Couplage mouvements articulées - vision stéréoscopique : une approche projective

Andreas Ruf

INSTITUT NATIONAL POLYTECHNIQUE (INP) de GRENOBLE

THESE

pour obtenir le grade de
DOCTEUR de l'INPG

Ecole Doctorale: **Mathématique, Science et Technologie de l'Information**
Spécialité: **Informatique (Imagerie, Vision et Robotique)**

présentée et soutenue publiquement le 15. février 2001 par

Andreas RUF

Closing the loop between articulated motion and stereo vision: a projective approach

Thèse préparée au sein du laboratoire GRAVIR–IMAG et INRIA Rhône-Alpes
sous la direction de

Radu HORAUD

Jury :

Président :	**Philippe** CINQUIN
Rapporteurs :	**François** CHAUMETTE
	Gerald SOMMER
Examinateurs :	**James** CROWLEY
	Olivier FAUGERAS
Directeur de thèse:	**Radu** HORAUD

Acknowledgements

I'm very grateful to Radu Horaud for advising me during the time of this thesis. He indicated directions which turned out very fruitful, and generously offered the liberty and the time I needed to explore and to conclude.

I'd like to thank the reseachers in the MOVI project for their cordial reception and support. In particular, I'd like to thank in particular Gabriella, to whom I owe a quick start and many open discussions, Bill and Peter, who selflessly agreed to proof-read the manuscript during a very busy season, and Frederick, to whom I owe excellent experimental results thanks to his DEA project.

I'd like to thank the jury for their thorough work, and especially the reviewers Gerald Sommer and Francois Chaumette for the quick and careful reading, and thoughtful comments.

Finally, there are all the friends who accompagnied and carried me through the ups and downs of life during this period, und meine Familie, bei der ich mich dafür entschuldige, dass wir uns zu selten sehen konnten.

Couplage mouvements articulés — vision stéréoscopique: une approche projective

Résumé *Cette thèse propose un nouvel éclairage sur le couplage de la vision stéréoscopique avec le mouvement articulé. Elle développe tout particulièrement de nouvelles solutions s'appuyant sur la géométrie projective. Ceci permet notamment l'utilisation des caméras non-calibrées.*

La contribution de cette thèse est d'abord une nouvelle formulation géo - métrique du problème. Il en découle des solutions pratiques pour la modélisation, l'étalonnage, et la commande du système. Ces solutions suivent une approche "coordinate-free", généralisées en une approche "calibration-free". C'est à dire, elles sont indépendantes du choix des repères et de l'étalonnage des caméras.

Sur le plan pratique, un tel système projectif peut fonctionner sans connaissance a priori, et l'auto-étalonnage projectif se fait de manière automatique. Les modèles proposés sont issues d'une unification projective de la géométrie du système plutôt que d'une stratification métrique. Pour un asservissement visuel, les modèles obtenus sont équivalents aux modèles classiques. D'autres questions pratiques ont ainsi pu être abordées d'une nouvelle manière: la commande de trajectoires visibles et méchaniquement realisables dans l'espace de travail entier.

Sur le plan théorique, un cadre mathématique complet est introduit. Il donne á l'ensemble des objets impliqués dans un asservissement visuel une représentation projective. Sont notamment etudiés, l'action des joints rotoïdes et prismatiques, les mouvements rigides et articulés, ainsi que la notion associée de vitesse projective. Le calcul de ces représentations est de plus explicité.

mots clés: asservissement visuel, stéréo non-calibrée, cinématique
spécialité: Informatique (Imagérie, Vision, Robotique)

GRAVIR-IMAG
INRIA Rhône-Alpes, movi
655, avenue de l'Europe
38330 Montbonnot St.-Martin

Closing the loop between articulated motion and stereo vision: a projective approach

Abstract *In this thesis, the object of study are robot-vision systems consisting of robot manipulators and stereo cameras. It investigates how to model, calibrate, and operate such systems, where operation refers especially to visual servoing of a tool with respect to an object to be manipulated.*

The main contributions of this thesis are to introduce a novel, "projective" geometrical formalization of these three problems, and to develop novel coordinate- and calibration-free solutions to them.

The impact in practice: the "projective" systems can operate with less priori knowledge. Only metrically unstratified but projectively unified parameters are needed to "calibrate" a fully operational model of the system. This can be done automatically with a self-calibration method. As far as visual servoing is concerned, the projective models are equivalent to classical ones. Also, their particular properties allow new answers to open questions in visual servoing to be found, such as how to control trajectories that are at the same time visually and mechanically feasible over the entire work-space.

The impact on theory: a novel and complete projective framework has been introduced, that allows calibration-free representations for all basic geometrical objects relevant to visual servoing, or to camera-based perception-action cycles in general. Most important are the projective representations of rigid- and articulated motion, as well as the respective notion of projective velocities. Effective computational methods for the numerical treatment of these representations were also developed.

keywords: visual servoing, uncalibrated stereo, coordinate-free kinematics

GRAVIR-IMAG
INRIA Rhône-Alpes, movi
655, avenue de l'Europe
38330 Montbonnot St.-Martin

Kopplung von artikulierter Bewegung und Stereosehen: ein projektiver Ansatz

Zusammenfassung *Die vorliegende Dissertation betrachtet Hand-Auge-Systeme bestehend aus einem Roboterarm und einer Stereokamera, mit dem Ziel einer sichtsystemgestützten Führung und Regelung des Roboters. Die Fragestellung umfaßt dabei die Umsetzungschritte Modellierung, Kalibrierung und Betrieb eines solchen Systems, wobei die Positionierung von Werkzeugen bezüglich der zu manipulierenden Werkstücke als Anwendungsbeispiel herangezogen wird.*

Der wissenschaftliche Beitrag besteht in der Formulierung eines neuartigen Ansatzes zur mathematischen Beschreibung der Systemgeometrie, dessen formale Darstellungen nur auf Mittel der projektive Geometrie zurückgreift. Dazu kommen durchgängig projektive Lösungen für die drei Umsetzungsschritte. Sie zeichnen sich insbesondere durch Unabhängigkeit sowohl von der Wahl des Koordinatensystems als auch von der Kamerakalibrierung aus.

Auswirkungen auf die Praxis können entsprechend "projektive" Hand-Auge-Systeme haben, die auch ohne Vorgaben zur Systemkalibrierung in Betrieb gehen können. Hierzu wird die zuvor noch in metrische und projective Parameter aufgetrennte Kalibrierung durch eine vereinheitlichte projektive Parametrierung ersetzt, die automatisch mit einer Selbstkalibriermethode bestimmt wird. Beide Parametrierungen sind, zumindest zum Zwecke der bildbasierten Regelung, gleichwertig. Weiterhin ergab diese Parametrierung auch neue Antworten auf noch offene Fragestellungen, wie die Roboterführung entlang von wohldefinierten Trajektorien, die sowohl unter mechanischen als auch unter visuellen Gesichtspunkten vom Hand-Auge-System realisierbar sind.

Auswirkungen auf die Theorie liegen in dem erstmals eingeführten projektiven Formalismus zur kalibrierunabhängigen Darstellung, welcher alle grundlegenden Bestandteile von Hand-Auge-Systemen umfaßt und somit als vollständig bezeichnet werden kann. Hervorzuheben sind hierbei die projektive Darstellung starrer und artikulierter Bewegungen sowie der entsprechenden Geschwindigkeitsbegriffe. Der theoretische Beitrag wird noch aufgewertet durch die Formulierung und Umsetzung der rechnerischen Methodik zur numerischen Behandlung der projektiven Darstellungen.

Contents

List of Figures

List of Publications

Journals

Andreas Ruf and Radu Horaud. Visual servoing of robot manipulators, Part I : Projective kinematics. *International Journal on Robotics Research*, 18(11):1101 – 1118, November 1999.

Conferences

Tutorial
Andreas Ruf and Radu Horaud. Coordination of Perception and Action: the projective approach. *European Conference on Computer Vision*, Dublin, Ireland, June 2000.
`http://www.inrialpes.fr/movi/people/Ruf/rufECCV00b.pdf`

Invitation
Andreas Ruf. Representation of Perception-Action Cycles in 3D projective frames. *Algebraic Frames for the Perception-Action Cycle (AFPAC'00), Kiel, Germany, September 2000.*
`http://www.inrialpes.fr/movi/people/Ruf/rufAFPAC.pdf`

Papers
Andreas Ruf and Radu Horaud. Vision-based guidance and control of robots in projective space. In Vernon David, editor, *Proceedings of the 6th European Conference on Computer Vision (ECCV'00)*, volume II of *Lecture Notes in Computer Science*, pages 50–83, Dublin, Ireland, June 2000. Springer.

Andreas Ruf, Frédérick Martin, Bart Lamiroy, and Radu Horaud. Visual control using projective kinematics. In John J. Hollerbach and Daniel E. Koditscheck, editors, *Proceeding of the International Symposium on Robotics Research, (ISRR'99)*, Lecture Notes in Computer Science, pages 97–104, Salt Lake City - UT, October 1999. Springer.

Andreas Ruf and Radu Horaud. Rigid and articulated motion seen with an uncalibrated stereo rig. In Tsotsos John, editor, *Proceedings of the 7th International Conference on Computer Vision (ICCV'99)*, volume II, pages

789 – 796, Kerkyra, Greece, September 1999. IEEE Computer Society.

Andreas Ruf and Radu Horaud. Projective rotations applied to non-metric pan-tilt head. In *Proceedings of the Intl. Conf. on Computer Vision and Pattern Recognition, (CVPR'99)*, volume I, pages 144 – 150, Fort Collins, CO, June 1999. IEEE Computer Society.

Andreas Ruf, Gabriella Csurka, and Radu Horaud. Projective translations and affine stereo calibration. In *Proceedings of the Intl. Conf. on Computer Vision and Pattern Recognition, (CVPR'98)*, pages 475 – 481, Santa Barbara, CA, June 1998. IEEE Computer Society.

Gabriela Csurka, David Demirdjian, Andreas Ruf, and Radu Horaud. Closed-form solutions for the Euclidean calibration of a stereo rig. In Burkhardt Hans and Neumann Bernd, editors, *Proceedings of the 5th European Conference on Computer Vision, (ECCV'98)*, volume 1, pages 426 – 442, Freiburg, Germany, June 1998. Springer Verlag.

Workshops

Ruf Andreas. Visual Trajectories from Uncalibrated Images. at *Intl. Conf. on Intelligent Robots and Systems, (IROS 1997)*, In *Proc. of Workshop on New Trends in Image-based Robot Servoing, pages 83 – 91, Grenoble, France, September 1997. INRIA.*

Ruf Andreas. Non-metric geometry of robotic manipulation. at *European Conference on Computer Vision, (ECCV 1998)*, Workshop on Visually guided robots: Current research issues and industrial needs Freiburg, Germany, June 1998.

Chapter 1

Introduction

1.1 Motivation

The human capacity to control movements based on visual perceptions is called **hand-eye coordination**. By analogy, a technical system build from video cameras, robotic actuators, and computing devices should also be capable of hand-eye control. A technology called *"visual servoing"* implements such *"perception-action-cycles"* as a closed-loop robot control, where feedback and set-points are video-images captured on-line.

Human hand-eye coordination nicely exemplifies the various characteristics and potentials of this paradigm. From primitive tasks like reaching and grasping, through still simple tasks like tracking, pointing, and shooting, to skilled craftsmanship and precision manufacturing, the various complexities of hand-eye skills remind of an evolution from gatherer, through hunter or warrior, to craftsmen. The latter are a good example how humans acquire skills very efficiently from learning-by-doing or, more precisely, from learning-by-seeing. For example, photos, sketches, or exploded-views in assembly instructions for take-away furniture or construction-toys (Fig. 1.1) can be seen as high-level "programs" for sequences of hand-eye skills, that almost anybody can follow. Finally, standardized engineering-drawings, CAD/CAM file formats, and in particular the connection of CNC machining with vision-based quality-control marks the transition from human towards technical hand-eye-coordination.

Technical hand-eye-coordination should ideally be as flexible, effective, and intelligent as its human counterpart, while retaining the speed, efficiency, and power of machines. These dual goals mean that hand-eye-coordination is at the crossroads between artificial intelligence and machine vision in computer science, robot- and video-technology in engineering science, and fundamental control theory, mechanics, and geometry in applied mathematics. In very general terms, the overall aim is to produce a desired 3D motion of the manipulator by observing its position and motion in the

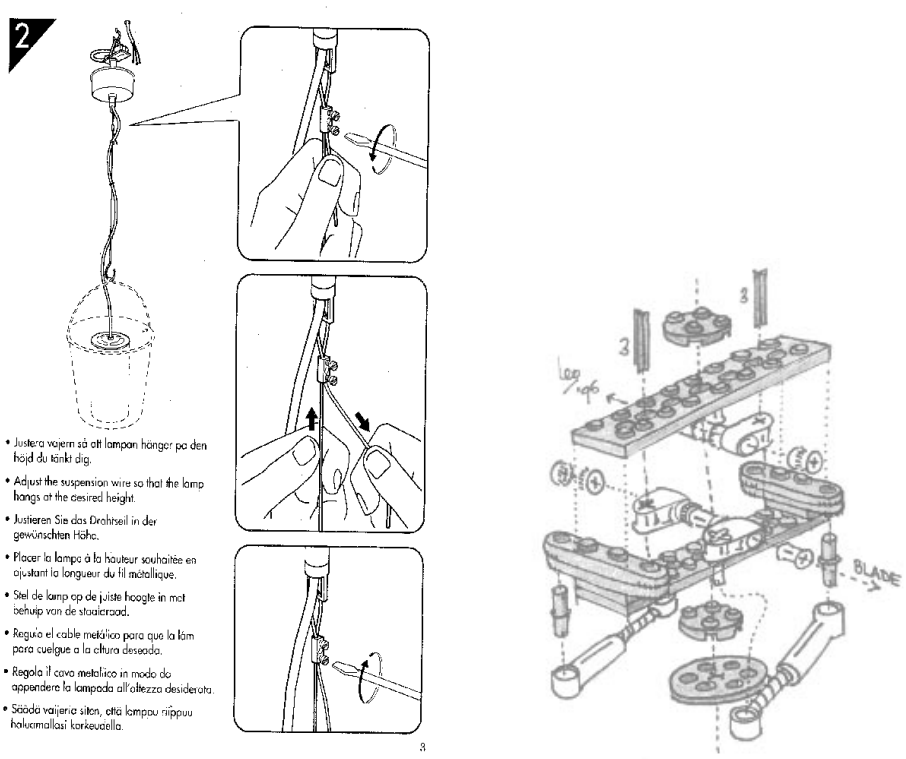

Figure 1.1: A hand-eye sketch from an assembly manual for a do-it-yourself lamp (left), and an exploded-view describing a construction-toy (right).

images, and using these observations to calculate suitable controls for the actuators. For expressing the interaction between perceptions and actions, it is therefore necessary to model the geometry of the system, and to calibrate it.

From a merely scientific point-of-view, **Euclidean geometry** provides ready-to-use concepts like rigid motion, kinematics, and linear camera optics, which make the geometric modeling of the robot-camera system straightforward. However, "*calibration*" – identifying all parameters such a **visual servoing system** under real-world conditions – remains a difficult and time-consuming problem, both, technically and methodologically.

Distances measured in a Euclidean ambient space allow for the control-error to be defined in workspace, for it to be visually measured, and for a corresponding workspace-based control-law to be derived. Whenever such a control-loop servos the robot actuators to drive the distance-error to zero, it is ensured that the hand-eye-coordinated action or motion has achieved to desired target. Think of an assembly task specified for instance in terms of a drawing, e.g. of take-away furniture. It would be considered completed as soon as the video-image shows the various parts to have the desired align-

ment. Here, all the visual servoing loop does is to control the robot towards a position at which the video image captured by the on-line camera is in perfect alignment with the default image on the drawing. Obviously, this requires the on-line camera to be calibrated with respect to the drawing. This is related to the problem of camera calibration, for which various approaches already exist.

Off-the-shelf calibration uses manufacturer data, calibration devices, and on-site manual measurements to identify and estimate the model parameters. Although the results obtained often lack stability and accuracy, and in some cases only a coarse guess is possible, the robustness inherent in the hand-eye approach ensures that the control still converges in most cases. However, systems that require accurate high-speed tracking of complex trajectories for global operability across the whole 3D-workspace, usually are only feasible if the quality of calibration is better than coarse.

More **recent calibration techniques**, especially for self-calibration of video cameras, use recent results in "uncalibrated" computer vision to identify camera parameters without prior knowledge of either the present environment, or the factory-given parts of the system. Although these techniques are rapidly becoming mature, they apply only to the "eye", while the "hand", the arm, and in particular the hand-eye relation still require an external calibration in form of a sophisticated procedure, called "*hand-eye-calibration*". Moreover, a prior kinematic model of the robot is still required, e.g. from the manufacturer, deviations from which greatly influence the overall coherency and accuracy of the system and its calibration.

Coordinate-free methods seek formal representations, distance metrics, and computational algorithms that are independent of ad-hoc choices of coordinate-systems. Geometrically, the latter often refer to a profusion of frames associated with the various components to be modeled. For instance, a classical model would allocate frames on robot-base and -hand, on each linkage, on the tool, and on the camera. The hand-eye link or Denavit-Hartenberg parameters of each linkage are examples of representations that fail to be coordinate-free. A coordinate-free representation, in contrast, would be based on joints rather than linkages, and would represent them by their respective group actions on the ambient three-space. Such representations often have fewer singularities, and greater generality, giving a theory that is both more powerful and more flexible, revealing the inherent nature of the underlying problem.

From an abstract scientific point of view, **projective geometry** describes the geometric space of optical rays through a center of projection, and provides the mathematical concepts for perspective projection, geometrical infinity, coordinate-systems based on cross-ratios, and respective perspective transformations. Recent research in computer vision has successfully used these tools to model "uncalibrated" cameras, and to coherently describe the geometry of multiple uncalibrated cameras. In this context, "uncalibrated"

means "calibration-free", i.e. the metric parameters describing the camera lens, the physical dimensions of the CCD, and their relation with the frame grabber output are entirely unknown. These metric parameters are replaced by projective ones, such as the linear product with a projection matrices. For instance, the epipolar geometry of an uncalibrated camera pair can be encoded as a pair of projective matrices, which in turn allow stereo triangulation of the environment in a three-dimensional projective space.

Image-based methods are a class of techniques, for which projective calculations are particularly well-suited.

For instance, two images of a scene are equivalent to a 3D projective reconstruction. Images taken from different view-points, using possible different lenses or cameras, or different acquisition systems, can be related to the original one by a projective transformation. As everything can so be expressed in terms of images coordinates, the metric information does somehow "cancel out". In this sense, the metric calibration and the displacement of the cameras become transparent, i.e. they no longer appear explicitly in the equations.

For example, as soon as a pair of images depicts an object-tool alignment, it corresponds to a **calibration-free** representation of the underlying assembly task, which is already well-suited for a visual servoing to be applied.

Calibration- and coordinate-free methods are analogous in the sense that they aim produce algebraic frames suitable for representing the geometry underlying the problem, and sometimes even geometric algebras naturally arising from it. In hand-eye-coordination, they are complementary, in the sense that coordinate-free representations of the action domain and calibration-free representations of the perception domain cover both aspects of hand-eye coordination. Moreover, the similarity of the two paradigms suggests that their fusion is worth investigating, in particular in the context of stereo-based hand-eye systems.

1.2 Problem formulation

The thesis seeks to give new solutions to the problem of hand-eye coordination, with special emphasis on the visual servoing of a robotic cell consisting of a stereo camera and a manipulator arm (Fig. 1.3). One common application is grasping an object from camera observations of the work-space (Fig. 1.2). In contrast to existing approaches, the general case of "uncalibrated" cameras is considered, which necessitates the integration of projective geometry with known techniques for visual servo control. The uncalibrated stereo rig spans a projective ambient space, whereas robot kinematics are commonly formulated in Euclidean space. One can ask whether a projective analogue to classical kinematics exists, and whether such an approach is

Figure 1.2: "*Visual servoing*" stands for hand-eye coordination in a closed control loop. Its set-point is a target image e.g. of a grasp, and its error-signal is usually the distance between current and target image.

computationally feasible. More generally, can the integration of calibration-free vision and coordinate-free mechanics bring about a unification of representations, as well ?

