombre fini de rayons lumineux à enregistrer. Au moment de l'acquisition, la caméra suit une trajectoire bien définie avec des pas constants. Chaque image, c'est-à-dire tous les rayons lumineux discrétisés qui traversent la surface (r, s) vont être liés à un point discret du plan (u, v) . Par conséquent pour chaque point discret du plan (u, v) , on obtient une image qui correspond à un tableau 2D des rayons optiques (Figure 2.16 a). Pour obtenir une nouvelle vue, on commence par la décomposer en éléments discrets sur le plan (r, s) . Pour un point discret du plan (r, s) il existe un ensemble de rayons connectés à une zone particulière des imagerie du plan (u, v) . On utilise cet ensemble de rayons pour calculer par interpolation le point discret du plan (r, s) (Figure 2.16 b). On utilise cette même méthode pour calculer tous les points discrets du plan (r, s) afin d'obtenir la nouvelle vue. La nouvelle vue est un ré-assemblage des parties des rayons lumineux enregistrés. On détermine la vue en se plaçant sur un noeud d'intersection de tous les rayons à l'intérieur de l'espace (u, v, r, s)

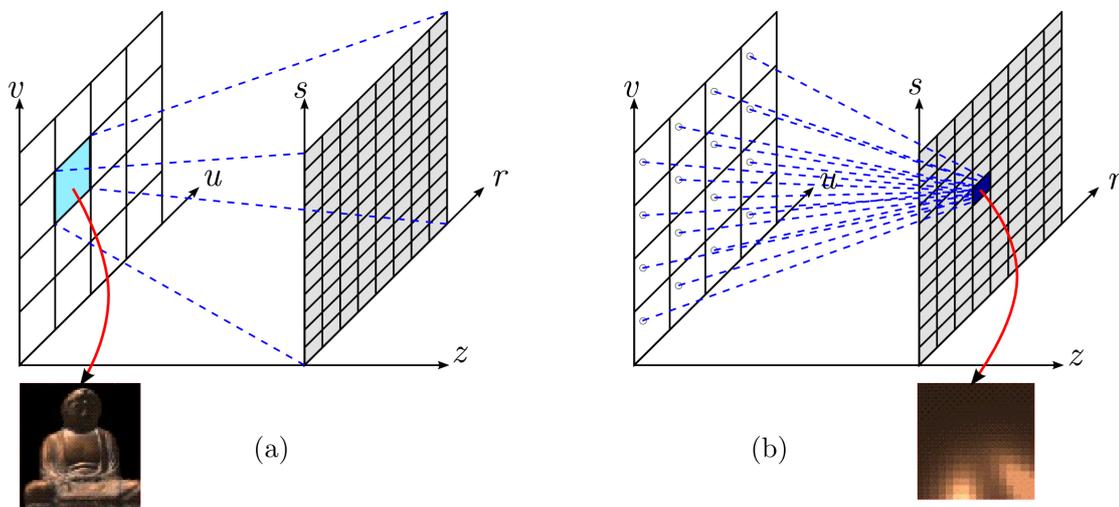


FIG. 2.16 – Fonctionnement du Light field : (a) principe de l'acquisition de chaque rayon des images dans le plan (u, v) avec un exemple d'imagette obtenue, (b) principe du rendu d'une nouvelle vue pour un seul élément discret du plan (r, s) avec un exemple d'échantillon obtenu.

pour tous les éléments discrets. C'est comme si le plan image glissait par rapport au plan focal. La position de la nouvelle vue de synthèse doit être dans la zone d'espace où les rayons ont été enregistrés, c'est-à-dire l'espace de la boîte englobante. Le rendu d'une nouvelle image demande un temps de calcul important, mais il peut très bien s'effectuer en temps réel (P. Sloan, M. Cohen et S. Gortler 1997, H. Lensch 1999).

2.5.4.2 Le *concentric mosaics*

H.-Y. Shum et L.-W. He (1999) présentent une technique basée sur une fonction plénoptique de dimension 3 appelé *concentric mosaics*. Le *concentric mosaics* est basé sur la construction de plusieurs mosaïques (voir §2.5.4.3). Une mosaïque est un ensemble d'images prises séparément puis assemblées les unes avec les autres pour former une image panoramique de l'environnement. Le *concentric mosaics* utilise plusieurs mosaïques acquises par une caméra qui se déplace sur des cercles concentriques de différents rayons. La caméra est placée soit tangente à la trajectoire soit sur la normale (Figure 2.17).

Le *light field* et *Lumigraph* permettaient d'avoir une nouvelle vue dans l'espace de la boîte englobante. Avec le *concentric mosaics* de nouveaux points de vue se trouvent à l'intérieur d'un disque, la nouvelle fonction plénoptique s'écrit $P_3(\alpha, C_i, V_z)$ avec α l'angle de rotation de la caméra, R_i le rayon des cercles concentriques et V_z la hauteur verticale. Le calcul de la nouvelle vue peut se faire de différentes manières. Dans le cas du *concentric mosaics* classique de H.-Y. Shum et L.-W. He (1999), le rendu est directement calculé pour chaque nouvelle ligne verticale (raie) de l'image. Les nouvelles raies sont interpolées à partir de deux raies provenant de mosaïques concentriques voisines. Cependant il existe une déformation verticale dans le rendu qui peut être estompé par

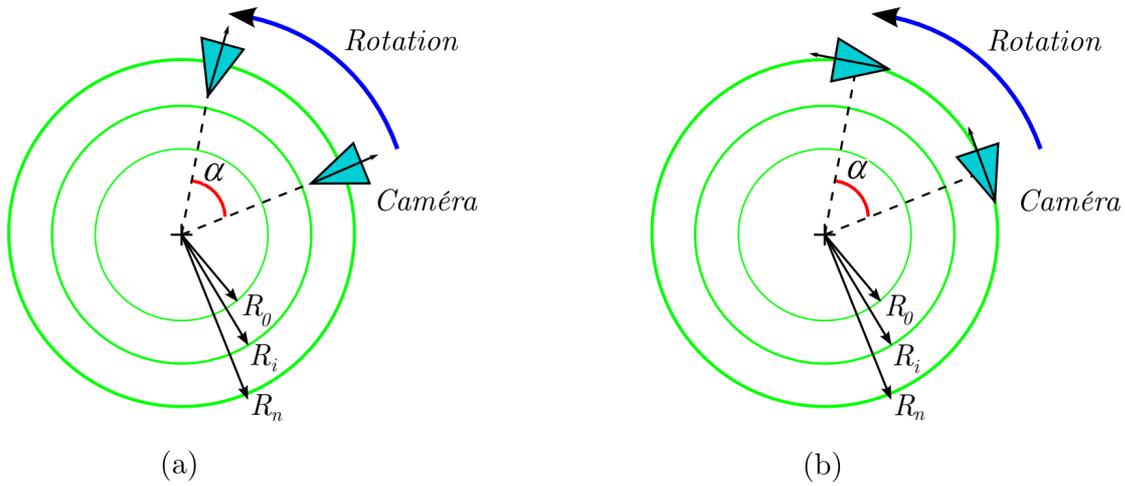


FIG. 2.17 – Processus d'acquisition par *concentric mosaics* pour différents rayons C_i , la caméra suit le cercle pour trajectoire avec un angle α selon deux méthodes : (a) la caméra est à la normale du cercle, (b) la caméra est sur la tangente du cercle.

une correction de profondeur présenté par J.-X. Chai et al. (2000), leur méthode sélectionne la meilleure raie à utiliser pour le rendu. La nouvelle vue obtenue accepte des déplacements dans le panoramique de gauche à droite mais aussi des déplacements suivant des petites rotations autour de la caméra. Cette petite rotation est possible grâce à l'observation de la scène sur des valeurs légèrement différentes d'angle de vue des cercles concentriques lorsque la caméra est placée tangente ou normale à la trajectoire. Récemment A. Rav-Acha et al. (2005) proposent un autre mode de rendu appelé *Dynamosaicing*. Le *Dynamosaicing* réalise une vidéo panoramique, c'est-à-dire que dans l'image panoramique certains objets sont dynamiques. L'exemple pris par A. Rav-Acha et al. (2005) montrent une chute d'eau panoramique (Figure 2.18). Pendant que vous vous déplacez dans le panoramique l'eau s'écoule des chutes. Pour obtenir ce résultat, différentes raies dans l'espace et le temps sont judicieusement sélectionnées au moment du rendu. Contrairement à ce que l'on pourrait croire, par l'ajout d'un paramètre temporel, la fonction plénoptique n'est pas de dimension 4. L'observation ne se fait pas au même instant t lors de l'acquisition, c'est-à-dire, si on observe la vidéo à l'instant t , l'eau qui s'écoule à gauche du panoramique est à l'instant t , mais l'eau qui s'écoule à droite du panoramique est à l'instant $t + \Delta t$ (dans l'hypothèse que le balayage de l'acquisition des images soit de gauche à droite). Pour que la fonction plénoptique soit de dimension 4 toutes les mosaïques concentriques doivent être connues pour n'importe quel instant t .

On peut également obtenir un panoramique stéréoscopique toujours par le biais de la sélection des raies pour le rendu. Cette méthode a été proposée par Shmuel Peleg et Moshe Ben-Ezra (1999). Par le biais de la légère différence d'angle de vue des placements des caméras (tangente ou normale), on peut en retirer deux mosaïques concentriques gauche et droite. Ces deux mosaïques peuvent être montées en stéréo-anaglyphe pour un rendu visuel 3D (via des lunettes bicolors cyan et rouge).



FIG. 2.18 – Vidéo panoramique par *Dynamosaicing*. (Image A. Rav-Acha et al. ©2005 IEEE)

2.5.4.3 Le *mosaicing*

Le *mosaicing* est une technique qui a la fonction plénoptique la plus simple, seulement 2 dimensions. Elle consiste à construire une image mosaïque (panoramique) de la scène à partir d'un ensemble de petites images qui représentent des portions de cette scène. Le *mosaicing* apparaît dans la littérature avec D. Milgram (1975) et le besoin de diminuer l'impact visuel des bords d'images satellites assemblées pour une carte géographique. Le panoramique utilisé en tant que vue virtuelle apparaît avec S. Chen (1995) et le logiciel de navigation virtuelle Quicktime® VR. Il existe plusieurs méthodes pour acquérir les différentes images qui constituent la mosaïque. La première présentée par S. Chen (1995) consiste à utiliser une caméra avec un point de vue fixe dans l'espace. La caméra balaie la scène par rotation lacet-tangage centrée sur elle-même (Figure 2.19). La fonction plénoptique s'écrit donc sous une forme à 2 dimensions $P_2(\theta, \phi)$. La deuxième méthode consiste directement à obtenir le panoramique sans balayage avec des technologies de caméra omnidirectionnel initialement présentées par V. Nalwa (1996) ou par un type de lentille *fish-eye* par Y. Xiong et K. Turkowski (1997). Dans ce cas, la fonction plénoptique prend une variable supplémentaire qui est le temps t . La dernière méthode d'acquisition utilise une fonction plénoptique du type $P_2(V_x, V_y)$. Appelé *manifold mosaics* ou encore *VideoBrushTM*, S. Peleg et J. Herman (1997) présentent une méthode où la caméra balaie la scène avec un point de vue libre, c'est-à-dire dans l'espace. Le *mosaicing* assemble toutes les images dans une même surface définie comme référentiel, c'est pourquoi, même si la caméra est libre dans l'espace la fonction plénoptique ne dépasse pas plus de 2 dimensions.

La construction de mosaïque se compose de quatre étapes. La première est le recalage géométrique et photométrique des images dans une même référence (Figure 2.19). Elle permet de corriger les variations d'illumination (photométrie) et les transformations géométriques provoquées par le changement de vue entre chaque acquisition et de placer chaque image l'une par rapport à l'autre. Le recalage peut se faire directement si on connaît le mouvement de la caméra (B. Rousso, S. Peleg et I. Finci 1997, Philippe Blanc et Eric Savaria 2001) sinon il faut estimer la matrice de transformation projective également appelée la matrice de colinéation. Elle s'estime soit par corrélation de trans-

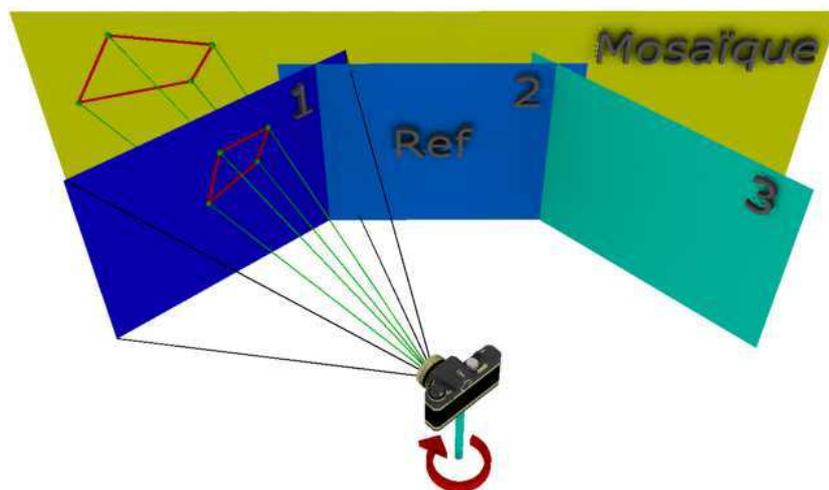


FIG. 2.19 – Illustration du recalage : l'image 1, l'image 2 et l'image 3 sont projetées dans la mosaïque avec pour référence l'image 2.

formation (Richard Szeliski et Heung-Yeung Shum 1997), soit en utilisant des indices invariants dans chaque image (D. Capel 2001). La deuxième étape consiste à reprojeter toutes les images dans un référentiel soit planaire, cylindrique ou sphérique en fonction du type de rendu que l'on désire. L'étape suivante appelée *blending* fusionne toutes les images dans la mosaïque de telle sorte que les bordures des images se fondent complètement en une seule image. La dernière étape permet d'avoir un panoramique esthétique, le détournage découpe la mosaïque pour qu'elle soit de forme rectangulaire. En fonction de la reprojection choisie la nouvelle vue évolue sur une surface planaire, sphérique ou cylindrique.

2.5.5 Rendu à partir d'images avec géométrie implicite

Dans le cas de la sous catégorie rendu à partir d'images avec géométrie implicite, la géométrie utilisée dans la scène pour reconstruire une nouvelle vue n'est pas directement disponible, l'information manquante de la géométrie est reconstruite par des calculs de projection. Bien souvent, la reconstruction de la géométrie est obtenue par l'utilisation du calibrage faible des caméras, mais à aucun moment on a accès à de l'information dans l'espace $3D$. La notion de calibrage faible, c'est-à-dire la détermination de la géométrie relative entre plusieurs vues est très importante dans le domaine du rendu à partir d'images (Y. Genc 1999). Nous aurons l'occasion d'en rediscuter dans le prochain chapitre. Cependant, certaines techniques nécessitent de connaître soit complètement, soit partiellement, les paramètres intrinsèques des caméras. Les nouvelles vues sont obtenues directement

par la relation entre les indices invariants appariés des vues d'origine. Généralement, les techniques de rendu avec géométrie implicite nécessitent beaucoup moins de vues que le rendu sans géométrie. Les différentes techniques de rendu seront présentées dans l'ordre suivant, l'interpolation de vue, le morphage de vue, la vue par triangulation commune, le transfert par colinéation, le transfert épipolaire et enfin le transfert trifocal.

2.5.5.1 Interpolation de vue

La première méthode de rendu par géométrie implicite revient à S. Chen et L. Williams (1993). Ils proposent une méthode qui permet d'obtenir une vue arbitrairement placée entre les deux vues réelles. Pour cela, il faut déterminer une correspondance dense, c'est-à-dire déterminer la correspondance entre tous les pixels de la première image avec la deuxième. Elle est obtenue par des méthodes de flot optique (*optical flow*). La nouvelle vue est calculée par l'interpolation du mouvement estimé des pixels dans chaque image réelle. Cette méthode est très contraignante, il faut obligatoirement que la nouvelle vue soit sur la ligne de base des caméras (ligne virtuelle entre les deux centres optiques des caméras). Une méthode récente élimine cette restriction en donnant plus de liberté à la nouvelle vue (K. Kimura et H. Saito 2005). Cependant le risque d'ambiguïté dans le rendu de la nouvelle vue oblige les deux caméras réelles à être très proches l'une de l'autre.

2.5.5.2 Morphage de vue

Le morphage de vue, en anglais le *view morphing*, est basé sur des techniques de morphage, présenté par S. Seitz et C. Dyer (1996). Cette technique permet de reconstruire, comme l'interpolation de vue de S. Chen et L. Williams (1993), une caméra virtuelle placée sur la ligne de base des caméras. Dans le cas du *view morphing*, il existe une cohérence géométrique qui permet d'avoir des distances plus importantes entre les deux caméras réelles. La cohérence géométrique est obtenue par une reprojection de la combinaison linéaire des images rectifiées des caméras (Figure 2.20).

Le *view morphing* se compose en trois étapes. La première étape correspond à une rectification épipolaire de chaque image réelle. Pour chaque image I_0 et I_1 il faut déterminer les matrices de rectification épipolaire respective H_0 et H_1 . Ce qui permet, par transformation projective, d'obtenir les images \hat{I}_0 et \hat{I}_1 rectifiées, c'est-à-dire que les images sont sur le même plan parallèle et que chaque point de l'image de gauche a la même ordonnée dans l'image de droite, il y a une cohérence géométrique. La deuxième étape calcule par des fonctions de morphage une nouvelle vue par interpolation linéaire sans l'utilisation de correspondance dense. Cette nouvelle image \hat{I}_s est exprimée dans le repère rectifié. La dernière étape consiste à la replacer dans son repère d'origine via une nouvelle matrice de transformation projective H_s . Toutes les matrices de transformation sont calculées par les matrices de projection des caméras. S. Seitz et C. Dyer (1996) proposent deux méthodes, une par calibrage fort et l'autre sans calibrage. Exactement comme l'interpolation de vue, la nouvelle vue virtuelle doit être sur la ligne de base. D. Scharstein (1996) étend cette limite à un plan, récemment J. Xiao et M. Shah (2004)

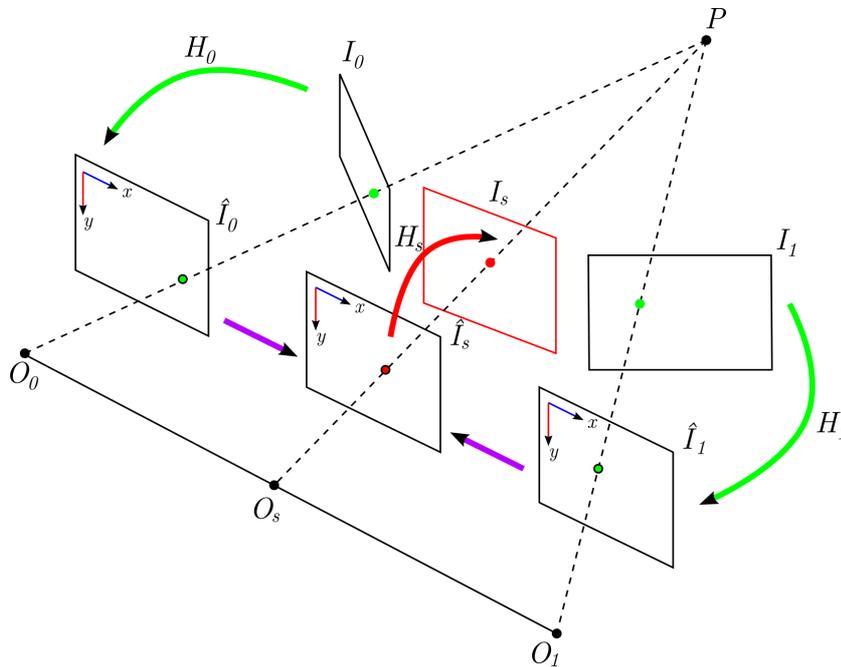


FIG. 2.20 – Principe du view morphing.

proposent d'utiliser trois caméras, permettant ainsi de positionner la nouvelle vue sur le plan des trois centres optiques des caméras.

2.5.5.3 Vue par triangulation commune

L'idée de M. Lhuillier et L. Quan (1999) est de pallier à la dense correspondance de l'interpolation de vue, par une triangulation de Delaunay, donc à un nombre de correspondances plus faible. La vue par triangulation commune (*Joint view triangulation*) comporte quatre étapes. La première étape consiste à déterminer un nombre de points d'intérêt élevé dans chaque image, et en faire la correspondance. Cette correspondance quasi-dense permet d'initialiser la deuxième étape, qui consiste en la création d'un quadrillage régulier à partir des points d'intérêt dans la première image. Pour chaque quadrillage de la première image et son correspondant dans la deuxième image, la matrice de colinéation est estimée par un algorithme robuste RANSAC (M. Fischler et R. Bolles 1981). En utilisant cette contrainte géométrique locale via la matrice de colinéation, chaque point du quadrillage de la deuxième image est rectifié avec les points d'intérêt les plus proches. Les points des quadrillages de la deuxième image sont modifiés de telle sorte à obtenir un quadrillage parfaitement continu sans trou. L'étape suivante consiste à la création d'un maillage par triangulation de Delaunay sur les sommets et les arêtes des quadrillages existants sur les deux images (Figure 2.21). La dernière étape est le rendu de la nouvelle vue. Un nouveau maillage est calculé pour déterminer des nouveaux triangles

en fonction des triangles communs de chaque image. La texture de chaque triangle est calculée à partir d'un ré-échantillonnage des triangles d'origine.

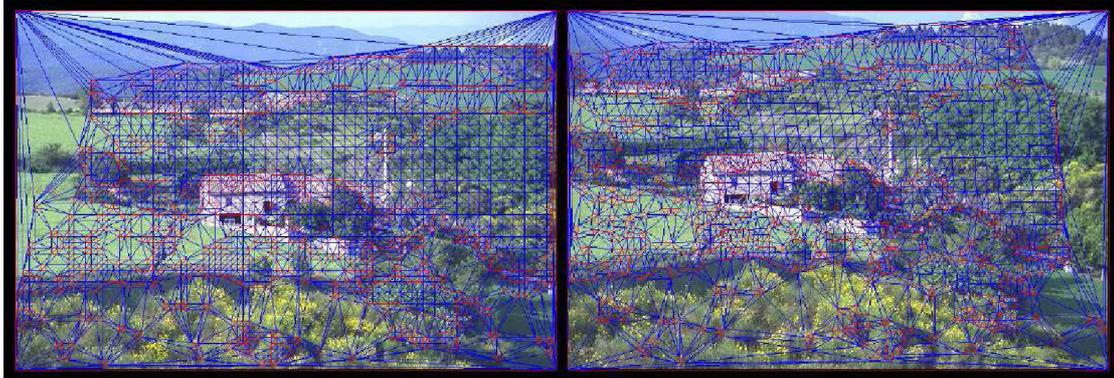


FIG. 2.21 – Résultat d'un maillage par triangulation de Delaunay sur les deux images dans la méthode du *Joint view triangulation*. (Images M. Lhuillier et L. Quan ©1999 IEEE)

2.5.5.4 Transfert par colinéation

Les techniques de transfert sont des méthodes qui utilisent différentes contraintes géométriques pour transférer les pixels d'origine de chaque image réelle dans la vue virtuelle désirée. La première technique utilise la matrice de colinéation comme contrainte géométrique, c'est-à-dire une simple transformation projective. Le transfert par colinéation est présenté par B. Boufama (2000). Cette technique utilise deux plans physiques dans la scène pour calculer différentes matrices de colinéation. Pour cela 4 points sont définis pour chaque plan physique dans la première image. Après une mise en correspondance avec la deuxième image puis une troisième, une collection de matrices de colinéation est estimée. Ces différentes matrices de transformation projective permettent de calculer la relation entre les points de la troisième image en fonction des points de la première et de la deuxième. La première étape de la synthèse d'une nouvelle vue consiste à supprimer la troisième vue, qui était utilisée seulement pour définir la collection de matrices de colinéation. Ensuite 3 couples de points appariés dans les deux images restantes, sont utilisés pour définir un plan virtuel dans l'espace. Ce plan virtuel définit par projection l'emplacement de la vue virtuelle désirée. Chaque point de la troisième vue virtuelle est calculé par l'intermédiaire du plan virtuel et des différentes matrices de colinéation pré-établies. Cette méthode nécessite des contraintes de fonctionnement importantes, il faut obligatoirement trouver deux plans physiques dans la scène, utiliser une troisième vue et réussir à placer la vue désirée par un plan virtuel dans l'image.

2.5.5.5 Transfert épipolaire

Le transfert épipolaire utilise une contrainte géométrique différente. Deux vues d'un système de stéréo-vision sont intrinsèquement liées pour la géométrie épipolaire (voir §3.2). La géométrie épipolaire définit la relation géométrique relative de deux vues par l'intermédiaire d'une matrice appelée matrice fondamentale. Cette matrice définit la dualité point ligne dans les vues, c'est-à-dire, chaque point de l'image de gauche est projeté en une ligne dans l'image de droite et vice versa. Cette ligne définit l'ensemble des positions possibles de la projection du point de l'image de gauche, dans l'image de droite. Toutes ces lignes appelées lignes épipolaires se coupent en un seul point appelé épipole. S. Laveau et O. Faugeras (1994) utilisent la contrainte épipolaire pour générer de nouvelles vues. L'idée de la technique est basée sur un système à trois vues, deux réelles et une virtuelle. Pour cela il faut avoir la connaissance de la géométrie épipolaire entre toutes les vues. Si la collection de matrices fondamentales est connue, le point de l'image de gauche se projette en une droite dans la vue virtuelle et le point de l'image de droite se projette également en une droite dans la vue virtuelle. L'intersection des deux droites définit la position du point dans l'image virtuelle. Si on désire reconstruire l'ensemble des pixels de l'image virtuelle il faut connaître la correspondance dense des images. Toute la difficulté de cette technique réside dans la définition par l'utilisateur de la vue virtuelle. Plusieurs possibilités sont envisageables pour la définir : soit l'utilisateur choisit la vue virtuelle par le positionnement des nouveaux épipoles dans les deux vues réelles, soit en utilisant une troisième vue réelle pour la supprimer ensuite. Certaines positions de vues génèrent des déformations dans la nouvelle image, ceci peut être résolu par la connaissance des paramètres intrinsèques des caméras. Dans l'image résultat, des trous apparaissent, en effet les pixels sont transférés à partir des vues réelles et en fonction de la position de la nouvelle vue, certains pixels ne sont pas tous accessibles. Pour résoudre ce problème, une méthode de lancer de rayon (*Ray tracing*) est nécessaire. Cette étape appelée aussi *backward mapping*, utilise le trajet inverse du transfert des pixels pour garantir que chaque pixel de la nouvelle vue reçoive une valeur d'intensité, évitant ainsi les trous dans l'image (Figure 2.22).

Il existe différentes variantes dans la littérature basées sur le transfert épipolaire. La technique du *Line-of-Sight* de M. Irani, T. Hassner et P. Anandan (2002) utilise une méthode de lancer de rayon par contrainte épipolaire. Même si elle nécessite au moins une dizaine d'images pour calculer la nouvelle vue, on est garanti d'avoir une image sans trou avec une position de la nouvelle caméra très simple à définir. Pour cela il suffit de tracer un segment qui définit la direction de ce que regarde la nouvelle caméra dans les différentes vues d'origine. Il existe une amélioration du transfert épipolaire par A. Fusiello et al. (2003) qui permet par une méthode simple et automatique de placer le point de vue en utilisant une réplique de la géométrie épipolaire des deux vues de référence.

2.5.5.6 Transfert trifocal

Il existe une dernière méthode pour transférer les pixels dans une nouvelle vue. La géométrie épipolaire définit un système à deux vues de référence. En ajoutant une



FIG. 2.22 – Couple d’images stéréo à gauche et le rendu à droite de la nouvelle vue par transfert épipolaire et *backward mapping*. (Images S. Laveau et O. Faugeras ©1994 INRIA)

troisième vue on peut définir la géométrie épipolaire entre chaque couple de vues et ainsi l’étendre à une géométrie à trois vues appelée géométrie trifocale. S. Avidan et A. Shashua (1997) proposent de l’utiliser pour reconstruire une nouvelle vue. La relation de chaque pixel dans un système à trois vues est définie à partir d’un tenseur trifocal, c’est-à-dire que si on connaît les couples de points correspondants dans deux vues de référence et le tenseur trifocal qui régie le système, il est possible de connaître la position de ces points dans la troisième vue. Les auteurs utilisent trois vues réelles pour estimer le tenseur trifocal et en déterminer une quatrième virtuelle. La spécification de la vue virtuelle se fait par une matrice de rotation et de translation par le calibrage fort des caméras. Peu de temps plus tard S. Avidan et A. Shashua (1998) ont proposé une alternative qui permet d’amorcer un tenseur trifocal à partir de seulement deux vues réelles. Le placement de la nouvelle vue se fait toujours par le biais d’un calibrage fort des caméras. Comme dans le transfert épipolaire, l’image résultat est sujette aux trous, c’est-à-dire que tous les pixels de l’image virtuelle ne reçoivent pas d’intensité. C’est pourquoi les auteurs utilisent des artifices de texturage par peinture. Keith Connor et Ian Reid (2002) proposent une méthode de *backward mapping* spécifiée pour le transfert trifocal. Le placement de la vue est facile si le calibrage fort a été déterminé pour chaque caméra. Sinon l’extrapolation de la position de la caméra virtuelle est limitée (Figure 2.23). Premièrement elle est limitée par le placement manuel et arbitraire de la nouvelle vue qui est très difficile à réaliser, et deuxièmement il faut connaître parfaitement la correspondance dense des deux images de référence.

2.5.6 Rendu à partir d’images avec géométrie explicite

La dernière sous catégorie présentée ici du rendu à partir d’images utilise l’information explicite de la géométrie dans les images, c’est-à-dire que le rendu à partir d’images avec géométrie explicite va puiser directement l’information 3D dans les images. L’information 3D peut se trouver sous la forme d’information de profondeur, de ligne de vue ou



FIG. 2.23 – Couple d’images stéréo à gauche et le rendu à droite de la nouvelle vue par transfert trifocal. (Images K. Connor et I. Reid ©2002 BMVC)

encore directement par des coordonnées $3D$ (modèle). Le rendu à partir d’images avec géométrie explicite est très proche du rendu à partir de modèle puisqu’elle consiste à obtenir une nouvelle image à partir des textures (image $2D$) des images originales via une connaissance parfois du modèle de l’objet. Si le rendu à partir de modèle est plus lié à la partie géométrique du rendu, le rendu à partir d’images avec géométrie explicite quand à lui s’inscrit dans un rendu orienté texture (image). L’utilisation d’image comme texture fait que les différentes techniques ont leur place dans la catégorie rendu à partir d’images. Les sous-sections suivantes présentent différentes techniques à savoir, le nuage de panneaux, la déformation $3D$, le calque d’image par profondeur, la texture par dépendance de vue.

2.5.6.1 Nuage de panneaux

La technique de panneaux (*Billboard*) était largement utilisée dans les premiers jeux vidéo $3D$ (Doom, Duke Nukem, ...). Limité par la puissance de calcul des cartes graphiques, il fallait trouver une solution pour afficher rapidement des objets $3D$ dans un monde virtuel. Pour cela un simple *sprite* (texture $2D$ de l’objet) était plaqué sur un panneau dans l’espace virtuel du jeu. L’orientation du panneau suivait la direction de l’observateur donnant ainsi l’impression d’avoir un objet $3D$ dans l’espace virtuel. Pour toutes les vues, l’image de l’objet était la même quelles que soient les directions. Une amélioration considérable de ce principe est présentée par X. Decoret et al. (2003), au lieu d’utiliser un panneau seul, les objets sont représentés par un nuage de panneaux. Pour chaque modèle texturé d’objet, l’algorithme détermine un ensemble de plans qui caractérise l’objet. Chaque plan, définira un panneau qui prendra la texture associée à chaque plan avec des zones de transparence. Le nuage de panneaux est assemblé pour former la représentation de l’objet. La partie transparente des panneaux permet d’obtenir des modèles d’objets complexes. Cette technique permet le rendu en temps réel d’objets complexes avec le moins d’information possible. Le modèle de l’hélicoptère (Figure 2.24) illustre ce principe ; sa représentation classique compte 5138 polygones contre seulement 32 panneaux.

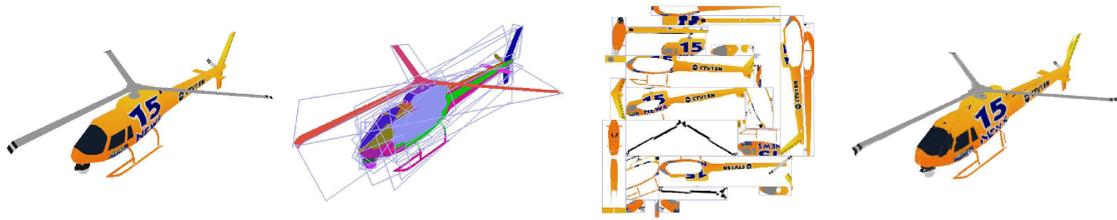


FIG. 2.24 – De gauche à droite : le modèle 3D original composé de 5138 polygones ; la localisation de chaque panneau ; une partie des 32 panneaux texturés mis à plat ; l'objet reconstitué par les différents panneaux. (Images X. Decoret et al. ©2003 ACM)

2.5.6.2 Déformation 3D



FIG. 2.25 – De gauche à droite : texture de l'image et carte de profondeur ; texture de l'image avant et après transformation. (Images M. M. Oliveira 2000)

La technique de déformation 3D est basée sur la modélisation par texturage (*mapping*). La toute première technique de déformation 3D est présentée par J. Blinn (1978). La texture bosselée (*bump mapping*) utilise la texture de l'objet pour simuler l'apparence de déformation sur la surface de l'objet. Comme le lissage de Gouraud, ceci n'est qu'une impression, en effet, la géométrie de l'objet n'est pas déformée, c'est seulement en jouant sur l'intensité de chaque pixel que le rendu donne une impression de déformation. M. Oliveira (2000) regroupe la méthode du *bump mapping* avec celle de la transformation de texture (L. McMillan 1997) pour aboutir à une solution qui permet d'obtenir une réelle déformation géométrique par texture. Cette technique, la texture relief permet de façonner les objets en connaissant leur carte de profondeur. En plus de la texture classique de l'objet, une texture relief est définie de ce même objet. La texture relief va modifier géométriquement l'objet en fonction de l'intensité de cette texture. La texture photométrique de l'objet quand à elle, est transformée de telle sorte à épouser parfaitement la nouvelle surface géométrique de l'objet (Figure 2.25). Ce principe peut être

utilisé pour représenter un objet en 3 dimensions (Figure 2.26). Pour cela il faut 6 textures de relief qui correspondent à une texture de chaque face de la boîte englobante de l'objet. Les deux images pré-transformées à la Figure 2.26 définissent la future texture qui une fois transformée donnera l'illusion de la représentation de la statue selon une nouvelle vue. Vous pouvez le vérifier de vous même, en regardant ce manuscrit de côté, les deux images pré-transformées subissent une transformation complète (comme les deux quadrilatères) pour obtenir l'image finale de la nouvelle vue de la statue.

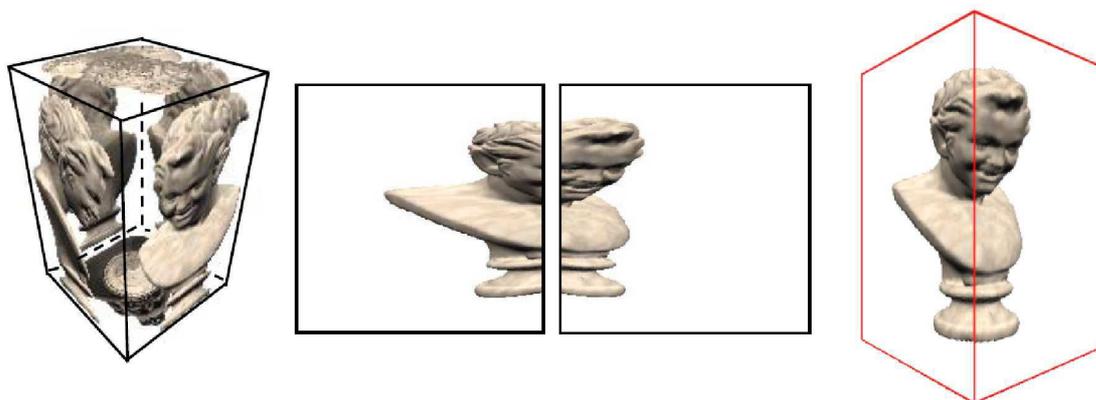


FIG. 2.26 – De gauche à droite : représentation de l'objet par 6 textures de relief ; exemple de pré-transformation de texture ; nouvelle vue texturée à partir de deux quadrilatères de texture transformée. (*Images M. M. Oliveira 2000*)

2.5.6.3 Calque d'image par profondeur

Parallèlement, dans la gestion des textures, il existe par endroit dans l'image résultat un manque d'information dans les images de référence pour certaines parties de la nouvelle vue. Ce problème est lié à l'occultation des objets entre eux. Le problème d'occultation provient du fait que pour un même rayon optique, il existe différentes valeurs d'intensité en fonction de la vue. Ce phénomène appelé aussi phénomène d'ombre, génère des tâches sans information derrière les objets ou personnages de la scène. L. McMillan (1997) propose d'utiliser une transformation de texture pour faire en sorte que les parties non connues de la scène soient calculées. Par le biais de la connaissance de la profondeur de tous les points de la scène, cette méthode donne l'illusion de compléter la nouvelle vue. La transformation de texture permet de supprimer ce problème pour des valeurs d'occultation limitées. J. Shade et al. (1998) proposent une méthode basée sur une représentation de la scène par des calques d'images par profondeur appelé LDI (*Lay-ered Depth Image*). Ils utilisent pour chaque image source une décomposition de l'image en plusieurs calques en fonction de la profondeur de chacun (S. Baker, R. Szeliski et P. Anandan 1998). Un exemple de représentation de calque d'image par profondeur est donné à la Figure 2.27.



FIG. 2.27 – De gauche à droite : image originale ; définition des calques en fonction de leur profondeur ; représentation de la scène sous forme de calques texturés. (Images S. Baker, R. Szeliski et P. Anandan ©1998 IEEE)

La méthode du LDI enregistre les parties visibles des calques pour chaque image mais aussi les parties non visibles en utilisant les autres images source. Au final, la méthode permet de tendre vers une texture complète de chaque calque (dépend fortement du nombre d'images source). Par la suite, la scène est représentée par l'association des différents calques placés en fonction de leur profondeur dans la scène. Maintenant il est possible de connaître l'intensité de chaque pixel sur l'image de rendu, c'est-à-dire sans problème d'occultation par le biais des calques. Une amélioration est proposée par C. Chang, G. Bishop et A. Lastra (1999), qui permet d'utiliser cette fois-ci un ensemble de calques organisé sous la forme d'un arbre hiérarchique. Pour conserver le même échantillonnage que les images de référence, chaque pixel de la nouvelle vue reçoit l'intensité définie en fonction de l'arbre hiérarchique des calques en mémoire. Cette dernière méthode permet de faire des rendus de scènes complexes.

2.5.6.4 Texture par dépendance de vue

La complexité de la texture est là pour augmenter le niveau de photo-réalisme dans les scènes synthétiques ou pour coller parfaitement au modèle réel. Dans le cas de scènes réelles il faut extraire la texture des images de référence, pour les appliquer ensuite sur leurs modèles géométriques. Étant donné que la texture observée dépend de la vue, il faut un certain nombre d'images source pour être garanti de recouvrir complètement le modèle par une texture. Par exemple, P. Debevec, C. Taylor et J. Malik (1996) proposent l'algorithme *Façade* qui permet d'extraire la texture de bâtiment, pour synthétiser des oeuvres architecturales. La méthode consiste à transformer et à composer les différentes textures des images source pour les appliquer sur leur modèle virtuel. Pour cela il faut retrouver manuellement les informations géométriques dans les images source ainsi qu'estimer automatiquement les cartes de profondeur dans la scène pour calculer le modèle de l'objet observé et les textures associées (Figure 2.28). Plus tard, ils proposent un algorithme en trois étapes, diminuant le coût en calcul de l'algorithme en déterminant seulement les polygones visibles de la vue virtuelle à calculer (P. Debevec, Y. Yu et G. Borshukov 1998). Dans une méthode similaire, basée sur la technique du *Lumigraph*, C. Buehler et al. (2001) utilisent un mélange des différentes textures des images source en

fonction de l'angle de position relative, la résolution et le champ de vision de la nouvelle vue.



FIG. 2.28 – De gauche à droite : photographie originale avec marquage des arêtes ; modèle reconstruit ; reprojection du modèles fil de fer sur la photographie ; rendu de l'image de synthèse. (image P. E. Debevec, C. J. Taylor et J. Malik ©1996 ACM)

Pour un maximum de photo-réalisme c'est-à-dire une image le plus fidèle possible à la scène réelle, les techniques de la vision artificielle ne sont pas suffisamment robustes pour extraire des textures qui dépendent de la vue elle même. Les différents phénomènes physiques qui sont difficiles à modéliser sont les effets d'éclairage, les réflexions, les réfractions et les transparences. Par exemple les effets d'illumination d'un objet dépendent de la position et de l'orientation de la caméra virtuelle. C'est pourquoi on s'est intéressé rapidement à développer des textures capable de rendre ces effets particuliers dans le rendu d'images de synthèse. La première technique de rendu d'illumination est le modèle de B. Phong (1975). L'illumination de Phong est un modèle local, c'est-à-dire qui calcule l'intensité en chaque point, et combine trois éléments : la lumière diffuse (modèle lambertien), la lumière spéculaire et la lumière ambiante. Par cette méthode il est possible de calculer pour n'importe quelle vue les effets spéculaires dans l'objet. D. Lischinski et A. Rappoport (1998) proposent d'utiliser une technique de plusieurs calques d'image par profondeur en parallèle pour générer les différents effets d'éclairage dans une scène non diffuse, c'est-à-dire des surfaces de type réflexion spéculaire ou réflexion glacé.

Au niveau de la texture des objets, la gestion des réflexions et des réfractions se réalise par lancer de rayons (*ray tracing*), c'est-à-dire que pour chaque pixel de l'image résultat un rayon lumineux y est associé et parcourt l'espace de la scène en respectant les lois de la physique sur la réflexion et la réfraction optique. Cette technique est très lourde en temps de calcul, car l'image résultante dépend de tous les objets environnants. Il suffit que la scène compte plusieurs objets transparents ou réfléchissants avec plusieurs



FIG. 2.29 – De gauche à droite : les coordonnées de texture de l'image de l'environnement codées sous forme de couleur d'un théière ; rendu final de la théière ; rendu final d'un tore. (*Images W. Heidrich et al. 1999*)

éclairages pour que le temps de calcul de la nouvelle vue explose. W. Heidrich et al. (1999) proposent de dissocier géométrie et illumination de la scène. Les coordonnées des pixels de la texture de l'environnement sont directement calculées et codées sur la texture de l'objet. Cette méthode est basée sur une technique de *light field* (Figure 2.29). Au moment du rendu, cela génère une texture représentant la réfraction de l'objet. L'illumination est ajoutée par un même principe dans un second temps. Cette méthode permet d'obtenir des rendus complexes de réfraction et de réflexion en temps réel.

2.6 Synthèse d'images en microscopie photonique

En microscopie photonique, il est encore très rare de trouver des techniques de rendu à partir d'images. Dans la littérature, il existe quelques techniques comme du mosaïcing en biomicroscopie photonique dans le domaine de l'ophtalmologie (J. Asmuth et al. 2001). Il y a aussi une autre utilisation de mosaïcing en collant bout à bout des images provenant d'un microscope photonique couplé à un miroir pour balayer sur une zone plus importante que son champ de vision (Benjamin Potsaid, Yves Bellouard et John Wen 2002). On peut également trouver une technique de tomographie en sélectionnant les couches par la profondeur de champ, appliquée à de la reconstruction de surface de pièces mécaniques (Jan Wedekind 2004). Récemment, la technique de *light field* est appliquée à un microscope photonique spécialement modifié pour discrétiser les images collectées avec M. Levoy et al. (2006). Malgré le peu de synthèse de nouvelles vues en microscopie photonique, le rendu à partir d'images commence à faire sa place dans ce domaine, poussé par une demande de plus en plus croissante d'améliorer et contrôler les images obtenues.

Les sous-sections précédentes présentaient les trois catégories de la synthèse de nouvelles vues par rendu à partir d'images : le rendu sans géométrie, le rendu avec géométrie implicite et le rendu avec géométrie explicite. Le rendu sans géométrie nécessite une quantité importante d'images source pour calculer une vue virtuelle. Ainsi la technique

du *light field* nécessite quelques gigaoctets de mémoire pour stocker tous les rayons lumineux. Les techniques les plus intéressantes en terme de quantité de données sont celles qui ont une fonction plénoptique avec le moins de dimension. Contrairement au rendu sans géométrie, celles avec géométrie implicite nécessite peu de vues, deux au minimum. Par contre la qualité de la vue retrouvée dépend de la précision de la mise en correspondance de chaque pixel dans les images de référence. De plus, pour obtenir une image de qualité complète, une correspondance dense et des techniques de *ray tracing* sont nécessaires. Quand au rendu avec géométrie explicite une connaissance du modèle est parfois nécessaire, si ce n'est pas le cas, plusieurs images source sont nécessaires afin de reconstruire l'information de texturage et de géométrie des différents objets de la scène. Cette dernière catégorie est orientée transformation et composition de texturage d'objets complexes.

D'après les contraintes évoquées dans le paragraphe *Spécificités d'un microscope photonique* (§2.3), ils nous est impossible de placer plusieurs microscopes photoniques dans une station. Par conséquent, notre stratégie s'oriente sur une limitation à 2 vues passives, c'est-à-dire fixes dans le système, et 1 vue active, c'est-à-dire pilotée dans l'espace par des axes motorisés. Cela permet d'avoir un système de vision distribué à la fois en temps réel (par les 2 vues passives) et à la fois en temps déporté (par balayage de la vue active). Différentes configurations peuvent être envisagées pour le système distribué temps réel, soit un système éclaté ou monobloc. Un système éclaté correspond à plusieurs microscopes photoniques compacts, comme plusieurs TIMM placés dans la station. Un système monobloc correspondrait soit à une binoculaire stéréoscopique ou un microscope photonique stéréoscopique, c'est-à-dire un bloc compact générant deux vues différentes par dissociation de chemin optique.

Deux méthodes dans les techniques du rendu à partir d'images semblent intéressantes par rapport à notre problématique. La première permettrait d'augmenter virtuellement le champ de vision du microscope photonique. Le *mosaïcing* présente des avantages. Tout d'abord sa fonction plénoptique n'est que de dimension 2 ce qui limite le nombre d'images source nécessaire au rendu ainsi que le stockage de l'information. Ensuite par le fait d'utiliser une vue active, c'est-à-dire en balayage, on pourrait augmenter le champ de vision sans perdre de résolution si précieuse pour la micromanipulation. L'utilisation de cette technique permettrait d'avoir une image globale de l'espace de travail avec une grande résolution, et ainsi pouvoir centraliser toutes les informations dans un même référentiel en vue de superviser une station de micromanipulation.

La deuxième technique qui semble pertinente est le transfert trifocal. Dans un contexte où les microscopes photoniques stéréoscopiques sont de plus en plus présents, il semble intéressant d'utiliser cette caractéristique encore très peu exploitée. Parmi les différentes techniques de rendu par géométrie implicite, qui ne nécessitent que deux vues, le transfert trifocal est le plus adapté au système de vision utilisé. Les techniques telles que le morphage de vue ou la vue par triangulation commune peuvent être classées comme techniques d'interpolation c'est-à-dire que la nouvelle vue est limitée dans l'espace intérieur des deux vues de référence. Ce qui ne correspond pas au type de système de vision que l'on utilise, car généralement l'angle entre les vues d'un stéréo-microscope photonique est seulement de quelques degrés (pour la binoculaire Leica de la station

SAMMI l'angle est de 10°). Seule la méthode de transfert propose des nouvelles vues par extrapolation c'est-à-dire éloignées et en dehors de l'espace intérieur des deux vues de référence. La technique de transfert trifocal est plus robuste que les autres techniques en particulier le transfert épipolaire qui présente des dégénérescences dans l'image résultat en fonction de certaines positions singulières de la nouvelle vue, voir à cet effet le comparatif de A. Habed et B. Boufama (2000) et de Keith Connor et Ian Reid (2002). La technique du transfert trifocal permettra de libérer l'espace de travail de la station. A partir de seulement deux vues de dessus, nous serons dans la capacité de reconstruire les différentes vues utiles au pilotage de la station.

2.7 Conclusion

Dans un premier temps nous avons listé quelques papiers issus de la littérature qui présentent des exemples concrets d'utilisation de systèmes de perception dans des tâches de micromanipulation et de micro-assemblage. De cette liste découle deux constats. Le premier constat est le suivant : pour une supervision et un contrôle total de la station de micromanipulation, un système de perception visuel distribué est nécessaire (vue locale, vue globale, vue de dessus, vue de côté, etc). Le deuxième constat est la présence forte du microscope photonique dans la plupart des applications. Il traduit le besoin de grande précision des vues pour la micromanipulation, avec un encombrement important mais acceptable. Dans un deuxième temps, ce chapitre a mis en évidence les problèmes liés aux propriétés des microscopes photoniques : encombrement important, faible distance de travail, faible profondeur de champ, faible champ de vision et forte dépendance à l'illumination. Ces propriétés entraînent des contraintes qui justifient et renforcent le fait d'utiliser un système de perception distribué. Un système multi-vues permet de fusionner les informations complémentaires perçues dans la station. Dans un dernier temps, nous avons exploré un domaine de reconstruction $2D$ de la vision appelé la synthèse de nouvelles vues. De cet aperçu nous proposons d'utiliser et d'améliorer deux techniques de ce domaine particulier. Le mosaicing, une technique sans géométrie, permettra d'augmenter le champ de vision d'un microscope photonique. La deuxième technique, le transfert trifocal, permettra quand à lui, de reconstruire n'importe quelle vue de l'espace à partir d'un stéréo microscope photonique. Le but étant d'alléger la structure distribuée du système de vision et ainsi de libérer l'espace de travail. Cette dernière technique utilise de la géométrie implicite dans la scène. Cela signifie qu'une étape de calibrage est nécessaire à son fonctionnement. Le chapitre suivant développera la méthode classique du calibrage faible d'un stéréo microscope et en parallèle nous présenterons nos apports et nos améliorations.

Auto-Calibrage Stéréoscopique Faible en Microscopie Photonique

Ce chapitre est dédié à la présentation du calibrage. Dans un premier temps nous expliquerons ce que signifie l'auto-calibrage d'un système de vision et pourquoi il est appelé faible. Nous présenterons le procédé que nous avons élaboré en vue de l'appliquer à un système de vision par microscopie photonique. Le procédé comprend des algorithmes classiques de la vision artificielle combinés avec de nouveaux algorithmes que nous avons développés.

3.1 Introduction

Parmi les deux méthodes que nous avons choisi d'exploiter, la construction de mosaïque et le transfert trifocal, une seule comporte une étape de calibrage. Le transfert trifocal, utilise de l'information géométrique implicite dans la scène par le biais d'un système de vision stéréoscopique. Ce chapitre sera entièrement consacré au calibrage de ce système. Par contre, certains des algorithmes de vision développés ici, sont communs aux deux techniques de synthèse de nouvelles vues. Par exemple, la méthode pour détecter des points d'intérêt dans une image est la même dans la construction de mosaïque que dans le transfert trifocal. C'est pourquoi ce chapitre est une manière de présenter les algorithmes qui accompagneront la lecture des autres chapitres, comme une sorte de boîte à outils, par le biais de l'application du calibrage du système de vision pour le transfert trifocal. Ce chapitre présentera les étapes successives du calibrage de façon linéaire. Dans les sections *détection d'indices* et *mise en correspondance* nous proposerons parallèlement notre apport avec une amélioration de la détection d'indices et une nouvelle méthode de mise en correspondance.

Le transfert trifocal utilise un système de vision stéréoscopique binoculaire, c'est-à-dire un système de perception visuelle distribué composé de deux vues. Le calibrage stéréoscopique faible, communément appelé, le calibrage faible consiste à déterminer la géométrie relative entre ces deux vues, c'est-à-dire, les relations de projection du système

de deux vues qui regarde la même scène (S. Laveau 1996). Cette géométrie particulière est appelée géométrie épipolaire. Elle est déterminée par une étape de calibrage. Ce calibrage est qualifié de faible parce qu'on estime simplement la relation relative entre les plans projectifs ($\mathcal{P}^2 \rightarrow \mathcal{P}^2$), c'est-à-dire d'image à image, par opposition au calibrage fort, qui lui détermine la relation absolue entre l'espace projectif et le plan projectif de chaque vue ($\mathcal{P}^3 \rightarrow \mathcal{P}^2$), c'est-à-dire le modèle complet de chaque vue dans l'espace de la scène (C. Zeller 1996). Ce modèle exprimé sous forme d'une matrice de projection perspective comprend deux sous-ensembles appelés paramètres intrinsèques et paramètres extrinsèques. Les paramètres intrinsèques représentent l'application linéaire de l'espace projectif vers le plan projectif exprimant la transformation perspective. Les paramètres extrinsèques représentent une transformation rigide (3 rotations et 3 translations) entre le repère de la scène et le repère du plan image. Dans le cas où le système est stéréoscopique, on connaît parfaitement la géométrie absolue des deux vues dans l'espace projectif. Le calibrage peut être aussi qualifié d'auto-calibrage, cela indique simplement que nous n'avons pas la connaissance dimensionnelle sur les objets observés de la scène. Dans le cas du calibrage faible cette information n'est pas utilisée, on ne connaît ni la taille ni la géométrie des objets. Par contre, pour un calibrage fort, on utilise couramment des mires de calibrage qui représentent des formes géométriques simples (carré, disque, etc.) avec des dimensions connues, par exemple chaque disque fait x mm de diamètre et sont séparés par d mm. Ce type de calibrage est très difficile à mettre en œuvre en microscopie, pour deux raisons. La première est la nécessité de fabriquer des mires aux dimensions parfaitement connues (utilisant des technologies microfabrication) et la deuxième est que le calibrage fort complet nécessite plusieurs images avec différentes postures des mires de calibrage dans l'espace, ce qui est quasiment impossible avec des moyens simples de micromanipulation. C'est pour cette raison précise que les techniques de synthèse de nouvelles vues sont orientées vers des méthodes de calibrage faible.

Le calibrage faible se résume par la détermination de la géométrie épipolaire. Cette géométrie, dans le cas d'un calibrage faible est caractérisée par une matrice appelée fondamentale (H. Longuet-Higgins 1981). Si le calibrage était partiellement fort, c'est-à-dire utilisant seulement la connaissance des paramètres intrinsèques de la caméra, cette géométrie serait caractérisée par une matrice appelée essentielle (Q.-T. Luong et O.D. Faugeras 1996), qui n'est autre que la matrice fondamentale calibrée. Pour la présentation du calibrage faible, la section suivante sera consacrée à la géométrie épipolaire. Comme celle-ci détermine la relation géométrique relative des points dans chaque vue, pour la déterminer, nous allons étudier la position des couples de points dans chaque vue. En déterminant les points d'intérêt dans chaque image puis en trouvant leur correspondance, on peut remonter à la relation entre chaque point et donc à la géométrie épipolaire.

3.2 Géométrie Epipolaire

La géométrie épipolaire définit la relation géométrique relative entre deux vues. Prenons un système de vision stéréoscopie binoculaire, c'est-à-dire à deux vues représenté à la Figure 3.1. Les deux vues $\psi\{O, \psi_I\}$ et $\psi'\{O', \psi'_{I'}\}$ sont respectivement la vue de

gauche et la vue de droite. Chaque vue a son centre optique noté O pour la vue ψ et O' pour la vue ψ' , le segment passant par ces deux centres est la ligne de base du système stéréoscopique. Par convention mathématique, les centres optiques sont placés derrière les plans image alors que dans la réalité ils sont placés devant. Chaque vue a également son plan image ψ_I pour la vue ψ et ψ'_I pour la vue ψ' .

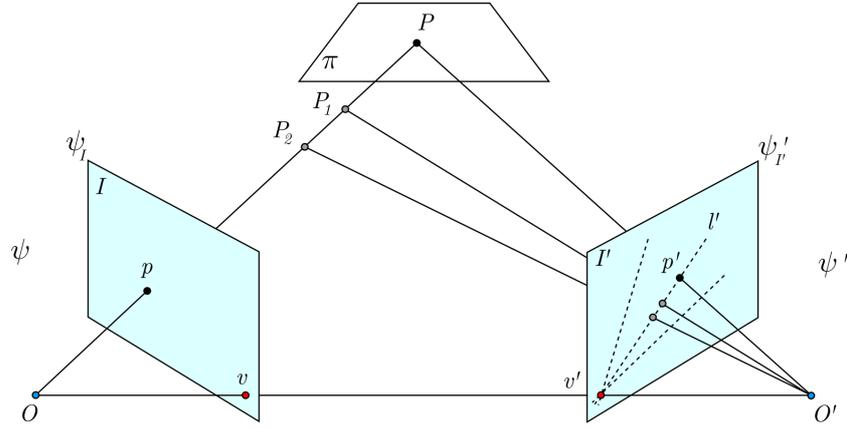


FIG. 3.1 – Géométrie épipolaire

Soit un point de l'espace projectif \mathcal{P}^3 noté P appartenant à un plan π . Le point P se projette à travers la première vue ψ en passant dans le plan projectif \mathcal{P}^2 selon le rayon optique $[PO]$ contraint par le centre optique O . De cette projection en découle un point p sur le plan image ψ_I . Ce même point P est également observé par la vue ψ' . Il se projette à travers ψ' selon le rayon optique $[PO']$ contraint par le centre optique O' . Le point P se projette donc dans le plan image ψ'_I en un point noté p' . Étant donné que le point P suit le rayon optique $[PO]$, il existe un ensemble de points P_1, P_2, \dots qui correspond à la même projection dans le plan image ψ_I . Cet ensemble de points P_i se projette également dans le plan image ψ'_I , passant ainsi par une ligne qui correspond à la projection du rayon optique $[PO]$. Cette ligne l' est appelée ligne épipolaire. Elle représente l'existence de la dualité point/ligne des deux vues : l'homologue d'un point p de l'image I de la vue ψ se trouve sur une ligne connue dans l'image I' de la vue ψ' . La réciproque est vraie, le point p' a aussi son dual qui se trouve sous forme d'une ligne épipolaire dans I . Chaque point d'une image subit une transformation par un morphisme projectif, car il y a changement de forme (point/ligne) au moment de la projection. Ce morphisme projectif s'exprime par une matrice appelée matrice fondamentale F de dimension 3×3 et de rang 2. Si pour une paire de vues pour laquelle la matrice fondamentale est connue alors on parle de calibrage faible. On peut déterminer chaque ligne épipolaire par cette matrice :

$$l' = Fp \quad l = F^T p' \quad (3.1)$$

avec l la ligne épipolaire dans l'image I . Cette relation décrit une contrainte entre le point p et son correspondant p' , car p existe sur l' et p' existe sur l , c'est-à-dire que le

plan (p, p', O, O') définit les deux lignes épipolaires. Ce plan traduit l'action de la matrice fondamentale qui satisfait une contrainte algébrique coplanaire appelée contrainte épipolaire, et définie par :

$$p'^T F p = 0 \quad (3.2)$$

Cette contrainte est une équation de forme bilinéaire, c'est-à-dire qu'elle dépend d'un système d'équations linéaires comprenant deux points qui sont p et p' . Maintenant supposons un objet composé d'un ensemble de points P de l'espace projectif \mathcal{P}^3 , ils n'appartiennent pas tous au même plan π . Chaque point P projeté dans le plan image ψ_I trouvera son dual dans un ensemble de lignes épipolaires dans le plan image $\psi'_{I'}$. Ces lignes provoquées par le faisceau de plans qui coupe le plan image $\psi'_{I'}$, se croisent en un même point appelé épipole v' . Cet épipole est le pivot de rotation des différentes contraintes épipolaires des faisceaux de plan (p, p', O, O') . Physiquement, l'épipole représente la projection du centre optique de O dans le plan image $\psi'_{I'}$. Inversement, l'épipole v est la projection du centre optique O' dans le plan image ψ_I . L'épipole est directement calculée par une décomposition en valeur singulière de la matrice fondamentale F :

$$F = U \Sigma V^T \quad (3.3)$$

avec Σ les valeurs singulières, (u_1, u_2, u_3) les vecteurs colonnes de U et (v_1, v_2, v_3) les vecteurs colonnes de V . L'épipole v' correspond au vecteur u_3 et l'épipole v au vecteur v_3 . Mathématiquement, les épipoles sont les noyaux de la matrice fondamentale car ils résolvent les équations :

$$F v = 0 \quad F^T v' = 0 \quad (3.4)$$

Connaissant la position de p dans I et la géométrie épipolaire, on peut retrouver la position du point p' dans I' . Deux cas de figure peuvent se présenter en fonction de l'appartenance ou non des points P de l'espace projectif \mathcal{P}^3 à un même plan π .

Si chaque point P appartient au plan π , il existe une relation géométrique entre le p de I et le p' de I' . Par le biais de la transformation entre les deux plans images ψ_I et $\psi'_{I'}$, on peut déterminer la position de p' à partir de p et vice versa. Cette transformation géométrique est de type projective, et s'exprime par une matrice de transformation projective dans \mathcal{P}^2 appelée matrice de colinéation G de dimension 3×3 . Si p et p' sont les projections du point P appartenant à π alors :

$$p' \sim G p \quad (3.5)$$

où le signe \sim signifie que l'égalité est définie à un facteur d'échelle λ près. Maintenant dans le cas où chaque point P n'appartient pas au plan π , la projection doit être rectifiée par le parallaxe. Le parallaxe projectif est constitué de l'épipole et du parallaxe relatif. Le parallaxe relatif est la distance de la normale du point P par rapport au plan π , c'est en quelque sorte la profondeur relative par rapport au plan π . Dans ce cas le point p' s'écrit en fonction de p :

$$p' \sim Gp + \delta v' \quad (3.6)$$

avec le terme $\delta v'$ le parallaxe projectif et δ le parallaxe relatif. Les épipoles sont des points particuliers dans les plans images, cela dit il n'en reste pas moins des points, c'est-à-dire que les épipoles v et v' sont aussi reliées par la transformation projective de la matrice de colinéation G , on a :

$$v' \sim Gv \quad (3.7)$$

De même il existe aussi une relation entre la matrice fondamentale et la matrice de colinéation faisant intervenir les épipoles. La matrice fondamentale peut s'écrire ainsi :

$$F \sim [v']_{\times} G \quad (3.8)$$

avec l'opérateur matriciel de symétrie croisée $[*]_{\times}$ qui traduit le produit vectoriel (voir glossaire des notations).

Le calibrage faible doit nous permettre de connaître la géométrie épipolaire du système, et la matrice fondamentale en est la clé de voûte. Réaliser le calibrage faible d'un système de vision stéréoscopique binoculaire se résume à déterminer la matrice fondamentale. Pour cela on utilise un ensemble de points p et p' pré-établi pour remonter à F . Parfois la détermination des points avec exactitude s'avère difficile, c'est pourquoi bien souvent on utilise un objet qui fait office de mire pour aider à extraire ces points.

3.3 Mire de calibrage par objet naturel

La détermination de la matrice fondamentale repose sur la connaissance d'un ensemble de couples de points (p, p') . Pour cela, la méthode consiste à utiliser les objets de la scène. On extrait des points d'intérêt sur les objets dans chaque image I et I' , cela peut être des coins, des arêtes, ou encore des intersections d'arêtes. Ensuite pour chaque point on va déterminer son jumeau dans l'autre image, c'est ce qu'on appelle la mise en correspondance. Par exemple, si le point p est le coin d'un objet dans l'image I on doit déterminer le point p' qui doit être positionné sur le même coin de l'objet dans l'image I' . C'est pourquoi les points d'intérêt sont qualifiés d'indices invariants de la scène. Généralement en calibrage classique on utilise des mires damiers perpendiculaires. Un des problèmes en microscopie photonique avec des objets quelconques, c'est qu'ils se prêtent difficilement à la mise en correspondance de points. Leurs images sont faiblement texturées avec des formes géométriques répétitives. Ce qui implique qu'on détecte un faible nombre de points d'intérêt et que le peu de texture rend difficile la comparaison entre les points pour la mise en correspondance. De plus la qualité de la matrice fondamentale dépend de l'exactitude de la mise en correspondance des points. Un autre problème est l'échantillonnage des points détectés, la matrice fondamentale traduit la géométrie épipolaire pour des points dans l'espace. Cela signifie que l'estimation de l'ensemble des couples de points pour la calculer nécessite d'avoir un échantillonnage dans l'espace uniformément distribué. Or, si nous analysons une image en utilisant des

points détectés sur une pièce artificielle de micro-assemblage nous avons un ensemble de points appartenant au dessus de la pièce et un autre à la profondeur du plan de travail, ce qui, pour une estimation de F de qualité est loin d'être suffisant. L'idéal est d'avoir un ensemble de couples de points parfaitement répartis dans l'espace et surtout appartenant à plusieurs profondeurs.

Dans une station de micromanipulation le calibrage est quelque chose de délicat, qu'il se rapporte à la robotique (G. Li et N. Xi 2002) ou à la vision (J. Feddema et T. Christenson 1999). Dans le cas du calibrage des imageurs photoniques il est impossible d'utiliser les traditionnels damiers perpendiculaires. Une méthode consiste à utiliser le préhenseur de micromanipulation pour créer une mire virtuelle dans l'espace (M. Ammi, V. Fremont et A. Ferreira 2005). Un des inconvénients majeurs de cette méthode est que pour obtenir un calibrage de qualité, un nombre important de points est nécessaire, or déterminer par exemple 200 points par le biais de l'extrémité d'une pince de micromanipulation devient un procédé fastidieux et long. Y. Zhou et B. Nelson (1999) proposent un calibrage fort d'un microscope photonique en utilisant une mire de calibrage réalisée en microtechnologie. Elle est constituée d'une grille de carrés sur une plaque de verre, chaque carré fait $10 \mu m$ de côté. Du fait que la mire est parallèle au plan de travail (aucun point en profondeur) le calibrage est simplifié par des approximations. Par contre, pour le calibrage d'un stéréo microscope, la difficulté est différente. La géométrie de deux vues, traduit le parallaxe et la profondeur des points dans la scène, c'est-à-dire que pour ce calibrage faible on ne peut se passer de points en profondeur, l'artifice de travailler avec une mire plane est impossible.

Il nous faut déterminer un objet qui fera office de mire d'auto-calibrage. Cette mire idéale doit satisfaire certaines conditions :

- elle doit être texturée, pour permettre d'obtenir suffisamment d'informations photométriques et déterminer la correspondance de chaque point par corrélation,
- elle ne doit pas avoir de motif ou de texture répétitif, pour éviter le risque d'ambiguïté pendant la mise en correspondance,
- elle doit être de taille acceptable pour être observée en microscopie photonique,
- elle doit comporter deux zones distinctes, une zone plane qui va permettre l'estimation de la matrice de colinéation, et une zone avec du relief pour l'estimation de la matrice fondamentale.

Nous proposons une mire qui répond parfaitement à ces attentes et à ces conditions : cet objet est une goutte d'eau recouverte de limaille de nickel. Quelques exemples de gouttes d'eau sont présentées à la Figure 3.2.

La méthode de réalisation est simple, il suffit de lâcher une goutte d'eau dans un plan de travail recouvert de limaille de nickel. Par capillarité et tension de surface, la limaille vient recouvrir complètement la goutte d'eau, en fonction de sa taille, on peut obtenir une sphère complète délicatement posée dans l'espace de travail. Le système de dépose se fait à la main, ce qui implique qu'il est très difficile de contrôler la taille des gouttes. On obtient des gouttes d'eau de $1 mm$ à $300 \mu m$, ce qui est suffisant pour la catégorie de stéréo microscope photonique habituellement utilisé en micromanipulation. La limaille de nickel procure une texture riche et non répétitive, on est proche des techniques de mouchetage d'objet utilisées dans certains cas de reconstruction 3D. La mire repose sur

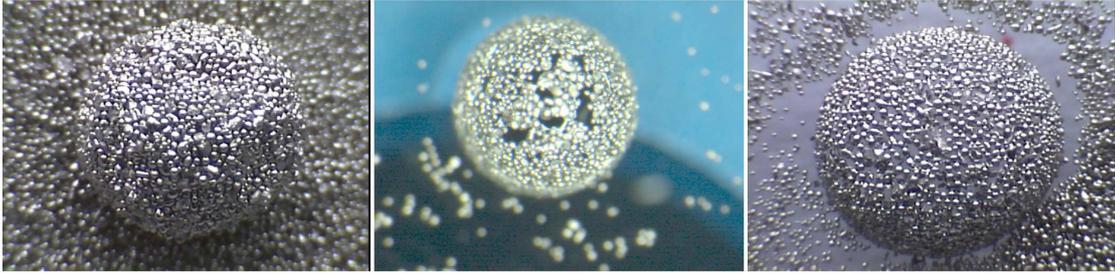
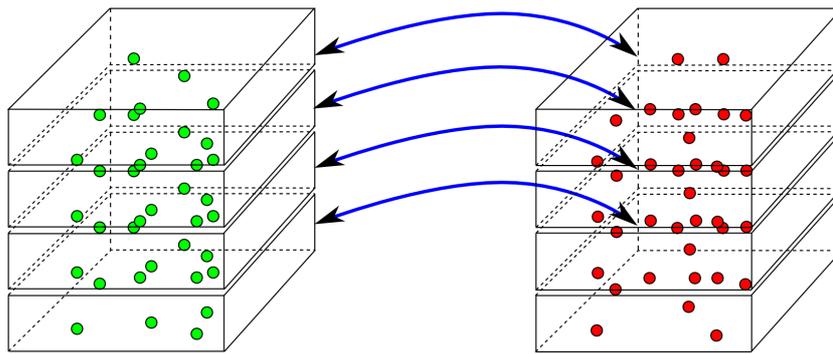


FIG. 3.2 – Exemple de mires de calibrage pour stéréo microscope.

un lit de limaille, ce plan permettra d'obtenir la matrice de colinéation. Pour la matrice fondamentale nous pourrions utiliser la géométrie sphérique ou elliptique (cela dépend de la taille de la goutte) pour obtenir des points à différentes profondeurs dans la scène.

3.4 Acquisition d'images par couche

FIG. 3.3 – Détection et mise en correspondance de l'ensemble des points p et p' séparément par couche de profondeur de champ.

L'utilisation d'une mire avec un fort relief pose un problème pour l'acquisition des images. En microscopie photonique, la profondeur de champ est faible, il est alors impossible d'observer la mire sur toute sa profondeur en une seule fois. Pour contourner ce problème nous allons utiliser une méthode d'acquisition par couche, c'est-à-dire que l'on va balayer la scène dans sa profondeur. En faisant bien correspondre les volumes de chaque profondeur de champ, il est possible de minimiser le nombre d'acquisitions. Pour chaque couche on capture les images gauche et droite du stéréo microscope photonique. La stratégie consiste à déterminer les points d'intérêt et à réaliser la mise en correspondance pour chaque couple de couche (Figure 3.3). Le fait d'isoler les couches pendant la mise en correspondance permet de limiter les risques d'erreur d'appariement. Par exemple, un point de la dernière couche ne peut être confondu avec un des points de la première couche. Cela nous permet de rajouter une contrainte (profondeur des points)

pendant la mise en correspondance. Pour notre exemple de calibrage nous prenons trois images à différentes profondeurs d'une mire de $300\ \mu\text{m}$ de diamètre, ce qui suffit pour obtenir toute la profondeur de la goutte d'eau (Figure 3.4).