A detailed formulation of the problems investigated is as follows:

- Given an uncalibrated stereo system capturing projective observations of a dynamic scene, is a notion of "projective motion" well-defined, and how can it be represented, calculated, and accurately estimated? How can this be done for general rigid-object motions, for elementary robot-joint motions, and for general articulated motions? Given an articulated mechanism, is a notion of "projective kinematics" well-defined, how can it be represented, calculated, and accurately estimated – "projective robot calibration" ? Can the forward and inverse kinematics be solved projectively, i.e. in terms of mappings between joint-space and projective-space motions? What are the infinitesimal analogues of this?

- Given a robot manipulator operating inside a projectively observed workspace, is a notion of "projective control-error" well-defined? How can a law for "projective servo control" be derived and calculated? Does it converge? How can this be done for set-points and features in a single image, in an image pair, and directly in projective space? Considering joint-velocity control, is a notion of "projective dynamics" well-defined[1], how can projective velocities be represented, calcu-

[1]We will consider only first-order motion parameters, here. Acceleration and forces will not be taken in account.

lated, and applied? Velocities of rigid-object motion, of points, and also of planes? Based on a dynamic model of the interaction between joint-space and image-space, can the forward dynamics, mapping joint-velocities onto image-space velocities, and the inverse dynamics, mapping the image-velocities required to annihilate the control-error onto the required joint-velocity command be solved projectively?

- Given an alignment task in a calibration-free representation, is a notion of "projective trajectory" well-defined, and how can it be represented, calculated, and adapted to the task? Given an alignment of all six degrees of freedom of a rigid body, is a notion of "projective Cartesian motion" [2] well-defined, and how can it be represented, derived, and calculated? Based on the object's projective displacement or alternatively on the projective dynamics of object points and -planes, how can the alignment be decomposed into projective translations and projective rotations, and can the robot motions that drive these elementary motions be used to produce the desired Cartesian trajectory? Is the solution still image-based or can a "direct projective control" be done? Can the trajectory approach cope with restricted visibility, large rotations, and global validity in work-spaces?

1.3 Organization of the manuscript

The manuscript is organized in 7 core chapters, enclosed by an introductory and a concluding chapter, followed by a number of appendices. Below, a chapter-by-chapter overview on contributions and impact is given. The organization of the chapters themselves follows the general guideline: algebraic characterization, geometric interpretation, analytical and differential properties, computational and numerical methods, experimental validation and evaluation.

- **Fundamentals**: In this precursory chapter, the concepts fundamental to this thesis are recapitulated and restated using the terminology and formalisms adopted also in the succeeding chapters. Basically, these are the projective geometry of uncalibrated stereo vision, and respective 3D-reconstruction in projective space. The rigid-stereo assumption is made, on which most results of this thesis are based. Equally important are the geometry and representations of the displacement group as well as its differential geometry and the velocities of rigid motion. The integration of these two concepts then allows the notion of projective motion to be defined as the homographic transformations arising from rigid motion observed in a projective space, i.e. captured

[2] Cartesian motions move a fixed point on the body along a straight line and rotate the body only about this point.

Figure 1.3: Modelling of a robotic work-cell using a stereo camera pair.

with a rigid stereo rig. Finally, the basic computational methodology for calculating reconstructions and homographies is given in great generality, allowing for instance also line and plane features to be considered.

- **Projective Translations**: This chapter introduces the calibration-free representation for the projective translation group, and characterizes the properties of this Lie group and its Lie algebra. It provides efficient computational techniques for calibration, forward- and inverse kinematics of prismatic joints, as well as for affine stereo calibration, which range from algebraic closed-forms, linear methods, to numerical optimization. The methods have been validated and evaluated experimentally.

- **Projective Rotations**: This chapter introduces the coordinate- and calibration-free representation for the projective rotation group, and characterizes the properties of this Lie group and its Lie algebra. "Rotation" refers to an axis which may have an arbitrary position in space, e.g. the single joints of a six-axis industrial manipulator. Efficient computational techniques for calibration, forward- and inverse kinematics of revolute joints are provided, as well as for a pan-tilt actuated stereo rig. The methods range from algebraic closed-forms, linear methods, to numerical optimization. The basic method is validated and evaluated experimentally on a pan-tilt mechanism.

- **Projective Kinematics**: This chapter combines the results of the two preceding ones to introduce coordinate- and calibration-free representations for projective articulated motion, in form of a product-of-exponentials. This comprises projective representations for the static as well as dynamic kinematics of a robotic mechanism. The latter has the form of a Jacobian relating joint-velocities to velocities of projective motion, or to image velocities. The chapter provides efficient computational techniques for stereo-based self-calibration of a robot manipulator, and for its forward kinematics as well as forward and inverse kinematics of the velocity model. The methods range from the sound analytic form of the Jacobian and forward kinematics, over a linear matrix representation of the joints-to-image Jacobian, to a large bundle-adjustment for the non-linear refinement of the calibration accuracy. Additionally, a stratification of the kinematics is investigated. The methods are validated and evaluated experimentally in a self-calibration experiment of a six-axis industrial manipulator.

- **Projective Inverse Kinematics**: This chapter relies on the developments in the preceding chapters and details solutions to the inverse kinematic problem for standard industrial manipulators. The contribution is to solve this problem from a projective kinematic model, only, which is more difficult than in the metric case, since only angular but no distance measures are available. It is a projective solution which renounces stratifying the representations. The main interest is in finding a restricted initial solution for initializing a subsequent numerical refinement. Moreover, the modular geometric solution had a strong impact on the subsequent chapter, in particular on the algorithm for finding the elementary components of an alignment task.

- **Projective Trajectories**: Two approaches to generate the trajectories of Cartesian motions from projective data are developed. The first is based on a projective displacement and affine stratification to identify a translational component. The second is based on projective velocities, and on constraining the motion of a face on the object under consideration. The same trigonometric formulae as in the inverse projective kinematics are arising. Cartesian trajectories are then created as a product-of-exponentials of projective translations and one or two concentric rotations. Additionally, the visibility of the face is taken into account. The methods are tested on simulated realistic data.

- **Projective Visual Control**: Two distinct approaches to visual robot control are developed. The first consists in image-based visual servoing based on a projective kinematic model. It is presented in great generality where the numerous varieties of this approach are shown as special cases of a coordinate-free formulation of the system. In partic-

ular, the equivalence of the hand-eye and independent-eye systems is shown. Similarly, a calibration-free formulation is possible, assuming nothing more than a rigid structure associated with the tool, while camera and robot configurations now may vary.

Besides the 2D-approaches, also a 3D approach can be considered, where a projective instead of a Euclidean space requires to revisit the definition of control-error in 3D-points.

Second, the results of the trajectory chapter have given rise to a directly computed feed-forward for trajectory control of Cartesian motion. The computations essentials correspond to those done in trajectory generation, while the remaining degrees-of-freedom are considered as a feedback error, which can be used to avoid singularities or constrained visibility.

The basic results are validated on a real visual servoing system, and the advanced techniques are tested against simulated data.

Notations

For a complete reference on the notations employed throughout this thesis, please refer to appendix B. Here, only the basic typesetting conventions for equations shall be stated. Plain Roman type stands for scalars a, b, c, indices i, j, k, and coefficients h_{21}, k_{12}, where Greek letters denote angles α, θ and scale-factors γ, λ, μ. Bold type stands for matrices $\mathbf{T}$, $\mathbf{H}$, $\mathbf{J}$, $\mathbf{P}$, which sometimes are accentuated with $\bar{}$, $\hat{}$, $\check{}$. Bold italic stands for vectors $\boldsymbol{q}$, $\boldsymbol{M}$, $\boldsymbol{S}$, $\boldsymbol{m}$, $\boldsymbol{s}$, where in the context of point-vectors, uppercase denotes 3D-points while lowercase denotes 2D-points. Finally, calligraphic type is used for naming frames, e.g. $\mathcal{E}$, $\mathcal{A}$, $\mathcal{P}$.

1.4 State-of-the-Art

The following synopsis on the state-of-the-art aims to situate this work with respect to the main approaches and the key-papers in the field. It moreover aims to relate it to previously published work which has inspired or influenced the research, and to distinguish it from approaches that show a certain degree of similarity. Since an innovation coming out of this thesis is a novel combination of robot control and manipulator kinematics with computer vision and projective geometry, the bodies of knowledge proper to both these fields have been of equal importance.

Concerning the research literature on visual robot control, please consult [42] and references there, or [9] in the quite representative collection [33], for an exhaustive overview on the research literature on visual robot control. Above that, the theoretical concepts underlying task- and sensor-based control are abstracted and formalized as the "*task-function approach*" introduced in the textbook [65]. Finally, two rather geometrical textbooks on robotics and kinematics, [66] and [57], have been a rich source of inspiration, and are quite close to the paradigms adopted in this thesis. More classical treatments are [5], [10], or the very didactic, but concise [55]. Additionally, the related problem of robot calibration is well covered by [56].

Concerning literature on computer vision or formerly "robot vision" [40], a number of excellent textbooks are already available for an in-depth and rather complete coverage of nowadays available solutions and computational methods: [37], [20], or [45] and more recently [29], which are very close to the scientific approach adopted in this thesis. In addition, a new collection of research work concerned with unifying the geometric approaches to robotic as well as vision problems is since recently available in [69].

One of the scientific pillars this thesis has been founded on is the notion of "projective motion", more precisely "projective rigid motion". It can be considered as a unifed formal and computational framework related to the classical problems in computer vision, such as pose, structure-from-motion (SFM), and binocular stereo vision. In this context, it is of interest what level of prior knowledge of camera calibration is required, since a fundamental question addressed in this thesis is what degree of calibration is inherent to the visual robot control, and what requirements have been made in the past just for the sake of technical convenience.

Object pose [27] from a single image requires both, a metric model of the object and the camera's intrinsic parameters in these metric units. This notion has given rise to the so-called position-based approach to visual servoing, see e.g. [80], for instance, or [75] which emphasizes more the pose-tracking problem. Structure from motion (SFM) [48] – more precisely structure and motion from an image pair – requires the intrinsic parameters, only, and yields a metric model of the object and the camera's motion in between the images, up-to a length scale, only. Visual servoing based on this has

been proposed in [73] for instance. A highly relevant special case is that of a planar object and its partial pose captured by the homography between two images of one and the same plane [6]. A respective visual servoing law has been introduced in [52], and extended to non-planar objects.

When internal camera calibration is no longer available, but only image correspondences, the essential matrix representing SFM becomes the fundamental matrix representing the epipolar geometry of an image pair [49]. It is determined from at least eight points, and linear estimation is feasible and efficient [31]. Basic visual "homing" based on epipoles is actually proposed in [1]. Above that, the fundamental matrix allows for three-dimensional reconstruction of scene-structure. In the uncalibrated case, [22], [30], it allows to recover structure and camera motion up to a fixed but unknown projective transformation of the ambient space. Approaches to visual servoing which are trying to cope with such ambiguities are described in [25], [24], and [46]. A theoretical framework based on geometric algebras has been proposed in [2].

Reconstruction can be extended to a series of images, where respective factorization-based methods for multi-view reconstruction have successfully been applied , first to affine [74], and then to projective cameras [71]. Common to most of these formulations is that the camera moving in or around a static scene is considered, and that it is camera motion which is recovered. In contrast, these methods cannot be applied immediately to a dynamic scene containing multiple objects moving relative to each other. Therefore, visual robot control, especially if an independently mounted camera is considered [39], is demanding rather for a sensorial system that yields a dynamic three-dimensional reconstruction of a dynamically changing environment. In the most general case, uncalibrated reconstruction is projective, and the respective notion of three-dimensional dynamics in a projective ambient space is "projective motion", as proposed in this thesis. In case of Euclidean three-space, discrete motion and pose is recapitulated in [82]. In the 2D projective case, the continuous motion seen by a single uncalibrated camera has been developed up to first-order in [77].

Projective motion output can be provided by stereo vision, which is considered in combination with a computational method for triangulation as a proper device for triangulation, briefly called a "stereo camera". The article [32] introduced such a method for triangulation, and discusses its properties when applied to ambient spaces of various geometries, from Euclidean to projective. Besides this polynomial solution in closed-form, a general iterative method, and also methods for reconstruction of lines have been proposed in [28] and [34], which are closely related to section 2.6.1, and a respective method for iteratively estimating projective motion [15], now from various features is developed in section 2.6.2.

A state-of-the-art application of projective motion is self-calibration of a rigid stereo rig [17], [83]. This can be done for the stereo camera undergoing

various types of rigid motion, among them ground-plane motions [11], [3] with respective degenerate situations [14]. The methods are closely related to the stratification hierarchy of geometric ambient spaces [21]. Among the special projective motions, pure translations and pure rotations with its axis in a general position – actually a general ground-plane motion is a non coordinate-free instance of a pure rotation – have so far been neglected. Finally, they have been studied in-depth by the author [58], [59], where in particular their potential for representing articulated motion and mechanisms has been exploited. Such motions, sometimes called "singular" motions, have been investigated with respect to degeneracies of self-calibration [72], and respective algorithms for partial or constrained estimation of epipolar geometries [78], camera calibration, or partial reconstructions [50].

However, those works are still using stratified representations, separating camera calibration and Euclidean representations of motion, although equivalent projective representations, to which Euclidean and affine camera parameters are transparent, can indeed be defined, and have been introduced by the author [60]. Moreover, projective representations of articulated motion and the kinematics of the underlying mechanism are equally possible, and respective calibration techniques have been demonstrated as feasible and accurate [62]. Above that, the notion of projective velocities of a point, a plane, or a rigid-body moving in a projective ambient space have been introduced in [61], (section 2.4.3) and [63]. In addition, the relationship between the chosen frame of the ambient space and the corresponding motion representations has been revealed (see also A.3), which allows to show for instance that visual servoing of hand-eye and independent-eye systems is strictly equivalent from a theoretical point-of-view (section 8.2.1). Last but not least, the group property of projective displacements, i.e. of projective rigid motion, its differential geometry as a Lie group, and its algebraic structure generated by the underlying Lie algebra have been identified [61]. Besides [18], few work has tried to explore the group-theoretical foundations of the visual servoing problem.

Most of the common approaches to visual servoing are neglecting the question of modeling or "kinematic calibration" of the robot actuator. Either an idealized Cartesian robot is considered [19], [51], or a six-axis industrial manipulator is operated in the manufacturer's Cartesian mode [26]. The effect of singularities [53], coupling [8], or inaccurate kinematics, also in the robot-camera linkage, are investigated rarely, and often no further than to the argument of asymptotic stability about the convergence of the closed loop.

Few approaches are controlling the robot-joints directly, and if, the coupling between the robot posture and the visual Jacobian is either covered by a coarse numerical approximation of a linear interaction model [44], which is valid only locally, or by an a-priori estimation of the interaction model, mostly locally around the goal and the robot's configuration, there [70].

Convergence of such control laws has been observed, but little can be said about its performance in work-space, or about trajectories and convergence properties, there.

The same could be said about a "coarse calibration" of the camera and the hand-eye linkage, which generally means that factory given values for focal length, pixel size, and coarse, manual measurements of the hand-eye parameters are utilized. Again, the closed-loop will assure a narrow convergence for the visual goal to be reached, however, nothing can be said about performance with respect to the workspace goal, especially in case of an "endpoint-open" system (see [42]).

"Uncalibrated", in first instance, refers to a visual task or goal which is formulated independent of the camera parameters. Clearly, projective invariance is the underlying concept, and [25], [24] tries to exploit such basic invariants for a number of simple alignment tasks. In [39] finally, a tool-to-object alignment in all 6 degrees-of-freedom is developed, based on a 5-point basis of projective space. This concept is also applied in [46]. The fact that "endpoint-closed" (see [42]) goals are considered implies the visual goal to be sufficient for a workspace-goal to be achieved.

Common to all these approaches is the "stratification" of projective camera parameters and projected visual motion against robot parameters or articulated motion. This stratification is most evident in [46], where the linear relationship between a projective frame for the work-space and a Euclidean frame for the robot is considered. However, the stratification of the model is vanishing as soon as it is contracted for instance to a robot-image Jacobian in matrix form [67], [68].

Therefore, the derivation of visual servoing laws based on projective kinematics representing the actuator, and on projective motion representing the demanded task and trajectory no longer requires a stratified model, unless for notational or technical convenience. It still results in respective interaction matrices, which are strictly equivalent and in which stratified parameters are transparent. Moreover, it allows for seamless integration with uncalibrated task-representations, including constrained trajectories, and projective calibration of robots.

Such an unstratified or "non-metric" visual servoing system obviously does change the calibration problem. Commonly, this problem has to be solved for the camera [23], the robot [79], and the camera-robot linkage [76]. The drawback of a stratified step-by-step calibration is that errors in the previous stage will affect the results of the succeeding steps, e.g. inaccurate kinematics rather quickly cause a hand-eye calibration to degenerate. Work already exists which tries to overcome this, [38], however rarely to an extent that calibration identifies also the kinematic parameters. Especially self-calibration techniques suffer from their generality and tend to instabilities due to the few and weak constraints available [72]. In contrast, the projective self-calibration of a stereo-robot system, as proposed in this thesis, first

establishes an unstratified model for each robot joint, which yields already an operational and accurate system. Then, it combines them into a consistent, but still unstratified model of the kinematic robot-stereo interaction. Finally, if convenient, an a-posteriori stratification of the representations can be recovered, as well.

Chapter 2

Fundamentals

2.1 Perspective camera and $\mathbb{P}^2$

This section describes how the geometry of a perspective camera can be modeled algebraically in terms of projective plane and its transformation group, the latter allowing in addition the properties of the CCD video-sensor and the frame-grabbing device to be modeled.

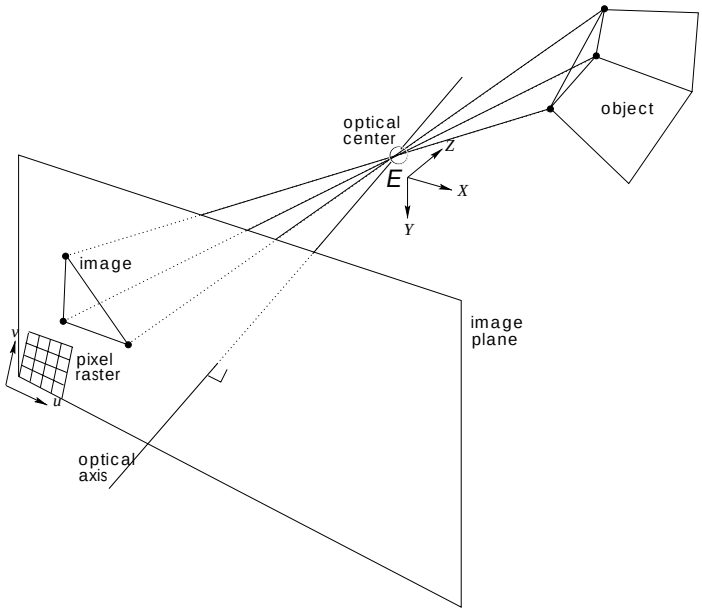

Figure 2.1: Pinhole camera with Euclidean frame.

Euclidean camera frame

For representing a camera or a stereo camera pair, each camera is associated with a Euclidean frame, denoted $\mathcal{E}$. It is allocated onto the camera in a

standard manner (Fig. 2.1). The optical center becomes the origin, the optical axis becomes the Z-axis, and the focal plane is taken as X-Y plane, where CCD-scanlines become the X-axis, but CCD-columns can be skew to the Y-axis. In such a frame, depth along the Z-axis is always orthogonal to the image-plane, hence to X- and Y-axis.