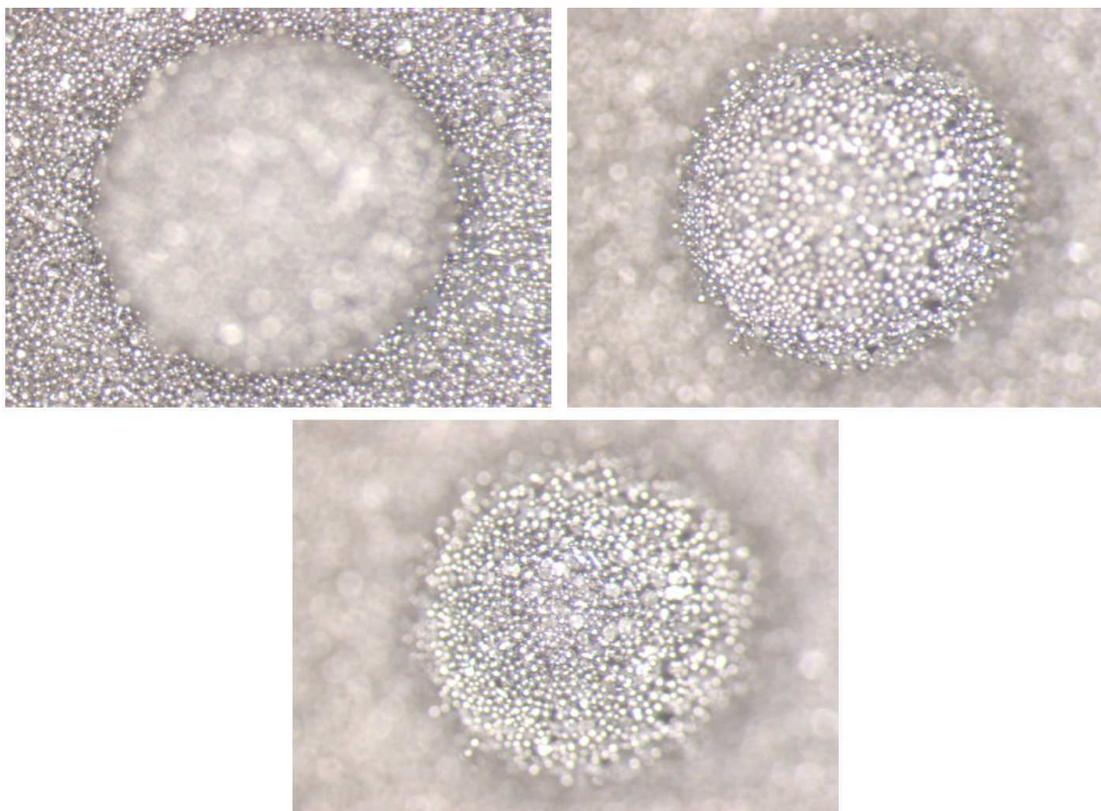


FIG. 3.4 – Les trois images de notre mire de $300\mu\text{m}$ choisie pour le calibrage avec seulement 3 couches (il n'y a que les images de droite représentées).

3.5 Détecteur de points d'intérêt

Après l'étape d'acquisition en couche, il faut déterminer pour chacune d'entre-elles les points d'intérêt entre l'image de gauche et l'image de droite. Ces points seront utilisés pour la mise en correspondance et la détermination des différentes informations nécessaires au calibrage. Leur particularité est de détecter un point susceptible d'être d'intérêt c'est-à-dire contenant de l'information intéressante sur la topologie de l'objet ou de la scène, c'est par exemple les coins ou les arêtes des objets. Ils sont utilisés dans de nombreux domaines de la vision, tels que le suivi de cible, la reconnaissance, le calibrage, etc. Le premier détecteur de points (*corner detector*) fut présenté par H. Moravec (1977). Aujourd'hui il existe de nombreux détecteurs dans la littérature mais seulement trois d'entre eux sont populaires. Le détecteur de Harris de C. Harris et M. Stephens

(1988) est basé sur une fonction d'auto-corrélation approximée par une dérivée au premier ordre de l'image. La base du détecteur de Harris repose sur le détecteur de Moravec. Le détecteur SUSAN (*Smallest Univalued Segment Assimilating Nucleus*) de S.M. Smith et J.M. Brady (1995) est basé sur un type de convolution avec un masque circulaire, il peut très bien détecter des coins comme des arrêtes. Le dernier détecteur de Farzin Mokhtarian et Riku Suomela (1998) est le CSS (*Curvature Scale Space*) qui est basé sur le détecteur de contours de Canny (J. Canny 1986). Comme le montre C. Schmid, R. Mohr et C. Bauckhage (1998) le détecteur de Harris est robuste au changement d'illumination. C'est donc ce dernier que nous allons utiliser.

3.5.1 Détecteur de Harris

Le détecteur de Harris présenté par C. Harris et M. Stephens (1988) est basé sur une fonction d'auto-corrélation, c'est-à-dire qu'il analyse les changements d'intensité lumineuse dans l'image. En réalité, on utilise une approximation bilinéaire de la fonction d'auto-corrélation par le biais de petites variations autour de chaque pixel. Supposons le changement d'intensité et sa variation spatiale (u, v) , la fonction d'auto-corrélation E peut s'écrire :

$$E(u, v) = \sum_x \sum_y W(x, y) [I(x + u, y + v) - I(x, y)]^2 \quad (3.9)$$

avec $W(x, y)$ la fenêtre de corrélation de coefficients rectangulaire ou Gaussienne et I l'intensité de l'image. Pour une petite variation de (u, v) nous avons une approximation bilinéaire de la forme :

$$E(u, v) \cong (u, v) M \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.10)$$

où M est la matrice 2×2 calculée à partir de la dérivation de l'image I limitée aux premiers termes :

$$M = \sum_x \sum_y W(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (3.11)$$

avec I_x et I_y les dérivées partielles suivant respectivement l'axe x et y :

$$I_x = \frac{\partial I(x, y)}{\partial x} \quad I_y = \frac{\partial I(x, y)}{\partial y} \quad (3.12)$$

Pour tous les pixels de l'image I une approximation bilinéaire M est calculée. De cette approximation on en déduit la réponse du détecteur de Harris par l'équation suivante :

$$R = \det M - k (\text{trace } M)^2 \quad (3.13)$$

où, R est la réponse du détecteur, $\det M = \lambda_1 \lambda_2$ et $\text{trace } M = \lambda_1 + \lambda_2$ avec λ_1 et λ_2 les valeurs propres de la matrice M . k est une constante définie de façon empirique et égale à 0,04 après différents essais. En fonction de la réponse du détecteur et des valeurs

propres, il est possible de déterminer si la région analysée par la fenêtre est un sommet ($R > 0$), une arête ($R < 0$) ou une zone plate ($R \sim 0$), comme le montre la Figure 3.5

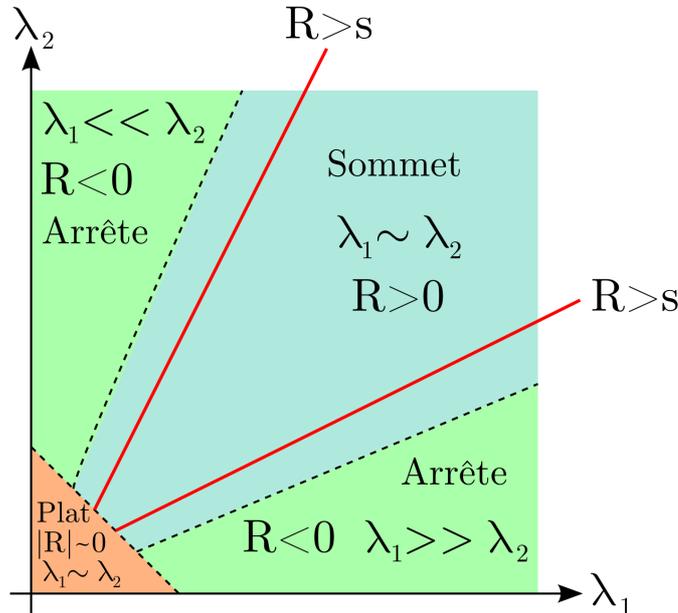


FIG. 3.5 – Définition d'un sommet, d'une arête ou d'une zone plate en fonction de la réponse du détecteur et des valeurs propres.

Il est possible de régler de façon empirique le nombre de points détectés en introduisant un seuil dans la réponse du détecteur. Concrètement l'utilisation du détecteur de Harris est très simple. Après avoir calculé les différents gradients I_x^2 , I_y^2 et I_{xy} sur toute l'image avec un passage par une fenêtre gaussienne, l'algorithme de la réponse du détecteur se résume en une ligne, par l'équation (3.13). Il ne reste plus qu'à récupérer les points par seuillage de l'image de la réponse du détecteur. L'illustration de la Figure 3.6 propose un exemple de détection par la méthode Harris sur une des images du calibrage avec une goutte d'eau. La figure montre l'image originale avec certaines parties floues, ce sont celles qui ne sont pas dans la profondeur de champ. Elle montre également l'image de la réponse du détecteur. C'est à partir de tous les maximums de cette réponse que l'on détermine de façon empirique le nombre de points désiré par seuillage. La dernière image de la figure représente l'image après seuillage, chaque point représenté est un point d'intérêt détecté.

3.5.2 Détecteur Harris Simplexe

Le calibrage nécessite un nombre de points minimum pour qu'il soit de qualité. Le problème avec le détecteur de Harris, c'est que le nombre de points détecté N_{pd} est dépendant, naturellement du seuil s , plus il augmente et plus le nombre de points détecté diminue. Il dépend également de la réponse du détecteur, c'est-à-dire qu'en fonction de

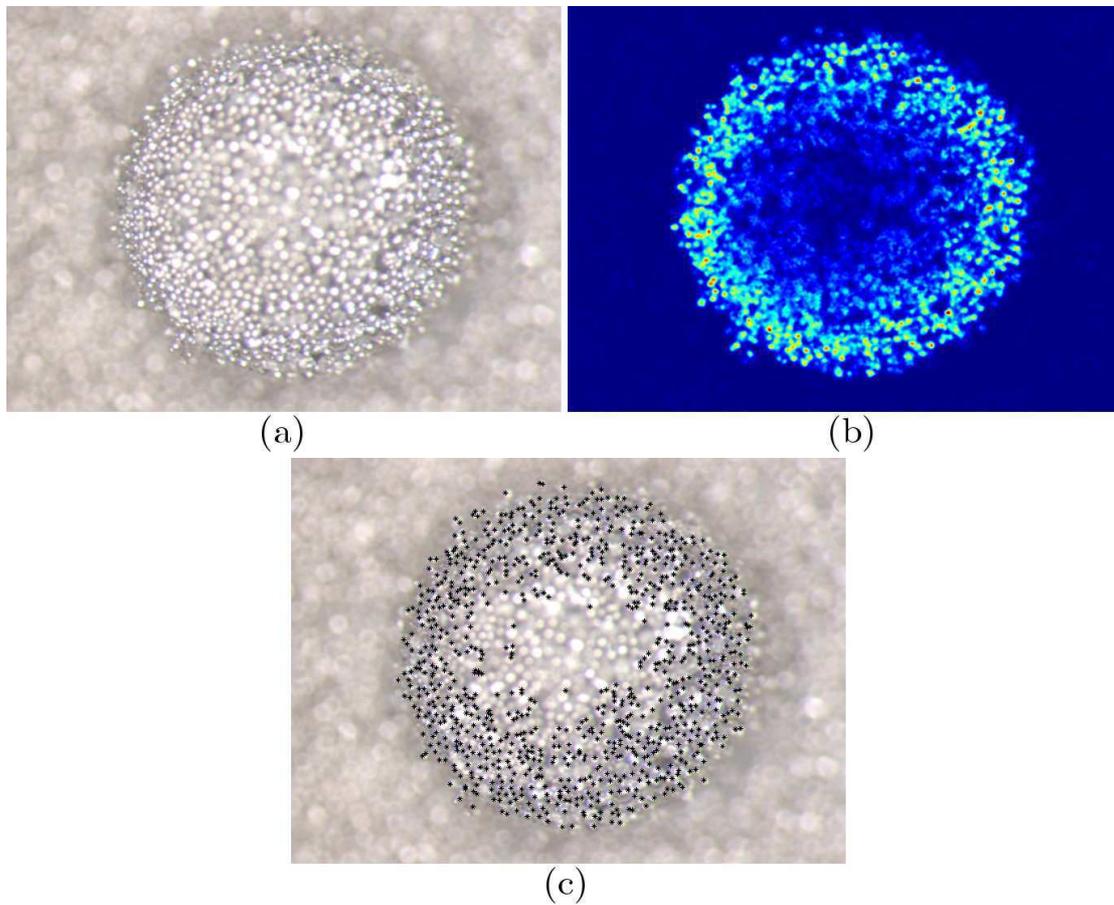


FIG. 3.6 – Exemple de détection de points par le détecteur de Harris : (a) une des images de la mire de calibration, (b) image résultat de la réponse du détecteur (en fausse couleur), (c) et résultat des 907 points détectés sur l'image originale ($s = 0,02$).

l'image analysée, pour un même seuil, le nombre de points diffère. Nous souhaitons que le calibration soit automatisé, donc la détection de point d'intérêts doit être mise en œuvre sur plusieurs images. Or dans le cas de ce détecteur on ne peut pas garantir un nombre minimum de points. C'est pour cette raison que nous proposons de modifier le détecteur Harris afin de permettre de déterminer *a priori* le nombre de points désiré Np^* dans une image. Tout le problème réside dans la détermination de la valeur du seuil s^* qui nous donne le nombre de points désiré Np^* . Pour déterminer le seuil à sélectionner nous choisissons une méthode d'optimisation. Le nombre de point Np_d est une fonction de la réponse du détecteur : $Np_d = f(R(s))$. La comparaison entre le nombre de points désiré Np^* et le nombre de points détecté $f(R(s))$ est une fonction de la forme exponentielle décroissante en fonction de s . Pour déterminer s^* nous choisissons de minimiser l'erreur entre Np^* et Np_d ; comme elle est de la forme exponentielle décroissante et qu'il n'y a pas de minimum, une astuce mathématique, par le biais de la valeur absolue, permet de

créer un minimum local par rapport au nombre de points désiré. Au final nous devons minimiser la fonction suivante :

$$|Np^* - f(R(s))| \quad (3.14)$$

Pour effectuer la minimisation de cette fonction nous allons utiliser une méthode d'optimisation par simplexe de type Nelder-Mead (J. Nelder et R. Mead 1965). Cette méthode compare la valeur d'une fonction objectif avec zéro et ne nécessite aucune dérivation de la fonction contrairement aux méthodes classiques. Un simplexe de dimension n est un polytope de dimension n , c'est-à-dire une enveloppe convexe de $n + 1$ sommets formant un repère affine dans un espace euclidien \mathbb{R}^n . Par exemple, un triangle est un simplexe de dimension 2. Dans la méthode de Nelder-Mead, le simplexe change de forme d'itération en itération par réflexion, expansion, contraction interne et contraction externe. En fonction de l'erreur entre Np^* et Np_d le simplexe va chercher directement le seuil correspondant. Pour optimiser la rapidité de la recherche, la réponse du détecteur R propre à l'image est calculée une seule fois. La détermination des points dans R ne dépend que du seuil. La Figure 3.7 montre le schéma bloc de la boucle de recherche du détecteur Harris simplexe.

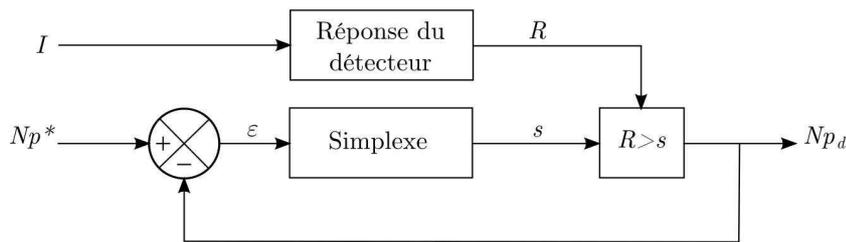


FIG. 3.7 – Le schéma bloc du détecteur Harris simplexe.

Par exemple, avec une des images de notre mire, le détecteur de Harris déterminait 907 points pour un seuil $s = 0,02$. Avec le détecteur Harris simplexe pour la même image, nous lui donnons le nombre de points que nous désirons, par exemple $Np^* = 200$ points, le simplexe détermine automatiquement le seuil à $s = 0,05725$. Une des contraintes de la méthode du simplexe et de l'initialiser au voisinage de la solution pour garantir sa convergence, pour compenser ce problème nous rajoutons une boucle d'incrément sur le seuil, si le simplexe ne trouve pas de solution autour de ce seuil, l'incrément se déplace et un nouveau simplexe est initialisé. On peut résumer le Harris simplexe par cet algorithme :

```

I : Image
Np* : Nombre de points désiré
k : Coefficient empirique de valeur 0,04

 $R \leftarrow (I_x^2 \times I_y^2 - I_{xy} \times I_{xy}) - k \times (I_x^2 + I_y^2)^2$ 
 $s \leftarrow 0$ 
Tant que (Solution= 0) faire
    Tant que (FinDeRecherche = 0) faire
         $I_s \leftarrow \text{seuillage}(s, R)$ 
         $p_d \leftarrow \text{centre des crêtes de } I_s$ 
         $Np_d \leftarrow \text{card}(p_d)$ 
         $\epsilon \leftarrow |Np^* - Np_d|$ 
         $[s, \text{FinDeRecherche}, \text{Solution}] = \text{simplexe}(\epsilon)$ 
    Fait
     $s \leftarrow s + 0,1$ 
Fait
pd : Ensemble des points d'intérêt détecté

```

Algorithme 1: Harris simplexe

Dans l'algorithme présenté, la fonction simplexe donne la nouvelle valeur du seuil s en fonction de l'erreur ϵ . Elle renvoie également une information sur la fin de recherche par la variable $FinDeRecherche = 1$. Cette information peut prendre la valeur 1 dans deux cas. Soit le simplexe ne trouve pas de solution et s'arrête au bout d'un certain nombre d'itérations pré-défini. Soit il arrête sa recherche car il estime avoir trouvé une solution qui satisfait un critère d'encadrement de précision, à ce moment la fonction retourne l'information supplémentaire $Solution = 1$. Dans le cas où il ne trouve pas de solution et arrête sa recherche la valeur d'initialisation du seuil s est incrémentée d'un pas de 0,1 en espérant se rapprocher de la solution recherchée. Appliquons cet algorithme aux trois images de la mire réalisée précédemment. Pour chaque paire d'images qui représente une couche, on contraint un nombre de points désirés à 500 points. Pour chaque couple I_i et I'_i (gauche et droite), des couches de bas (indice 1) en haut (indice 3) de la mire, le Harris simplexe nous renvoie les valeurs de seuil suivantes : pour I_1 , $s = 0,0635$ et I'_1 , $s = 0,067266$; pour I_2 , $s = 0,039125$ et I'_2 , $s = 0,04025$ et enfin pour I_3 , $s = 0,034813$ et I'_3 , $s = 0,04025$. En représentant chaque point pour chaque image, on obtient bien 500 points par image (Figure 3.8). On remarque que les points détectés correspondent seulement aux parties nettes de l'objet, dans la profondeur de champ. Ce qui nous permet de sélectionner un ensemble de points de la mire en fonction des différentes couches de profondeur. Au total pour ce calibrage nous avons un nombre de 3000 points (500 points par couche et par image) qui se situent à différentes profondeurs dans la scène. Il reste à mettre en correspondance les points de chaque couche.

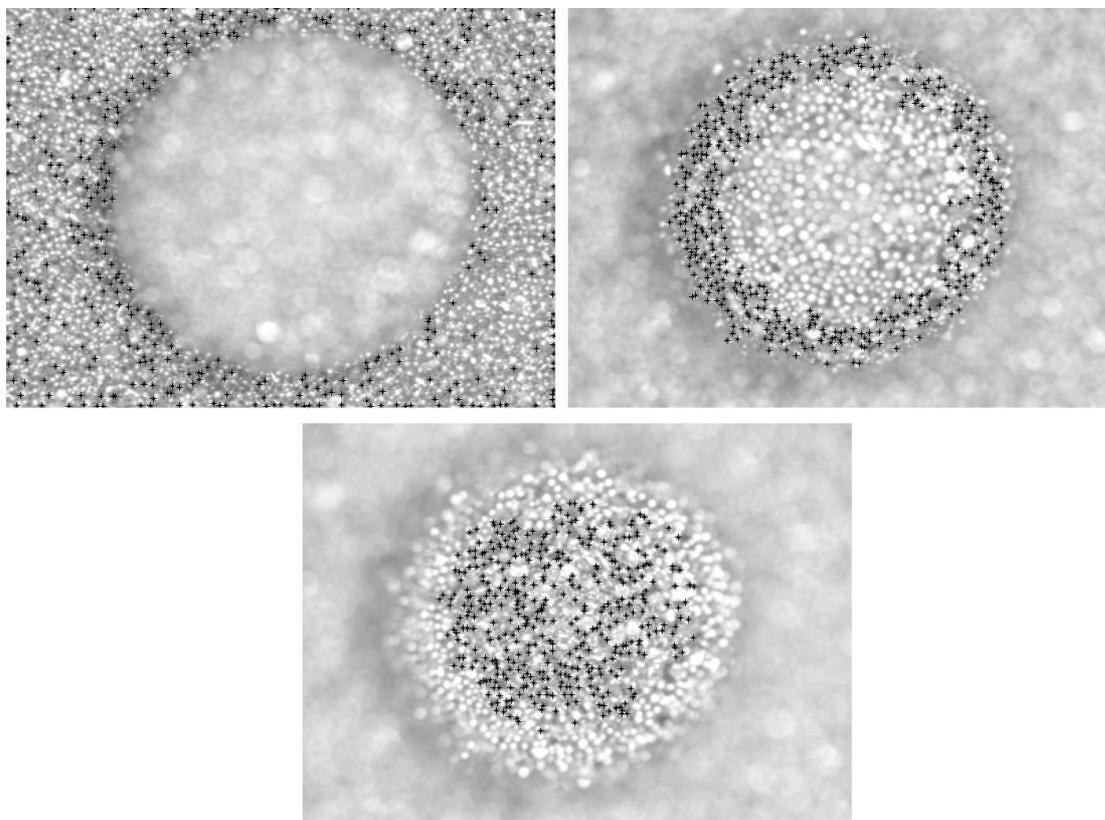


FIG. 3.8 – Exemples de résultat du détecteur Harris simplexe, pour chaque image, la consigne était de 500 points.

3.6 Mise en correspondance

L'étape de mise en correspondance permet de retrouver le correspondant de chaque point d'une image dans une autre. Par exemple, dans une image quelconque, si nous avons un point d'intérêt sur le coin d'un objet, la mise en correspondance consiste à déterminer son jumeau, c'est-à-dire le même coin de cet objet, parmi un ensemble de points d'une autre image. C'est une étape capitale dans le domaine de la vision, beaucoup de méthode de vision sont basées sur une première étape de mise en correspondance. Il existe plusieurs types de mise en correspondance : elle peut être dense, en prenant la totalité des points de l'image ; quasi-dense en prenant un peu moins que la totalité des pixels (un sur deux) ; ou encore d'intérêt en ne prenant que des points particuliers de l'image. La mise en correspondance permet d'extraire de l'information de déplacement et de profondeur de la scène, par le biais de l'analyse de la disparité entre les points appariés. La disparité correspond au déplacement dans l'image d'un point par rapport à son homologue dans l'autre image. Elle varie en fonction de la profondeur, plus la disparité entre deux points est faible et plus ces points sont éloignés de la source d'image. Ce phénomène est dû à la stéréoscopie, en regardant votre pouce bras tendu en alternant l'ouverture de vos

yeux droit et gauche, votre pouce se décale légèrement à gauche ou à droite. Par contre, la même expérience avec un objet très éloigné le fait apparaître avec un déplacement pratiquement nul. Ce phénomène est la base de la vue stéréoscopique. C'est pourquoi elle suscite toujours un vif intérêt dans la communauté de la vision artificielle.

Pour déterminer le correspondant de chaque point, nous allons utiliser l'information photométrique autour de ce point via une zone (fenêtre) et la comparer avec toutes les zones autour des points dans les autres images. Les deux zones les plus ressemblantes déterminent deux points correspondants l'un avec l'autre. Cette méthode est la technique de mise en correspondance par corrélation. Considérons deux images gauche et droite d'une scène, contenant un ensemble de points par image, la mise en correspondance des ensembles de points se décompose en trois temps pour chaque point de l'image de gauche :

- il faut définir une fenêtre de corrélation centrée sur le point, et faire la même chose autour de tous les points de l'image de droite,
- il faut calculer la fonction de corrélation entre la fenêtre de l'image de gauche avec toutes celles de l'image de droite,
- il faut chercher parmi toutes les valeurs de corrélation de droite celles qui correspondent au maximum de ressemblance avec la fenêtre de gauche.

Le pivot important des différentes étapes de la mise en correspondance est la fonction de corrélation, c'est elle qui permet de calculer le critère de ressemblance entre toutes les fenêtres définies, et par conséquent c'est elle qui détermine si la mise en correspondance est robuste ou pas.

3.6.1 Mise en corrélation par ZNSSD et ZNCC

Il existe une multitude de fonctions de corrélation, qui satisfont plus ou moins les critères nécessaires à une bonne correspondance. Ces critères sont les garants de la bonne détermination du correspondant parmi tous les points candidats. La fonction de corrélation demande une certaine robustesse en fonction du bruit dans l'image : photométrique (bruit du signal), géométrique (transformation projective) et matériel (sources d'image réglées différemment). Il existe deux familles de calcul du critère de corrélation fondées, soit sur la distance euclidienne entre les deux vecteurs basés sur les intensités des fenêtres, soit sur le produit scalaire de ces deux vecteurs. Le premier critère, par distance euclidienne est l'un des algorithmes le plus connu, le SSD (*Sum of Squared Differences*). Cette fonction de corrélation calcule le carré de la somme des différences entre les fenêtres. Le deuxième critère calcule le produit scalaire, c'est l'algorithme CC (*Cross-Correlation*). Tous deux peuvent gagner en robustesse en corrigeant la différence d'illumination des deux sources d'images. Ces deux nouveaux critères le ZSSD (*Zero-mean Sum of Squared Differences*) et le ZCC (*Zero-mean Cross-Correlation*), corrigent l'intensité de chaque fenêtre de telle sorte à obtenir une moyenne nulle, ce qui permet d'analyser la ressemblance de chaque fenêtre indépendamment du changement d'illumination entre les vues. L'ordre de grandeur de la valeur du critère de corrélation calculé dépend des intensités de chaque fenêtre, c'est-à-dire que pour une scène donnée l'ordre de grandeur du critère pour un point correspondant peut être différent de l'ordre de grandeur du critère des

autres points correspondants. Ceci ne permet pas, par un indice de confiance, de juger de la qualité de la mise en correspondance, ce qui pose un problème si on désire filtrer les points correspondants suspicieux. C'est pourquoi, on normalise le critère de corrélation, en utilisant ces deux critères le ZNSSD (*Zero-mean Normalized Sum of Squared Differences*) et le ZNCC (*Zero-mean Normalized Cross-Correlation*). Dans ce cas on peut comparer la valeur des critères pour chaque point mis en correspondance pour juger de la confiance de l'appariement. Ces deux derniers critères sont les plus populaires et les plus utilisés, même si le ZNCC semble être un peu plus robuste que le ZNSSD, comme le montre F. Devernay (1997). Ces deux critères conduisent à des résultats très similaires. Le critère du ZNSSD est défini de la sorte :

$$c_{x,y} = - \frac{\sum_{i,j} [(I(x+i, y+j) - \bar{I}) - (I'(x'+i, y'+j) - \bar{I}')]^2}{\sqrt{\sum_{i,j} [I(x+i, y+j) - \bar{I}]^2} \sqrt{\sum_{i,j} [I'(x'+i, y'+j) - \bar{I}']^2}} \quad (3.15)$$

où \bar{I} et \bar{I}' sont les moyennes des images I et I' . De même le critère ZNCC est défini par l'équation suivante :

$$c_{x,y} = - \frac{\sum_{i,j} [(I(x+i, y+j) - \bar{I}) (I'(x'+i, y'+j) - \bar{I}')]^2}{\sqrt{\sum_{i,j} [I(x+i, y+j) - \bar{I}]^2} \sqrt{\sum_{i,j} [I'(x'+i, y'+j) - \bar{I}']^2}} \quad (3.16)$$

Même si ces deux critères sont très populaires, il existe d'autres techniques de mise en correspondance. Elles ne sont pas présentées ici parce qu'elles s'adressent à des correspondances de type dense avec une détermination de manière globale avec *post-processing*, souvent utilisées dans un but de reconstruction de surfaces par carte des profondeurs par stéréovision (F. Devernay 1997). Dans notre cas, on désire juste chercher la position de points d'intérêt et donc ne nécessitant pas de méthode globale élaborée. Un recueil avec une comparaison de toutes les techniques de correspondance dense par stéréovision est présenté par (D. Scharstein et R. Szeliski 2002). Par contre même si le ZNSSD ou le ZNCC est suffisant, il n'en est pas moins important de vérifier leur robustesse avec des images de scènes de microscopie photonique. En effet les problèmes de réflexions spéculaires des objets, des parties floues de la scène et du peu de texture diminuent la probabilité de trouver le bon correspondant pour chaque point. C'est pour cela que nous proposons de rajouter encore une spécificité aux ZNSSD et ZNCC par une nouvelle amélioration.

3.6.2 Mise en corrélation par III-ZNSSD et III-ZNCC

Le critère de corrélation en lui même est satisfaisant et ne peut plus être amélioré de façon conséquente sur la mise en correspondance. Par contre, plusieurs études ont été menées sur la fenêtre de corrélation elle même. On trouve l'emploi de plusieurs fenêtres asymétriques par exemple dans (A. Fusiello, V. Roberto et E. Trucco 2000), ou encore

l'utilisation de fenêtres multiples adaptatives et récursives présentés par S. Chan, Y.-P. Wong et J. Daniel (2003). Il existe une comparaison plus complète des différentes fenêtres dans le papier de P. Aschwanden et W. Guggenbuhl (1992). Quoi qu'il en soit, nous proposons une nouvelle fenêtre de corrélation non récursive, unique et beaucoup plus simple et qui donne de meilleurs résultats. Elle est similaire aux méthodes de multi-résolutions mais présente l'avantage de ne pas utiliser une pyramide d'images (images avec différentes résolutions) et donc d'être beaucoup plus rapide. Pour la présenter nous allons partir de la définition d'une fenêtre de corrélation classique rectangulaire pour remonter à la définition de cette nouvelle fenêtre que nous appelons \mathbb{III} (« Cha »). Soit la fonction continue de la fenêtre de corrélation rectangulaire $f_w(x, y)$. Dans une image l'espace est discrétisé (numérisé) par les pixels. La fenêtre de corrélation dans l'image devient :

$$W_{\square} = f_w(x, y) \cdot \mathbb{III}_{\Delta x, \Delta y}(x, y) \quad (3.17)$$

où W_{\square} est la fonction de la fenêtre de corrélation rectangulaire dans l'image et $\mathbb{III}_{\Delta x, \Delta y}(x, y)$ un peigne de Dirac dans un espace (x, y) de dimension 2 :

$$\begin{aligned} \mathbb{III}_{\Delta x, \Delta y}(x, y) &= \mathbb{III}_{\Delta x}(x) \cdot \mathbb{III}_{\Delta y}(y) \\ &= \sum_{m=-\infty}^{+\infty} \delta(x - m\Delta x) \cdot \sum_{n=-\infty}^{+\infty} \delta(y - n\Delta y) \end{aligned} \quad (3.18)$$

c'est-à-dire la résultant de la multiplication de deux peignes de Dirac d'une dimension selon deux axes différents, $\mathbb{III}_{\Delta x}(x)$ peigne de Dirac sur l'axe x et $\mathbb{III}_{\Delta y}(y)$ peigne de Dirac sur l'axe y . Chaque peigne est échantillonné par un intervalle suivant x et y défini respectivement par Δx et Δy qui représente l'intervalle défini par les pixels (Figure 3.9). La nouvelle fenêtre $W_{\mathbb{III}}$ que nous proposons est une modification de la taille de la fenêtre et de la période d'échantillonnage de la fenêtre rectangulaire W_{\square} . La fenêtre se trouve sur-échantillonnée par un nouveau peigne de Dirac par un multiple entier k de la période d'échantillonnage Δx et Δy . Elle se trouve également sur-dimensionnée par le même multiple entier k . Notre nouvelle fenêtre est définie par cette équation :

$$W_{\mathbb{III}} = W_{k\square}^{\mathbb{I}} \cdot \mathbb{III}_{k\Delta x, k\Delta y}(x, y) \quad (3.19)$$

avec $\mathbb{III}_{k\Delta x, k\Delta y}(x, y)$ le nouveau peigne de Dirac 2D à la fréquence d'échantillonnage $k\Delta x$ et $k\Delta y$, et $W_{k\square}^{\mathbb{I}}$, la fenêtre discrète W_{\square} re-dimensionnée à l'échelle par k et forcée à une taille impaire \mathbb{I} . La dimension de cette nouvelle fenêtre, représentée mathématiquement par le cardinal selon x et selon y est définie par les équations suivante :

$$\begin{aligned} \text{card}_x(W_{k\square}^{\mathbb{I}}) &= k \times \text{card}_x(W_{\square}) - \text{mod}_3(k \times \text{card}_x(W_{\square})) \\ \text{card}_y(W_{k\square}^{\mathbb{I}}) &= k \times \text{card}_y(W_{\square}) - \text{mod}_3(k \times \text{card}_y(W_{\square})) \end{aligned} \quad (3.20)$$

avec $\text{card}_x(*)$ et $\text{card}_y(*)$ le cardinal selon x et y respectivement et $\text{mod}_n(*)$ le modulo n . Ces deux équations traduisent le fait que la dimension de la nouvelle fenêtre doit

être impaire quel que soit l'axe. La Figure 3.9 montre la topologie finale d'une fenêtre de corrélation III. Elle est éclatée en petites parties fenêtres et donc plus grande que la fenêtre originale. Elle est plus intéressante que la fenêtre classique par ce qu'elle a un champ de recherche moins local, en ayant toujours le même nombre de pixels, c'est-à-dire le nombre de calculs est le même qu'une fenêtre classique $\text{card}_{xy}(W_{\square}) = \text{card}_{xy}(W_{\text{III}})$.

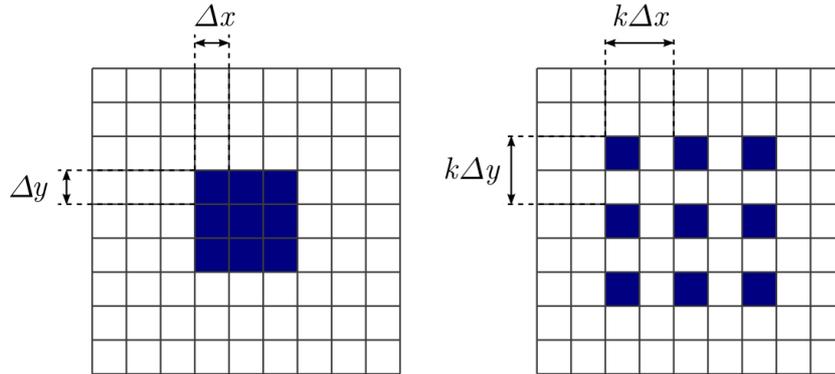


FIG. 3.9 – A gauche, la fenêtre de corrélation rectangulaire. A droite, la fenêtre de corrélation III avec un facteur $k = 2$.

Un autre avantage de ce style de fenêtre c'est qu'elle est adaptable à tous types de fonction de corrélation, aussi bien ZNSSD que ZNCC. Pour tester l'efficacité de cette nouvelle fenêtre de corrélation nous allons utiliser un benchmark. Dans un premier temps, nous allons utiliser une paire d'images stéréoscopiques célèbre pour être utilisée dans plusieurs expériences en vision. Nous allons utiliser la paire d'images de Tsukuba (Figure 3.10), elle permet à la communauté de tester des algorithmes sur une image commune, facilitant les comparaisons entre techniques.

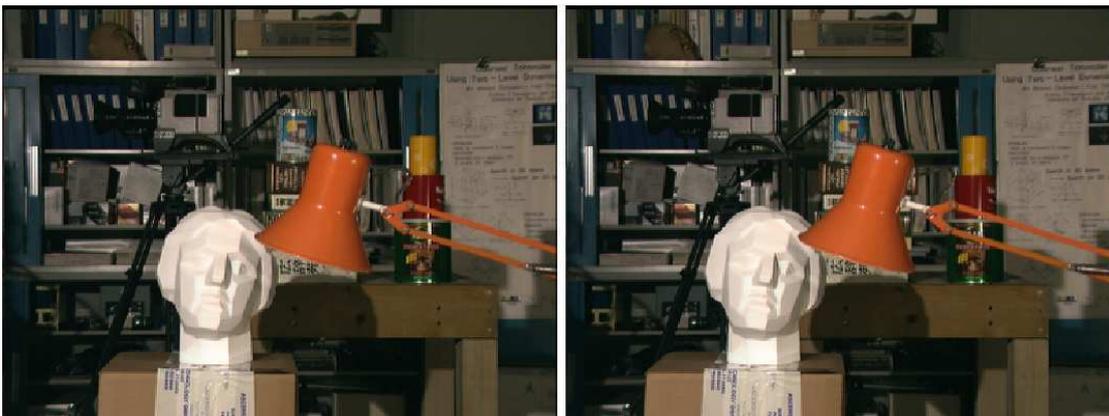


FIG. 3.10 – Images stéréoscopiques de Tsukuba (384×288 pixels), utilisées comme image de test.

Le benchmark développé ici, permet de comparer l'erreur de mise en correspondance en fonction du type de fenêtres et des différentes tailles. Nous allons utiliser un benchmark de type *forward-backward*. Le processus opératoire est le suivant :

- définir une grille de 2500 points qui recouvre toute la surface de l'image de gauche,
- chercher par corrélation la position de chaque point dans l'image de droite et ainsi retrouver la grille correspondante,
- utiliser cette nouvelle grille de l'image de droite pour de nouveau chercher par corrélation la position de chaque point dans l'image originale de gauche,
- calculer l'erreur moyenne des distances euclidienne entre les points de la grille d'origine et la grille estimée.

La recherche par corrélation s'effectue deux fois, une fois dans le sens de l'image de gauche vers l'image de droite, c'est le *forward*, puis dans le sens image de droite vers image de gauche, c'est le *backward*. La différence de distance entre les points des deux grilles représente l'erreur de corrélation pendant la recherche. Nous choisissons arbitrairement le critère de corrélation ZNSSD. Nous avons tracé à la Figure 3.11, le résultat des erreurs moyennes des 2500 points pour des tailles utiles de fenêtres allant de 3×3 pixels jusqu'à 21×21 pixels. Les courbes représentent le test avec une fenêtre classique et avec une fenêtre de type III.

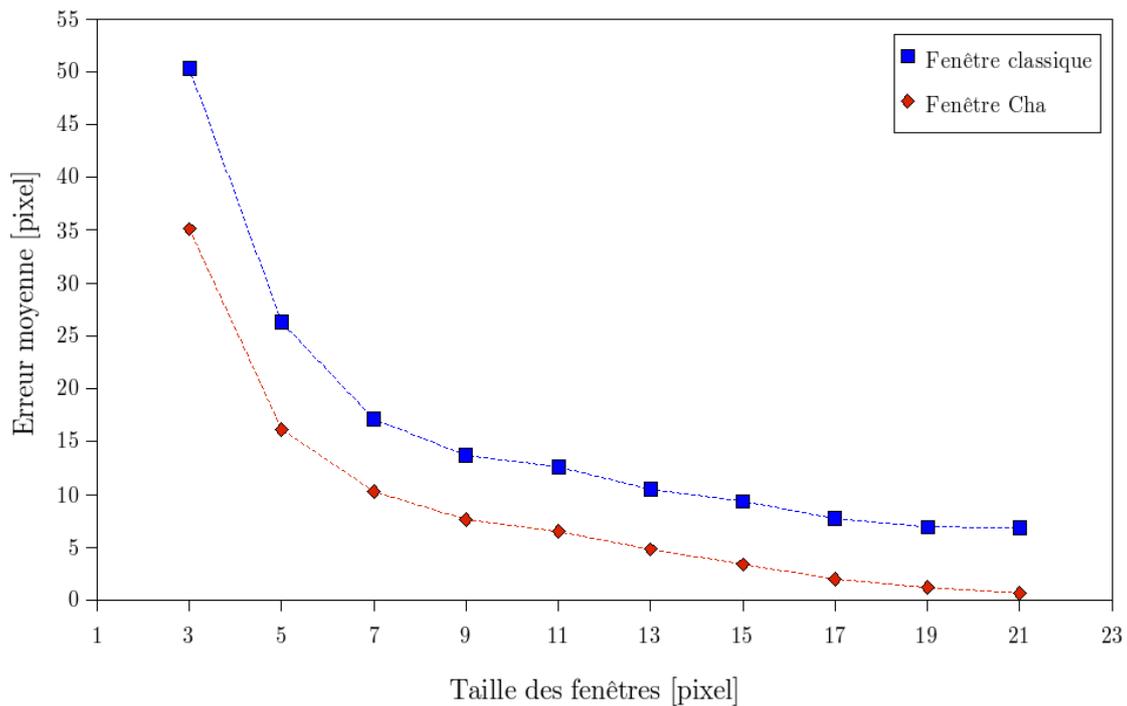


FIG. 3.11 – Erreur moyenne de la mise en correspondance sur les images de Tsukuba, avec un test qui comprend la recherche de 2500 points selon différentes tailles utiles de fenêtre pour une fenêtre rectangulaire et une fenêtre III (critère de corrélation ZNSSD).

La fenêtre III donne de meilleurs résultats quelle que soit sa taille par rapport à la fenêtre rectangulaire classique. Par exemple, pour une taille utile de 21×21 pixels l'erreur moyenne est de 6,86 pixels pour une fenêtre rectangulaire contre seulement 0,67 pixel pour une fenêtre de type III, ce qui correspond à une amélioration des résultats de 90,2%. Si on considère l'ensemble des tailles de fenêtres testées c'est-à-dire de 3×3 pixels à 21×21 pixels, en moyenne le III-ZNSSD apporte une amélioration de 56,7% sur l'erreur moyenne de la mise en correspondance. L'amélioration de la mise en correspondance par corrélation avec une III fenêtre est sans équivoque.

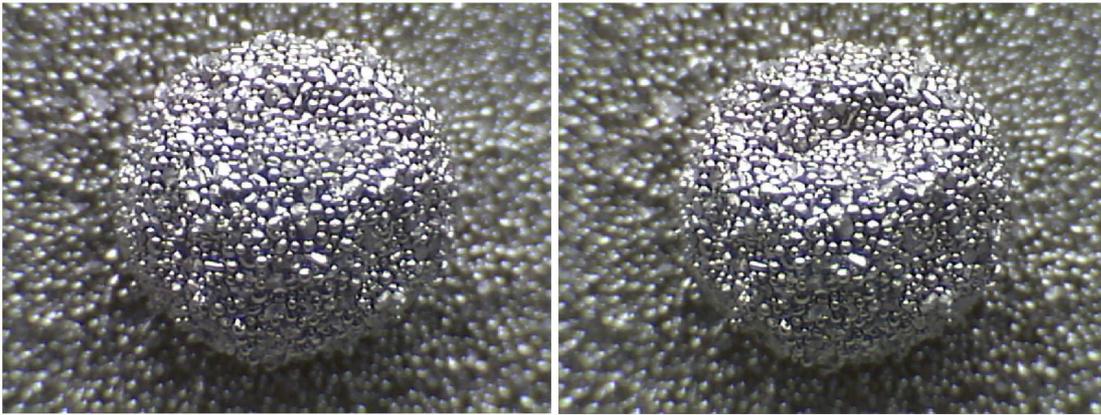


FIG. 3.12 – Images stéréoscopiques d'une mire de 1 mm de diamètre (768×576) pixels, utilisées comme images de test.

Il est intéressant de vérifier cette amélioration mais cette fois-ci sur une image provenant d'un microscope photonique avec une de nos mires. La grille des points est définie sur toute l'image or les images de calibrage en couche ont des zones floues. Ce flou n'est pas important pour le calibrage car les points détectés par le Harris simplexe sont sur la partie nette seulement, cela risque par contre de limiter la comparaison entre les fenêtres. Pour obtenir un couple d'images nettes nous diminuons le grossissement du stéréo microscope photonique et nous utilisons une mire plus grande (1 mm de diamètre) qui permet en une seule couche d'avoir l'ensemble de la mire (Figure 3.12). Le graphique résultat du benchmark sur les images de la mire se trouve à la Figure 3.13.

La première remarque, c'est que pour une taille de fenêtre supérieure à 15×15 pixels les résultats sont pratiquement équivalents pour les deux types de fenêtre (III demeurant légèrement supérieur), mais pour les autres tailles de fenêtre, les résultats du III sont meilleurs. Par exemple avec la taille de 9×9 pixels le benchmark donne une valeur d'erreur moyenne de 128,8 pixels pour une fenêtre rectangulaire classique, tandis que la fenêtre III ne donne que 67,6 pixels, ce qui représente une amélioration de 47,5%. Sur l'ensemble des tailles de fenêtre testé la méthode du III-ZNSSD permet une amélioration globale de 21,4%. L'amélioration de la fenêtre III ne joue pas que sur le plan de la qualité de la mise en correspondance mais aussi sur le plan de la rapidité. En effet à erreur moyenne équivalente, pour une taille de la fenêtre rectangulaire classique de $17 \times 17 = 289$ pixels, la taille utile de la fenêtre III n'est que de $9 \times 9 = 81$ pixels, soit

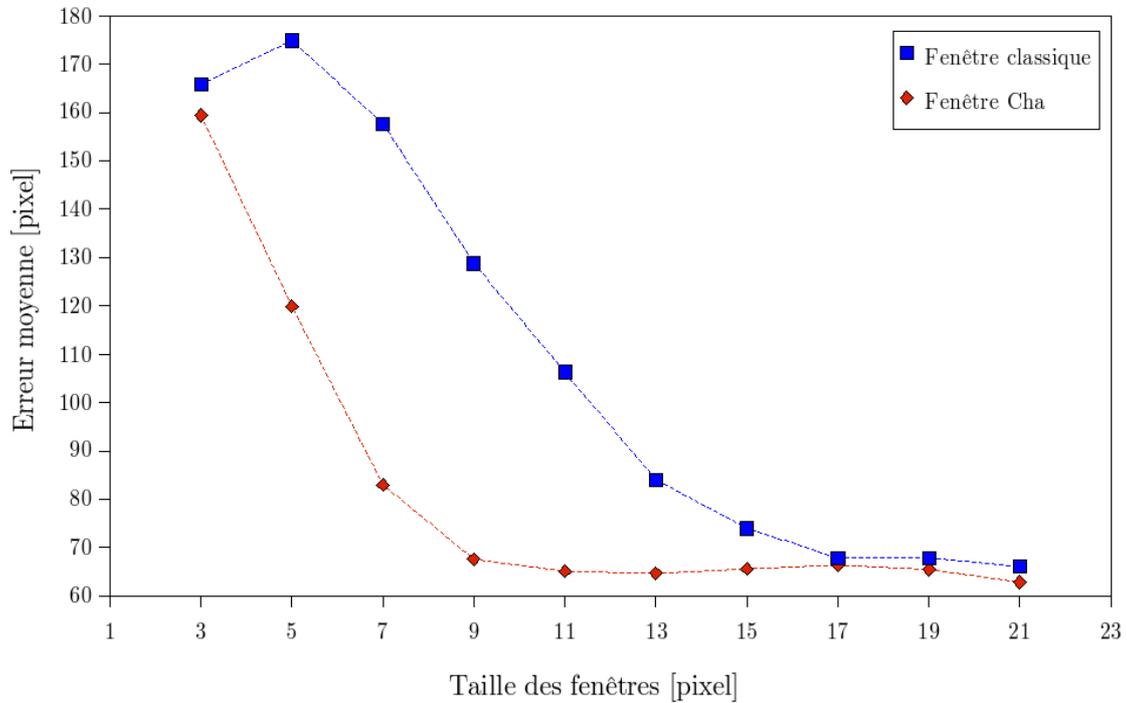


FIG. 3.13 – Erreur moyenne de la mise en correspondance sur les images d’une mire, avec un test qui comprend la recherche de 2500 points selon différentes tailles utiles de fenêtre, pour une fenêtre rectangulaire classique et une fenêtre III (critère de corrélation ZNSSD).

une rapidité potentielle de 3,56 fois plus grande. Si on compare les deux courbes entre un couple d’images d’une scène banale et un couple d’images d’une scène de microscopie photonique, on s’aperçoit que les ordres de grandeur entre les erreurs moyennes sont différents. Par exemple pour une fenêtre III de 9×9 pixels pour les images de Tsukuba l’ordre de grandeur est de $\simeq 9$ pixels d’erreur moyenne contre $\simeq 65$ pixels pour une image de notre mire. Il est donc important de renforcer la robustesse et le contrôle de la mise en correspondance pour garantir un calibrage de qualité. C’est dans cette optique que nous utilisons une recherche par corrélation couplée à une méthode de relaxation par contrainte de voisinage.

3.6.3 Mise en correspondance par corrélation avec une contrainte de voisinage

Conformément à la nouvelle fenêtre de corrélation nous allons faire la mise en correspondance pour chaque ensemble de points des différentes couches gauches et droites de la mire. Nous avons mis en évidence que malgré les améliorations de la mise en correspondance par corrélation, le risque d’erreur de faux appariement est encore élevé. En plus de travailler avec une contrainte de profondeur des points (en effectuant une mise

en correspondance entre couche correspondante), nous allons ajouter une contrainte supplémentaire : la contrainte du voisin le plus proche qui permet, d'une part de gagner du temps dans la mise en correspondance et d'autre part de garantir un bon appariement entre les points. Cette contrainte est basée sur le bon sens et l'hypothèse que comme les vues stéréoscopiques sont proches l'une de l'autre, cela signifie que nous avons une faible disparité, le point que l'on recherche dans l'image de droite doit forcément être un des voisins de cette même position dans l'image de gauche. Elle permet d'augmenter la rapidité de la mise en correspondance, car au lieu de calculer la corrélation pour tout l'ensemble des points, on la calcule seulement pour quelques uns. Elle permet également de diminuer les sources d'erreurs d'appariement car dans une scène il y a des ambiguïtés, le point ne peut pas être comparé à des points qui sont à une distance exagérée et donc potentiellement erroné. La comparaison des points voisins doit se faire sur plusieurs candidats, en effet un seul point voisin ne permettrait pas de garantir l'exatititude d'une bonne correspondance. Il existe différentes manières de procéder, Z. Zhang et al. (1995) utilise une méthode de relaxation par étiquetage et note les points candidats par minimisation d'énergie entre ces points et le point recherché. Pour gagner en rapidité et en simplicité nous allons utiliser une méthode semblable à la méthode de relaxation mais en utilisant les distances euclidiennes directes. La Figure 3.14 montre le procédé, pour chaque point p de l'image de gauche dont on recherche le correspondant dans l'image de droite on définit à la même position un cercle de rayon r dans l'image de droite. Le cercle va se relaxer, c'est-à-dire s'agrandir tant qu'un nombre pré-défini de points voisins candidats ne se trouvent pas à l'intérieur de ce cercle.

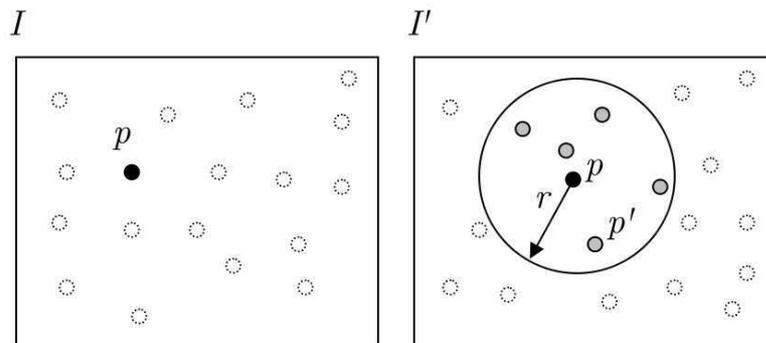


FIG. 3.14 – Procédé du choix des candidats pour la mise en correspondance par relaxation de la distance Euclidienne entre la proximité des points.

Une étape de mesure de la distance euclidienne est nécessaire entre le point recherché et tous les points potentiellement candidats à la mise en correspondance. Une fois qu'un point est apparié avec un de l'ensemble de l'image de droite, il est supprimé de cet ensemble. Cela permet encore une fois d'augmenter la rapidité de recherche et de limiter les erreurs en supprimant tout risque d'ambiguïté. Supposons l'image de gauche I et l'image de droite I' ainsi que leurs ensembles de points d'intérêt respectifs $\{p\}$ et $\{p'\}$. Notre algorithme de mise en correspondance est présenté dans l'algorithme 2.

I, I' : Image de gauche et image de droite
 $\{p\}, \{p'\}$: Ensemble de points d'intérêt de l'image I et I' respectivement
 nc_{min} : Nombre minimum de points candidat
 w : Taille de la fenêtre de corrélation

Pour chaque point p **faire**

| [*Création de la fenêtre de corrélation autour de p*]
 $W_{III} \leftarrow III_w(I, p)$
 | [*Calcul de la distance euclidienne entre chaque point*]
Pour chaque point p' **faire**

| $d \leftarrow dist_E(p; p')$
Fin Pour
 $rayon \leftarrow 0$
 | [*Relaxation du rayon du cercle des candidats*]
Tant que ($card(candidats) \leq nc_{min}$) **faire**

| [*Détermination des points candidats*]
 $candidats \leftarrow trouver(dist_E < rayon, p')$
 $rayon \leftarrow rayon + 1$
Fait
 | [*Corrélation des candidats*]
Pour chaque point candidat **faire**

| [*Création de la fenêtre de corrélation autour du candidat*]
 $W_{IIIc} \leftarrow III_w(I', candidats)$
 $c_{xy} = ZNSSD(W_{III}, W_{IIIc})$
Fin Pour
 | [*Choix du point apparié*]
 $p'_c \leftarrow max(c_{xy})$
 supprime p'_c dans $\{p'\}$
Fin Pour

$\{p\}, \{p'_c\}$: Ensemble des points d'intérêts apparié de l'image I et I'

Algorithme 2: Mise en correspondance par contrainte de voisinage

On applique cet algorithme aux couples d'images des trois couches de la mire de calibrage. Pour chaque couple de couche on obtient la correspondance c'est-à-dire qu'on détermine l'appariement de chaque point dans chaque image (Figure 3.15). On obtient, dans notre cas six ensembles de points, les ensembles $\{p_1, p_2, p_3\}$ des points des images de gauche et leurs points correspondants associés $\{p'_1, p'_2, p'_3\}$. Pour le calibrage nous devons regrouper tous ces points en deux grands ensembles p et p' , comme si ce n'était qu'une seule image de gauche et de droite. Le regroupement se fait par une simple fonction d'union entre chaque ensemble :

$$p = \bigcup_{i=1}^3 p_i \quad p' = \bigcup_{i=1}^3 p'_i \quad (3.21)$$

Une fois que les points d'intérêt sont appariés il ne reste plus qu'à calculer le calibrage faible à partir de ces points. Dans ce calibrage nous allons estimer les deux matrices importantes de la géométrie à deux vues, la matrice de colinéation et la matrice fondamentale.

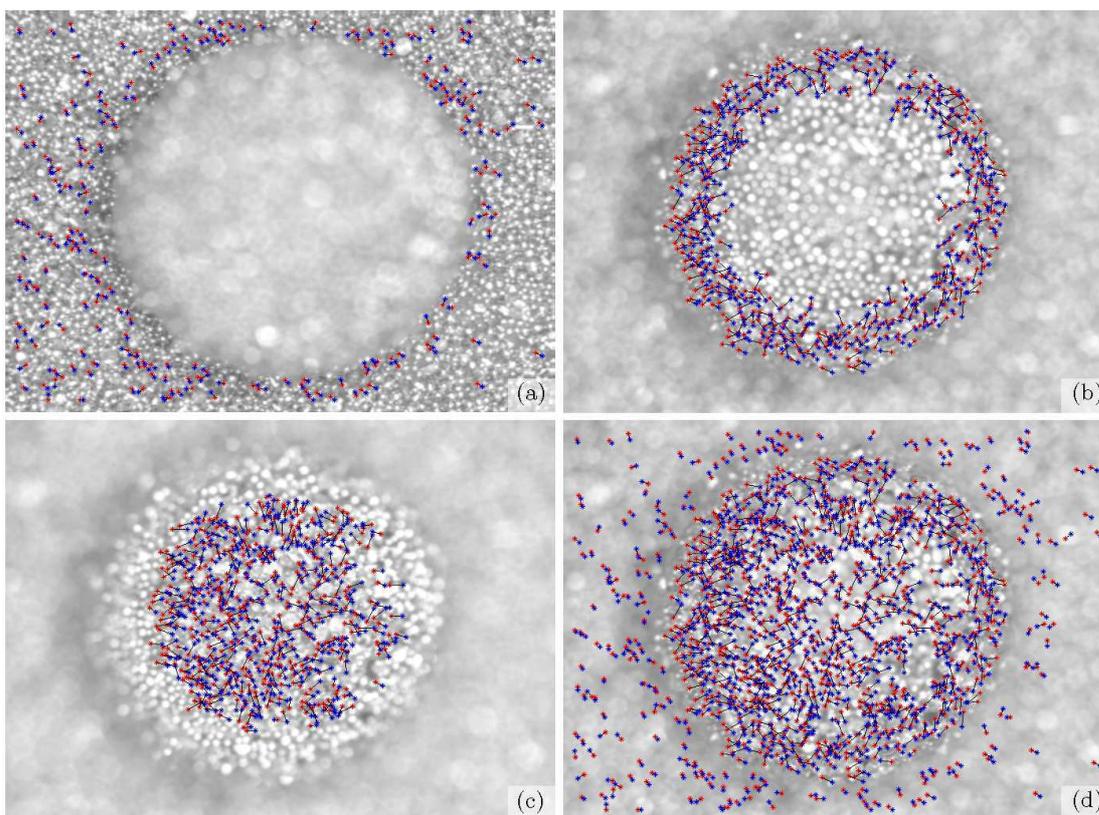


FIG. 3.15 – Mise en correspondance des images de la mire : (a), (b) et (c) respectivement les couples de couches 1, 2 et 3, (d) regroupement de tous les points correspondants des trois couches.

3.7 Calcul du Calibrage

Normalement le calibrage faible se résume à l'estimation de la matrice fondamentale. Dans beaucoup de techniques la matrice de colinéation est également utilisée, c'est pourquoi nous allons par la même occasion la calculer. Dans le cas de l'estimation de la matrice fondamentale nous avons besoin de points à différentes profondeurs dans la scène.

Nous allons utiliser l'ensemble regroupé de points appariés (p, p') . Par contre pour la matrice de colinéation, elle doit être estimée avec des points qui appartiennent à un même plan physique dans la scène. Dans ce cas nous allons utiliser seulement l'ensemble des points (p_1, p'_1) correspondants aux images de la première couche, qui représente le plan de travail. C'est pour cette raison, que dans notre cahier des charges de la mire il était stipulé la présence d'une zone avec différentes profondeurs pour la matrice fondamentale mais également une zone plan pour l'estimation de la matrice de colinéation.

3.7.1 Estimation de la matrice de colinéation

Pour estimer la matrice de colinéation nous allons utiliser la méthode DLT (*Direct Linear Transformation*) de J. Semple et G. Kneebone (1952). Elle permet de déterminer directement la transformation linéaire à partir des ensembles de points appariés. Étant donné l'ensemble des points de l'image de gauche p et son correspondant dans l'image de droite p' et la matrice de colinéation G qui est la transformation projective de la vue $\psi \rightarrow \psi'$, nous pouvons écrire la relation suivante :

$$p' \sim Gp \quad (3.22)$$

avec \sim l'égalité à un facteur λ près. Connaissant p et p' il nous faut calculer G , pour cela nous réécrivons l'égalité précédente :

$$[p']_{\times} Gp = 0 \quad (3.23)$$

une fois la notation étendue cela nous donne,

$$\begin{bmatrix} 0 & -1 & y' \\ 1 & 0 & -x' \\ -y' & x' & 0 \end{bmatrix} \begin{bmatrix} g_1^1 & g_2^1 & g_3^1 \\ g_1^2 & g_2^2 & g_3^2 \\ g_1^3 & g_2^3 & g_3^3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \quad (3.24)$$

où $p = (x, y, 1)^T$ et $p' = (x', y', 1)^T$ et g_i^j les éléments de la matrice G . Après développement nous obtenons trois égalités sous forme compacte :

$$\begin{aligned} y'(g_i^3 p^i) - (g_i^2 p^i) &= 0 \\ (g_i^1 p^i) - x'(g_i^3 p^i) &= 0 \\ x'(g_i^2 p^i) - y'(g_i^1 p^i) &= 0 \end{aligned} \quad (3.25)$$

avec (g_i^1, g_i^2, g_i^3) les vecteurs lignes des éléments de la matrice G et p^i le i ème élément du vecteur p . Nous utilisons une notation compacte appelée notation Einstein, elle traduit le produit scalaire sous forme de sommes cachées (pour plus de précisions se reporter au glossaire des notations avec l'expression $a_i b^i$). De ces trois égalités nous voulons extraire les éléments de g_i^j sous la forme d'un unique vecteur colonne. Pour cela nous devons transposer les vecteurs lignes (g_i^1, g_i^2, g_i^3) en vecteurs colonnes par $(g_i^{1T}, g_i^{2T}, g_i^{3T})$. En utilisant la propriété $g_i^n p^i = p^{iT} g_i^{nT}$, on réécrit les trois égalités :

$$\begin{aligned}
y'(p^{iT} g_i^{3T}) - (p^{iT} g_i^{2T}) &= 0 \\
(p^{iT} g_i^{1T}) - x'(p^{iT} g_i^{3T}) &= 0 \\
x'(p^{iT} g_i^{2T}) - y'(p^{iT} g_i^{1T}) &= 0
\end{aligned} \tag{3.26}$$

maintenant nous pouvons factoriser p et p' pour isoler les éléments de la matrice G sous forme d'un unique vecteur colonne :

$$\begin{bmatrix} 0_{1 \times 3} & -p^{iT} & y'p^{iT} \\ p^{iT} & 0_{1 \times 3} & -x'p^{iT} \\ -y'p^{iT} & x'p^{iT} & 0_{1 \times 3} \end{bmatrix} \begin{bmatrix} g_i^{1T} \\ g_i^{2T} \\ g_i^{3T} \end{bmatrix} = 0 \tag{3.27}$$

De ces trois équations il a été démontré que seules deux sont linéairement indépendantes. Il est classique d'omettre la dernière équation pour l'estimation de la matrice de colinéation (I. Sutherland 1963). Au final nous obtenons les deux équations suivantes :

$$\begin{bmatrix} 0_{1 \times 3} & -p^{iT} & y'p^{iT} \\ p^{iT} & 0_{1 \times 3} & -x'p^{iT} \end{bmatrix} \begin{bmatrix} g_i^{1T} \\ g_i^{2T} \\ g_i^{3T} \end{bmatrix} = 0 \tag{3.28}$$

Le développement et la factorisation finale a eu pour but de mettre la relation de transformation projective sous la forme,

$$Ag = 0 \tag{3.29}$$

c'est par le biais de cette forme que nous pourront extraire les éléments de la matrice de colinéation g_i^j qui est le noyau de A . Il faut un minimum de 4 points pour avoir la possibilité de calculer la matrice de colinéation, la méthode DLT est souvent aussi appelée méthode des 4 points. Dans notre cas nous voulons une estimation de qualité, par conséquent nous allons utiliser bien plus de 4 points. Pour utiliser n couples de points il faut construire la matrice A avec les éléments des couples des n points de la façon suivante :

$$\begin{bmatrix} 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1 y'_1 & y_1 y'_1 & y'_1 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ \vdots & \vdots \\ 0 & 0 & 0 & -x_n & -y_n & -1 & x_n y'_n & y_n y'_n & y'_n \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \end{bmatrix} \begin{bmatrix} g_1^1 \\ g_1^2 \\ g_1^3 \\ g_2^1 \\ g_2^2 \\ g_2^3 \\ g_3^1 \\ g_3^2 \\ g_3^3 \end{bmatrix} = 0 \tag{3.30}$$

Au final on obtient une matrice A de dimension $[2n \times 9]$ pour le vecteur g de la matrice G de dimension $[9 \times 1]$. L'extraction des éléments du vecteur g est obtenue par une décomposition en valeurs singulières de la matrice A :

$$A = U\Sigma V^T \quad (3.31)$$

Avec U de taille $[2n \times 2n]$, Σ de taille $[2n \times n]$ et V de taille $[n \times n]$. Les valeurs des éléments de la matrice de colinéation g_i^j se retrouvent dans le dernier vecteur colonne de V (colonne où la valeur singulière est la plus faible). Une fois les éléments obtenus on les remet sous une forme matricielle G . Cette méthode est la plus simple possible pour estimer la matrice de colinéation, par contre les résultats obtenus dépendent des emplacements des points dans l'image, c'est-à-dire que le résultat n'est pas invariant aux transformations similaires dans l'image. Pour compenser ce problème, une étape de normalisation des points est nécessaire. L'algorithme DLT normalisé est présenté dans le livre de R. Hartley et A. Zisserman (2000). Il consiste en une normalisation par une échelle isotrope. Dans un premier temps, il faut traduire tous les points de l'image vers une origine définie comme étant le centre de gravité de l'ensemble des points. Ensuite les points subissent un changement d'échelle pour que la distance moyenne des points et de l'origine soit égale à $\sqrt{2}$. Les transformations sont appliquées à chaque image indépendamment. Après avoir estimé la matrice de colinéation il ne faut pas oublier de la dé-normaliser par la transformation utilisée pour les points p' de la deuxième image. Pour que la méthode d'estimation soit claire non pas seulement d'un point de vue théorique mais aussi d'un point de vue pratique nous proposons son code dans le langage Matlab/Octave¹ :

```

0000 % Définit la matrice A et le vecteur nulle [1x3]
0001 A = zeros(2*size(p1,2),9);
0002 0 = [0 0 0];
0003 % Constitue la matrice A avec tous les points p1=p et p2=p'
0004 for n = 1:size(p1,2)
0005     x2 = p2(1,n);
0006     y2 = p2(2,n);
0007     A(2*n-1,:) = [ 0          -p1(:,n)'    y2*p1 ];
0008     A(2*n  ,:) = [ p1(:,n)'    0          -x2*p1 ];
0009 end
0010 % Décomposition en valeur singulière de A
0011 [U,S,V] = svd(A);
0012 % Retaille les éléments de g sous forme de matrice G [3x3]
0013 G = reshape(V(:,9),3,3)';
0014 % Ecriture de G sous forme homogène
0015 G = G./G(3,3);

```

Dans le cas pratique, on homogénéise toujours la matrice de colinéation car elle est définie à un facteur d'échelle λ près. Par conséquent, rendre homogène la matrice de colinéation G s'effectue en divisant par un facteur d'échelle telle que la valeur g_3^3 soit

¹GNU Octave : logiciel de calcul numérique comparable à MATLAB et à Scilab. Ce logiciel est distribué sous licence GPL (Licence publique générale GNU). <http://www.octave.org>.

égale à 1. Le facteur d'échelle est souvent source d'erreur dans la mise en application des transformations projectives. Par exemple, pour transformer un ensemble de points dans une autre vue en utilisant une matrice de colinéation, il ne faut pas oublier d'homogénéiser aussi les nouveaux points calculés, c'est-à-dire pour un ensemble de points $p = (x, y, 1)^T$, après transformation par une colinéation G , nous obtenons l'ensemble de points $p' = (\lambda x', \lambda y', \lambda)$. Si ces points ne sont pas remis sous forme homogène, leurs positions dans l'image cible sont erronées.