The frame $\mathcal{E}$ is spanning an ambient space which is Euclidean, and in which a Euclidean point has coordinates $(X, Y, Z)^\top$. It projects onto the image point $(x, y)^\top$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{fX}{Z} \\ \frac{fY}{Z} \end{pmatrix}, \qquad \text{with } f \text{ the focal length.}$$

Here, $(x, y)^\top$ are coordinates in a plane, which has the principal point as origin, and the x- and y-axis aligned with $\mathcal{E}$. Note, dividing by the depth Z makes this projection a non-linear one.

Homogeneous coordinates

Image coordinates can be formally extended to homogeneous ones by an additional third row $(x, y, 1)^\top$. This allows the 3-vector (X, Y, Z) to represent a point in $\mathcal{E}$, and at the same time its image projection or equivalently the direction of its optical ray. In this context, the vector-space is divided into equivalence classes of non-zero 3-vectors with a free, unknown scale λ.

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{pmatrix} \simeq \lambda \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \quad \text{with } f = 1.$$

Thus, as soon as the length unit is set to f, homogeneous coordinates implicitly express the perspective projection of a pinhole camera (Fig. 2.1).

Infinity

Usually, the representatives for such equivalence classes are chosen to be vectors with a 1 on the third row, since it is non-zero for all visible points. Additionally, the general or "projective" homogeneous coordinates allow optical rays to be represented that are actually "invisible", i.e. that lie in the focal plane, and so have $Z = 0$. Formally, these are 3-vectors with a 0 on the third row. Geometrically, they form a "*line at infinity*" holding the "*vanishing points*" or the "*directions*" of the plane, where opposite directions, i.e. antipodal vanishing points are identified.

$$\begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \lim_{\tau \to \infty} \frac{1}{\tau} \begin{pmatrix} \tau x \\ \tau y \\ 1 \end{pmatrix}.$$

The set of non-zero homogeneous three-vectors are thus a representation of the projective plane $\mathbb{P}^2$. This geometric space is used in the sequel to model pinhole cameras and their image-planes.

Intrinsic camera geometry

In practice, there is still a transformation between the Euclidean image-plane $(x, y, 1)^\top$ as a geometric entity, and the video-image $(u, v, 1)^\top$ as a 2D-signal captured by the CCD and processed by the frame grabber. In (2.1), such a video-image is described in pixels of width $1/k_u$ and height $1/k_v$, with skew k_{uv}, where u_0 and v_0 are the pixel-coordinates of the principal point.

$$\mathbf{K} = \begin{bmatrix} k_u & k_{uv} & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.1}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \simeq \mathbf{K} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{2.2}$$

Please note, $\mathbf{K}$ is strictly speaking an affine transformation or an "*affinity*", as the last row shows, whereas $\simeq$ denotes projective equality up-to-scale.

Image transformations

The homogeneous coordinates in the projective plane $\mathbb{P}^2$ allow to represent projective transformations linearly in terms of the projective group $PGL(2)$, whose elements are "*homographies*" or 2D-"*projectivities*". They are represented as 3×3 homogeneous matrices $\mathbf{H} \in PGL(2)$ in such a context[1]. For example, the transformations of an image resulting from a camera rotating about its optical center are always such projectivities. In consequence, misalignments between the optical axis and the image-plane can so be modeled, e.g. a CCD failing to be exactly parallel to the focal plane.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \simeq= \mathbf{K} \quad \mathbf{R} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{2.3}$$

Still, algebraic QR-decomposition allows an arbitrary homography $\mathbf{H}$ to be rewritten in form of an upper triangular matrix $\mathbf{K}$ times an orthonormal matrix $\mathbf{R}$, i.e. a rotation. Similarly, the later introduced three-dimensional projective space (2.7) will allow for a linear representation of perspective projection, although it is analytically a non-linear one.

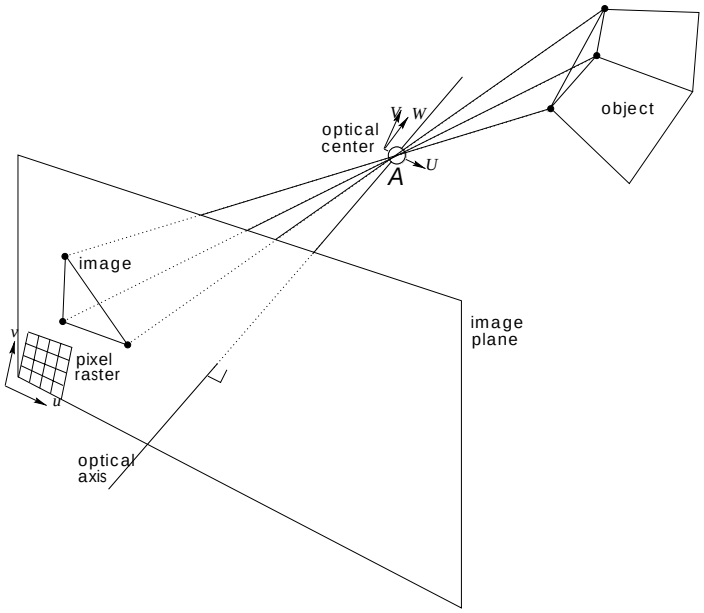

Figure 2.2: Pinhole camera with affine frame.

Affine camera frame

Based on $\mathbf{K}$, an affine camera frame $\mathcal{A}$, is well-defined, which can be thought of as being allocated onto the camera (Fig. 2.2). It spans an affine ambient space in which 3D-points are represented as $(U, V, W)^\top$, where U and V hold pixel-coordinates, and W holds depth-coordinates parallel to the optical axis in f units. It is indeed a general affine frame and no longer a Euclidean one, since the U- V-axes are skew, and since their scales differ, especially from the one on the W-axis.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{U}{W} \\ \frac{V}{W} \\ 1 \end{pmatrix} \simeq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} U \\ V \\ W \end{pmatrix}.$$

Note, in this particular frame, $\mathcal{A}$, the camera parameters are trivially the identity $\mathbf{I}$.

2.2 Stereo camera and $\mathbb{P}^3$

Consider now a pair of pinhole cameras mounted onto a stereo rig, briefly called a "*stereo camera*" (Fig. 2.3). The basic properties of such a system are recapitulated in this section. It is presented as a triangulation device which

[1]The notation $\mathbf{H}$ is mainly used for space homographies, introduced later. Normally, the dimension of a homography can be distinguished from its mathematical context.

can be operated either in metric, in affine, or in projective mode, depending on the present degree of calibration. Such a device is used to recover a reconstruction of the three-dimensional workspace in the respective ambient space: Euclidean, affine, or projective camera-space, denoted $\mathcal{E}$, $\mathcal{A}$, or $\mathcal{P}$.

This section concludes with the "*rigid stereo assumption*", which most results of this thesis are based on. Briefly, it states that a stereo camera with a fixed geometry results in the transformations between the respective ambient spaces to be constant, as well.

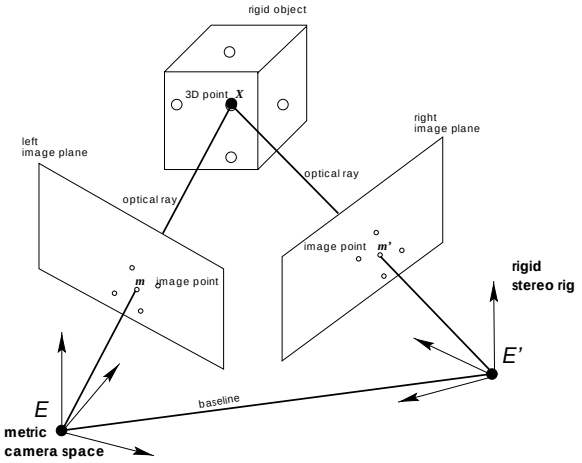

Figure 2.3: Stereo camera with Euclidean frame.

Extrinsic geometry

Consider a pair of Euclidean frames, $\mathcal{E}$ and $\mathcal{E}'$, on the left and right camera. Then, the extrinsic geometry of the rig is described by the pose of the right camera with respect to the left one. This is expressed as a rotation $\mathbf{R}$ followed by the translation $\boldsymbol{t}$, which would move the right frame onto the left one, or analogously, which transforms the coordinates in the left ambient space to the right ones, indicated by a prime:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{t}.$$

The formal extension to a homogeneous fourth coordinate – see section 2.3 for a rigorous treatment – allows this transformation, which is non-linear in the summed form above, to be represented linearly as a single multiplication with a 4×4 matrix:

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R} & \boldsymbol{t} \\ \mathbf{o}^\top & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{t} \\ 1 \end{pmatrix}. \tag{2.4}$$

Note, such matrices represent always a displacement, and thus represent the so-called "*special Euclidean group*" $SE(3)$.

Infinity

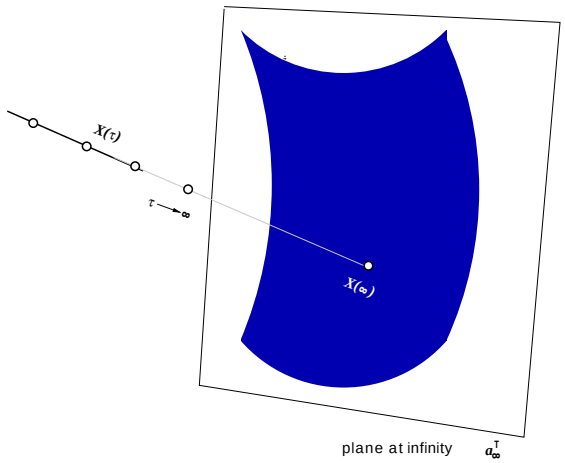

Figure 2.4: Plane at infinity in projective space $\mathbb{P}^3$.

In a Euclidean, and also in an affine context, a point homogenized to 1 is always a finite point, whereas points at infinity formally have a 0 on the forth row. Geometrically, they can be interpreted as vanishing points or "*directional*" points of an endlessly[2] continued translation along $\boldsymbol{t} = (X, Y, Z)^\top$ (Fig. 2.4). Since τ is unsigned, these directions are unoriented, and two antipodal vanishing points are thus coincident.

$$\begin{pmatrix} X \\ Y \\ Z \\ 0 \end{pmatrix} = \lim_{\tau \to \infty} \frac{1}{\tau} \begin{pmatrix} \tau X \\ \tau Y \\ \tau Z \\ 1 \end{pmatrix}. \tag{2.5}$$

Please note, this simple formal distinction between finite and infinite points is only valid in Euclidean or affine frames. A necessary condition on Euclidean and affine transformations is therefore that they leave this property invariant, which is indeed assured by the fourth row of a respective matrix equal $(0, 0, 0, 1)$.

$$\begin{pmatrix} * \\ * \\ * \\ \rho \end{pmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} * \\ * \\ * \\ \rho \end{pmatrix}.$$

So, $\boldsymbol{a}_\infty^\top = (0, 0, 0, 1)$ necessarily are the Euclidean coordinates of the "*plane at infinity*" which contains all directional points.

Euclidean stereo camera

For the moment, let us continue with the classical modelling of a stereo camera, but let us use rigorously $\mathbb{P}^3$ and $\mathbb{P}^2$ to represent points in three-space and in the image-planes, respectively.

[2]This limit has to be understood as the limit of a normalized coordinate vector.

For a vector $\boldsymbol{X}$ holding homogeneous Euclidean coordinates in $\mathcal{E}$, the left and right cameras can be written as in (2.7), where $\mathbf{K}'$ are the right intrinsic parameters:

$$\boldsymbol{m} \simeq \quad \mathbf{P}_E \quad \boldsymbol{X}, \qquad \boldsymbol{m}' \simeq \quad \mathbf{P}'_E \quad \boldsymbol{X}, \qquad \text{where} \tag{2.6}$$

$$\mathbf{P}_E = \left[\begin{array}{c|c} \mathbf{K} & \mathbf{0} \end{array}\right], \qquad \mathbf{P}'_E = \left[\begin{array}{c|c} \mathbf{K}'\mathbf{R} & \mathbf{K}'\boldsymbol{t} \end{array}\right]. \tag{2.7}$$

Affine stereo camera

For a vector $\boldsymbol{N}$ holding homogeneous coordinates in $\mathcal{A}$, the left projection matrix is the trivial one, and the right projection becomes respectively

$$\boldsymbol{m} \simeq \quad \mathbf{P}_A \quad \boldsymbol{N}, \qquad \boldsymbol{m}' \simeq \quad \mathbf{P}'_A \quad \boldsymbol{N}, \qquad \text{where} \tag{2.8}$$

$$\mathbf{P}_A = \left[\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right], \qquad \mathbf{P}'_A = \left[\begin{array}{c|c} \mathbf{K}'\mathbf{R}\mathbf{K}^{-1} & \mathbf{K}'\boldsymbol{t} \end{array}\right]. \tag{2.9}$$

Interestingly, for points at infinity $\boldsymbol{N}_\infty = (U, V, W, 0)^\top$, the so-called infinity-homography $\mathbf{H}_\infty$ is well-defined, which maps their left images $\boldsymbol{m}_\infty$ onto right images $\boldsymbol{m}'_\infty$:

$$\mathbf{H}_\infty = \mathbf{K}'\mathbf{R}\mathbf{K}^{-1}, \qquad \boldsymbol{m}'_\infty = \mathbf{H}_\infty \boldsymbol{m}_\infty.$$

Additionally, since $\boldsymbol{t}$ relates to the left optical center in $\mathcal{E}'$, the right epipole equals $\boldsymbol{e}' = -\mathbf{K}'\boldsymbol{t}$.

Projective stereo camera and epipolar geometry

Consider a number of 3D points $\boldsymbol{M}$, and their matching left and right images $\boldsymbol{m}$ and $\boldsymbol{m}'$. Convince yourself that the two optical centers together with the 3D-point $\boldsymbol{M}$ form a plane which cuts the two image-planes in a line, each (Fig. 2.5). Its algebraic formulation is the epipolar constraint in form of a 3×3 matrix, the so-called "*fundamental matrix*" $\mathbf{F}$.

$$\boldsymbol{m}^\top \mathbf{F} \boldsymbol{m}' = 0. \tag{2.10}$$

It maps a right image point $\boldsymbol{m}'$ to its corresponding epipolar line $\mathbf{F}\boldsymbol{m}'$ in the left image, on which $\boldsymbol{m}$ must lie. It has been shown that at least 8 such point-correspondences determine the so-called epipolar geometry of the stereo camera (Fig. 2.5).

In the "*uncalibrated*" or "*projective*" case, the intrinsic parameters $\mathbf{K}$, $\mathbf{K}'$, and the extrinsic parameters $\mathbf{R}$, $\boldsymbol{t}$ are supposed not to be known. In spite of that, a pair of projection matrices can be derived, which is consistent with

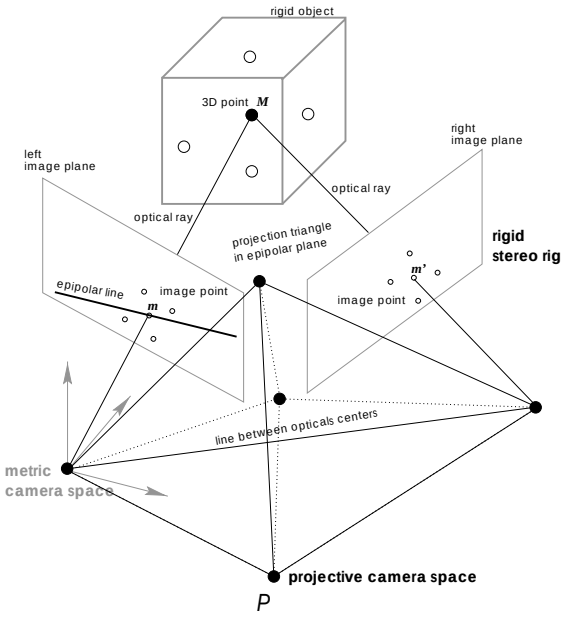

Figure 2.5: Stereo camera pair with epipolar geometry and projective frame.

the present epipolar geometry. In [30], it is shown how a right projection matrix consistent with $\mathbf{F}$ can be defined,

$$\boldsymbol{m} \simeq \quad \mathbf{P}_P \quad \boldsymbol{M}, \qquad \boldsymbol{m}' \simeq \quad \mathbf{P}'_P \quad \boldsymbol{M}, \qquad \text{where} \tag{2.11}$$

$$\mathbf{P}_P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{P}'_P = \left[[\boldsymbol{e}']\mathbf{F} + \boldsymbol{e}'\boldsymbol{a}^\top \middle| \boldsymbol{e}' \right], \tag{2.12}$$

and $(\boldsymbol{a}^\top, 1)$ is an arbitrarily fixed 4-vector.

Triangulation

From a more abstract point-of-view, the above three projections (2.7),(2.9), (2.12) are three different models of one and the same stereo camera pair. They differ in the ambient space they are expressed in, Euclidean, affine, or projective. So, they differ in the camera frame a respective 3D-reconstruction refers to: $\mathcal{E}$, $\mathcal{A}$, or $\mathcal{P}$. Geometrically, each of the equations (2.6), (2.8), or (2.11), describes a pair of optical rays (see section 2.6.1). Solving homogeneous equations like the one in (2.13) for $\boldsymbol{X}$, $\boldsymbol{N}$, or $\boldsymbol{M}$, amounts to intersecting these rays, and yields a triangulation [32] of the respective ambient space relative to its proper coordinate frame.

$$\text{projective:} \qquad \begin{bmatrix} \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{bmatrix} \cdot \mathbf{P}_P \\ \begin{bmatrix} 1 & 0 & -u' \\ 0 & 1 & -v' \end{bmatrix} \cdot \mathbf{P}'_P \end{bmatrix} \rho \boldsymbol{M} = \mathbf{o}. \tag{2.13}$$

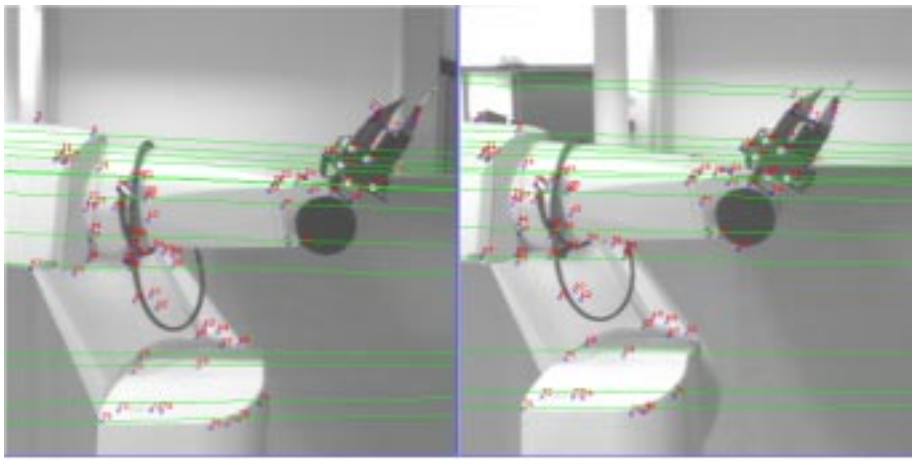

Figure 2.6: Stereo image-pair of a robot arm, matches, and their epipolar lines illustrating the pair's epipolar geometry.