3.7.2 Estimation de la matrice fondamentale

La matrice fondamentale va s'estimer de façon très similaire à la matrice de colinéation, c'est-à-dire en utilisant une décomposition en valeurs singulières. La matrice fondamentale est calculée par l'algorithme des 8 points normalisés introduit par H. Longuet-Higgins (1981) et finalisé par R. Hartley (1997). Ce n'est pas la seule méthode d'estimation de la matrice fondamentale, il existe différents algorithmes dans la littérature, quelques uns sont proposés dans l'ouvrage de R. Hartley et A. Zisserman (2000). La méthode des 8 points que nous avons choisi est la plus populaire, car elle est simple et donne de bons résultats. Pour appliquer cette méthode comme son nom l'indique il faut un minimum de 8 couples de points appariés (p, p') . La matrice fondamentale \hat{F} est définie par l'équation :

$$p'^T \hat{F} p = 0 \quad (3.32)$$

En écrivant complètement avec les éléments nous obtenons :

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} \hat{f}_1^1 & \hat{f}_2^1 & \hat{f}_3^1 \\ \hat{f}_1^2 & \hat{f}_2^2 & \hat{f}_3^2 \\ \hat{f}_1^3 & \hat{f}_2^3 & \hat{f}_3^3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \quad (3.33)$$

\hat{f}_i^j sont les éléments de la matrice fondamentale \hat{F} . Après développement nous obtenons une seule égalité de cette forme,

$$x' \hat{f}_i^1 p^i + y' \hat{f}_i^2 p^i + \hat{f}_i^3 p^i = 0 \quad (3.34)$$

où $(\hat{f}_i^1, \hat{f}_i^2, \hat{f}_i^3)$ sont les vecteurs lignes de la matrice \hat{F} . Pour les mêmes raisons que la matrice de colinéation nous souhaitons isoler les éléments de la matrice \hat{F} sont forme d'un vecteur colonne unique dans le but d'appliquer la décomposition en valeurs singulières. En utilisant la propriété $\hat{f}_i^n p^i = p^{iT} \hat{f}_i^{nT}$ nous réécrivons l'équation précédente dans sa nouvelle forme :

$$x' p^{iT} \hat{f}_i^{1T} + y' p^{iT} \hat{f}_i^{2T} + p^{iT} \hat{f}_i^{3T} = 0 \quad (3.35)$$

Désormais on peut factoriser p et p' pour isoler les éléments de la matrice fondamentale, nous obtenons :

$$\begin{bmatrix} x'p^{iT} & y'p^{iT} & p^{iT} \end{bmatrix} \begin{bmatrix} \hat{f}_i^{1T} \\ \hat{f}_i^{2T} \\ \hat{f}_i^{3T} \end{bmatrix} = 0 \quad (3.36)$$

Le but de ces calculs, de développement suivi d'une factorisation, nous permet de retomber sur la forme nécessaire à la décomposition en valeurs singulières, du type,

$$A\hat{f} = 0 \quad (3.37)$$

avec \hat{f} le vecteur colonne des éléments de la matrice fondamentale \hat{f}_i^j . Comme dans la méthode DLT pour la matrice de colinéation nous devons construire la matrice particulière A avec les n couples de points (p, p') mis en correspondance. La constitution de la matrice finale est la suivante :

$$\begin{bmatrix} x_1x'_1 & x'_1y_1 & x'_1 & x_1y'_1 & y_1y'_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots \\ x_nx'_n & x'_ny_n & x'_n & x_ny'_n & y_ny'_n & y'_n & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} \hat{f}_1^1 \\ \hat{f}_2^1 \\ \hat{f}_3^1 \\ \hat{f}_1^2 \\ \hat{f}_2^2 \\ \hat{f}_3^2 \\ \hat{f}_1^3 \\ \hat{f}_2^3 \\ \hat{f}_3^3 \end{bmatrix} = 0 \quad (3.38)$$

La matrice A est de dimension $[n \times 9]$ et le vecteur \hat{f} de dimension $[9 \times 1]$. L'étape suivante consiste à récupérer les éléments du vecteur \hat{f} par une décomposition en valeurs singulières. Par contre, la matrice F a la particularité d'être une matrice singulière de rang 2, or ce que nous obtenons c'est une matrice \hat{F} qui n'a pas cette propriété. Nous devons donc forcer \hat{F} à devenir de rang 2. La première étape est de récupérer la matrice \hat{F} par décomposition en valeurs singulières :

$$A = U_A \Sigma_A V_A^T \quad (3.39)$$

Les valeurs des éléments du vecteur colonne \hat{f} se trouvent dans la dernière colonne de V_A^T qui correspond au vecteur de la plus petite valeur singulière de Σ_A . Les éléments de \hat{f} sont placés sous forme matricielle dans \hat{F} . Maintenant nous devons contraindre la matrice \hat{F} pour obtenir la matrice F de rang 2. F doit minimiser la norme de Frobenius $\|\hat{F} - F\|$ suivant la condition $\det(F) = 0$. Pour cela nous allons de nouveau décomposer en valeurs singulières la matrice \hat{F} :

$$\hat{F} = U_{\hat{F}} \Sigma_{\hat{F}} V_{\hat{F}}^T \quad (3.40)$$

Dans la matrice $\Sigma_{\hat{F}}$ on trouve les trois valeurs singulières sur la diagonale,

$$\Sigma_{\hat{F}} = \text{diag}(\sigma_1, \sigma_2, \sigma_3) \quad (3.41)$$

avec $(\sigma_1, \sigma_2, \sigma_3)$ les valeurs singulières de \hat{F} . Pour contraindre $\Sigma_{\hat{F}}$ au rang 2 et obtenir la nouvelle matrice nous allons contraindre la dernière valeur singulière à $\sigma_3 = 0$ puis recomposer la nouvelle matrice F en gardant les matrices $U_{\hat{F}}$ et $V_{\hat{F}}^T$. Nous obtenons la nouvelle matrice fondamentale :

$$F = U_{\hat{F}}[\text{diag}(\sigma_1, \sigma_2, \sigma_3)]V_{\hat{F}}^T \quad (3.42)$$

Comme pour la matrice de colinéation, la même étape de normalisation des points permet d'avoir de meilleurs résultats sur F . Sous Octave/Matlab, l'algorithme le plus simple peut s'écrire de la façon suivante :

```
0001 % Constitution de la matrice A avec p1=p et p2=p'
0002 A = [p2(1,:)'.*p1(1,)', p2(1,:)'.*p1(2,)', p2(1,:)'. ...
0003         p2(2,:)'.*p1(1,)', p2(2,:)'.*p1(2,)', p2(2,:)'. ...
0004         p1(1,)', p1(2,)', ones(size(p1,2),1)];
0005 % Première décomposition en valeur singulière
0006 [U,D,V] = svd(A);
0007 % Extraction de la matrice fondamentale
0008 F = reshape(V(:,9),3,3)';
0009 % Deuxième décomposition en valeur singulière
0010 [U,D,V] = svd(F);
0011 % Force la matrice fondamentale au rang 2
0012 F = U*diag([D(1,1) D(2,2) 0])*V';
```

La matrice de colinéation et la matrice fondamentale doivent être déterminées avec précision. Malgré les méthodes robustes employées, il y a toujours du bruit et donc des erreurs de mise en correspondance. Ces erreurs se répercuteront directement sur l'estimation des matrices G et F . Pour obtenir une estimation robuste par l'algorithme 4 et 8 points nous allons ajouter une étape supplémentaire. Il existe différentes méthodes d'optimisation du procédé d'estimation de la matrice de colinéation (A. Agarwal, C. Jawahar et P. Narayanan 2005), il en existe davantage pour l'estimation de la matrice fondamentale (R. Hartley et A. Zisserman 2000) et de façon plus générale appliquée à la vision robotique (E. Malis et E. Marchand 2005). Une méthode rapide en terme de calcul et robuste dans l'estimation, est la méthode RANSAC qui connaît depuis sa création un franc succès.

3.8 Estimation robuste par RANSAC

L'algorithme RANSAC (*RANdom SAmple Consensus*), a été introduit par M. Fischler et R. Bolles (1981). Il s'agit d'une méthode d'estimation robuste par vote probabiliste. Il permet d'obtenir un modèle avec précision à partir d'un ensemble de points appariés comportant un pourcentage de faux appariements. A partir de sous-ensembles de points un modèle estimé est constitué puis comparé au reste des sous-ensembles et enfin les points ne correspondant pas à ce modèle sont écartés. Petit à petit le modèle s'affine par filtrage des points. Cette méthode peut être adaptée à différents types

de modèles, comme par exemple la matrice de colinéation ou la matrice fondamentale. L'algorithme RANSAC comporte quatre grandes étapes itératives :

- choisir aléatoirement un sous-ensemble S_{e_n} de couples de points dans l'ensemble S (initialisé par l'ensemble des couples de points (p, p')) comprenant des erreurs d'appariement,
- estimer le modèle \hat{M}_n à partir du sous-ensemble S_{e_n} ,
- calculer les points estimés \hat{p} à partir du modèle \hat{M}_n et des points p ,
- retirer les couples de points de l'ensemble S pour lesquels l'erreur entre les points p' et les points estimés \hat{p} est supérieure à un seuil s_ϵ pré-déterminé.

L'algorithme ré-itére ces quatre étapes jusqu'à ce que tous les points estimés aient une erreur inférieure au seuil établi. Par exemple dans le cas de l'estimation de la matrice de colinéation, un sous-ensemble de 4 couples de points est choisi dans l'ensemble S . Ensuite en utilisant la méthode DLT on estime le modèle \hat{G} de ce sous-ensemble. On calcule les points estimés par ce modèle : $\hat{p}' = \hat{G}p$. L'erreur est mesurée en utilisant la distance euclidienne entre les points estimés \hat{p}' et les points contenant des erreurs p' . Si la distance est supérieure à un seuil s_ϵ les couples de points sont supprimés de l'ensemble S . Et on ré-itére ce processus jusqu'à obtenir des points estimés qui rentrent dans la tolérance du seuil s_ϵ . Au fur et à mesure des itérations, l'ensemble S est filtré des faux appariements. La dernière étape de RANSAC utilise l'ensemble S filtré pour calculer la matrice de colinéation. On utilise le même procédé avec l'estimation de la matrice fondamentale, sauf qu'on utilise 8 points pour calculer F et qu'on mesure cette fois-ci la distance euclidienne entre la ligne épipolaire estimée \hat{l} et la coordonnée de p' .

L'algorithme RANSAC donne des résultats très robustes, par exemple pour 20% de faux appariements, il suffit de 5 tirages aléatoires seulement pour obtenir une probabilité de trouver le bon modèle de 95%. RANSAC est capable d'estimer un modèle avec 50% d'erreur de mise en correspondance dans un ensemble de couples de points. Pour déterminer avec précision le nombre de tirages aléatoires nécessaire pour une probabilité P_A d'avoir le bon modèle, il faut appliquer la relation suivante :

$$t_i = \frac{\log(1 - P_A)}{\log(1 - (1 - \epsilon\%)^N)} \quad (3.43)$$

avec t_i le nombre de tirages aléatoires, $\epsilon\%$ le pourcentage d'erreur de mise en correspondance et N le nombre de points nécessaire à la détermination du modèle. Dans l'ouvrage de R. Hartley et A. Zisserman (2000), on trouve un petit abaque pour déterminer le nombre de tirages aléatoires minimum pour une probabilité de 0,99 pour différentes valeurs de N et $\epsilon\%$.

Nous appliquons la méthode d'estimation robuste RANSAC à nos deux ensembles de points appariés provenant de la mire. La Figure 3.16 représente le résultat après le filtrage par RANSAC. Dans le cas de la matrice de colinéation tous les points ne sont pas utilisés, il est donc normal dans l'image résultat de ne pas trouver de point sur la goutte d'eau texturée elle même. Par contre quel que soit le modèle utilisé, matrice de colinéation ou matrice fondamentale, RANSAC supprime les erreurs d'appariement. On obtient deux ensembles de points pour G et F beaucoup moins importants qu'au départ. C'est pourquoi il est important de détecter un nombre important de points dès le début,

car le filtrage réduit la taille des ensembles appariés. La dernière étape du calibrage micrométrique est consacrée à la vérification et à la quantification de la qualité des deux matrices estimées par RANSAC.

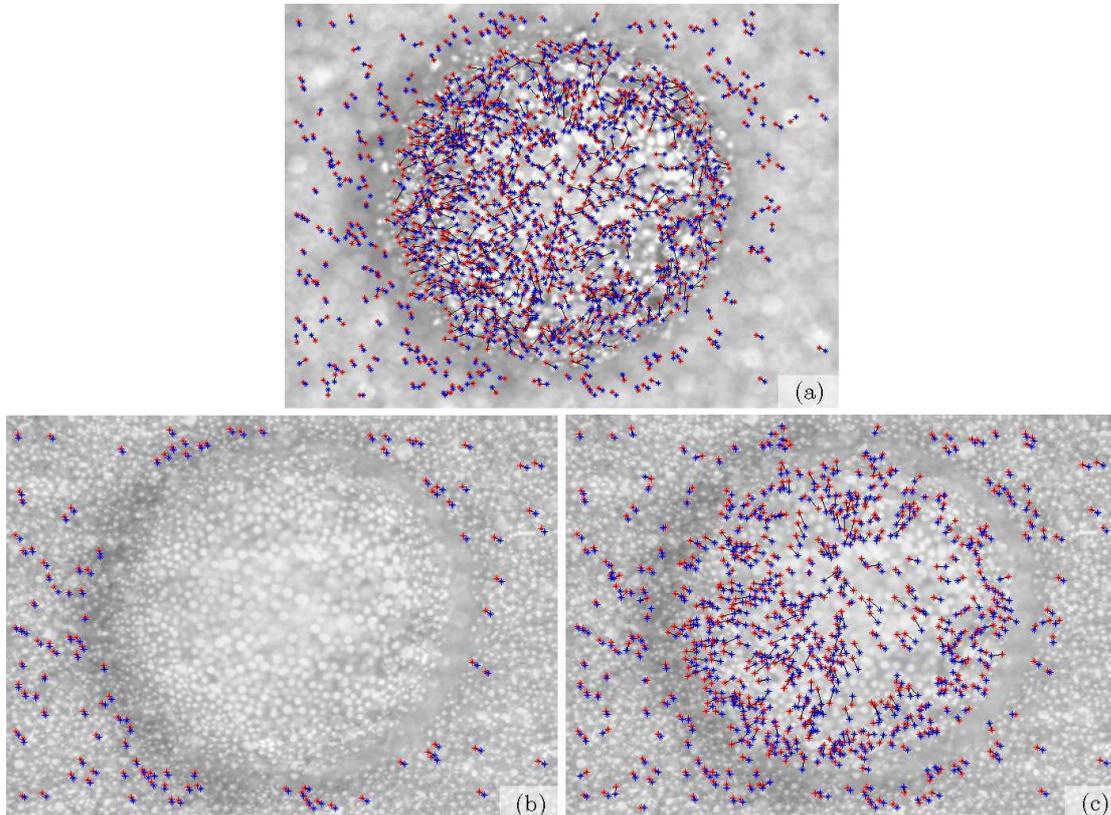


FIG. 3.16 – Estimation robuste par RANSAC : (a) représentation de tous les couples de points appariés, (b) les couples de points utilisés pour estimer G (après RANSAC), (c) les couples de points utilisés pour estimer F (après RANSAC).

3.9 Quantification de la qualité du calibrage

Nous allons quantifier la qualité du calibrage pour la matrice fondamentale et la matrice de colinéation. Le but est de déterminer une valeur qui reflète l'exactitude du modèle obtenu. Le nombre de tirages aléatoires de RANSAC dépend du pourcentage d'erreur de mise en correspondance. Or il est très difficile de quantifier après la mise en correspondance la quantité d'erreurs d'appariement. Si le nombre de tirages aléatoires n'est pas suffisant le résultat du modèle comporte une erreur importante. Effectivement quand on exécute RANSAC, pour son premier tirage des couples de points, il peut tomber sur un sous-ensemble de points sans erreur puis lors d'une autre exécution son premier tirage peut sélectionner un sous-ensemble de points avec beaucoup d'erreur. En fonction

des premiers tirages le résultat du modèle sera différent. La solution consiste à faire un nombre de tirages aléatoires maximum quelle que soit l'image analysée, par exemple pour la matrice de colinéation avec 50% d'erreurs d'appariement il faut 1177 tirages aléatoires pour avoir une probabilité de 0,99 de trouver le bon modèle. Un autre facteur de variation du résultat est le réglage du seuil s_ϵ , qui est choisi de façon empirique, un seuil mal choisi diminue le nombre de tirages aléatoires, car RANSAC s'arrête en estimant avoir trouvé le bon modèle. Pour toutes ces raisons il est nécessaire de vérifier *a posteriori* si le modèle estimé correspond bien au couple de points utilisés.

Pour déterminer la qualité de la matrice F nous allons utiliser la propriété de dualité point/ligne. Pour chaque point p de l'image de gauche I nous traçons les lignes épipolaires correspondantes l' dans l'image de droite I' . Ensuite nous calculons la distance euclidienne entre la ligne l' et le point p' , comme le fait RANSAC pour estimer l'erreur du modèle des sous-ensembles. Cette matrice sera estimée à partir de 573 points. La moyenne des distances euclidienne entre tous ces points définit l'erreur du modèle estimé par RANSAC. Pour mesurer les déviations du modèle F par rapport à différentes estimations par RANSAC, nous allons calculer cette erreur pour 20000 estimations de F . Nous représentons sa distribution à la Figure 3.17.

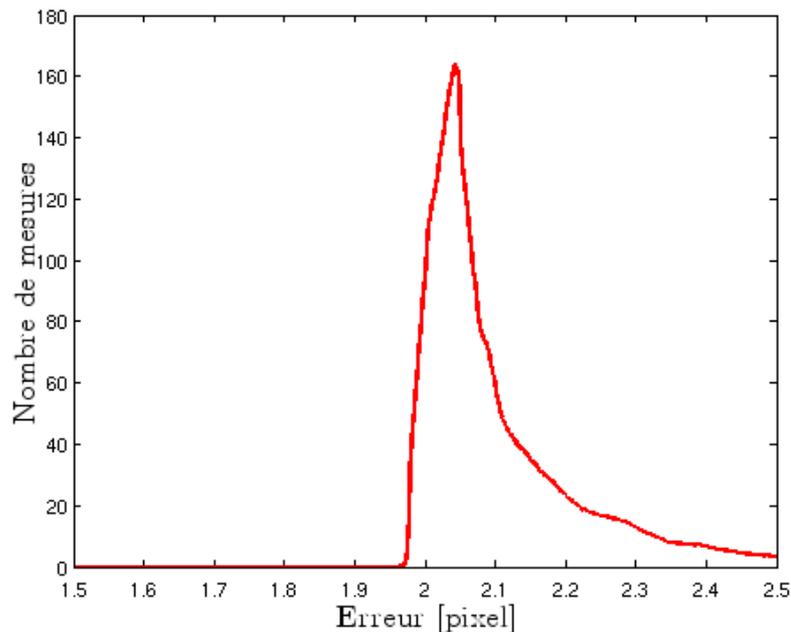


FIG. 3.17 – Distribution des erreurs pour 20000 estimations différentes de F par RANSAC (erreur calculée avec 513 points).

De cette distribution à peu près symétrique on en déduit quelques valeurs significatives. Premièrement la valeur moyenne de l'erreur sur l'ensemble des estimations est de 2,08 pixels. Cette valeur est importante si on la compare à la taille de l'image utilisée qui est de 640×480 pixels. La deuxième valeur que l'on peut en déduire est l'écart type, de 0,34 pixel. Cette valeur est l'image directe de la précision de l'estimation du modèle.

Sa faible valeur montre que l'ensemble de nos méthodes pour estimer la matrice fondamentale sont de bonnes qualités. Nous utilisons le même principe de mesure pour estimer l'erreur du modèle de la matrice de colinéation. Cette fois-ci l'erreur est la distance entre la position d'un point estimé par la matrice de colinéation et sa position réelle. De même on représente à la Figure 3.18 la distribution de l'erreur pour 20000 estimations de la matrice de colinéation à partir d'un échantillon de 513 points.

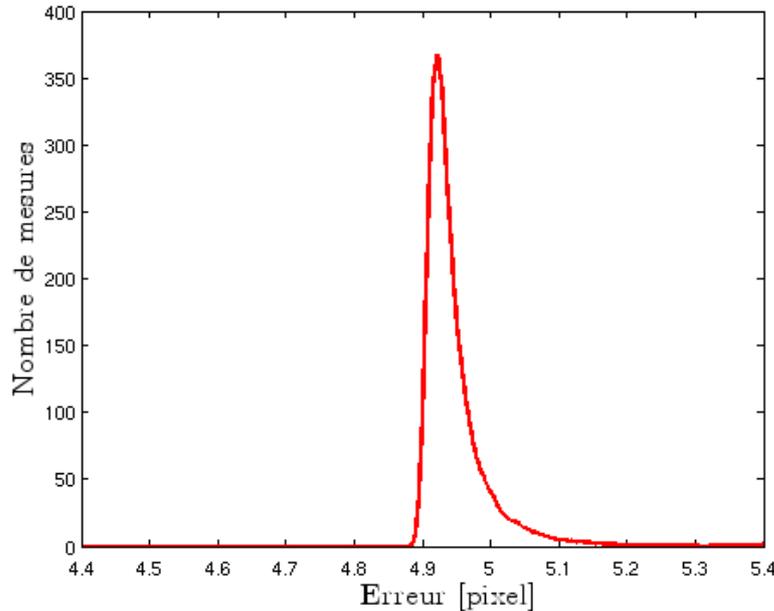


FIG. 3.18 – Distribution des erreurs pour 20000 estimations différentes de A par RANSAC (erreur calculée avec 513 points).

Cette distribution est également quasi symétrique. La valeur de l'erreur moyenne est de 4,93 pixels et l'écart type de 0,055 pixel. La précision du modèle estimé est bien meilleur que celle pour la matrice fondamentale. Cette différence peut s'expliquer par le fait que la matrice fondamentale est estimée à partir de points qui appartiennent à différentes couches de profondeur dans la scène. Pour obtenir ces différentes couches on joue sur la focal du microscope photonique qui par ses propriétés modifie légèrement le facteur d'échelle des images. Par conséquent la matrice fondamentale est estimée à partir d'un ensemble de points qui comporte un bruit plus important que la matrice de colinéation, car celle-ci est estimée à partir d'un seul couple d'images.

3.10 Conclusion

Dans ce chapitre nous avons présenté différents algorithmes qui seront appelés tout au long du reste du mémoire dans les applications des techniques de synthèse de vues. Ces algorithmes ont été présentés autour du calibrage faible d'un système stéréoscopique binoculaire en contexte de microscopie photonique. Le calibrage stéréoscopique faible,

qui correspond à l'estimation de la matrice fondamentale, n'est pas choisi par hasard, il est directement impliqué dans la réalisation du transfert trifocal utilisé dans les futurs chapitres. Il en est de même de la méthode présentée pour la matrice de colinéation. Nous avons mis l'accent sur notre apport dans la méthode du calibrage, à savoir : la mise en place d'une mire originale, le détecteur Harris simplexe, la nouvelle fenêtre de corrélation III. Pour mettre en évidence le côté pratique du calibrage, nous avons proposé tout au long des sections l'écriture de méthodes sous forme algorithmique et pour certaines sous forme de langage Matlab/Octave. Pour les fonctions classiques, Harris, la méthode des 8 points normalisés et la méthode des 4 points normalisés, il n'est pas nécessaire de les implémenter : elles sont présentes sur internet, par exemple dans la bibliothèque développée par P. Kovesi. Cette *toolbox* est écrite en langage Matlab et compatible avec le langage Octave. Pour les nouvelles fonctions, nous les avons implémentées sous Matlab et créé notre *toolbox* maison MCV (*Micromanipulation and Computer Vision*).

Le chapitre suivant présente la première méthode de synthèse de vues, la construction de mosaïques. Cette méthode nous permet d'augmenter le champ de vision d'un microscope photonique et faciliter la supervision d'une station de micromanipulation.

Synthèse de Nouvelles Vues par Construction de Mosaïques

La première technique de synthèse de nouvelles vues présentée dans ce chapitre est la construction de mosaïques d'images. Cette méthode va permettre d'augmenter le champ de vision du microscope photonique. Nous présenterons comment la réaliser selon deux cas de figure. Elle sera appliquée à des scènes de micro-assemblage pour la supervision de stations de micromanipulation. Nous développerons comment tous les flux vidéos de la station pourront être centralisés dans le même référentiel que constitue l'image mosaïque de l'espace de travail.

4.1 Introduction

Notre objectif est d'augmenter le champ de vision du microscope photonique par reconstruction de l'espace de travail avec une grande résolution. Pour cela nous allons utiliser la technique du rendu à partir d'images nommée *mosaicing* d'images (construction de mosaïques). Il s'agit de la construction d'une image mosaïque de la scène observée à partir d'un ensemble d'images de cette scène, chaque image représentant une partie de la scène. Le champ de vision est virtuellement augmenté sans perte de résolution des images de départ. Dans ce domaine on trouve beaucoup d'applications dans la littérature :

- construction de cartes aériennes (D. Milgram 1975, D. Milgram 1977, S. Peleg 1981), ou sous-marines (Richard Marks, Stephen Rock et Michael Lee 1994, N. Gracias et J. Santos-Victor 2001, M. Borgetto 2005),
- création de photos panoramiques de 360° avec l'utilisation d'un logiciel comme Quicktime©VR qui permet une immersion virtuelle dans l'environnement (S. Chen et L. Williams 1993, Z. Zhu et al. 2006),

- construction de larges scènes virtuelles (P. Heckbert 1989, R. Kumar, P. Anandan et K. Hanna 1994, Richard Szeliski 1994),
- compression de séquences vidéos (M. Irani et al. 1996),
- numérisation de grands documents (Anthony Zappalá, Andrew Gee et Michael Taylor 1997, M. Pilu et F. Isgro 2002, G. Kumar et al. 2004, P. Shivakumara et al. 2006),
- stabilisation d'images dans des applications industrielles (C. Guestrin, F. Cozman et M. Simoes 1998),
- système de navigation en robotique mobile dans un environnement extérieur (J. Zheng et S. Tsuji 1992, N. R. Gracias 2002, H. Morita et al. 2006),
- augmentation du champ de vision des systèmes d'images, en particulier les vidéos microscopes photoniques (J. Asmuth et al. 2001, B. Potsaid, Y. Bellouard et J. Wen 2005, B. Appleton, A. Bradley et M. Wildermoth 2005).

On trouve également des logiciels commerciaux, avec des fonctions de création de panorama, qui sont proposés dans l'achat de certains appareils photos numériques.

Le *mosaicing* d'images se fait en deux étapes principales : le recalage d'images, où chaque image est rectifiée par rapport à une même référence, et le rendu de l'image, où les valeurs de tous les pixels de la mosaïque sont déterminées. La première étape, le recalage, consiste à corriger la déformation géométrique entre chaque image provoquée par le déplacement de la source d'images pendant l'acquisition. Cette étape permet de garder une cohérence géométrique de la mosaïque par rapport à une image de référence pour l'assemblage de toutes les images. Elle inclut éventuellement la reprojection des images selon le mode de visualisation souhaité (planaire, cylindrique ou sphérique). L'étape de rendu quand à elle correspond à une étape de traitement photométrique, c'est-à-dire le traitement de l'intensité des pixels de la mosaïque. Elle comprend la fusion des intensités des pixels dans les zones de recouvrement des images, la correction de l'illumination entre les images, le détournage de la mosaïque finale. La qualité de la mosaïque dépend de différents paramètres géométriques et photométriques, comme la distorsion optique, le parallaxe, le déplacement d'objets dans la scène, les réflexions spéculaires, les changements d'illumination, le vignetage¹, etc. Le premier facteur de détérioration visuelle, est la présence dans la mosaïque des bords des images assemblées (*seam* : littéralement couture en français). Une sous étape importante appelée fusion (*blending*) permet d'estomper la transition des images qui composent la mosaïque. Les différentes étapes de *mosaicing* sont ordonnées de la façon suivante :

- recalage géométrique (éventuellement photométrique) des images par rapport à une même référence,
- reprojection de tous les pixels des images dans un environnement spécifique, plan, cylindrique ou sphérique,

¹Vignetage : assombrissement de la périphérie d'une image provoqué soit par une insuffisance de l'objectif photographique, soit par l'utilisation d'un objectif dont le cercle-image ne couvre pas totalement le format du film. *source* <http://fr.wikipedia.org>.

- étape de la fusion (*blending*), qui corrige les différences d'illumination et fusionne les images pour obtenir une mosaïque sans transitions visibles dans l'image finale,
- la dernière étape est le détournage, qui découpe la mosaïque de telle sorte que l'image soit rectangulaire et la plus esthétique possible.

Il existe deux configurations possibles pour la réalisation d'une mosaïque. Selon l'abondance ou pas de texture dans la scène. Dans le cas où la scène est fortement texturée, on peut utiliser cette information photométrique pour déterminer la transformation entre chaque paire d'images pour le recalage. Dans ce cas il y a deux méthodes, une directe par corrélation de transformation projective et une indirecte en utilisant des indices invariants dans la scène. Maintenant, dans la configuration où la scène contient très peu de texture, par exemple quelques objets répartis dans un grand espace de travail sans texture, le déplacement de la source d'images est obtenu en utilisant le mouvement calibré de l'actionneur. Cette configuration nécessite un système de positionnement contrôlable pour déplacer la source d'images et une détermination du rapport métrique/pixel de la scène observée.

Nous allons présenter les deux configurations possibles. Dans le cas de la construction de mosaïque à faible texture, on utilise le recalage par mouvement calibré. Nous présenterons notre méthode de détermination du rapport métrique/pixel. Dans le cas de la construction de mosaïque à forte texture, nous évoquerons la méthode directe par corrélation de transformation projective. Nous discuterons pourquoi cette méthode ne sera pas retenue pour nos applications. Nous détaillerons alors plus amplement la méthode de recalage par l'étude des indices invariants. Nous utiliserons les méthodes proposées dans le chapitre 3 sur le calibrage faible. Par contre, dans la suite de la création d'une mosaïque à forte texture, nous proposerons de nouveaux algorithmes avec la fusion adaptative et une méthode de détournage automatique. Nous présenterons également un apport d'un point de vue applicatif, avec la réalisation de mosaïque à partir d'images issues d'un microscope photonique et la supervision d'une station de micromanipulation par incrustation en temps réel des images des sources d'images statiques ou mobiles dans la mosaïque.

4.2 Construction de mosaïque à partir d'images à faible texture

Cette catégorie de mosaïque ne peut être réalisée par les méthodes conventionnelles. En effet, l'information qui permet de déterminer le déplacement de la source d'images entre deux images est la texture, c'est-à-dire l'information photométrique de la scène (objets et fond). Dans certains cas, en micromanipulation, certaines zones du champ de travail n'ont pas de texture, comme une grande partie des scènes non naturelles. Par exemple dans l'environnement, hormis le substrat et les quelques microproduits à manipuler (parfois un seul) il n'y a rien d'autre. Ces zones sans texture rendent la reconstruction de l'espace de travail impossible par les méthodes habituelles. Comment déterminer le déplacement de la source d'images avec deux images complètement vides d'information ?

La solution commence par l'utilisation d'une source d'images montée sur des tables de positionnement pilotables formant un manipulateur cartésien (x,y,z) ainsi on connaît avec une grande exactitude le déplacement de la source d'images. L'environnement est balayé et les images sont assemblées comme pour un scanner de documents papiers.

4.2.1 Recalage par mouvement calibré

Quand le déplacement de la caméra est parfaitement connu, c'est-à-dire que les transformations entre les images sont connues, le recalage est immédiat et chaque image est rectifiée. Si la transformation est une simple translation, les images n'ont même pas besoin d'être rectifiées, l'image de la scène est découpée en bandelettes qui sont directement alignées. Cette particularité de joindre bord à bord directement les images de l'environnement pour reconstruire une mosaïque s'appelle le *stitching*. Elle a été utilisée par B. Rousso, S. Peleg et I. Finci (1997), S. Peleg et J. Herman (1997), Philippe Blanc, Eric Savaria et Farid Oudyi (2001) pour des mosaïques d'images satellitaires et Benjamin Potsaid, Yves Bellouard et John Wen (2003) pour l'augmentation du champ de travail dans des applications de micromanipulation biologique. L'avantage de cette approche est de pouvoir être utilisée sur des scènes qui contiennent peu de texture.

Toute la difficulté dans cette méthode est de connaître avec une grande précision le mouvement à donner à la caméra pour obtenir des images qui se mettent bout à bout pour construire la mosaïque. Il faut connaître la relation métrique/pixel en fonction des différentes distances de travail. Pour cela, il existe une étape essentielle à cette méthode, c'est la détermination de la taille d'un pixel de l'image. Le mode opératoire consiste à utiliser une mire calibrée, c'est-à-dire un objet dont on connaît parfaitement les dimensions. Il ne reste plus qu'à faire le rapport entre le nombre de pixels de l'objet et sa taille en mètre pour obtenir la taille d'un pixel. Il est important de déterminer avec précision la dimension de l'objet en pixels dans l'image. Pour cela deux techniques s'offrent à nous pour augmenter la précision de mesure, et avoir une mesure sub-pixélique. La première est l'interpolation d'image et la deuxième la super-résolution.

4.2.2 Interpolation vs Super-résolution

La limitation de la résolution des mires (en pixels) observées est déterminée par la taille de l'image elle-même ainsi que la résolution de l'optique de la source d'images. Pour augmenter la précision d'une mesure faite dans cette image, elle doit être de type sub-pixélique, c'est-à-dire avec une résolution inférieure à un pixel.

Pour cela on peut utiliser une méthode d'interpolation d'image. Cette méthode est réglée en trois temps. La première étape consiste à ré-échantillonner l'image, donc à l'agrandir numériquement de façon uniforme selon les deux dimensions, le ratio de la taille de l'image se trouve donc conservé. L'agrandissement de l'image n'est pas un simple zoom numérique qui agrandit visuellement la taille du pixel (pixélisation). Dans le cas de l'interpolation, la taille du pixel est conservée, elle est juste ré-échantillonnée. On obtient une image avec de l'information manquante (trous) entre chaque échantillon. Par exemple une interpolation par 2 nous permet d'obtenir une image 2 fois plus grande

avec un trou un pixel sur deux. La deuxième étape de l'interpolation consiste comme son nom l'indique à interpoler les valeurs des pixels manquants entre chaque échantillon. Les nouvelles valeurs interpolées sont calculées par une fonction polynomiale à un degré n plus ou moins important à partir des échantillons voisins. Le résultat permet d'obtenir une image plus grande que l'originale avec une définition très fine. Les mesures effectuées dans cette nouvelle image permettent de définir avec une plus grande résolution les caractéristiques des objets dans l'image.

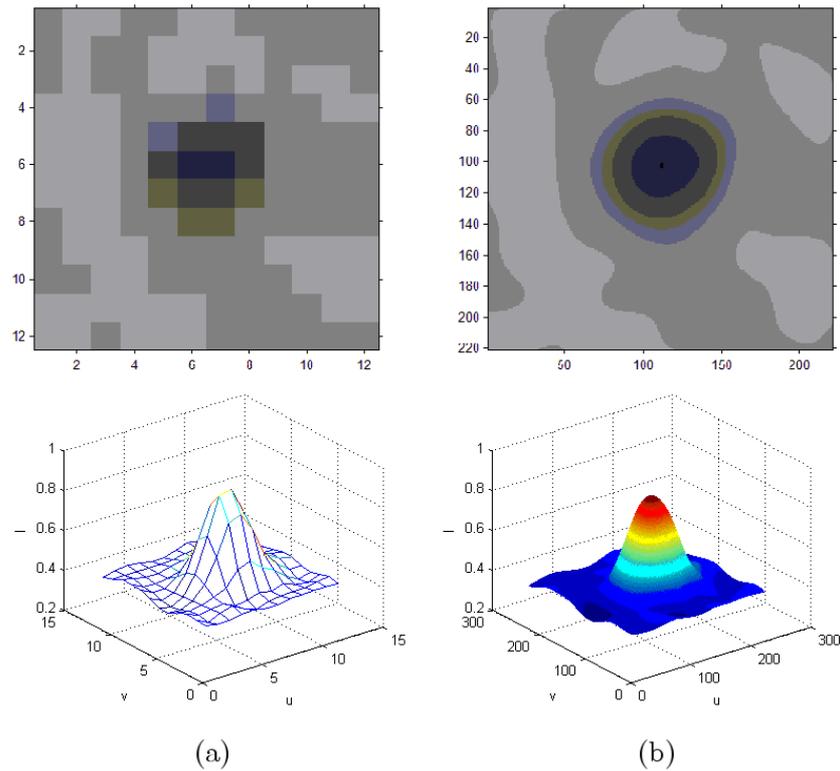


FIG. 4.1 – Exemple d'interpolation 2D : (a) image d'un disque en résolution d'origine et sa représentation en altitude, (b) image du même disque par interpolation $\times 20$ et sa représentation en altitude.

L'exemple de la Figure 4.1 où l'image est échantillonnée avec une taille 20 fois plus importante montre à quel point la définition de la nouvelle image est meilleure. Dans cet exemple il est facile de remarquer que la mesure du centre du disque est plus précise. Dans l'image originale le centre du disque a les coordonnées $x = 6$ pixels et $y = 7$ pixels. Dans l'image interpolée le centre du disque a les coordonnées suivantes $x_i = 110$ pixels et $y_i = 121$ pixels. La dernière étape d'une méthode d'interpolation est d'exprimer les mesures effectuées dans l'image interpolée dans le repère original. Le centre de la mire mesuré par interpolation 20 fois, est égale à $x = x_i/20 = 110/20 = 5,5$ pixels et de la même façon $y = 6,05$ pixels. La localisation est plus précise du fait que le résultat donne une mesure sub-pixélique avec une définition maximale inversement proportionnelle au

taux d'interpolation, dans ce cas de résolution $1/20 = 0,05$ pixel. Il faut faire attention aux mesures par interpolation, ce n'est pas parce que le taux d'interpolation est de 1000 fois, que la mesure est garantie à 0,001 pixel près. C'est juste la capacité de la définition qui peut atteindre cette valeur.

L'interpolation n'est pas la seule méthode qui permet d'augmenter la résolution des mesures dans l'image. On en retrouve une dans le domaine de la restauration d'images, appelée super-résolution. La super-résolution apparaît avec R. Tsai et T. Huang (1984). Elle consiste à retrouver une image en haute résolution (fidélité du détail) à partir d'un ensemble d'images basse résolution par extrapolation de la bande passante. La super-résolution est un système MISO (*Multi-Input Single-Output*). Une image haute résolution est une image avec un spectre contenant des hautes fréquences spatiales. Elle permet de restaurer une image fortement aliasée (spectre replié) provoqué par un sous-échantillonnage et donc un non respect du théorème de Shannon-Nyquist, ce qui provoque une perte d'information dans l'image, traduit visuellement par une pixélisation. Cette technique de restauration permet à partir d'un ensemble d'images basse résolution, de retrouver une image de plus grande résolution, c'est-à-dire de zoomer sur une image de façon numérique et non optique. La Figure 4.2 montre un exemple de super-résolution, à partir d'une séquence vidéo de 300 images basse résolution elle permet de retrouver une image haute résolution.

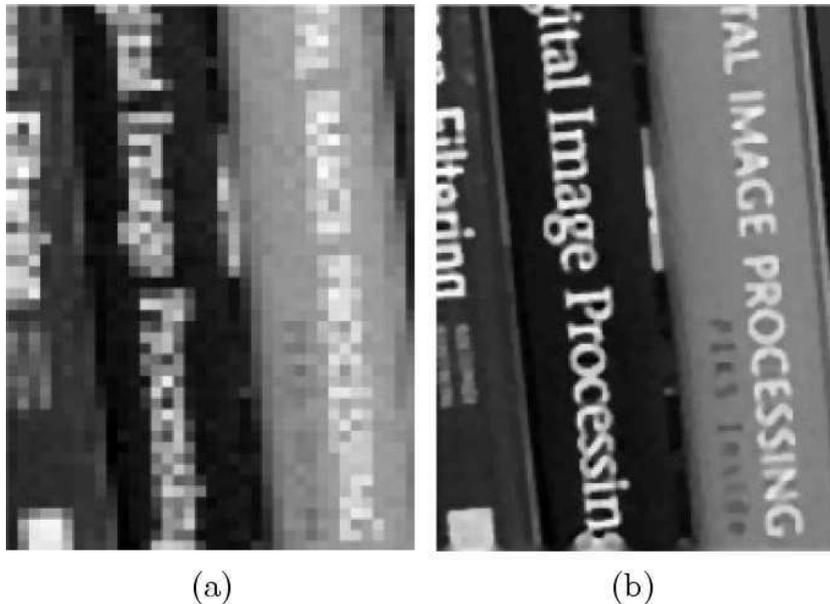


FIG. 4.2 – Exemple de super-résolution : (a) une des images originales basse résolution d'une séquence vidéo de 300 images, (b) la restauration par super-résolution. (*Images S. Farsiu et al. ©2004 J. Wiley & Sons, Inc.*)

Il existe différentes techniques de super-résolution. La méthode fréquentielle s'appuie sur les propriétés de translation de la transformée de Fourier, ainsi que la théorie de

l'échantillonnage. Présentée par R. Tsai et T. Huang (1984), elle consiste en la suppression de l'aliasing des images basse résolution par les coefficients de la transformée de Fourier discrète. Les images sont enregistrées avec un mouvement global de translation inter-images sub-pixéllique. Pour des mouvements plus complexes, W. Su et S. Kim (1994) proposent de découper l'image en sous blocs et d'estimer le mouvement de translation de chacun.

La méthode d'échantillonnage multicanaux propose une reconstruction basée sur le théorème d'échantillonnage généralisé de A. Papoulis (1977) et modifié de J. Brown (1981). Le premier exemple d'application en super-résolution est donné par H. Ur et D. Gross (1992). D'après le théorème de Papoulis, le signal original peut être reconstruit à partir de ces K signaux dégradés. Pour cela, on passe les K signaux dans K filtres appropriés et on additionne les sorties. Pour la super-résolution, il suffit de considérer que les K filtres correspondent à la fonction de transfert optique du système ainsi qu'à une translation globale de l'image.

La technique d'interpolation à partir d'une grille d'échantillonnage non-uniforme est basée sur une interpolation spatiale (comme précédemment) d'échantillons non-uniformément répartis. Par la connaissance du mouvement sub-pixéllique dans la séquence d'images acquises, il est possible de rapporter tous les pixels des images de la séquence dans le même référentiel. On obtient un sur-échantillonnage des images, mais dont les pixels ne sont pas uniformément répartis. Cette image est ensuite interpolée et ré-échantillonnée sur une grille uniforme. On peut alors appliquer sur cette image haute résolution une méthode classique de restauration d'image pour déconvoluer la fonction de transfert optique (D. Keren, S. Peleg et R. Brada 1988, T. Komatsu et al. 1993, P. Magain, F. Courbin et S. Sohy 1998).

Le problème de super-résolution peut-être modélisé par l'inversion d'un modèle de dégradation. Une image quelconque peut se modéliser selon un modèle de dégradation :

$$\underline{Y}(t) = M_d(t)\underline{X}(t) + \underline{V}(t) \quad (4.1)$$

Où $\underline{Y}(t)$ est l'image collectée en basse résolution, $M_d(t)$ représente le système de vision (effet de mouvement, flou, effet de couleur, sous-échantillonnage), $\underline{V}(t)$ représente le bruit et $\underline{X}(t)$ représente l'image originale (super-résolue) à estimer. Cette méthode consiste à modéliser le problème physique, puis à l'inverser pour obtenir l'image d'origine. Une des premières méthodes qui utilise la modélisation de la dégradation est présentée par S. Peleg, D. Keren et L. Schweitzer (1987) avec la méthode *Iterative Back-Projection* (IBP). Ce modèle est amélioré en intégrant les problèmes d'occultation et de transparence par M. Irani et S. Peleg (1993). Plusieurs techniques de filtrage inverse sont utilisées, soit par un filtre de Wiener (A. Erdem, M. Sezan et M. Ozkan 1992) ou par des filtres adaptatifs (A. Patti, A. Tekalp et M. Sezan 1998).

Des méthodes stochastiques sont également utilisées. Plus particulièrement l'estimation statistique par une approche Bayésienne. Elle permet d'introduire des connaissances *a priori* sur la solution sous forme de densité de probabilité. On estime une image super-résolue en utilisant des algorithmes de recherche de maximum *a posteriori* (MAP). Cette approche a été utilisée par R. Schultz et R. Stevenson (1996) avec une distribution de

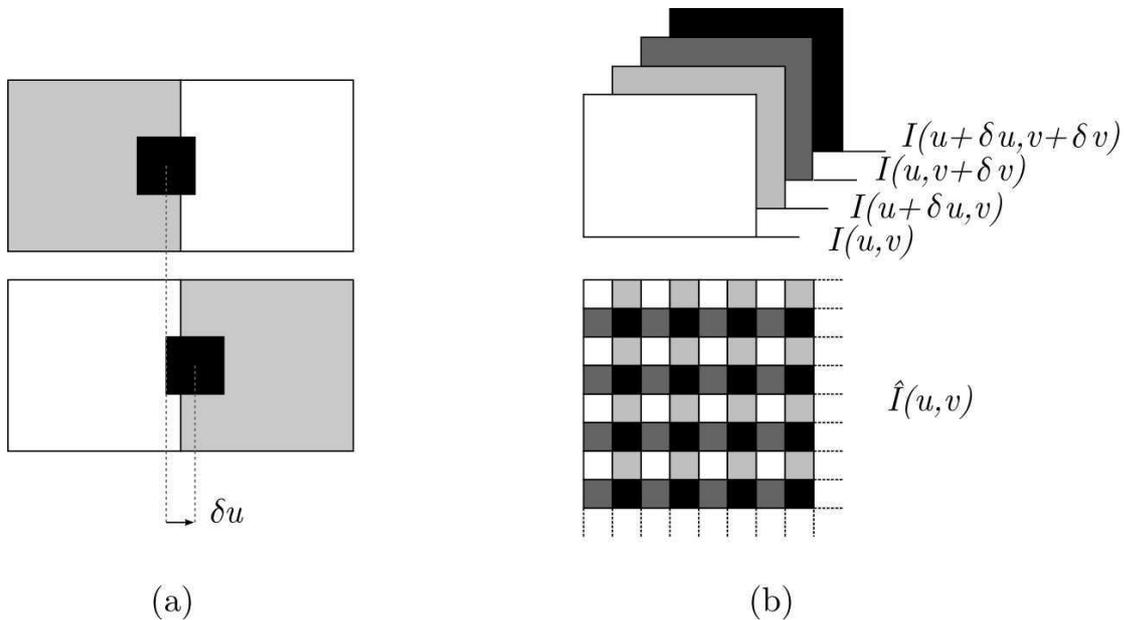


FIG. 4.3 – Méthode de super-résolution par *shift-and-add* : (a) information du pixel gauche et droite en fonction du chevauchement d'un élément déplacé sub-pixéliquement, (b) exemple de reconstruction de l'image (u, v) à partir de 4 images déplacées de façon sub-pixélique selon les deux directions u et v .

Huber ou par R. Hardie, K. Barnard et E. Armstrong (1997) avec une distribution gaussienne. On peut aussi retrouver une image super-résolue en utilisant d'autres types d'algorithmes de recherche comme le maximum de vraisemblance (*maximum likelihood*). Pour plus de détails on peut se référer aux travaux de B. Tom et A. Katsaggelos (1994).

Il existe aussi une méthode basée sur la théorie des ensembles, essentiellement la méthode de projection sur des ensembles convexes (POCS : *Projection Onto Convex Sets*). Le principal intérêt de la méthode POCS est la simplicité avec laquelle il est possible d'introduire des informations *a priori* de natures très différentes sur la solution désirée. Il suffit pour cela de définir un ensemble convexe qui les incorpore. La méthode des POCS dans le domaine de la super-résolution est l'œuvre de D. Youla et H. Webb (1982), repris dans l'estimation de modèle de dégradation par P. Eren, M. Sezan et A. Tekalp (1997).

La dernière méthode est une technique relativement récente et simple. Le *Shift-and-Add*, présenté par Michael Elad et Yacov Hel-Or (2001), utilise directement le capteur de la source d'images. En jouant sur l'échantillonnage des rayons lumineux continus qui traversent le capteur et plus particulièrement ceux qui sont à cheval entre deux pixels il est possible d'augmenter la résolution de l'image. Par exemple, si un rayon lumineux est à cheval sur deux pixels mais que sa position est décentrée plus à gauche, seul le pixel de gauche aura l'information du rayon lumineux. En déplaçant la source d'images de façon

sub-pixéllique à un moment donné, le rayon lumineux sera un peu plus décentré à droite, et à ce moment l'information du pixel bascule, sur le pixel de droite (Figure 4.3-a). Cette méthode nécessite soit de connaître le déplacement sub-pixéllique d'une source d'images libre, soit de contrôler la position de la source d'images avec précision. Chaque image déplacée sub-pixélliquement est ré-échantillonnée sur une grille. La Figure 4.3-b montre l'acquisition de 4 images avec des déplacements sub-pixélliques selon les deux directions (u, v) . Les images originales sont décalées selon leur déplacement dans l'espace, c'est la partie *Shift* (décalage) de la méthode, puis ajouter dans un même référentiel *Add* (ajout). Si on désire augmenter la précision de l'image de n fois, on fait une acquisition de chaque image avec un déplacement de $1/n$ pixel selon les deux directions. Un exemple d'utilisation de la méthode *Shift-and-Add* est présenté par Sina Farsiu et al. (2004). Cette approche est rapide à mettre en oeuvre lorsque le déplacement de la source d'image est contrôlable avec une grande précision, ce qui correspond à notre cas.

D'après la liste non exhaustive des différentes techniques de super-résolution, nous en sélectionnons une seule, la méthode par *Shift-and-Add*. Nous gardons également la méthode d'interpolation, ces deux méthodes vont nous permettre d'augmenter la précision de nos mesures pour la détermination du rapport métrique/pixel. Afin d'en garder une seule, nous implémentons les deux méthodes et nous allons les comparer sur une même image. Le choix du grossissement de l'image est fixé à 10. Pour la méthode *Shift-and-Add* nous balayons une partie d'un pignon de montre avec des déplacements sub-pixélliques de $1/10$ de pixels selon les deux directions u et v . Au final nous obtenons 100 images de basse résolution, qui sont par la suite décalées et ajoutées dans un même repère pour obtenir l'image super-résolue. Dans le cas de l'interpolation une seule image de basse résolution est nécessaire, nous utilisons une interpolation de type bicubique, qui correspond à l'interpolation de l'intensité d'un pixel en fonction des intensités des 8 pixels connexes. La comparaison des deux images avec un grossissement par 10 est représenté à la Figure 4.4.

On remarque que les résultats sont très proches l'un de l'autre. Par la méthode de super-résolution *Shift-and-Add*, l'image est légèrement pixelisée par bloc de pixels qui correspond à la taille d'un pixel d'origine. En effet dans chaque bloc on compte les 100 pixels des acquisitions obtenues par mouvement sub-pixéllique. La méthode par interpolation est très proche de la méthode par super-résolution. Cependant elle impose beaucoup moins de contraintes à l'acquisition. Pour que la super-résolution devienne intéressante au niveau de la résolution, il faudrait ajouter une étape supplémentaire de restauration pour diminuer le flou de l'image.

En conclusion, la technique de super-résolution est très alléchante mais nécessite une mise en oeuvre complexe et longue. Elle pourrait pourtant apporter beaucoup en microscopie photonique, par exemple pour obtenir un microscope photonique de haute résolution à partir d'un microscope photonique de basse résolution permettant ainsi de garder ces propriétés physiques (grande profondeur de champ). Pour un choix de simplicité nous ne gardons que la méthode d'interpolation pour augmenter la précision de nos mesures dans l'image.

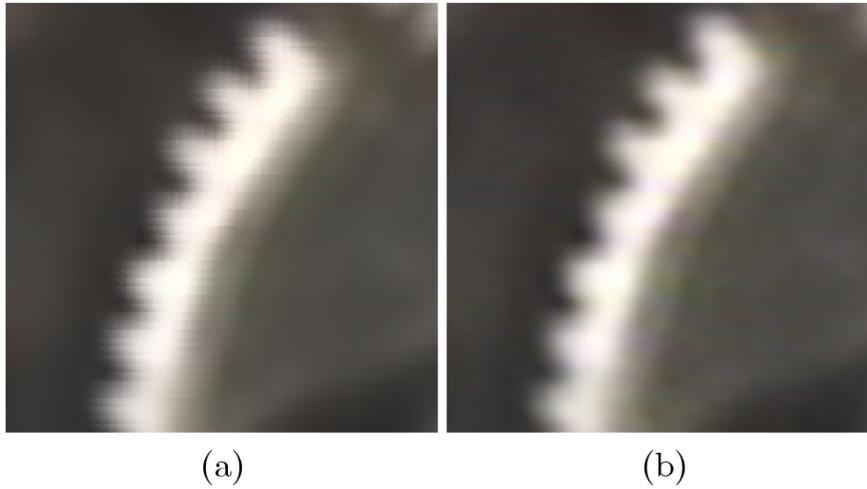


FIG. 4.4 – Grossissement $10\times$ d’une partie d’un pignon de montre : (a) par super-résolution *Shift-and-Add*, (b) par interpolation bicubique.

4.2.3 Étalonnage par détection de contour à partir d’un motif carré

L’étalonnage a pour but de déterminer le rapport métrique/pixel, il correspond à l’inverse de la résolution de la source d’images dans les conditions d’utilisation. Il permet d’ajuster le déplacement de la source d’images afin de mettre bout à bout les images pour avoir une image mosaïque. La première méthode employée ici est fondée sur l’utilisation d’une mire réalisée en microtechnologie. Celle-ci se compose d’un masque perforé de motifs carrés. L’éclairage de la mire est de type diascopique (éclairage par en dessous). Nous allons utiliser les contours de la mire ainsi qu’une analyse robuste des points pour déterminer en pixels la taille de la mire. La mesure sera effectuée dans une image interpolée pour augmenter la résolution et l’exactitude de la mesure. La première étape consiste à acquérir une image de qualité par la source d’images. Pour cela nous allons utiliser un filtre passe-bas. Ce type de filtrage peut se faire de deux manières, soit de façon spatiale, c’est-à-dire directement sur l’image, soit de façon temporelle avec un ensemble d’images. La méthode temporelle permet de mieux supprimer les bruits provenant du capteur de la caméra. Ce filtre consiste en la fusion de plusieurs images d’une même scène, l’image filtrée peut s’écrire de la façon suivante,

$$I_f(u, v) = \frac{1}{n} \sum_1^n I_n(u, v) \quad (4.2)$$

Avec I_n les différentes images prises à des instant différents. L’exemple de la Figure 4.5 représente une image filtrée à partir d’une séquence vidéo de 100 images. Chaque image originale est sévèrement bruitée, dans la réalité les caméras qui équipent les microscopes photoniques sont de meilleure qualité.

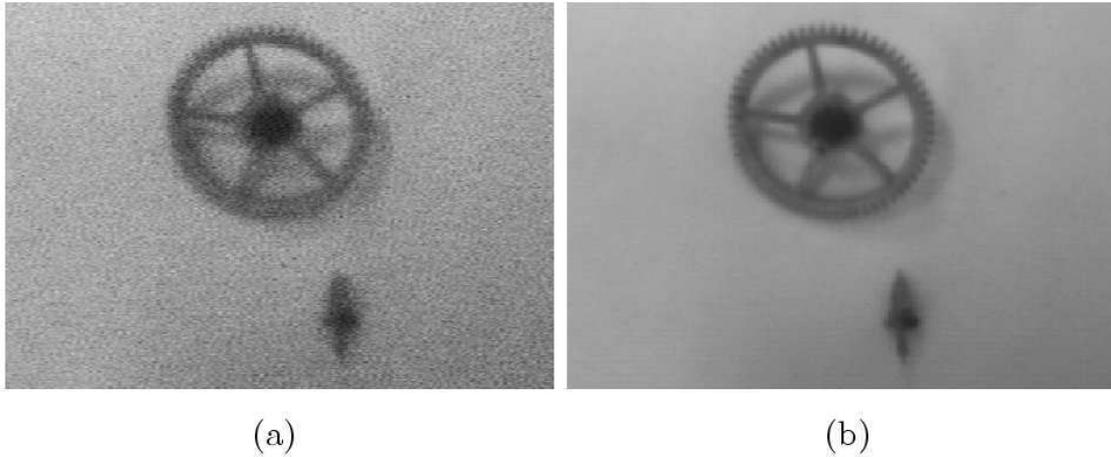


FIG. 4.5 – Image d'un pignon et d'un axe de montre : (a) une des images originales de la séquence vidéo, (b) application d'un filtre passe bas temporel avec les 100 images de la séquence vidéo.

Cette approche temporelle est utilisée essentiellement pour obtenir des images statiques, pour des images de scènes mobiles on choisira plutôt le filtrage spatial.

Nous allons maintenant déterminer les arêtes du masque carré. La notion de contour étant liée à celle de variation, il est évident qu'une telle définition nous amène tout naturellement vers une évaluation de la variation de l'intensité en chaque pixel. Le gradient, en un pixel d'une image numérique, est un vecteur caractérisé par son amplitude et sa direction, il se définit comme suit,

$$\vec{\nabla}I = \frac{\partial I}{\partial x} \vec{x} + \frac{\partial I}{\partial y} \vec{y} \quad (4.3)$$

avec $\vec{\nabla}I$ le vecteur du gradient de l'image I . Généralement dans le traitement de l'image on utilise seulement le module du gradient pour pouvoir le représenter sous forme d'une image. Il s'écrit :

$$|\vec{\nabla}I| = \left[\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2 \right]^{1/2} \quad (4.4)$$

Nous allons utiliser seulement l'arête supérieure et l'arête inférieure, par conséquent nous allons utiliser seulement un calcul de variation mono-dimensionnelle discret, c'est-à-dire la variation dans une seule direction dans l'image. Notre image de contour selon l'axe y devient :

$$\frac{\partial I}{\partial x} = I(x, y + 1) - I(x, y) \quad (4.5)$$

Ce qui revient à appliquer un seul masque de convolution pour obtenir les arêtes supérieure et inférieure de la mire. Au final, on obtient les points de contour des deux

arêtes du carré de la mire. Tous les points des arêtes ne sont pas pris en compte à cause de la diffraction et de l'interpolation des coins de la mire qui s'arrondissent. Pour cela on délimite les points des arêtes à la portion linéaire des bords, en même temps cela nous permet de supprimer les erreurs de détermination de contour. La Figure 4.6 montre les trois premières étapes effectuées, l'interpolation suivie de la détection des contours supérieur et inférieur et enfin le filtrage des points des arêtes du carré.

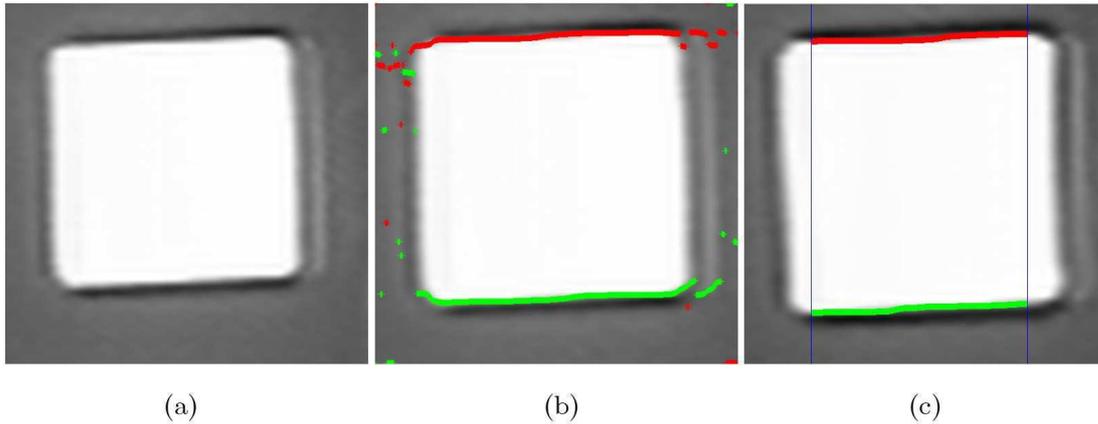


FIG. 4.6 – Premières étapes pour la mesure de la relation métrique/pixel : (a) interpolation $\times 10$ de l'image de la mire, (b) détection des points de contour selon l'axe y , (c) limitation des points pour obtenir seulement l'arête supérieure et inférieure.

Maintenant pour chaque ensemble de points nous allons déterminer les droites de régression pour obtenir les équations des droites des arêtes supérieure et inférieure du carré. La méthode la plus connue pour estimer l'équation d'une droite par régression est la méthode des moindres carrés. Pour augmenter la robustesse du procédé de mesure, nous allons utiliser une variante, les moindres carrés médians (P. Rousseeuw 1984). En effet, la méthode des moindres carrés n'est pas robuste aux données erronées. Par exemple, la présence d'un seul point aberrant peut faire échouer l'estimation d'une droite (Figure 4.7).

La méthode des moindres carrés médians permet l'ajustement de modèles linéaires même en présence de données aberrantes. Cette méthode est basée sur trois étapes : recherche d'une première estimation fiable, élimination des données aberrantes et estimation finale des paramètres du modèle par la méthode des moindres carrés classiques mais en exploitant uniquement les données filtrées. Cette méthode rappelle la méthode de RANSAC utilisé dans le chapitre 3 pour le calibrage. Un certain nombre de p -uplets sont alors tirés au hasard. Pour chacun d'eux, les paramètres du modèle sont estimés. Par exemple avec deux points (x_i, y_i) et (x_j, y_j) , les paramètres de la droite sont :

$$a_{ij} = \frac{y_j - y_i}{x_j - x_i} \tag{4.6}$$

$$b_{ij} = y_i - a_{ij}x_i$$

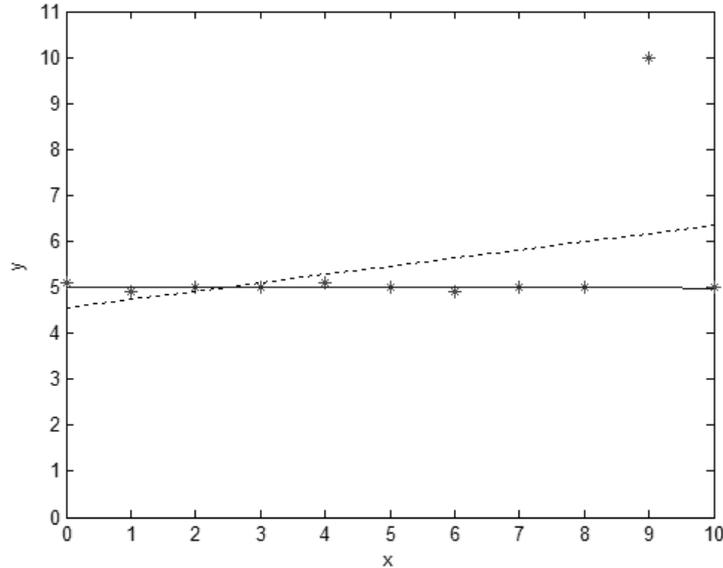


FIG. 4.7 – Comparaison entre une droite de régression par moindres carrés (trait pointillé) et par moindres carrés médian (trait plein).

Les écarts au modèle pour toutes les données sont alors calculés puis classés par ordre croissant ($k \in [1, n]$),

$$e_k = y_k - a_{ij}x_k - b_{ij} \quad (4.7)$$

En première approximation, nous retiendrons comme paramètres du modèle, ceux obtenus avec le p-uplet conduisant à la valeur médiane du vecteur des écarts la plus faible. Ces paramètres correspondent à une bonne approximation et peuvent être utilisés pour filtrer les données. Si on s'autorise une probabilité d'erreur égale à Q , le nombre de p-uplets aléatoires à tester est donné par le plus petit entier q tel que :

$$[1 - (1 - \epsilon\%)^p]^q \leq Q \quad (4.8)$$

Avec $\epsilon\%$ le pourcentage estimé de données aberrantes. Un résultat de statistique, nous fournit une estimation de l'écart type des écarts au modèle à partir de la médiane la plus faible que nous noterons m ,

$$\sigma = 1,4826 \left(1 + \frac{5}{q-p}\right) \sqrt{m} \quad (4.9)$$

Cette estimation de σ est fiable sous l'hypothèse que la distribution des écarts au modèle des données non aberrantes suit une loi normale. Cette estimation peut tolérer jusqu'à 50 % de données aberrantes. L'élimination des données erronées est réalisée en ne conservant que celles pour lesquelles l'écart au modèle est inférieur à $2,5\sigma$. Ce qui

correspond à retenir 99 % des données dont l'écart au modèle suit une loi normale d'écart-type σ . Maintenant on peut estimer de façon précise le modèle. Cette dernière est simplement obtenue par une minimisation au sens des moindres carrés à partir des données filtrées. On applique les moindres carrés médians aux deux ensembles de points obtenus sur l'arête supérieure et inférieure de la forme carré de la mire. On obtient les deux droites de régression qui déterminent avec précision et robustesse les arêtes de la mire carré. La mire est placée de façon grossière dans la scène, cela signifie que l'on n'a pas contrôlé l'orientation de la mire, il faut par conséquent en tenir compte dans la détermination de la largeur de la mire. On calcule le centre d'une des droites en tenant compte de son inclinaison par rapport à l'axe horizontal. A partir de ce centre on détermine la droite perpendiculaire en prenant l'inverse du coefficient directeur de la droite de l'arête supérieure. Le croisement de cette droite perpendiculaire aux deux arêtes définit le segment qui correspond à la largeur du carré de la mire. La Figure 4.8 montre les dernières étapes pour l'obtenir.

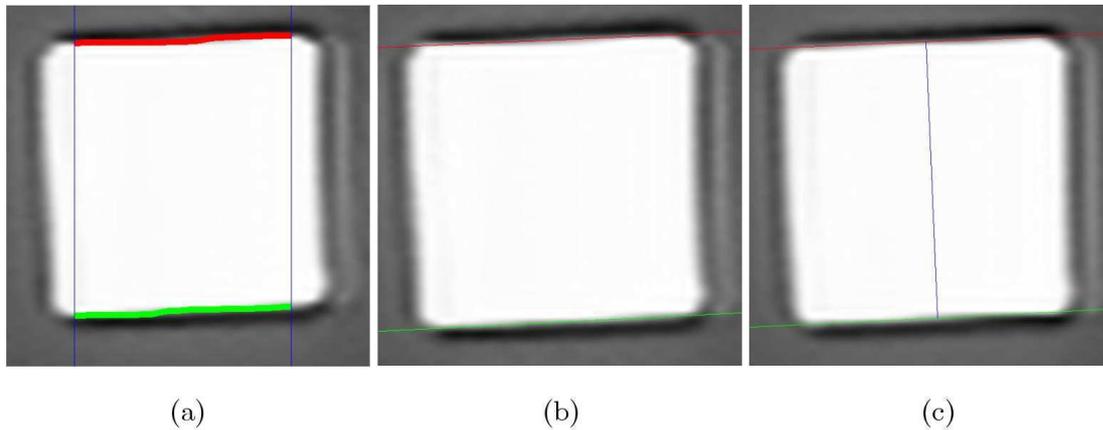


FIG. 4.8 – Les dernières étapes de la mesure de la relation métrique/pixel : (a) ensemble des points de l'arête supérieure et inférieure de la mire, (b) détermination des droites de régression, des arêtes, par moindre carré médian, (c) on utilise le segment perpendiculaire aux arêtes de la mire pour mesurer sa largeur.

Connaissant la taille réelle du motif de la mire réalisée en microtechnologie, on peut calculer le rapport métrique/pixel. On mesure la largeur du carré en pixels et on le divise par sa largeur réelle. Pour valider et vérifier la détermination de la taille du pixel nous allons pour plusieurs mesures déterminer le rapport métrique/pixel puis calculer la largeur réelle de la mire. Les différentes mesures vont nous permettre de connaître aussi la répétabilité de la méthode. On réalise un ensemble de 500 mesures, nous obtenons les résultats suivant :

$$\begin{aligned}
 \text{Moyenne} &= 2052,48 \mu\text{m} \\
 \text{Ecart type} &= 0,291 \mu\text{m} \\
 \text{Erreur} &= 52,48 \mu\text{m}
 \end{aligned}
 \tag{4.10}$$

La première remarque va directement sur l'erreur de la mesure qui très importante par rapport à la taille réelle de la mire. Par contre l'écart type est très faible, ce qui montre une certaine robustesse dans la détermination de la taille de la mire. Pour expliquer l'erreur importante de la mesure nous émettons deux hypothèses par rapport au système d'éclairage diascopique. Notre première hypothèse se réfère au phénomène de diffraction. Lorsque la lumière traverse le masque perforé de la mire par en dessous, la lumière est légèrement déviée par les bordures. Le résultat est que la taille de la mire dans l'image ne correspond pas totalement à sa taille réelle. Cette hypothèse n'est pas en grande partie la cause de l'erreur de mesure car la diffraction joue un rôle trop minime sur la mire. Ce qui nous permet d'introduire notre deuxième hypothèse, basée sur la nature de l'éclairage diascopique. Notre système d'éclairage est composé d'un bloc à base de mini-tubes fluorescents au néon. Même si ce type d'éclairage a un rayonnement thermique plus faible que les lampes à incandescence, la température du plateau où est déposée la mire n'est pas complètement basse. La mire est réalisée en microtechnologie par un dépôt de métaux sur une plaque de verre. Prenons par exemple un type de verre très utilisé dans les laboratoires pour sa résistance à la chaleur et aux agents chimiques, les verres borosilicate, le plus connu est le Pyrex (1915). Il est composé de 80% de silice, 13% d'anhydride borique, 4% de soude et 3% d'alumine. Son coefficient de dilatation est de $32.10^{-7}K^{-1}$. Si nous prenons notre erreur moyenne de $52,48 \mu m$, cela correspondrait à une élévation de la température de la mire de $16,4^{\circ}C$. Après une journée de fonctionnement, l'ordre de grandeur d'élévation de la température du système d'éclairage ne semble pas aberrant.

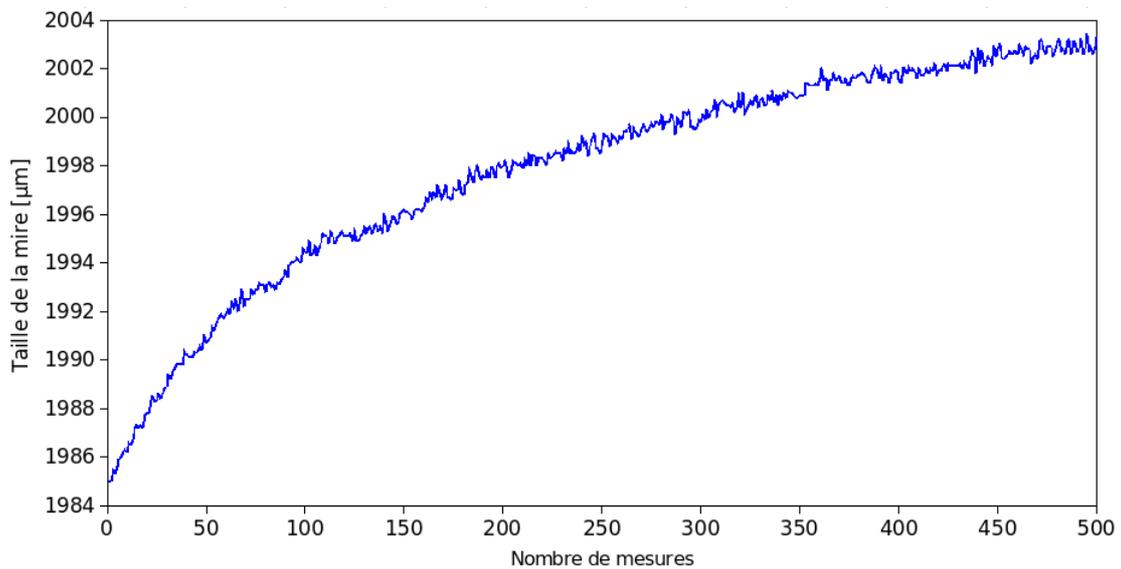


FIG. 4.9 – Graphique représentant la taille de la mire en fonction des différentes mesures dans le temps.

Pour vérifier cette hypothèse nous mesurons la taille de la mire avec 500 mesures intercalées dans le temps de chacune 10 s. Nous débutons les mesures en allumant l'éclairage juste avant, donc à froid. Nous traçons les résultats à la Figure 4.9. Cette figure montre que la taille de la mire se dilate progressivement de mesure en mesure, c'est-à-dire dans le temps. La forme de la courbe représente un premier ordre, typique dans ce système qui monte en température. Il n'est pas difficile d'imaginer que la taille de la mire va se stabiliser une fois que le système d'éclairage a atteint sa température finale. Notre hypothèse est vérifiée, la température de l'éclairage diascopique agit directement sur la mire et donc sur nos mesures d'étalonnage. La solution consiste à utiliser une méthode d'étalonnage qui utilise un système d'éclairage épiscopique, diminuant l'interaction du rayonnement thermique, sans pour autant le supprimer, seul les problèmes de diffractions disparaîtront.