Homographies

Remember, the "*projective three-space*" $\mathbb{P}^3$ is the set of all finite and infinite points. They are represented by equivalence classes of non-zero, homogeneous 4-vectors $\boldsymbol{M} = \rho\,(U, V, W, 1)^\top$ with a well-defined scale ρ. The corresponding transformation group are the spatial "*homographies*" or $3D$-"*projectivities*", always denoted as $\mathbf{H} \in PGL(3)$. This group is commonly written as 4×4 homography matrices, which in general will alter a vector's forth row

$$\boldsymbol{M}' = \gamma \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \boldsymbol{M}.$$

It is worthwhile regarding how such homographies are well-defined. Since each point has 3 "*degrees-of-freedom*" (dof), and the homography matrix has 15 dof, 5 points in general position are necessary and sufficient for a matrix $\mathbf{H}$ to be well-defined, up to a scalar γ, however. So, five pairs of corresponding 4-vectors $\boldsymbol{A}_p$ and $\boldsymbol{B}_p$, which actually amount to a pair of corresponding projective frames, allow the linear transformation between the two frames, $\mathcal{A}$ and $\mathcal{B}$, to be calculated algebraically as follows [12].

$$\lambda_p \boldsymbol{A}_p = \mathbf{H}\boldsymbol{B}_p, \qquad p = 1, \ldots, 5$$

$$[\mathbf{B}] = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 & \boldsymbol{B}_3 & \boldsymbol{B}_4 \end{bmatrix}, \quad [\mathbf{A}] = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 & \boldsymbol{A}_4 \end{bmatrix}$$

In the first instance, four points define an intermediate 4×4 matrix $\mathbf{H}^*$ related to the solution by four scalars $\mathbf{D} = \text{diag}\,(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ on the columns.

$$\mathbf{H}^* = [\mathbf{A}]\mathbf{D}[\mathbf{B}]^{-1}.$$

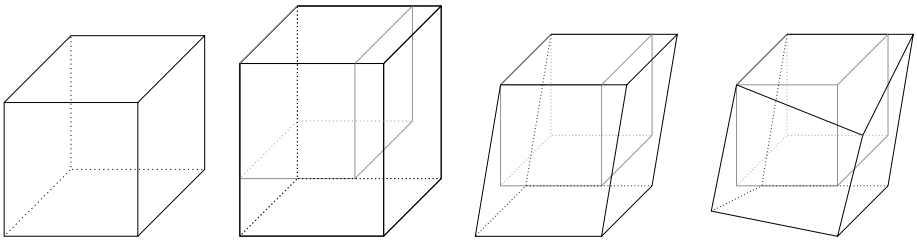

Figure 2.7: Illustration of levels in stratification hierarchy (left to right): Euclidean, scaled Euclidean, affine, projective.

They are in fact determined by the fifth point:

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4)^\top = \text{diag } (\boldsymbol{A}_5)^{-1}\mathbf{H}^*\boldsymbol{B}_5, \qquad \text{such that}$$
$$\mathbf{H} = \lambda_5 \mathbf{A}\text{diag } (\lambda_1, \lambda_2, \lambda_3, \lambda_4)\mathbf{B}^{-1}.$$

Basis of projective frame

The fact that the number of points defining a 3D-projectivity is five implies that a projective frame requires as well five points in general position to be well-defined. In other words, a "*basis*" of $\mathbb{P}^3$ consists of five 4-vectors, where the standard notation of such a basis is the following

$$\boldsymbol{E}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{E}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{E}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \boldsymbol{E}_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \boldsymbol{E}_5 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

As long as this projective frame is aligned with a Euclidean space, the standard basis has a metric interpretation: $\boldsymbol{E}_1$, $\boldsymbol{E}_2$, $\boldsymbol{E}_3$ are X-, Y-, Z-directions, $\boldsymbol{E}_4$ is the origin, and $\boldsymbol{E}_5$ is the unity point, which relates the coordinates' scales. As soon as the frame has suffered a projective skew, this interpretation of projective coordinates is no longer valid.

In a preceding paragraph, a way to allocate a projective frame $\mathcal{P}$ onto a stereo camera has been described. This frame is no longer aligned with $\mathcal{E}$, so that the Euclidean meaning of the basis-vector is lost. Still, an "*upgrade homography*" $\mathbf{H}_{PE}$ (2.18) from $\mathcal{P}$ to $\mathcal{E}$ can be introduced that relates the projective to the Euclidean camera frame, i.e. that holds the calibration data. Later yet, this "*calibration homography*" will give rise to a special class of homographies that represents rigid motion in projective space: "*projective displacements*".

Stratification hierarchy

The relationship between the three camera frames and respective reconstructions can be understood formally by means of inserting a generic homography $\mathbf{H}$:

$$\boldsymbol{m} = (\mathbf{P}\mathbf{H}^{-1}) \cdot (\mathbf{H}\boldsymbol{M}).$$

On the one hand, this allows the different forms of projection matrices to be related (below, first column). On the other hand, it allows the different reconstructions to be upgraded a-posteriori (below, second column).

$$\mathbf{P}_A = \mathbf{P}\mathbf{H}_{PA}^{-1}, \qquad \boldsymbol{N} = \mathbf{H}_{PA}\boldsymbol{M}, \qquad \mathbf{H}_{PA} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \boldsymbol{a}_\infty^\top & & & 1 \end{bmatrix}. \tag{2.14}$$

$$\mathbf{P}_E = \mathbf{P}_A\mathbf{H}_{AE}^{-1}, \qquad \boldsymbol{X} = \mathbf{H}_{AE}\boldsymbol{N}, \qquad \mathbf{H}_{AE} = \begin{bmatrix} & & & 0 \\ & \mathbf{K}^{-1} & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.15}$$

$$\mathbf{P}_E = \mathbf{P}_P\mathbf{H}_{PE}^{-1}, \qquad \boldsymbol{X} = \mathbf{H}_{PE}\boldsymbol{M}, \qquad \mathbf{H}_{PE} = \begin{bmatrix} & & 0 \\ & \mathbf{K}^{-1} & 0 \\ & & 0 \\ \boldsymbol{a}_\infty^\top & & 1 \end{bmatrix}. \tag{2.16}$$

In detail, applying the inverse of a homography from the right onto a projection matrix transforms the cameras between the three ambient spaces. Applying these homographies from the left onto a reconstruction upgrades the latter to an ambient space that is on a higher level in the so-called "*stratification hierarchy*", [21].

On the projective level of this hierarchy, only coincidence and collinearity of geometric entities, i.e. points, planes, etc. are defined, as well as their cross-ratios. On the affine level, this is upgraded to the notion of parallelism of lines and planes, and to length ratios of parallel entities. Basically, this requires the infinity to be identified. On the Euclidean level, this is upgraded to notions like perpendicularity, angle, and distance.

Please note the difference between a projective reconstruction upgraded a-posteriori to Euclidean and an a-priori Euclidean reconstruction. In the second case, since the optical rays are described in a Euclidean frame, for example the length of a normal common to both rays is available. So, the spatial distance metric can be used already for triangulation. In the projective case, such an error measure is not available. Therefore, the objective used by most estimation techniques developed in this thesis (see sections on "*Numerical Estimation*") is a reprojection-error of image measurements. So, they allow to work in any sort of ambient space.

Definition 1 (Rigid Stereo Assumption) *As soon as the triangulation device is a stereo camera having a rigid linkage between two cameras with constant intrinsic parameters, the "rigid-stereo assumption" is fulfilled. The cameras have a constant epipolar geometry* $\mathbf{F}$*, which defines the upgrade homography* $\mathbf{H}_{PE}$ *relating the projective to the Euclidean ambient space,* $\mathcal{P}$ *to* $\mathcal{E}$*. This matrix contains, besides left camera parameters* $\mathbf{K}$*, the coordinates*

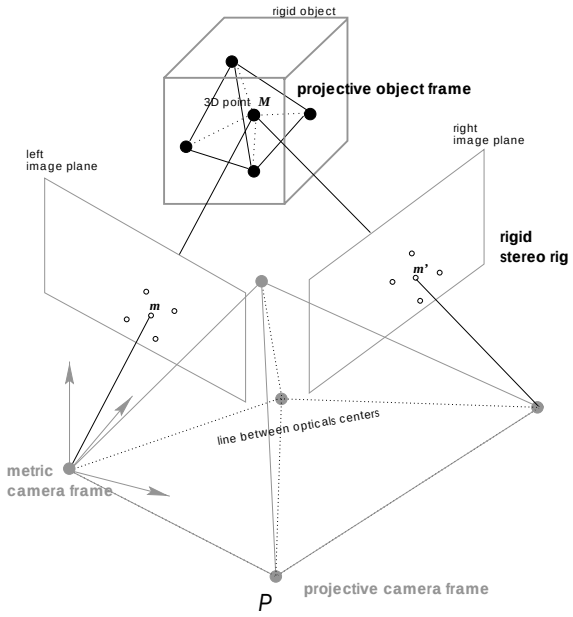

Figure 2.8: Virtual rigid-stereo assumption based on a rigid structure in the scene.

of the infinity plane $\left(\boldsymbol{a}_\infty^T, 1\right)$ *in* $\mathcal{P}$.

$$\mathbf{H}_{PE} = \begin{bmatrix} \mathbf{K}^{-1} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \boldsymbol{a}_\infty^\top & 1 \end{bmatrix}, \tag{2.17}$$

$$\boldsymbol{X} \simeq \quad \mathbf{H}_{PE} \quad \cdot \boldsymbol{M}. \tag{2.18}$$

Alternatively, *the rigid-stereo assumption is "***virtually***" fulfilled as soon as a general rigid structure of five 3D-points is always present in the ambient space (Fig. 2.8). In this case, these points are assigned the standard coordinates* $\boldsymbol{E}_{1\ldots5}$*, and they so form a fixed projective basis. It allows the triangulation to be expressed with respect to this fixed projective frame by means of a further* 4×4 *homography, so that* $\boldsymbol{X} \simeq \mathbf{H}_{PE}\mathbf{H}_{PP}\boldsymbol{E}$

$$\boldsymbol{M}_p \simeq \quad \mathbf{H}_{PP} \quad \boldsymbol{E}_p, \quad \textit{with} \tag{2.19}$$

$$\mathbf{H}_{PP} = \begin{bmatrix} \frac{\gamma_1}{\mu_1}\boldsymbol{M}_1 & \frac{\gamma_2}{\mu_2}\boldsymbol{M}_2 & \frac{\gamma_3}{\mu_3}\boldsymbol{M}_3 & \frac{\gamma_4}{\mu_4}\boldsymbol{M}_4 \end{bmatrix}, \quad \textit{where} \tag{2.20}$$

$$\boldsymbol{M}_5 = (\mu_1, \mu_2, \mu_3, \mu_4)^\top,$$

$$\boldsymbol{M}_1 + \boldsymbol{M}_2 + \boldsymbol{M}_3 + \boldsymbol{M}_4 = (\gamma_1, \gamma_2, \gamma_3, \gamma_4)^\top,$$

$$\boldsymbol{X} \simeq \quad \mathbf{H}_{PE}\mathbf{H}_{PP} \quad \cdot \boldsymbol{E}. \tag{2.21}$$

2.3 Displacements, rigid motion, and $SE(3)$

In this section, the theory of representing the group of rigid displacements is briefly sketched. The aim is to draw a line between the concepts of a group on the one side, and the matrix representation of a group on the other side. Since they are also Lie groups, the presentation in 2.4 covers the differential geometry of these groups, and their relation with the corresponding Lie algebras and their representations. Particularly one-dimensional subgroups are essential for this thesis, and have given rise to chapters 3 and 4.

Although displacements are traditionally understood as the transformation group of Euclidean three-space, they are objects of mathematical study on their own. There exists actually a number of algebraic representations, e.g. dual quaternions [13], which are no longer directly related to a three-space. The mathematical field underlying this is "*representation theory*", a brief summary of which is provided in appendix A.4. The reason why reference is made to this rather abstract field of pure mathematics is that it allows to see a central topic of this thesis – the group of projective displacements – in a brighter light, namely as a representation of rigid displacements within the transformation group of projective space.

2.3.1 The Lie group $SE(3)$ – displacement group

The rigid motions in three-space form a group: the special Euclidean group $SE(3)$. It is the semi-direct product of the rotation group $SO(3)$ and the translation group $\mathbb{R}^3$, both three-dimensional. For the moment, let the notations $\mathbf{R}$ and $\boldsymbol{t}$ refer to abstract elements of these two groups, and let $\circ$ and $+$ refer to the abstract operations of these groups, respectively. Consider further $SE(3)$, and let for the moment the notation $\mathbf{T}_{RT}$ refer to an abstract group element, which stands for the semi-direct product $(\mathbf{R}, \boldsymbol{t})$ of two basic elements. The abstract group operation in $SE(3)$ is then defined in terms of the operations in these two subgroups:

$$\mathbf{T}_{RT_1} \circ \mathbf{T}_{RT_2} = (\mathbf{R}_1, \boldsymbol{t}_1) \circ (\mathbf{R}_2, \boldsymbol{t}_2) = (\mathbf{R}_1 \circ \mathbf{R}_2, \boldsymbol{t}_1 + \mathbf{R}_1 \cdot \boldsymbol{t}_2). \tag{2.22}$$

Note that the operation $\mathbf{R} \cdot \boldsymbol{t}$ already amounts to an action of the special orthogonal group $SO(3)$ onto $\mathbb{R}^3$. Its matrix representation are the orthogonal 3×3 matrices ($\mathbf{R}\mathbf{R}^\top = \mathbf{I}$) of unit determinant (see chapter 4). This matrix-multiplication is a linear action on the three-dimensional space of translations. The representation of the translation group is a translation vector $\boldsymbol{t} \in \mathbb{R}^3$ and the operation is the vector-addition.

An equivalent representation of $\mathbb{R}^3$ are the matrices $\mathbf{T}_T$ (2.23), embedded in the 4×4 matrix group $GL(4)$. The usual matrix-multiplication is a linear

action which, in case of translations, is furthermore commutative.

$$\boldsymbol{t} \rightarrow \mathbf{T}_T = \begin{bmatrix} \mathbf{I} & \boldsymbol{t} \\ \mathbf{o}^\top & 1 \end{bmatrix}, \tag{2.23}$$

$$\mathbf{T}_{T1} \cdot \mathbf{T}_{T2} = \begin{bmatrix} \mathbf{I} & \boldsymbol{t}_1 + \boldsymbol{t}_2 \\ \mathbf{o}^\top & 1 \end{bmatrix}. \tag{2.24}$$

Please note, matrices with $(0, 0, 0, 1)$ on the fourth row are called *"homogeneous"*. General homogeneous matrices are a representation of the three-dimensional affine group, a subgroup of which is $SE(3)$ – the special Euclidean group.

Then, a corresponding representation of $SO(3)$ are the matrices $\mathbf{T}_R$ (2.25), also embedded in the homogeneous matrix group. Their multiplication however is no longer commutative, neither among two different $\mathbf{T}_R$-matrices, nor between $\mathbf{T}_R$ and $\mathbf{T}_T$.

$$\mathbf{R} \rightarrow \mathbf{T}_R = \begin{bmatrix} \mathbf{R} & \mathbf{o} \\ \mathbf{o}^\top & 1 \end{bmatrix}, \tag{2.25}$$

$$\mathbf{T}_{R1} \cdot \mathbf{T}_{R2} = \begin{bmatrix} \mathbf{R}_1\mathbf{R}_2 & \mathbf{o} \\ \mathbf{o}^\top & 1 \end{bmatrix}. \tag{2.26}$$

Since both basic groups of $SE(3)$ have been embedded in $GL(4)$, it is hence straight-forward to write the semi-direct product and the group operation of $SE(3)$ in matrix form:

$$(\mathbf{R}, \boldsymbol{t}) \rightarrow \mathbf{T}_{RT} = \mathbf{T}_T\mathbf{T}_R = \begin{bmatrix} \mathbf{R} & \boldsymbol{t} \\ \mathbf{o}^\top & 1 \end{bmatrix}, \tag{2.27}$$

$$\mathbf{T}_{R_1T_1}\mathbf{T}_{R_2T_2} = \mathbf{T}_{T_1}\mathbf{T}_{R_1}\mathbf{T}_{T_2}\mathbf{T}_{R_2} = \begin{bmatrix} \mathbf{R}_1\mathbf{R}_2 & \boldsymbol{t}_1 + \mathbf{R}_1\boldsymbol{t}_2 \\ \mathbf{o}^\top & 1 \end{bmatrix}. \tag{2.28}$$

Besides the action of the group onto the group itself, the homogeneous form $\boldsymbol{X} = (X, Y, Z, 1)^\top$ of a three-vector allows the same matrix to act linearly on homogeneous point-vectors $\boldsymbol{X}$

$$\mathbf{T}_{RT}\boldsymbol{X} = \left(\mathbf{R}\,(X, Y, Z)^\top + \boldsymbol{t}, 1\right)^\top. \tag{2.29}$$

2.3.2 One-dimensional subgroups

Let us now study the one-dimensional subgroups of $SE(3)$:

Translation

As soon as translations are restricted to a fixed 3D direction $\boldsymbol{t}$, they give rise to the one-dimensional subgroup $\mathbf{T}_T(\tau)$, where τ is the 'deflection' or norm of $\boldsymbol{t}$. In a more general context, chapter 3 shows rigorously that such

a subgroup is isomorphic to the additive group $\mathbb{R}^1$. In the context of this thesis, such subgroups are used to model prismatic joints of 1 degree-of-freedom.

$$\tau \to \mathbf{T}_T(\tau) = \begin{bmatrix} \mathbf{I} & \tau\boldsymbol{t} \\ \mathbf{o}^\top & 1 \end{bmatrix}.$$

Rotation

As soon as rotations are restricted to be around an axis in a fixed position in space, they give rise to a one-dimensional subgroup $\mathbf{T}_R(\theta)$, where θ is the 'angle' of rotation. In the general context of chapter 4, it is shown rigorously that this group is isomorphic to the multiplicative group $\mathbb{C}$. In the context of this thesis, such subgroups are used to model revolute joints of 1 degree-of-freedom. Take for example the standard embedding of $SO(3)$ in $SE(3)$, i.e. the axis fixed at the origin, and suppose furthermore the axis of $\mathbf{R}$ to have the fixed direction $\boldsymbol{w}$.

$$\theta \to \mathbf{T}_R(\theta) = \begin{bmatrix} \mathbf{R}(\theta) & \mathbf{o} \\ \mathbf{o}^\top & 1 \end{bmatrix}.$$

Screw motion

As soon as rotation axis and translations are restricted to have the **same direction**, $\boldsymbol{t} \parallel \boldsymbol{w}$, the resulting motion is a screwing motion[3]. If there is furthermore a constant ratio $p = \frac{\theta}{2\pi \cdot \tau}$ between angular and linear component, called "*pitch*", a one-dimensional subgroup $\mathbf{T}_S(\theta)$ of screwing motions is obtained.

$$\theta \to \mathbf{T}_S(\theta, p) = \mathbf{T}_T(\tfrac{\theta}{2\pi \cdot p})\mathbf{T}_R(\theta), \qquad \text{where } \boldsymbol{t} \parallel \boldsymbol{w}. \tag{2.30}$$

It is isomorphic to $\mathbb{R}$. However, its embedding in $GL(4)$ is not the same as for $\mathbf{T}_T(\tau)$.

These basic subgroups are used in chapters 3 and 4 to represent respective robot joints: prismatic and revolute joints. Above, the exemple equations (2.23),(2.25), (2.30) are embeddings of a subgroup in $SE(3)$ which geometrically have a respective joint axis through the coordinate origin. On the one hand, each one-dimensional subgroup has a "canonical" embedding in $SE(3)$ as soon as the additional restriction that the axis direction is in Z-direction holds. Then, the corresponding matrix transformation is in Jordan form (see appendix A.5):

$$\mathbf{J}_T = \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&1 \\ 0&0&0&1 \end{bmatrix}, \quad \mathbf{J}_R = \begin{bmatrix} \cos&-\sin&0&0 \\ \sin&\cos&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix}, \quad \mathbf{J}_{RT} = \begin{bmatrix} \cos&-\sin&0&0 \\ \sin&\cos&0&0 \\ 0&0&1&1 \\ 0&0&0&1 \end{bmatrix}. \tag{2.31}$$

[3]The continuous motion which realizes a given displacement by a rotational motion about its screw axis, and a simultaneous translation

On the other hand, there are many possible embeddings of these groups in $SE(3)$. First, any point in three-space gives rise to a different instantiation of a subgroup, apart from the translation group. Then, any direction – once of translation, once of the rotation axis – equally instantiates new subgroups, where in case of a screw axis, any group is also instantiated with a pitch p. Alternatively, such instantiation of the canonical subgroup could be regarded as positioning a joint, actually its axis, in general position at a posture $\mathbf{S}^{-1}$. Algebraically, conjugating the Jordan form by the adjoint map (see appendix A.5) of $\mathbf{S}$ faithfully expresses this.

$$\mathrm{ad}_{\mathbf{S}}(\mathbf{J}) = \mathbf{S}^{-1}\mathbf{J}\mathbf{S} \tag{2.32}$$

Numerous variations of such forms are analyzed in great detail in chapters 3 and 4.