4.2.4 Étalonnage par mesure de surface d'un motif circulaire

Nous allons utiliser une mire réalisée en microtechnologie, composée d'un disque argenté opaque d'un diamètre de 2000 μm . Le système d'éclairage est de type épiscopique (au dessus) composé de plusieurs mini-tubes fluorescents. La stratégie pour déterminer la taille de la mire sera différente, car sa forme est différente de celle mise en œuvre précédemment. La mire est un disque opaque. Le début du traitement est identique à la mire carrée, on fait plusieurs acquisitions pour filtrer l'image, puis on applique une interpolation bicubique pour augmenter la précision des mesures. Contrairement à la mire carrée nous n'allons pas utiliser le contour de l'objet, mais sa surface. L'image filtrée et interpolée de la mire est binarisée (noir ou blanc) avec un seuil automatique. On calcule la surface du disque en utilisant le moment M_{00} de l'image, rappel du calcul du moment :

$$M_{mn} = \sum_{i=1}^{|I(x,y)|_x} \sum_{j=1}^{|I(x,y)|_y} x_i^m y_j^n I(x_i, y_j) \quad (4.11)$$

D'après la surface mesurée dans l'image du disque il est facile d'utiliser la propriété géométrique qui lie la surface au rayon d'un disque ($S = \pi r^2$) pour déterminer le diamètre de celui-ci. La connaissance du diamètre en pixels (image) et du diamètre en unité métrique (scène) permet de calculer le rapport métrique/pixel. Tout comme la méthode précédente nous effectuons 500 mesures afin d'évaluer la répétabilité et la précision des mesures. Les mesures sont réalisées avec un éclairage à chaud, c'est-à-dire que nous avons attendu que l'éclairage atteigne sa température de fonctionnement pour débiter les mesures. Nous obtenons les résultats suivants :

$$\begin{aligned} \text{Moyenne} &= 1995,05 \mu m \\ \text{Ecart type} &= 4,061 \mu m \\ \text{Erreur} &= 4,95 \mu m \end{aligned} \quad (4.12)$$

L'erreur moyenne de la mesure par éclairage diascopique était de 52,48 μm , maintenant par éclairage épiscopique elle n'est plus que de 4,95 μm . Le changement du

système d'éclairage permet bien de diminuer l'erreur de mesure, sans pour autant la supprimer complètement. Par contre, la méthode du motif carré nous donnait un écart type de $0,291 \mu m$, maintenant cette valeur atteints $4,061 \mu m$. Cette augmentation de l'écart type est directement lié à la méthode de détermination de la taille de la mire. Dans notre cas, la méthode par la mesure de surface est moins robuste et moins répétable que la méthode de détection de contour et d'estimation robuste. Quoiqu'il en soit, un écart type et une erreur moyenne d'environ $4 \mu m$ sur une mire d'un diamètre de $2000 \mu m$, sont tout à fait convenables et suffisants pour un étalonnage. Par rapport à la taille du disque nous obtenons une erreur moyenne de $0,24\%$.

La principale caractéristique de ces méthodes réside dans le fait que le rapport métrique/pixel du système optique dépend de la focale et de la distance scène-source d'images. Pour nous permettre, de connaître ce rapport à n'importe quel zoom (distance) nous allons calculer ce rapport pour différentes valeurs de grandissement et en construire un modèle.

4.2.5 Étalonnage dynamique

Le rapport métrique/pixel dépend directement de la distance de travail pour certaine source d'image, ce qui est notre cas. Par conséquent, pour chaque distance de travail on doit connaître la relation métrique/pixel, ce qui oblige à créer un modèle d'évolution de la taille du pixel en fonction de la distance de travail (ou du zoom). Pour la mesure du rapport nous allons utiliser la méthode précédente, avec un éclairage épiscopique et une mire formée d'un disque opaque de $2000 \mu m$. On monte la source d'images sur une table de positionnement permettant de contrôler les translations de la source d'images selon les trois directions (x, y, z) . Nous allons calculer la taille du pixel réel dans l'image S_p pour différentes valeurs de distance de travail Z_m . Pour se garantir une mesure parfaite, il faut, de manière automatique, gérer le centrage de la mire au centre de l'image. Cela permet d'une part de se placer dans une zone d'image où les distorsions optiques sont minimales, et d'autre part, de garder la mire dans le champ de vision de la source d'images. Pour se faire nous implémentons un asservissement visuel de type *look-and-move*. Après chaque mesure de la taille du pixel S_p , nous calculons les coordonnées du centre de gravité du disque de la mire (\bar{x}, \bar{y}) en utilisant les moments $(M_{10}/M_{00}, M_{01}/M_{00})$. En utilisant les erreurs de la position de la mire $(\epsilon_{\bar{x}}, \epsilon_{\bar{y}})$ par rapport au centre de l'image et la taille du pixel S_p , les tables de positionnement corrigent la position de la source d'images. Nous mesurons 500 tailles de pixel différentes en fonction de plusieurs distances de travail dans un intervalle bien précis (zone linéaire) qui correspond aux zones de travail susceptibles d'être reconstruites. Nous représentons les résultats des mesures à la Figure 4.10.

On note que la variation du rapport S_p en fonction de la distance de travail Z_m est linéaire dans notre intervalle d'utilisation. Par conséquent, nous allons en déduire un modèle affine. Il sera estimé en utilisant une régression par moindres carrés médians. Nous obtenons la fonction suivante :

$$S_p = -1,251.10^{-3} \times Z_m + 90,236 \quad (4.13)$$

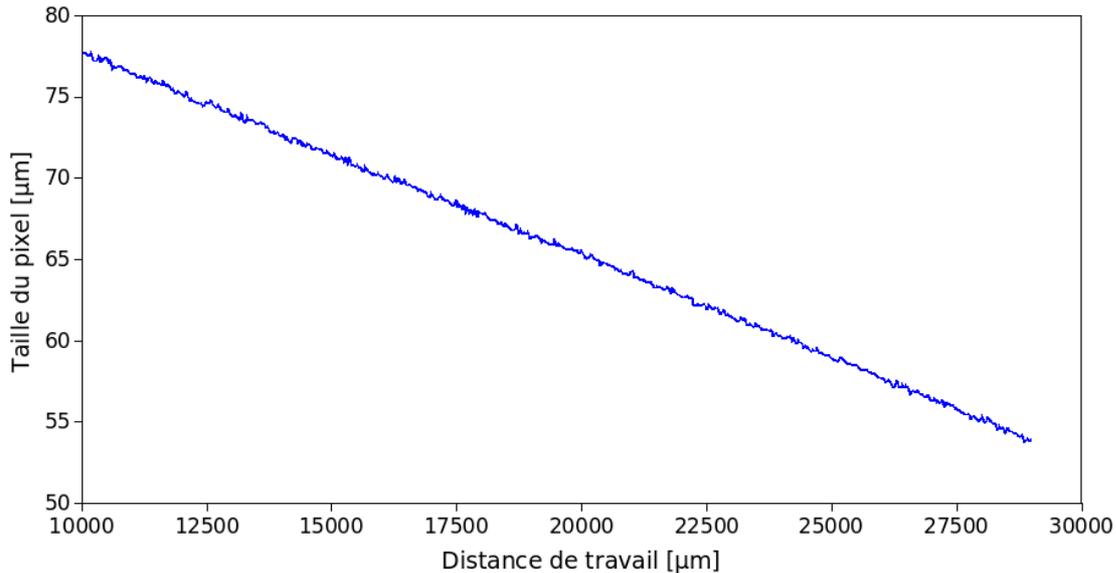


FIG. 4.10 – Taille du pixel en fonction de la distance de travail de la source d’image.

Maintenant, dans une plage déterminée de distance de travail, nous pouvons déterminer la relation métrique/pixel par le biais de notre modèle établi. Ce modèle évite d’étalonner à chaque fois que la distance de travail change. Il ne reste plus qu’à appliquer la méthode de construction de mosaïque par mouvement calibré sur une scène faiblement texturée.

4.2.6 Exemple d’application

Pour appliquer la méthode de construction de mosaïque par mouvement calibré, nous allons utiliser une scène qui comporte très peu de texture. Le système est composé des éléments suivants (Figure 4.11) :

- une mini-caméra endoscopique : longueur de 20 *mm*, diamètre de 5 *mm*, avec un angle de vue de 90° et un capteur CCD couleur (8 bits) de résolution 768×576 pixels,
- trois tables de positionnement montées en (x, y, z) avec une résolution de $2,5\mu m$ et une course de 55 *mm* par table,
- un système de contrôle (SMS60) pour piloter les moteurs pas à pas des tables de positionnement,
- une carte d’acquisition Matrox Meteor II pour récupérer les images de la mini-caméra,
- un PC avec le logiciel Matlab pour commander l’ensemble du système via la liaison série connectée au contrôleur SMS60.

La méthode par mouvement calibré utilise le contrôle du déplacement de la source d’images pour assembler les images directement bout à bout. La mini-caméra est montée

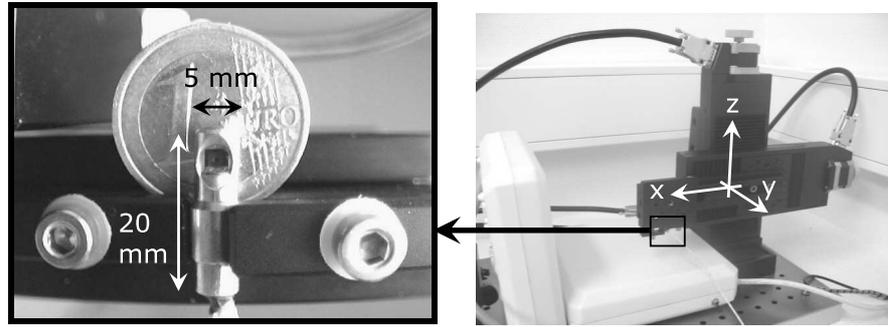


FIG. 4.11 – La mini-caméra endoscopique embarquée sur les tables de positionnement montées en (x, y, z) .

sur les tables de positionnement (Figure 4.11) et le système peut balayer l'espace de travail en obtenant un ensemble d'images locales. Pour cela nous allons étalonner les déplacements des tables pour quels correspondent à la largeur de chaque image. Le pas de déplacement d_m (métrique) correspond à la largeur de l'image d_p (déplacement en pixels). On peut écrire :

$$d_m = d_p \times S_p \quad (4.14)$$

S_p est le rapport métrique/pixel qui dépend de la distance de travail Z_m . Connaissant le modèle établi précédemment, le déplacement de la source d'images s'écrit :

$$d_m = d_p(-1,251.10^{-3} \times Z_m + 90,236) \quad (4.15)$$

Les images délivrées par le système de vision sont fortement soumises aux distorsions optiques provenant du grand angle de vue de notre mini-caméra (90°). Pour diminuer ces distorsions, dans notre mosaïque finale, nous découpons une petite zone autour du centre de chaque image locale, car les distorsions y sont plus faibles. En effet la projection du centre optique est situé à peu près au centre de l'image. Si on découpe dans chaque image, le déplacement en pixel n'est plus la largeur de l'image, mais la largeur de la zone à découper. On applique le balayage avec le mouvement calibré à une scène contenant un bâti de montre et quelques engrenages, le reste de la scène est complètement vide. Le rapport métrique/pixel (facteur d'échelle) est estimé à $50 \mu m/\text{pixel}$. On balaye l'ensemble de la scène avec les courses maximums des tables de déplacement soit une surface de $55 \text{ mm} \times 55 \text{ mm}$. L'image résultat obtenue fait une taille de 1100×1100 pixels, elle est représentée à la Figure 4.12.

4.2.7 Conclusion

La construction de mosaïque par la méthode du mouvement calibré est la seule qui permet de créer une mosaïque si l'image manque d'informations. Une grande partie des zones de la scène montrée dans l'image de la Figure 4.12 ne contient aucune texture, il était donc impossible d'estimer le déplacement en utilisant l'information qui n'existe

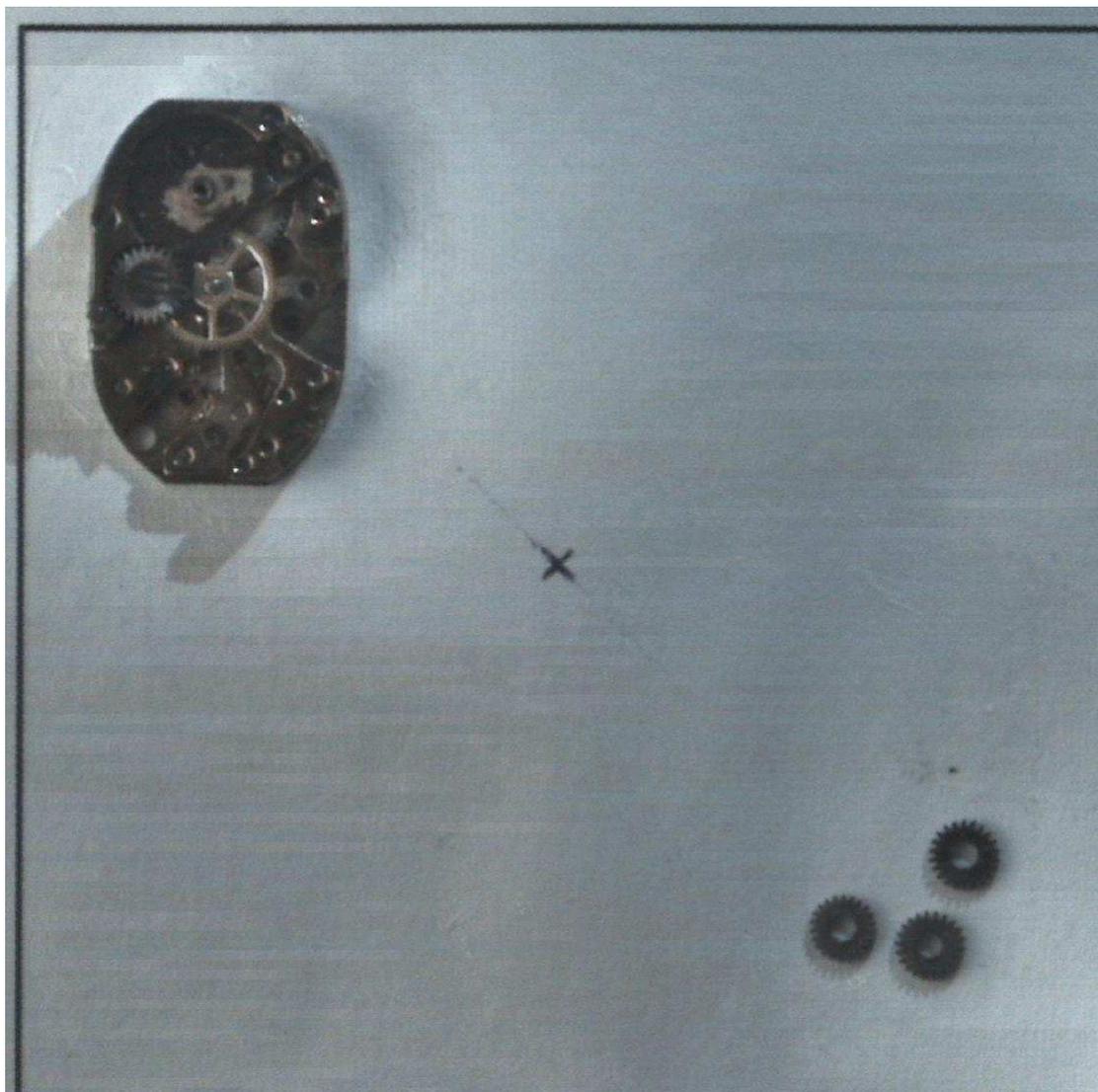


FIG. 4.12 – Construction d’une mosaïque par mouvement calibré d’une scène de $55\text{ mm} \times 55\text{ mm}$ avec un facteur d’échelle de $50\ \mu\text{m}/\text{pixel}$ soit une image de dimensions de 1100×1100 pixels.

pas dans l'image. Cette méthode a certains inconvénients, elle nécessite d'une part un matériel spécial pour contrôler la translation de la source d'images et d'autre part un étalonnage précis du système. En plus la source d'images doit être le plus parallèle possible à la scène (contrainte fronto-parallèle) sinon les images assemblées sont décalées. L'inconvénient majeur est l'étalonnage, à ces dimensions il existe tellement de perturbation, notamment sur l'éclairage, qu'il est très difficile d'obtenir des résultats précis. Par exemple dans le cas d'une mosaïque très longue, si l'étalonnage n'est pas effectué dans un environnement contrôlé (température ambiante, rayonnement éclairage, etc.) les erreurs sur la relation métrique/pixel vont s'accumuler tout au long du balayage d'image. Et comme les petits ruisseaux forment les grandes rivières, les dernières images assemblées dans la mosaïque ont des erreurs de décalage de plusieurs pixels. Pour s'affranchir de la conduction thermique de l'éclairage dans l'air ambiant il faut utiliser des éclairages spéciaux appelés éclairage froid, souvent composés de diodes électroluminescentes ou de systèmes à lampes déportées par fibres optiques. Cependant, pendant les essais nous avons remarqué que même, un éclairage froid engendrait des variations sur les mesures. Il faut également faire attention aux matériaux utilisés pour les mires. Il est bien évident qu'à cette échelle seul un procédé de microtechnologie permet de créer des mires, dans ce cas tous les matériaux ne sont pas exploitables, il faut quand même le garder en mémoire lors de la conception de mire. En effet très fréquemment lors d'achat de microscope (pourtant de marque), on vous donne une mire en microtechnologie fait d'une plaque de verre avec des dépôts de métaux qui se place sur un éclairage diascopique (chaud) pour étalonner votre matériel fraîchement acquis! La solution ultime pour garantir qu'il n'y a pas de décalage entre les images assemblées dans la mosaïque est d'effectuer le recalage directement par l'image, bien sûr il faut que l'information soit suffisante dans l'image.

4.3 Construction de mosaïques à partir d'images à forte texture

Maintenant dans l'hypothèse où les images de la scène sont très texturées c'est-à-dire avec suffisamment d'informations pour pouvoir estimer le déplacement de la source d'images directement entre les images. Dans ce cas, il existe deux autres méthodes pour recalculer les images. La méthode directe utilise une fonction de corrélation par transformation projective, d'itération en itération elle recalcule les images dans un même référentiel. Elle est également appelée méthode référencée intensité car elle utilise l'intensité de tous les pixels pour le recalage. La deuxième méthode de recalage utilise les points d'intérêts dans les images pour estimer la transformation entre elle, c'est la méthode référencée indices invariants.

4.4 Recalage à partir de l'intensité

Cette méthode, estime la transformation projective (matrice de colinéation), entre chaque paire d'images, par une méthode itérative. Elle utilise une fonction de corrélation entre les images pour corriger leurs différences géométriques afin de converger vers

des images transformées dans un même référentiel, donc recalé. Un grand nombre de méthodes de ce type a été développé. La première méthode remonte à E. Barnea et H. Silverman (1972) qui proposaient d'utiliser une corrélation spatiale appelée norme L1. Puis par la suite C. Kuglin et D. Hines (1975) utilisent une corrélation de phase par FFT (*Fast Fourier Transform*) qui permet d'estimer la translation entre deux images en utilisant les propriétés de l'espace de Fourier. Szeliski et Shum introduisent une corrélation par transformation projective (Richard Szeliski 1994, Richard Szeliski et Heung-Yeung Shum 1997, Heung-Yeung Shum et Richard Szeliski 1997). Cette corrélation utilise la fonction de corrélation de la deuxième image transformée pour estimer les nouvelles transformations à lui appliquer, et de façon itérative l'image se recalé par rapport à la première. Ce qui revient à estimer la matrice de colinéation G entre les deux images à recalé. Pour cela il faut déterminer les 8 paramètres que constituent cette matrice (le 9^{ième} paramètre est égale à 1). La matrice G est initialisée arbitrairement, puis elle est mise à jour en fonction de l'expression suivante :

$$G \leftarrow (\mathcal{I} + D_{inc})G \quad (4.16)$$

avec \mathcal{I} la matrice identité et D_{inc} la matrice incrémentale. La matrice incrémentale est estimée par rapport à l'erreur de transformation entre les deux images à recalé I et I' . Elle va petit à petit transformer l'image I' dans le même référentiel que l'image I . La méthode directe se résume à un problème de minimisation de l'erreur $E(d)$ entre I et la transformation courante de l'image \tilde{I}' provenant de l'image originale I' . L'erreur de la fonction de corrélation est définie comme une somme des différences des intensités pour chaque pixel des images,

$$E(d) = \sum_i \left[\tilde{I}'(\tilde{p}') - I(p) \right]^2 \quad (4.17)$$

avec \tilde{p}' les points de l'image \tilde{I}' et p les points de l'image I . Cette erreur peut être approximée par le premier ordre de la série de Taylor :

$$E(d) \approx \sum_i \left[\tilde{I}'(p') + \nabla \tilde{I}'(p') \frac{\partial \tilde{p}'}{\partial d} d - I(p) \right]^2 \quad (4.18)$$

avec p' les points de l'image I' et $d = (d_0, \dots, d_8)$ le vecteur des éléments de la matrice incrémentale. On obtient, après une écriture compacte, l'erreur de transformation en fonction de la matrice incrémentale,

$$E(d) = \sum_i \left[g_i^T J_i^T d + e_i \right]^2 \quad (4.19)$$

où $e_i = \tilde{I}'(p') - I(p)$ est l'erreur de l'intensité, $g_i^T = \nabla \tilde{I}'(p')$ est le gradient de \tilde{I}' et $J_i = \frac{\partial \tilde{p}'}{\partial d}$ est le Jacobien des points ré-échantillonnés \tilde{p}' . L'équation précédente est un problème de type moindre carré, il existe une simple solution en posant l'équation sous la forme,

$$Ad = -b \quad (4.20)$$

avec,

$$A = \sum_i J_i g_i g_i^T J_i^T \quad (4.21)$$

et avec la matrice hessienne (matrice carrée de ses dérivées partielles secondes),

$$b = \sum_i e_i J_i g_i \quad (4.22)$$

qui représente le gradient accumulé ou le résidu. Ces équations peuvent être résolues en utilisant un solveur SPD (*Symmetric Positive Definite*) comme en utilisant une décomposition de Cholesky. Via cette résolution, la matrice incrémentale D_{inc} est calculée et la matrice de colinéation G est mise à jour pour obtenir une nouvelle image \tilde{I}' , ainsi de suite jusqu'à tendre vers une erreur de transformation entre les deux images proche d'un seuil pré-défini. La Figure 4.13 montre un résumé sous forme de diagramme de la méthode directe initialement présentée dans Heung-Yeung Shum et Richard Szeliski (1997). Dans certains cas la méthode directe ne peut travailler qu'avec des images qui ont des petites transformations, c'est-à-dire que les deux images à recalage doivent être proches l'une de l'autre. De plus, un des inconvénients de cette méthode itérative, est qu'il est possible de converger vers un minimum local. La principale raison est le nombre important de paramètres qui donnent trop de degrés de liberté dans l'estimation du recalage. C'est pour cela que l'on préfère choisir un modèle de transformation plus restreint que les 8 paramètres de la matrice de colinéation, par exemple un modèle à 3 paramètres. En fait, ces paramètres définissent une transformation à partir des trois angles de rotation de l'image. Moins il y a de paramètres à utiliser et plus la méthode est rapide et robuste face aux erreurs. Malgré ce modèle plus simple, pour des longues séquences d'images, les erreurs de recalage s'accumulent pour aboutir à une erreur finale importante mais que l'on peut réduire par une méthode d'alignement global. C'est en partie pour ces différentes raisons (risque de convergence vers un minimum local et erreur d'alignement global), que nous allons nous orienter vers une méthode alternative basée sur les indices invariants.

4.5 Recalage à partir des indices invariants

Dans cette méthode de recalage, la matrice de colinéation est estimée en utilisant les relations géométriques de points d'intérêts appariés entre les deux images à rectifier. On retrouve la première utilisation d'indices invariants pour la construction de mosaïque dans le papier de I. Zoghلامي, O. Faugeras et R. Deriche (1997). A partir des paires de points en correspondance, la transformation projective via la matrice de colinéation peut être estimée de façon linéaire (méthode DLT voir le chap. 3 sur le calibrage), et affiné par des approches statistiques ou probabilistes, comme l'utilisation d'estimateur de maximum de ressemblance (ML : *Maximum Likelihood*) par David Capel et Andrew

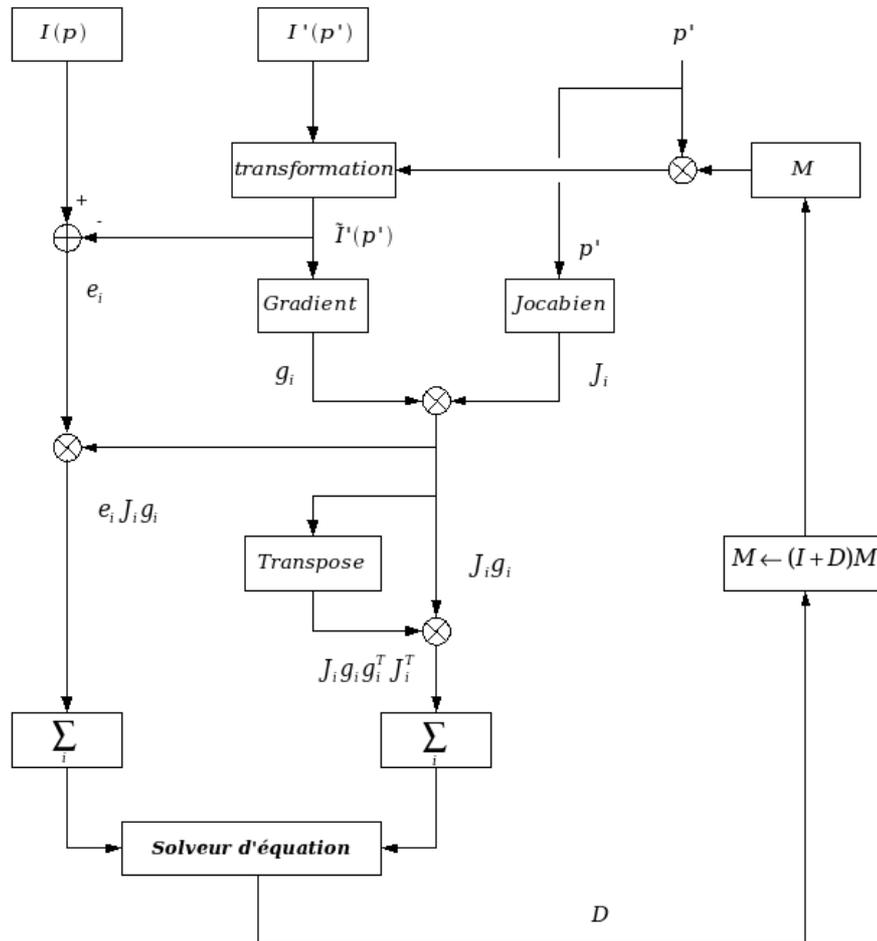


FIG. 4.13 – Diagramme résumant la méthode directe pour le recalage d'images.

Zisserman (2001). Comme pour l'estimation de la matrice de colinéation dans le chapitre du calibrage, les différentes étapes du recalage sont :

- détection d'indices invariants dans les deux images à recaler,
- mise en correspondance des indices,
- filtrage des erreurs de mise en correspondance avec RANSAC,
- estimation de la transformation projective par la méthode linéaire direct DLT,
- transformation de l'image à recaler par la matrice de colinéation estimée.

Étant donné que les grandes étapes, hormis la transformation de l'image, sont présentées et développées dans la partie calibrage, nous allons simplement rappeler le déroulement de chaque étape. Pour étayer les étapes, nous allons illustrer chaque résultat en s'appuyant sur deux images à recaler provenant d'un microscope photonique. Les deux images originales de départ sont représentées à la Figure 4.14. La paire d'images représente une partie d'une puce électronique d'EPROM (*Erasable Programmable Read Only Memory*),

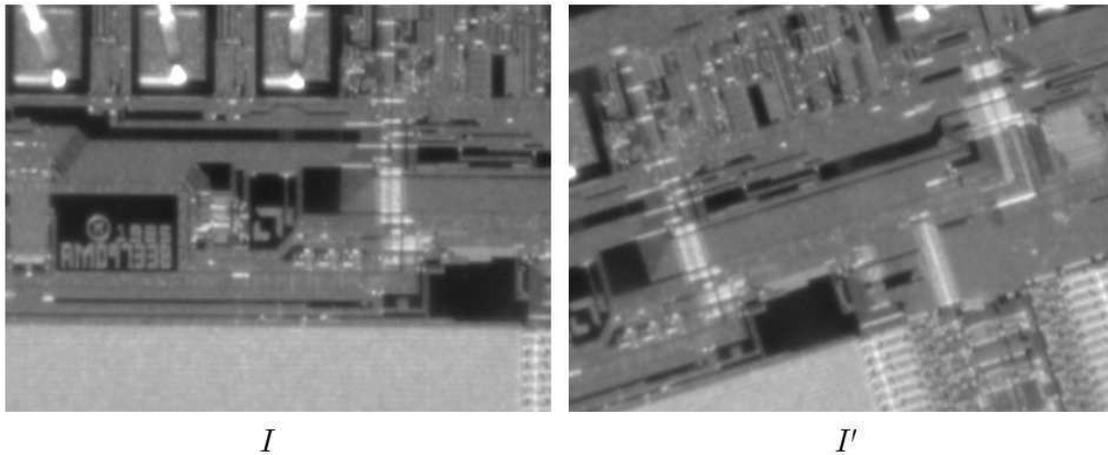


FIG. 4.14 – Paire d’images d’une puce électronique d’EPROM (facteur d’échelle 450 nm/pixel , taille de chaque image 640×480 pixels).

elles ont un facteur d’échelle de 450 nm/pixel pour des images qui ont une dimension de 640×480 pixels. Le but étant de recalibrer l’image I' par rapport à l’image de référence I .

Dans la suite de cette section nous allons décrire le procédé de recalage pour seulement deux images I et I' . Le procédé est identique sur un ensemble d’images prises deux à deux (I', I''), (I'', I'''), etc.

4.5.1 Détection des indices invariants

Nous utilisons l’algorithme du Harris Simplexe, basé sur le détecteur de coin Harris. Nous réglons le nombre de points désirés à 400 points. Nous obtenons pour les deux images deux ensembles de points, représentés à la Figure 4.15.

4.5.2 Mise en correspondances des indices

Nous appliquons le III-ZNSSD, notre méthode de corrélation, sur les deux ensembles de points afin d’en déterminer leur correspondance. En ayant la connaissance de qui correspond avec qui, il sera facile d’estimer la transformation projective subie sur les ensembles de points. Le résultat de la mise en correspondance est présenté à la Figure 4.16-a.

4.5.3 Estimation robuste de la transformation projective

Malgré l’utilisation du III-ZNSSD il reste toujours des erreurs d’appariement, qui proviennent en grande partie du bruit de l’image et des textures répétitives des objets observés. Comme dans le chapitre pour le calibrage nous allons utiliser l’algorithme RANSAC, pour filtrer les ensembles de points appariés en utilisant un modèle de transformation projective. Cette transformation modélisée par la matrice de colinéation est

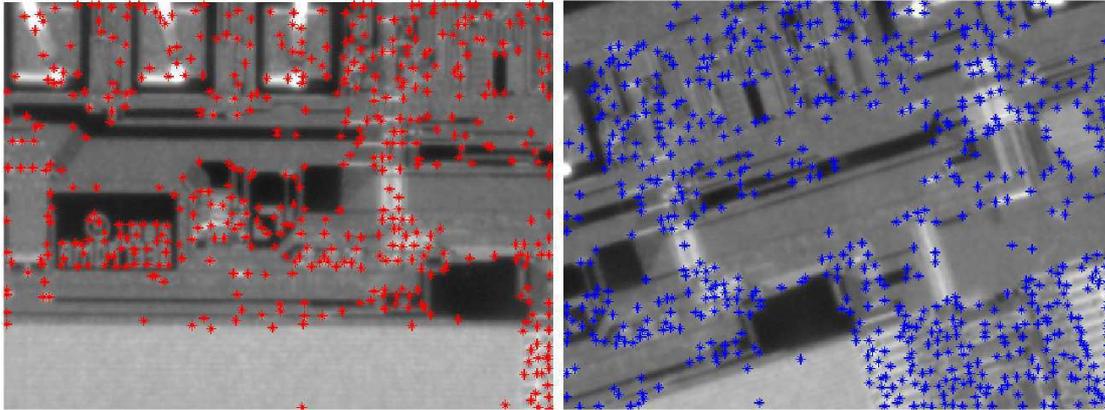


FIG. 4.15 – Indices invariants détectés dans les images avec le détecteur Harris simplexe (400 points).

calculée par la méthode classique du DLT. Après filtrage, la matrice de colinéation est estimée une dernière fois avec les deux ensembles de point appariés. La différence entre, avant et après RANSAC, est montrée à la Figure 4.16. Elle met bien en valeur l'action du filtrage de la méthode RANSAC. Au final nous obtenons la matrice de colinéation de la transformation de l'image I à l'image I' , il ne reste plus qu'à rectifier l'image I' par rapport à l'image I .

4.5.4 Transformation de l'image

Dès que la matrice de colinéation G est connue entre les images I et I' ($I' = GI$) nous pouvons transformer l'image I' pour la ramener dans le repère de l'image I . Connaissant la transformation projective de matrice G de I vers I' la transformation de I' vers I est de matrice G_r inverse de G , donc $G_r = G^{-1}$. Il existe deux méthodes pour transformer une image en fonction d'une transformation géométrique quelconque. La première méthode est appelée *forward mapping* (plaquage de texture vers l'avant), elle consiste à transformer l'ensemble des points p' de l'image I' en une nouvelle image I'_r composée des pixels p'_r en utilisant directement la formule $I'_r = G_r I'$. L'étape de transformation d'une image est une étape gourmande en temps de calcul, car elle nécessite de transformer tous les points de l'image, ce qui correspond à $|I'(x', y')|_x \times |I'(x', y')|_y$ soit pour une image de 640×480 pixels, 307200 points à calculer. Les nouvelles positions des points sont calculées par la multiplication par G_r , or dans ce cas les positions obtenues peuvent être des valeurs non entières, c'est-à-dire qu'il faut positionner un point avec des coordonnées à virgule (ex. : $x'_r = 12,234$ et $y'_r = 233,725$) sur une grille (image résultat) avec seulement des valeurs entières. Il se pose des questions dans ce cas : quelle position à affecter réellement au pixel dans l'image résultat, mais aussi quelle intensité lui attribuer. Pour cela nous utilisons plusieurs niveaux d'interpolation polynomiale. En fonction de la position plus ou moins proche des pixels connexes (voisin) l'intensité de ce pixel prendra des valeurs plus ou moins composées de ces pixels connexes. La première

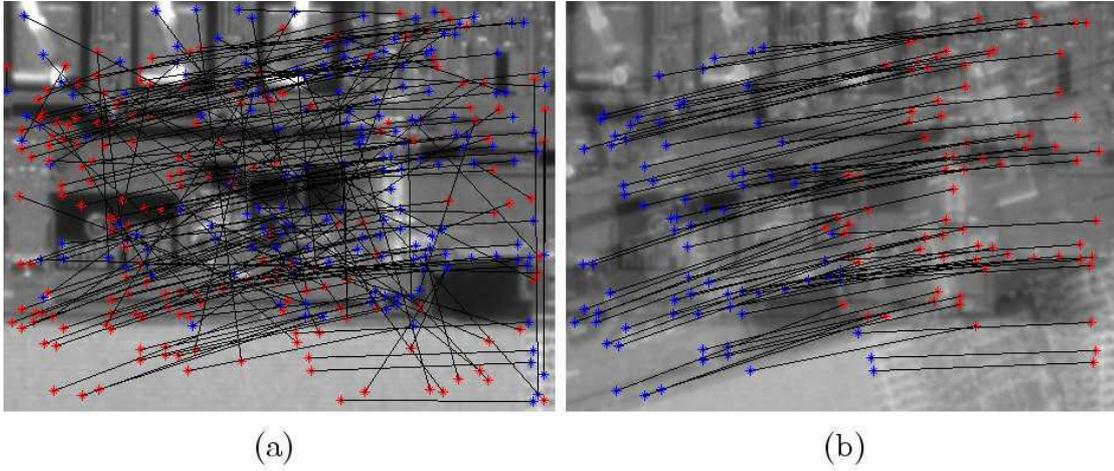


FIG. 4.16 – Représentation des points appariés : (a) avant RANSAC, (b) après RANSAC.

interpolation, d'ordre 0, appelée interpolation au voisin le plus proche, est la plus simple des interpolations :

$$\widehat{I}'_r(\hat{x}'_r, \hat{y}'_r) \approx I'_r(\lfloor x'_r \rfloor, \lfloor y'_r \rfloor) \quad (4.23)$$

avec \widehat{I}'_r l'image interpolée à partir de l'image transformée I'_r par G_r et $\lfloor * \rfloor$ la fonction d'arrondi à la valeur entière. Cette interpolation est la plus simple car on tronque la virgule à la valeur la plus proche, et pour l'affectation de l'intensité du pixel elle ne tient pas compte des autres intensités des pixels connexes. Par exemple cette interpolation si elle est utilisée dans le cas d'un changement de taille d'une image, laisse apparaître du aliasing c'est-à-dire des effets de créneau sur l'image résultat. Bien évidemment cette solution simple ne permet pas d'avoir une image résultat de grande qualité. L'interpolation d'ordre 1, nommé l'interpolation bilinéaire, cette fois-ci tient compte de l'intensité des quatre pixels connexes pour calculer l'intensité du pixel à placer dans l'image résultat. Elle est définie par la fonction suivante :

$$\widehat{I}'_r(\hat{x}'_r, \hat{y}'_r) \approx \begin{bmatrix} 1 - x'_r & x'_r \end{bmatrix} \begin{bmatrix} I'_r(x_{r1}, y_{r1}) & I'_r(x_{r2}, y_{r1}) \\ I'_r(x_{r2}, y_{r2}) & I'_r(x_{r1}, y_{r2}) \end{bmatrix} \begin{bmatrix} 1 - y'_r \\ y'_r \end{bmatrix} \quad (4.24)$$

avec $I'_r(x_{ri}, y_{ri})$ les quatre pixels connexes (Nord, Sud, Est et Ouest). Cette fonction d'interpolation, définit l'intensité du pixel, en fonction des pixels connexes. Après développement de l'équation précédente, nous obtenons la fonction polynomiale suivante :

$$\widehat{I}'_r(\hat{x}'_r, \hat{y}'_r) \approx cc_1 + cc_2 x'_r + cc_3 y'_r + cc_4 x'_r y'_r \quad (4.25)$$

avec a_i les coefficients d'intensités des pixels connexes, c'est-à-dire leurs intensités. L'interpolation bilinéaire n'utilise que les quatre pixels connexes les plus proches. On peut

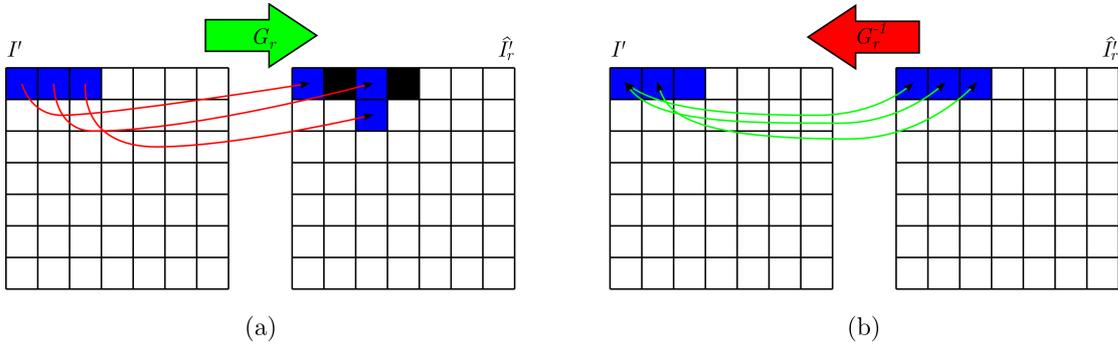


FIG. 4.17 – Méthode de transformation d’une image : (a) *forward mapping*, (b) *backward mapping*.

également utiliser des pixels connexes supplémentaires de plus en plus loin par rapport au pixel de référence, comme par exemple ceux placés à la position Nord-Est, Sud-Est, Sud-Ouest et Nord-Ouest. Dans ce cas l’interpolation utilise les 8 pixels connexes les plus proches. Nous obtenons donc une interpolation de degré supérieur à 1, appelée interpolation bicubique. Elle est définie par la fonction polynomiale suivante :

$$\widehat{I}_r(\hat{x}'_r, \hat{y}'_r) \approx \sum_{i=0}^3 \sum_{j=0}^3 cc_{ij} x_r^i y_r^j \quad (4.26)$$

avec cc_{ij} les différents coefficients d’intensité des pixels connexes. Cette dernière interpolation est la plus utilisée, car elle permet d’estimer l’intensité du pixel recherché en fonction de l’environnement complet de celui-ci, c’est-à-dire avec ces 8 pixels connexes. Malgré les efforts pour déterminer l’intensité du pixel le plus juste possible pendant la transformation, la méthode par *forward mapping* a un inconvénient majeur, lié au fait d’arrondir la position du pixel calculé à une valeur entière. En effet, il se peut que dans l’image résultat, certains pixels ne reçoivent aucune affectation d’intensité. L’image résultat contient alors des trous (manque d’information), des pixels noirs sont répartis un peu partout dans l’image. Le mécanisme de cette détérioration de l’image est représentée à la Figure 4.17-a.

Il existe une deuxième méthode de transformation d’image permettant de contrecarrer l’inconvénient précédent. La méthode *backward mapping* (plaquage de texture vers l’arrière) comme son nom l’indique utilise un procédé de projection en arrière. Au lieu de projeter tous les points de l’image I' vers l’image résultat \widehat{I}_r , on va déterminer l’origine de projection dans l’image I' pour chaque point de l’image résultat \widehat{I}_r . Cela permet de garantir que tous les points de l’image résultat auront une affectation d’intensité. Le procédé est représenté à la Figure 4.17-b. Étant donné la recherche en arrière, c’est la transformation inverse G_r^{-1} qui est utilisée pour déterminer le projeté de chaque point de l’image recalé. Nous avons besoin de balayer tous les points de l’image résultat, or nous ne connaissons pas la taille ni la forme exacte de cette image car elles dépendent de la transformation elle-même. Cette méthode est un peu plus compliquée que la méthode

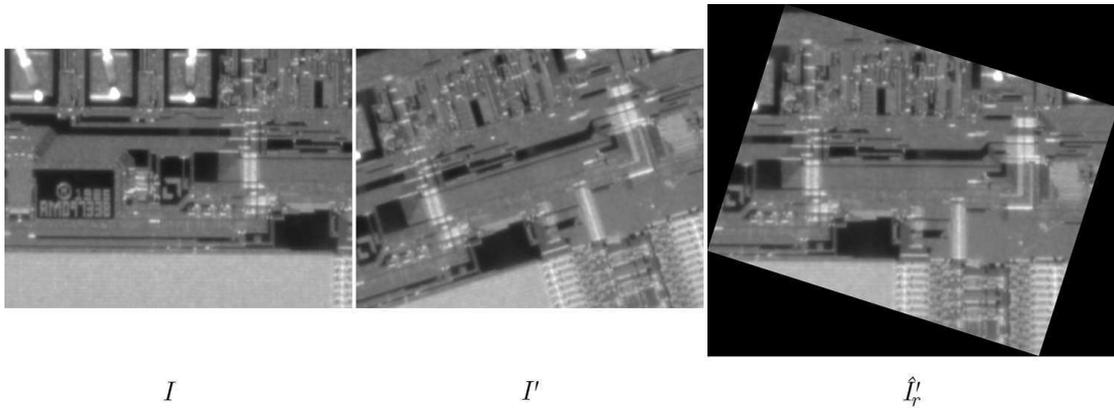


FIG. 4.18 – Le couple d’images I et I' original, avec l’image \hat{I}'_r le résultat du recalage de I' en fonction I avec une interpolation de type bicubique.

par *forward mapping* car elle a besoin de plusieurs étapes pour être mise en place. Les étapes sont les suivantes :

- la première étape consiste à déterminer le contour de l’image résultat ; en prenant les quatre coins de l’image I' et par la transformation géométrique G_r nous calculons exactement la position des quatre nouveaux coins de l’image recalé \hat{I}'_r ,
- à partir de ce contour, l’image recalée est créée et initialisée arbitrairement à 0, c’est-à-dire une image vide,
- pour chaque point appartenant à l’intérieur du contour de l’image \hat{I}'_r , on utilise la transformation inverse G_r^{-1} pour aller chercher la position du pixel dans l’image d’origine I' ,
- par interpolation, on calcule l’intensité du pixel à transférer en fonction des pixels connexes,
- on recopie l’intensité du pixel interpolé dans l’image \hat{I}'_r .

Au final nous obtenons une image modifiée géométriquement par la transformation choisie sans trou et avec anti-aliasing par le biais de l’interpolation qui évite le phénomène de marche d’escalier dans l’image.

L’étape du recalage a pour but de transformer toutes les images par rapport à une image de référence. La Figure 4.18 présente le résultat du recalage d’une image originale utilisée. Ce recalage va permettre dans la suite d’assembler toutes les images entre elles pour construire la mosaïque : c’est l’étape de fusion (*blending* en anglais)

4.6 Fusion

La fusion (*blending*) est la principale tâche du procédé de rendu de la mosaïque. L’objectif est d’obtenir une mosaïque sans « couture » c’est-à-dire sans transitions visibles entre les imagerie qui la composent. Le rendu calcule les valeurs des pixels de la mosaïque à partir d’une fonction des valeurs des pixels des imagerie d’entrée. Toute la

difficulté réside dans la détermination de cette fonction. Il existe beaucoup de fonctions de fusion, appelées masques de fusion puisqu'elles sont définies au-dessus d'une région bidimensionnelle, ou encore canaux ou *layer* alpha dans le domaine de l'image numérique car ce sont des calques, exactement comme les trois couleurs qui composent une image (rouge, vert, bleu), sauf qu'elles contiennent les coefficients de fondu de l'image. Il existe différentes fonctions dans la littérature, par exemple D. Capel (2001) en présente plusieurs :

- filtre moyen, le pixel rendu est la moyenne des valeurs des pixels des images d'entrée,
- filtre médian, le rendu du pixel correspond à la valeur médiane des pixels d'entrée,
- filtre de mode, la valeur du pixel rendu est déterminée par la moyenne des modes de l'histogramme des imagerie, il s'agit en fait d'une méthode robuste de calcul du filtre moyen,
- proche du centre de l'image, la valeur du pixel des imagerie le plus proche du centre de la mosaïque est choisie,
- filtre en fonction du pas, une fonction de pondération est associée à chaque image d'entrée, son maximum est défini au centre de l'image et à partir de celui-ci, pas par pas, la pondération décroît jusqu'au bord de l'image vers 0,
- filtre flexible, comme le filtre en fonction du pas, une fonction de pondération est associée à chaque image d'entrée, par contre les différentes pondérations tiennent compte du recouvrement des autres images.

La dernière méthode donne de meilleur résultat, car elle tient compte de la disposition de toutes les images de la mosaïque pour déterminer les meilleures pondérations à affecter à chaque pixel. Habituellement les pondérations sont définies empiriquement (D. Capel 2001, W. Zhao 2006), mais certains papiers proposent d'utiliser des méthodes d'optimisation pour les définir (D. Milgram 1975, D. Milgram 1977, S. Peleg 1981, A. Levin et al. 2004, N. Gracias et al. 2006). La fonction de pondération est directement appliquée sur les images d'entrée, c'est-à-dire que le pixel rendu est la somme pondérée des pixels d'entrée (D. Capel 2001, W. Zhao 2006). Ces fonctions peuvent être aussi utilisées dans différentes bandes de fréquence, à la fin elles sont combinées pour donner la mosaïque finale. Cette solution est une approche de type multi-résolution. On retrouve plusieurs méthodes de type multi-résolution : des méthodes de Laplacien pyramidale (E. Adelson et al. 1984) ou encore des méthodes utilisant la décomposition en ondelettes (C.-T. Hsu et J.-L. Wu 1996).

Nous proposons une nouvelle méthode de fusion de type filtre flexible. Actuellement la fonction de pondération est définie sur toute la surface des images d'entrée. Nous considérons que seule la région de chevauchement entre les images d'entrée devraient bénéficier de coefficients de pondération. La partie de l'image sans pondération serait copiée directement dans la mosaïque. Cette méthode augmenterait la rapidité du rendu de la mosaïque. Nous proposons alors la fusion adaptative, qui permet de réaliser un *blending* avec seulement un rendu par pondération sur les zones de recouvrement des images. Pour chaque image d'entrée utilisée dans le procédé du *blending*, un calque de pondération est défini par image. Évidemment, les mêmes transformations géométriques entre les images seront appliquées entre les calques associés aux images. Nous cherchons

à calculer le calque de pondération $C_p^{\prime r}$ de l'image \widehat{I}_r , qui va permettre de fusionner les images I et \widehat{I}_r . La première étape correspond à créer un calque C_p' de pondération arbitrairement initialisé sur l'image sans transformation (non recalé) \widehat{I}' . Nous utilisons la méthode du filtre en fonction du pas. Les coefficients de pondération du calque C_p' sont définis par le produit et la symétrisation d'une fonction de type rampe horizontale et verticale :

$$C_p' = [f_h(x, y) + s_{x>\gamma}(f_h)] \times [f_v(x, y) + s_{y>\gamma}(f_v)] \quad (4.27)$$

avec f_h et f_v les fonctions rampes respectivement définies sur l'axe horizontal et vertical et $s_{x>\gamma}(f_h)$ et $s_{y>\gamma}(f_v)$ respectivement la symétrie de $f_v(x, y)$ et de $f_h(x, y)$ selon les axes de symétrie définis par les lignes $x = \gamma$ et $y = \gamma$ (Figure 4.19). Par exemple la fonction de type rampe horizontale est définie de la façon suivante :

$$f_h(x, y) = \begin{cases} 0, & 0 < x < \alpha \quad \forall y \\ \frac{1}{\beta-\alpha}x + \frac{\alpha}{\alpha-\beta}, & \alpha < x < \beta \quad \forall y \\ 1. & \beta < x < \gamma \quad \forall y \end{cases} \quad (4.28)$$

Les paramètres α , β et γ sont définis par l'utilisateur, ils définissent le début, la fin et la largeur du fondu en jouant sur les coefficients de pondération du calque. Ils peuvent être ainsi ajustés en fonction des textures des images d'entrée. Une illustration du profil 1D de la fonction symétrique horizontale de type rampe est représentée à la Figure 4.19.

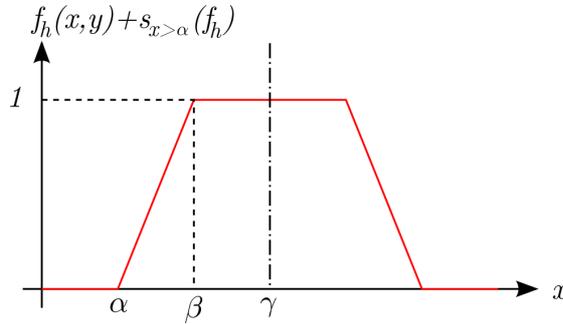


FIG. 4.19 – Le profil de la fonction $f_h(x, y) + s_{x>\gamma}(f_h)$ selon l'axe horizontal.

La fonction symétrique de type rampe est une fonction 2D, elle représente donc une image contenant les coefficients de pondération. Le calque résultat des pondérations est le produit entre les deux fonctions 2D horizontale et verticale. La Figure 4.20 représente le résultat obtenu du calque C_p' , il est présenté sous sa forme image (2D) mais aussi sous forme de niveau (3D) par rapport au pondération, pour mettre en valeur les variations des coefficients dans le calque.

La particularité de ce calque est d'avoir une évolution des pondérations quadratiques seulement sur les quatre coins du masque, de la forme suivante :

$$C_p'(x, y) = \begin{cases} \frac{(x-\alpha)(y-\alpha)}{(\beta-\alpha)^2} & \alpha < x < \beta \\ & \alpha < y < \beta \end{cases} \quad (4.29)$$

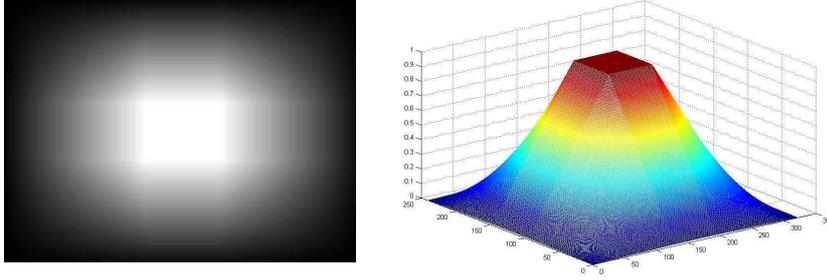


FIG. 4.20 – Représentation 2D et 3D du claqué C'_p pour $\alpha = 0$, $\beta = 0,8$ et $\gamma = 1$.

Maintenant nous allons utiliser ce calque de pondération pour calculer le calque C'_p qui n'est qu'autre que le masque de pondération C'_p recalé, de telle sorte que le calque qui contient les coefficients pour la fusion C'_p corresponde bien à la géométrie de l'image \hat{I}_r . D'un point de vue algorithmique nous avons besoin de connaître les pixels susceptibles d'être calculés par rendu. Par exemple dans l'image recalée, des zones sans information en noir ne sont pas utilisées dans le procédé, nous allons donc lui associer un calque de transfert C'_t . Ce calque contient une information binaire, 0 pour les parties de l'image non utilisées et 1 pour les pixels appartenant à l'image utile. Ce calque permet d'extraire rapidement (pour les pixels = 1) les pixels de l'image d'origine recalée \hat{I}_r . A défaut le calque de transfert C_t de l'image I contient des 1 dans toute l'image car toute sa surface est utile. Notre algorithme de fusion adaptative définit des pondérations de fondus seulement pour les zones de recouvrement. Pour définir cette zone nous allons utiliser l'intersection des deux calques de transfert C_t et C'_t . Ce calque de recouvrement noté C_o pour calque de recouvrement (*overlapping*) est défini par l'intersection suivante :

$$C_o(x_o, y_o) = C_t(x_t, y_t) \cap C'_t(x'_t, y'_t) \quad (4.30)$$

Avec \cap qui définit la partie commune des calques. Le calque de recouvrement C_o , traduit que seule cette zone aura besoin d'un fondu entre les images d'entrée. Nous l'utilisons donc pour délimiter les zones de pondération dans le calque de coefficient C'_p . Les parties du calque qui n'ont pas besoin de fondu prendront la valeur du calque de transfert, pour être copiées directement dans l'image résultat (mosaïque). Le calque de fusion final C_b est estimé en fonction du calque de pondération recalé C'_p , du calque de recouvrement entre les deux images à assembler C_o et enfin du calque de transfert recalé C'_t . Pour obtenir ce calque de fusion il faut faire l'union entre l'intersection du calque de pondération C'_p et le calque de recouvrement C_o avec l'intersection du calque de transfert C'_t et le calque complémentaire (à 1) du recouvrement C_o . Ce qui définit le calque de fusion de la façon suivante :

$$C_b(x_b, y_b) = [C'_p(x'_p, y'_p) \cap C_o(x_o, y_o)] \cup [C'_t(x'_t, y'_t) \cap C_o^c(x_o, y_o)] \quad (4.31)$$

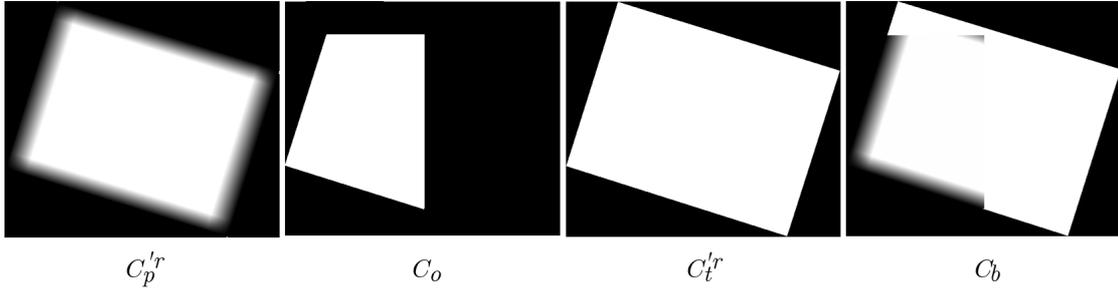


FIG. 4.21 – Représentation des différents calques de la méthode de fusion adaptative : $C_p^{r'}$ le calque des coefficients de pondération du fondu, C_o le calque de recouvrement entre les deux images d'entrée, $C_t^{r'}$ le calque de transfert de l'image recalée et C_b le calque de fusion finale.

avec l'opérateur C_o^{\complement} qui est le complément du calque C_o et \cup qui définit la somme des calques. La Figure 4.21 représente sous forme d'images la représentation de chaque calque utilisé pour assembler et fusionner nos deux images d'exemples.

Le calque de fusion final contient tous les coefficients permettant la fusion entre les deux images d'entrée. La procédure d'assemblage se fait en plusieurs étapes. La première consiste à copier la première image de référence I dans un espace de travail neutre, cette espace contiendra la mosaïque finale. Pour avoir une neutralité dans la couleur de fond de l'espace de travail, nous choisissons une couleur souvent utilisée dans le retouche photo ou l'incrustation d'imagette (*sprite*), la couleur magenta. La deuxième étape consiste à calculer le rendu de l'image recalée \hat{I}_r avec le calque de fusion. Cette nouvelle image I_{rendu} est exprimée comme suit :

$$I_{rendu}(x_{rendu}, y_{rendu}) = I(x, y)C_b^{\complement}(x_b, y_b) \cup \hat{I}_r(\hat{x}'_r, \hat{y}'_r)C_b(x_b, y_b) \quad (4.32)$$

Elle est composée du produit des coefficients de pondération C_b et de l'image recalée. Par contre si on désire une transition très douce entre les images d'entrée, il faut obligatoirement faire l'union avec les coefficients de pondération complémentaire avec l'image de référence qui se trouve déjà dans l'espace de travail. La dernière étape consiste tout simplement à copier l'image rendu I_{rendu} dans l'image résultat par écrasement, c'est-à-dire pour deux pixels existants déjà dans la mosaïque c'est l'intensité du pixel de I_{rendu} qui est propriétaire car il intègre déjà l'information du fondu avec l'image se trouvant déjà dans l'espace de travail.

Nous appliquons notre méthode de fusion adaptative à nos deux images exemples. Le résultat est présenté à la Figure 4.22. Ce résultat montre qu'il est impossible de dissocier la frontière des deux images fusionnées. Il existe une continuité parfaite entre les bords des images d'entrées. Nous illustrons les performances de cette méthode simple et rapide de *blending* en comparaison avec deux autres méthodes tout aussi simple à la Figure 4.23. Ce comparatif est réalisé entre deux images quelconques qui représentent le dos d'un livre. Il existe une petite erreur de recalage voulue pour mettre en valeur les bords des deux images. La méthode par recopie directe ne permet pas de supprimer

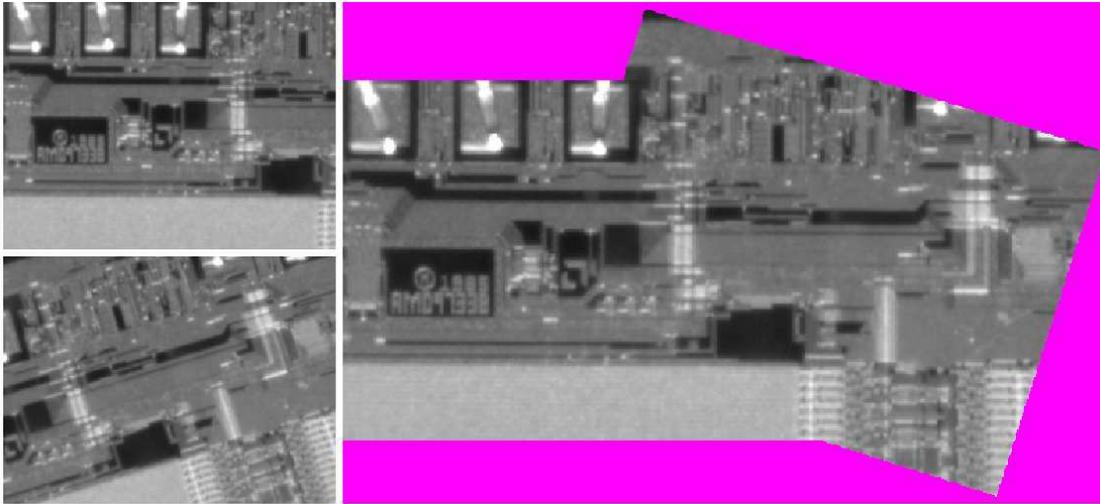


FIG. 4.22 – A gauche le couple d’images originales, à droite l’image de l’espace de travail représentant la mosaïque obtenue par recalage et fusion.

les effets de « contours ». Par contre en utilisant un filtre de type moyenneur, les deux écarts entre les images très visibles s’estompent légèrement sans pour autant le supprimer complètement. Seule, la méthode adaptative permet de passer d’une image à l’autre avec une transition très souple. Ce qui permet dans ce cas d’effacer l’erreur de recalage entre les deux images, car la méthode agit comme un pont, rejoignant les deux rives des images en les fusionnant progressivement en fonction de leur bordure respective.



FIG. 4.23 – Comparaison entre des approches simples de *blending* de deux images quelconques : à gauche, la méthode de recopie directe ; au centre, la méthode par moyennage ; et à droite la méthode adaptative.

Pour résumer et simplifier l’implémentation de la méthode adaptative, nous proposons le détail de l’algorithme pour un rendu d’une vue complète à partir de n images d’entrées, il s’écrit :

```

 $I_0$  : l'image de référence
 $I_n$  : les images à fusionner
 $E_t$  : l'image de l'environnement de travail vide

copier  $E_t \leftarrow I_0$ 
calculer les calques  $C'_p$  et  $C'_t$ 
Pour chaque image  $I_n$  faire
     $G_r = \text{colinéation}(I_{n-1}, I_n)$ 
     $\widehat{I}_r = \text{transformation}(G_r, I_n)$ 
     $C'^r_p = \text{transformation}(G_r, C'_p)$ 
     $C'^r_t = \text{transformation}(G_r, C'_t)$ 
     $C_o = (C'_t + C'^r_t)/2$ 
     $C_b \leftarrow C'^r_p$ 
    Pour chaque pixel de  $C'^r_t = 1$  faire
        Si (pixel de  $C_o \neq 1$ ) Alors
             $C_b \leftarrow 1$ 
        Fin Si
    Fin Pour
    Pour chaque pixel de  $C'^r_t = 1$  faire
         $E_t = E_t \times (1 - C_b) + \widehat{I}_r \times C_b$ 
    Fin Pour
Fin Pour
 $E_t$  : l'image de l'environnement de travail avec la mosaïque

```

Algorithme 3: La fusion adaptative

Quelques mots de précision sur l'algorithme et plus particulièrement sur le calcul du masque de recouvrement C_o . Dans l'algorithme il est écrit $C_o = (C'_t + C'^r_t)/2$, précisons la nature de C'_t et de C'^r_t . Ce sont deux masques de transfert qui contiennent les valeurs 0 et 1 pour la présence de pixels appartenant à l'image. Pour calculer la zone de recouvrement on additionne ces deux calques, ainsi la zone de recouvrement va prendre la valeur de $1 + 1 = 2$, la zone des images reste à 1 et l'image non utile à 0. Par la suite le fait de diviser cette addition par deux permet de normaliser le calque pour qu'il soit compris entre 0 et 1. Donc, maintenant la zone de recouvrement est de valeur 1, les zones images utiles 0,5 et le reste à 0. Ainsi pour effacer les coefficients de pondération sur la zone de non recouvrement il suffit de tester si chaque pixel de $C_o \neq 1$, comme fait dans l'algorithme. Le résultat obtenu de notre algorithme est en fait l'espace de travail contenant la mosaïque formée des images d'entrées, comme présenté dans la Figure 4.22. L'extraction de la mosaïque est relativement simple, en effet elle est dans un environnement contenant une couleur neutre (magenta), qui permet facilement de ne récupérer que l'image nous intéressant. Par contre l'image de la mosaïque n'est pas une image parfaitement rectangulaire, sa forme finale dépend de la position des images

recalées. Pour obtenir une image plus esthétique, de forme rectangulaire, qui par la suite nous permettra d'exploiter la mosaïque, une nouvelle étape est requise, c'est le détourage automatique.

4.7 Détourage automatique

L'étape finale du détourage automatique consiste à déterminer automatiquement la zone maximale de la partie de l'image qui peut être découpée de forme rectangulaire sur la mosaïque dans l'espace de travail. Toute la difficulté réside dans la détermination des quatre coins du rectangle de détourage, car le détourage en lui-même est très simple ce n'est juste qu'une recopie d'image. Nous présentons une méthode simple et rapide pour obtenir les quatre coins du rectangle qui maximisent la surface utile de la mosaïque, de telle sorte à en avoir le moins possible à couper. Notre méthode consiste dans un premier temps à créer un masque binaire à partir de l'image utile dans l'espace de travail. Pour cela dans une nouvelle image nous plaçons la valeur 0 quand la couleur du pixel de l'espace de travail est neutre (magenta) sinon la valeur 1. On obtient un masque binaire qui représente la zone utile de la mosaïque. A partir de ce masque nous calculons, via les moments de l'image, le centre de gravité de l'image afin de déterminer le centre de la mosaïque (\bar{x}, \bar{y}) . Ce centre de gravité va nous permettre de définir 4 quadrants de travail (q_1, q_2, q_3, q_4) dans la mosaïque, un par coin du rectangle à déterminer (Figure 4.24-a).

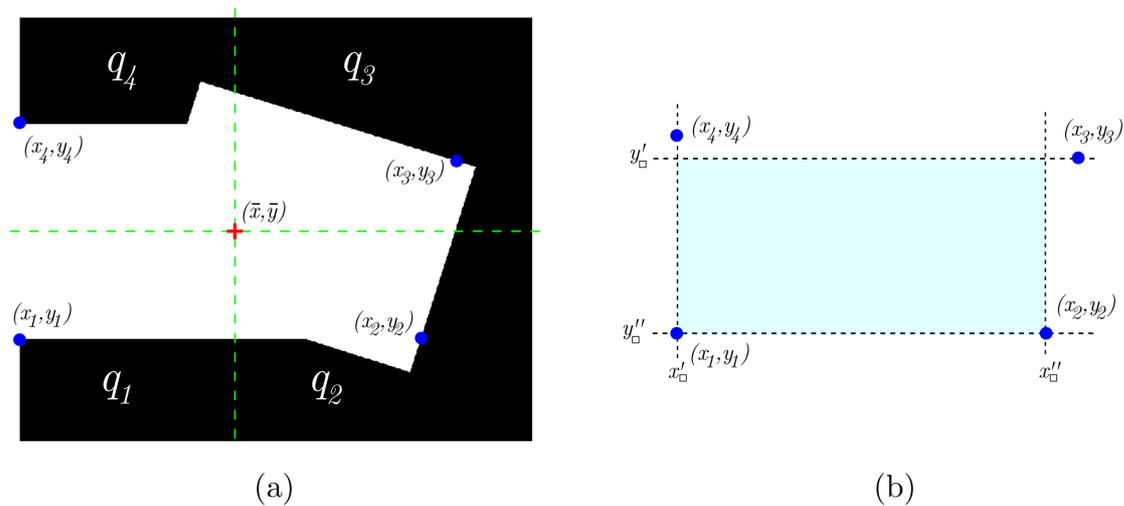


FIG. 4.24 – Méthode du détourage automatique : (a) Les quatre quadrants définis par le centre de gravité avec les points de chaque quadrant qui maximisent la surface utile, (b) définition du rectangle maximal à partir des quatre points des quadrants.

Dans chaque quadrant nous allons calculer la surface utile pour chaque coordonnée des points de bordure de la mosaïque. Pour cela, on calcule l'aire du rectangle définie par le centre de la mosaïque et le point de bordure de la mosaïque. Cela revient à multiplier la coordonnée y du point de bordure avec la taille du segment qu'il est possible de concevoir

à l'intérieur de la mosaïque associé à ce point. Ce segment est calculé en ajoutant la valeur de chaque pixel de la ligne (0 ou 1) jusqu'au bord de la mosaïque. Pour une coordonnée y donnée l'aire du rectangle inclus dans l'image utile du quadrant q_2 , formé par la ligne horizontale correspondante s'écrit :

$$A_{q_2}(y) = y \sum_{x=\bar{x}}^{|I^\phi(x,y)|_x} I^\phi(x,y) \quad \forall y \in [\bar{y}; |I^\phi(x,y)|_y] \quad (4.33)$$

avec I^ϕ l'image binaire selon que le pixel de l'image I est utile ou non. La mesure de l'aire va permettre de déterminer les coordonnées du rectangle ayant la plus grande surface dans chaque quadrant. Pour cela on analyse l'évolution de chaque rectangle en fonction des coordonnées des points de bordure de la mosaïque. Par la suite il suffit de chercher la coordonnée y du point de bordure dans chaque quadrant pour laquelle la surface du rectangle est maximale. Un exemple de courbe de mesure de l'aire pour le quadrant q_3 est représenté à la Figure 4.25

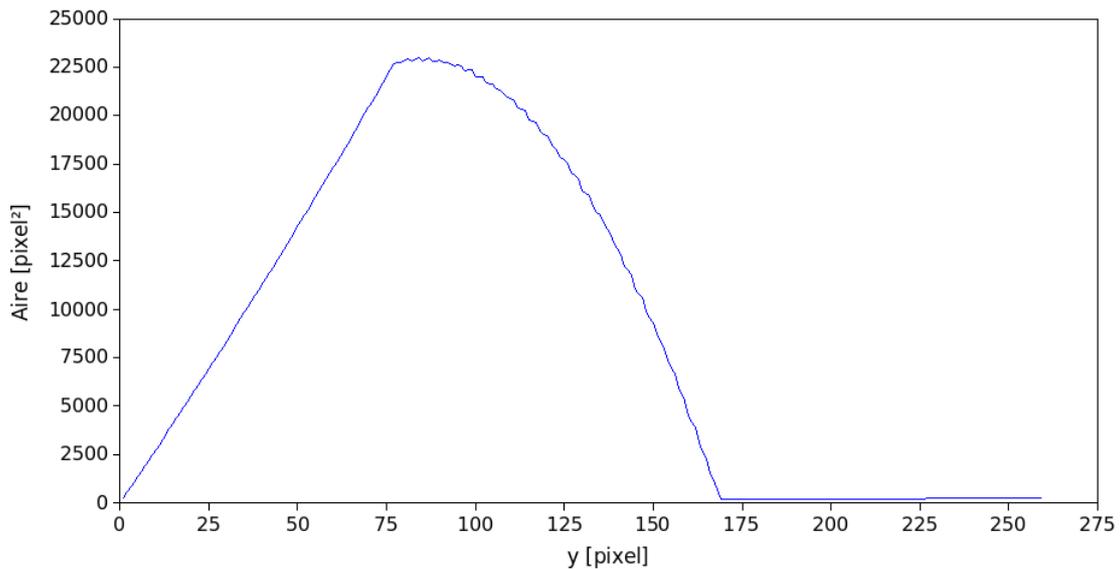


FIG. 4.25 – Evolution de l'aire du rectangle de la surface utile dans le quadrant q_3 en fonction de la coordonnée y de la ligne de balayage dans le masque.

Nous obtenons 4 points pour lesquels chacun d'eux maximise une surface rectangulaire dans leur quadrant respectif. Étant donné que la forme finale de la mosaïque doit être un rectangle, il faut uniformiser les différentes abscisses et coordonnées de chacun de ces points afin d'extraire les quatre coins d'un rectangle. Pour uniformiser nous allons chercher le rectangle qui peut être englobé par le quadrilatère défini par les points des quatre quadrants. Par une simple règle mathématique nous pouvons déterminer l'abscisse et la coordonnée minimale et maximale du rectangle de détourage :

$$\begin{aligned}
 x'_{\square} &= \max(x_1, x_4) \\
 x''_{\square} &= \min(x_2, x_3) \\
 y'_{\square} &= \max(y_3, y_4) \\
 y''_{\square} &= \min(y_1, y_2)
 \end{aligned}
 \tag{4.34}$$

avec $x'_{\square}, x''_{\square}, y'_{\square}$ et y''_{\square} , les abscisses et ordonnées de la boîte englobante de détournage, et (x_i, y_i) le point du sous rectangle du quadrant q_i . A partir des abscisses et des coordonnées de la boîte de détournage nous découpons cette zone dans l'espace de travail et la copions directement dans une nouvelle image. Pour des raisons de clarté nous proposons le résumé de notre méthode de détournage automatique sous forme d'un algorithme présenté ci-joint :

E_t : l'image de l'espace de travail avec la mosaïque

$masque = \text{binairisation}(E_t)$

$(\bar{x}, \bar{y}) = \text{barycentre}(masque)$

$q_i = \text{quadrants}(\bar{x}, \bar{y})$

Pour chaque quadrants q_i **faire**

$hauteur = 0$

Pour chaque ordonnée y depuis \bar{y} **faire**

$hauteur = hauteur + 1$

$largeur = 0$

Pour chaque abscisse x depuis \bar{x} **faire**

$largeur = largeur + masque(x, y)$

Fin Pour

$A(y) = hauteur \times largeur$

$p(y) \leftarrow x$

Fin Pour

$y_i = \max(A(y))$

$x_i = p(y_i)$

Fin Pour

$x'_{\square} = \max(x_1, x_4)$

$x''_{\square} = \min(x_2, x_3)$

$y'_{\square} = \max(y_3, y_4)$

$y''_{\square} = \min(y_1, y_2)$

$I_{mos} \leftarrow \text{détournage}(E_t, [x'_{\square}, x''_{\square}, y'_{\square}, y''_{\square}])$

I_{mos} : Image mosaïque rectangulaire

Algorithme 4: Détournage automatique

Nous appliquons cette méthode de détournage à l'espace de travail contenant la mosaïque de nos deux images de départ. Le résultat de la mosaïque rectangulaire est montré à la

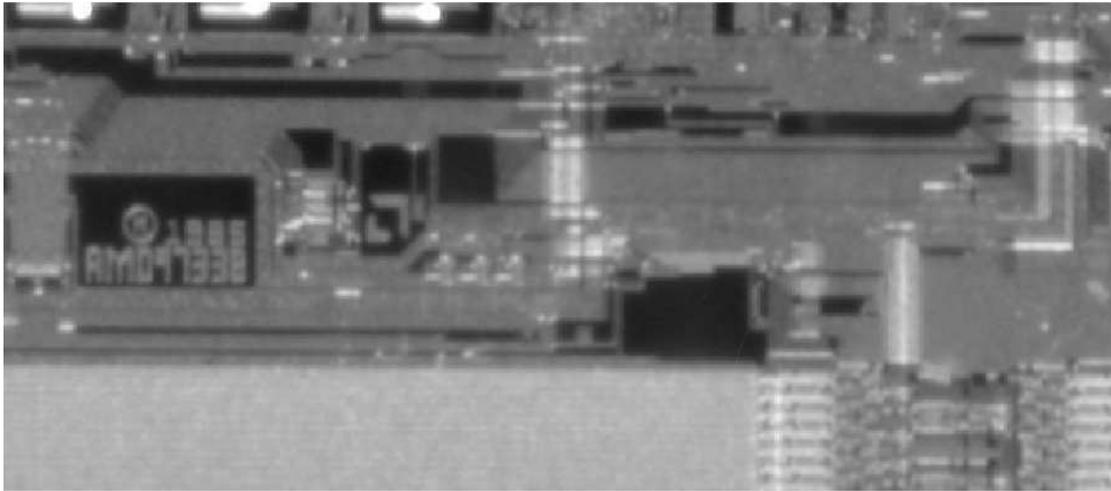


FIG. 4.26 – Mosaïque des deux images de l'Eprom après détournage.

Figure 4.26. Un des inconvénients du détournage par notre méthode utilisant les quatre quadrants, est que l'optimisation de la surface utile est estimée de façon locale par quadrant. Par conséquent, la surface globale n'est pas optimale. Cependant cette approche est suffisante pour traiter les images mosaïques dans le contexte de la micromanipulation, plus particulièrement du fait que les images d'entrée ne subissent que très peu de rotation. Les mosaïques sont alors pratiquement rectangulaires et donc assez faciles à détourner.

4.8 Application en micromanipulation

Nous venons de présenter la méthode de construction de mosaïques par indices invariants, maintenant nous allons pouvoir l'appliquer dans un contexte de micromanipulation. L'objectif de la construction est d'augmenter le champ de vision d'un microscope photonique, permettant ainsi à l'utilisateur de surveiller l'espace de travail en entier tout en ayant la haute résolution des images locales. Le contexte de l'application est relativement simple. Nous allons balayer la scène de micromanipulation en faisant l'acquisition de plusieurs images par le biais d'un microscope photonique. Toutes les images locales collectées seront assemblées et fusionnées en une seule image globale, une image mosaïque.

4.8.1 Dispositif

Le dispositif expérimental est composé du même système de tables de déplacement (x, y, z) utilisé dans l'exemple applicatif de la construction de mosaïques par mouvement calibré. Les tables de positionnement ont une résolution de $2,5 \mu m$ et une course de $55 mm$ chacune. Une boucle d'asservissement bas niveau sur les moteurs pas à pas des

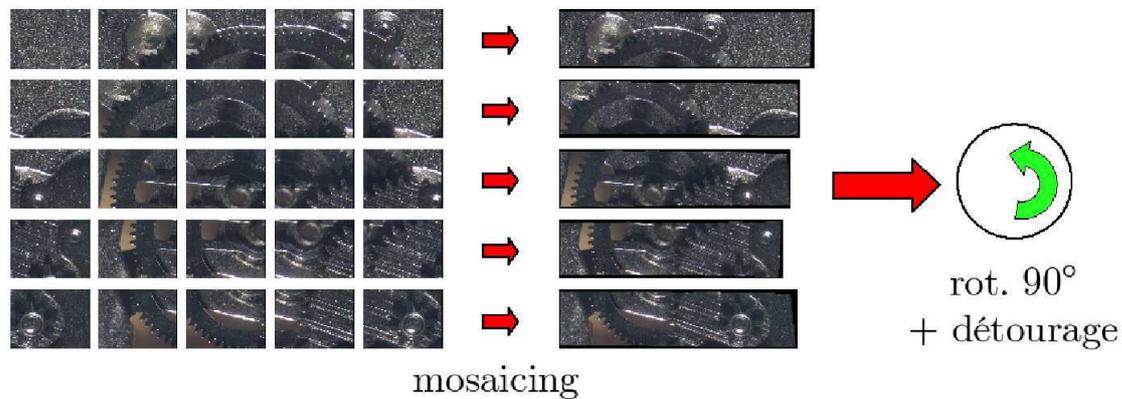


FIG. 4.27 – Première étape d’assemblage du mosaïque matricielle, la constitution des bandes mosaïques.

tables est réalisée par le contrôleur SMS60. Nous nous contentons de spécifier les coordonnées articulaires via une liaison RS232 entre le PC et le contrôleur SMS60. Nous allons utiliser un mini-microscope photonique TIMM400 de dimension de longueur 155 mm et de diamètre 22 mm . Il est équipé d’un objectif $6x$ et d’un capteur CCD monochrome de $1/4''$ pour une résolution d’image de 540×600 pixels. L’éclairage est composé de panneaux lumineux à base de mini-tubes fluorescents au néon. Pour récupérer les images nous utilisons des convertisseurs sous forme de boîtiers extérieurs. Ils convertissent les signaux analogiques PAL en signaux numériques IEEE1394 (*firewire*) directement connectés sur les ports *firewire* du PC. Toute l’architecture du système expérimental est composée de briques standards de banc d’essai optique. Ces briques permettent de garantir une grande flexibilité sur les différentes architectures expérimentales, elles permettent de changer la configuration du système de vision sans racheter du matériel ou de développer des pièces dédiées. Au niveau logiciel, toute la partie expérimentale est développée sur Matlab dans notre *toolbox* de vision pour la micromanipulation. Elle contient toutes les fonctions de la construction de mosaïques mais également du pilotage du contrôleur des tables SMS60.

4.8.2 Mosaïque matricielle

La partie explicative sur les différentes méthodes de construction de mosaïques a toujours été développée pour seulement deux images. Dans cette partie nous allons étendre le procédé de deux images à un ensemble d’images provenant d’un balayage de forme matricielle. Le balayage assure un déplacement de type zig-zag manuel constant entre toutes les images avec une zone de recouvrement définie arbitrairement sans calibrage. Par conséquent l’ensemble des images constituent l’équivalent d’une matrice d’images de la scène. La mosaïque ne sera plus un simple panoramique, c’est-à-dire une bande d’images balayées de gauche à droite, mais plusieurs bandes d’images balayées également de haut en bas. La méthode présentée précédemment entre une paire d’images est suffisant pour construire la mosaïque de toutes ces images. La première étape consiste à assembler

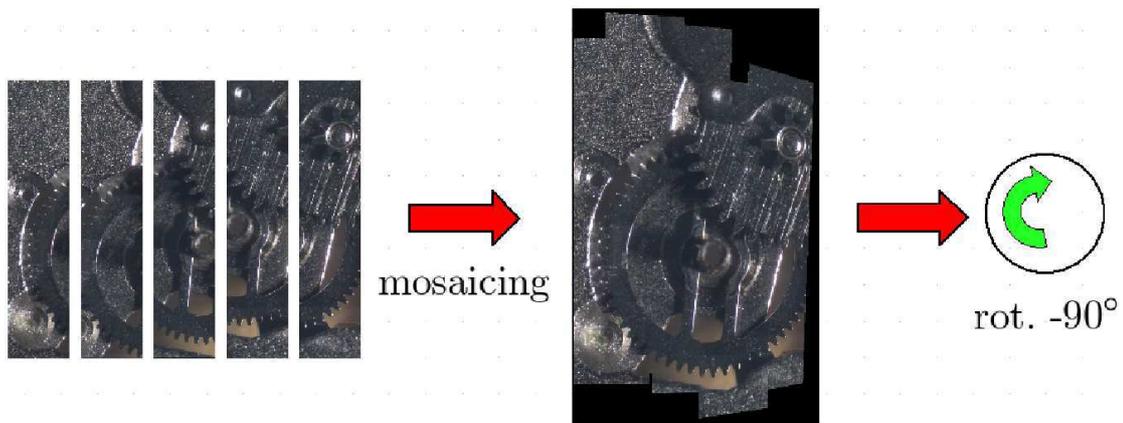


FIG. 4.28 – Assemblage des bandes mosaïque.

deux à deux les images de chaque ligne de la matrice d'image pour obtenir des bandes de mosaïques. Par exemple dans la Figure 4.27, nous avons une matrice d'image 5×5 images. Après la première étape nous obtenons 5 sous-mosaïques. Pour les préparer pour le reste de la procédure d'assemblage, elle sont toutes détournées pour obtenir des bandes parfaitement rectangulaires.

L'algorithme que nous avons prototypé sous Matlab, pour des raisons de facilité, ne permet que d'assembler des images de gauche à droite. Or dans ce cas pour assembler les différentes bandes l'assemblage doit être de haut en bas. L'astuce consiste à faire une rotation de 90° de toutes les bandes, pour que l'assemblage soit dans le bon sens, de gauche à droite. La suite correspond tout simplement à l'assemblage des bandes deux à deux pour obtenir la mosaïque finale (Figure 4.28). Étant donné que les bandes ont subies une rotation, pour revenir à l'orientation originale nous effectuons de nouveau une rotation mais en sens inverse de -90° .

Le tout dernier procédé sur la mosaïque finale est un détournage afin d'avoir une image parfaitement rectangulaire (Figure 4.29). Lorsqu'on regarde l'image avant et après détournage, on remarque que l'on perd beaucoup d'informations. Cela vient du fait que les bandes ne sont pas toutes de la même longueur. Comme nous utilisons un balayage non

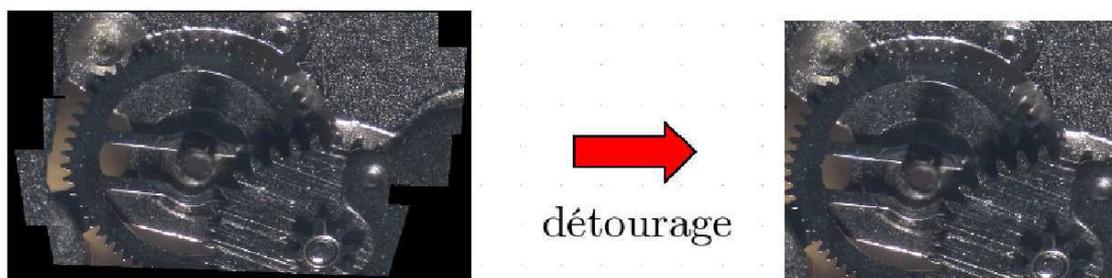


FIG. 4.29 – Détournage final sur la mosaïque complète de l'espace de travail.

calibré et réalisé de façon arbitraire, la zone balayée n'est pas optimale. Dans la suite pour toutes les mosaïques réalisées nous utiliserons dorénavant un balayage de type zig-zag automatique, toujours non calibré. Nous définissons un déplacement métrique arbitrairement en faisant attention qu'il y ait suffisamment de recouvrement entre les images pour qu'elles puissent être assemblées. Le fait de passer à un balayage automatique va permettre de minimiser le gaspillage photométrique c'est-à-dire l'acquisition d'image qui au final sera supprimée par détourage.

Pour illustrer le procédé de réalisation de mosaïque de grand espace de travail, nous allons utiliser la puce électronique de l'EPRM, dans les sections précédentes nous avons utilisé seulement deux images, cette fois-ci nous prenons un ensemble d'images matricielles. La Figure 4.30 présente l'image de la mosaïque obtenue. Le facteur d'échelle est $2,5187 \mu\text{m}/\text{pixel}$ pour des images d'entrée de 640×480 pixels. Cette mosaïque est composée de 5×3 images locales. Entre elles il existe un recouvrement d'environ 50%, ce qui nous donne une image globale finale de 1865×977 pixels soit un champ de vision de $11,28 \text{ mm}^2$.

4.8.3 Autofocus

Dès le moment où on utilise un microscope photonique on se heurt à une profondeur de champ très faible. Comme la scène observée n'est pas forcément plane les images apparaissent floues par moment. Effet on retrouve des irrégularités en fonction des objets et des pièces que le microscope photonique survole. Par exemple si on balaie l'espace de travail d'un bâti de montre pour observer toute la mécanique à l'intérieur, il existe des pignons à différents niveaux qui peuvent se retrouver en dehors de la profondeur de champ du microscope photonique. Le focus du mini-microscope photonique utilisé dans notre expérimentation se règle par une molette manuelle en bout du tube du microscope photonique. Par contre on peut également régler, pour de petites variations, le focus en modifiant la distance de l'objectif avec les objets observés. Notre solution consiste donc à rechercher le meilleur focus pendant le balayage en jouant sur la distance entre le mini-microscope photonique et l'espace de travail. Cela implique de mesurer le focus pendant l'acquisition. A chaque nouvelle position de balayage, le mini-microscope photonique effectue un mouvement de descente et de montée. Pour chaque profondeur on capture une image et on mesure le focus. Nous choisissons un algorithme de mesure de focus de type variance normalisée (F. Groen, IT. Young et G. Lighthart 1985, T. Yeo, SO Jayasooriah et R. Sinniah 1993), pour sa simplicité et parce qu'il donne d'excellents résultats. Pour obtenir un comparatif des différents algorithmes de mesure de focus utilisés dans la microscopie se référer à l'article de Y. Sun, S. Duthaler et B. Nelson (2005b). La variance normalisée s'écrit de la façon suivante :

$$\mathcal{F} = \frac{1}{|I(x,y)|_x \times |I(x,y)|_y \times \bar{I}} \sum_{x=1}^{|I(x,y)|_x} \sum_{y=1}^{|I(x,y)|_y} (I(x,y) - \bar{I})^2 \quad (4.35)$$

Comme nous avons mesuré les focus pour les différentes profondeurs de l'imageur, il est facile de sélectionner celle qui correspond à un focus maximum, c'est-à-dire à l'image

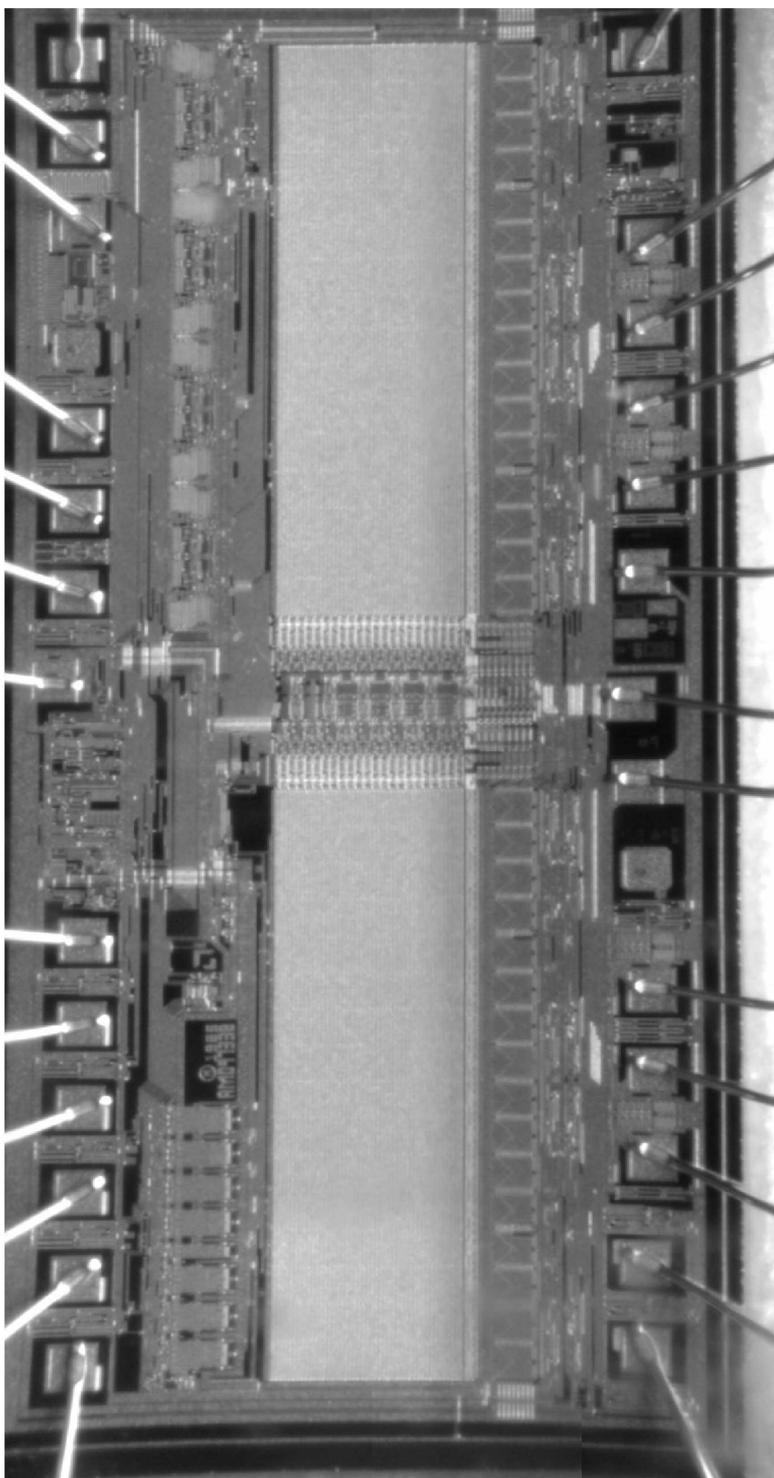


FIG. 4.30 – Mosaique d'une puce d'EPROM constituée de 5×3 images (facteur d'échelle $2,51 \mu\text{m}/\text{pixel}$, de dimension de 1865×977 pixels et d'un champ de vision de $4,6 \text{ mm} \times 2,4 \text{ mm}$).

la plus nette. A ce moment, la table de déplacement positionne le microscope photonique à la position nécessaire, puis nous prenons une acquisition de l'image qui sera utilisée pour composer la mosaïque. L'inconvénient de mettre en place un auto-focus pendant le balayage est qu'il faut plus de temps pour obtenir l'ensemble des images d'entrée. Pour chaque position on effectue un balayage supplémentaire sur Z . Cependant, nous sommes certains d'obtenir des images d'entrée nettes pour la construction de la mosaïque. Il ne faut pas oublier que si une des images est floue, il sera très difficile de trouver des points d'intérêt, et cela peut mettre en péril l'ensemble du processus automatique de construction.

4.8.4 Incrustation de vidéos

L'inconvénient de la construction de mosaïque c'est que l'image globale ainsi réalisée est statique. Pour des images esthétiques panoramiques de paysage cela ne pose aucun problème, mais quand le but est de surveiller et de contrôler des systèmes censés faire de la micromanipulation en temps réel il y a alors un problème. Une solution consisterait à avoir plusieurs imageurs qui observent la scène au même instant et à calculer une mosaïque en temps réel. Cette solution est irréalisable avec des systèmes de type microscope photonique, problèmes d'encombrement et de coût. Notre philosophie générale était d'avoir le maximum de vues avec le minimum d'imageurs, par conséquent nous sommes orientés vers une solution présentée par Masakatsu Kourogi et al. (1999), elle est très simple et consiste à surveiller les zones susceptibles d'être non stationnaires, c'est-à-dire des objets bougeant dans la scène, puis à en faire l'acquisition et l'incrustation en temps réel dans la mosaïque pré-établie de l'environnement. Le procédé consiste dans un premier temps à réaliser la mosaïque, le fond, de l'espace de travail de la micromanipulation, puis de déterminer les zones potentiellement dynamiques, comme le magasin d'objets ou la zone d'assemblage des microproduits. Et enfin d'observer ces zones via des microscopes photoniques et d'incruster l'image locale dans le fond de l'environnement. La difficulté de ce procédé est en fait lié à la localisation de l'image locale dans le référentiel de l'image globale.

4.8.4.1 Incrustation à partir d'imageurs statiques

Le premier cas de localisation de l'image locale dans l'image globale est l'utilisation de sources d'images passives où la position de la zone observée est statique. Par exemple le magasin en microproduits est toujours au même endroit dans l'espace de travail. Dans ce cas la source d'images est placée à la position désirée d'observation. L'initialisation du repérage de la vue locale dans la vue globale, est calculée en prenant une acquisition de l'image locale puis par corrélation on la recherche dans la mosaïque. Du fait que l'on recherche l'image entière par corrélation nous utilisons le III-ZNSSD pour diminuer le nombre de pixels à analyser. Une des contraintes d'utiliser une corrélation pour déterminer le point d'insertion de l'image locale dans l'image globale, c'est qu'elle doit être ressemblante. Cela veut dire qu'au moment de l'acquisition de la première initialisation, il faut vérifier que la scène observée soit identique à la scène pré-établie pendant la con-

struction. Si par exemple un objet est présent alors que dans le fond il ne s'y trouvait pas, l'occultation peut engendrer des problèmes au niveau de la corrélation et ne pas permettre de retrouver exactement la position de l'image locale dans la mosaïque. Après la recherche par corrélation, la position de l'image locale est mise en mémoire. Elle sera ré-utilisée à chaque acquisition pour incruster l'image locale dans l'image globale. Pour l'incrustation nous ré-utilisons notre méthode de fusion adaptative, permettant d'avoir un fondu entre les images temps réel du microscope photonique et la mosaïque.

Cette méthode n'est valide que si les objets observés restent en dessous de la position du microscope photonique. Si un préhenseur tel qu'une micropince, qui a une position dynamique, sort du champ de vision du microscope photonique, il disparaît fondu dans la mosaïque, car l'incrustation de l'image locale est statique. Cela nous amène à présenter le deuxième cas de localisation de l'image locale.

4.8.4.2 Incrustation à partir d'imageurs mobiles

Dans le cas où on désire incruster l'image d'un objet qui a une position dynamique dans l'environnement il faut bien évidemment mettre à jour le lieu d'incrustation dans la mosaïque. Ce cas de figure nécessite d'avoir un système de vision actif, c'est-à-dire que le microscope photonique doit avoir la capacité de se déplacer dans l'espace de travail. Par exemple si on souhaite visualiser la pince à tout moment dans l'image globale, il faut d'une part, que l'imageur suive la pince et que d'autre part cette vue mobile soit placée au bon endroit dans la mosaïque. Pour cela nous allons utiliser différents algorithmes de suivi de cibles pour que notre système soit capable de suivre l'objet à observer dans l'espace de travail. Nous présentons seulement trois méthodes existantes dans la littérature, pour leur simplicité de mise en œuvre.

Lucas-Kanade La méthode Lucas-Kanade présentée par B. Lucas et T. Kanade (1981) figure comme une des plus anciennes méthodes de flot optique (*optical flow*). Elle permet le suivi de cible dans un flux vidéo. À partir d'une zone prédéfinie dans l'image, la méthode Lucas-Kanade estime de façon itérative le déplacement de cette zone dans l'image suivante du flux vidéo. Soit l'image courante d'un flux vidéo I et son image suivante I' . La méthode Lucas-Kanade consiste à minimiser l'erreur de déplacement (u, v) d'une région entre deux images :

$$E_{rr}(u, v) = \sum_{x \in ROI} \sum_{y \in ROI} [I(x + u, y + v) - I']^2 \quad (4.36)$$

avec E_{rr} l'erreur de déplacement de la région ROI (*Region Of Interest*). Cette erreur peut être décomposée et approximée de la façon suivante,

$$E_{rr}(u, v) = \sum_{x \in ROI} \sum_{y \in ROI} [I_x u + I_y v - I_t]^2 \quad (4.37)$$

avec $I_x u + I_y v$ qui représente le gradient spatial et I_t' un gradient temporel :

$$I_x = \frac{\partial I(x,y)}{\partial x} \quad I_y = \frac{\partial I(x,y)}{\partial y} \quad I_t = I' - I \quad (4.38)$$

L'objectif est de minimiser l'erreur E_{rr} et d'en extraire le déplacement (u, v) . Si l'erreur tend vers 0 on peut factoriser et mettre sous forme matricielle le déplacement (u, v) comme suit :

$$\begin{bmatrix} \Sigma I_x^2 & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \Sigma I_t I_x \\ \Sigma I_t I_y \end{bmatrix} \quad (4.39)$$

Si on pose la première matrice de cette équation comme étant M et la dernière comme étant N , le déplacement (u, v) est égale à :

$$\begin{bmatrix} u \\ v \end{bmatrix} = M^{-1}N \quad (4.40)$$

Le schéma classique de Lucas-Kanade consiste à calculer le premier déplacement (u, v) entre la région des deux images, puis de modifier par transformation l'image I vers l'image I' en fonction des déplacements qui viennent d'être calculés. Dans les prochaines itérations on re-calcule le déplacement, qui doit être plus faible. D'itération en itération, de transformation en transformation, l'erreur de déplacement est accumulée. Après un nombre d'itérations défini par l'utilisateur (en général de 5 à 10 itérations) nous obtenons le déplacement inter-image de la région analysée. Le nombre d'itérations est choisi en fonction de la finesse du déplacement que l'on souhaite. Pour un algorithme rapide on met 1 à 2 itérations, mais le déplacement n'est alors pas précis. Une méthode rapide et récente présentée par A. Rav-Acha et S. Peleg (2006) permet d'utiliser un algorithme Lucas-Kanade sans itération. La méthode classique de Luca-Kanade itérative est spécialement optimisée pour des flux vidéo car il recherche de petits déplacements, typiquement le déplacement d'objets inter-image dans une vidéo. L'algorithme Lucas-Kanade suppose que l'objet dans l'image I s'est déplacé dans l'image I' de façon très proche. Cette contrainte implique que l'objet à suivre doit avoir une vitesse et une accélération de faible valeur, sinon son déplacement inter-image est trop important par rapport au rafraîchissement de l'imageur. Et dans ce cas la méthode Lucas-Kanade ne parvient pas à déterminer le déplacement de l'objet, il perd sa cible. Dans notre cas, la vitesse des micropréhenseurs est relativement lente et constante, ce qui est parfait pour appliquer cet algorithme. Nous l'appliquons à une vidéo d'une micropince développée au Laboratoire d'Automatique de Besançon qui se déplace de gauche à droite dans l'espace de travail. Nous sélectionnons deux zones centrées autour des deux points qui sont placés sur les deux extrémités des doigts de la micropince. A titre indicatif la taille de la pièce vue dans la scène fait $400 \mu m$ de large. La Figure 4.31 représente quelque images de la vidéo du suivi, nous avons utilisé un Lucas-Kanade avec une zone de 51×51 pixels avec une recherche de 2 itérations.

Indices invariants Les indices invariants peuvent être également utilisés pour le suivi d'objet. Par exemple, dans notre cas, on prend l'image d'un doigt de la pince et on y détecte les points les plus dignes d'intérêts en ne cherchant qu'un ensemble de points

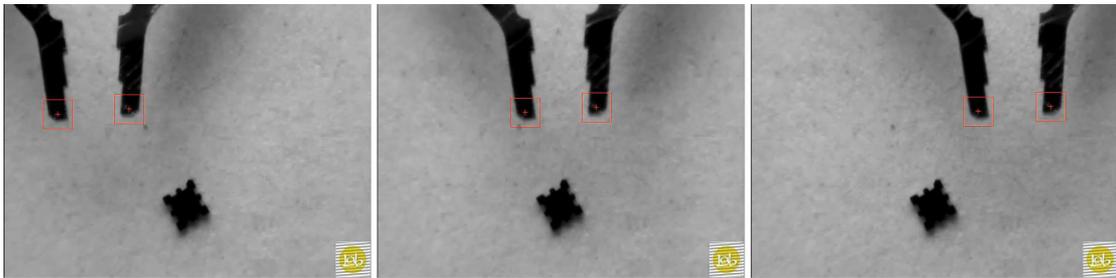


FIG. 4.31 – Suivi de la micropinçe par la méthode Lucas-Kanade (masque de 51×51 pixels pour 2 itérations).

restreints (une dizaine). Cette image et ces points seront placés en mémoire pendant tout le suivi de la pince dans un dictionnaire de signatures. A chaque image de la vidéo où la micropinçe se déplace, on estime plusieurs indices invariants dans l'image. Cet ensemble d'indices va définir une signature spécifique à l'objet observé dans l'image. Cette signature est comparée dans le dictionnaire des signatures préalablement établi. Comme il existe la signature du doigt de la pince dans le dictionnaire, plusieurs points sont reconnus et permettent d'identifier la signature de l'objet inconnu dans l'image comme étant celle du doigt de la pince. Cette méthode permet de suivre dans chaque image de la vidéo le déplacement de l'objet répertorié dans le dictionnaire. L'inconvénient dans cette méthode c'est qu'il faut faire la mise en correspondance des indices pour identifier la signature dans le dictionnaire, et donc le risque d'erreurs d'appariement. Pour limiter l'impact de ce risque sur le suivi il suffit d'introduire un seuil de reconnaissance c'est-à-dire de considérer qu'il faut un minimum de points reconnus pour être garanti de la validité de l'objet. Il existe aussi d'autres inconvénients liés à l'invariance des indices. En effet, si la micropinçe subit une rotation, les masques de corrélation définis autour des indices ne correspondent plus à l'image définie en mémoire. Il est donc impératif en fonction des cas de figure d'utiliser des indices avec des niveaux d'invariance plus ou moins élevés (invariance en rotation ou à l'échelle). On peut citer quelques méthodes très populaires d'indices invariants comme les SIFT (D. Lowe 1999), les SURF (H. Bay, T. Tuytelaars et L. Gool 2006) ou encore l'utilisation de moments invariants couplés avec un détecteur d'Harris (A. Sluzek, Md Islam et P. Annamalai 2006). Dans notre cas nous allons utiliser simplement une méthode classique avec un détecteur de Harris et une mise en correspondance par ZNSSD. Étant donné que dans notre cas les trajectoires de la micropinçe sont des pures translations, la non-invariance aux rotations ne posera pas de problème, qui plus est les pièces mécaniques dans la scène ont des fortes ressemblances et par conséquent trop d'invariance en particulier en rotation pourrait apporter des ambiguïtés. Sur la même vidéo présentée précédemment nous illustrons la méthode par indices invariants à la Figure 4.32.

Sous-fenêtrage La dernière méthode que nous présenterons pour le suivi de cibles est une méthode récente et assez simple. Elle est présentée par W. Ding et al. (2006), et

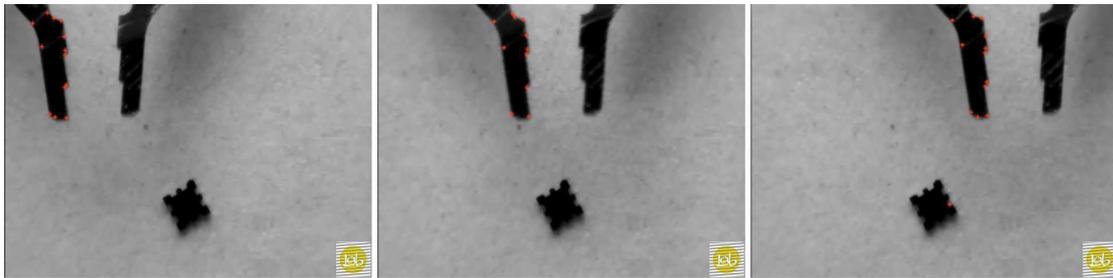


FIG. 4.32 – Suivi de la micropinçe en utilisant des indices invariants comparé à un modèle pré-enregistré (masque de corrélation de taille de 29×29 pixels).

consiste à recentrer une sous-fenêtre par rapport à l'objet à suivre. La première étape consiste à définir cette sous-fenêtre qui doit englober l'objet que l'on désire suivre. A chaque image du flux vidéo on utilise une méthode de segmentation de telle sorte à extraire l'objet à suivre du fond de la sous-fenêtre. A partir de cette segmentation on définit un indice permettant de mesurer la position de l'objet dans la sous-fenêtre à chaque image. Par exemple cela peut être simplement le centre de gravité de l'objet extrait. En utilisant la différence entre la position de l'indice à l'image précédente et celle à l'image courante, la sous-fenêtre se replace pour que l'indice soit centré par rapport à la nouvelle image. Cela permet d'image en image de suivre le déplacement de la sous-fenêtre centrée sur l'objet suivi. Si l'objet à suivre a des déplacements trop importants, il y a un risque qu'il sorte de la sous-fenêtre. Pour cela W. Ding et al. (2006) utilise un estimateur simple pour prédire la position de la sous-fenêtre dans l'image suivante, de telle sorte garantir que l'objet reste toujours à l'intérieur de la sous-fenêtre. On peut également utiliser des méthodes d'estimateur plus complexes comme un filtre de Kalman.

La grande difficulté de la méthode de sous-fenêtrage est de garantir l'extraction de l'objet à suivre par segmentation par rapport au fond. Nous utilisons cette méthode sur la vidéo de la micropinçe en train de se déplacer. La micropinçe a une luminance plus foncée que le fond de l'espace de travail, par conséquent la segmentation utilisée dans notre cas est un simple seuillage binaire. L'indice pour estimer le déplacement de l'objet est son centre de gravité calculé par la méthode des moments de l'image. Pour la prédiction de la sous-fenêtre, on se contente de prendre en compte la vitesse de l'objet. Ce qui nous permet d'avoir toujours la pinçe dans la zone de la sous-fenêtre. Des images de la séquence de suivi par sous-fenêtrage sont présentées à la Figure 4.33 avec une sous-fenêtre de dimensions 51×51 pixels.

Conclusion Parmi les trois méthodes présentées pour le suivi de cible, une est plus robuste que les autres, le suivi par Lucas-Kanade. Cependant la méthode la plus rapide est la méthode par sous-fenêtrage car elle utilise des traitements de l'image simples de type seuillage, moment d'image, etc. Par contre elle peut très bien devenir complexe lorsque la texture de fond de la sous-fenêtre ne permet pas une segmentation facile. La méthode par indices peut être très intéressante lorsqu'il existe plusieurs objets à détecter

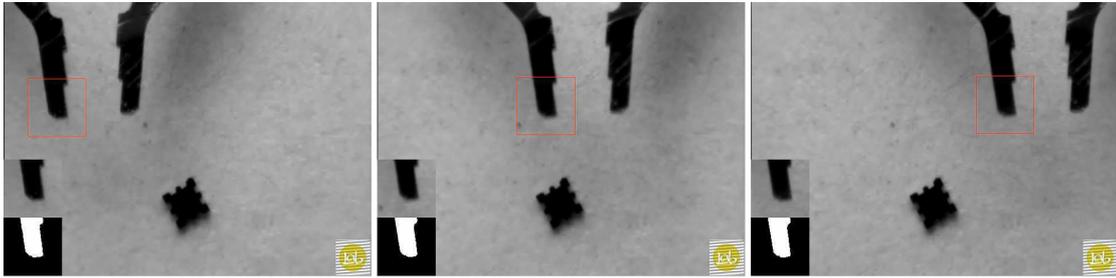


FIG. 4.33 – Suivi de la micropinche par sous-fenêtrage, en bas à gauche de chaque image, l'image de la sous-fenêtre ainsi que son seuillage sont rappelés (sous-fenêtre de taille de 51×51 pixels).

et à suivre dans la même scène. Par contre les indices invariants utilisés doivent être choisis correctement en fonction des objets, de leurs mouvements et de leurs textures. Pour la suite de nos applications nous utiliserons une méthode de type Lucas-Kanade car elle est plus appropriée à nos types de scène. Les motifs trop répétitifs (pièce mécanique) de la scène ne permettent pas d'utiliser un suivi par indices invariants de façon robuste. De même pour la segmentation et le suivi par sous-fenêtrage, la texture abondante et faiblement contrastée risque de perturber le bon déroulement du suivi.

4.8.5 Supervision de station de micromanipulation

Nous allons mettre en application la construction de la mosaïque et l'incrustation d'imageurs statiques dans le but de superviser une scène de micromanipulation. La scène représente la surface d'une piste d'un circuit électronique d'une montre. La mosaïque est réalisée à partir d'un ensemble d'images composées de 6×6 images, de dimensions 768×576 pixels chacune avec une zone de recouvrement d'environ 50%. Le facteur d'échelle de la mosaïque finale est de $3 \mu\text{m}/\text{pixel}$ soit une image finale de 1865×977 pixels pour un champ de vision de $5,8 \text{ mm} \times 4,7 \text{ mm} = 27,26 \text{ mm}^2$. La micromanipulation consiste en la dépose de billes de polystyrène de diamètre de $250 \mu\text{m}$ par une aiguille. La zone de dépose est parfaitement connue, par conséquent un mini-microscope photonique (TIMM400c) est placé au-dessus de cette zone. Par corrélation on détermine la position d'incrustation dans la mosaïque. On utilise une méthode de fusion pour incruster en temps réel les vues locales dans la vue globale. Une des images de la séquence vidéo de la micromanipulation est représentée à la Figure 4.34.

La deuxième application, consiste à avoir deux sources d'images dans la mosaïque, un imageur statique et un imageur mobile. La scène de micromanipulation représente le mécanisme d'une montre. On considère cette scène comme une station de la micromanipulation. Le but est la surveillance des tâches d'assemblage des pignons de montre. Nous avons une vue statique qui doit surveiller la micromanipulation d'un axe de pignon sur une zone connue, par une micropinche et une vue mobile qui doit suivre les déplacements d'une seconde micropinche dans l'espace de travail. La mosaïque est composée de 4×4 images de taille de 640×480 pixels avec un recouvrement d'environ 50%, soit une image

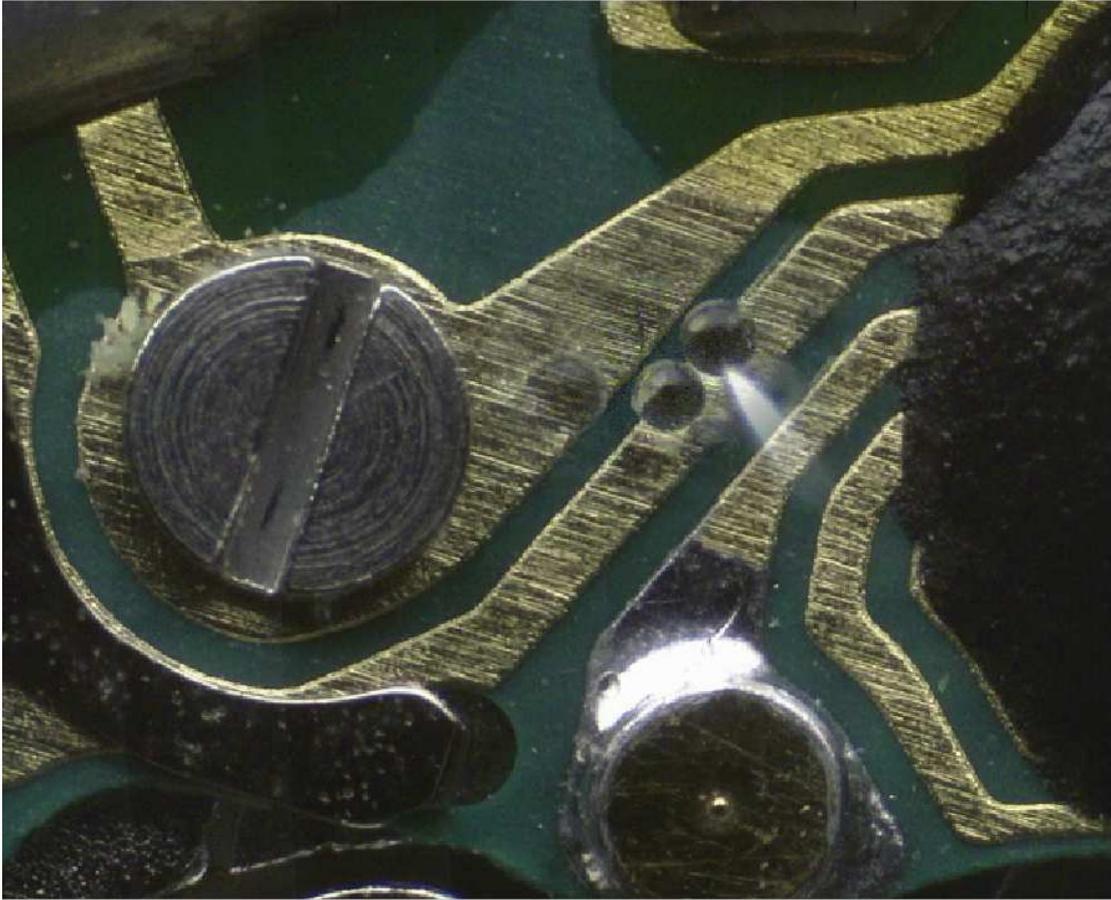


FIG. 4.34 – Image de la séquence vidéo de l’incrustation d’une vue statique dans une mosaïque.

finale de 1315×986 pixels de facteur d’échelle $8,495 \mu\text{m}/\text{pixel}$, ce qui nous donne un champ de vision de $11,17 \text{ mm} \times 8,37 \text{ mm} = 93,5 \text{ mm}^2$. L’initialisation du système de vision consiste à placer tous les mini-microscopes photoniques (TIMM400c) au dessus des zones initiales des micropinces. Une première image pour chaque mini-microscope photonique est prise afin de localiser la position initiale d’incrustation dans la mosaïque, par corrélation. Pour la vue statique cette position est placée en mémoire et ne changera pas pendant toute la durée de la supervision. Par contre, pour la vue mobile, deux points de référence sont placés par l’utilisateur sur les extrémités des doigts de la micropince. Un algorithme de suivi de type Lucas-Kanade est utilisé pour déterminer le déplacement de la micropince pour le recadrage du mini-microscope au-dessus de la pince mais également pour mettre à jour la position d’incrustation dans la mosaïque. Pour illustrer nous représentons deux images de la séquence vidéo à deux instants différents à la Figure 4.35.

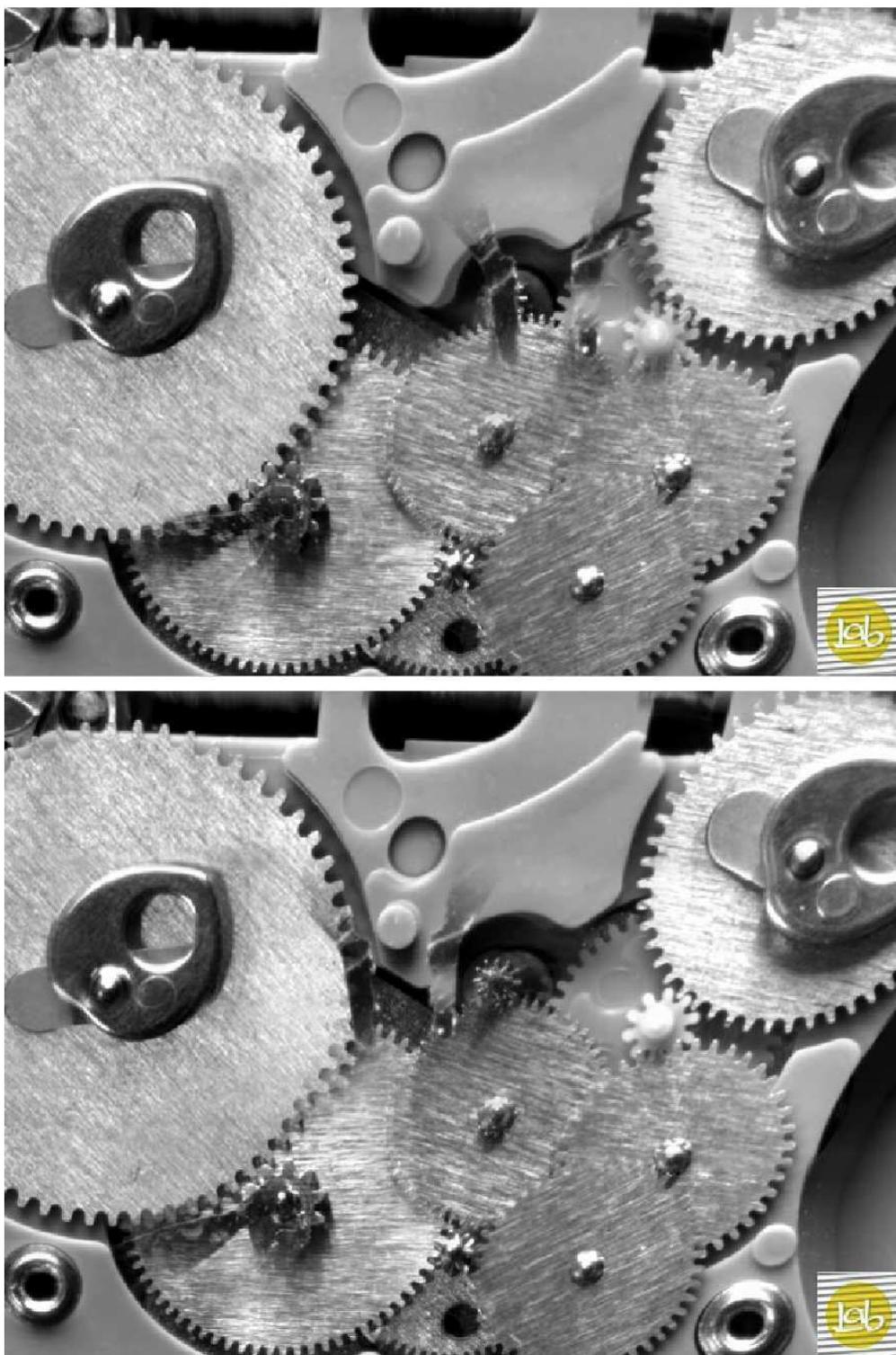


FIG. 4.35 – Deux images de la séquence vidéo de l'incrustation d'une vue statique et dynamique dans une mosaïque.

4.9 Conclusion

Le faible champ de vision d'un microscope photonique est une contrainte importante dans le contrôle et la supervision de tâche de micromanipulation et de micro-assemblage. En effet si le champ de vision est centré sur la micropince, cela est parfait pour la précision de la tâche de micromanipulation, mais par contre l'information globale est manquante. Où se trouve le magasin de microproduit par rapport à la pince? Où est la cible du micro-assemblage dans l'espace de travail? A l'inverse, si on utilise une source d'images avec un grand champ de vision on perd en résolution. La solution directe est de rajouter plusieurs sources d'images dans le système de vision afin d'avoir l'information globale et locale à la fois. Mais avoir des vues dispatchées et représentées dans plusieurs images différentes pose un problème de référentiel. Nous proposons d'utiliser des méthodes permettant d'avoir à la fois toutes les vues dans un même référentiel qui comporte l'information globale avec la résolution d'une vue locale. La vue globale est obtenue par une méthode de rendu d'images à partir d'images par construction de mosaïques (*mosaic-ing*). Nous avons proposé une méthode de construction à partir d'images à faible texture propre à la microscopie. Dans le domaine de la création de mosaïques par indices invariants nous avons utilisé nos nouveaux algorithmes présentés dans les chapitres précédents pour augmenter la robustesse de leur réalisation. Nous avons également présenté une nouvelle méthode de fusion adaptative et un nouveau algorithme de détournage automatique. Notre apport se trouve également dans le domaine des applications. Nous avons réalisé des mosaïques de scène par microscopie avec incrustation statique ou dynamique des zones de micro-assemblage en vue de la supervision de stations de micromanipulation. L'incrustation d'une vue locale dans la vue globale permet également en plus d'être dans un même référentiel, de limiter l'occultation des micropinces dans l'espace de travail. En effet seule l'extrémité, la partie utile de l'outil est incrustée dans la mosaïque, ce qui évite d'occulter l'espace de travail avec les parties non utiles des micropinces. Tout le travail de balayage et de création des mosaïques a été prototypé sous Matlab. Pour des raisons d'application et d'intégration, le système de supervision temps réel d'incrustation et le suivi des cibles ont été implémentés dans notre logiciel *MosaicingRTI* (*Mosaicing and Real-Time Inlaying*), développé en C++ avec la bibliothèque de vision libre et ouvert OpenCV. Le logiciel utilise la carte du fond de la scène de l'espace de travail pré-établie par Matlab et incruste en temps réel avec une vitesse d'environ 10 Hz la vue locale dans la mosaïque, une capture d'écran est montrée à la Figure 4.36.

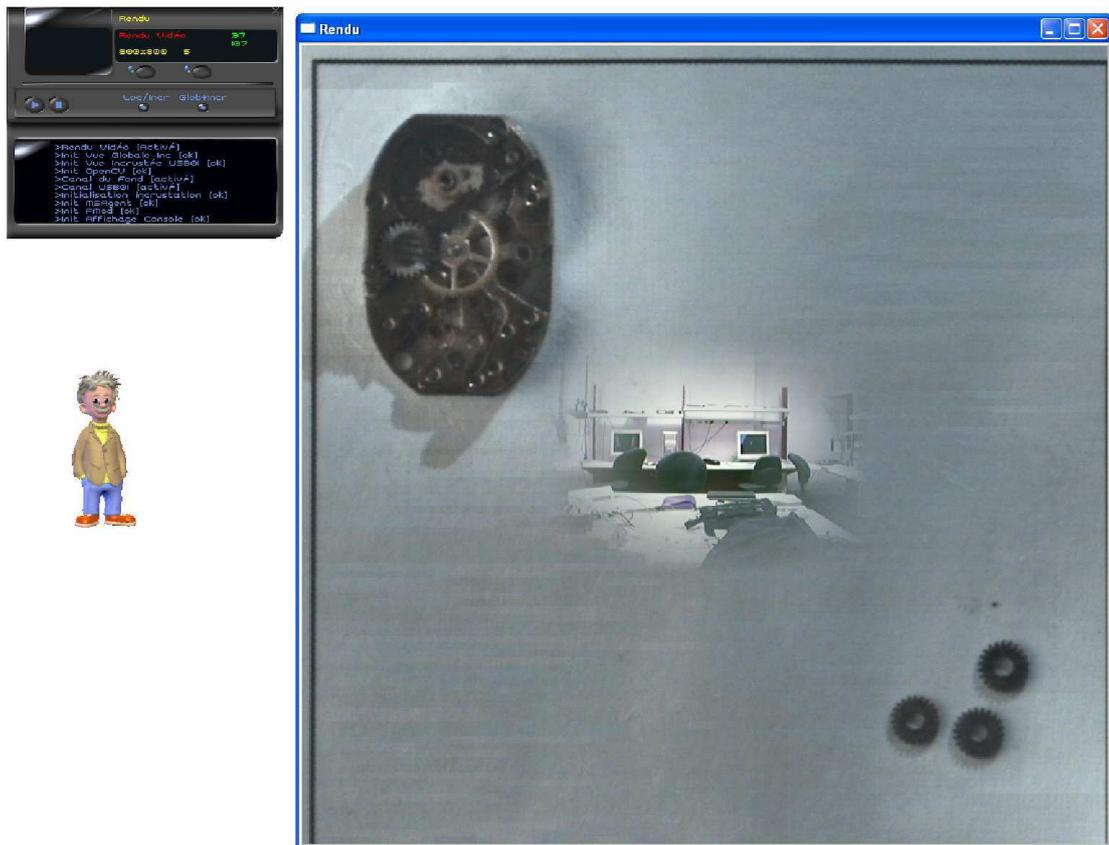


FIG. 4.36 – Le logiciel *MosaicingRTI* (*mosaicing* et incrustation temps réel).

Synthèse de Nouvelles Vues par Transfert Trifocal

La dernière technique de synthèse de nouvelles vues que nous présenterons dans ce chapitre est le transfert trifocal. Cette méthode va nous permettre de construire un maximum de vues virtuelles de l'environnement avec un minimum de microscopes photoniques. Nous montrerons comment elles peuvent remplacer une vue de côté réelle dans une application de micromanipulation.

5.1 Introduction

L'encombrement des microscopes photoniques est un sérieux problème dans une station de micromanipulation. Pourtant il est très difficile de se séparer de certaines vues utiles dans le contrôle des micropinces. Par exemple une vue de dessus permet le positionnement d'une micropince dans le plan (x, y) et la vue de côté permet le contrôle sur l'axe z . Une amélioration des microscopes photoniques classiques est apparue récemment avec la stéréoscopie, mais celle-ci est encore peu exploitée. Nous allons utiliser cette nouvelle caractéristique des microscopes photoniques pour reconstruire les autres vues utiles dans le pilotage d'une station de micromanipulation. Au final, un seul système de microscopie vidéo sera utilisé, les autres vues, comme la vue de côté, seront reconstruites à partir des deux images stéréoscopiques de la vue de dessus. Ce choix va nous permettre de libérer l'espace de travail par le biais d'une minimisation du nombre d'imageurs nécessaire. La méthode qui nous a plu, tant par son aspect original que prometteur est le transfert trifocal. Néanmoins, il existe une limitation à la position de certaines nouvelles vues, car il est impossible de reconstruire de l'information à laquelle on a pas accès. Le transfert trifocal est en quelque sorte boudé dans la communauté au profit des reconstructions classiques 3D. En effet la puissance de calcul de nos ordinateurs actuels fait que des méthodes de reconstruction très lourdes sont à la mode. Dans notre cas, où le contexte nous contraint à des calibrages difficiles et la nécessité, dans le futur, de disposer de systèmes temps-réel, le transfert trifocal nous a séduit.

Il permet simplement sans passer par la construction d'un modèle 3D, de reconstruire une image à partir d'autres images. Bien évidemment cette technique très peu connue a donc suscité très peu de littérature. Les quelques références en notre possession sont très évasives sur les détails de la méthode en elle-même. C'est pourquoi ce chapitre sera plus théorique que le précédent, car notre apport se trouve sur l'éclaircissement et la mise en application du transfert trifocal. Pour ne pas être noyé sous un ensemble de théories, nous allons illustrer ce chapitre par des exemples d'applications en présentant à chaque fois une partie du code pour mettre en pratique la méthode décrite dans chaque section. Les exemples utiliseront le même couple d'images tout au long de la partie théorique. Dans la section application à la micromanipulation, nous utiliserons d'autres images qui auront un rapport direct avec notre contexte. Pour l'instant nous utiliserons les images tests avec leurs mires de calibrage faible présentées à la Figure 5.1.

Durant ce chapitre nous présenterons également une nouvelle méthode automatique de positionnement de la vue de synthèse sans calibrage, ce qui nous permettra de l'exploiter dans notre contexte de micromanipulation. Nous présenterons aussi de nouveaux algorithmes de rendu, l'un permettant le rendu dans le cas d'une vue positionnée sans calibrage et l'autre évitant l'utilisation du *backward mapping* pour optimiser la qualité du texturage sur le rendu d'une image complète. La preuve de son utilisation dans un système de microscopie photonique sera validée par l'asservissement en position d'une micropince par une vue de côté, qui sera purement virtuelle et reconstruite par les images stéréoscopiques de la vue de dessus.

Toutes les expressions de ce chapitre seront écrites sous la notation Einstein, voir le glossaire des notations avec l'expression $a;p^i$.

5.2 Géométrie trifocale

La création de nouvelles vues par transfert trifocal est basée sur un système à trois vues. La géométrie qui les régit s'appelle la géométrie trifocale, géométrie qui est un peu plus fournie en littérature que le transfert en lui-même. Ce n'est en fait que l'extension de la géométrie épipolaire (2 vues) vers trois vues. Effectivement, un système de trois vues est constitué de trois sous systèmes de deux vues. Pour la compréhension de la méthode de reconstruction nous allons présenter la géométrie trifocale ainsi que ses propriétés.

5.2.1 Extension de la géométrie épipolaire

La géométrie trifocale est l'extension de la géométrie épipolaire à 3 vues. Considérons un système de 3 vues ψ , ψ' et ψ'' (Figure 5.2). Un point de l'espace projectif $P \in \mathcal{P}^3$ est projeté en un point $p = (x, y, 1)^T$ dans la vue ψ , $p' = (x', y', 1)^T$ dans la vue ψ' et $p'' = (x'', y'', 1)^T$ dans la vue ψ'' . Considérons les éléments suivants en prenant la vue ψ comme référence :

- A et B sont les matrices de colinéation correspondant respectivement à la transformation projective $\psi \rightarrow \psi'$ et $\psi \rightarrow \psi''$,
- v' et v'' les épipoles, c'est-à-dire les projections du centre optique O respectivement dans ψ' et ψ'' .

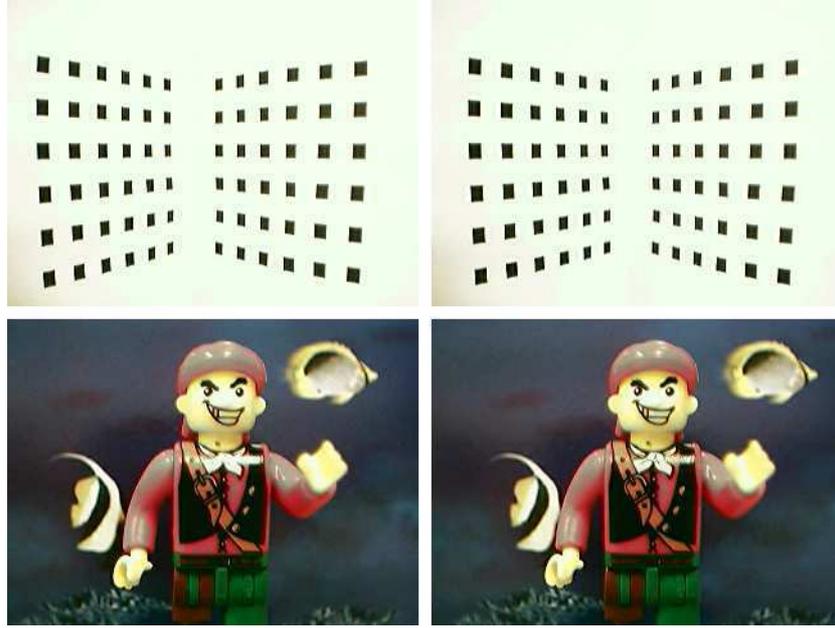


FIG. 5.1 – Couple d’images stéréoscopiques utilisé pour l’illustration du transfert trifocal : en haut, les mire pour le stéréo-calibrage faible, en bas, les images réelles.

D’après la géométrie épipolaire nous pouvons énoncer les relations suivantes en considérant les sous-systèmes stéréoscopiques (ψ, ψ') et (ψ, ψ'') :

$$\begin{aligned} p' &\cong Ap + \delta v' \\ p'' &\cong Bp + \delta v'' \end{aligned} \quad (5.1)$$

Avec δ le paralaxe relatif. Ces relations expriment le lien de projection du point p dans les vues ψ' et ψ'' en fonction de la transformation projective et du parallaxe projectif.

5.2.2 Équations trilinéaires

Si le point $P \in \mathcal{P}^3$ se déplace dans l’espace, les points p, p' et p'' vont se déplacer dans le plan image \mathcal{P}^2 . Les trois points sont intimement liés au même point P . Cette relation qui fait intervenir les trois points projetés dans les trois vues, est la trilinearité. Elle définit la contrainte entre les 3 vues. D’après A. Shashua (1994) chaque coordonnée de p, p' et p'' satisfait une paire d’équations trilinéaires de la forme suivante :

$$\begin{aligned} x'' (\alpha_1 x + \alpha_2 y + \alpha_3) + x'' x' (\alpha_4 x + \alpha_5 y + \alpha_6) + \\ x' (\alpha_7 x + \alpha_8 y + \alpha_9) + \alpha_{10} x + \alpha_{11} y + \alpha_{12} = 0 \end{aligned} \quad (5.2)$$

et,

$$\begin{aligned} y'' (\beta_1 x + \beta_2 y + \beta_3) + y'' x' (\beta_4 x + \beta_5 y + \beta_6) + \\ x' (\beta_7 x + \beta_8 y + \beta_9) + \beta_{10} x + \beta_{11} y + \beta_{12} = 0 \end{aligned} \quad (5.3)$$

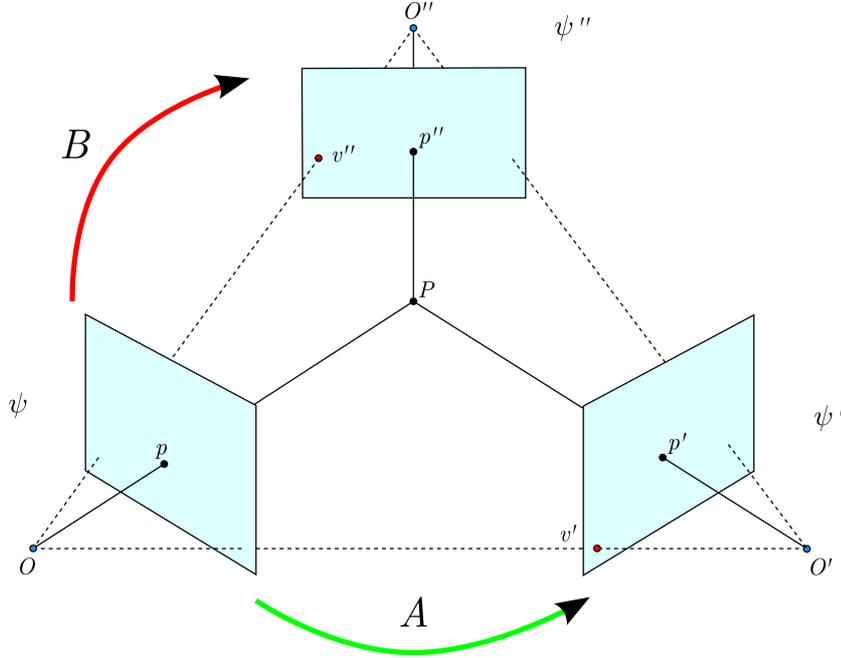


FIG. 5.2 – Géométrie trifocal.

où les coefficients α_i et β_i pour $i \in [1; 12]$ sont des constantes, qui ne dépendent pas du point P . Les coefficients des deux équations trilinéaires peuvent être estimés à partir des relations de la géométrie épipolaire entre chaque paire de vues. Pour cela on s'appuie sur une constante qui relie les vues entre elles : le parallaxe projectif permet de corriger la transformation projective pour déterminer la projection du point p dans chaque vue. Ce parallaxe dépend de la position de la vue, par contre, il est composé du parallaxe relatif δ , qui lui, exprime la distance normale au plan π , et ne dépend pas de la position et de l'orientation des vues. En effet la distance dans l'espace d'un point à un plan restera toujours constant dans l'absolu quel que soit l'angle de vue d'observation. Pour chaque équation de la géométrie épipolaire entre (ψ, ψ') et (ψ, ψ'') on peut donc isoler le parallaxe relatif car si on observe le même point c'est la même valeur (A. Shashua et N. Navad 1994). D'après les équations de la géométrie épipolaire (5.1) nous isolons le parallaxe relatif et nous obtenons cet ensemble d'égalités :

$$\delta = \frac{v'^1 - x'v'^3}{(x'a_i^3 - a_i^1) p^i} = \frac{v'^2 - y'v'^3}{(y'a_i^3 - a_i^2) p^i} = \frac{y'v'^1 - x'v'^2}{(x'a_i^2 - y'a_i^1) p^i}, \quad (5.4)$$

et,

$$\delta = \frac{v''^1 - x''v''^3}{(x''b_i^3 - b_i^1) p^i} = \frac{v''^2 - y''v''^3}{(y''b_i^3 - b_i^2) p^i} = \frac{y''v''^1 - x''v''^2}{(x''b_i^2 - y''b_i^1) p^i}, \quad (5.5)$$

où b_i^1, b_i^2, b_i^3 et a_i^1, a_i^2, a_i^3 sont les lignes des matrices A et B , et avec $v' = (v'^1, v'^2, v'^3)$ et $v'' = (v''^1, v''^2, v''^3)$. Pour trouver les équations trilinéaires il faut développer toutes les

égalités possibles, ce qui fait un total de 9 équations. Par exemple les premiers termes des équations (5.4) et (5.5), donnent :

$$\begin{aligned} x'' (v'^1 b_i^3 - v''^3 a_i^1) p^i - x'' x' (v'^3 b_i^3 - v''^3 a_i^3) p^i + \\ x' (v'^3 b_i^1 - v''^1 a_i^3) p^i - (v'^1 b_i^1 - v''^1 a_i^1) p^i = 0. \end{aligned} \quad (5.6)$$

Si on compare cette équation aux équations trinéaires (5.2) et (5.3) nous retrouvons la même forme. Après avoir développé les 9 équations nous obtenons seulement 4 équations linéairement indépendantes comprenant 27 coefficients distincts.

5.2.3 Tenseur trinéaire

Les 27 coefficients des 4 équations trinéaires peuvent être écrit sous la forme de notation tensorielle. Cette notation permettra par la suite d'exprimer les relations de façon plus compacte. Chaque facteur des 4 équations trinéaires peut être écrit de façon canonique, par les coefficients du tenseur trinéaire comme suit :

$$\mathcal{T}_i^{jk} = v'^j b_i^k - v''^k a_i^j \quad (5.7)$$

où \mathcal{T}_i^{jk} est le coefficient du tenseur trinéaire, a_i^j et b_i^k sont les éléments des matrices de colinéations A et B pour $i, j, k \in [1; 3]$ (j et k sont les indexes des lignes et i l'index des colonnes). Le tenseur trinéaire \mathcal{T} est une « super » matrice de dimension $3 \times 3 \times 3$ soit 27 coefficients, les indices sont décrits dans la Figure 5.3. Les 4 équations trinéaires peuvent s'écrire avec le tenseur de la façon suivante :

$$\begin{aligned} x'' \mathcal{T}_i^{13} p^i - x'' x' \mathcal{T}_i^{33} p^i + x' \mathcal{T}_i^{31} p^i - \mathcal{T}_i^{11} p^i &= 0 \\ y'' \mathcal{T}_i^{13} p^i - y'' x' \mathcal{T}_i^{33} p^i + x' \mathcal{T}_i^{32} p^i - \mathcal{T}_i^{12} p^i &= 0 \\ x'' \mathcal{T}_i^{23} p^i - x'' y' \mathcal{T}_i^{33} p^i + y' \mathcal{T}_i^{31} p^i - \mathcal{T}_i^{21} p^i &= 0 \\ y'' \mathcal{T}_i^{23} p^i - y'' y' \mathcal{T}_i^{33} p^i + y' \mathcal{T}_i^{32} p^i - \mathcal{T}_i^{22} p^i &= 0 \end{aligned} \quad (5.8)$$

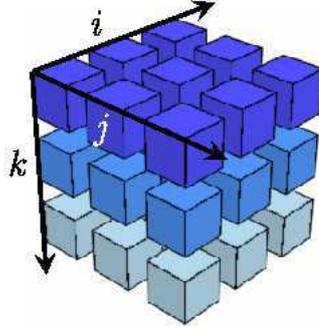
Les 4 équations que nous venons de présenter, proviennent d'une correspondance de type point/point/point, c'est-à-dire que la trinéarité, dans ce cas, est définie pour des projections de points. Les équations trinéaires expriment la relation de projection entre chaque vue et par rapport à la vue de référence ψ . Une des propriétés de la géométrie trifocale est d'avoir également des relations entre projections de lignes. On retrouve différentes correspondances, présentées dans l'ouvrage de référence de R. Hartley et A. Zisserman (2000), que voici :

correspondance point/point/point,

$$[p']_{\times} (p^i \mathcal{T}_i^{jk}) [p'']_{\times} = 0_{3 \times 3} \quad (5.9)$$

correspondance point/point/ligne,

$$[p']_{\times} (p^i \mathcal{T}_i^{jk}) l'' = 0 \quad (5.10)$$

FIG. 5.3 – Indices de notation du tenseur trilinéaire \mathcal{T} .

correspondance point/ligne/point,

$$l'^T (p^i \mathcal{T}_i^{jk}) [p'']_{\times} = 0^T \quad (5.11)$$

correspondance point/ligne/ligne,

$$l'^T (p^i \mathcal{T}_i^{jk}) l'' = 0 \quad (5.12)$$

correspondance ligne/ligne/ligne,

$$(l'^T [p^1 \mathcal{T}_1^{jk}, p^2 \mathcal{T}_2^{jk}, p^3 \mathcal{T}_3^{jk}] l'') [l]_{\times} = 0^T \quad (5.13)$$

Le tenseur trilinéaire présenté ici est exprimé d'après la vue de référence ψ . En réalité il existe 6 tenseurs différents en fonction des combinaisons des vues prises en référence, par exemple le tenseur utilisé ici est le tenseur $\mathcal{T}(\psi, \psi', \psi'')$ qui sera différent du tenseur $\mathcal{T}(\psi, \psi'', \psi')$. Il existe des règles pour passer d'un tenseur à un autre, nous le verrons par la suite. Il peut se décomposer en sous matrices en fonction de l'orientation de découpe dans le tenseur. Par exemple les sous matrices \mathbf{l}_i représentent les matrices du tenseur trilinéaire lorsque l'index i est fixé, on obtient :

$$\mathcal{T}_i^{jk} = \{\mathcal{T}_1^{jk}, \mathcal{T}_2^{jk}, \mathcal{T}_3^{jk}\} = \{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3\} \quad (5.14)$$

Nous avons également les sous matrices \mathbf{K}_i qui représentent les matrices du tenseur trilinéaire lorsque l'index k est fixé, on obtient :

$$\mathcal{T}_i^{jk} = \{\mathcal{T}_i^{j1}, \mathcal{T}_i^{j2}, \mathcal{T}_i^{j3}\} = \{\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3\} \quad (5.15)$$

Ces sous matrices ont des propriétés particulières notamment dans le changement de la vue de référence d'un tenseur trilinéaire. Pour plus d'informations sur les propriétés du tenseur trifocal se référer à l'ouvrage de R. Hartley et A. Zisserman (2000) et C. Ressel (2003). Finalement le tenseur trilinéaire ou trifocal représente un modèle qui régit les projections dans un système composé de trois vues. Ce système est modélisé par les éléments de la géométrie relative à une configuration de vue précise. Le tenseur trilinéaire

peut être estimé de différentes façons à partir de la correspondance des points appariés des trois vues : la méthode linéaire directe (C. Ressl 2003), la méthode itérative par minimisation des moindres carrés (R. Hartley 1994) ou encore l’algorithme des 6 points (P. Torr et A. Zisserman 1997)... Pour avoir un large spectre des méthodes on peut consulter (R. Hartley et A. Zisserman 2000).