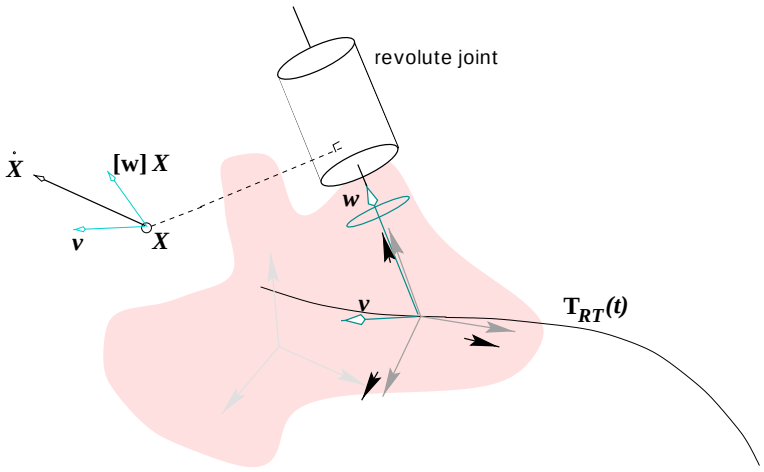

Figure 2.9: Velocity screw in one of its canonical frames.

2.3.3 Projective displacements

Above, nothing has been said about the form and group of a matrix $\mathbf{S}$ in the adjoint map. As long as $\mathbf{S} \in SE(3)$, the result always is a rigid motion. As long as $\mathbf{S}$ is homogeneous, i.e. a general affinity, so is the result. In case of a general matrix $\mathbf{S} \in GL(4)$, so is the result in general, although special cases can be identified in which $ad_{\mathbf{S}}(\mathbf{J})$ has a non-general form, e.g. is affine.

Now, the focus is on a particular adjoint map, which will be central to the definition and study of projective motion, as it is developed throughout this thesis. In section 2.2, the rigid-stereo assumption is made, and it is shown that the projective and Euclidean ambient spaces of such a device are related by the upgrade homography $\mathbf{H}_{PE}$. Consequently, this matrix allows to relate the representations of rigid structure in terms of the point-coordinates in both ambient spaces. Now, it is shown how the rigid-stereo

assumption moreover allows to represent rigid motion in an ambient space which is no longer a Euclidean one. This is achieved by substituting the projective for the Euclidean coordinates of a point, and by subsequently considering the homographies $\boldsymbol{M}' = \mathbf{H}_{RT}\boldsymbol{M}$ for a set of rigidly moving points their projective coordinates before and after the motion, $\boldsymbol{M}$ with $\boldsymbol{M}'$. This allows the notion of "projective motion" to be defined, which represents by means of homography matrices rigid motions of/in a projective ambient space.

$$\begin{aligned} \boldsymbol{X}' &= \mathbf{T}_{RT}\boldsymbol{X}, \\ \mathbf{H}_{PE}\boldsymbol{M}' &= \mathbf{T}_{RT}\mathbf{H}_{PE}\boldsymbol{M}. \end{aligned}$$

In practice, such homographies can be estimated linearly from image data or a respective triangulations. Obviously, they can have general form with a forth row other than $(0, 0, 0, 1)^\top$.

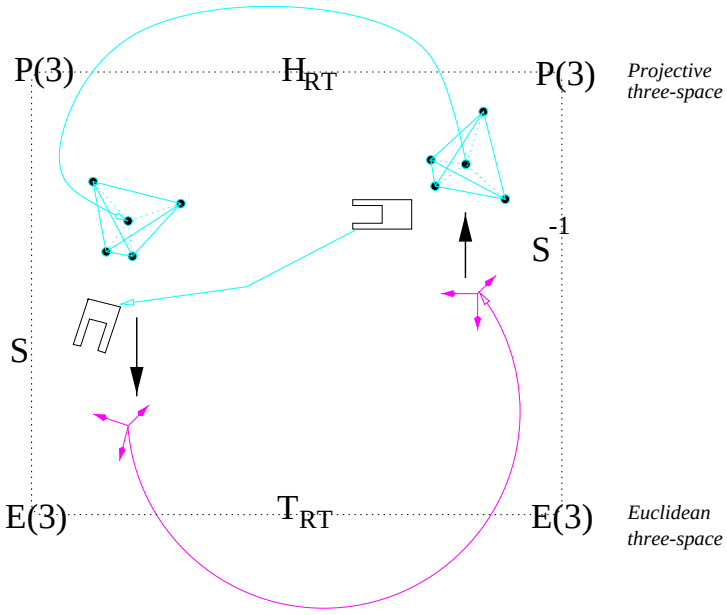

Figure 2.10: Schema depicting the algebraic construction underlying homographies of projective rigid motion.

Definition 2 (projective displacement) *Under the rigid - stereo assumption* $\mathbf{H}_{PE}$ *(2.18), a projective displacement is defined as the projective motion* $\gamma\mathbf{H}_{RT}$ *resulting from a displacement* $\mathbf{T}_{RT} \in SE(3)$ *of a rigid point-structure. This homography writes as (2.33), and it is called "projective displacement" in the normalized form having unit determinant, or equivalently a unit scale*

γ.

$$\mathbf{T}_{RT} = \begin{bmatrix} \mathbf{R} & \boldsymbol{t} \\ \mathbf{o} & 1 \end{bmatrix},$$
$$\gamma \mathbf{H}_{RT} = \mathbf{H}_{PE}^{-1} \cdot \mathbf{T}_{RT} \cdot \mathbf{H}_{PE}, \qquad \gamma = 1. \tag{2.33}$$

Thanks to the properties of similar matrices, the normalization factor $1/\gamma$ *can be calculated from a given displacement homography* $\mathbf{H}$ *by*

$$\gamma = (\det \mathbf{H})^{1/4} \text{ sign } (\text{tr } \mathbf{H}). \tag{2.34}$$

In fact, $\mathbf{H}_{RT}$ *can be considered as a 4-dimensional matrix representation of* $SE(3)$ *within the projective group.*

2.4 Velocity of rigid motion and $se(3)$

In kinematics and mechanics, the velocity of a rigid body or its "*kinematic screw*" is commonly written as a pair of three-vectors $\boldsymbol{w}$ and $\boldsymbol{v}$, representing respectively the body's "*angular velocity*" and its "*linear*" or "*translation velocity*". In the context of a mechanical system, the norms of these vectors refer to physical velocities

$$\omega_{[rad/s]} = ||\boldsymbol{w}||, \qquad v_{[m/s]} = ||\boldsymbol{v}||. \tag{2.35}$$

For a point at $(X, Y, Z)^\top$ on this body, its (linear) point-velocity is

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = [\boldsymbol{w}]_\times \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{v}, \quad \text{where} \tag{2.36}$$
$$[\boldsymbol{w}]_\times = \begin{bmatrix} 0 & -\boldsymbol{w}_z & \boldsymbol{w}_y \\ \boldsymbol{w}_z & 0 & -\boldsymbol{w}_x \\ -\boldsymbol{w}_y & \boldsymbol{w}_x & 0 \end{bmatrix} \quad \text{is antisymmetric.} \tag{2.37}$$

Geometrically, $\boldsymbol{w}$ is the rotational axis of the body, as seen in its own inertial frame, whereas $\boldsymbol{v}$ is the velocity of the body's center, as seen from an outside reference frame. (See (2.42) spatial definition of rigid body velocity.)

2.4.1 Lie algebra $se(3)$

Algebraically, we are dealing with two Lie algebras. For the rotation group $SO(3)$ this is $so(3)$, where $[\boldsymbol{w}]_\times$ is a matrix representation of this Lie algebra: the antisymmetric matrices. For the translation group $\mathbb{R}^3$ this is the Lie algebra $\mathbb{R}^3$. Together they form $se(3)$, the Lie algebra of the displacement group $SE(3)$. Its elements are often denoted as 6-vectors

$$\begin{pmatrix} \boldsymbol{w} \\ \boldsymbol{v} \end{pmatrix} \in se(3). \tag{2.38}$$

A 4×4 matrix representation of $se(3)$ can also be defined, which, as soon as Euclidean three-space is denoted homogeneously, acts on a point by left multiplication

$$(\boldsymbol{w}, \boldsymbol{v})^\top \to \hat{\mathbf{T}}_{RT} = \begin{bmatrix} [\boldsymbol{w}]_\times & \boldsymbol{v} \\ \mathbf{o}^\top & 0 \end{bmatrix}, \tag{2.39}$$

$$\hat{\mathbf{T}}_{RT} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} [\boldsymbol{w}]_\times \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \boldsymbol{v} \\ 0 \end{pmatrix}. \tag{2.40}$$

The above representations, especially the antisymmetric one for $so(3)$, rely on a Euclidean ambient space. Consider affine camera space, contrastingly. The intrinsic parameters (2.1) give rise to a slightly more general representation $\hat{\mathbf{A}}_{RT}$ of a kinematic screw. Since both frames, $\mathcal{E}$ and $\mathcal{A}$, are related by eq. (2.14), this representation is well-defined from the 3×3 matrices $\mathbf{K}, \mathbf{K}'$, only

$$\begin{pmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \\ 0 \end{pmatrix} = \begin{bmatrix} \mathbf{K}[\boldsymbol{w}]_\times \mathbf{K}'^{-1} & \mathbf{K}\boldsymbol{v} \\ 0 \;\; 0 \;\; 0 & 0 \end{bmatrix} \begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix}. \tag{2.41}$$

Clearly, $\mathbf{K}[\boldsymbol{w}]_\times \mathbf{K}'^{-1}$ is no longer antisymmetric, but it keeps the similarity properties of $[\boldsymbol{w}]_\times$, like a zero trace, while $\mathbf{K}\boldsymbol{v}$ is the affine translation vector.

2.4.2 Velocity of rigid motion

A continuous rigid motion is described by the continuously varying displacement $\mathbf{T}_{RT}(t)$, acting over time on a number of points $\boldsymbol{X}$ on a rigid structure. The trajectories $\boldsymbol{X}'(t)$ of such points have, assuming differentiability at t, a tangent $\dot{\boldsymbol{X}}'(t)$.

$$\dot{\boldsymbol{X}}'(t) = \dot{\mathbf{T}}_{RT}(t)\boldsymbol{X}$$

There are two "natural" ways to define a velocity of this motion:

Definition 3 (spatial velocity) *First, a velocity as seen by a static observer in the ambient space can be defined, where a $\hat{}$-accent indicates this particular meaning of an operator matrix $\hat{\mathbf{T}} \in se(3)$:*

$$\hat{\mathbf{T}} = \dot{\mathbf{T}}_{RT}(t)\mathbf{T}_{RT}^{-1}(t). \tag{2.42}$$

In this case, the coordinates of the "moving" points have to be differentiated. Since they are relative to the fixed spatial reference frame, so is the resulting velocity.

$$\begin{aligned} \boldsymbol{X}'(t) &= \mathbf{T}_{RT}(t)\boldsymbol{X}, \\ \dot{\boldsymbol{X}}'(t) &= \dot{\mathbf{T}}_{RT}(t)\boldsymbol{X}, \\ \dot{\boldsymbol{X}}'(t) &= \dot{\mathbf{T}}_{RT}(t)\mathbf{T}_{RT}^{-1}(t) \cdot \boldsymbol{X}'(t). \end{aligned} \tag{2.43}$$

In detail, the resulting velocity screw has a matrix representation $\hat{\mathbf{T}} \in se(3)$ *which geometrically is expressed relative to the spatial reference frame, i.e. it refers to the frame at time instant* $t = 0$.

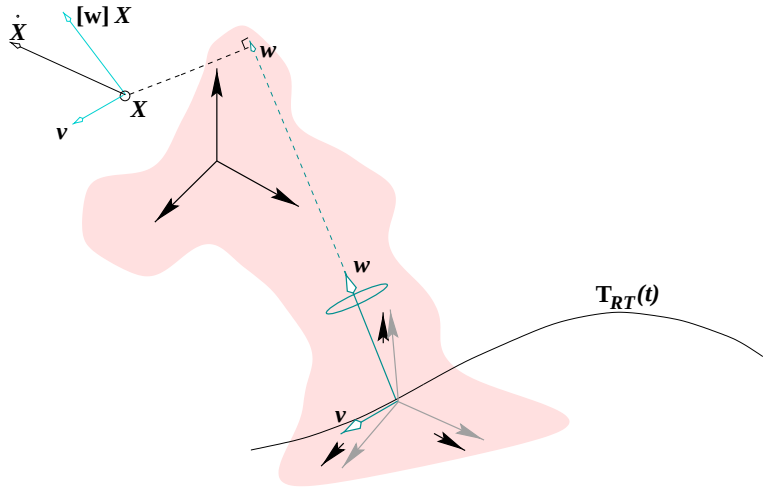

Figure 2.11: Velocity screw in the spatial reference frame.

The geometric interpretation of $\hat{\mathbf{T}}$ in terms of $\boldsymbol{v}$,$\boldsymbol{w}$ is rather counter-intuitive: $\boldsymbol{v}$ is the linear velocity of the spatial frame, as if it were rigidly moving with the object, i.e. a point traveling at t through the origin would have the velocity $\boldsymbol{v}$. The velocity $\boldsymbol{w}$ is the angular velocity at t acting on the spatial frame, as well, i.e. as if it were rigidly moving with the object. Both 3-vectors refer to spatial coordinates.

Definition 4 (body velocity) *Second, a velocity as seen by an observer on the moving body, i.e. coinciding with the current position of the moving frame, can be defined, where a* ˇ *accent indicates this particular meaning of the operator matrix* $\check{\mathbf{T}} \in se(3)$*:*

$$\check{\mathbf{T}} = \mathbf{T}_{RT}^{-1}(t)\dot{\mathbf{T}}_{RT}(t). \tag{2.44}$$

In this case, the coordinates of points in the original object frame have to be differentiated, after beign introduced using the inverse. Since these coordinates are relative to the body frame, so is the resulting velocity.

$$\begin{aligned}
\mathbf{T}_{RT}\boldsymbol{X}(t) &= \mathbf{T}_{RT}(t)\boldsymbol{X}, \\
\mathbf{T}_{RT}\dot{\boldsymbol{X}}(t) &= \dot{\mathbf{T}}_{RT}(t)\boldsymbol{X}, \\
\dot{\boldsymbol{X}}(t) &= \mathbf{T}_{RT}^{-1}(t)\dot{\mathbf{T}}_{RT}(t)\cdot\boldsymbol{X}.
\end{aligned} \tag{2.45}$$

In detail, the resulting velocity screw has a matrix representation $\check{\mathbf{T}} \in se(3)$ *which geometrically is expressed relative to the moving reference frame, i.e. it refers to the frame at time instant* t.

The geometric interpretation of $\check{\mathbf{T}}$ in terms of $\boldsymbol{v}$,$\boldsymbol{w}$ is as one commonly expects: $\boldsymbol{v}$ is the inertial linear velocity of the body frame, i.e. its origin is traveling at t away from the spatial origin with velocity $\boldsymbol{v}$. The vector $\boldsymbol{w}$ is the angular velocity acting at t directly on the body frame, itself. Both 3-vectors refer at any instant t to inertial body coordinates.

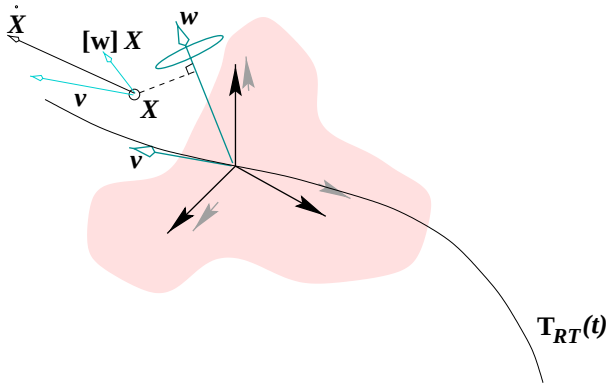

Figure 2.12: Velocity screw in the moving body-frame.

In each of the formulations (2.43) and (2.45), the coordinates of the moving points refer to a single frame, respectively. In case of $\boldsymbol{X}'(t)$, this is the spatial frame of an observer (2.43), in case of $-\boldsymbol{X}(t)$, this is the body frame (2.45) of a moving object. However, the velocity of a rigid body, or instantaneous rigid motion in general, emerges independent of a particular set of points and their particular coordinates. Still, the above chosen frames, e.g. a body frame, still reflect the idea of a moving object.

Therefore, in the same way as a rigid motion can be described for an arbitrary, just "virtual" frame in the ambient space, so can a velocity screw, which has then to be seen as an independent geometric entity (2.4). Consequently, it can also be transferred from one reference frame to another. In detail, for two frames related by $\mathbf{S}^{-1}$, such a "*screw-transfer*" is formally expressed by the adjoint map (see appendix A.3) applied to a matrix representation of $se(3)$. Instantaneous rigid motion as a geometric object is often called a "*twist*".

$$\mathrm{ad}_{\mathbf{S}}(\hat{\mathbf{T}}') = \mathbf{S}^{-1} \cdot \hat{\mathbf{T}} \cdot \mathbf{S}. \tag{2.46}$$

2.4.3 Projective representations of $se(3)$

The affine case (2.41) has been a first example for a representation independent of a Euclidean ambient space. In order to understand the fundamental difference in generalizing further to a projective representation, please note that the matrix operators $\hat{\mathbf{T}}_{RT}$ and $\hat{\mathbf{A}}_{RT}$ have always zeros on the forth row. Necessarily so, the velocity 4-vectors $\dot{\boldsymbol{N}}$ determined by such an opera-

tor (2.40) have always a forth row equal to 0. This means for a homogeneous point-vector $(U, V, W, 1)^\top$ that its forth row always remains 1.

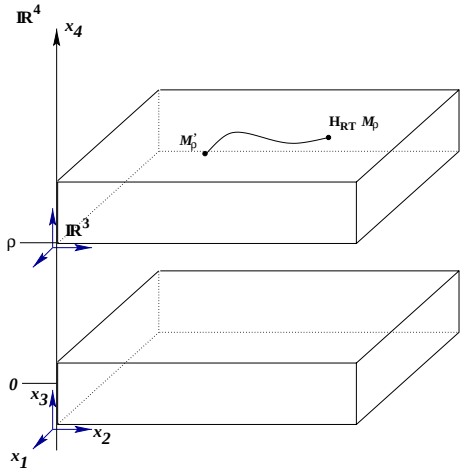

Figure 2.13: Orbit in $\mathbb{R}^4$ under action of projective displacements, drawn for a rigidly moving projective point.

Orbit of projective points

Therefore, a generalization to the projective camera frame $\mathcal{P}$ is no longer straight-forward, since the homogeneous coordinates $\boldsymbol{M}$ of a projective point contain an unknown scale factor ρ. This scalar ambiguity is also arising in the upgrade from projective to Euclidean coordinates, $\mathbf{H}_{PE}$ (2.16). In detail, the vector $\left(\boldsymbol{a}_\infty^\top, 1\right)$ representing the plane at infinity allows to extract this scale, but this plane is unknown in the uncalibrated case.

Definition 5 (displacement orbit and its height) *Each projective 4 - vector $\boldsymbol{M}_\rho$ has a scalar ρ as an implicit property, called the "height" of the point.*

$$\rho \boldsymbol{X} = \mathbf{H}_{PE} \boldsymbol{M}_\rho, \qquad \rho = \left(\boldsymbol{a}_\infty^\top, 1\right) \boldsymbol{M}_\rho. \tag{2.47}$$

In a projective ambient space, neither $\left(\boldsymbol{a}_\infty^\top, 1\right)$, nor the scale ρ in each reconstructed point are known. However, the height of a 4-vector $\boldsymbol{M}_\rho$ is invariant to left multiplications with the projective displacement group as long as the rigid-stereo assumption (2.18) holds. Consequently, the orbit of a vector $\boldsymbol{M}_\rho$ under action of projective displacements lies entirely within the hyperplane $\left(\boldsymbol{a}_\infty^\top, 1\right) \cdot \boldsymbol{M}_\rho = \rho$ of $\mathbb{R}^4$. Hence, ρ is also the height associated with the entire displacement orbit of a point, i.e. an "orbital height".

$$\boldsymbol{X}' = \mathbf{T}_{RT}\boldsymbol{X}, \qquad \boldsymbol{M}' \simeq \mathbf{H}_{RT}\boldsymbol{M}, \qquad \text{while}$$
$$\mathbf{H}_{RT}\boldsymbol{M}_\rho = \mathbf{H}_{PE}^{-1}\mathbf{T}_{RT}\mathbf{H}_{PE}\boldsymbol{M}_\rho = \mathbf{H}_{PE}^{-1}\mathbf{T}_{RT}\rho\boldsymbol{X} = \rho\mathbf{H}_{PE}^{-1}\boldsymbol{X}' = \boldsymbol{M}'_\rho.$$

Consider now the action of a continuously varying and differentiable projective displacement $\mathbf{H}_{RT}$ onto a point-vector $\boldsymbol{M}_\rho$, whose entire trajectory necessarily lies at the above identified height-orbit.

Definition 6 (Projective point-velocity) *A "projective point-velocity" is well-defined by $\dot{\boldsymbol{M}}_\rho(t)$ for each point on a rigid orbit $\boldsymbol{M}_\rho(t)$. It is related by $\mathbf{H}_{PE}$ to its metric velocity (2.36) up to the fixed but unknown height of $\boldsymbol{M}$.*

$$\rho\dot{\boldsymbol{X}} = \mathbf{H}_{PE}\dot{\boldsymbol{M}}_\rho. \tag{2.48}$$

However, a meaningful norm of a projective point-velocity like in (2.35) cannot be defined, not even up to a scalar.

Definition 7 (Velocity of projective rigid motion) *Under the rigid - stereo assumption, the velocity of a projective rigid motion can be defined, either as a "spatial velocity" in the ambient space, or as a "body velocity" of the moving rigid point-structure. These velocities are represented as a 4×4 matrix, called "tangent operators".*

$$\hat{\mathbf{H}}_{RT} = \mathbf{H}_{PE}^{-1} \quad \hat{\mathbf{T}} \quad \mathbf{H}_{PE}, \tag{2.49}$$
$$\check{\mathbf{H}}_{RT} = \mathbf{H}_{PE}^{-1} \quad \check{\mathbf{T}} \quad \mathbf{H}_{PE}, \qquad \textit{where} \tag{2.50}$$
$$\hat{\mathbf{T}}, \check{\mathbf{T}} \in se(3), \begin{bmatrix} [\boldsymbol{w}]_\times & \boldsymbol{v} \\ 0\ 0\ 0 & 0 \end{bmatrix}. \tag{2.51}$$

They are a matrix representation of the Lie algebra of projective displacements. The action of these operators on a projective displacement yields the tangent to the trajectory in $\mathbb{R}^{4\times4}$

$$\dot{\mathbf{H}}_{RT} = \hat{\mathbf{H}}_{RT}\mathbf{H}_{RT}(t). \tag{2.52}$$

The action of these operators on a point yields its projective point velocity:

$$\dot{\boldsymbol{M}}_\rho = \hat{\mathbf{H}}_{RT}\boldsymbol{M}_\rho. \tag{2.53}$$

2.5 Duality

In this section, the principle of "*duality*" is introduced in terms of dual representations for all the geometric entities used in the context of this thesis. These are points and lines in the projective image-plane and correspondingly points and planes in projective three-space. Furthermore, 3D-lines or

any linear geometric primitive are introduced in a very general way, either in a point-like, or dually in a plane-like representation. Such representations allow linear and linear iterative methods for homography estimation to be formulated, which in particular allows very general sorts of feature correspondence to be incorporated, e.g. line-to-line, point-on-line, etc.

Eventually, the 4×4 homographies acting on points are dualized, resulting in homographies acting on (hyper-)planes. A very important case are the projective displacements and especially their generators, being Lie groups with corresponding Lie algebras, for which the duals will be derived explicitly.

2.5.1 Duality in $\mathbb{P}^2$

In a projective plane, the linear geometric locii are points and lines, where the latter are the "hyperplanes" of $\mathbb{P}^2$. They are represented by homogeneous 3-vectors, which are distinguished as columns $\boldsymbol{m}$ for points and rows $\boldsymbol{l}^\top$ for lines. Then, coincidence between points and lines writes as a homogeneous linear form

$$\boldsymbol{l}^\top \boldsymbol{m} = 0, \quad \text{or} \quad \boldsymbol{m}^\top \boldsymbol{l} = 0.$$

The dual representation $\boldsymbol{m}^\perp$ of a point, which is a plane-alike representation, writes as the orthogonal complement of $\boldsymbol{m}$, and it is calculated as the transpose of the null-space of $\boldsymbol{m}^\top$,

$$\boldsymbol{m}^\perp = \begin{bmatrix} \boldsymbol{l}_1^\top \\ \boldsymbol{l}_2^\top \end{bmatrix}, \quad \text{where} \tag{2.54}$$

$$\boldsymbol{m}^\top \quad \boldsymbol{m}^\perp \quad = \mathbf{o}. \tag{2.55}$$

Geometrically, this amounts to writing a point as the intersection of two lines in $\mathbb{P}^2$. The basis-vectors of $\boldsymbol{m}^\perp$ can be chosen freely, while a convenient choice is for the resulting linear forms to express some meaningful geometric distance, e.g. a pixel-coordinate or a distance to an epipolar line.

$$\begin{bmatrix} -1 & 0 & u^* \\ 0 & -1 & v^* \end{bmatrix} \begin{pmatrix} \lambda u \\ \lambda v \\ \lambda \end{pmatrix} = \lambda \begin{pmatrix} u^* - u \\ v^* - v \end{pmatrix}. \tag{2.56}$$

The dual representation $\boldsymbol{l}^\perp$ of a line is a pair of point-vectors spanning the null-space of $\boldsymbol{l}^\top$.

$$\boldsymbol{l}^\perp = \begin{bmatrix} \boldsymbol{m}_1 & \boldsymbol{m}_2 \end{bmatrix}, \quad \text{where} \tag{2.57}$$

$$\boldsymbol{l}^\top \quad \boldsymbol{l}^\perp \quad = \mathbf{o}. \tag{2.58}$$

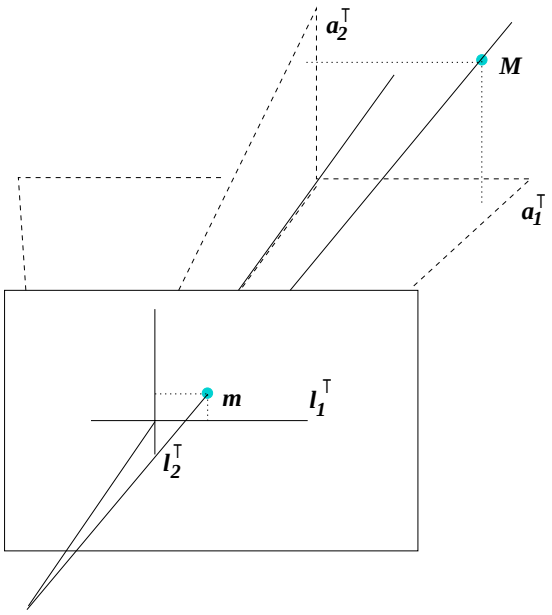

Figure 2.14: The dual of an image point, written to calculate a Euclidean reprojection-error in the pixel-plane.

2.5.2 Duality in $\mathbb{P}^3$

The geometric locii of projective three-space $\mathbb{P}^3$ which can be represented by single vectors, columns $\boldsymbol{A}$ or rows $\boldsymbol{a}^T$, are 3D-points or 3D-planes, where the latter are the "hyperplanes" of this space. Moreover, a 3D-line can be represented in this context either as the span of two points or as the intersection of two planes. In strict analogy to $\mathbb{P}^2$, the dual representations are formally characterized as an orthogonal complement and calculated as a null-space.

$$\begin{bmatrix} \boldsymbol{a}_1^\top \\ \boldsymbol{a}_2^\top \\ \boldsymbol{a}_3^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_1 \end{bmatrix} = 0, \tag{2.59}$$

$$\begin{bmatrix} \boldsymbol{a}_1^\top \\ \boldsymbol{a}_2^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix} = 0, \tag{2.60}$$

$$\begin{bmatrix} \boldsymbol{a}_1^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \end{bmatrix} = 0. \tag{2.61}$$

Geometrically, the point-alike representation is the span of a number of points, and the plane-alike representation is the intersection of a number of planes.

2.5.3 Duality of multi-columns and multi-rows in $\mathbb{P}^n$

This section introduces "*point-like*" and "*plane-like*" representations for any linear, k-dimensional geometric primitive in a n-dimensional projective space $\mathbb{P}^n$. This includes all primitives from points, to lines, planes, ..., until hyperplanes. It fully generalizes and formalizes the examples in two and three dimensions, given above. To help intuition, just remain in projective three-space $\mathbb{P}^3$.

The practical importance of this section is to provide a unified formalism for expressing a large variety of alignment-constrains, such as line-onto-line, point-on-plane, etc. This allows for instance to represent an alignment task for visual servoing. Thanks to the homogeneous formulation of these constraints, the corresponding alignment homographies, and respective visual servoing commands can be calculated in a projective ambient space.

Multi-column

As common, a point is written as a homogeneous[4] "*column-vector*" $\boldsymbol{A}$ of size $N = n + 1$, and a hyperplane by a homogeneous "*row-vector*" $\boldsymbol{a}^\top$, also of size N.

A k-dimensional subspace of $\mathbb{P}^n$ is the span of $K = k + 1$ points $\boldsymbol{A} = \alpha_1 \boldsymbol{A}_1 + \alpha_2 \boldsymbol{A}_2 + \cdots + \alpha_{k+1} \boldsymbol{A}_{k+1}$, where at least one α does not vanish. The subspace corresponds to the homogeneous range $[\boldsymbol{A}_1, \ldots, \boldsymbol{A}_K] \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_K \end{pmatrix}$ of a $N \times K$ matrix, called a "*multi-column*", denoted briefly as $[[\mathbf{A}]]$

$$[[\mathbf{A}]] = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_K], \tag{2.62}$$

Multi-row

In particular, a $n - 1$ dimensional subspace is a hyperplane $\boldsymbol{a}^\top$ spanned by a n multi-column $[\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n]$. A general point $\boldsymbol{A}$ on a hyperplane is constrained by the vanishing $n + 1$ determinant

$$\left| \boldsymbol{A}_1, \ldots, \boldsymbol{A}_n, \ \boldsymbol{A} \right| = 0, \tag{2.63}$$

such that the minors, when developing along column $n + 1$, yields the coefficients of the row $\boldsymbol{a}^\top$ in the usual plane-equation

$$\boldsymbol{a}^\top \ \boldsymbol{A} = 0. \tag{2.64}$$

This point-in-plane incidence is the basic constraint, that will be exploited for homography estimation in section 2.6.2.

[4] Homogeneous vectors or transforms mean non-zero vectors or non-singular square matrices of size $N = n + 1$, defined up to scale.

Generally, a k-dimensional subspace ($K = k + 1$) can be completed to a hyperplane by adding $n - K$ independent columns $\boldsymbol{A}_{K+1}, \ldots, \boldsymbol{A}_n$ to the K multi-column. This results in a single n multi-column $[\mathbf{A}]_i$

$$[\ \mathbf{A}\ ]_i = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_K, \boldsymbol{A}_{K+1}, \ldots, \boldsymbol{A}_n] \tag{2.65}$$

and thus in one hyperplane-constraint

$$\Big|[\mathbf{A}]_i,\ \boldsymbol{A}\Big| = 0, i.e. \qquad \boldsymbol{a}_i^\top \boldsymbol{A} = 0. \tag{2.66}$$

A total of $n - k$ independent multi-columns $[\mathbf{A}]_i$ can be build in this way, and the respective $n - k$ independent linear constraints uniquely define the subspace by

$$\begin{bmatrix} \boldsymbol{a}_1^\top \\ \vdots \\ \boldsymbol{a}_{n-k}^\top \end{bmatrix} \boldsymbol{A} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad \text{i.e.} \qquad [[\ \boldsymbol{a}\ ]] \cdot \boldsymbol{A} = \mathbf{o}. \tag{2.67}$$

The $(n - k) \times (n + 1)$ matrix $[[\mathbf{a}]]$ is called a $n - k$ "*multi-row*". Now, the null-space of the multi-row coincides with the original subspace.

In summary, a k-dimensional subspace of $\mathbb{P}^n$ is represented by the point-wise span or range of a $k + 1$ multi-column, or dually by the plane-wise intersection or the kernel of a $n - k$ multi-row.

For instance in $\mathbb{P}^2$, a point is a 1 (multi)-column or a 2 multi-row, and a line is a 2 multi-column or a 1 (multi)-row. In $\mathbb{P}^3$, a point also is a 1 (multi)-column, but a 3 multi-row, a line is a 2 multi-column or a 2 multi-row, and a plane is a 3 multi-column or a 1 (multi)-row.

2.5.4 Duals of homographies

Consider now a 4×4 projectivity $\mathbf{H}$ acting on a point $\boldsymbol{A}$. Since it is acting at the same time from the right onto $\boldsymbol{a}^\top$, its dual is necessarily $\boldsymbol{H}^{-\top}$, which is then a plane-homography, acting by left-multiplication on a plane-vector $\boldsymbol{a}$:

$$\begin{aligned} \mathbf{H}^\perp = \ \mathbf{H}^{-\top}, \qquad & \text{where} \\ \boldsymbol{a}^\top \cdot \boldsymbol{A} = \boldsymbol{a}^\top \mathbf{H}^{-1} \cdot \mathbf{H}\boldsymbol{A} &= \left(\mathbf{H}^{-\top}\boldsymbol{a}\right)^\top \cdot \mathbf{H}\boldsymbol{A} \end{aligned} \tag{2.68}$$

For the especial case of homographies defined in terms of a conjugate form, the dual is necessarily the conjugate of the dual Jordan matrix, as this can be seen by dualizing a similarity form:

$$\begin{aligned} \mathbf{H} = \mathbf{S}^{-1}\mathbf{J}\mathbf{S}. \qquad & (\mathbf{S}^{-1}\mathbf{J}\mathbf{S})^{-\top} = \\ \mathbf{H}^\perp = \mathbf{S}^\top \mathbf{J}^{-\top} \mathbf{S}^{-\top}. & \end{aligned} \tag{2.69}$$

The fact that $\mathbf{H}^{\perp}$ is acting on planes implies that it fails to cover planar motions in such a plane. Naively, this suggests that it cannot fully describe all projective motions, since there might be plane whose inside is altered but whose position as a plane remains invariant. However, the fact that five general planes are required for $\mathbf{H}^{\perp}$ to be well-defined ensures indeed that no subspace remains unconstrained.

In case of $\mathbf{H}$ being defined in terms of a matrix exponential, as this is arising in the context of the exponential representation of a Lie group, the dual of the underlying exponent matrix $\hat{\mathbf{H}}$, as the generator of the underlying Lie algebra, is necessarily $-\hat{\mathbf{H}}^{\top}$ (2.70):

$$\begin{aligned} \mathbf{H} &= \exp(\hat{\mathbf{H}}), \qquad \exp(\hat{\mathbf{H}})^{-\top} = \exp(-\hat{\mathbf{H}}^{\top}) \\ \hat{\mathbf{H}}^{\perp} &= -\hat{\mathbf{H}}^{\top} \end{aligned} \tag{2.70}$$

Additionally, it is worthwhile to consider the plane-like characterization of continuous projective motions $\mathbf{H}_{RT}^{-\top}(t)$ and their tangent space. In fact, the respective operator is $\hat{\mathbf{H}}_{RT}^{\perp}$, and applying it to a plane that is moving under action of $\mathbf{H}_{RT}^{-\top}(t)$ yields the derivative of the plane's coordinates

$$\dot{\boldsymbol{a}} = -\hat{\mathbf{H}}_{RT}^{\top}\mathbf{H}_{RT}^{-\top}(t)\boldsymbol{a} = \hat{\mathbf{H}}_{RT}^{\perp}\boldsymbol{a}(t)$$

In this sense, $\hat{\mathbf{H}}^{\perp}$ is the plane-velocity of a projective motion, and $\hat{\mathbf{H}}^{\perp}\boldsymbol{a}(t)$ is the velocity of a so moving plane. Most importantly, an invariant plane, although if points are moving within this plane, has a vanishing velocity. A very common example is the ground-plane of a moving vehicle or the infinity plane of an arbitrary rigid motion. Later in chapter 7, the dual matrix operators $-\hat{\mathbf{H}}_j^{\top}$ arising from projective robotic motion will be used to constrain the velocity of some well-chosen planes, aiming for the robot to drive along well-defined trajectories.

2.6 Numerical estimation

2.6.1 Projective reconstruction of points and lines

A straight-forward application of these dual representations is projective reconstruction of points, and furthermore of lines, as well. The reconstructions will be relative to a projective camera frame implicitly fixed by a pair of projection matrices, $\mathbf{P}$ and $\mathbf{P}'$ consistent with the fundamental matrix.

$$\begin{bmatrix} \boldsymbol{m}^{\perp} \cdot \mathbf{P} \\ \boldsymbol{m}'^{\perp} \cdot \mathbf{P}' \end{bmatrix} \boldsymbol{M} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{2.71}$$

$$\begin{bmatrix} \boldsymbol{l}^{\perp} \cdot \mathbf{P} \\ \boldsymbol{l}'^{\perp} \cdot \mathbf{P}' \end{bmatrix} \begin{bmatrix} \boldsymbol{L}_1 & \boldsymbol{L}_2 \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{2.72}$$

These linear homogeneous equations can be solved in a least-squares sense using standard eigenvector calculations or singular value decomposition. Please note, the two-dimensional null-space in the line case is written as a 4×2 matrix $\begin{bmatrix} \boldsymbol{L}_1 & \boldsymbol{L}_2 \end{bmatrix}$, which is in fact a pair of 3D-points spanning the reconstructed 3D-line.

Based on this, the reconstruction-constraints can be interpreted as the intersection of four planes in the reconstructed 3D-point (2.71), or as the intersection of two interpretation planes in the reconstructed 3D-line (2.72). Basically, the linear forms are already a plane-like description of the primitive to be reconstructed, which is transformed explicitly to a point-like description by solving the dualization.

Eventually, it is straight-forward to extend this to an iterative weighting scheme, which will assure the least-squares solution to be computed with respect to a meaningful Euclidean or maximum-likelihood measure. Such a method would back-project the 3D-solution $\boldsymbol{M}$ or $\boldsymbol{L}$ onto $\mathbb{P}^2$, and renormalize correspondingly the projection matrices within the constraint matrix for $\mathbb{P}^2$-coordinates to have pixel scales in the next iteration

$$\begin{pmatrix} \lambda u \\ \lambda v \\ \lambda \end{pmatrix} = \mathbf{P}\boldsymbol{M}, \quad \text{in next iteration use} \qquad 1/\lambda\mathbf{P} \text{ and } 1/\lambda\mathbf{P}'. \tag{2.73}$$

At convergence, the perfectly renormalized form of the linear constraints like (2.71) calculates the stereo reprojection-error, for which a least-squares solution can be found easily.