L’idée consiste à utiliser le modèle pour calculer une vue de synthèse. Par exemple dans la correspondance point/point/point, si on connaît le tenseur trilinéaire \mathcal{T} , le point p et son correspondant p' , il est facile, en théorie, de calculer la position du point p'' dans la vue ψ'' .

5.3 Construction de nouvelles vues par transfert trifocal

L’idée de créer une vue virtuelle en utilisant le tenseur trilinéaire revient à S. Avidan et A. Shashua (1997). A cette époque ils utilisent un système de trois vues réelles. Le tenseur est estimé à partir de ces trois vues de référence. Ensuite il estime la vue virtuelle pour la comparer à la troisième vue réelle. Récemment on retrouve une application de ce procédé, qui consiste à connaître une configuration à trois vues pour en synthétiser l’une d’entre elle, présenté par T. Molinier, D. Fofi et P. Gorria (2006), le transfert trifocal est utilisé ici pour le texturage d’objets 3D par reprojection. De cette application, S. Avidan et A. Shashua (1997) remarquent très vite que ce procédé n’a aucun intérêt fondamental, en effet si on a la connaissance de la géométrie des trois vues, il suffit d’utiliser les lignes épipolaires qui se croisent dans la troisième vue, pour déterminer les points de la nouvelle vue. C’est la méthode de transfert épipolaire de S. Laveau et O. Faugeras (1994). C’est pourquoi S. Avidan et A. Shashua (1998) proposent cette fois-ci d’initialiser un tenseur trilinéaire seulement à partir de deux vues. Pourtant le tenseur trilinéaire est en théorie exprimé par un système à trois vues.

5.3.1 Tenseur trilinéaire à partir de deux vues

En fait il utilise une subtilité de configuration du système de référence des vues pour amorcer un tenseur à partir de deux vues. Cette subtilité consiste à supposer que deux des trois vues sont fusionnées. Dans notre système de vues composé de ψ , ψ' et ψ'' , la deuxième vue sera donc confondue avec la troisième, on a $\psi' = \psi''$ (Figure 5.4). Dans ce cas la matrice de colinéation A de la transformation $\psi \rightarrow \psi'$ et la matrice de colinéation B de la transformation $\psi \rightarrow \psi''$ sont identiques. De même les épipoles v' et v'' sont confondus donc de même valeur. Dans ce cas les coefficients du tenseur trilinéaire de ce système $\tilde{\mathcal{T}}_i^{jk}(\psi, \psi', \psi'' = \psi')$ deviennent :

$$\tilde{\mathcal{T}}_i^{jk} = v'^j a_i^k - v'^k a_i^j \quad (5.16)$$

Le tenseur $\tilde{\mathcal{T}}$ est le tenseur d’amorçage qui va permettre d’avoir un tenseur de base, pour ensuite retrouver un système de trois vues. Ce tenseur a une particularité, il est estimé seulement à partir de deux vues, par conséquent il inclut beaucoup moins d’information que le tenseur \mathcal{T} estimé à partir de trois vues. On peut le démontrer de la façon

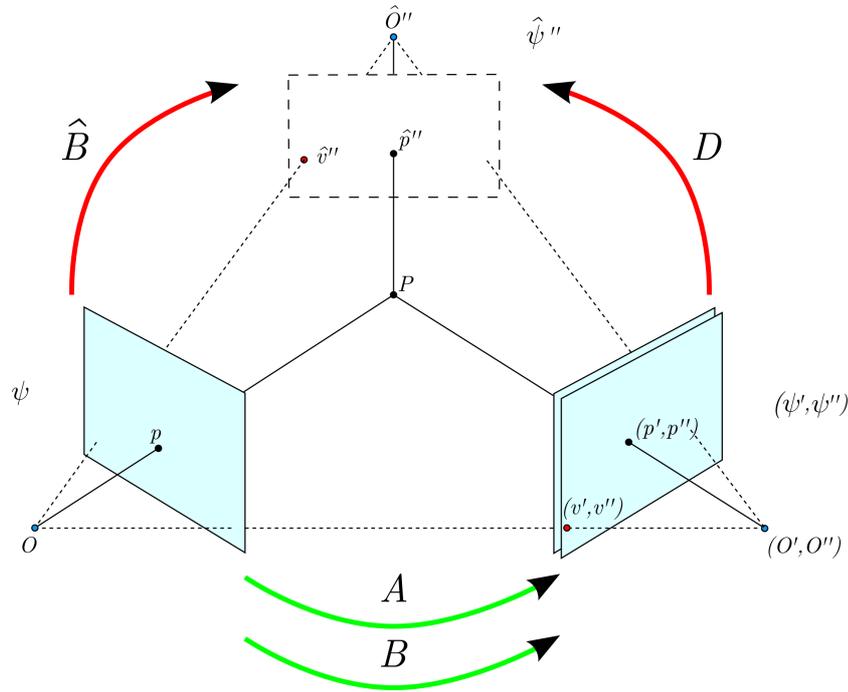


FIG. 5.4 – Principe de la synthèse de nouvelle vue à partir de deux vues.

suivante. La matrice fondamentale entre ψ et ψ' peut s'exprimer en fonction de l'épipôle v' et de la matrice de colinéation A comme cela :

$$F = [v']_{\times} A \quad (5.17)$$

Cela signifie que le tenseur \tilde{T} peut être estimé seulement à partir des éléments de la matrice F ,

$$\tilde{T}_i^{jk} = \epsilon^{ljk} F_{li} \quad (5.18)$$

avec ϵ^{ljk} l'opérateur Levi-Civita (produit croisé tensoriel). Cette équation permet de montrer que le tenseur \tilde{T} n'intègre qu'une matrice fondamentale. Ce qui signifie que nous sommes dans une configuration singulière car elle n'inclut que la géométrie relative de ψ et ψ' .

Exemple Nous utilisons les mires de calibration stéréoscopique, de la Figure 5.1, pour estimer la matrice fondamentale F puis en déduire l'épipôle v' ainsi que la matrice de colinéation A d'après les méthodes présentées dans le chapitre 3 sur le calibration stéréoscopique faible. Connaissant v' et A nous pouvons calculer le tenseur \tilde{T} avec l'équation (5.16), exemple du code Matlab/Octave :

```
0000 % Tenseur T d'indices (k,i,j)
```

```

0001 T=zeros(3,3,3);
0002 for j=1:3
0003     for k=1:3
0004         for i=1:3
0005             T(k,i,j)=vp(k)*A(j,i)-vp(j)*A(k,i);
0006         end
0007     end
0008 end

```

Remarque importante par rapport aux indices utilisés dans Matlab/Octave, les matrices sont décrites par (*ligne, colonne*), par conséquent le tenseur à des indices du type (k, i, j) .

5.3.2 Nouveau tenseur

Le but recherché est de transformer le tenseur d'amorçage \tilde{T} pour se placer dans une nouvelle configuration de vue $(\psi, \psi', \hat{\psi}'')$ avec $\hat{\psi}''$ la vue virtuelle. Pour cela supposons le système avec la configuration finale composée de la vue virtuelle (Figure 5.4). On pose \hat{T} le tenseur estimé de la configuration des vues ψ, ψ' et $\hat{\psi}''$. Les coefficients de ce tenseur trilinéaire s'écrivent comme cela :

$$\hat{T}_i^{jk} = v'^j \hat{b}_i^k - \hat{v}''^k a_i^j \quad (5.19)$$

avec \hat{b}_i^k les éléments de la matrice de colinéation \hat{B} de $\psi \rightarrow \hat{\psi}''$ et \hat{v}'' l'épipole de $\hat{\psi}''$ engendré par la projection du centre optique O . Nous allons exprimer ce tenseur en fonction du tenseur d'amorçage, c'est-à-dire en utilisant la configuration à deux vues. Soit D la matrice de colinéation qui transforme la vue ψ'' en la vue $\hat{\psi}''$. La matrice de transformation \hat{B} peut s'exprimer indirectement avec la double transformation en passant par la vue ψ'' de la façon suivante :

$$\hat{B} = D.B \quad (5.20)$$

Nous allons remplacer la matrice \hat{B} par les deux matrices de colinéation D et B dans le calcul des coefficients du tenseur \hat{T} . Dans les travaux de S. Avidan et A. Shashua (1998), il est proposé un résultat qui permet d'obtenir le tenseur \hat{T} exprimé à partir du tenseur des deux vues \tilde{T} . Il n'existe aucune littérature permettant d'avoir la démonstration de ce résultat, en plus elle nécessite une astuce mathématique ne permettant pas d'avoir le résultat simplement, c'est pourquoi nous allons le détailler. Notre méthode va consister à dérouler le développement de l'équation de la forme canonique du tenseur \hat{T} en remplaçant la matrice de colinéation \hat{B} par DB , jusqu'à être bloqué dans le développement. On fait de même mais cette fois-ci en remontant dans le développement du résultat de S. Avidan et A. Shashua (1998), jusqu'à être bloqué également. Lorsque les deux équations sont pratiquement de la même forme, il nous suffit de déterminer ce qui avait été injecté dans l'équation de départ pour résoudre entièrement la démonstration, que voici. Soit la matrice de colinéation \hat{B} composée de D et B :

$$\hat{b}_i^k = d_l^k b_i^l \quad (5.21)$$

On remplace cette expression dans le calcul des coefficients du tenseur \hat{T}_i^{jk} :

$$\hat{T}_i^{jk} = v'^j (d_l^k b_i^l) - \hat{v}''^k a_i^j \quad (5.22)$$

Nous allons injecter un terme nul dans cette expression :

$$\hat{T}_i^{jk} = v'^j d_l^k b_i^l - \hat{v}''^k a_i^j + \underbrace{v''^l d_l^k a_i^j - v''^l d_l^k a_i^j}_0 \quad (5.23)$$

Après développement et factorisation, un des facteurs est une forme canonique de notre tenseur d'amorçage \tilde{T} ($A = B$ et $v' = v''$) :

$$\hat{T}_i^{jk} = v'^j d_l^k b_i^l - v''^l d_l^k a_i^j + v''^l d_l^k a_i^j - \hat{v}''^k a_i^j \quad (5.24)$$

$$= d_l^k \underbrace{(v'^j b_i^l - v''^l a_i^j)}_{\tilde{T}_i^{jl}} + (v''^l d_l^k - \hat{v}''^k) a_i^j \quad (5.25)$$

Le facteur $(v''^l d_l^k - \hat{v}''^k)$ représente la différence de position entre l'épipole transformé $v''D$ et l'épipole \hat{v}'' , c'est-à-dire que la différence entre v'' et \hat{v}'' représente tout simplement un vecteur de translation entre ces deux épipoles dans l'espace projectif. Cela nous donne :

$$\hat{T}_i^{jk} = d_l^k \tilde{T}_i^{jl} + \underbrace{(v''^l d_l^k - \hat{v}''^k)}_{t^k} a_i^j \quad (5.26)$$

avec $t^k = v''^l d_l^k - \hat{v}''^k$ les éléments du vecteur de translation entre v'' et \hat{v}'' dans \mathcal{P}^3 . Au final nous obtenons le nouveau tenseur d'un système à trois vues dont une virtuelle, exprimée en fonction des éléments de deux vues :

$$\hat{T}_i^{jk} = d_l^k \tilde{T}_i^{jl} + t^k a_i^j \quad (5.27)$$

Le tenseur du système virtuel dépend du tenseur d'amorçage des deux vues réelles mais également de la matrice de colinéation A de $\psi \rightarrow \psi'$ et du vecteur de translation t^k . Le point important à ce stade est de savoir comment spécifier la position de la nouvelle vue dans l'espace.

5.3.3 Placement de la vue virtuelle

Les seuls paramètres utilisables pour la spécification de la vue virtuelle c'est la matrice de colinéation D et le vecteur de translation t . La méthode de placement que l'on attend, qui est la plus logique, est d'utiliser la matrice de colinéation D pour placer la vue dans l'espace projectif \mathcal{P}^3 , dans ce cas le vecteur de translation est égale à 0 et on obtient le nouveau tenseur suivant :

$$\hat{\mathcal{T}}_i^{jk} = d_i^k \tilde{\mathcal{T}}_i^{jl} \quad (5.28)$$

Or une difficulté survient, en effet il est impossible de spécifier, de façon intuitive, une vue à une position précise par une matrice de colinéation. La solution à ce problème consiste à spécifier la vue dans l'espace euclidien \mathcal{R}^3 . Le placement de la vue se trouve découpé en une matrice de rotation euclidienne R et en un vecteur de translation euclidien t , le nouveau tenseur est le suivant :

$$\hat{\mathcal{T}}_i^{jk} = r_i^k \tilde{\mathcal{T}}_i^{jl} + t^k a_i^j \quad (5.29)$$

Pour déterminer les paramètres de rotation et de translation dans l'espace euclidien il existe deux méthodes, une qui consiste à utiliser le calibrage stéréoscopique fort, et une autre le calibrage faible. Nous allons évoquer ces deux méthodes avec leurs avantages et leurs inconvénients. Par la suite nous présenterons notre méthode, par calibrage faible, qui permet de positionner la caméra de façon plus intuitive.

5.3.3.1 Méthode par calibrage stéréoscopique fort

La méthode par placement euclidien de la vue virtuelle avec un calibrage fort est la méthode la plus souvent utilisée car elle est la plus facile à appliquer. Cette méthode a été présentée initialement par S. Avidan et A. Shashua (1998) et rappelée récemment par la suite dans H. Mayer (2002). Elle consiste à définir la matrice de rotation R directement par le calcul du vecteur d'approximation pour des petits angles de rotation autour de chaque axe de la vue virtuelle $\Omega = (\Omega_X, \Omega_Y, \Omega_Z)^T$ par l'opération $[\Omega]_\times$:

$$R = \begin{bmatrix} 1 & \Omega_Z & -\Omega_Y \\ -\Omega_Z & 1 & \Omega_X \\ \Omega_Y & -\Omega_X & 1 \end{bmatrix} \quad (5.30)$$

Avant d'appliquer la rotation R il faut déterminer l'orientation et la position initiale de la vue ψ'' dans l'espace euclidien. Elle est déterminée par le calibrage stéréoscopique fort. Ce type de calibrage permet de connaître la géométrie absolue entre les deux vues réelles ψ et ψ' . Pour cela il faut utiliser la matrice essentielle E qui est la matrice fondamentale calibrée :

$$E = K^T F K \quad (5.31)$$

avec K la matrice des paramètres intrinsèques de la caméra (calibrage fort). Par le biais de la matrice essentielle on peut extraire la rotation et la translation qui existe entre les deux vues réelles, car elle s'écrit de la façon suivante :

$$E = [t_0]_\times R_0 \quad (5.32)$$

avec t_0 un vecteur de translation et R_0 une matrice de rotation dans l'espace euclidien \mathcal{R}^3 . Donc en connaissant l'orientation d'origine de la vue ψ' il est possible de spécifier l'orientation de la vue virtuelle $\hat{\psi}''$ par R . Maintenant la difficulté réside dans

la détermination du vecteur de translation t dans l'espace euclidien \mathcal{R}^3 . Ce vecteur s'écrit $v''^l r_l^k - \hat{v}''^k$, or on ne connaît pas la position finale dans l'espace de \hat{v}'' . C'est pourquoi la valeur du vecteur de translation t est choisie de façon empirique pour qu'elle corresponde avec la vue attendue de la vue virtuelle. S. Avidan et A. Shashua (1997) proposent une variante en utilisant la transformation initiale entre les deux vues réelles comme incrément de transformation. Pour cela la vue virtuelle est placée par des pas de transformation de valeur R_0 et t_0 . Si le déplacement est important entre les vues ψ et ψ' , ils proposent de définir une trajectoire de la vue virtuelle avec des points clés, correspondant à chaque incrément et à utiliser une interpolation pour estimer les paramètres entre chaque pas de la matrice de rotation R et du vecteur de translation t . Cette méthode, qui nécessite un calibrage fort, ne permet pas d'obtenir une trajectoire totalement libre de la nouvelle vue. Étant donné qu'elle est recopiée sur le déplacement et l'orientation des deux vues réelles certains mouvements sont impossibles à obtenir.

5.3.3.2 Méthode par calibrage stéréoscopique faible

S. Avidan et A. Shashua (1997) ont proposé également une autre méthode pour estimer la position et l'orientation de la vue initiale sans passer par la matrice essentielle, c'est-à-dire sans calibrage fort des caméras. Elle consiste à calculer la matrice de rotation R_0 entre ψ et ψ' sans calibrage. Dans l'hypothèse que les vues réelles soient proches l'une de l'autre, B. Rousso et al. (1996) montre qu'il est possible d'extraire à partir du tenseur trilinéaire, les rotations entre les deux vues. Il en fait la démonstration dans une application de stabilisation vidéo. Soit le tenseur trilinéaire estimé à partir de deux vues \tilde{T} , les valeurs des angles de rotation se définissent de la façon suivante :

$$\Omega_X = k^{-1} \det \begin{bmatrix} \tilde{T}_2^{j3} \\ \tilde{T}_2^{j3} + \tilde{T}_3^{j2} \\ \tilde{T}_3^{j3} + \tilde{T}_2^{j2} \end{bmatrix} \quad (5.33)$$

et,

$$\Omega_Y = k^{-1} \det \begin{bmatrix} -\tilde{T}_1^{j3} \\ \tilde{T}_2^{j3} + \tilde{T}_3^{j2} \\ \tilde{T}_3^{j3} - \tilde{T}_2^{j2} \end{bmatrix} \quad (5.34)$$

et,

$$\Omega_Z = k^{-1} \det \begin{bmatrix} \tilde{T}_1^{j2} \\ \tilde{T}_2^{j3} + \tilde{T}_3^{j2} \\ \tilde{T}_3^{j3} - \tilde{T}_2^{j2} \end{bmatrix} \quad (5.35)$$

avec,

$$k = \det \begin{bmatrix} \tilde{T}_2^{j2} \\ \tilde{T}_2^{j3} + \tilde{T}_3^{j2} \\ \tilde{T}_3^{j3} - \tilde{T}_2^{j2} \end{bmatrix} \quad (5.36)$$

avec par exemple $\tilde{\mathcal{T}}_2^{j2}$ la forme compactée de $\{\tilde{\mathcal{T}}_2^{12}, \tilde{\mathcal{T}}_2^{22}, \tilde{\mathcal{T}}_2^{32}\}$. Après le calcul de la matrice R_0 à partir de l'extraction des angles de rotation des deux vues réelles la procédure est quasi la même que dans la méthode avec calibrage. En effet si cette méthode permet de calculer R_0 , le vecteur de translation t_0 entre ψ et ψ' est inconnu et donc le vecteur de placement t aussi. Les auteurs, très vagues sur le paramétrage de t , précisent qu'il est spécifié par l'utilisateur, c'est-à-dire que sa valeur est empirique et donc basée sur le savoir faire de l'utilisateur en fonction de la configuration du système. Une solution pourrait néanmoins permettre une estimation de la translation entre ψ et ψ' par la disparité. La disparité correspond à la différence de position en pixel d'un couple de points homologues dans les plans images de deux vues (N. Ayache 1989),

$$\delta_{pp'} = x' - x \quad (5.37)$$

avec $\delta_{pp'}$ la disparité, x et x' les abscisses des points respectivement p et p' dans les images I et I' . Pour estimer le vecteur de translation quelques informations sont nécessaires. C'est un peu un calibrage partiel, en effet dans l'ouvrage de référence O. Faugeras (1993), on retrouve l'équation suivante,

$$\delta = \frac{d_{OO'} f}{dz} \quad (5.38)$$

avec $d_{OO'}$ la distance entre les centres optiques des vues ψ et ψ' , f la distance focale (qui doit être identique sur les deux vues) et dz la distance entre l'objectif des caméras et l'objet observé. Donc pour calculer le vecteur de translation entre les deux vues réelles, c'est-à-dire la distance ente les centres optiques $d_{OO'}$ il faut des informations sur la configuration du système de vision (distance qui sépare les caméras) ainsi que la distance focale de chaque caméra.

La seule et unique méthode sans calibrage est de spécifier la vue dans l'espace projectif, en utilisant la matrice de colinéation D de l'équation (5.27). Comme expliqué au début de cette section, ce mode de placement n'est pas intuitif. C'est pourquoi S. Avidan et A. Shashua (1998) utilisent une matrice de colinéation D_r qui ne contient que les rotations de la vue virtuelle. Pour déterminer la matrice D_r ils utilisent un plan virtuel ou physique dans l'image. En effet, dans le calibrage faible un plan physique dans la scène est utilisé pour obtenir la matrice de colinéation A . On peut très bien utiliser ce plan et simuler une rotation autour de celui-ci. Cela signifie que l'utilisateur choisit dans l'image 4 points qui définissent ce plan. Par la suite il imagine la projection de ces 4 points dans l'image virtuelle. Par exemple si les 4 points formaient un plan de l'image de 100×100 pixels, une rotation sur Ω_X de 45° de la vue virtuelle transformerait le plan à une dimension de $100 \times \cos(45) = 71 \text{ pixels} \times 100 \text{ pixels}$. En spécifiant les 4 points du plan qu'observerait la vue virtuelle il est facile de déterminer la transformation projective entre ces deux plans pour obtenir la matrice D_r . Pour le vecteur de translation, même combat que dans la méthode précédente, il doit être spécifié empiriquement par l'utilisateur.

L'ensemble des méthodes du placement de la vue virtuelle, dans un cas de calibrage stéréoscopique faible, n'est pas concluant. En effet le placement de la vue virtuelle est

basée sur du tâtonnement, il est très difficile pour une scène quelconque de choisir automatiquement, par exemple la vue à 45° . C'est pourquoi, dans notre contexte nous avons été obligé de développer une nouvelle approche de placement de la vue virtuelle.

5.3.3.3 Notre méthode avec calibrage stéréoscopique faible

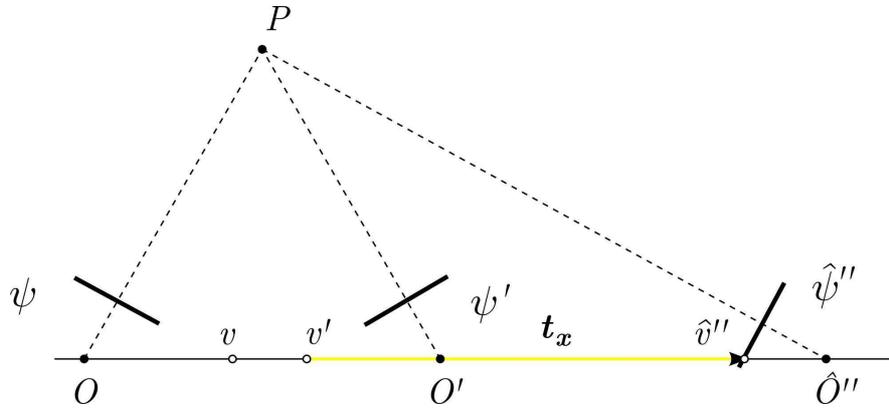
Nous sommes toujours dans un système dont on ne connaît que le calibrage faible. Dans les méthodes précédentes plusieurs placements dans l'espace projectif \mathcal{P}^3 sont possibles : avec une matrice de colinéation D complète et un vecteur de translation $t = 0$, avec une matrice de colinéation incomplète D_r (juste les rotations) et un vecteur de translation t de valeur empirique. Notre méthode va consister à utiliser seulement le vecteur de translation t pour spécifier la position de la vue virtuelle. L'orientation de cette vue va se contraindre automatiquement par le biais du lien géométrique dans l'espace projectif entre la nouvelle vue et la configuration du système.

Soit la configuration de deux vues réelles ψ et ψ' à la Figure 5.5. Nous définissons le point P de l'espace projectif \mathcal{P}^3 comme étant le centre de rotation entre les vues ψ et ψ' dans \mathcal{P}^3 . L'orientation de la vue ψ' est contraint par la géométrie épipolaire. On suppose au départ que la vue virtuelle $\hat{\psi}''$ est confondue avec la vue ψ' . Nous fixons la matrice de rotation en tant que matrice identité \mathcal{I} , le nouveau tenseur s'exprime ainsi :

$$\hat{\mathcal{T}}_i^{jk} = \mathcal{I}\tilde{\mathcal{T}}_i^{jl} + t^k\alpha_i^j \quad (5.39)$$

Si on applique la translation t_x selon l'axe X de l'espace projectif via le vecteur t^k , l'épipole de la vue virtuelle $\hat{\psi}''$ va se déplacer. Comme cette vue virtuelle subit une contrainte liée à la géométrie trifocale avec les deux autres vues réelles, elle va se déplacer également. Le centre optique \hat{O}'' se déplace pour conserver la contrainte lié au centre P qui est l'axe de rotation entre les deux vues réelles. La vue virtuelle se reconfigure de telle sorte que son orientation soit toujours dans l'axe du centre P quelle que soit la translation de l'épipole (Figure 5.5). Cette auto-orientation vient de différentes contraintes : épipolaire (projection du centre optique O dans $\hat{\psi}''$) ou physique (distance entre le plan image et le centre optique \hat{O}'' constant). Cela a pour conséquence de forcer la vue $\hat{\psi}''$ à subir une rotation par rapport à elle même autour de son axe Y de l'espace projectif. Il est également possible de faire exactement le même procédé avec la translation t_y selon l'axe Y de l'espace projectif, pour contraindre la vue virtuelle à subir une rotation autour de son axe X de l'espace projectif.

Donc en jouant juste sur le vecteur de translation t , il nous est possible de forcer une rotation de la vue virtuelle par rapport au centre P . Par contre cette approche nécessite des dispositions particulières dans l'estimation de la matrice de colinéation. En effet le point P défini comme centre, représente l'axe de rotation entre les vues ψ et ψ' et est ni plus ni moins que l'axe de rotation du plan physique (ou virtuelle) utilisé pour estimer la matrice de colinéation. Or si les points de l'objet observé sont complètement en dehors de cet axe, l'objet va sortir du champ de la vue virtuelle. Comme présenté dans la Figure 5.6, il faut se garantir que le plan π utilisé dans l'estimation de la matrice de colinéation soit centré par rapport à l'objet (ou à la scène) que l'on va observer, afin de garantir le meilleur résultat possible.

FIG. 5.5 – Placement de la vue virtuelle en fonction de t_x .

Exemple A partir du tenseur \tilde{T} que nous avons calculé dans l'exemple précédent, de la matrice de colinéation A , nous allons déterminer le nouveau tenseur \hat{T}_i^{jk} pour une vue virtuelle placée par le vecteur de translation t . Arbitrairement nous choisissons de faire subir à la vue virtuelle une rotation sur l'axe Y de \mathcal{P}^3 , en utilisant l'élément t_x pour une valeur de 1000, dont voici le code Matlab/Octave :

```

0000 % Nouveau Tenseur newT d'indices (k,i,j)
0001 newT=zeros(3,3,3);
0002 % Définition de tk avec tx=1000
0003 tk=[1000;0;0];
0004 % Contraint D en identité
0005 D=eye(3,3);
0006 % Calcul le nouveau tenseur newT
0007 for j=1:3
0008     for k=1:3
0009         for i=1:3
0010             newT(k,i,j)=(D(k,1)*T(1,i,j)+D(k,2)*T(2,i,j) ...
0011                 +D(k,3)*T(3,i,j))+t(k,1)*A(j,i);
0012         end
0013     end
0014 end

```

Avec, dans l'algorithme, T qui correspond au tenseur d'amorçage \tilde{T} , A à la matrice de colinéation et \mathbf{newT} au nouveau tenseur \hat{T} .

5.3.3.4 Zone de validité du placement

La méthode de placement par le vecteur de translation t^k n'est pas valide pour toutes les positions de la vue virtuelle. En effet il existe une bande de singularité pour les vues virtuelles en interpolation, c'est-à-dire qui sont placées sur la ligne de base (ligne entre

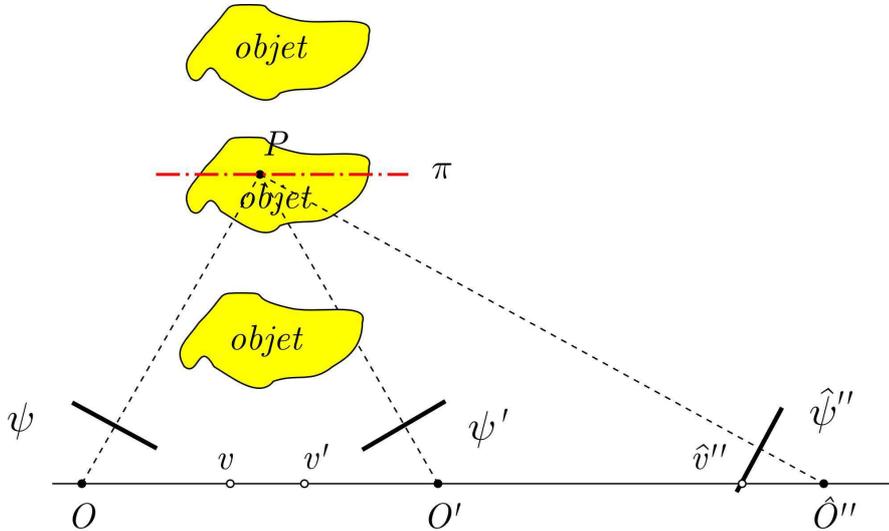


FIG. 5.6 – Placement de l'objet ou de la scène en fonction du plan π pour l'estimation de la matrice de colinéation.

O et O') des deux vues réelles. Cette bande de placement produit une géométrie à la contrainte incohérente (Figure 5.7). Dans le cas où une vue virtuelle est placée en interpolation de vue, si les contraintes doivent être respectées la vue virtuelle doit à la fois être orientée vers le centre P et à la fois avoir un épipole sur la ligne de base. Or si la vue virtuelle est orientée vers le centre P l'épipole est repoussé à l'infini et ne permet pas d'être sur la ligne de base entre les deux centres optiques O et O' . Le placement de la vue virtuelle en interpolation est donc impossible du fait de cette singularité de configuration. Le placement de la vue n'est valide que pour des extrapolations, c'est-à-dire une vue virtuelle en dehors de la bande entre les centres optiques des deux vues réelles. Pendant le déplacement de la vue virtuelle il faut éviter d'avoir une trajectoire qui passe dans la bande de singularité. Par exemple si la vue virtuelle se déplace de la vue ψ à la vue ψ' (de gauche à droite), elle est obligée de passer par la bande de singularité. Pour éviter une telle configuration nous pouvons utiliser deux modèles de placement. Un pour le placement de la vue virtuelle dans la bande d'extrapolation correspondant à la position de ψ vers $-\infty$ sur l'axe x et un autre pour la position ψ' vers $+\infty$ sur l'axe x . Chaque modèle est déterminé par le calcul d'un nouveau tenseur en prenant comme vue confondue avec la vue virtuelle, ψ ou ψ' . Dans le cas d'une trajectoire de ψ vers ψ' , par exemple, le premier modèle est utilisé pour placer la vue virtuelle entre la position ψ et $-\infty$. Lorsque la position de la vue virtuelle arrive à la position de la vue ψ pour rentrer dans la bande de singularité on place sa position directement en sortie de la bande à la position de la vue ψ' en permutant sur le modèle correspondant. Pour minimiser la bande de singularité on peut choisir une configuration où les vues réelles sont très proches l'une de l'autre. Cela permet d'obtenir une meilleure continuité dans la trajectoire tout en rejetant les positions singulières de la vue virtuelle.

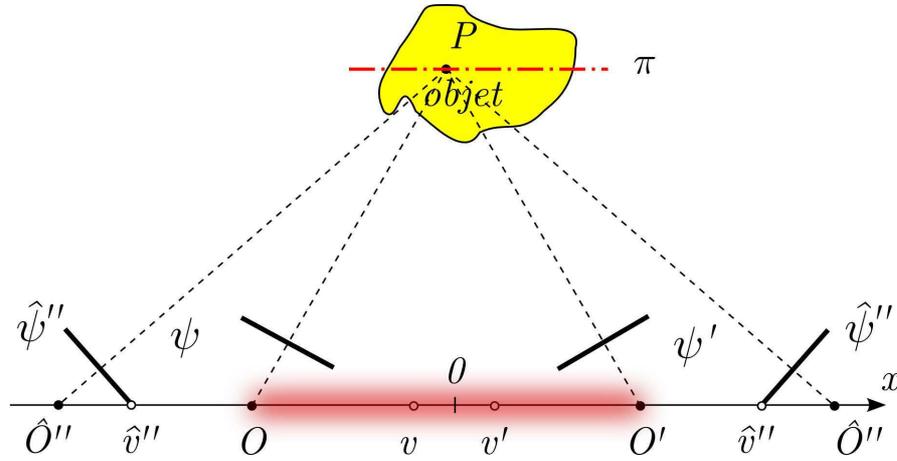


FIG. 5.7 – Bande de placement où la validité est incohérente.

5.3.3.5 Placement automatique de la vue

La quantification du vecteur de translation t^k est toujours empirique, car il dépend de la géométrie des deux vues réelles. Il est donc très difficile de savoir qu'elle valeur choisir pour obtenir une vue virtuelle avec une rotation précise. Première conséquence de notre méthode de placement c'est que la vue ne peut pas atteindre une rotation supérieure ou égale à 90° . En effet quand t_x tend vers l'infini, l'angle de rotation de la vue tend vers 90° . Deuxième conséquence, le placement de la vue virtuelle n'est contrôlée que par le vecteur t^k . Dans les méthodes précédentes, après spécification de la matrice de rotation D , la valeur de t_k devait être calculée ou trouvée de façon empirique pour coller à la rotation que la vue virtuelle avait subie. Dans notre méthode seul ce paramètre joue sur la rotation de la vue. Cela va nous permettre de contrôler la quantité à affecter au vecteur t_k en fonction de l'angle désiré pour la rotation de la vue virtuelle. Nous allons procéder de la façon suivante. Par création d'un objet factice dans l'image quasiment parallèle au plan de la vue ψ' , nous allons pour plusieurs valeurs de t_x (idem pour t_y) mesurer la taille de cet objet dans la vue virtuelle. En déduisant l'angle de rotation entre la vue réelle ψ' et la vue virtuelle $\hat{\psi}''$ par le biais de la variation de la taille de cet objet factice, nous allons créer un modèle mathématique de l'angle de rotation en fonction de t_x . Ce qui nous permettra pour n'importe quel angle souhaité de déduire directement la valeur du vecteur de translation t_k .

Pour cela nous sommes obligés de poser un modèle équivalent simplifié du placement de la vue virtuelle. Considérons le modèle équivalent présenté à la Figure 5.8. Soit θ_x l'angle de rotation entre la vue réelle ψ' et la vue virtuelle $\hat{\psi}''$. Ne pouvant exprimer cet angle en fonction de t_x nous posons le modèle équivalent avec le nouvel axe de rotation par le point P^* placé à la perpendiculaire de l'épipole v' . La distance entre P^* et v' représente la hauteur h_0 du triangle $P^*v'\hat{v}''$, rectangle en v' . Ce modèle permet d'exprimer l'angle rotation θ_x^* , entre les deux épipoles v' et \hat{v}'' par rapport au nouvel axe établi au point P^* . θ_x^* est l'approximation de l'angle réel θ_x . Le modèle équivalent

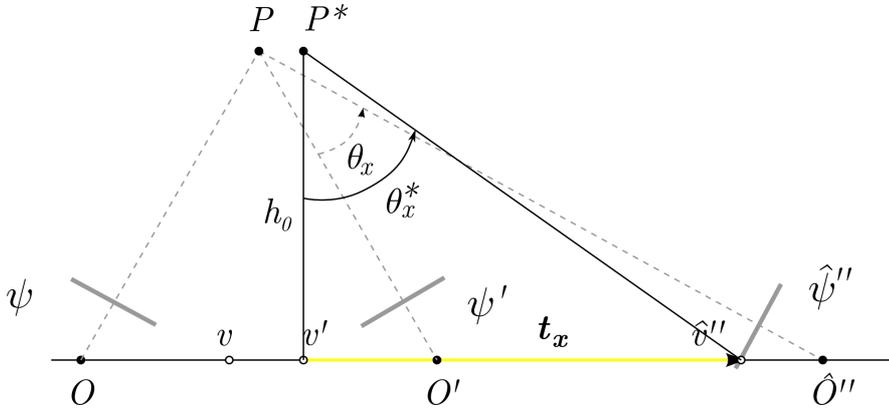


FIG. 5.8 – Configuration du modèle équivalent simplifié du placement de la vue virtuelle.

permet d'exprimer l'angle θ_x^* en fonction de h_0 et du vecteur de translation t_x , de la façon suivante :

$$t_x = h_0 \tan \theta_x^* \quad (5.40)$$

Nous avons la hauteur h_0 qui est inconnue dans cette équation. Pour pouvoir l'estimer nous allons utiliser un objet factice ou réel dans la scène. Soit un objet factice ou réel dans l'espace \mathcal{P}^3 de longueur L_0 projeté dans le plan image \mathcal{P}^2 de la vue réelle ψ' . Lorsque cet objet est projeté dans l'image de la vue virtuelle, qui subit une rotation, sa longueur va diminuer en fonction de l'angle de rotation. On peut exprimer sa longueur L comme suit :

$$L(t_x) = L_0 \cos \theta_x(t_x) \quad (5.41)$$

Pour obtenir h_0 , il nous faut déterminer avec un seul angle la valeur de t_x . Par exemple pour l'angle réel $\theta_x = 60^\circ$ qu'elle est la valeur de t_x . Pour un angle de 60° la longueur de l'objet factice sera, d'après la trigonométrie, d'une valeur de $L_0/2$. Donc en utilisant une méthode d'optimisation, comme le simplexe de Nelder-Mead, nous pouvons chercher quelle est la valeur de t_x qui permet de minimiser une fonction, lorsque la longueur de l'objet factice est de moitié. Une fois cette valeur déterminée de t_x , on suppose l'équivalence approximatif de θ_x avec θ_x^* pour calculer la hauteur h_0 . Ainsi nous obtenons un modèle équivalent, qui en fonction de l'angle désiré, nous donne la valeur du vecteur de translation à appliquer à la nouvelle vue.

En réalité dans ce placement automatique, nous avons présenté deux méthodes. Une rapide mais approximative, qui consiste à créer un modèle équivalent du système permettant d'obtenir par la suite facilement la valeur de t^k . Une autre précise mais plus longue, qui utilise une méthode d'optimisation pour déterminer l'angle exacte en mesurant le rapport de longueur de l'objet factice pour chaque position de recherche de la nouvelle vue.

Application Nous allons utiliser nos images exemples, pour expliquer une méthode d'estimation d'objet factice. Ensuite nous estimerons son modèle équivalent simple et nous le comparerons avec des mesures directes sur l'objet factice, afin de le valider. Premièrement nous allons voir comment construire un objet factice purement virtuel dans l'image et parallèle au plan image de la vue ψ' . Sur la première image exemple nous lui appliquons un détecteur d'arêtes de type Canny (J. Canny 1986). Pour chaque point du contour de l'image I associée à la vue ψ on cherche son correspondant dans l'image I' associée à la vue ψ' . Pour augmenter la rapidité de la mise en correspondance par corrélation on utilise la méthode de Zhang (Z. Zhang et al. 1995). Cette méthode consiste à rechercher un point sur une trajectoire à une dimension au lieu d'une zone à deux dimensions en utilisant la ligne épipolaire. En effet si le calibrage faible est connu, nous connaissons la matrice fondamentale, il est donc possible de calculer la ligne épipolaire qui passe forcément par le point que l'on recherche dans l'image I' . Après cette étape on connaît les points de contour dans le couple d'images (I, I') . A partir des points appariés des contours p et p' nous allons calculer la disparité. Le but étant d'extraire un plan virtuel parallèle en utilisant les points de même disparité. On calcule la disparité comme la distance Euclidienne entre p et p' ,

$$\delta_{pp'} = \sqrt{(y' - y)^2 + (x' - x)^2} \quad (5.42)$$

avec $\delta_{pp'}$ la disparité entre les points $p = (x, y, 1)^T$ et $p' = (x', y', 1)^T$. On représente la disparité sous forme de calque en fonction du produit entre x et y , le résultat est présenté à la Figure 5.9. Chaque calque montre les points de même disparité, ces points peuvent être considérés comme appartenant à un même plan parallèle au plan image de l'imageur (contrainte fronto-parallèle). Pour la création de l'objet factice nous voulons extraire le plan qui contient la plus grande population de points. Pour le choisir, nous représentons sous forme d'histogramme la fréquence des points en fonction de la disparité à la Figure 5.9.

On isole le calque intéressant par l'extraction des points concernés dans l'ensemble des points p' , qui correspond aux points de la vue où nous allons créer l'objet factice. Nous obtenons un ensemble de points de même disparité qui s'écrit de la forme :

$$p_\delta = \left\{ p'_i \mid \delta(p_i) = \max[\tilde{f}(\delta_{pp'})] \right\} \quad (5.43)$$

avec p_δ les points extraits, $\tilde{f}(\delta_{pp'})$ l'histogramme des disparités, $\delta(p_i)$ la disparité et p'_i les points du contour de l'image I' . Les points extraits sont montrés à la Figure 5.10, elle montre une des images originales ainsi que l'image des points de contour avec les points extraits en sur-impression.

A partir de l'ensemble des points que nous avons extrait, nous allons créer un objet factice. En utilisant la disparité on est certain que ces points appartiennent, à peu près, au même plan avec une contrainte fronto-parallèle respectée. A partir des points p_δ nous allons déterminer un objet de type segment. Un objet simple est suffisant pour notre application, en effet on désire juste l'angle de rotation de la vue virtuelle par rapport à notre système de deux vues virtuelles selon l'axe Y . Dans le cas où nous aurions voulu

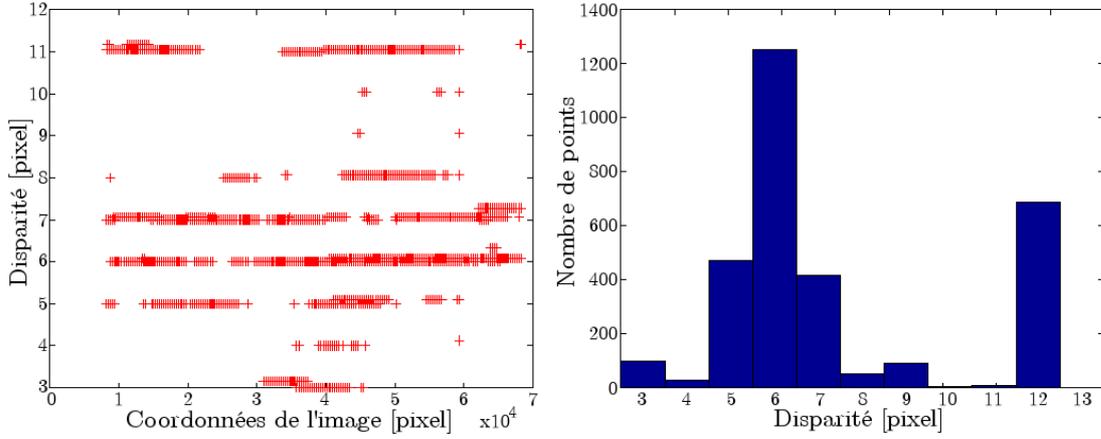


FIG. 5.9 – À gauche, la carte des calques de même disparité, à droite, l’histogramme des disparités.

un degré de liberté supplémentaire avec, par exemple, la mesure de l’angle selon l’axe X , nous aurions créé un objet avec une forme de type carré ou croix. Ce type d’objet nous aurait permis de mesurer plusieurs angles dans la même image. Pour définir ce segment, nous allons partager notre ensemble de points de telle sorte à pouvoir calculer, à partir de ces sous-ensembles, des points clés qui définiront l’objet. Pour définir deux sous-ensembles nous allons utiliser le centre de gravité du nuage de points que constitue p_δ , en utilisant cette fonction :

$$\bar{c} = (\bar{x}, \bar{y}, 1)^T = \left(\frac{1}{\text{card}(p_\delta)} \sum_{i=1}^{\text{card}(p_\delta)} p_{\delta\ i}^1, \frac{1}{\text{card}(p_\delta)} \sum_{i=1}^{\text{card}(p_\delta)} p_{\delta\ i}^2, 1 \right)^T \quad (5.44)$$

avec \bar{c} le centre de gravité du nuage de points p_δ . À partir de ce centre, on définit deux sous-ensembles : un à gauche, Ω_g et un à droite Ω_d . La séparation des sous-ensembles est réalisée à partir de l’abscisse du centre de gravité de la façon suivante :

$$\Omega_g = \{p_{\delta\ i} | p_{\delta\ i}^1 > \bar{x}\} \quad (5.45)$$

et,

$$\Omega_d = \{p_{\delta\ i} | p_{\delta\ i}^1 < \bar{x}\} \quad (5.46)$$

Ces deux sous-ensembles affichent clairement la séparation par rapport à la droite de séparation verticale constituée de l’abscisse de \bar{c} . La Figure 5.11-a montre les deux sous-ensembles obtenus, avec les points en gris pour l’ensemble de gauche, et les points en noir pour l’ensemble de droite.

À partir de là, on calcule le centre de gravité de chaque sous-ensemble juste sur l’axe des abscisses, de telle sorte à obtenir un segment purement horizontal dans l’image passant par le point \bar{c} . Ces centres de gravité vont définir les points clés du segment,



FIG. 5.10 – A gauche, l'image de la vue droite originale, à droite, les points de contour avec les points p_δ en noir.

avec le centre p_g de l'ensemble de gauche Ω_g et le centre p_d de l'ensemble de droite Ω_d . Chaque point p_g et p_d est obtenu de la façon suivante :

$$p_g = (x_g, \bar{y}, 1)^T = \left(\frac{1}{\text{card}(\Omega_g)} \sum_{i=1}^{\text{card}(\Omega_g)} \Omega_g^1_{i, \bar{y}, 1} \right)^T \quad (5.47)$$

et,

$$p_d = (x_d, \bar{y}, 1)^T = \left(\frac{1}{\text{card}(\Omega_d)} \sum_{i=1}^{\text{card}(\Omega_d)} \Omega_d^1_{i, \bar{y}, 1} \right)^T \quad (5.48)$$

Au final nous obtenons le segment factice composé des points p_g, \bar{c}, p_d . Ce segment est tracé à la Figure 5.11-b. C'est grâce à lui que nous allons pouvoir modéliser le déplacement de la vue virtuelle afin de prédire la valeur à choisir pour le vecteur de translation t^k . La taille du segment dans l'image diminue en fonction de t^k , car cet objet factice est projeté dans la vue virtuelle qui subit une rotation dans l'espace. L'évolution de la taille de ce segment en fonction de t^k est tracé à la Figure 5.12-a. En utilisant la méthode décrite précédemment nous utilisons une méthode d'optimisation du type simplexe de Nelder-Mead pour rechercher avec précision une valeur particulière de l'angle de rotation en mesurant la taille du segment. Par exemple, quand le segment a une longueur de moitié cela correspond à une rotation de 60° . Par le biais de cette valeur mesurée, nous créons le modèle de rotation en fonction de t^x . Ce modèle est un modèle équivalent. Pour vérifier dans quelle mesure il est valide, nous traçons sur le même graphique, à la Figure 5.12-b, l'angle de vue obtenu par le modèle équivalent en fonction de t^x et les mesures réelles des angles de rotation pour chaque valeur de t^x . Malgré le fait d'utiliser un modèle équivalent approximatif, le modèle est très proche des mesures réelles. Il existe juste cependant un petit décalage d'erreur soit négatif, soit positif, sur l'angle de rotation en fonction de la valeur de t^x .



FIG. 5.11 – Création de l'objet factice : (a) séparation des points en deux sous-ensembles par rapport à son centre de gravité, (b) représentation des points clés du segment factice.

5.3.4 Rendu

Maintenant nous supposons que la vue virtuelle est positionnée à l'endroit désiré par l'utilisateur. Par conséquent, le tenseur \hat{T} associé à ce nouveau système de configuration est connu. Dans le début de ce chapitre, nous avons précisé, que si on connaissait un tenseur \hat{T} ainsi que l'ensemble des points mis en correspondance p et p' il était possible de retrouver les points \hat{p}'' dans l'image \hat{I}'' . Cette étape de reconstruction de l'information de la nouvelle vue correspond à une étape de rendu. Elle est découpée en deux parties, une géométrique et une photométrique. Afin de présenter progressivement l'étape de rendu, nous allons dans un premier temps nous focaliser sur la partie géométrique. Ce qui signifie que nous allons étudier simplement la position des points projetés dans la vue virtuelle sans se soucier de l'information de l'intensité des pixels qu'ils transportent. Et pour cela, nous allons utiliser simplement les informations des contours des deux images réelles. Les points des contours des objets montreront si géométriquement la silhouette de ces objets est cohérente dans la vue virtuelle. Une fois la géométrie validée, nous nous intéresserons à l'aspect photométrique du rendu. Dans ce mode de rendu nous allons projeté non plus des contours mais bien l'ensemble de l'image texturée.

5.3.4.1 Forte extrapolation

Pour l'instant on utilise que les points de contour de l'image. Soit p un point de contour de l'image I et p' le point homologue de contour de l'image I' . Pour chaque couple de points (p, p') , on peut calculer à partir du nouveau tenseur \hat{T} , le point correspondant dans la vue virtuelle \hat{p}'' . Ce point noté \hat{p}'' est transféré dans la vue virtuelle par la relation point/point/point du tenseur trilineaire de la façon suivante :

$$\hat{x}'' = \frac{x' \hat{T}_i^{31} p^i - \hat{T}_i^{11} p^i}{x' \hat{T}_i^{33} p^i - \hat{T}_i^{13} p^i} \quad (5.49)$$

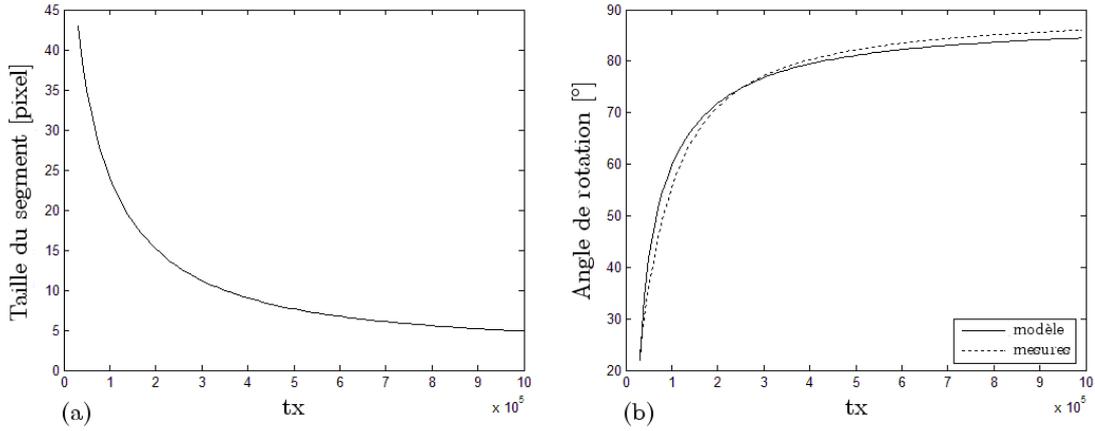


FIG. 5.12 – Mesures du segment factice : (a) taille du segment en fonction de t^x , (b) comparaison entre le modèle de placement et les mesures réelles.

et,

$$\hat{y}'' = \frac{x' \hat{T}_i^{32} p^i - \hat{T}_i^{12} p^i}{x' \hat{T}_i^{33} p^i - \hat{T}_i^{13} p^i} \quad (5.50)$$

avec $\hat{p}'' = (\hat{x}'', \hat{y}'', 1)^T$. Le choix du type de placement de la vue virtuelle est très important par rapport au rendu. En effet, ces deux équations sont totalement valides si on a utilisé un placement de la vue avec un calibrage fort. Par contre, dans notre cas, où la vue est spécifiée par une contrainte très spécifique, nous avons un phénomène de détérioration qui apparaît dans les fortes extrapolations. On entend par forte extrapolation, une configuration où la vue virtuelle est à une position très éloignée par rapport à celles des deux vues réelles. Lorsqu'on atteint une valeur du vecteur de translation correspondant à un angle important d'extrapolation, la géométrie de la vue subit une dégénérescence. Les points des objets ont une expansion en fonction de l'angle de rotation. Au final, nous obtenons une image virtuelle avec une perte de cohésion, il est alors impossible de distinguer les silhouettes des objets. La Figure 5.13 montre un exemple de ce phénomène. La première image représente les points de contour d'une de nos images exemples. La seconde est le rendu de la vue virtuelle pour une forte extrapolation d'environ 75° . On remarque que les points se sont dispersés dans l'image, générant une incohérence géométrique. Par exemple les points de contour des poissons sont maintenant hors champ.

Pour pouvoir palier ce problème, il faut déjà en connaître les causes. Cette perte de cohérence géométrique dans la reprojection est propre à notre placement de la vue virtuelle par le vecteur de translation t . La première approche aurait été de penser que cela vient de la translation de la vue virtuelle vers l'infini. Effectivement, plus nous désirons un angle d'extrapolation important et plus la vue virtuelle par le biais de t s'éloigne de l'objet observé. Mais cette hypothèse ne permet pas de comprendre pourquoi les points sont en expansion et non pas l'inverse. En toute logique si on s'éloigne de l'objet observé, il va

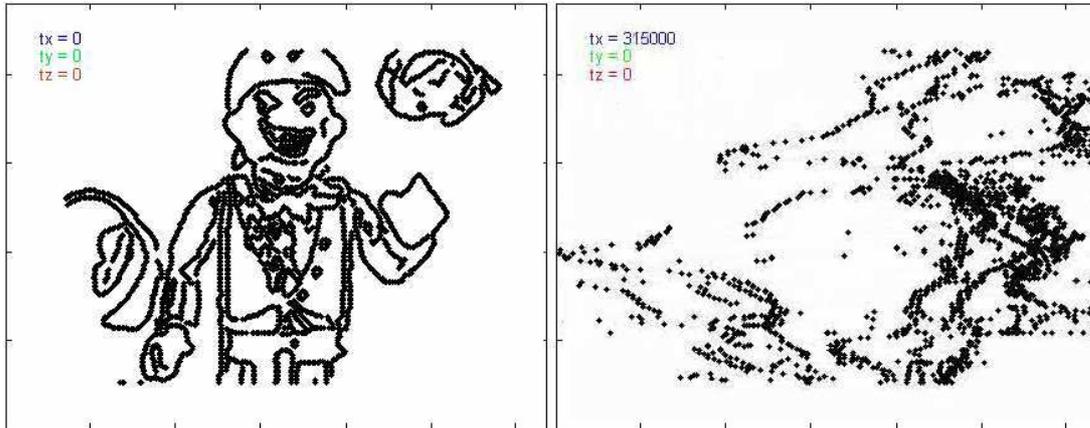


FIG. 5.13 – A gauche, les points de contours pour une vue à 0° , à droite, les mêmes points de contours pour une vue à 75° .

devenir de plus en plus petit dans l'image, donc les points de contour devraient tous se contracter et se regrouper dans l'image et non s'étendre. En réalité, l'explication est tout autre bien qu'elle soit issue de l'éloignement de la vue virtuelle. La Figure 5.14 présente le mécanisme d'expansion des points de la vue virtuelle. Considérons un ensemble de points qui décrit l'objet dans une des vues réelles (Figure 5.14-a), l'image de cette vue a un facteur d'échelle de valeur λ_1 . Pour l'estimation du tenseur d'armorçage, nous avons utilisé des éléments de cette image, par conséquent le tenseur est en quelque sorte défini pour se facteur d'échelle. Après avoir placé la vue virtuelle avec une forte extrapolation par le vecteur de translation t , on détermine le nouveau tenseur. Or maintenant ce tenseur intègre une vue virtuelle qui est très éloignée de l'objet observé. Par conséquent l'image théorique donnée par la vue virtuelle a un facteur d'échelle λ_i qui est inconnu et beaucoup plus petit que λ_1 (Figure 5.14-b). Au moment de la reprojection des points dans la vue virtuelle, pour qu'elle soit cohérente tous les points doivent être homogènes. Ce qui nous demande une étape de homogénéisation des points pour que leurs coordonnées z soient égale à un. En fait cette étape correspond à rectifier le facteur d'échelle λ_i au facteur d'échelle λ_1 (Figure 5.14-c). Étant donné que λ_i est inférieur à λ_1 , il existe entre les deux, un facteur de correction non nul λ_k de valeur :

$$\lambda_k = \frac{\lambda_1}{\lambda_i} \quad (5.51)$$

Or les points utilisés pour la reprojection p et p' sont déjà à un facteur d'échelle λ_1 . C'est donc ce facteur d'échelle λ_k au moment de l'homogénéisation qui est responsable de l'expansion des points. Car l'image de la vue virtuelle est à un facteur d'échelle différent des points utilisés pour la reprojection.

Par conséquent, pour corriger ce problème, nous allons introduire un facteur de correction d'échelle. Il y a deux possibilités sur l'endroit d'intervention du facteur de correction. Soit on affecte une correction aux points d'origine dans les images ou soit on corrige

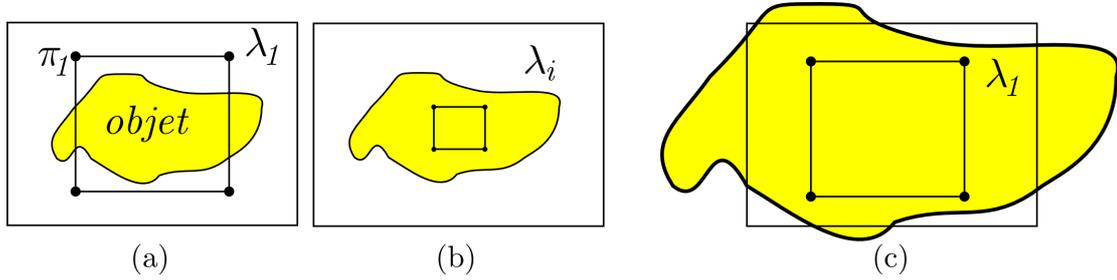


FIG. 5.14 – Mécanisme de la perte de cohésion en forte extrapolation : (a) image de la scène avec le facteur d'échelle λ_1 , (b) plan π_1 au facteur d'échelle λ_1 utilisé pour l'estimation du tenseur trilinéaire, (c) image de la même scène mais avec une position de la vue virtuelle très éloignée.

directement pendant le calcul la position de chaque point dans la vue virtuelle. Comme il est impossible de connaître avec exactitude le facteur d'échelle λ_i pour une vue virtuelle quelconque, nous choisissons d'appliquer le facteur de correction pendant la reprojexion des points dans la vue virtuelle. Pour cela nous allons utiliser une contrainte basée sur le principe que l'objet observé doit toujours être dans le champ de la vue virtuelle. Donc en déterminant les limites des points de l'objet pendant la reprojexion on peut déterminer le facteur de correction à apporter pour qu'il tienne dans l'image. En plus de la correction d'échelle nous allons ajouter une correction de positionnement. En effet, dans les sections précédentes nous avons précisé que l'objet observé devait être placé au centre du plan utilisé dans l'estimation du tenseur. Si ce n'était pas le cas, l'objet pendant l'extrapolation sort du champ (voir §5.3.3.3). Nous allons donc déterminer le centre de gravité de l'objet et lui appliquer cette nouvelle correction pour qu'il soit toujours placé au centre de l'image. Au final, l'ensemble des points \hat{p}'' sont calculés et corrigés de la manière suivante :

$$\hat{x}'' = \lambda_x \frac{x' \hat{T}_i^{31} p^i - \hat{T}_i^{11} p^i}{x' \hat{T}_i^{33} p^i - \hat{T}_i^{13} p^i} - \Delta_x \quad (5.52)$$

et,

$$\hat{y}'' = \lambda_y \frac{x' \hat{T}_i^{32} p^i - \hat{T}_i^{12} p^i}{x' \hat{T}_i^{33} p^i - \hat{T}_i^{13} p^i} - \Delta_y \quad (5.53)$$

avec (λ_x, λ_y) les facteurs de correction d'échelle respectivement selon les axes x et y et avec (Δ_x, Δ_y) le facteur de correction de centrage respectivement selon les axes x et y . Pour déterminer les facteurs de correction de centrage nous utilisons un simple calcul de centre de gravité sur le nuage de points projetés dans la vue virtuelle :

$$\Delta_x = \frac{1}{\text{card}(\hat{p}'')} \sum_{i=1}^{\text{card}(\hat{p}'')} \hat{x}_i'' \quad (5.54)$$

et,

$$\Delta_y = \frac{1}{\text{card}(\hat{p}'')} \sum_{i=1}^{\text{card}(\hat{p}'')} \hat{y}_i'' \quad (5.55)$$

Pour les facteurs de correction d'échelle nous allons utiliser le rapport de la taille du nuage de points par rapport à la taille de l'image \hat{I}'' . Les facteurs de correction doivent intervenir seulement quand la taille de l'objet commence à sortir de l'image. C'est pourquoi nous avons deux valeurs pour le facteur d'échelle. La première est de valeur égale à 1, elle intervient quand l'objet ne dépasse pas de l'image c'est-à-dire lorsque l'extrapolation est faible. La deuxième valeur est appliquée seulement quand l'objet dépasse de la fenêtre pour de fortes extrapolations. Sa valeur est calculée de telle sorte que l'ensemble de l'objet tienne dans l'image. Les deux facteurs de correction d'échelle se calculent de cette manière :

$$\lambda_x = \begin{cases} 1, & \text{si } \| \max(\hat{x}'') - \min(\hat{x}'') \| \leq \text{card}_x(\hat{I}'') \\ \frac{\text{card}_x(\hat{I}'')}{\| \max(\hat{x}'') - \min(\hat{x}'') \|} & \text{si } \| \max(\hat{x}'') - \min(\hat{x}'') \| > \text{card}_x(\hat{I}'') \end{cases} \quad (5.56)$$

et,

$$\lambda_y = \begin{cases} 1, & \text{si } \| \max(\hat{y}'') - \min(\hat{y}'') \| \leq \text{card}_y(\hat{I}'') \\ \frac{\text{card}_y(\hat{I}'')}{\| \max(\hat{y}'') - \min(\hat{y}'') \|} & \text{si } \| \max(\hat{y}'') - \min(\hat{y}'') \| > \text{card}_y(\hat{I}'') \end{cases} \quad (5.57)$$

On applique les facteurs de correction à nos images exemple avec le même angle d'extrapolation. La Figure 5.15 montre la comparaison entre avant et après l'application des facteurs de correction sur le rendu. Il n'existe plus de problème de perte de cohésion. La géométrie est parfaite, et elle permet d'obtenir des angles d'extrapolation importants. Dans la littérature, il est très rare d'obtenir des résultats d'extrapolation de cette importance avec seulement deux vues et sans calibrage fort. Maintenant que la géométrie est cohérente, nous pouvons dans la prochaine section travailler sur l'aspect photométrique du rendu, ce qui signifie que la texture complète des vues réelles sera calculée dans la vue virtuelle. Mais avant, nous présentons un exemple de mise en application de notre méthode de reprojection.

Exemple Nous reprenons les paramètres du dernier exemple de ce chapitre, à savoir : le nouveau tenseur du système de configuration avec une vue virtuelle et les ensembles de points appariés des deux vues réelles. L'exemple du code proposé est écrit pour Matlab/Octave . Il consiste à calculer la position des points de contour de l'objet dans la vue virtuelle. Nous appliquons seulement la correction pour l'axe x pour diminuer la longueur du code, mais il est évidemment identique sur l'axe y en remplaçant seulement les abscisses par les ordonnées.

```
0000 % Calcul des points de la vue virtuelle
0001 for n=1:size(p2,2)
```

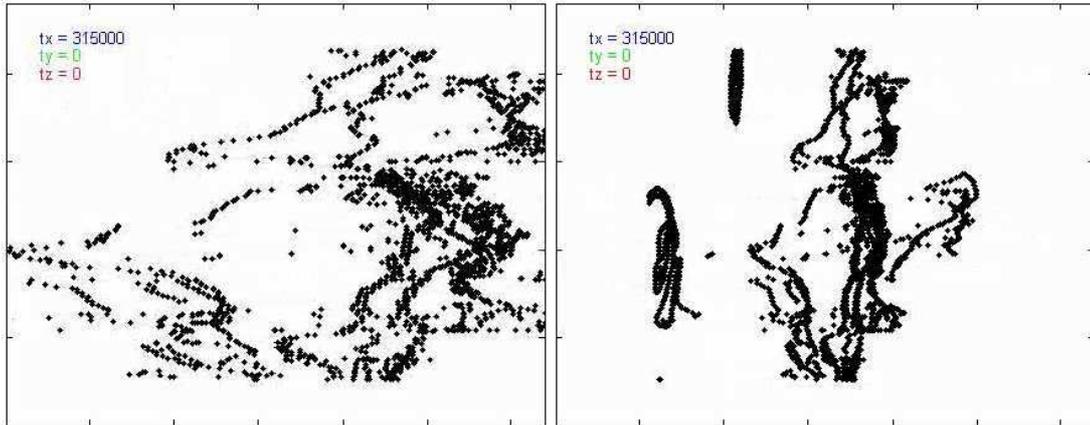


FIG. 5.15 – A gauche, les points de contours pour une vue à 75° sans facteur de correction, à droite, la même vue avec facteur de correction.

```

0002   p3(1,n)=(newT(1,:,1)*p1(:,n)-p2(1,n)*newT(1,:,3)*p1(:,n)) ...
0003           /(newT(3,:,1)*p1(:,n)-p2(1,n)*newT(3,:,3)*p1(:,n));
0004   p3(2,n)=(newT(2,:,1)*p1(:,n)-p2(1,n)*newT(2,:,3)*p1(:,n)) ...
0005           /(newT(3,:,1)*p1(:,n)-p2(1,n)*newT(3,:,3)*p1(:,n));
0006   p3(3,n)=1
0007 end
0008 % Correction du facteur d'échelle seulement sur x
0009 fe=max(p3(1,:))-min(p3(1,:));
0010 if fe>size(I3,2)
0011   p3(1,:)=p3(1,:)/(fe/size(I3,2));
0012 end
0013 % Correction du centrage seulement sur x
0014 p3(1,:)=p3(1,:)-mean(p3(1,:))+size(I3,2)/2;

```

avec, dans l'algorithme, newT le tenseur trilinéaire \hat{T} , p_1 , p_2 et p_3 les points respectifs de p , p' et \hat{p}'' , fe le facteur d'échelle et enfin I_3 l'image \hat{I}'' de la vue virtuelle. Au final on obtient l'ensemble des points \hat{p}'' de l'image virtuelle.

5.3.4.2 Forward mapping

La reprojection proposée précédemment est basée sur un rendu de type géométrique, il est partiel. En effet nous connaissons la position de chaque point dans l'image synthétisé mais à aucun moment nous ne nous sommes préoccupés de l'intensité à affecter à ces points : nous allons augmenter la complexité de notre rendu en incorporant la partie photométrique dans la partie géométrique. Pour cela nous devons considérer non plus les points de contour de l'image mais bien tous les points. Le résultat permettra d'obtenir une image texturée de la vue virtuelle. Dans le rendu propre au méthode de transfert il existe un procédé simple pour réaliser le rendu de tous les points. Il est identique à

celui utilisé pour calculer les points de contours dans la vue virtuelle, c'est la technique du *Forward mapping* (texturage avant). Cette technique consiste à calculer, pour chaque point des images réelles, la position du point dans l'image virtuelle puis à lui attribuer la valeur d'intensité des points des images réelles. L'intensité est calculée soit en prenant directement la valeur d'un point soit en combinant les valeurs des deux. Au final on prend la texture des images réelles et on l'applique point par point dans l'image virtuelle. Mais avant de pouvoir transférer tous les points dans notre vue virtuelle il faut connaître la mise en correspondance entre tous les points p et p' . Cette dense correspondance est la première difficulté de ce procédé. Avec les contours nous avons utilisé la méthode de Z. Zhang et al. (1995), qui consiste par une méthode de corrélation à chercher le point homologue sur la ligne épipolaire, permettant ainsi d'augmenter l'efficacité et la rapidité de la recherche. Seulement dans ce cas, cela serait trop long à calculer pour chaque point la ligne épipolaire correspondante. C'est pourquoi il existe une technique appelée rectification épipolaire, qu'on retrouve dans N. Ayache et C. Hansen (1988) et O. Faugeras et L. Robert (1993), qui consiste à projeter les deux vues stéréoscopiques dans un même plan. Par transformation des plans images, ils deviennent parallèles entre eux. Ainsi les faisceaux de lignes épipolaires concordent parfaitement entre les deux images. Cela implique que les lignes épipolaires sont parallèles entre elles et qu'un point quelconque a la même ordonnée dans les deux images. Le but de cette rectification est d'optimiser la mise en correspondance en cherchant l'homologue d'un point seulement le long d'une ligne horizontale. Au lieu de calculer chaque ligne épipolaire pour chaque point comme effectué par la méthode de Zhang, on rectifie directement la géométrie de l'image pour que ces lignes soient connues à l'avance. Il existe plusieurs méthodes de rectification épipolaire, nous en présentons une, qui est la plus connue et présentée par R. Hartley (1999).