Renormalization of the reprojection-error based on such an iterative linear scheme has been employed for many estimation problems throughout this thesis. Besides reconstruction, homography estimation in $\mathbb{P}^3$ is the most frequent application, where the objective is renormalized to correspond to an image-based reprojection-error. In addition to most known estimation algorithms, homographies between points, lines, planes, or even mixed correspondences can be calculated from a single set of equations.

2.6.2 Homographies between points, lines, and planes

This section treats the estimation of the homography between two projective frames from a heterogeneous set of matching constraints, e.g. points, lines, planes in $\mathbb{P}^3$. To achieve this, the primitives are represented in the first frame by a set of points as their span, and in the second frame by a set of plane as their intersection. Corresponding incidence constraints then yield a system of linear, homogeneous equations in the coefficients of the homography matrix. It is easily solved by singular-value decomposition. Additionally, the solution can be iterated to approximate a respective image-error.

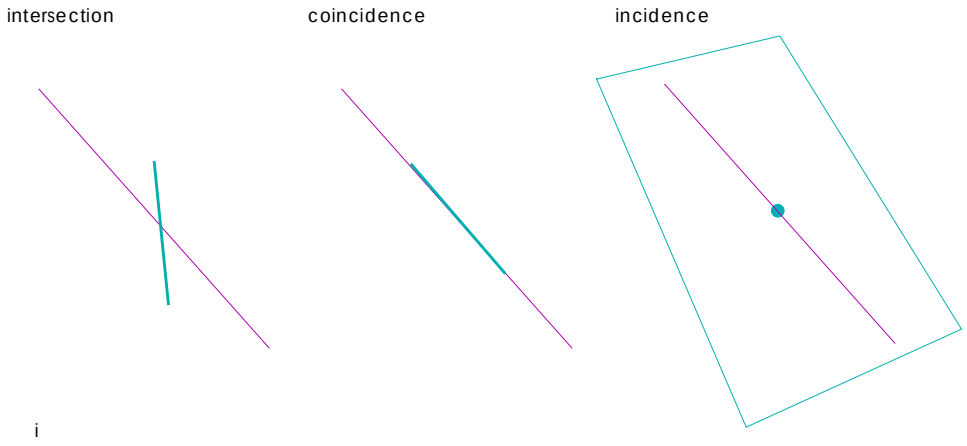

Figure 2.15: Intersection, incidence, and coincidence relation between linear geometric features, i.e. points, lines, planes, ...

Intersection, Incidence, Coincidence

This section introduces a constraint-matrix $\mathbf{C}$, that allows to describe intersection, incidence (inclusion) and strict coincidence of geometric primitives from their multi-vector representations. For two primitives $\mathcal{L}$ and $\mathcal{K}$ of subspace-dimensions l and k, they are represented by a $L = n - l$ multi-row $[[\ \mathbf{a}\ ]]$ and a $K = k + 1$ multi-column $[[\ \mathbf{A}\ ]]$. This gives rise to a $L \times K$ matrix $\mathbf{C}$, called the "*incidence-matrix*".

$$\mathbf{C} = \big[\big[\ \mathbf{a}\ \big]\big]\big[\big[\ \mathbf{A}\ \big]\big], \text{where } L \leq K \text{ w.l.o.g.} \tag{2.74}$$

The solution $\boldsymbol{x}$ of the homogeneous matrix equation $\mathbf{C} \cdot \boldsymbol{x}$ then describes the intersection of $\mathcal{K}$ and $\mathcal{L}$.

$$\begin{bmatrix} \boldsymbol{a}_1^\top \\ \vdots \\ \boldsymbol{a}_L^\top \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{A}_1, \ldots, \boldsymbol{A}_K \end{bmatrix} \cdot \boldsymbol{x} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, \text{ i.e.} \tag{2.75}$$

$$[[\ \mathbf{a}\ ]]_{L \times N} \cdot [[\ \mathbf{A}\ ]]_{N \times K} \cdot \boldsymbol{x} = \boldsymbol{O}_L \tag{2.76}$$

In detail, the two primitives are "*disjunct*" in the case of full $rank(\mathbf{C}) = L$. They do "*intersect*" if the rank is inferior to L, and their intersection is a subspace of dimension $d = L - rank(\mathbf{C}) - 1$. The latter writes as the homogeneous range of $[[\mathbf{A}]]\boldsymbol{x}$, where $\boldsymbol{x}$ is spanning a subspace of the multi-column $[[\mathbf{A}]]$. The intersection is "proper" if $d < min(k, l)$, and incidence or coincidence, otherwise. If $k = l = d$, then the two primitives do exactly coincide, otherwise one is included within the other. Consequently, $rank(\mathbf{C}) = 0$ is necessary and sufficient for subspaces of the same dimension to coincide. Thus call

$$\big[\big[\mathbf{a}\big]\big]\big[\big[\mathbf{A}\big]\big] = [\mathbf{O}]_{L \times K}, \quad \mathbf{C}_{L \times K} = \mathbf{O}_{L \times K} \tag{2.77}$$

the "*coincidence-constraint*" of multi-row and multi-column [[**a**]] and [[**A**]].

In contrast to coincidence, an inclusion, which is weaker than coincident, can be expressed by (2.77), as well. In detail, represent the lower-dimensional primitive by [[**A**]] and the higher-dimensional one by [[**a**]]. This asymmetry of the incidence-relations unfortunately restricts the possibilities of combining numerous incidence constraints (Fig. 2.16), when estimating a homography **H** as described later in section 2.6.2.

To further illustrate the introduced multi-vector constraint, the example of line-on-line coincidence in $\mathbb{P}^3$ is now given. A line through two points $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ is a one-dimensional subspace that is represented by the homogeneous range of a 2-multi-column

$$\begin{bmatrix}\boldsymbol{A}_1, & \boldsymbol{A}_2\end{bmatrix}\begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix} = \boldsymbol{M}, \qquad \text{i.e.} \quad [[\ \mathbf{A}\ ]]\ \begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix} = \boldsymbol{M}, \tag{2.78}$$

or dually by the kernel of a 2-multi-row

$$\begin{bmatrix}\boldsymbol{a}_1^T\\ \boldsymbol{a}_2^T\end{bmatrix}\boldsymbol{M} = \begin{bmatrix}0\\ 0\end{bmatrix}, \qquad \text{i.e.} \quad [[\ \mathbf{a}\ ]]\ \boldsymbol{M} = 0, \tag{2.79}$$

i.e. the intersection of two planes. For this line and a second line $[\ \mathbf{B}\ ] = [\ \boldsymbol{B}_1\ \boldsymbol{B}_2\ ]$ to coincide, the coincidence-condition below must hold:

$$[[\ \mathbf{a}\ ]]\ [[\ \mathbf{B}\ ]] = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}.$$

In consequence, coincidence of lines in $\mathbb{P}^3$ amounts to four homogeneous constraints

$$\boldsymbol{a}_1^T\boldsymbol{B}_1 = 0 \quad , \quad \boldsymbol{a}_2^T\boldsymbol{B}_1 = 0, \tag{2.80}$$
$$\boldsymbol{a}_1^T\boldsymbol{B}_2 = 0 \quad , \quad \boldsymbol{a}_2^T\boldsymbol{B}_2 = 0. \tag{2.81}$$

Coincidence is weakened to incidence of line-on-point, $\boldsymbol{A}$ on $\boldsymbol{B}_1$, if merely (2.80) is considered. The alternative case, expressing point-on-line by augmenting [[**a**]] to

$$\begin{bmatrix}\boldsymbol{a}_1^T\\ \boldsymbol{a}_2^T\\ \boldsymbol{a}_3^T\end{bmatrix}\ [[\ \mathbf{B}\ ]] = 0$$

is an example for an invalid constraint, since the two points in **B** need not lying on each of the three planes intersection in point $\boldsymbol{A}$. Once again, the incidence-relation is of asymmetric nature, i.e. they require the lower-dimensional subspace to be expressed by the multi-row.

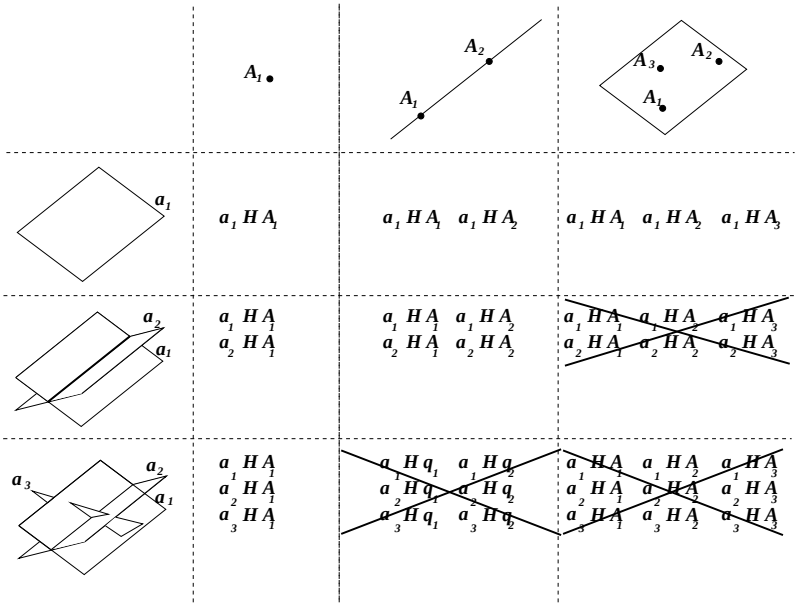

Figure 2.16: Multi-columns and multi-rows to represent coincidence and incidence constraints of matched primitives between two geometric frames.

Projectivity

In this section, a computational method is detailed allowing to estimate the homography between two projective frames from matched primitives, i.e. from the corresponding coincidence- and asymmetric incidence-constraints. For instance, one could constrain a number of coplanar points in one frame to lie on a given plane in the second frame, while at the same time two lines have to coincide.

A homography of $\mathbb{P}^n$ is represented by a homogeneous, non-singular, square matrix $\mathbf{H}$ of size $N = n + 1$

$$\mathbf{H} = \begin{bmatrix} h_{11} & \dots & h_{1N} \\ \vdots & & \vdots \\ h_{N1} & \dots & h_{NN} \end{bmatrix}, \tag{2.82}$$

or equivalently by the N^2 vector[5]

$$\boldsymbol{h} = (h_{11}, \dots, h_{1N}, \dots \ \dots, h_{N1}, \dots, h_{NN})^\top \tag{2.83}$$

composed of the rows of $\mathbf{H}$. A point $\boldsymbol{A}$ in a first frame is transformed into a second frame by $\boldsymbol{B} = \mathbf{H}\boldsymbol{A}$. Let now be $\boldsymbol{b}^\top = (b_1, b_2, \dots, b_N)$ a hyperplane

[5]The more concise tensor notation using a "Kronecker" product has been omitted here for the sake of a more intuitive notation of the equations.

in the second frame, such that a hyperplane-constraint like (2.64) becomes $\boldsymbol{b}^\top \mathbf{H} \boldsymbol{A} = 0$, or

$$\left(b_1 \boldsymbol{A}^\top, b_2 \boldsymbol{A}^\top, \ldots, b_N \boldsymbol{A}^\top\right) \boldsymbol{h} = 0, \tag{2.84}$$

$$\boldsymbol{c} \quad \boldsymbol{h} = 0, \tag{2.85}$$

in the unknown vector $\boldsymbol{h}$, where the N^2 vector $\boldsymbol{c}$ is a homogeneous constraint on the coefficients of $\mathbf{H}$.

Each element of the zero-matrix in a coincidence constraint (2.77) yields one equation $\boldsymbol{c}_i \cdot \boldsymbol{h} = 0$ like (2.85), i.e. $L \cdot K$ equations in total. The $\boldsymbol{c}_i$ are a vector representation of the incidence matrix $\mathbf{C}$. A total of $N^2 - 1$ independent equations like (2.85) are necessary to estimate $\mathbf{h}$.

The method is not restricted to the minimal number of constraints. Generally, a number r of constraints $\boldsymbol{c}_j$ yields a $r \times N^2$ system of linear homogeneous equations, where these constraints could be independent or not, and possibly inconsistent due to numerical artefact or noise in the measurements.

$$\begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_r \end{bmatrix} \boldsymbol{h} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{N^2}, \quad \text{i.e.} \quad [[\mathbf{C}]]^{r \times N^2} \boldsymbol{h} = \mathbf{o}^{r \times 1}. \tag{2.86}$$

Assuming that they fully constrain the projectivity $\mathbf{H}$, an approximate solution is the eigenvector associated with its smallest eigenvalue.

In practice, the singular value decomposition can be used to find a solution in the last column of $\mathbf{V}$.

$$\mathbf{C} = \mathbf{U}(\sigma_1, \ldots, \sigma_{n+1}) \mathbf{V}^T, \sigma_1 \geq \cdots \geq \sigma_{N^2} \tag{2.87}$$

Renormalization

The example of projective reconstruction in section (2.6.1) has already shown how the spatial distance between a point and a plane can be expressed as a distance between an image-point and an image-line, which actually is the intersection of the 3D-plane with the image plane. In the stereo case, three such "image-distances" fully constrain the spatial position of a point. This suggest to construct the multi-row $[[\mathbf{a}]]$ such that it represent an image-based error-measure. For instance, a 3D-line $\begin{bmatrix} \boldsymbol{a}_1^\top \\ \boldsymbol{a}_2^\top \end{bmatrix}$ can be constructed from its left- and right images as in (2.72). As soon as the scale λ of backprojected projective image points, e.g. of $\boldsymbol{P}\mathbf{H}\boldsymbol{B}_1$ is known, the respective rows in the equation matrix can be rescaled by $1/\lambda$, such that each $\boldsymbol{c}_j \boldsymbol{h}$ amounts to an image-distance at pixel-scale. This process can be iterated until all scales converge to $\lambda = 1$, so that the respective least-squares solution minimizes a distance in image pixel.

Chapter 3

Projective Translations

This chapter is devoted to the theory and practice of purely translational motion, and especially to such projective motions. It develops in formal detail their matrix representation as well as respective numerical methods. Translations as a motion are of practical importance since they occur often and naturally in real-world and production environments, e.g. vehicles, conveyor belts, and especially robot prismatic joints. In the context of this thesis, they have given rise to various practical methods, one for affine self-calibration of a stereo camera, one for kinematic self-calibration of prismatic joints, and finally one for visual servoing along translational trajectories.

3.1 Projective representations

3.1.1 Definition

Definition 8 (projective translation) *Under the rigid-stereo assumption* $\mathbf{H}_{PE}$ *(2.18), a projective translation is defined in (3.1) as the projective motion* $\gamma\mathbf{H}_T$ *resulting from a pure translation* $\mathbf{T}_T$ *about vector* $\boldsymbol{t} = (t_x, t_y, t_z)^\top$*. This homography writes as (3.1) and it is called "projective translation" in the normalized form with unit determinant or equivalently with unit scale* γ*.*

$$\mathbf{T}_T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\gamma\mathbf{H}_T = \mathbf{H}_{PE}^{-1} \quad \cdot \quad \mathbf{T}_T \quad \cdot \quad \mathbf{H}_{PE}, \qquad \gamma = 1. \tag{3.1}$$

Thanks to the properties of similar matrices, the normalization factor $1/\gamma$ *can be calculated from a calculated translation homography* $\mathbf{H}$ *by*

$$\begin{aligned} \gamma &= 1/4(\text{tr } \mathbf{H}), \quad \textit{or} \\ \gamma &= (\det \mathbf{H})^{1/4} \text{ sign } (\text{tr } \mathbf{H}). \end{aligned} \tag{3.2}$$

A more intuitive way to understand this is to interpret (3.1) from right to left as a sequence of transformations acting on a projective frame. Imaging this

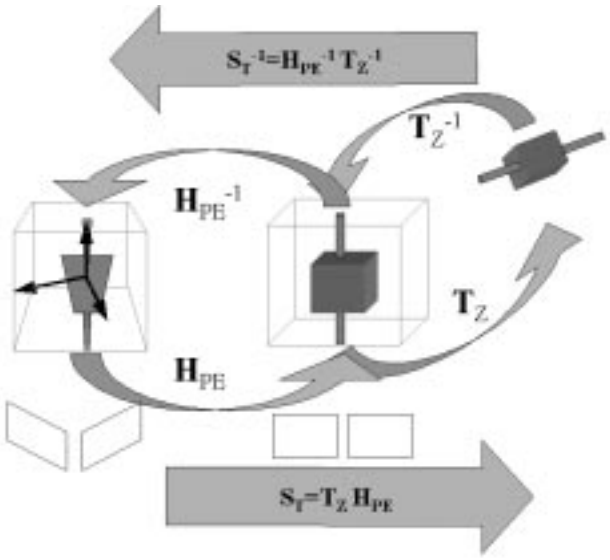

Figure 3.1: Geometric interpretation of Jordan form of projective translation.

frame as five points on the unit cube, for instance. Initially in projective, the upgrade $\mathbf{H}_{PE}$ "rectifies" the projective distortion. Being in Euclidean then, $\mathbf{T}_T$ rigidly translates the points. Finally, the inverse undoes the rectification and returns to projective (Fig. 3.1).

For the formal translation to be along the z-axis, a rotation $\mathbf{T}_Z$ has to be adjoined (3.3) without it affecting the motion, at all. The so constructed decomposition of a projective translation has almost "*Jordan form*" up to a parameter τ, which holds the length or "deflection" of the translation:

$$\mathbf{H}_T(\tau) = \mathbf{H}_{PE}^{-1}\mathbf{T}_Z^{-1} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{T}_Z\mathbf{H}_{PE}. \tag{3.3}$$

3.1.2 Jordan form

Theorem 1 (Jordan normal form of projective translations)
A projective translation always has a Jordan matrix of form $\mathbf{J}_T$ *(3.4). The regular matrix form* $\mathbf{C}_T$ *of ten parameters (3.6) defines the matrix family* $\mathbf{S}_{CT} = \mathbf{C}_T\mathbf{S}_T$ *(3.7), which characterizes all possible Jordan decompositions (3.5) of a projective translation, where* $\mathbf{S}_T$ *is an arbitrary member of this family taken as seed.*

$$\mathbf{J}_T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.4}$$

$$\mathbf{H}_T = \mathbf{S}_{CT}^{-1} \quad \mathbf{J}_T \quad \mathbf{S}_{CT}. \tag{3.5}$$

$$\mathbf{C}_T = \quad \mu \begin{bmatrix} c_{11} & c_{12} & 0 & c_{14} \\ c_{21} & c_{22} & 0 & c_{24} \\ c_{31} & c_{32} & 1 & c_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad , \tag{3.6}$$

$$\mathbf{H}_T = \quad \mathbf{S}_C^{-1}\mathbf{C}_T^{-1} \qquad \mathbf{J}_T \qquad \mathbf{C}_T\mathbf{S}_T. \tag{3.7}$$

Proof:
Looking at equation (3.1), $\mathbf{T}_T$ and by similarity also $\mathbf{H}_T$ have the characteristic polynomial $(1-\lambda)^4 = 0$ and hence a quadruple eigenvalue $\lambda = 1$. Since $rank(\mathbf{T}_T - \mathbf{I}) = 1$ and $rank(\mathbf{T}_T - \mathbf{I})^2 = 0$, their common Jordan matrix [41] has one Jordan block of order two, $\mathbf{J}_2 = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$, and two of order one, $\mathbf{J}_1 = [1]$. Among the possible permutations of blocks, the one having the form of a rigid displacement is chosen as Jordan matrix. q.e.d. (3.4)

Looking at equation (3.3), and rescaling by $1/\tau$ shows the matrix

$$\mathbf{S}_{ET} = \text{diag}\,(\tfrac{1}{\tau}, \tfrac{1}{\tau}, \tfrac{1}{\tau}, 1)\mathbf{T}_Z\mathbf{H}_{PE}. \tag{3.8}$$

to be a possible decomposition of $\mathbf{H}_T$, which can be taken as seed.