Rectification épipolaire La méthode proposée par R. Hartley (1999) consiste à placer les épipoles à l'infini, permettant ainsi de forcer les plans images à devenir parallèles. Cette rectification est basée sur la géométrie épipolaire, donc on part du principe qu'elle est connue. La première étape consiste à placer le premier épipole v' de la vue ψ' à l'infini. Pour cela il faut préparer et positionner l'épipole en question correctement pour le placer à l'infini. Au départ, on place le centre de l'image sur son origine en vue de lui faire subir une rotation par rapport à elle même. On suppose que le point de projection du centre optique sur le plan image est au centre de l'image. Dans ce cas le changement de repère s'effectue par la translation suivante :

$$t = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{card_x(I')}{2} \\ 0 & 1 & -\frac{card_y(I')}{2} \\ 0 & 0 & 1 \end{bmatrix} \quad (5.58)$$

avec d_x et d_y le déplacement du repère selon x et y . Ce déplacement de repère va permettre d'effectuer des rotations par rapport au centre de l'image pour placer l'épipole sur l'axe des abscisses du plan image avant de la déplacer à l'infini. La projection de l'épipole v' sur cet axe est obtenue par une rotation autour de l'axe z du repère plan

image de I' , avec un sens anti-trigonométrique ou trigonométrique en fonction d'où il est positionné. On obtient l'angle de rotation en fonction de l'épipole :

$$\Psi = \tan^{-1} \left(\frac{v'^1}{v'^2} \right) \quad (5.59)$$

avec Ψ l'angle de rotation et $v' = (v'^1, v'^2, 1)^T$. Avec cet angle de rotation nous calculons la matrice de rotation autour de l'axe z du repère image :

$$R = \begin{bmatrix} \cos\Psi & \sin\Psi & 0 \\ -\sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.60)$$

Maintenant l'épipole est placé sur l'axe x du repère image, il ne reste plus qu'à le placer à l'infini. Les nouvelles coordonnées de l'épipole se calculent suivant la transformation :

$$\hat{v}' = Rtv' \quad (5.61)$$

Pour placer l'épipole \hat{v}' Hartley propose d'utiliser une approximation du premier ordre d'un Jacobien. Le but étant d'obtenir la valeur $\hat{v}'^3 = 0$ alors qu'elle est égale 1 (homogène). Cette approximation est exprimée par la matrice de transformation \mathcal{G} qui s'écrit :

$$\mathcal{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{\hat{v}'^1} & 0 & 1 \end{bmatrix} \quad (5.62)$$

A partir de là, nous connaissons la matrice de transformation projective H' à appliquer à l'image I' pour qu'elle soit rectifiée. Cela correspond à la succession des transformations calculées précédemment, de la façon suivante :

$$H' = \mathcal{G}Rt \quad (5.63)$$

Maintenant il faut forcer la première vue ψ à venir s'aligner par rapport à ψ' pour qu'elle soit dans le même plan parallèle. Une solution consiste à trouver la matrice de transformation projective H par minimisation des moindres carrés de la distance euclidienne entre les points des deux vues :

$$\sum_i \text{dist}_E(Hp_i, H'p'_i) \quad (5.64)$$

Sinon, on peut également transformer l'image I' puis définir des indices invariants que l'on vient rechercher dans l'image encore non transformée I . A partir de ces indices mis en correspondance, on calcule la matrice de colinéation entre les deux images et on l'utilise pour forcer I à se recalculer dans le même repère de I' . Au final, nous obtenons H et H' qui permettent la rectification épipolaire des deux vues ψ et ψ' .

Application Nous allons appliquer la rectification épipolaire à un couple d'images stéréoscopiques pour comprendre son utilité dans la dense correspondance. Afin de mettre en valeur les résultats nous allons choisir deux images différentes de celles utilisées jusqu'à maintenant. En effet, les images que nous utilisons depuis le début sont pratiquement déjà rectifiées du fait de leur configuration, ce qui n'est pas assez visuel pour illustrer la rectification épipolaire. Les images stéréoscopiques originales sont présentées à la Figure 5.16.



FIG. 5.16 – Couple d'images stéréoscopiques utilisé pour le test de rectification épipolaire.

En utilisant les transformations H et H' décrites précédemment nous transformons les deux images avec une méthode d'interpolation de type bicubique pour avoir un résultat de qualité. Les deux images rectifiées sont présentées dans la Figure 5.17. On peut vérifier, avec les lignes tracées dans la figure, que les deux images sont alignées. La première ligne passe par les yeux du personnage dans I et de même dans I' , la deuxième passe par le crochet dans les deux images et également pour la ligne passant par ses pieds. Bien évidemment il existe des erreurs d'approximation et d'estimation sur la rectification, mais cette étape est vraiment capitale pour la rapidité de la dense correspondance.

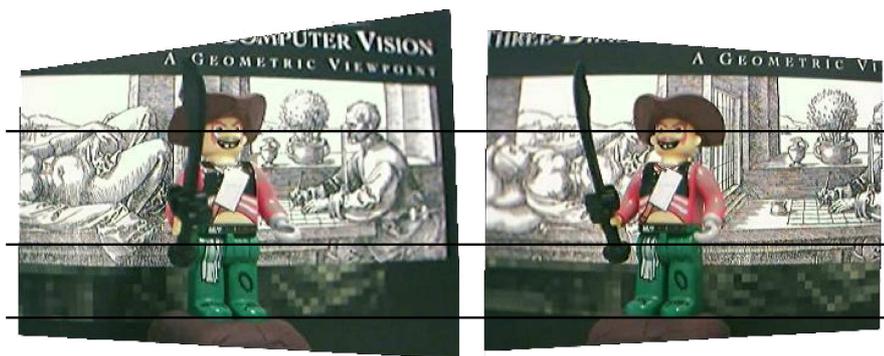


FIG. 5.17 – Couples d'images stéréoscopiques après rectification épipolaire.

Dense correspondance La rectification épipolaire appliquée aux deux images que nous utilisons depuis le début, nous permet de chercher efficacement chaque correspon-

dant dans chaque image. Il existe diverses techniques de dense correspondance, pour plus d'information il existe un comparatif de toutes les techniques présenté dans D. Scharstein et R. Szeliski (2002). Une étude complète sur la corrélation par stéréo vision est présentée par F. Devernay (1997). Notre méthode consiste à utiliser une méthode de corrélation de type *ZNSSD* ou *ZNCC* pour déterminer chaque correspondant. Nous utilisons également une méthode d'interpolation sur le signal du résidu de corrélation afin de déterminer les points correspondants avec une précision sub-pixélique. Un comparatif entre la disparité sans et avec interpolation de recherche est montré à la Figure 5.18. On remarque qu'il existe un gradient plus important avec une mise en correspondance sub-pixélique. Ce point est très important, en effet, dans le rendu si la correspondance est connue seulement au pixel près nous avons un phénomène de calque c'est-à-dire pendant la rotation de la vue, l'objet apparaît comme étant tronçonné en plusieurs couches de calque. Ce phénomène vient du fait que l'échantillonnage de la profondeur, existante dans l'information de la disparité est trop grossière. Si on connaît de façon sub-pixélique la mise en correspondance, le rendu de la vue affiche des points avec une plus grande variété de profondeur. Ce qui signifie que l'effet de couche de calque laisse la place à une continuité des surfaces dans la vue virtuelle.

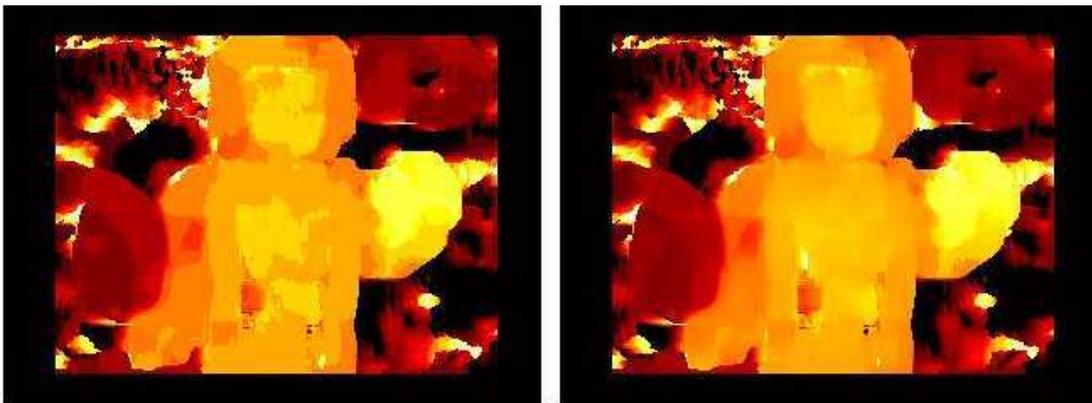


FIG. 5.18 – A gauche, la carte de disparité sans recherche sub-pixélique, à droite, la même carte mais avec une recherche sub-pixélique.

Z-Buffer Connaissant la correspondance entre chaque point, nous pouvons calculer la projection de tous les points de l'image dans la vue virtuelle. Une image résultat est présentée à la Figure 5.19-a. Cependant, comme on peut le constater sur cette figure, il existe un phénomène d'écrasement des pixels. Il y a plusieurs pixels transférés pour une même position d'un point dans l'image virtuelle. Si un pixel est déjà affecté à une position, et qu'un autre pixel est transféré sur cette même position, il écrase l'information précédente. Cela a un effet de masquage : des objets de la scène se placent au premier plan alors qu'ils devraient se trouver en arrière plan. Typiquement sur notre image résultat, le fond avec les poissons se trouvant en arrière du personnage, se trouvent devant lui, en le masquant sur certaines parties. Ce phénomène est provoqué par le manque d'information

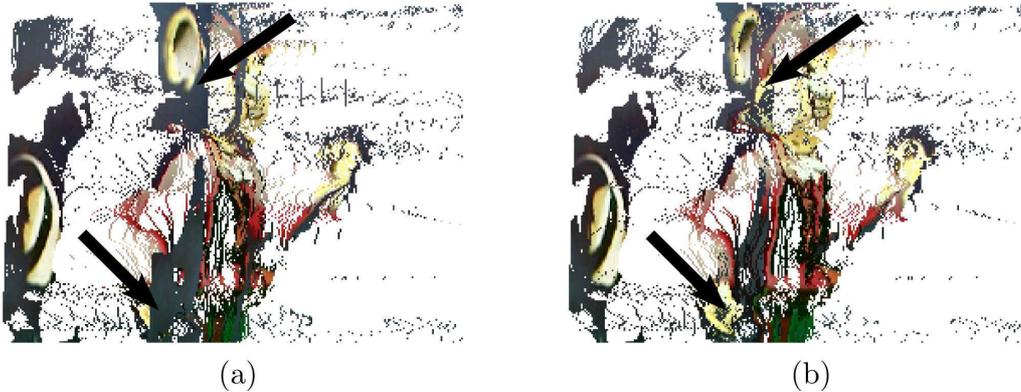


FIG. 5.19 – Rendu par *forward mapping* : (a) illustration du phénomène d'écrasement des pixels, (b) le même rendu avec une technique de *z-buffer*.

sur nos points reconstruits. En effet dans les méthodes de transfert nous n'utilisons pas d'information 3D explicite, donc on ne reconstruit pas la position des points dans l'espace, tout est projeté dans une image 2D. La technique pour palier ce problème est une des plus anciennes de l'infographie. Elle se nomme le *z-buffering* elle consiste à utiliser un *z-buffer* (tampon de profondeur). Dans le domaine du rendu de l'image de synthèse la problématique d'affichage des arêtes et des polygones cachés d'un objet a débuté il y a bien longtemps avec A. Appel (1968) et I. Sutherland, R. Sproull et R. Schumaker (1974). Le *z-buffer* présenté pendant la thèse de E. Catmull (1974), propose une solution en affectant une priorité de rendu sur chaque pixel. La priorité est affectée en fonction de sa profondeur dans la scène c'est-à-dire en plus de stocker l'information de couleurs de chaque pixel, il faut également stocker sa profondeur dans le *z-buffer*. Au moment du rendu on commence par calculer les plus éloignés jusqu'au plus proche, ainsi les points du premier plan vont écraser ceux en arrière plan. Bien évidemment il existe des méthodes d'architecture, d'optimisation et d'estimation des règles d'affichage dans un rendu. Dans ce domaine les pionniers ont été C. Jones (1971), J. Clark (1976) et D. Meagher (1982). Le *z-buffer* n'est pas la seule technique existante, plusieurs sont présentées sous forme d'état de l'art dans l'article de D. Cohen-Or et al. (2003). Aujourd'hui ce problème suscite toujours autant d'intérêt avec l'apparition du rendu temps réel.

Dans notre cas, nous allons utiliser le *z-buffer* pour palier l'écrasement des pixels reprojétés dans la vue virtuelle. Avant, il faut connaître la profondeur de chaque point. Or, dans notre cas, il est impossible de l'avoir, par contre nous connaissons sa valeur relative. Si nous connaissons la dense correspondance de nos images, nous connaissons également la carte de disparité de l'image. Et comme la disparité est en faite l'image de la profondeur du couple de points, nous pouvons l'utiliser dans le *z-buffer*. La méthode est simple, le rendu commence par calculer les couples de points avec la plus faible disparité jusqu'à la plus forte. La Figure 5.19-b représente le résultat avec la méthode du *z-buffering*. On remarque que les parties de l'arrière plan cette fois-ci sont effectivement placées derrière les objets du premier plan.

Résultat final En utilisant la dense correspondance filtrée des deux images exemples que nous utilisons depuis le début de ce chapitre, nous allons appliquer le transfert trifocal avec un rendu de type *forward mapping*. Le filtrage est obtenu avec la valeur du résidu normalisé de la corrélation pendant la correspondance, on seuille tous les points qui ne sont pas suffisamment ressemblants. La vue générée est le fruit d'un rendu avec l'ensemble des étapes présentées ici : dense correspondance, calcul de disparité, filtrage et rendu de type *forward mapping* avec *z-buffering*. La Figure 5.20 montre une des images originales ainsi que la vue virtuelle rendu pour un angle quelconque par une vue du côté gauche de l'objet ($t_x < 0$). Le filtrage a été très sélectif afin de ne pas avoir de point gênant à l'observation du rendu. C'est pourquoi une partie de la texture de l'image a été supprimée.



FIG. 5.20 – Rendu par *forward mapping* avec à gauche une des images originales et à droite le rendu final de la vue virtuelle.

L'inconvénient majeur d'un rendu de type *forward mapping* est la génération de trous dans l'image. Le phénomène est exactement identique pour toute transformation de l'image. Dans ce contexte, l'approximation des positions des points n'est pas la seule responsable de ce phénomène. En effet il est amplifié par tous les problèmes d'occultation. Prenons l'image originale d'une des vues de la figure précédente, on connaît très bien l'information photométrique du bras du personnage tendu devant lui par exemple. Au moment de la reconstruction, la technique *forward* consiste à calculer tous les points du bras de l'image, dans ce cas seulement la partie vue de devant. Donc les parties latérales de ce bras ne sont pas transférées car non texturées dans les images originales. Ce qui implique une zone non texturée dans la vue virtuelle car à aucun moment on n'a prévu de calculer cette zone. Comme dans la transformation d'image par interpolation la technique du rendu par *backward mapping* s'impose.

5.3.4.3 Backward mapping

A l'inverse du *forward*, le *backward* va calculer l'information à transférer pour chaque pixel de la vue virtuelle. Le *backward mapping* (texturage arrière) permet d'adapter le

texturage du rendu en fonction de la vue virtuelle. Dans la figure précédente, la méthode consistait à prendre les points du bras du personnage pour les transférer dans le rendu, ce qui a pour conséquence un manque de texture lorsqu'on observe ce bras de côté. Maintenant la technique consiste à prendre les points du bras observé de côté et de rechercher l'information dans les images originales. Bien évidemment même le *backward mapping* ne permet pas de supprimer les problèmes liés aux occultations, mais elle permet d'améliorer considérablement la qualité de l'image en supprimant les approximations dues au transfert. Il faut pour chaque point de la nouvelle vue retrouver les points correspondants dans les vues d'origine. La problématique liée à l'application du *backward mapping* est d'obtenir le transfert inverse avec le tenseur trilinéaire. La première étape pour inverser le transfert est décrite dans un papier présenté par Keith Connor et Ian Reid (2002). Il faut changer la vue de référence. Pour l'instant, le tenseur \hat{T} est exprimé pour une configuration $(\psi, \psi', \hat{\psi}'')$. Mais la nouvelle vue de travail est la vue $\hat{\psi}''$, car c'est à partir de ces points que nous allons retrouver les équivalents dans les vues ψ et ψ' par transfert. Il faut transformer le tenseur $\hat{T}(\psi, \psi', \hat{\psi}'')$ dans la nouvelle configuration $\hat{T}(\hat{\psi}'', \psi, \psi')$. Pour ce faire, nous allons utiliser une méthode proposée par S. Avidan et A. Shashua (1996) pour dériver la vue de référence dans le tenseur. Il existe d'autres méthodes dont une présentée dans R. Hartley et A. Zisserman (2000) que nous n'utiliserons pas pour des raisons de complexité. Leurs méthodes proposées permettent d'obtenir deux opérateurs, le premier O_{23} qui permet de passer d'une configuration $\mathcal{T}(1, 2, 3)$ à $\mathcal{T}(1, 3, 2)$ et le deuxième O_{12} qui permet de passer d'une configuration $\mathcal{T}(1, 2, 3)$ à $\mathcal{T}(2, 1, 3)$. Dans notre cas le tenseur doit subir les transformations suivantes :

$$\hat{T}_i^{jk}(\psi, \psi', \hat{\psi}'') \xrightarrow{O_{23}} \hat{T}_i^{jk}(\psi, \hat{\psi}'', \psi') \xrightarrow{O_{12}} \hat{T}_i^{jk}(\hat{\psi}'', \psi, \psi') \quad (5.65)$$

La première transformation utilise le premier opérateur O_{23} qui est obtenue directement par les coefficients du tenseur de départ, plus particulièrement par ces sous matrices \mathbf{l}_i de la façon suivante :

$$\hat{T}_i^{jk}(\psi, \hat{\psi}'', \psi') = O_{23} \hat{T}_i^{jk}(\psi, \psi', \hat{\psi}'') = O_{23} \{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3\} = \{\mathbf{l}_1^T, \mathbf{l}_2^T, \mathbf{l}_3^T\} \quad (5.66)$$

Ce n'est qu'une réorganisation des sous matrices \mathbf{l}_i par transposition. La transformation suivante, celle avec l'opérateur O_{12} s'obtient cette fois-ci avec les sous matrices \mathbf{K}_i comme suit :

$$\hat{T}_i^{jk}(\hat{\psi}'', \psi, \psi') = O_{12} \hat{T}_i^{jk}(\psi, \hat{\psi}'', \psi') = O_{12} \{\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3\} = \{\lambda_1 \mathbf{K}_1^{-1}, \lambda_2 \mathbf{K}_2^{-1}, \lambda_3 \mathbf{K}_3^{-1}\} \quad (5.67)$$

cette fois-ci il n'y a pas qu'une réorganisation, en effet les nouvelles sous matrices utilisées nécessitent une rectification par des facteurs d'échelle λ_i . Ces facteurs d'échelle peuvent être calculés à partir des sous matrices \mathbf{K}_i ainsi qu'en utilisant les points p et p' . Le premier facteur d'échelle λ_1 est de valeur 1 car il définit le facteur d'échelle globale par contre pour λ_2 il faut procéder comme suit :

$$\lambda_2 = \frac{\frac{([\mathbf{K}_1 p] \times p')^T ([p'] \times \mathbf{K}_2 p)}{\|[p'] \times \mathbf{K}_2 p\|^2}}{\frac{([\mathbf{K}_1^{-1} p'] \times p)^T ([p] \times \mathbf{K}_2^{-1} p')}{\|[p] \times \mathbf{K}_2^{-1} p'\|^2}} \quad (5.68)$$

On recommence l'opération pour le facteur d'échelle λ_3 avec les sous matrices \mathbf{K}_1 et \mathbf{K}_3 . A partir du nouveau tenseur $\hat{\mathcal{T}}(\hat{\psi}'', \psi, \psi')$ il existe différentes méthodes pour calculer la position du point dans les images d'origine. Keith Connor et Ian Reid (2002) proposent d'utiliser une matrice notée \mathcal{J} qui exprime une corrélation entre les deux vues ψ et ψ' d'origine. Pour chaque point \hat{p}'' de l'image \hat{I}'' , ils calculent cette matrice avec le tenseur associé :

$$\mathcal{J} = \hat{\mathcal{T}}_i^{jk}(\hat{\psi}'', \psi, \psi') \hat{p}''^k \quad (5.69)$$

A partir de cette matrice \mathcal{J} , ils en déduisent les deux lignes épipolaires l_{0i} et l'_{0j} des vues respectives ψ et ψ' . Les deux points recherchés dans les images originales existent sur ces deux lignes épipolaires. En utilisant la connaissance de la dense correspondance, ils utilisent les points correspondant à ligne épipolaire l_{0i} dans l'autre vue ψ' . Pour chacun de ces points ils vérifient que la relation entre chaque ligne épipolaire et la matrice \mathcal{J} est valide, pour déterminer si le point en cours est bien le point que l'on recherche. Si c'est le cas son intensité est transférée dans la vue virtuelle. Cette méthode est très contraignante, en effet elle nécessite la connaissance parfaite de la dense correspondance, sinon il est impossible de trouver avec précision les points que l'on recherche. En plus cette méthode revient à déterminer le croisement entre la ligne épipolaire l_{0i} et sa correspondante l'_{0i} dans la même vue. Or si dans la configuration les vues sont très proches, les lignes épipolaires sont pratiquement parallèles, ce qui augmente l'erreur de trouver le point d'intersection des deux lignes. Maintenant si les vues sont trop éloignées, elle mettent en péril la mise en correspondance obtenue par corrélation. La méthode de Keith Connor et Ian Reid (2002) est très fragile et peu robuste, c'est pourquoi les images résultats dans leurs travaux montrent seulement des nouvelles vues avec de faible extrapolation. Il existe une autre méthode de *backward mapping* pour le transfert trifocal, elle est présentée par H. Li et R. Hartley (2005). Elle est très similaire à celle de Connor, par contre elle ne nécessite pas de dense correspondance, en contre partie pour obtenir des résultats corrects il faut au moins 5 images réelles. De plus, leur résultat montre également des extrapolations de faible valeur. Pour gérer les occultations on retrouve également un papier de W.H. Lin, K. Sengupta et R. Sharma (2003) qui utilisent des reprojections par transfert trifocal des différentes segmentations (avant et arrière plan) des images originales. Un transfert par *backward mapping* ne présente pas que des avantages, par exemple dans les travaux de A. Fitzgibbon, Y. Wexler et A. Zisserman (2003), il est discuté de la génération d'écho dans la synthèse d'images. Le phénomène d'écho est la recopie répétitive de la texture dans les zones d'occultation. Reprenons l'exemple du bras de notre personnage dans la section précédente. Quand la méthode *backward* va chercher les points correspondants à la partie latérale du bras, elle va trouver les points originaux qui sont alignés avec le poing du personnage car vus de face. Par conséquent, l'information sera quand même transférée, et à la place d'obtenir la texture du bras sur

son côté nous allons avoir une texture du poing répété. Ce phénomène est normal car il est bien évident qu'il est impossible de reconstruire de l'information que l'on n'avait pas au départ. C'est pourquoi notre motivation nous a poussé à développer une nouvelle méthode, autre que le *backward*, qui intervient en post-processing, comme un deuxième rendu après le transfert par *forward mapping*.

5.3.4.4 Algorithme du *painting bridge*

Il existe différentes méthodes pour obtenir un rendu sans défaut. Elles ont pour but d'améliorer l'aspect photométrique de certaines parties de l'image obtenue par le transfert trifocal. Nous nous plaçons dans un contexte de rendu de type *forward mapping*. Nous savons que ce type de rendu engendre, deux problèmes, l'un étant l'écrasement des pixels et l'autre étant la génération de trous dans l'image.

La méthode du *z-buffering* permet de régler le problème de l'écrasement des pixels en balayant la carte de disparité. Si pour une raison quelconque la dense correspondance n'est pas connue, il existe également une technique utilisée appelée *modified painter's algorithm* (algorithme du peintre modifié) et présenté par L. McMillan et G. Bishop (1995a). Elle consiste à balayer les pixels de l'image pour les organiser comme dans le cas du *z-buffer* sauf que la valeur qui est utilisée n'est pas déterminée par la profondeur obtenue par une dense correspondance mais en utilisant directement la géométrie épipolaire.

Pour régler le problème de génération de trous il existe également différents algorithmes. Citons l'utilisation de filtre moyenneur appliqué au transfert trifocal (W.H. Lin, K. Sengupta et R. Sharma 2003). Un autre algorithme appelé *splatting* est présenté avec différentes variantes par M. Levoy et T. Whitted (1985) et N. Greene et P. Heckbert (1986). Son nom est tiré du mot anglais *splatter* (éclaboussure). Il découle de l'onomatopée anglaise que ferait une balle pleine de peinture lancée contre un mur, « splat ! ». Elle laisse une empreinte d'une certaine taille avec la couleur correspondant à la peinture que contenait la balle. C'est exactement le principe de la méthode de *splatting*. Pour chaque pixel de l'image originale qui est projeté dans l'image finale on lui associe une empreinte spéciale de différente taille (supérieur à un pixel) que l'on va appliquer dans l'image résultat. Donc chaque *splat*, l'élément projeté dans l'image, contribue dans plusieurs pixels de l'image finale. Ce qui permet de supprimer certains trous en les bouchant. Cette méthode est très intéressante car très rapide au moment du rendu. Différentes empreintes de *splat* peuvent être utilisées, par exemple dans le papier de N. Greene et P. Heckbert (1986) des formes elliptiques sont utilisées avec une couleur obtenue par pondération. Malgré tous ces avantages, le *splatting* ne permet pas de combler les grandes parties sans texture, en forte extrapolation, à moins de prendre des tailles d'empreintes importantes, qui risque alors de détériorer la qualité de l'image. En effet plus l'empreinte est grande, plus elle est efficace pour combler les trous, mais plus elle détériore l'image finale en créant des pâtés de couleur (pixélisation).

Nous avons développé un nouveau algorithme, appelé *painting bridge* (pont de peinture) pour résoudre ce problème de trous. Il intervient dans le rendu sous forme de *post-processing* c'est-à-dire après le premier rendu. Il consiste à repérer et à mémoriser

la taille et la position des trous dans l'image, puis une règle propre au *painting bridge*, que nous verrons plus tard, est appliquée pour combler les trous.

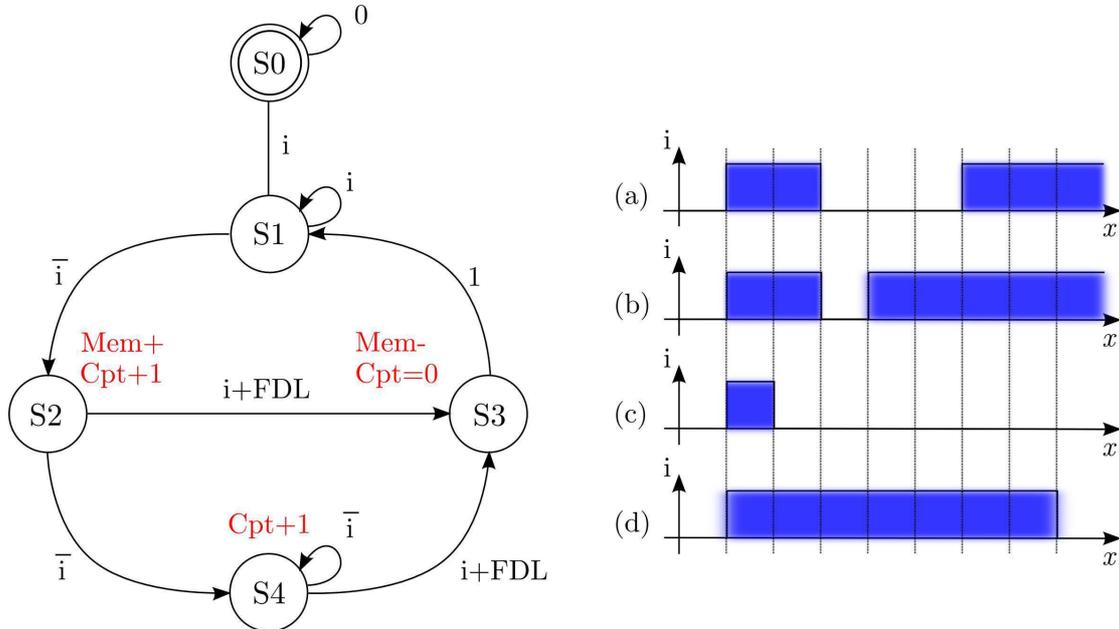


FIG. 5.21 – A gauche, machine à état de la détection des trous de la méthode du *painting bridge*, à droite, quelques exemples de profil de lignes de l'image contenant des trous.

La première étape consiste, pour chaque ligne de l'image résultat, à repérer les trous ainsi que la couleur des deux pixels de ses bords. Pour cela nous utilisons la machine à états présentée à la Figure 5.21. Prenons le profil (a) de la Figure 5.21, elle représente l'intensité de l'image contenant des trous pour une de ces lignes ($y = \text{cst}$). L'intensité i représente l'intensité d'un pixel utile, et l'intensité \bar{i} représente l'intensité du fond de l'image de travail (magenta pour notre cas). Dans la machine à états on part de l'étape initiale S_0 , on commence à balayer le profil de la ligne suivant x . Lorsque qu'on rencontre l'intensité i d'un premier bloc de pixel, nous entrons dans l'étape S_1 . Tant que l'intensité est égale à i nous restons dans cette étape. Pendant le balayage, un trou est détecté par \bar{i} , à ce moment l'étape S_2 est active. L'action de cette étape lance $Mem+$ qui correspond à la mémorisation de la position ainsi que l'intensité exacte du pixel définissant un bord de trou. Dans cette étape un compteur est incrémenté ($Cpt + 1$). Dans le cas du profil (a) le prochain pixel est toujours un trou, donc l'étape S_4 devient active, et continue d'incrémenter le compteur tant qu'il y a une intensité de pixel \bar{i} . L'étape S_2 a permis de détecter le front descendant de i pour la mémorisation du bord du trou. Après trois pixels, un nouveau bloc de pixels d'intensité i amène à l'étape S_3 . A ce moment, l'action consiste à mémoriser par $Mem-$ la position du pixel en cours, son intensité exacte et la valeur du compteur, qui correspond à la taille du trou, puis le compteur est remis à zéro. Ensuite on revient à l'étape S_1 pour un autre cycle. Maintenant, dans le cas où le trou ne fait qu'un pixel, ce qui correspond au profil (b), la détection du premier pixel

du trou nous place dans l'étape $S2$, comme il est de largeur 1, l'intensité suivante est de valeur i , dans ce cas on se retrouve directement dans l'étape $S3$. Dans le cas où le trou est si important qu'il va jusqu'à la fin de la ligne, comme le profil (c), alors, pendant que le compteur incrémente dans l'étape $S4$ nous arrivons à la fin de la ligne FDL . C'est pourquoi dans la condition du passage de l'étape $S4$ à $S3$ nous plaçons la variable fin de ligne FDL . On fait exactement la même chose pour la condition de l'étape $S2$ à $S3$ lorsque le trou ne fait qu'un pixel, comme le montre le profil (d). Au final on obtient pour chaque ligne de l'image la taille des trous ainsi que la position et l'intensité de chacun de ces bords.

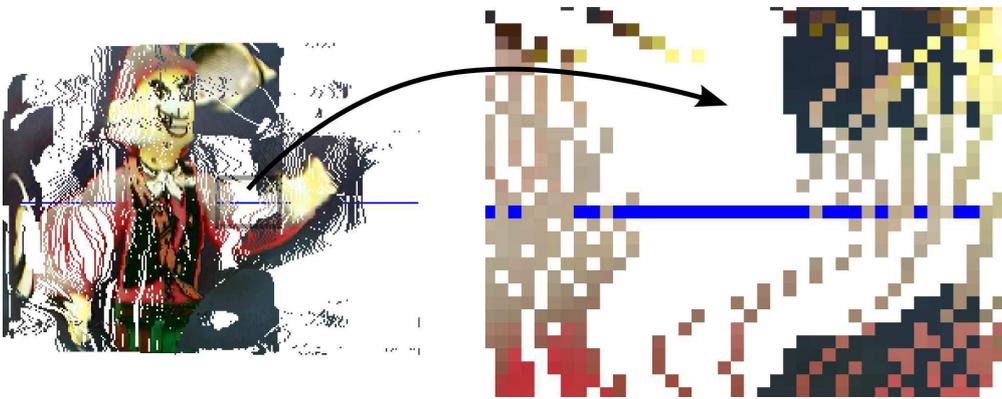


FIG. 5.22 – A gauche, représentation des trous détectés pour une ligne, à droite, le grossissement d'une partie de cette même ligne.

Connaissant les parties à combler, il ne reste plus qu'à définir les règles. Le *painting bridge* (pont de peinture) comme son nom l'indique, consiste à construire un pont photométrique entre les deux bords des trous détectés. Pour cela on utilise une règle qui définit la validité du pont en fonction de la distance des deux bords, le pont n'est construit que lorsque la taille du trou est inférieure à un seuil donné. Le but de la limitation de la longueur de la jonction entre les bords est d'éviter la fusion de deux objets distincts dans l'image. Le *painting bridge* doit corriger les problèmes du *forward mapping* ainsi que les zones d'occultation sans pour autant combler et fusionner la totalité de l'image. La Figure 5.23-a montre un exemple type de notre méthode. Soit deux lignes quelconques, avec deux trous de tailles 6 et 8 pixels. Après recherche par la machine à états on repère la dimension des trous ainsi que la position et l'intensité de chaque bord. Dans le cas de cette figure, le *painting bridge* est réglé pour créer des ponts d'une dimension égale ou inférieure à 6 pixels. Par conséquent, seul le trou de 6 pixels sera comblé (Figure 5.23-b). Pour rejoindre les deux bords du trou à combler on calcule pour chaque pixel du pont l'intensité qui correspond à une continuité photométrique.

Pour obtenir cette continuité nous utilisons la mémorisation de l'intensité des deux pixels du bord. Par choix, nous utilisons une variation linéaire entre l'intensité de chaque bord du trou, mais d'autres fonctions peuvent être utilisées, comme une fonction sig-

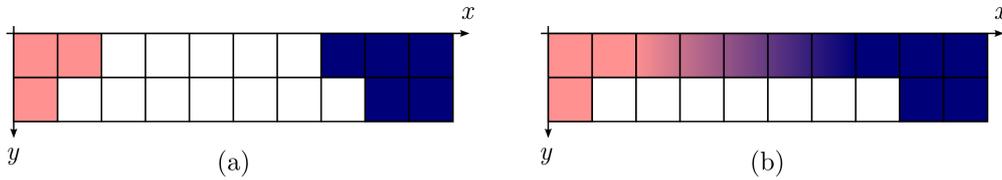


FIG. 5.23 – A gauche, deux lignes quelconques avec deux trous de tailles différentes, à droite, l'application sur ces lignes du *painting bridge* pour une longueur de pont limitée à 6 pixels.

moïde. Si le premier bord du trou a une intensité i_0 et le dernier bord une intensité i_n , le j -ième pixel à combler dans le trou aura l'intensité i_j suivante :

$$i_j = i_0 + j \left(\frac{i_n - i_0}{n} \right) \quad (5.70)$$

avec n l'index du dernier bord.

Application Pour l'application du *painting bridge* nous reprenons l'image rendue de la vue virtuelle de notre couple d'images test par *forward mapping*. Ce résultat, va subir un post-traitement pour corriger les problèmes de trous générés par l'approximation et l'occultation. Pour des raisons de rapidité de calcul sur le rendu en texture complète par *forward mapping*, et par la suite avec le *painting bridge*, nous avons choisi de ne pas utiliser le langage Matlab/Octave pour réaliser l'application que nous présentons ici. Ce langage interprété, parfait pour le prototypage rapide, est très mal adapté pour le rendu d'image, ce qui est devenu un frein pour obtenir des résultats rapides sur de petites machines.

Vu notre besoin de rapidité de prototypage nous nous sommes orientés vers un langage interprété multi-plateforme libre, facile de mise en main, d'une intuitivité proche de Matlab.

Les résultats que nous présentons ont été obtenus à partir d'une implémentation en langage Python sur une machine avec un OS (*Operating System*) GNU/Linux (Figure 5.24). Ce langage multi-plateforme a un code pseudo-compilé qui permet d'atteindre des temps de calcul plus que satisfaisant avec une rapidité de prototypage proche de Matlab/Octave. La première image de la Figure 5.24 correspond au rendu obtenu avec un *forward mapping* seul. On note bien l'existence de trous dûs aux approximations et aux occultations. La deuxième image montre l'application du *painting bridge* avec un seuil de taille de pont de 15 pixels. Les trous provenant d'approximation ont été corrigés, par contre certaines zones non texturées provenant de l'occultation restent non comblées car la taille des trous est supérieure au seuil du pont. Pour combler complètement le manque d'information il faut augmenter le seuil du pont. Un seuil de 100 pixels donne la dernière image de la Figure 5.24. On y constate alors un étalement de la texture, c'est le risque encouru lorsque l'on augmente trop le seuil.

Ces résultats permettent de valider l'ensemble de la chaîne du rendu de vue virtuelle par transfert trifocal. Maintenant nous allons le re-cadrer par rapport à notre contexte



FIG. 5.24 – A gauche, le rendu par *forward mapping*, au centre, une seconde passe de rendu par *painting bridge* avec un seuil de pont de 15 pixels, à droite, le même rendu mais avec un seuil de pont de 100 pixels.

de microscopie photonique, nous allons voir comment le mettre en application pour la micromanipulation.

5.4 Application en Micromanipulation

Nous allons utiliser le transfert trifocal pour libérer l'espace de travail dans une station de micromanipulation. En général le positionnement d'une micropince en vue de manipuler un micro-objet utilise deux vues, une vue de dessus pour contrôler le placement sur les axes x et y et une vue de côté pour contrôler la position de la pince sur l'axe z (Figure 5.25-a). Nos recherches permettent de remplacer cette configuration classique pour une configuration stéréo vision où les deux imageurs sont frontales. À partir des deux vues réelles frontales nous allons utiliser le transfert trifocal pour reconstruire la vue virtuelle de côté qui permettra le contrôle de la pince sur l'axe z (Figure 5.25-b). Les deux imageurs peuvent être regroupés en un seul système d'images en utilisant la fonction stéréoscopique de notre microscope photonique. Le fait de reconstruire l'image d'une vue qui n'existe plus physiquement, car l'imageur de côté a été déplacé, nous permet de libérer l'espace de travail pour placer des éléments plus importants dans la station, comme d'autres micropinces. Nous montrerons la validité de la reconstruction de la vue côté en mettant en place un système d'asservissement visuel simple pour contrôler la pince via la vue virtuelle. Le procédé de reconstruction a été validé sur différentes plates formes de micromanipulation, dont celles de l'équipe SAMMI comprenant un stéréo microscope photonique de type Leica. Pour des raisons de disponibilité nous avons monté notre propre manipulation comprenant deux mini-microscopes photoniques de type TIMM montés dans une configuration stéréoscopique. Nous avons nos tables de positionnement qui nous permettent le contrôle d'une micropince. Les deux arguments de la limite d'utilisation du transfert trifocal se trouvent dans la mise en correspondance et le calibrage. La nécessité de connaître la correspondance entre les points des vues réelles, obtenue par corrélation, implique pour des résultats optimaux, un faible angle entre les vues gauche et droite. Par exemple sur le stéréo microscope photonique Leica il est de 10° , pour les TIMM montés en stéréo il atteint 30° . La deuxième limitation, est le

calibrage faible stéréoscopique, il faut avoir la capacité de le calculer, pour ce faire nous avons utilisé les méthodes présentées dans le chapitre 3.

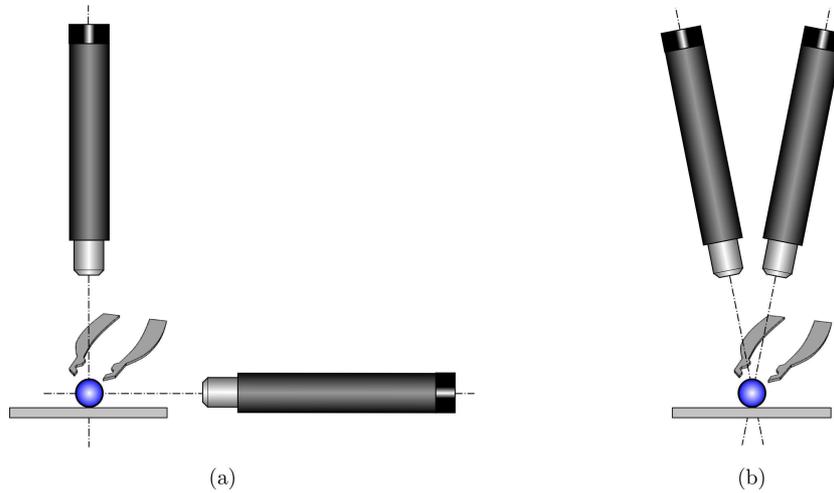


FIG. 5.25 – Configuration du système de vision d’une station de micromanipulation : (a) configuration classique, (b) configuration stéréo vision.

5.4.1 Vue de côté

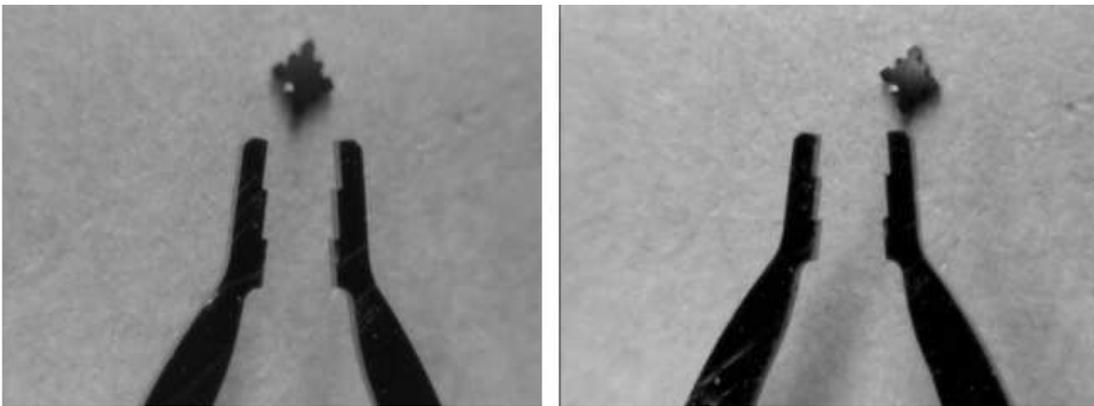


FIG. 5.26 – Un exemple de couple d’images obtenue à partir de notre système de vision.

L’objectif de la création de la vue virtuelle de côté est de contrôler la position de la micropince. Nous n’avons pas besoin de la texture complète de la pince pour déterminer sa position dans l’image. Par conséquent, nous allons réduire le nombre d’informations à transférer dans la vue virtuelle : seuls les points de contour seront utilisés dans le rendu. Un exemple de couple d’images que l’on peut obtenir à partir de notre système de vision est présenté à la Figure 5.26. La scène est constituée d’une micropince et d’une

micropièce carré de $400 \mu m$ de côté. La pince se trouve à environ $500 \mu m$ au dessus de la pièce. Après un calibrage du système, on utilise un détecteur de contours de type Canny pour avoir les points de contours de l'image de gauche. On effectue une correspondance par III-ZNSSD avec une recherche sub-pixélique pour déterminer les points homologues du contour de l'image de droite. On amorce le tenseur trilinéaire à partir des deux vues. Par reprojection contrôlée on cherche la vue de côté par minimisation d'un objet factice. La valeur du vecteur de translation obtenue est mis en mémoire, pour être réutilisée plus tard. On utilise un rendu de type *forward mapping* simple à partir des points de contour. On montre, à la Figure 5.27, des images résultats pour plusieurs valeurs d'angles d'extrapolation. Sur les vues de côté purement virtuelles nous observons bien que la micropinçe et la micropièce ne sont pas dans le même plan : la distance en pixels entre la micropinçe et la micropièce correspond à la position de la pince sur l'axe z . Il ne reste plus qu'à quantifier cette distance de telle sorte à asservir la micropinçe en position de capture de l'objet.

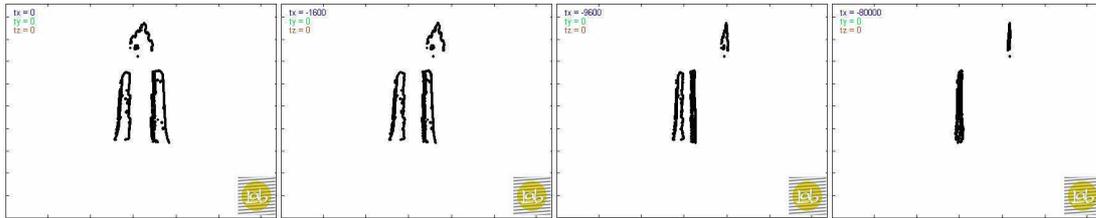


FIG. 5.27 – Vue virtuelle de la micropinçe pour différents angles de vue entre la vue de face et la vue de côtés.

5.4.2 Mesure de distance

Par choix nous voulons que la pince se positionne automatiquement au même niveau que la micropièce dans le but de la saisir. La première chose est de placer la vue virtuelle à un angle d'extrapolation proche de 90° pour être sûr d'observer la vue de côté. Ensuite il faut déterminer la distance en pixel qui sépare les deux objets, micropièce et micropinçe. Pour cela nous allons utiliser une technique mise en pratique précédemment pour estimer un objet factice dans l'image. Soient p_c les points de contour rendus dans l'image virtuelle. Nous calculons le centre de gravité du nuage de points, nous obtenons les coordonnées (\bar{x}, \bar{y}) . Nous séparons les points en deux groupes par le biais de ce centre, ceux qui sont à gauche et ceux qui sont à droite. En fonction de la vue virtuelle générée, côté gauche ou côté droit, chaque ensemble correspond soit à la micropièce soit à la micropinçe. En mesurant la distance euclidienne entre les centres de gravité de chaque sous-ensemble nous déterminons la distance en pixel entre la micropièce et la micropinçe. Dans notre cas nous avons une distance de 686 pixels, représentée dans la Figure 5.28. La distance réelle mesurée par les tables de positionnement nous donne une distance métrique de $424 \mu m$. Ce qui nous fait un rapport métrique/pixel de $618 nm/pixel$ dans la vue virtuelle. Or nous avons un rapport métrique/pixel de $11 \mu m/pixel$ dans les vues réelles de dessus.

Cela signifie qu'une utilisation d'images stéréoscopique avec un transfert trifocal permet d'appréhender la profondeur avec une résolution supérieure.

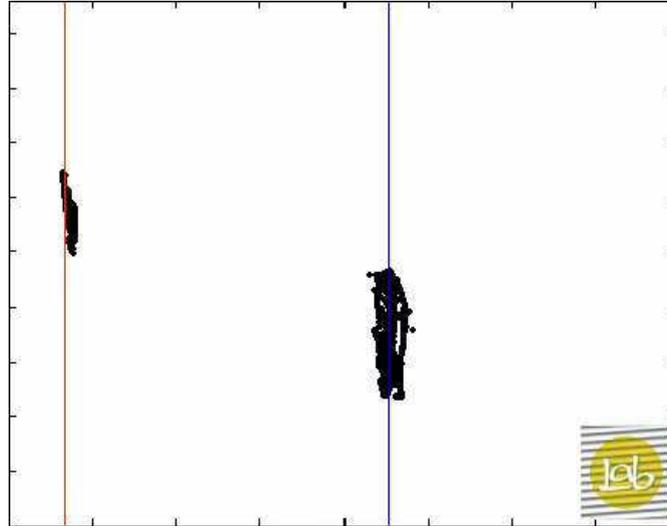


FIG. 5.28 – Vue de côté virtuelle de la micromanipulation, avec la position de la micro-pièce à gauche et la position de la micropinçe à droite.

L'augmentation de la résolution vient de la mise en correspondance entre les points des deux images réelles. L'évolution des points projetés dans la vue virtuelle est déterminée par le transfert trifocal à partir des points des images réelles. En fonction de la géométrie trifocale et de l'information de déplacement entre les points des images réelles les points rendus dans l'image virtuelle seront placés à différents endroits. C'est donc la disparité entre les points réels qui donne la résolution des points de la vue virtuelle. Étant donné que la mise en correspondance a été réalisée par une corrélation avec une interpolation par 20, il est donc normal de se retrouver avec une résolution plus importante.

5.4.3 Asservissement en position

Notre dernière étape consiste à créer une boucle d'asservissement pour que la micropinçe vienne se positionner à la même altitude que la micro-pièce pour la saisir. Pour cela nous allons utiliser un asservissement simple de type *look-and-move*. La philosophie de ce type d'asservissement consiste à mesurer par l'observation (*look*) puis à se déplacer (*move*). À chaque itération on mesure la position entre la micropinçe et la micro-pièce dans la vue virtuelle puis on déplace la pince dans l'espace. Il faut plusieurs itérations pour que la micropinçe converge vers son point désiré. Nous n'utilisons qu'une boucle d'asservissement sur l'axe z dans la vue virtuelle, car notre objectif est de montrer la faisabilité d'une commande par le biais d'un imageur virtuel. Pour mettre en pratique cela nous devons choisir la consigne de position que doit atteindre la micropinçe dans l'image. Il est impossible de sélectionner une consigne qui consiste à placer la micropinçe

et la micro-pièce de façon colinéaire dans la vue virtuelle car les points utilisés dans les images réelles sont les contours en vue de dessus, ce qui signifie que ce n'est que les parties supérieures des objets qui sont reprojctées dans la vue de côté. Or la micropince a une épaisseur de $280 \mu m$ et la micro-pièce une épaisseur de $100 \mu m$. Il faut donc intégrer un *offset* sur la consigne de position de la micropince pour qu'elle soit exactement au même niveau que la micro-pièce à saisir. Pour un rendu rapide nous utilisons seulement un point sur dix des contours des objets.

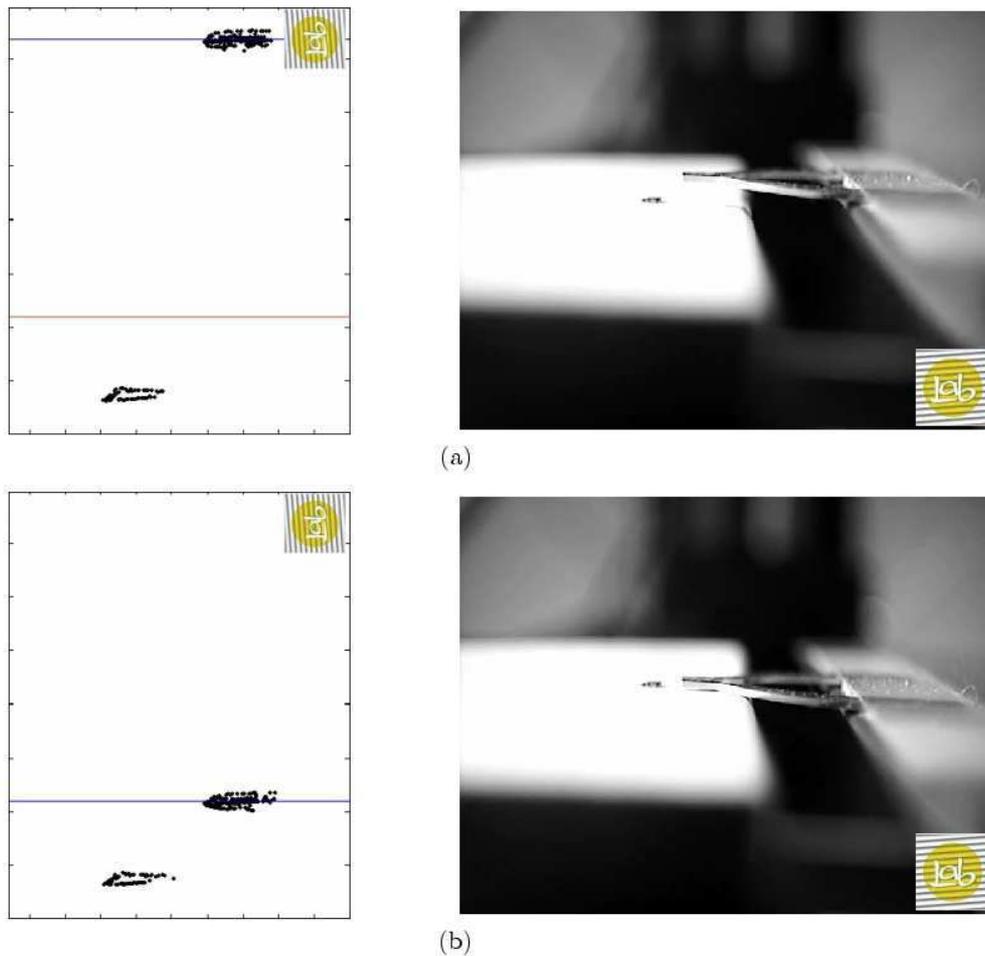


FIG. 5.29 – Asservissement visuel par une vue virtuelle : (a) vue virtuelle avec la pince en position initiale et la vérification sur la vue réelle correspondante, (b) vue virtuelle avec la pince en position finale et la vérification sur la vue réelle correspondante.

Le résultat de l'asservissement visuel par la vue virtuelle est représenté à deux moments à la Figure 5.29. La Figure 5.29-a montre le début de l'asservissement : la position courante de la micropince est représentée par un trait horizontal. On retrouve également un trait horizontal proche de la micropièce avec un *offset*, il correspond à la position de consigne que l'on désire atteindre. Une vue supplémentaire est fournie dans cet exem-

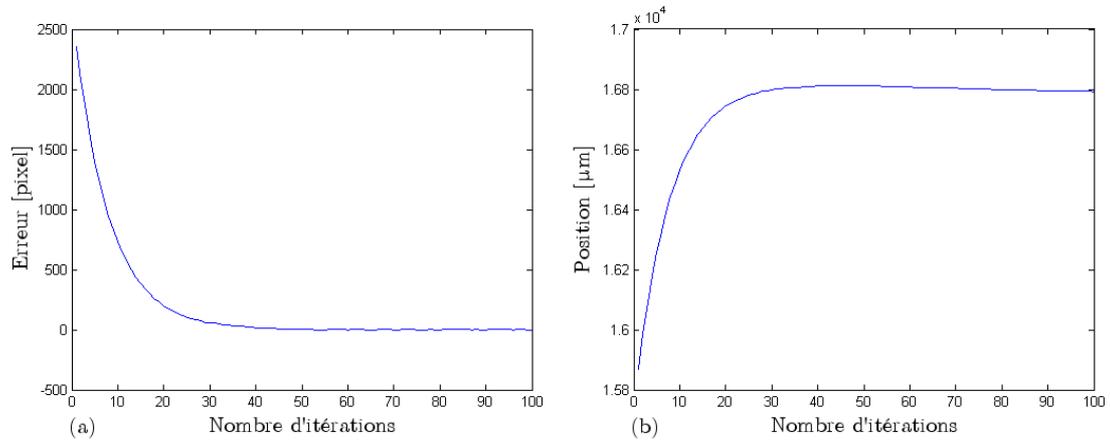


FIG. 5.30 – Asservissement visuel par une vue virtuelle : (a) erreur de position dans la vue virtuelle en pixel pendant l’asservissement, (b) position de la micropince sur l’axe z en μm pendant l’asservissement.

ple, c’est la vue réelle de côté. Elle est utilisée simplement pour vérifier le déplacement réel de la micropince. Pendant l’asservissement nous avons limité le déplacement de la micropince avec un coefficient de gain inférieur à 1 pour minimiser tout risque de dépassement de la consigne qui serait destructif pour la micropince. Après plusieurs itérations, la position de la micropince converge vers la position désirée. La Figure 5.29-b montre la fin de l’asservissement. La droite horizontale qui symbolise la position de la micropince est confondue avec le trait vertical de consigne. La vue de côté réelle, utilisée pour vérification, montre bien que la micropince est venue se placer sur le plateau de l’espace de travail à la même hauteur que la micropièce à saisir. Pendant l’asservissement visuel nous mesurons pour chaque itération l’erreur de position en pixel dans la vue virtuelle entre la micropince et la micropièce, elle est représentée par le graphique dans la Figure 5.30-a. Ce graphique montre la convergence progressive de l’erreur par une exponentielle décroissante jusqu’à la valeur 0. Cela montre que la micropince tend progressivement vers la position désirée. Nous avons également représenté l’évolution de la position de la micropince sur l’axe z par le biais de la table de positionnement, le graphique se trouve à la Figure 5.30-b. Ce graphique montre qu’il y a un très léger dépassement, le réglage de l’asservissement doit être très bien effectué car un dépassement trop important peut être destructif pour la micropince. Au final la micropince converge vers la position désirée.

Notre application montre bien la faisabilité du transfert trifocal dans le contrôle de station de micromanipulation. En utilisant des vues virtuelles nous sommes en capacité d’extraire de l’information pour contrôler le positionnement des micropinces. Nous avons utilisé un asservissement visuel en s’appuyant sur un microscope photonique virtuel. De plus cette vue virtuelle procure une résolution plus importante que les vues de dessus. A partir seulement d’un stéréo microscope photonique nous sommes dans la capacité de reconstruire tout un ensemble de vues virtuelles positionnées dans un domaine atteignable

qui représente une demi sphère. Cela permet de libérer l'espace de travail en supprimant des vues physiques pour les remplacer par des vues virtuelles.

5.5 Conclusion

L'encombrement des stations de micromanipulation est un problème important. Une réflexion poussée sur l'architecture matériel doit être entreprise. En effet les micropinces utilisées pour la micromanipulation et le micro-assemblage ont besoin de plusieurs degrés de liberté et plusieurs actionneurs doivent être placés dans la station. Si nous rajoutons en plus le système de vision qui doit donner toutes les vues nécessaires au contrôle et à la supervision de la station, nous avons un sur-encombrement de l'espace limitant la flexibilité et la reconfigurabilité de la station. Qui plus est les imageurs utilisés dans ce type de système de vision sont composés de microscopes photoniques dont l'encombrement est très important. Nous proposons une solution globale qui permet de reconstruire un maximum de vues de la station avec un minimum de microscopes photoniques. Pour cela, nous nous basons sur un système de vision composé d'un seul microscope photonique stéréoscopique. Les deux vues réelles sont utilisées pour synthétiser une vue virtuelle par transfert trifocal, méthode du rendu à partir d'images. Nous avons éclairci la théorie du transfert trifocal et développé une nouvelle méthode de placement de la vue virtuelle convenant à notre contexte de microscopie photonique. Afin d'optimiser la qualité du rendu de la vue nous incorporons des facteurs de correction dans la reprojection des points dans la vue virtuelle pour permettre un rendu de vue à forte extrapolation. Dans la littérature sur le transfert trifocal nous figurons comme les seuls capables d'obtenir des extrapolations qui frôlent les 90° sans le calibrage fort du système de vision. Le domaine atteignable du microscope photonique virtuelle ainsi créé se trouve dans une demi-sphère autour des imageurs réels (l'imageur peut se placer entre $] - 90^\circ; 90^\circ[$ selon la rotation autour de l'axe X et Y du centre d'extrapolation). Les techniques classiques de *forward mapping* engendrent des trous dans la texture de la vue virtuelle et les techniques classiques qui permettent de les corriger par *backward mapping* ont des contraintes telles qu'en pratique elles sont inutilisables. C'est pourquoi nous proposons une méthode de rendu en deux passes, la première comprend un rendu *forward mapping*, puis la seconde comprend l'application de notre algorithme de *painting bridge*. Il permet de corriger le phénomène de trous dans la texture de la vue virtuelle. Il est possible de l'appliquer verticalement ou horizontalement en fonction de l'extrapolation de la position de la vue virtuelle. Nous avons considéré comme un point important à l'obtention d'une texture complète de la vue reconstruite, car nous voulons proposer une méthode globale c'est-à-dire une vue telle qu'elle serait si le microscope photonique avait été réel, après libre à l'utilisateur de la traiter en fonction de l'information désirée, détection de contour, centre de gravité, segmentation, etc. Cela permet de l'utiliser pour différentes applications aussi bien pour de la supervision que pour de la commande référencée vision. C'est justement cette dernière que nous avons mise en pratique afin de montrer la faisabilité d'un asservissement d'une micropince par le biais d'une vue provenant d'un microscope photonique qui n'existe pas physiquement. Les résultats de cette application nous mon-

trent que notre méthode permet le contrôle de la station de micromanipulation avec une grande précision. Avec un seul stéréo microscope photonique nous sommes capables de positionner une ou plusieurs micropinces dans l'espace de travail, diminuant ainsi considérablement le nombre d'imageurs du système de vision et par conséquent l'encombrement. L'ensemble des travaux dans ce chapitre a abouti à un logiciel permettant de reconstruire une vue virtuelle texturée à partir de deux images et de son calibrage stéréoscopique faible. L'extrapolation de la vue complète est modifiable sur toutes les positions possibles en temps réel, permettant de se déplacer virtuellement autour de l'objet. Le logiciel TTR (*Trifocal Transfer Rendering*) a été intégré et développé en C++ avec les bibliothèques libres OpenCV et OpenGL. OpenCV nous permet d'utiliser un Lucas-Kanade pour obtenir la dense correspondance et la bibliothèque OpenGL nous permet de bénéficier de sa rapidité de rendu avec une méthode de *z-buffering*. Une capture d'écran du logiciel TTR est présentée à la Figure 5.31.



FIG. 5.31 – Le logiciel TTR *Trifocal Transfer Rendering*.

Conclusions et Perspectives

Ce chapitre présente la conclusion générale de l'ensemble des travaux effectués pendant la thèse. Il présente les apports dans chaque thématique ainsi que les perspectives à court et long termes.

6.1 Conclusion Générale

La production de produits à base de micro composants nécessite de maîtriser les tâches de micromanipulation et de micro-assemblage. Ces tâches sont réalisées dans des stations de micromanipulation qui constitueront, dans un futur proche, des systèmes automatisés de production. Comme dans tout système automatisé, une station de micromanipulation a besoin d'actionneurs et de capteurs. A l'heure actuelle, l'imageur principal (capteur+optique) utilisé pour sa résolution d'observation, est le vidéo microscope photonique. Toute personne ayant déjà manipulé des microproduits par observation avec un microscope photonique ressent les contraintes liées à cet outil. Cette thèse a permis dans un premier temps d'identifier clairement ces contraintes dans le contexte de micromanipulation qui nécessitent des placements de microproduits dans un espace 3D, ce qui est très différent des applications 2D de la biologie. Trois contraintes nous semblent très importantes, elles sont liées aux propriétés du microscope photonique : la faible profondeur de champ, le faible champ de vision et l'encombrement. Les algorithmes de vision classiques dans le but de contrôler et de surveiller une station de micromanipulation ne permettent pas de résoudre ces contraintes. Pour compenser ces contraintes nous sommes obligés d'ajouter n vues, c'est-à-dire n microscopes photoniques ou caméras. Il est bien évident que ce n'est pas une solution en soit. Car rajouter des vues dans le système de vision implique une diminution de l'espace de travail, une augmentation des coûts et une diminution de la flexibilité de la station. Parfois il est même impossible de rajouter des imageurs dans la station du fait de la faible distance de travail et de l'encombrement d'un microscope photonique. C'est pourquoi cette thèse a apporté des solutions de reconstruction, dans une philosophie qui consiste à produire les n vues nécessaires avec un minimum de microscopes photoniques dans la station.

Pour cela nous nous sommes appuyés sur un domaine de la vision artificielle dédiée à la reconstruction d'images 2D.

Cette thèse propose d'améliorer la perception visuelle d'une station de micromanipulation par des outils de vision avancée. Mais pour utiliser ces outils nous avons dû mettre en oeuvre le calibrage de tous les imageurs. Nous avons proposé un protocole d'auto stéréo calibrage faible avec de nouveaux algorithmes et de nouvelles méthodes pour améliorer le processus :

- mire réalisée par goutte d'eau recouverte de limaille de nickel,
- correspondance de points par couche de profondeur,
- détecteur Harris simplexe,
- corrélation par III-ZNSSD.

Par le biais de ce calibrage le premier outil de vision avancée, fut la construction de mosaïques (*mosaicing*). Cet outil nous a permis de reconstruire un microscope photonique virtuel qui a les propriétés d'un microscope photonique de grande résolution avec un grand champ de vision. Cet imageur virtuel permet d'observer tout l'espace de travail de la station avec la meilleure résolution possible. Nous avons proposé des méthodes d'incrustation des vues en temps réel des zones de l'espace de travail qui doivent être surveillées. Nous n'avons pas simplement utilisé l'outil de construction de mosaïques mais nous avons également contribué à l'amélioration du processus avec les apports suivants :

- estimation robuste du rapport métrique/pixel,
- construction de mosaïque à partir d'images à faible texture,
- détecteur Harris simplexe,
- corrélation par III-ZNSSD,
- fusion adaptative,
- détournage automatique,
- supervision de station de micromanipulation.

Nous avons présenté le moyen de pallier la première contrainte d'un microscope photonique en contexte de micromanipulation, le faible champ de vision. Les résultats ont été validés en expérimentation sur une station de micromanipulation factice prévue à cet effet. Les expérimentations ont permis de valider notre capacité de la supervision et du contrôle de station de micromanipulation. Une partie de la supervision de cette étude a été intégrée et implémentée dans le logiciel *MosaicingRTI*.

Cette thèse a permis également de présenter un outil très peu utilisé dans le domaine de la vision. Cet outil, le transfert trifocal, permet de reconstruire une vue virtuelle à partir de seulement de deux vues réelles sans reconstruction 3D explicite. En utilisant un microscope photonique stéréoscopique, ou deux microscopes photoniques montés en stéréoscopie, nous avons spécifié l'ensemble des étapes nécessaires pour synthétiser un microscope photonique virtuel. La vue virtuelle obtenue permet de nous fournir des images

provenant d'un imageur qui n'existe pas physiquement. Nous avons proposé un nouveau moyen de placement de la vue virtuelle par calibrage faible pour que le transfert trifocal soit adaptable dans notre contexte de travail. Dans une station de micromanipulation, cette méthode permet d'alléger le système de vision, car on reconstruit une multitude de vues virtuelles à partir de seulement deux vues réelles. En plus si pour des raisons d'encombrement il est impossible de placer un microscope photonique à un endroit précis de la station, le transfert trifocal permet d'accéder à des vues dans des zones inaccessibles. Notre travail a permis d'obtenir des résultats de qualité par cette méthode de transfert. Pour cela nous avons amélioré des techniques existantes et développé des nouvelles ; notre apport se résume en quelques points :

- éclaircissement de la théorie,
- placement de la vue virtuelle sans calibrage fort,
- placement automatique de vue virtuelle par objet factice,
- contrôle de reprojection en forte extrapolation,
- rendu par *painting bridge*,
- asservissement visuel dans la vue virtuelle.

Cette partie de la thèse consacrée au transfert trifocal, permet de compenser les problèmes d'encombrement des microscopes photoniques. En effet, dans la station de micromanipulation le nombre d'imageurs est réduit au minimum. Tous les résultats théoriques ont été présentés avec des images quelconques pour des raisons d'illustration plus explicites. Dans la littérature nous sommes les seuls à fournir des résultats en forte extrapolation par un placement de la vue sans calibrage fort et seulement à partir de deux images. Dans cette thèse nous avons présenté également l'application concrète de notre procédé dans le contexte de micromanipulation. Nous avons montré la capacité de nos méthodes à contrôler une micropince en position par asservissement visuel avec une vue virtuelle générée par le transfert trifocal à partir de deux vues de microscopes photoniques réels. Notre travail sur le transfert trifocal s'inscrit dans les travaux de la littérature, pour la plupart théorique, comme une application concrète de l'utilisation du transfert trifocal dans le domaine de la robotique en général. Une partie du procédé de reconstruction a été intégré et implémenté dans le logiciel TTR.

Tout au long de cette thèse nous avons identifié les problèmes liés aux systèmes d'imagerie photonique en micromanipulation. Puis nous les avons résolus en utilisant des outils de vision avancée. A chaque fois nous nous sommes efforcés de les présenter de manière claire et illustrée avec des exemples soit sous forme d'algorithmes, soit par du code, pour que cela soit exploitable et ré-utilisable. Nos méthodes et techniques ont toutes été validées sur des scènes concrètes de micromanipulation. Dans chaque partie, calibrage, construction de mosaïques et transfert trifocal nous n'avons pas simplement mis en application des techniques existantes dans un contexte de microscopie, nous avons également contribué directement à la vision artificielle avec l'amélioration de ces techniques qui sont par ailleurs ré-utilisables dans d'autres contextes que la micromanipulation, l'imagerie bio-médicale notamment.

6.2 Perspectives à court terme

Beaucoup de techniques présentées dans cette thèse sont encore à travailler, découvrir ou améliorer. Nous présentons ici, les premières perspectives qui seront la suite logique de cette thèse. Nous commençons par la présentation de deux nouvelles pistes dans le domaine de la synthèse de nouvelles vues qui pourraient être appliquées dans le contexte de microscopie. Nous terminerons avec les perspectives propres aux techniques présentées dans la thèse.

6.2.1 Synthèse de nouvelles vues par EDF

Cette technique permettrait de palier les problèmes liés à la faible profondeur de champ. La méthode *Extended Depth of Field* (EDF) d'extension de la profondeur de champ n'est pas à confondre avec la méthode de *depth-from-focus* qui elle est une méthode de reconstruction 3D par focus déjà exploitée commercialement. L'EDF permet de reconstruire une image 2D complètement nette de la scène à partir des images de différentes couches de profondeur de champ. On retrouve une méthode présentée par B. Forster et al. (2004) et A. Bradley et P. Bamford (2004), qui consiste à utiliser une décomposition en ondelettes sur toutes les images, puis à fusionner les coefficients qui correspondent aux parties nettes des images et par décomposition inverse, retrouver une image contenant toutes les parties nettes de chaque image. Cette méthode est facile d'utilisation et d'implémentation. Nous proposons un exemple à la Figure 6.1.

Ce procédé peut être amélioré en utilisant des stratégies différentes pour la sélection des coefficients utiles à la reconstruction d'une netteté parfaite. L'inconvénient est l'acquisition d'images à différentes profondeurs de champ qui rend le procédé assez lent. C'est pourquoi ce n'est pas encore une technique très utilisée car la dynamique du système optique est trop lent pour être exploité dans du contrôle de micromanipulation. Mais de nouveaux moyens optiques peuvent faire progresser le domaine de la video microscopie. En effet on commence à voir apparaître pour des applications d'appareils photos numériques des lentilles optiques souples en polymère baignées dans un liquide, ce qui permet de les basculer d'une forme concave à convexe avec une extrême rapidité. Ce type de lentille est utilisé pour un réglage quasi instantané de la profondeur de champ de l'appareil photo. On peut espérer un futur microscope photonique équipé de cette lentille souple pour faire des balayages du focus avec une extrême rapidité. Il semble très intéressant d'orienter des travaux dans ce sens, car la profondeur de champ est un réel problème en micromanipulation. Cette méthode s'inscrit dans une suite logique de cette thèse.

6.2.2 Synthèse de nouvelle vue par super-résolution

Reconstruire la profondeur est très intéressant mais supprimer cette contrainte est encore mieux. Dans le chapitre 3 nous avons évoqué une technique de restauration de l'image appelée super-résolution. Cet outil utilise un ensemble d'images basses résolutions avec des déplacements inter-images sub-pixéliques pour reconstruire une image en

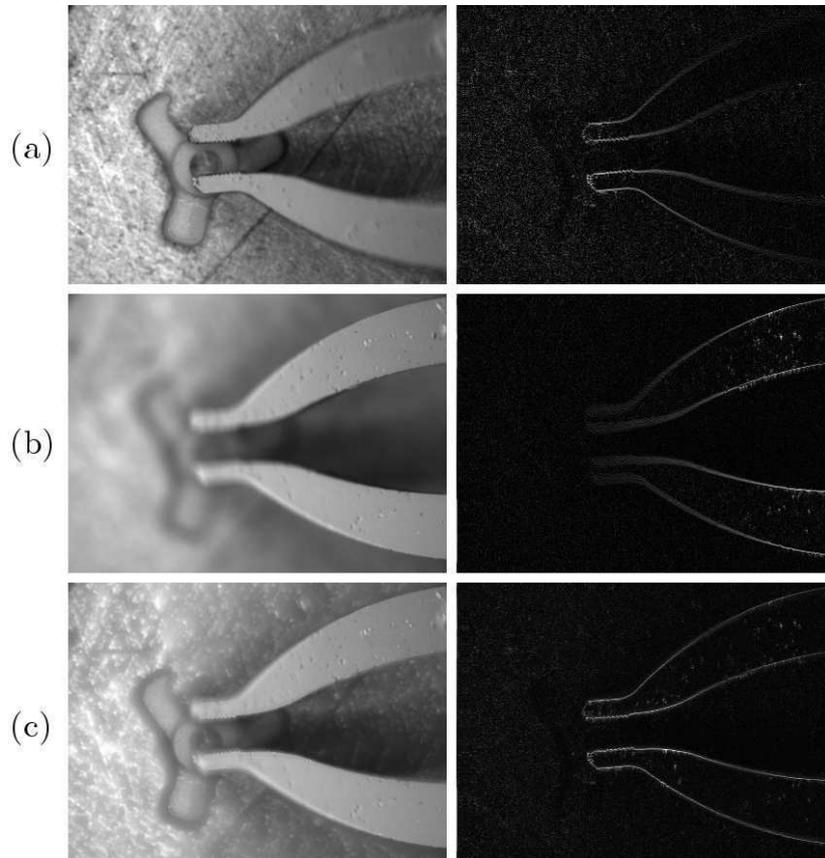


FIG. 6.1 – Extension de la profondeur de champ : (a) image originales nette sur sa parti gauche et représentation d’une partie des coefficients de la décomposition en ondelettes, (b) idem avec une image nette sur la partie droite, (c) reconstruction de la profondeur de champ, l’image est nette sur la partir droite et gauche.

haute résolution. Dans un microscope photonique plus le grandissement est important et plus la profondeur de champ est faible. Pourquoi ne pas utiliser les avantages d’un microscope photonique réel avec une grande profondeur de champ (faible résolution) et de le reconstruire virtuellement avec une grande résolution. En effet en utilisant des images provenant d’un microscope photonique de faible grandissement, la profondeur de champ va être importante. Ainsi les images montreront la totalité de l’espace de travail net. Pour pallier son faible grandissement une méthode de super-résolution pourrait être appliquée, permettant un facteur de grandissement important sur les images et de même qualité que si le zoom avait été optique. Le besoin de déplacement sub-pixélique peut être, par exemple, résolu en faisant vibrer soit l’imageur ou le capteur en lui-même par un système mécanique ou piézoélectrique pour avoir de faibles déplacements entre les images acquises. Au final le résultat permettrait d’avoir un microscope photonique virtuel de grande résolution avec une grande profondeur de champ.

6.2.3 Calibrage

Par rapport aux perspectives du calibrage de stéréo microscope photonique, il semble évident que sa réalisation peut être encore améliorée. En effet, la mire utilisée à partir d'une goutte d'eau recouverte par de la limaille de nickel a des dimensions qui ne permettent pas de faire le calibrage à une échelle de la scène inférieure à $300 \mu m$ de large. Pour l'étendre à d'autres échelles plus petites il serait intéressant de réaliser une mire de même propriété en microtechnologie. Pendant la durée de cette thèse, nous nous sommes orientés vers la réalisation d'une telle mire. Or malgré les techniques d'ingénierie en salle blanche, il existe une barrière technique dans la quantification et la réalisation de la texture de l'objet. En effet, la mire doit avoir une texture non répétitive totalement aléatoire faisant penser au technique de mouchetage. Sa réalisation et surtout la quantification de la taille du grain de la texture sont extrêmement difficiles à réaliser. Une étude plus approfondie doit être menée dans la réalisation de mires pour la stéréo microscopie. Une des autres perspectives importantes serait de franchir le pas du calibrage fort. Avec le moins de contraintes possibles, la plupart du temps le calibrage fort est obtenu par une vingtaines d'images montrant des postures différentes de la mire. Une voie possible serait de mettre en place un calibrage fort à partir d'une seule posture de mire facile à mettre en pratique dans une scène de micromanipulation.

6.2.4 Construction de mosaïques

Les différentes étapes que composent la construction de mosaïque sont encore très perfectibles. Par exemple une étude approfondie de la corrélation par III-ZNSSD dans le domaine fréquentielle permettrait de choisir la bonne taille du masque en fonction de la texture de l'image. Le détournage automatique pourrait aussi être amélioré avec une recherche d'optimisation de l'espace utile de façon globale et non locale dans les quadrants. On peut imaginer un rectangle qui se déforme par optimisation de la surface utile par le même principe qu'un simplexe. Dans l'incrustation de vues on pourra également approfondir la gestion du suivi de cible pour une meilleure robustesse. On pourrait aussi tenir compte du recalage géométrique et photométrique des vues incrustées dans la mosaïque. Cela permettrait de garantir le même facteur d'échelle entre l'image locale et globale ainsi que sa cohérence géométrique. On pourra également étendre la construction de mosaïque 2D en mosaïque 3D. Étant donné que l'on balaie l'espace de travail sur l'axe z pour le focus, pourquoi ne pas l'utiliser pour mettre en pratique une méthode de *depth-from-focus*, qui permettrait en plus la reconstruction de la surface de l'espace de travail en 3D. On pourrait simplement s'appuyer sur les travaux de V. Murino et al. (2000) et de M. Horstmann, T. Hermes et A. Winter (2003), où ils utilisent les images deux à deux pour reconstruire une mosaïque 3D. Avec le même balayage, et le même ensemble d'images originales on peut réaliser une mosaïque 3D de l'espace de travail. Par le balayage du focus on pourrait également combiner une reconstruction de champ de vision par construction de mosaïque et une reconstruction par profondeur de champ par EDF. Des méthodes de mesure qualitatives de la construction de la mosaïque peuvent

également être envisagées, comme mesurer la qualité de la fusion afin de la comparer avec d'autres techniques en vue de l'améliorer.

6.2.5 Transfert trifocal

Parmi toutes les sous-étapes pour mettre en application le transfert trifocal, une est la principale responsable de ses faibles performances en terme de rapidité : c'est la mise en correspondance. Qu'elle soit dense, quasi-dense ou d'intérêt, c'est elle qui englutit la plus grosse partie du temps de calcul. Une des améliorations serait d'implémenter ce type d'algorithme sur des systèmes DSP ou FPGA. On pourrait avoir un rendu en temps réel complet de la vue virtuelle. Néanmoins il est possible d'améliorer la rapidité du rendu sur une architecture PC classique. En effet, une des perspectives à apporter sur le transfert trifocal serait d'utiliser une méthode de transfert sans corrélation, c'est-à-dire sans connaître la correspondance des points. Cette méthode est possible dans un transfert avec un rendu de type *backward mapping* (R. Hartley et A. Zisserman 2000). Pour chaque point de la vue virtuelle il est possible d'obtenir les lignes épipolaires correspondante dans les deux vues réelles. Par le biais de la théorie de la géométrie trifocale, il existe une relation particulière entre les deux lignes et le point recherché. Il est donc possible de retrouver le point sans utiliser de mise en correspondance par corrélation, cela permettrait de gagner un temps considérable. Par contre supprimer la corrélation, c'est supprimer la carte de disparité et donc de rendre un rendu avec *z-buffer* inutilisable. Pour corriger cet inconvénient il faudrait également mettre en place une gestion des priorités d'affichage des pixels de la vue virtuelle à partir de la géométrie épipolaire, comme fait par L. McMillan et G. Bishop (1995a). D'autres aspects du rendu peuvent être également améliorés, filtrage des points, gestion des occultations, etc.

6.3 Perspectives à long terme

Quelques mots d'une perspective dans un futur très lointain... La micro-usine est constituée de stations de micromanipulation, où chacune de ces stations a la dimension d'un cube de 10 *cm* de côté. Chaque station réalise une étape différente de la fabrication du microproduit. Ces stations sont re-configurables, et pour garantir une grande flexibilité de production les stations peuvent se plugger dans différentes architectures en fonction de la gamme du microproduit à réaliser. Et la vision dans tout cela? Le système de vision est intégré dans chaque station sous une forme d'armature à la dimension de la station.

La Figure 6.2 montre un système de vision d'une station de micromanipulation contenant le vieux microrobot TRING (M. Rakotondrabe 2006). La deuxième image de cette figure montre un des imageurs incrustés dans le montant du système de vision. L'imageur est composé d'une lentille et d'un ensemble de fibres optiques de type microfibroscope. Après une amélioration technologique sur les fibres optiques, il est possible de déporter le capteur image tout en garantissant une résolution proche de nos anciens microscopes photoniques. Les douze imageurs placés dans les arêtes du cube de la station permettent une reconstruction fidèle en temps réel du moindre fait et geste à l'intérieur de la station. La scène reconstruite est utilisée pour la commande, la supervision de la station

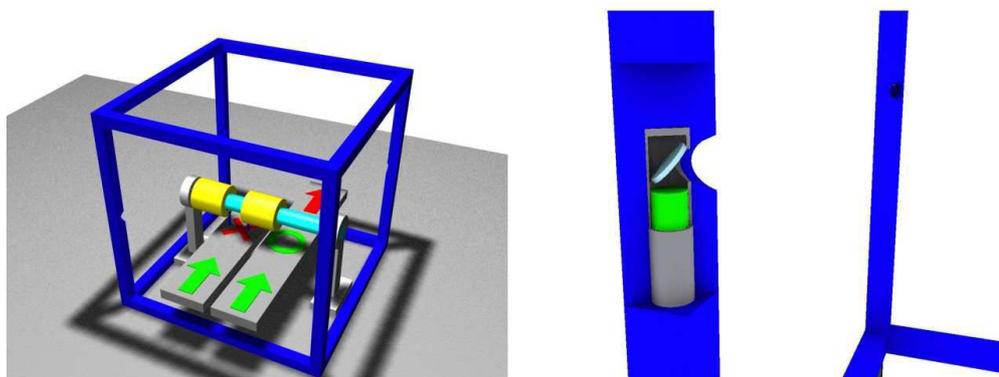


FIG. 6.2 – A gauche, le système de vision d'une station de micromanipulation, à droite, le détail d'un de ces imageurs.

et l'inspection par l'équipe de maintenance en se déplaçant virtuellement dans tous les recoins de la station. Comme chaque station a une tâche bien précise, la micro-usine a une architecture bien définie en fonction du flux de production des pièces (Figure 6.3). Les stations sont *plug-and-play*, cela permet donc de modifier à souhait l'architecture de la micro-usine. C'est pour cela que le système de vision intégré dans chaque station est également *plug-and-play*, comme le montre la Figure 6.3.

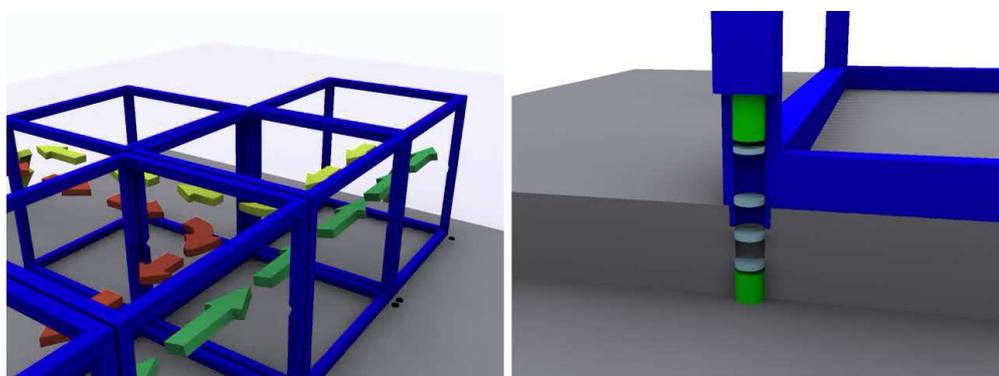


FIG. 6.3 – A gauche, un ensemble de stations de micromanipulation avec les flux de production représentés, à droite, le détail de la connectique du système de vision dans la micro-usine.

Un système de lentilles et de groupes de fibres optiques permettent d'interfacier le système de vision en plus de l'énergie et de l'information dans le socle de la micro-usine. Un superviseur détermine la position des cellules et reconstruit l'environnement de la micro-usine en vue de sa supervision. Chaque orifice de microfibroscope dans le socle de la micro-usine est connecté à une caméra directement branchée sur le ordinateur dédié au traitement du système de vision. Ce qui permet de remonter toutes les informations nécessaires au superviseur principal...

Cette vue d'esprit du futur système de vision de la micro-usine se heurte non seulement aux difficultés algorithmiques, mais aussi technologiques, humaines et financiers vu le coût d'une telle installation. La question n'est pas de savoir si un jour nous serons capable de développer d'un tel système, il est bien évident que oui, la réelle question est quand serons nous capable de produire un tel système de microproduction ? En attendant de voir cette pyramide de prouesses scientifiques et technologiques il reste encore beaucoup de pierres à poser, cette thèse en fait partie, pour arriver à la finalité ultime.

Bibliographie

- ADELSON, E. H. et BERGEN, J. R. (1991). « The Plenoptic Function and the Elements of Early Vision », *Computational Models of Visual Processing*, pp. 3–20.
- ADELSON, E. H. , ANDERSON, C. H. , BERGEN, J. R. , BURT, P. J. et OGDEN, J. M. (1984). « Pyramid Methods in Image Processing », *RCA Engineer*, **29**(6) : 33–41.
- AGARWAL, A. , JAWAHAR, C. V. et NARAYANAN, P. J. (2005). « A Survey of Planar Homography Estimation Techniques », *Technical Report IIIT/TR/2005/12*, International Institute of Information Technology, Hyderabad, India.
- AMAVASAI, B. P. , MEIKLE, S. et YATES, R. B. (2000). « A robust vision system for micro-object manipulation », *SPIE Proceedings of the Microrobotics and Microassembly II*, Vol. 4194, pp. 112–120.
- AMMI, M. , FREMONT, V. et FERREIRA, A. (2005). « Flexible Microscope Calibration using Virtual Pattern for 3-D Telemicromanipulation », *IEEE International Conference on Robotics and Automation*, pp. 3899–3904.
- APPEL, A. (1968). « Some Techniques for Shading Machine Renderings of Solids », *Proc. AFIPS Spring Joint Computer Conference*, Vol. 32, pp. 37–45.
- APPLETON, B. , BRADLEY, A. P. et WILDERMOTH, M. (2005). « Towards Optimal Image Stitching for Virtual Microscopy », *IEEE Proceedings of Digital Image Computing : Techniques and Applications (DICTA)*, pp. 299–306.
- ASCHWANDEN, P. et GUGGENBUHL, W. (1992). « Experimental results from a comparative study on correlation-type registration algorithms », *Robust Computer Vision*, pp. 268–289.
- ASMUTH, J. , MADJAROV, B. , SAJDA, P. et BERGER, J. (2001). « Mosaicking and Enhancement of Slitlamp Biomicroscope Fundus Images », *British Journal of Ophthalmology*, Vol. 85, pp. 563–565.
- AVIDAN, S. et SHASHUA, A. (1996). « Tensorial transfer : On the representation of $N > 3$ views of a 3D scenes », *Proceedings of the Image Understanding Workshop*.
- AVIDAN, S. et SHASHUA, A. (1997). « Novel view synthesis in tensor space », *IEEE Computer Vision and Pattern Recognition*, pp. 1034–1040.

- AVIDAN, S. et SHASHUA, A. (1998). « Novel View Synthesis by Cascading Trilinear Tensors », *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, Vol. 4, pp. 293–306.
- AYACHE, N. (1989). *Vision stéréoscopique et perception multisensorielle. Applications à la robotique mobile*, Collection Science Informatique, Paris.
- AYACHE, N. et HANSEN, C. (1988). « Rectification of Images for Binocular and Trinocular Stereovision », *IEEE International Conference on Pattern Recognition (ICPR)*, Vol. 1, Beijing, China, pp. 11–16.
- BAKER, S. , SZELISKI, R. et ANANDAN, P. (1998). « A layered approach to stereo reconstruction », *IEEE Computer Vision and Pattern Recognition (CVPR)*, Santa Barbara, pp. 434–441.
- BARNEA, E. I. et SILVERMAN, H. F. (1972). « A class of Algorithms for Fast Digital Image Registration », *IEEE Transactions on Computers*, Vol. C-21, pp. 179–186.
- BAY, H. , TUYTELAARS, T. et GOOL, L. Van (2006). « SURF : Speeded Up Robust Features », *Proceedings of European Conference on Computer Vision (ECCV)*.
- BLANC, Philippe et SAVARIA, Eric (2001). « Techniques de mosaïquage pour les systèmes spatiaux d'imagerie optique à haute résolution », *Revue des télécommunications d'Alcatel*, pp. 1–6.
- BLANC, Philippe , SAVARIA, Eric et OUDYI, Farid (2001). « Le mosaïquage d'images satellitaires optiques a haute resolution spatiale », *18e colloque sur le traitement du signal et des images (GRETSI'01)*, Vol. 2, Toulouse, France, pp. 251–254.
- BLINN, J. F. (1978). « Simulation of Wrinkled Surfaces », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Vol. 12, pp. 286–292.
- BOIVIN, S. et GAGALOWICZ, A. (2001). « Image-Based Rendering from a Single Image », *Technical report 4098*, INRIA.
- BORGETTO, M. (2005). *Contribution à la construction de mosaïques d'images sous-marines géo-référencées par l'introduction de méthodes de localisation*, PhD thesis, Université du Sud Toulon-Var.
- BOUFAMA, B. S. (2000). « The use of homographies for view synthesis », *IEEE International Conference on Pattern Recognition (ICPR)*, Vol. 1, pp. 563–566.
- BRADLEY, A. P. et BAMFORD, P. C. (2004). « A One-pass Extended Depth of Field Algorithm Based on the Over-complete Discrete Wavelet Transform », *Proceeding Image and Vision Computing New Zealand (IVCNZ)*, Akaroa, New Zealand, pp. 279–284.
- BROWN, J. (1981). « Multichannel sampling of low-pass signals », *IEEE Transactions on Circuits and Systems*, Vol. 28(2), pp. 101–106.
- BUEHLER, C. , BOSSE, M. , McMILLAN, L. , GORTLER, S. et COHEN, M. (2001). « Unstructured Lumigraph rendering », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Los Angeles, USA, pp. 425–432.
- BURSCHKA, D. , COBZAS, D. , DODDS, Z. , HAGER, G. , JAGERSAND, M. et YEREX, K. (2003). « Recent Methods for Image-based Modeling and Rendering », *Tutorial 1*, IEEE Virtual Reality.

- CANNY, J. F. (1986). « A computational approach to edge detection », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 679–698.
- CAPEL, D. P. (2001). *Image Mosaicing and Super-resolution*, PhD thesis, University of Oxford.
- CAPEL, David et ZISSERMAN, Andrew (2001). « Automated mosaicing with super-resolution zoom », *IEEE Computer vision and pattern recognition*, pp. 885–891.
- CATMULL, E. (1974). *A Subdivision Algorithm for Computer Display of Curved Surfaces*, PhD thesis, Univeristy of Utah.
- CHAI, J.-X. , TONG, X. , CHAN, S.-C. et SHUM, H.-Y. (2000). « Plenoptic sampling », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 307–318.
- CHAN, S. O.-Y. , WONG, Y.-P. et DANIEL, J. K. (2003). « Dense Stereo Correspondence Based On Recursive Adaptive Size Multi-Windowing », *Proceeding Image and Vision Computing New Zealand (IVCNZ)*, Palmerston North, New Zealand, pp. 256–259.
- CHANG, C. , BISHOP, G. et LASTRA, A. (1999). « LDI tree : A hierarchical representation for image-based rendering », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 291–298.
- CHEN, S. (1995). « QuickTime VR - an image-based approach to virtual environment navigation », *Computer Graphics (SIGGRAPH'95 Proceedings)*, pp. 29–38.
- CHEN, S. E. et WILLIAMS, L. (1993). « View Interpolation for Image Synthesis », *Computer Graphics (SIGGRAPH'93 Proceedings)*, pp. 279–288.
- CIBERT, C. (2005). *Fondamentaux d'optique et d'imagerie numérique à l'usage des microscopistes*, Cépaduès-éditions.
- CLARK, J. H. (1976). « Hierarchical Geometric Models for Visible Surface Algorithms », *Communications of the ACM*, **19**(10) : 547–554.
- CODOUREY, A. , RODRIGUEZ, M. et PAPPAS, I. (1997). « A Task-oriented Teleoperation System for Assembly in the Microworld », *Proceeding of International Conference on Advanced Robotics (ICAR)*.
- COHEN-OR, D. , CHRYSANTHOU, Y. L. , SILVA, C. T. et DURAND, F. (2003). « A Survey of Visibility for Walkthrough Applications », *IEEE Transactions on Visualization and Computer Graphics*, **9**(3) : 412–431.
- CONNOR, Keith et REID, Ian (2002). « Novel View Specification and Synthesis », *Proceeding of the British Machine Vision Conference*, Cardiff, England, pp. 243–252.
- DEBEVEC, P. E. , TAYLOR, C. J. et MALIK, J. (1996). « Modeling and rendering architecture from photographs : A hybrid geometry- and image-based approach », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 11–20.

- DEBEVEC, P. E. , YU, Y. et BORSHUKOV, G. (1998). « Efficient view-dependent image-based rendering with projective texture-mapping », *Eurographics Workshop on Rendering*, pp. 105–116.
- DECORET, X. , DURAND, F. , SILLION, F. X. et DORSEY, J. (2003). « Billboard clouds for extreme model simplification », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 689–696.
- DETTWILLER, L. (1997). *Les instruments d'optique, étude théorique, expérimentale et pratique*, Ellipses.
- DEVERNAY, F. (1997). *Vision stéréoscopique et propriétés différentielles des surfaces*, PhD thesis, Ecole Polytechnique.
- DING, W. , GONG, Z. , XIE, S. et ZOU, H. (2006). « Real-time Vision-based Object Tracking from a Moving Platform in the Air », *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, pp. 681–685.
- ELAD, Michael et HEL-OR, Yacov (2001). « A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur », *IEEE Transactions on image processing*, Vol. 10, pp. 1187–1193.
- ENIKOV, E. T. , CLARK, S. et MINKOV, L. (2003). « Micro-assembly cell with dual optical/computer vision control for electrostatic gripping of MEMS », *SPIE Proceedings of Intelligent Robots and Computer Vision XXI*, Vol. 5267, pp. 269–280.
- ERDEM, A. , SEZAN, M. et OZKAN, M. (1992). « Motion-compensated multiframe wiener restoration of blurred and noisy image sequences », *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 3, pp. 293–296.
- EREN, P. E. , SEZAN, M. I. et TEKALP, A. (1997). « Robust, Object-Based High-Resolution Image Reconstruction from Low-Resolution Video », *IEEE Transactions on Image Processing*, Vol. 6(10), pp. 1446–1451.
- EVERS-SENNE, J.-F. et KOCH, R. (2005). « Image-based rendering of complex scenes from a multi-camera rig », *IEE Vision, Image & Signal Processing (VISIP)*, **152**(4) : 470–480.
- FARSIU, Sina , ROBINSON, Dirk , ELAD, Michael et PEYMAN, Milanfar (2004). « Fast and Robust Multi-frame Super-resolution », *IEEE Transactions on Image Processing*, Vol. 13, pp. 1327–1344.
- FATIKOW, S. , BUERKLE, A. et SEYFRIED, J. (1999). « Automatic control system of a microrobot-based microassembly station using computer vision », *SPIE Proceedings of the Microrobotics and Microassembly*, Vol. 3834, pp. 11–22.
- FATIKOW, S. , SEYFRIED, J. , FAHLBUSCH, S. , BUERKLE, A. et SCHMOECKEL, F. (2000). « A Flexible Microrobot-Based Microassembly Station », *Journal of Intelligent and Robotic Systems*, **27** : 135–169.
- FAUGERAS, O. (1993). *Three-Dimensional Computer Vision*, The MIT Press.
- FAUGERAS, O. et ROBERT, L. (1993). « What can two images tell us about a third one ? », *Technical report 2018*, INRIA.

- FEDDEMA, J. T. et CHRISTENSON, T. R. (1999). « Parallel Assembly of High Aspect Ratio Microstructures », *SPIE Proceedings of the Microrobotics and Microassembly*, Vol. 3834, pp. 153–164.
- FEDDEMA, J. T. et SIMON, R. W. (1998). « Visual Servoing and CAD-Driven Microassembly », *IEEE Robotics & Automation Magazine*, pp. 18–24.
- FISCHLER, M. A. et BOLLES, R. C. (1981). « Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography », *Communications of the Association for Computing Machinery (ACM)*, **24**(6) : 381–395.
- FITZGIBBON, A. , WEXLER, Y. et ZISSERMAN, A. (2003). « Image-based rendering using image-based priors », *IEEE International Conference on Computer Vision (ICCV)*.
- FORSTER, B. , VILLE, D. Van De , BERENT, J. , SAGE, D. et UNSER, M. (2004). « Extended depth-of-focus for multi-channel microscopy images : a complex wavelet approach », *IEEE International Symposium on Biomedical Imaging (ISBI)*, Arlington VA, USA, pp. 660–663.
- FUSIELLO, A. , APRILE, M. , MARZOTTO, R. et MURINO, V. (2003). « Mosaic of a video shot with multiple moving objects », *IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain.
- FUSIELLO, A. , ROBERTO, V. et TRUCCO, E. (2000). « Symmetric stereo with multiple windowing », *International Journal of Pattern Recognition and Artificial Intelligence*, **14**(8) : 1053–1066.
- GENC, Y. (1999). *Weak calibration and image-based rendering algorithms*, PhD thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois.
- GLÜCKSTAD, J. , DARIA, V. R. , ERIKSEN, R. L. et RODRIGO, P. J. (2003). « Vision-guided manipulation of colloidal structures », *SPIE Proceedings of the Optical Pattern Recognition XIV*, Vol. 5106, pp. 46–52.
- GORTLER, S. J. , GRZESZCZUK, R. , SZELISKI, R. et COHEN, M. F. (1996). « The Lumigraph », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 43–54.
- GRACIAS, N. et SANTOS-VICTOR, J. (2001). « Underwater mosaicing and trajectory reconstructing using global alignment », *IEEE Conference Proceedings Oceans*, Honolulu.
- GRACIAS, N. , GLEASON, A. , NEGAHDARIPOUR, S. et MAHOOR, M. (2006). « Fast Image Blending using Watersheds and Graph Cuts », *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 469–478.
- GRACIAS, N. R. E. (2002). *Mosaic-based Visual Navigation for Autonomous Underwater Vehicles*, PhD thesis, University of Lisbon.
- GREENE, N. et HECKBERT, P. S. (1986). « Creating raster Omnimax images from multiple perspective views using the Elliptical Weighted Average filter », *IEEE Computer Graphics and Applications*, Vol. 6(6), pp. 21–27.

- GROEN, F. , YOUNG, IT. et LIGTHART, G. (1985). « A comparison of different focus functions for use in autofocus algorithms », *Cytometry*, Vol. 12, pp. 81–91.
- GUESTRIN, C. , COZMAN, F. et SIMOES, M. G. (1998). « Industrial applications of image mosaicing and stabilization », *International Conference on Knowledge-Based Intelligent Electronic Systems*, Adelaide, Australia, pp. 174–183.
- HABED, A. et BOUFAMA, B. (2000). « Novel View Synthesis : a comparative analysis study », *In Proceeding of Vision Interface (VI)*.
- HARDIE, R. C. , BARNARD, K. J. et ARMSTRONG, E. E. (1997). « Joint MAP Registration and High-Resolution Image Estimation Using a Sequence of Undersampled Images », *IEEE Transactions on Image Processing*, Vol. 6(12), pp. 1621–1633.
- HARRIS, C. et STEPHENS, M. (1988). « A combined corner and edge detector », *Proceeding of 4th Alvey Vision Conference*, pp. 147–151.
- HARTLEY, R. (1994). « Lines and points in three views - an integrated approach », *Proceedings of the ARPA IU Workshop*.
- HARTLEY, R. (1999). « Theory and Practice of Projective Rectification », *International Journal of Computer Vision*, **35**(2) : 115–127.
- HARTLEY, R. et ZISSERMAN, A. (2000). *Multiple View Geometry*, Cambridge University Press.
- HARTLEY, R. I. (1997). « In Defense of the Eight-Point Algorithm », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19(6), pp. 580–593.
- HAYNES, S. M. et JAIN, R. (1983). « Detection of moving edges », *Computer Vision, Graphics, and Image Processing*, Vol. 21(3), pp. 345–367.
- HECKBERT, P. (1989). « *Fundamentals of texture mapping and image warping* », Master's thesis, The University of California at Berkeley, USA.
- HEIDRICH, W. , LENSCH, H. , COHEN, M. F. et SEIDEL, H.-P. (1999). « Light field techniques for reflections and refractions », *Eurographics Workshop on Rendering*, pp. 195–205.
- HERNANDEZ, C. et SCHMITT, F. (2004). « Silhouette and Stereo Fusion for 3D Object Modeling », *Computer Vision and Image Understanding*, **96**(3) : 367–392.
- HORSTMANN, M. , HERMES, T. et WINTER, A. (2003). « 3D-Mosaicing for Virtual Reconstruction of Environments », *3D-NordOst*, pp. 41–48.
- HSU, C.-T. et WU, J.-L. (1996). « Multiresolution mosaic », *IEEE International Conference on Image Processing (ICIP)*, Vol. 3, pp. 743–746.
- HÖTZENDORFER, H. , FERROS, M. , GIOUROUDI, I. et BRENNER, W. (2006). « Micromanipulation System with Stereoscopic Imaging », *The IASTED International Conference on Robotics and Applications*, pp. 22–26.
- IRANI, M. et PELEG, S. (1993). « Motion analysis for image enhancement : Resolution, occlusion and transparency », *Journal of Visual Communication and Image Representation*, **4** : 324–335.

- IRANI, M. , ANANDAN, P. , BERGEN, J. , KUMAR, R. et HSU, S. (1996). « Mosaic Representations of Video Sequences and Their Applications », *Signal Processing : Image Communication, special issue on Image and Video Semantics : Processing, Analysis, and Application*, Vol. 8(4), pp. 327–351.
- IRANI, M. , HASSNER, T. et ANANDAN, P. (2002). « What does the scene look like from a scene point ? », *Proceedings of European Conference on Computer Vision (ECCV)*, Copenhagen, pp. 883–897.
- JONES, C. B. (1971). « A New Approach to the Hidden Line Problem », *Computer Journal*, **14**(3) : 232–237.
- KANG, S. B. (1997). « A Survey of Image-based Rendering Techniques », *Technical Report CRL 97/4*, Cambridge Research Laboratory.
- KASAYA, T. , MIYAZAKI, H. , SAITO, S. et SATO, T. (1999). « Micro Object Handling under SEM by Vision-based Automatic Control », *IEEE International Conference on Robotics and Automation*, pp. 2189–2196.
- KATOH, Y. , ARAI, T. et INOUE, K. (2001). « Auto Focusing and Tracking for Micro Manipulation », *Mecatronics'01*, Besançon, France, pp. 349–354.
- KEREN, D. , PELEG, S. et R. BRADA (1988). « Image sequence enhancement using subpixel displacements », *IEEE Computer Vision and Pattern Recognition*, pp. 742–746.
- KIM, B. , KANG, H. , KIM, D.-H. et PARK, J.-O. (2006). « A flexible microassembly system based on hybrid manipulation scheme for manufacturing photonics components », *International journal, advanced manufacturing technology*, **28** : 379–386.
- KIM, K. , KIM, D.-H. , LEE, S. J. et LEE, J. (2002). « Hybrid Microassembly System for Three-Dimensional MEMS Components », *International Workshop on Microfactories*, pp. 21–24.
- KIMURA, K. et SAITO, H. (2005). « Video Synthesis at Tennis Player Viewpoint from Multiple View Videos », *IEEE Virtual Reality (VR)*, Bonn, Germany, pp. 281–282.
- KOMATSU, T. , IGARASHI, T. , AIZAWA, K. et SAITO, T. (1993). « Very high resolution imaging scheme with multiple different aperture cameras », *Signal Processing : Image Communication*, Vol. 5, pp. 511–526.
- KOUROGI, Masakatsu , KURATAY, Takeshi , HOSHINOZ, Jun'ichi et MURAOKA, Yoichi (1999). « Real-Time Image Mosaicing from a Video Sequence », *IEEE International Conference on Image Processing*, pp. 133–137.
- KOVESI, P. (n.d.). <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/index.html>. Senior Lecturer School of Computer Science & Software Engineering The University of Western Australia.
- KUGLIN, C. D. et HINES, D. C. (1975). « The Phase Correlation Image Alignment Method », *Proceeding IEE International Conference on Cybernetic Society*, New York, USA, pp. 163–165.

- KUMAR, G. Hemantha , SHIVAKUMARA, P. , GURU, D. S. et NAGABHUSHAN, P. (2004). « Document image mosaicing : A novel approach », *SADHANA - Academy Proceedings in Engineering Sciences*, Vol. 29(3), India, pp. 329–341.
- KUMAR, R. , ANANDAN, P. et HANNA, K. (1994). « Shape recovery from multiple views : a parallax based approach », dans M. K. PUBLISHERS (éd.), *In Image Understanding Workshop*, Monterey, CA, pp. 947–955.
- LAVEAU, S. (1996). *Géométrie d'un système de n caméras. Théorie, estimation et applications.*, Thèse, Ecole polytechnique.
- LAVEAU, S. et FAUGERAS, O. (1994). « 3-D Scene Representation as a Collection of Images and Fundamental Matrices », *Rapport de recherche 2205*, INRIA.
- LEE, Seok Joo , KIM, Kyunghwan , KIM, Deok-Ho , PARK, Jong-Oh et PARK, Gwi-Tae (2001). « Recognizing and Tracking 3D-Shaped Micro Parts Using Multiple Views for Micromanipulation », *IEEE International Symposium on Micromechatronics and Human Science*, Nagoya in Japan, pp. 203–210.
- LENSCH, H. (1999). « *Techniques for Hardware-Accelerated Light Field Rendering* », Master thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg.
- LEVIN, A. , ZOMET, A. , PELEG, S. et WEISS, Y. (2004). « Semaless image stitching in the gradient domain », *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 377–389.
- LEVOY, M. et HANRAHAN, P. (1996). « Light field rendering », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 31–42.
- LEVOY, M. et WHITTED, T. (1985). « The use of points as a display primitive », *Technical report UNC 85-022*, University of North Carolina, Chapel Hill, NC.
- LEVOY, M. , NG, R. , ADAMS, A. , FOOTER, M. et HOROWITZ, M. (2006). « Light Field Microscopy », *ACM Transactions on Graphics (Proc. SIGGRAPH 2006)*, Vol. 25(3).
- LHULLIER, M. et QUAN, L. (1999). « Image Interpolation by Joint View Triangulation », *IEEE Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, pp. 139–145.
- LI, G. et XI, N. (2002). « Calibration of a Micromanipulation System », *IEEE International Conference on Intelligent Robots and Systems (IROS)*, EPFL, Lausanne, Switzerland, pp. 1742–1747.
- LI, H. et HARTLEY, R. (2005). « Inverse Tensor Transfer for Novel View Synthesis », *IEEE International Conference on Image Processing (ICIP)*, Vol. 2, pp. 97–100.
- LIN, W.H. , SENGUPTA, K. et SHARMA, R. (2003). « Augmented Reality with Occlusion Rendering Using Background-Foreground Segmentation and Trifocal Tensors », *IEEE International Conference on Multimedia & Expo (ICME)*, Baltimore, Maryland, USA, pp. 93–96.
- LISCHINSKI, D. et RAPPOPORT, A. (1998). « Image-based rendering for non-diffuse synthetic scenes », *Eurographics Workshop on Rendering*, pp. 301–314.
- LONGUET-HIGGINS, H. C. (1981). « A Computer Algorithm for Reconstruction a Scene from Two Projections », *Nature*, **293** : 133–135.

- LOWE, D. G. (1999). « Object Recognition from Local Scale-Invariant Features », *IEEE International Conference on Computer Vision (ICCV)*, pp. 1150–1157.
- LUCAS, B. D. et KANADE, T. (1981). « An Iterative Image Registration Technique with an Application to Stereo Vision », *Proceeding of Imaging Understanding Workshop*, pp. 121–130.
- LUONG, Q.-T. et FAUGERAS, O.D. (1996). « The Fundamental matrix : theory, algorithms, and stability analysis », *International Journal of Computer Vision*, **17**(1) : 43–76.
- MAGAIN, P. , COURBIN, F. et SOHY, S. (1998). « Deconvolution with Correct Sampling », *Astrophysical Journal*, **494** : 472–492.
- MALIS, E. et MARCHAND, E. (2005). « Méthodes robustes d'estimation pour la vision robotique », *5e Journées Nationales de la Recherche en Robotique*, pp. 51–60.
- MARKS, Richard , ROCK, Stephen et LEE, Michael (1994). « Real-Time Video Mosaicking of the Ocean Floor », *IEEE Symposium on Autonomous Underwater Vehicle Technology*.
- MAYER, H. (2002). « Estimation of and View Synthesis with the Trifocal Tensor », *Photogrammetric Computer Vision (PCV)*, Austria, pp. 211–217.
- MCMILLAN, L. (1997). *An image-based approach to three-dimensional computer graphics*, Phd thesis, University of North Carolina.
- MCMILLAN, L. et BISHOP, G. (1995a). « Head-tracked stereoscopic display using image warping », *SPIE Proceedings of the Stereoscopic Displays and Virtual Reality Systems II*, pp. 21–30.
- MCMILLAN, L. et BISHOP, G. (1995b). « Plenoptic modeling : An image-based rendering system », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 39–46.
- MEAGHER, D. (1982). « Efficient Synthetic Image Generation of Arbitrary 3D Objects », *IEEE International Conference on Pattern Recognition and Image Processing*, pp. 473–478.
- MILGRAM, D. L. (1975). « Computer Methods for Creating Photomosaics », *IEEE Transactions on Computers*, Vol. 24, pp. 1113–1119.
- MILGRAM, D. L. (1977). « Adaptive techniques for photomosaicking », *IEEE Transactions on Computers*, Vol. 26, pp. 1175–1180.
- MOKHTARIAN, Farzin et SUOMELA, Riku (1998). « Robust Image Corner Detection Through Curvature Scale Space », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(12) : 1376–1381.
- MOLINIER, T. , FOFI, D. et GORRIA, P. (2006). « Novel View Synthesis for Projective Texture Mapping on Real 3D Objects », *SPIE Proceedings of the Electronic Imaging - Machine Vision Applications in Industrial Inspection XIV*, San José, USA.
- MORAVEC, H. P. (1977). « Towards automatic visual obstacle avoidance », *Proceeding of the 5th International Joint Conference on Artificial Intelligent*, p. 584.

- MORITA, H. , HILD, M. , MIURA, J. et SHIRAI, Y. (2006). « Panoramic View-Based Navigation in Outdoor Environments Based on Support Vector Learning », *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, pp. 2302–2307.
- MURINO, V. , FUSIELLO, A. , IURETIGH, N. et PUPPO, E. (2000). « 3D mosaicing for environment reconstruction », *IEEE International Conference on Pattern Recognition (ICPR)*, pp. 362–366.
- NALWA, V. S. (1996). « A true omnidirectional viewer », *Technical report*, Bell Laboratories, Holmdel, NJ.
- NELDER, J. A. et MEAD, R. (1965). « A simplex method for function minimization », *Computer Journal*, **7** : 308–313.
- OLIVEIRA, M. M. (2000). *Relief texture mapping*, Phd thesis, University of North Carolina.
- PAPOULIS, A. (1977). « Generalized sampling expansion », *IEEE Transactions on Circuits and Systems*, Vol. 24(11), pp. 652–654.
- PATTI, A. J. , TEKALP, A. M. et SEZAN, M. I. (1998). « A new motion compensated reduced order model Kalman filter for space-varying restoration of progressive and interlaced video », *IEEE Transactions on Image Processing*, Vol. 7, pp. 543–554.
- PELEG, S. (1981). « Elimination of seams from photomosaics », *Computer Graphics and Image Processing (CGIP)*, Vol. 16, pp. 90–94.
- PELEG, S. et HERMAN, J. (1997). « Panoramic Mosaics by Manifold Projection », *IEEE Computer Vision and Pattern Recognition*, pp. 338–343.
- PELEG, S. , KEREN, D. et SCHWEITZER, L. (1987). « Improving image resolution by using subpixel motion », *Pattern Recognition Letters*, **5**(3) : 223–226.
- PELEG, Shmuel et BEN-EZRA, Moshe (1999). « Stereo Panorama with a Single Camera », *IEEE Computer Vision and Pattern Recognition*, Ft. Collins, pp. 395–401.
- PHONG, B. T. (1975). « Illumination for Computer Generated Pictures », *Communications of the ACM*, Vol. 18(6), pp. 311–317.
- PILU, M. et ISGRO, F. (2002). « A fast and reliable planar registration method with applications to document stitching », *Proceeding of the British Machine Vision Conference*, Cardiff, pp. 688–697.
- POTSAID, B. , BELLOUARD, Y. et WEN, J. T. (2005). « Adaptive Scanning Optical Microscope (ASOM) : A multidisciplinary optical microscope design for large field of view and high resolution imaging », *The International Electronic Journal of Optics*, **13**(17) : 6504–6518.
- POTSAID, Benjamin , BELLOUARD, Yves et WEN, John T. (2002). « Scanning Optical Mosaic Scope for Micro-Manipulation », *International Workshop on Microfactories*, Minneapolis, Minnesota, USA.
- POTSAID, Benjamin , BELLOUARD, Yves et WEN, John T. (2003). « Scanning Optical Mosaic Scope for Dynamic Biological Observations and Manipulation », *IEEE Intelligent Robotics Systems (IROS)*, Las Vegas, NV, USA.

- RAKOTONDRABE, M. (2006). *Développement et commande modulaire d'une station de microassemblage*, PhD thesis, Université de Franche-Comté, Besançon.
- RALIS, S. , NELSON, B. J. et VIKRAMADITYA, B. (2000). « Micropositioning of a Weakly Calibrated Microassembly System Using Coarse-To-Fine Visual Servoing Strategies », *IEEE Transactions on Electronic Packaging Manufacturing*, **23**(2) : 123 – 131.
- RAV-ACHA, A. et PELEG, S. (2006). « Lucas Kanade Without Iterative Warping », *IEEE International Conference on Image Processing (ICIP)*, pp. 1097–1100.
- RAV-ACHA, A. , PRITCH, Y. , LISCHINSKI, D. et PELEG, S. (2005). « Dynamosaics : Video Mosaics with Non-Chronological Time », *IEEE Computer Vision and Pattern Recognition (CVPR)*, San Diego, pp. 58–65.
- RESSL, C. (2003). *Geometry, Constraints and Computation of the Trifocal Tensor*, PhD thesis, Institut of Photogrammetry, University of Bonn.
- ROUSSEEUW, P. J. (1984). « Least Median of Squares Regression », *Journal of the American Statistical Association*, **79**(388) : 871–880.
- ROUSSO, B. , AVIDAN, S. , SHASHUA, A. et PELEG, S. (1996). « Robust recovery of camera rotation from three frames », *DARPA Image Understanding Workshop*, pp. 851–856.
- ROUSSO, B. , PELEG, S. et FINCI, I. (1997). « Mosaicing with generalized strips », *DARPA Image Understanding Workshop*, New Orleans, Louisiana, USA, pp. 255–260.
- SAITO, H. , BABA, S. et KANADE, T. (2003). « Appearance-Based Virtual View Generation From Multicamera Videos in the the 3-D Room », *IEEE Transactions on Multimedia*, Vol. 5, pp. 303–316.
- SANO, T. et YAMAMOTO, H. (2004). « Micromanipulation with Stereoscopic Imaging », *IEEE International Workshop an Imaging Systems and Techniques*, Stresa, Italy, pp. 48–52.
- SATO, T. , KAMEYA, T. , MIYAZAKI, H. et HATAMURA, Y. (1995). « Hand-Eye System in Nano Manipulation World », *IEEE International Conference on Robotics and Automation*, pp. 59–66.
- SCHARSTEIN, D. (1996). « Stereo vision for view synthesis », *IEEE Computer Vision and Pattern Recognition (CVPR)*, San Francisco, USA, pp. 852–857.
- SCHARSTEIN, D. et SZELISKI, R. (2002). « A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms », *International Journal of Computer Vision*, **47** : 7–42.
- SCHILP, J. , HARFENSTELLER, M. , JACOB, D. et SCHILP, M. (2003). « High-accuracy micro-assembly by intelligent vision systems and smart sensor integration », *SPIE Proceedings of the Optomechatronic Systems IV*, Vol. 5264, pp. 274–282.
- SCHMID, C. , MOHR, R. et BAUCKHAGE, C. (1998). « Comparing and evaluating interest points », *IEEE International Conference on Computer Vision*, pp. 230–235.

- SCHULTZ, R. R. et STEVENSON, R. L. (1996). « Extraction of high-resolution frames video sequences », *IEEE Transactions on Image Processing*, Vol. 5(6), pp. 996–1011.
- SEITZ, S. M. et DYER, C. R. (1996). « View Morphing », *Computer Graphics (SIGGRAPH'96 Proceedings)*, pp. 21–30.
- SEITZ, S. M. , CURLESS, B. , DIEBEL, J. , SCHARSTEIN, D. et SZELISKI, R. (2006). « A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms », *IEEE Computer Vision and Pattern Recognition (CVPR)*, Vol. 1, pp. 519–526.
- SEMPLE, J. G. et KNEEBONE, G. T. (1952). *Algebraic Projective Geometry*, Oxford Press.
- SHACKLOCK, A. et SUN, W. (2005). « Ephemera : Tracking of Motion in Microscope Views », *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 1847–1852.
- SHACKLOCK, Andrew (2004). « Multiple-View Multiple-Scale Navigation for Micro-Assembly », *IEEE International Conference on Robotics and Automation*, New Orleans, LA, pp. 902–907.
- SHADE, J. , GORTLER, S. , HE, L.-W. et SZELISKI, R. (1998). « Layered depth images », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Orlando, USA, pp. 231–242.
- SHASHUA, A. (1994). « Algebraic Functions For Recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, pp. 779–789.
- SHASHUA, A. et NAVAD, N. (1994). « Relative Affine Structure : Theory and Application to 3D Reconstruction from Perspective Views », *IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 483–489.
- SHIVAKUMARA, P. , KUMAR, G. H. , GURU, D. S. et NAGABHUSHAN, P. (2006). « Sliding window based approach for document image mosaicing », *Journal of Image and Vision Computing*, **24** : 94–100.
- SHUM, H.-Y. et HE, L.-W. (1999). « Rendering with Concentric Mosaics », *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Los Angeles, USA, pp. 299–306.
- SHUM, H.-Y. et KANG, S. B. (2000). « A Review of Image-based Rendering Techniques », *SPIE Proceedings of the Visual Communications and Image Processing*, pp. 2–13.
- SHUM, H.-Y. , CHAN, S.-C. et KANG, S. B. (2006). *Image-Based Rendering*, Springer.
- SHUM, Heung-Yeung et SZELISKI, Richard (1997). « Panoramic Image Mosaics », *Technical Report MSR-TR-97-23*, Microsoft Research.
- SIEVERS, T. et FATIKOW, S. (2005). « Visual Servoing of a Mobile Microrobot inside a Scanning Electron Microscope », *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Canada, pp. 1682–1686.
- SLAYTER, E. M. et SLAYTER, H. S. (1992). *Light and electron microscopy*, Cambridge University Press.

- SLOAN, P. P. , COHEN, M. F. et GORTLER, S. J. (1997). « Time critical Lumigraph rendering », *Symposium on Interactive 3D Graphics*, Providence, USA, pp. 17–23.
- SLUZEK, A. , ISLAM, Md S. et ANNAMALAI, P. (2006). « Using Interest Points for Robust Visual Detection and Identification of Objects in Complex Scenes », *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, pp. 5321–5326.
- SMITH, S.M. et BRADY, J.M. (1995). « SUSAN - A New Approach to Low Level Image Processing », *Internal Technical Report TR95SMS1c*, Defence Research Agency, Chobham Lane, Chertsey, Surrey, UK.
- SU, W. Y. et KIM, S. P. (1994). « High-resolution restoration of dynamic image sequences », *International Journal of Imaging Systems and Technology*, **5**(4) : 330–339.
- SUN, L. , XIE, H. , RONG, W. et CHEN, L. (2005a). « Task-Reconfigurable System for MEMS Assembly », *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 844–849.
- SUN, Y. , DUTHALER, S. et NELSON, B. J. (2005b). « Autofocusing Algorithm Selection in Computer Microscopy », *IEEE Intelligent Robotics Systems (IROS)*, Canada, pp. 419–425.
- SUTHERLAND, I. E. (1963). « Sketchpad : A man-machine graphical communications system », *Technical Report 296*, MIT Lincoln Laboratories.
- SUTHERLAND, I. E. , SPROULL, R. F. et SCHUMAKER, R. A. (1974). « A Characterization of Ten Hidden Surface Algorithms », *ACM Computer Surveys*, **6**(1) : 1–55.
- SZELISKI, Richard (1994). « Image Mosaicing for Tele-Reality Applications », *Technical Report CRL 94/2*, Digital Equipment Corporation, Cambridge Research Lab.
- SZELISKI, Richard et SHUM, Heung-Yeung (1997). « Creating full view panoramic image mosaics and environment maps », *Computer Graphics (SIGGRAPH'97 Proceedings)*, Vol. 31, pp. 251–258.
- TOM, B. C. et KATSAGGELOS, A. K. (1994). « Reconstruction of a high resolution image from multiple degraded mis-registered low resolution images », *SPIE Proceedings of the Visual Communications and Image Processing*, Vol. 2308, Chicago, pp. 971–981.
- TORR, P. et ZISSERMAN, A. (1997). « Robust parameterization and computation of the trifocal tensor », *Journal of Image and Vision Computing*, **15** : 591–605.
- TSAI, R. et HUANG, T. (1984). « Multiframe image restoration and registration », dans R. Y. TSAI et T. S. HUANG (éds), *Advances in Computer Vision and Image Processing*, Vol. 1, JAI Press Inc., pp. 317–339.
- UR, H. et GROSS, D. (1992). « Improved resolution from subpixel shifted pictures », *Computer Graphics and Image Processing : Graphical Models and Image Processing*, **54** : 181–186.
- VIKRAMADITYA, B. et NELSON, B. J. (1997). « Visually Guided Microassembly Using Optical Microscopes and Active Vision Techniques », *IEEE International Conference on Robotics and Automation*, pp. 3172–3177.

- VIKRAMADITYA, B. et NELSON, B. J. (1999). « Visually Servoed Micropositioning for Robotic Micromanipulation », *Journal of Microcomputer Applications*, **18**(1) : 23–31.
- WECK, M. et PESCHKE, C. (2004). « Equipment technology for flexible and automated micro-assembly », *Microsystem Technologies*, **10** : 241–246.
- WEDEKIND, Jan (2004). « Focus set based reconstruction of micro-objects », *IEEE International Conference Mechatronics and Robotics*, Aachen, Germany.
- XIAO, J. et SHAH, M. (2004). « Tri-view morphing », *Computer Vision and Image Understanding*, **96**(3) : 345–366.
- XIE, H. , CHEN, L. , SUN, L. et RONG, W. (2005). « Hybrid Vision-Force Control for Automatic Assembly of Miniaturized Gear System », *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1380–1385.
- XIE, H. , RONG, W. et SUN, L. (2006). « Wavelet-Based Focus Measure and 3-D Surface Reconstruction Method for Microscopy Images », *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, pp. 229–234.
- XIONG, Y. et TURKOWSKI, K. (1997). « Creating image-based VR using a self-calibrating fisheye lens », *IEEE Computer Vision and Pattern Recognition (CVPR)*, San Juan, Puerto Rico, pp. 237–243.
- YANG, G. , GAINES, J. A. et NELSON, B. J. (2005). « Optomechatronic Design of Microassembly Systems for Manufacturing Hybrid Microsystem », *IEEE Transactions on Industrial Electronics*, **52**(4) : 1013–1023.
- YANG, Ge , GAINES, James A. et NELSON, Bradley J. (2003). « A supervisory wafer-level 3D microassembly system for hybrid MEMS fabrication », *Journal of Intelligent and Robotic Systems*, **37** : 43–68.
- YEO, T. , JAYASOORIAH, SO et SINNIAH, R. (1993). « Autofocusing for tissue microscopy », *Image and Vision Computing*, **11** : 629–639.
- YESIN, K. B. et NELSON, B. J. (2002). « CAD Model Based Tracking of MEMS Components for 3D Microassembly », *International Workshop on Microfactories*, pp. 81–84.
- YOULA, D. et WEBB, H. (1982). « Image restoration by the method of convex projections : Part 1 - Theory », *IEEE Transactions on Medical Imaging*, Vol. MI-1(2), pp. 81–94.
- YU, S. et NELSON, B. J. (2001). « Autonomous Injection of Biological Cells Using Visual Servoing », *Experimental Robotics VII*, Vol. 271 of *Lecture Notes in Control and Information Sciences*, pp. 169–178.
- ZAPPALÁ, Anthony , GEE, Andrew et TAYLOR, Michael (1997). « Document mosaicing », *Proceeding of the British Machine Vision Conference*, Vol. 17, pp. 589–595.
- ZELLER, C. (1996). *Calibration Projective Affine et Euclidienne en Vision par Ordinateur*, PhD thesis, École Polytechnique.
- ZHANG, C. et CHEN, T. (2003). « A Survey on Image-Based Rendering », *Technical Report TR-AMP-03-03*, Electrical and Computer Engineering Carnegie Mellon University, Pittsburgh.

- ZHANG, Z. , DERICHE, R. , FAUGERAS, O. et LUONG, Q.-T. (1995). « A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry », *Technical report 2273*, INRIA.
- ZHAO, W. (2006). « Flexible image blending for image mosaicing with reduced artifacts », *International Journal of Pattern Recognition and Artificial Intelligence*, **20**(4) : 609–628.
- ZHENG, J. Y. et TSUJI, S. (1992). « Panoramic representation for route recognition by a mobile robot », *International Journal of Computer Vision*, **9** : 55–76.
- ZHOU, Y. et NELSON, B. J. (1999). « Calibration of a parametric model of an optical microscope », *Journal of Optical Engineering*, **38**(12) : 1989–1995.
- ZHOU, Y. , NELSON, B. J. et VIKRAMADITYA, B. (2000). « Integrating Optical Force Sensing with Visual Servoing for Microassembly », *Journal of Intelligent and Robotic Systems*, **28** : 259–276.
- ZHU, Z. , XU, G. , RISEMAN, E. M. et HANSON, A. R. (2006). « Fast construction of dynamic and multi-resolution 360° panoramas from video sequences », *Journal of Image and Vision Computing*, **24** : 13–26.
- ZOGLAMI, I. , FAUGERAS, O. et DERICHE, R. (1997). « Using geometric corners to build a 2D mosaic from a set of images », *IEEE Computer Vision and Pattern Recognition*, Puerto Rico, pp. 420–425.

Index

- A**
- acquisition de l'information 4
 - action et perception 9
 - algorithme génétique 20
 - aliasing 117
 - approche Bayésienne 97
 - asservissement visuel 17, 19, 107, 187
 - auto-calibrage 56
 - autofocus 16, 19, 20, 132
- B**
- backward mapping* ... 44, 45, 118, 146, 177
 - Bayer Bryce E. 6
 - benchmark forward-backward* 72
 - Billboard 46
 - Binnig Gerd 13, 14
 - binoculaire 23
 - biologie 20
 - Blending* 40
 - Boyle Willard 5
 - boîte englobante 128
 - bump mapping* 46
 - Bézier Pierre 8
- C**
- C++ 142
 - Calibrage
 - faible 40, 55–57, 155
 - fort 45, 56, 155
 - mire en damier perpendiculaire 59
 - mire goutte d'eau 60
 - mires 56, 152
 - Calque
 - d'image 45, 48
 - de fusion 122
 - de pondération 121
 - de recouvrement 122
 - de transfert 122
 - Caméra
 - bas prix 17
 - endoscopique 17, 108
 - omnidirectionnel 39
 - à tube 4, 6
 - canal alpha 119
 - capillarité 60
 - Carte
 - de disparité 176, 180
 - de profondeur 46
 - Catmull Ed 8
 - CCD 5, 130
 - cellule biologique 17, 20
 - centre optique 57
 - CMOS 6
 - cohérence géométrique 41, 167
 - computer graphics* 32
 - computer vision* 32
 - concentric mosaics* 36, 37
 - configuration singulière 152
 - contour actif 17, 20
 - Contrainte
 - fronto-parallèle 109, 163, 164
 - épipolaire 58
 - coordonnées 3D 45
 - Correction
 - d'échelle 169
 - de centrage 169

- Correspondance 59, 68
 d'intérêt 68
 dense 40, 51, 68, 175
 par corrélation 69
 par voisinage 75
 quasi-dense 42, 68
- Corrélation 20
 III-ZNSSD 115, 134
 CC 69
 de phase 111
 de transformation 40
 de transformation projective 111
 par stéréo vision 175
 spatiale norme L1 111
 SSD 69
 ZCC 69
 ZNCC 70, 175
 ZNSSD 70, 137, 175
 ZSSD 69
- Critère
 de corrélation 69
 de Rayleigh 12
 Shannon-Nyquist 96
- D**
- depth-from-focus* 17, 19
 dictionnaire 136
 diffraction 104
 disparité 157, 163
 distorsion optique 92
- Distribution
 de Huber 97
 gaussienne 97
- DLT 79, 114, 115
 droite de régression 101
 DSP 10, 17
 dualité point/ligne 57
Dynamosaicing 38
- Décomposition
 de Cholesky 112
 en ondelette 16
 en ondelettes 120
 en valeurs singulières 80, 83
- déformation 3D 45, 46
- dégénérescence de vue 167
- Détecteur
 Canny 63, 162
 CSS 63
 d'arête 17
 de points 62
 Harris 62
 Harris simplexe 64, 66, 115
 SIFT 137
 SURF 137
 SUSAN 63
- Détourage 40, 92
 automatique 126
- E**
- Echantillonnage
 multicanaux 96
 non-uniforme 97
 théorème généralisé 96
- écho d'image 179
- Eclairage
 diascopique 100, 104, 111
 froid 109
 épiscopique 105
- Effet
 d'éclairage 49
 de masquage 175
- épipole 43, 58
- EPROM 114
- équation trilinéaire 147
- espace projectif 57
- estimation robuste 84
- Eugene Frederick 10
- extrapolation de vue 160
- F**
- Faisceau
 de ligne épipolaire 172
 de plans épipolaire 58
- Fenêtre
 III 71
 adaptative 70
 asymétrique 70
 de corrélation 70

- multiple 70
 - réursive 70
 - fibre optique 18
 - fibroscope 17, 18
 - filtrage 19
 - Filtre
 - adaptatif 97
 - de Bayer 6
 - de Kalman 138
 - de mode 119
 - de Wiener 97
 - en fonction du pas 119
 - flexible 120
 - moyen 119
 - médian 119
 - passe-bas 100
 - Fonction
 - de corrélation 69
 - plénoptique 35, 51
 - sigmoïde 183
 - fonction plénoptique 35
 - forward mapping* 116, 172
 - FPGA 10
 - Friese-Greene William 10
 - Fusion 119
 - adaptative 120, 135
 - blending* 92
 - fécondation *in-vitro* 22
- G**
- Gerber Christoph 14
 - Gouraud Henri 8
 - goutte d'eau 60
 - Géométrie
 - trifocale 44, 146
 - épipolaire 43, 56, 146
- H**
- Hooke Robert 11
- I**
- iconoscope 4
 - Illumination
 - ambiante 28
 - de Phong 50
 - globale 28
 - Image
 - de synthèse 32
 - globale 135
 - locale 135
 - mosaïque 16, 39
 - panoramique 37
 - rectifié 41
 - satellitaire 94
 - satellite 39
 - Imageur
 - mobile 139
 - statique 139
 - incrustation 134
 - indice invariant 40, 59, 114, 136
 - infographie 32, 33
 - Interpolation
 - bicubique 99, 106, 117, 174
 - bilinéaire 117
 - d'image 94
 - de vue 40, 160
 - polynomiale 116
 - Iterative Back-Projection* 97
- J**
- Jacobien 112, 173
 - Janssen Hans 11
 - Janssen Zacharias 11
 - Joint view triangulation* 42
- K**
- Knoll Max 12
- L**
- lancer de rayon 44, 50
 - Laplacien pyramidale 120
 - Laser 18
 - trapping 18
 - layer alpha* 119
 - Layered Depth Image (LDI)* 48
 - Leica 20, 52, 184
 - lentille *fisheye* 39
 - Levi-Civita 152

- Light field* 36, 50, 51
- Ligne
 de base 40, 57
 de vue 45
 épipolaire 43, 57
- Line-of-Sight* 44
- Linux 183
- lissage de Gouraud 8, 46
- Logiciel
 de modélisation 33
 MosaicingRTI 142
 TTR 190
- look-and-move* 19, 107, 187
- Lucas-Kanade 135, 140, 190
- Lumigraph* 36, 49
- Lumière
 ambiante 50
 diffuse 50
 spéculaire 50
- M**
- machine à états 180
- manifold mosaics* 39
- mapping* 46
- Marr David 7
- Matlab . 81, 84, 89, 108, 142, 152, 159, 170
- Matrice
 de colinéation 40, 58, 79
 de morphisme projectif 57
 de projection 42
 de symétrie croisée 59
 de transformation projective... 40, 58
 essentielle 155
 fondamentale 57, 82
 Hessienne 112
- Matrox Meteor 108
- maximum *a posteriori* 97
- maximum likelihood* 97, 114
- McMullan Dennis 13
- micro-usine 1
- micromanipulation 1, 129, 183
- Microscope
 aberration chromatique 22
 champ de vision 22, 26
 confocal 12
 contraste 27
 cône d'acceptance 24
 distance de mise au point 22
 distance de travail 24, 25
 distance focale 21
 encombrement 22, 23
 grandissement 23
 illumination 22, 27
 lentille mince 21
 mini 17
 objectif 21
 objectif Mitutoyo 29
 oculaire 21
 ouverture numérique 24
 photonique 11
 photonique en champ proche 12
 photonique inversé 20
 profondeur de champ 22, 25, 61
 propriétés 22
 stéréoscopique 18, 20
 vision nette 22
 à balayage par sonde 13
 à effet tunnel 13
 à force atomique 14
 électronique en transmission 12
 électronique à balayage 12, 17
 électronique à balayage par
 transmission 12
- microscopie 11
- microvision 14
- Minsky Marvin 12
- mire calibrée 94
- modeleur 33
- modified painter's algorithm* 180
- Modèle
 affine 107
 CAO 17
 de dégradation 97
 lambertien 50
 plénoptique 36
 à partir d'images 32, 33
- modélisation par texturage 46
- moindres carrés médians 101

- moment d'une image 106
- montage stéréoscopique 30
- Moravec Hans 9
- morphage de vue 40, 41
- morphisme projectif 57
- mosaicing* 36, 37, 39, 51, 91
- Mosaïque 19
- matricielle 130
- Méthode
- d'optimisation 65
- de relaxation 76
- de Zhang 163, 172
- des 4 points 80
- des 6 points 150
- des 8 points 82
- des moindres carrés 101
- des moindres carrés médians 101
- directe 111
- N**
- nanotechnologie 14
- Newell Martin 8
- Norme
- de Frobenius 83
- L1 111
- Novel View Synthesis* 31
- nuage de panneaux 45, 46
- O**
- Oatley Charles 13
- objet factice 161, 162
- occultation 48, 177
- Octave 81, 84, 89, 152, 159, 170
- onde de De Broglie 12
- OpenCV 142, 190
- OpenGL 190
- optical flow* 40, 135
- P**
- painting bridge* 180
- panoramique stéréoscopique 38
- Parallaxe 58, 92
- projectif 58, 147
- relatif 58, 148
- paramètre intrinsèque 40, 44, 155
- peigne de Dirac 71
- perception distribué 30
- Phong B. 50
- photo-réalisme 49
- pick-and-place* 19
- Pirex 104
- plan projectif 57
- Pohl D. W. 12
- point d'intérêt 56, 59, 62
- post-processing* 70, 179, 180
- Projection Onto Convex Sets* 97
- PRONOMIA 2, 17
- Python 183
- Q**
- quadrant 126
- Quate Calvin 14
- Quicktime 39
- R**
- RANSAC 42, 84, 114, 115
- ray tracing* 44, 50, 51
- rayonnement thermique 104
- Recalage 92
- géométrique 40
- photométrique 40
- Reconstruction
- 2D 17, 52
- 3D 33
- rectification épipolaire 41, 172
- relation métrique/pixel 94, 103
- Rendu à partir d'images 32, 34
- avec géométrie explicite 51
- avec géométrie implicite 34, 51
- explicite 34
- sans géométrie 34, 35, 51
- rendu à partir de modèle 32, 33
- reprojection 167
- restauration d'image 95
- Roberts Lawrence G. 7
- Rohrer Heinrich 13
- ROI 135
- RS232 130

Ruska Ernst	12
Réflexion.....	49
glacé.....	50
spéculaire.....	28, 50, 70, 92
réfraction	49

S

SAMMI.....	1, 2, 17, 23, 52, 184
Scheinman Victor	10
<i>seam</i>	92
segmentation.....	20, 137
sensori-psycho-moteur.....	4
Shakey.....	9
<i>shift-and-add</i>	97
SIFT	137
signature d'objet.....	137
simplexe de Nelder-Mead	66, 162, 166
Smith George	5
SMS60.....	108, 130
solveur <i>Symmetric Positive Definite</i> ...	112
sous-fenêtrage.....	137
<i>splatting</i>	180
<i>sprite</i>	46
<i>stitching</i>	94
stéréo-anaglyphe.....	38
stéréoscopie.....	10
sub-pixélique.....	94, 175, 185
sub-pixéllique.....	97
suivie d'objet.....	17, 20, 135
super-résolution	94, 95
supervision.....	20, 139
SURF	137
Sutherland Ivan.....	8
Synge Edward Hutchinson	12
synthèse de nouvelle vues.....	31, 32
Systeme	
distribué.....	30
distribué monobloc.....	51
distribué temps déporté.....	30
distribué temps réel.....	30
distribué éclaté.....	51
pseudo distribué.....	30
temps déporté.....	51
temps réel.....	51

T

technique de panneaux.....	46
Tenseur	
d'amorçage.....	151
trifocal.....	45, 150
trilinéaire.....	149, 150
tension de surface.....	60
texturage par peinture.....	45
Texture.....	45
bosselé.....	46
par dépendance de vue.....	45
photométrique.....	46
relief.....	46
théière de Newell.....	9
TIMM400.....	17, 51, 130, 139, 184
tomographie	17
<i>Toolbox</i>	
Kovesi.....	89
MCV.....	89
Traitement	
de l'image.....	7
de l'information.....	6
Transfert.....	43
par colinéation.....	40, 43
trifocal.....	40, 52, 145
épipolaire.....	40, 43
Transformé	
de Fourier.....	96
de Fourier discrète.....	96
de Hough.....	19
en ondelette.....	16, 120
transparence.....	49
triangulation de Delaunay.....	42
trilinéarité.....	147

V

variance normalisé.....	132
verre borosilicate.....	104
<i>VideoBrush</i>	39
vidéo panoramique.....	38
<i>view morphing</i>	41
vignelage.....	92
vision artificielle.....	6, 8, 31-33
Von Ardenne Manfred.....	12

vote probabiliste 84

Vue

active 51

globale 27

locale 27

par triangulation commune 40, 42

passive 51

réelle 31

virtuelle 31, 151

W

wafer de silicium 28

Wheatstone Charles 10

Z

z-buffer 176

Zworykin Vladimir 4, 12

Résumé

La combinaison de microscope photonique et de caméra est largement utilisée dans les applications de micro-assemblage. Cet instrument indispensable comporte cinq propriétés : un faible champ de vision, une faible profondeur de champ, une faible distance de travail, une forte dépendance à l'éclairage et un encombrement important. La conséquence directe de ces propriétés dans un contexte de micro-assemblage est l'utilisation de système de vision distribué composé d'imageurs de caractéristiques différentes et complémentaires qui ont un coût et un encombrement important. C'est pour cela que notre approche consiste à reconstruire certaines vues indispensables au contrôle de la station par des techniques de rendu d'images, limitant ainsi le nombre de vidéo microscopes photoniques réels dans la station. Pour cela, nous proposons d'utiliser deux techniques, la construction d'images mosaïques et le transfert trifocal.

Avant toute chose le système de vision doit être calibré, nous présentons les étapes du calibrage stéréoscopique faible pour ce contexte de microscopie photonique ainsi que leurs améliorations par de nouveaux algorithmes.

La construction d'images mosaïques est une technique qui permet de reconstruire une image complète d'une scène à partir d'un ensemble d'images représentant chacune une petite partie de cette scène. Après avoir présenté de nouveaux algorithmes en vue d'améliorer la construction de mosaïques nous l'utilisons pour la supervision de station de micromanipulation.

Le transfert trifocal est une technique qui permet à partir d'un simple calibrage stéréoscopique faible de reconstruire une vue virtuelle à partir de seulement deux vues réelles sans le besoin d'information 3D explicite. Après un éclaircissement sur la théorie et la proposition de nouveaux algorithmes nous l'utilisons dans une boucle d'asservissement visuel de type *look-and-move*, pour contrôler le déplacement d'une micropince. La vue de côté virtuelle permettant le contrôle est reconstruite à partir des vues réelles provenant d'un stéréo microscope photonique.

Mots-clés : micromanipulation, micro-assemblage, microscope photonique, rendu à partir d'images, calibrage faible, construction de mosaïques, transfert trifocal.

Abstract

The combination of photon microscope and camera is used in the applications of micro-assembly. This essential instrument have five properties : a weak field of view, a low depth of field, a short working distance, a strong dependence to illumination and bulkiness. In that context of micro-assembly the vision system should be distributed and composed of complementary imagers are expensive and bulky. Therefore our approach consists in obtaining the essential views for the control of the station by the image based rendering. It decreases the number of photonic microscopes in the station. For that, we propose to used two techniques, the mosaicing and the trifocal transfer.

First the vision system calibration is required, we present the stages of stereoscopic weak calibration for that context of photonic microscopy with improved new algorithms.

Image mosaicing is a technique for the reconstruct a wide image of scene from a set of images that scene where each represents a small part of this scene. After new algorithms improving the construction of mosaic we use this technique for the supervision of micromanipulation station.

Trifocal transfer is a technique based on stereoscopic weak calibration for the reconstruction of a virtual view from only two real without 3D explicit information. After an explanation on the theory and a new algorithm proposed we used this technique in a the visual servoing application, to control the position of a microgripper. The virtual view allowing control is obtain from the real view coming from a stereoscopic photonic microscope.

Keywords : micromanipulation, micro-assembly, photonic microscope, image based rendering, weak calibration, mosaicing, trifocal transfer.