To see that $\mathbf{C}_T$ and $\mathbf{J}_T$ commute, verify

$$\mathbf{C}_T\mathbf{J}_T - \mathbf{J}_T\mathbf{C}_T = 0,$$

So, each matrix of form (3.6) yields a Jordan decomposition of $\mathbf{H}_T$, as long as it is of full rank. q.e.d. (3.7)

The decomposition $\mathbf{S}_{ET}$ introduced in (3.3) is a particular one. It transforms the space such that the Jordan matrix is acting on a Euclidean frame. In this particular case, the Jordan matrix is formally as well as geometrically a unit translation in z-direction. For a general decomposition however, this direct link with $\mathbf{H}_{PE}$ is lost.

Please note, projective transformations might be observed which have also the Jordan matrix $\mathbf{J}_T$, but which are generated other than by a pure translation, for instance if a change of perspective has occurred, e.g. zooming.

3.1.3 Eigenspaces and geometric interpretation

In the especial case of (3.8), the ambient space and its standard basis-vectors have the usual Euclidean semantics: the points $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$ are x, y, z directions at infinity, $\boldsymbol{e}_4$ is the origin o, and $\boldsymbol{e}_5$ is the unity point; the planes $\boldsymbol{e}_1^T$, $\boldsymbol{e}_2^T$, $\boldsymbol{e}_3^T$ represent the coordinate planes π_{yz}, π_{xz},π_{xy}, and $\boldsymbol{e}_4^T$ the plane at infinity $\boldsymbol{a}_\infty^\top$. Furthermore, the matrix $\mathbf{J}_T$ already has a rather convenient form for the invariant spaces of a translation to be characterized. Then, the invariant space of a projective translation can be directly characterized by the vectors in $\mathbf{S}_{CT}$, a Jordan decomposition of $\mathbf{H}_T$.

Along these lines, let's firstly consider $\mathbf{J}_T$ (3.10). It has a single three-dimensional eigenspace for $\lambda = 1$, spanned by $\mathbf{E}_T = \left[\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\right]$, the com-

plement of which is $\boldsymbol{e}_4^T$.

$$\begin{bmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&1\\ 0&0&0&1 \end{bmatrix} \cdot \begin{pmatrix} x\\ y\\ z\\ 0 \end{pmatrix} = \begin{pmatrix} x\\ y\\ z\\ 0 \end{pmatrix}, \qquad (0,0,0,1)^\top \cdot \big(x,y,z,0\big) = 0. \tag{3.9}$$

Geometrically, this amounts to the "*point-wise fixed plane*" of all directions at infinity $(X,Y,Z,0)^\top$, known as "*plane at infinity*" with standard coordinates $\boldsymbol{a}_\infty^\top = \boldsymbol{e}_4^T$.

Let's secondly consider the dual case (3.10). There is a single three-dimensional left-eigenspace for $\lambda = 1$, spanned by $\mathbf{F}_T = \big[\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_4\big]$, the orthogonal complement of which is $\boldsymbol{e}_3^\top$.

$$(x,y,0,t) \cdot \begin{bmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&1\\ 0&0&0&1 \end{bmatrix} = (x,y,0,t)\,, \qquad (0,0,1,0)^\top \cdot (x,y,0,t)^\top = 0 \tag{3.10}$$

Geometrically, this amounts to the "*family of fixed planes*" $(x,y,0,t)$, which are, in addition to the plane at infinity, all finite planes parallel to the z-direction. The intersection of this family is the z-direction at infinity $\boldsymbol{e}_3$, which in the general terms amounts to the "*direction of translation*".

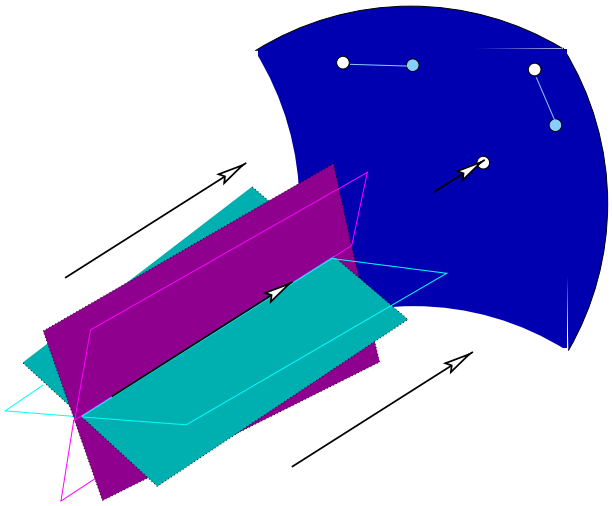

Figure 3.2: The point eigenspace of a translation spans the plane at infinity $\boldsymbol{a}_\infty^\top$, e.g. by three vanishing points in directions x, y, z. The plane eigenspace is spanned by $\boldsymbol{a}_\infty^\top$ and by two planes parallel to the translation, e.g. $\boldsymbol{a}_\infty^\top$, $\pi_{yz}^\top$, $\pi_{xz}^\top$. They intersect in the vanishing point in z-direction.

Consider finally the ambiguities introduced by the commutator $\mathbf{C}_T$ spanning the matrix family $\mathbf{S}_{CT}$ (3.5):

First, starting from the standard basis of the eigenspace $\mathbf{E}_T$, all admissible bases are fully characterized by the product $\mathbf{C}_T\mathbf{E}_T$, and by $\mathbf{F}_T\mathbf{C}_T$ in the dual case.

In the point case (3.11),

$$\begin{bmatrix} & & & \\ \boldsymbol{k}_1 & \boldsymbol{k}_2 & \boldsymbol{k}_3 & \boldsymbol{k}_4 \\ & & & \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} & 0 & c_{14} \\ c_{21} & c_{22} & 0 & c_{24} \\ c_{31} & c_{32} & 1 & c_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 0 \end{pmatrix}. \tag{3.11}$$

geometrically, the first and second column of $\mathbf{C}_T$ (3.6) allow a general pair of directions to replace the x and y directions as basis of the eigenspace, whereas the third column always remains the z-direction, i.e. the direction of translation. Additionally, as $\mathbf{C}_T$ has to be regular, an arbitrary but only a finite point $(c_{14}, c_{24}, c_{34}, 1)$ can become a new origin.

In the plane case (3.12),

$$(x, y, 0, t) \begin{bmatrix} c_{11} & c_{12} & 0 & c_{14} \\ c_{21} & c_{22} & 0 & c_{24} \\ c_{31} & c_{32} & 1 & c_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_1^T \\ \boldsymbol{h}_2^T \\ \boldsymbol{h}_3^T \\ \boldsymbol{h}_4^T \end{bmatrix} \tag{3.12}$$

the first and second row of $\mathbf{C}_T$ show that an arbitrary pair of finite planes parallel to the z-axis can replace the π_{yz} and π_{xz} planes as basis-vectors, while the third row shows in addition that any finite plane with orientation transversal to the direction can replace the plane π_{xy}. Most importantly, the forth row assures the plane at infinity to always be among the basis-vectors.

Finally, consider an arbitrary instance of $\mathbf{C}_T$, and have a look at a respective decomposition $\mathbf{S}_{CT}$, including matrix $\mathbf{H}_{PE}$. For all $\mathbf{S}_{CT}$, the rows $\boldsymbol{h}_j^\top$ and the columns $\boldsymbol{k}_i$ of $\mathbf{S}_{CT}^{-1}$, respectively, do now hold the projective coordinates of the just described geometric loci-i. It is essentially $\mathbf{S}_T$ that links the projective camera frame to the generic eigenvectors, similar to $\mathbf{H}_{PE}$.

In the point case (3.9) , $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ are a general pair of directions other than the direction of translation, which is $\boldsymbol{k}_3$, whereas $\boldsymbol{k}_4$ is always a finite point.

In the plane case (3.10), $\boldsymbol{h}_1^\top$ and $\boldsymbol{h}_2^\top$ are a general pair of finite planes. They are always parallel to the translation, whereas $\boldsymbol{h}_3^\top$ is also a finite plane, but is transversally oriented. The plane at infinity always is $\boldsymbol{h}_4^\top$.

Clearly, this is unaffected by the scalar μ in $\mathbf{C}_T$ (3.6) , as coordinates are homogeneous.

3.1.4 Generator

Theorem 2 (generator of projective translation)
Under the rigid-stereo assumption, a projective translation has the parameterization (3.13) in terms of a scalar τ and a rank one matrix $\hat{\mathbf{H}}_T$ (3.14), called "generator" or "tangent operator". This operator $\hat{\mathbf{H}}_T$ is unique up to the scaling of τ. It equals the outer product of two vectors $\boldsymbol{k}_3$ and $\boldsymbol{h}_4^\top$, which represent respectively the direction of translation and the infinity plane

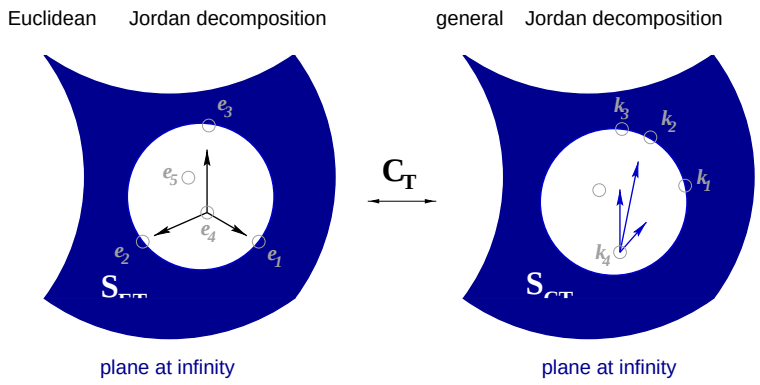

Figure 3.3: The multitude of frames compatible with the Jordan matrix of a pure translation.

associated with the stereo camera.

$$\mathbf{H}_T = \mathbf{I} + \tau\hat{\mathbf{H}}_T \tag{3.13}$$

$$\tau\hat{\mathbf{H}}_T = \boldsymbol{k}_3 \; \boldsymbol{h}_4^\top \tag{3.14}$$

Proof:
To prove this proposition, consider a matrix $\mathbf{S}_{CT}$ arising in (3.5) and denote its rows by $\boldsymbol{h}_i^\top$ and the columns of the inverse by $\boldsymbol{k}_j$, where obviously

$$\boldsymbol{h}_i^\top \boldsymbol{k}_j = \boldsymbol{k}_j^\top \boldsymbol{h}_i = \delta_{ij}, \text{ using Kronecker's } \delta_{ij}. \tag{3.15}$$

Subtracting the identity matrix from $\mathbf{J}_T$ and pulling it out of the similarity relation (3.5) yields:

$$\mathbf{H}_T = \begin{bmatrix} & & & \\ \boldsymbol{k}_1 & \boldsymbol{k}_2 & \boldsymbol{k}_3 & \boldsymbol{k}_4 \\ & & & \end{bmatrix} \begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \end{bmatrix} \begin{bmatrix} & \boldsymbol{h}_1^\top & \\ & \boldsymbol{h}_2^\top & \\ & \boldsymbol{h}_3^\top & \\ & \boldsymbol{h}_4^\top & \end{bmatrix} + \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix} \tag{3.16}$$

$$= \boldsymbol{k}_3 \cdot \boldsymbol{h}_4^\top + \mathbf{I}$$

$$= \mathbf{I} + \hat{\mathbf{H}}_T \tag{3.17}$$

q.e.d. (3.13)

Uniqueness of $\hat{\mathbf{H}}_T$ is still an issue, since changing $\mathbf{S}_{CT}$ to another $\mathbf{C}_T\mathbf{S}_T$ might also change $\boldsymbol{k}_3$ and $\boldsymbol{h}_4^\top$. However, since this affects the one side by μ and the inverse's side reciprocally by $1/\mu$, the outer product $\hat{\mathbf{H}}_T$ itself remains unaffected. A more general way to show this is to rewrite $\hat{\mathbf{H}}_T$ using the matrix $\hat{\mathbf{J}}_T$, which actually is the generator of z-translation in the standard representation of $se(3)$.

$$\hat{\mathbf{J}}_T = \begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&1 \\ 0&0&0&0 \end{bmatrix}, \tag{3.18}$$

$$\hat{\mathbf{H}}_T = \mathbf{S}_{CT}^{-1} \quad \hat{\mathbf{J}}_T \quad \mathbf{S}_{CT} = \boldsymbol{k}_3 \cdot \boldsymbol{h}_4^\top. \tag{3.19}$$

It is then sufficient to verify that $\mathbf{C}_T$ commutes with $\hat{\mathbf{J}}_T$ in order to show:

$$\mathbf{S}_T^{-1}\mathbf{C}_T^{-1} \cdot \hat{\mathbf{J}}_T \cdot \mathbf{C}_T\mathbf{S}_T = \mathbf{S}_T^{-1}\mathbf{C}_T^{-1}\mathbf{C}_T \cdot \hat{\mathbf{J}}_T \cdot \mathbf{S}_T = \hat{\mathbf{H}}_T. \tag{3.20}$$

q.e.d. (3.14)

There is no natural choice for the scale factor τ and the respective scaling of $\hat{\mathbf{H}}_T$. They can be either set to an a-priori given value for τ, e.g. a joint encoder reading, or can just be normalized to $\tau = 1$, assuming thus a unit translation. Another possibility is to normalize the generator to $||\hat{\mathbf{H}}_T|| = 1$ and to adjust τ correspondingly.

3.1.5 Exponential and logarithm

Theorem 3 (exponential form and logarithm)
The matrix exponential of the operator matrix exists and has the closed form (3.21). It maps a generator to a corresponding projective translation $\mathbf{H}_T(\tau)$. The matrix logarithm of a projective translation exists and has the closed forms (3.22) or (3.23). It thus maps the projective translation to the respective operator $\tau\hat{\mathbf{H}}_T$.

$$\exp\left(\tau\hat{\mathbf{H}}_T\right) = \mathbf{I} + \tau\hat{\mathbf{H}}_T = \mathbf{H}_T, \tag{3.21}$$

$$\log\left(\mathbf{H}_T\right) = \mathbf{H}_T - \mathbf{I} = \tau\hat{\mathbf{H}}_T, \tag{3.22}$$

$$\log\left(\mathbf{H}_T\right) = \tfrac{1}{2}\left(\mathbf{H}_T - \mathbf{H}_T^{-1}\right) = \tau\hat{\mathbf{H}}_T. \tag{3.23}$$

Proof:
First, consider the exponential (3.21). The matrix $\hat{\mathbf{H}}_T = \boldsymbol{k}_3\boldsymbol{h}_4^\top$ has rank equal to 1, trace equal to 0, and is nilpotent with order 1

$$\hat{\mathbf{H}}_T^2 = \mathbf{O}, \qquad \text{since } \boldsymbol{k}_3\boldsymbol{h}_4^\top\boldsymbol{k}_3\boldsymbol{h}_4^\top = \boldsymbol{k}_3\delta_{34}\boldsymbol{h}_4^\top = 0. \tag{3.24}$$

Thus, all super-linear terms in the formal matrix exponential (A.1) vanish, which demonstrates (3.21). q.e.d.

Second, consider the logarithm. The formulation in (3.22) follows directly from (3.13), whereas the formulation in (3.23) becomes evident after noticing that $\mathbf{H}_T^{-1} = \mathbf{I} - \tau\hat{\mathbf{H}}_T$. q.e.d.

3.1.6 Lie group and Lie algebra

Theorem 4 (Lie group and Lie algebra of projective translations)
The projective translations $\mathbf{H}_T(\tau)$ are a Lie group and the operators $\tau\hat{\mathbf{H}}_T$ are a corresponding Lie algebra. They are one dimensional and the group is Abelian. Both are isomorphic to $\mathbb{R}^1$, once the Lie group $\mathbb{R}^1$ and once the Lie algebra $\mathbb{R}^1$. The introduced 4×4 matrices, $\mathbf{H}_T(\tau)$ and $\tau\hat{\mathbf{H}}_T$, can be regarded as a respective 4-dimensional matrix representation.

Proof:
A group manifold ϕ associated with projective translations (3.21) is for instance

$$\phi : \mathbb{R} \to \mathbb{R}^{4\times 4} : \mathbf{H}_T(\tau) = \mathbf{I} + \tau\hat{\mathbf{H}}_T. \tag{3.25}$$

It is linear in $\tau \in \mathbb{R}$, and thus is trivially in C^∞, i.e. it is continuously differentiable or "smooth". The group axioms are directly verified below. Equations (3.27) and (3.29) also demonstrate the remaining conditions on the multiplication and inverse to be continuous and differentiable.

identity:		$\mathbf{H}_T(0)$	$= \mathbf{I}$	(3.26)
closure:	$\mathbf{H}_T(\tau_1)$	$\mathbf{H}_T(\tau_2)$	$= \mathbf{I} + \tau_1\hat{\mathbf{H}}_T + \tau_2\hat{\mathbf{H}}_T + \tau_1\tau_2\hat{\mathbf{H}}_T^2$	
			$= \mathbf{I} + (\tau_1 + \tau_2)\hat{\mathbf{H}}_T$	(3.27)
commutative:	$\mathbf{H}_T(\tau_2)$	$\mathbf{H}_T(\tau_1)$	$= \mathbf{I} + (\tau_1 + \tau_2)\hat{\mathbf{H}}_T$	(3.28)
inverse:	$\mathbf{H}_T(\tau)$	$\mathbf{H}_T(-\tau)$	$= \mathbf{I} + (\tau - \tau)\hat{\mathbf{H}}_T = \mathbf{I}$	(3.29)

A manifold ψ associated with the operators is for instance (3.30). It trivially assures the linear structure of the addition.

$$\psi : \mathbb{R} \to \mathbb{R}^{4\times 4} : \hat{\mathbf{H}}_T(\tau) = \tau\hat{\mathbf{H}}_T. \tag{3.30}$$

The product operation is the Lie bracket

$$\left[\hat{\mathbf{H}}_{T1}, \hat{\mathbf{H}}_{T2}\right] = \hat{\mathbf{H}}_{T2}\hat{\mathbf{H}}_{T1} - \hat{\mathbf{H}}_{T1}\hat{\mathbf{H}}_{T2},$$

which can be expressed using the product in $\mathbb{R}^{4\times 4}$. Due to the group's Abelian structure, it is always an annihilation

$$\left[\tau_1\hat{\mathbf{H}}_T, \tau_2\hat{\mathbf{H}}_T\right] = \mathbf{O}, \tag{3.31}$$

so that the bracket trivially fulfills the product axioms.

The above defined manifolds ϕ and ψ are both differentiable isomorphisms with respect to $\mathbb{R}^1$, once between this Lie group and $\mathbf{H}_T(\tau)$, and once between this Lie algebra and $\tau\hat{\mathbf{H}}_T$. The matrices act by direct multiplication of $\mathbb{R}^{4\times 4}$ onto itself. Additionally, the matrix forms allow also an action on projective three-space and on the group of projective displacements to be defined, as long as they have been defined with the same $\mathbf{H}_{PE}$. Conclusively, a series of 4-dimensional matrix representations of $\mathbb{R}^1$ has so been established. The latter is the abstract Lie group and Lie algebra which is underlying the projective translation group. q.e.d.

3.2 Numerical estimation

The homography of a projective translation is determined from image measures, only, but depends on them in an indirect i.e. non-linear manner. It can thus be calculated algebraically only by means of several intermediate steps, e.g. triangulatoin followed by homography estimation, followed by a SVD. Additionally, methods which are based on spatial data are preferable also from a practical point of view, since an elaborate system for continuous projective reconstruction of a dynamic scenes can be assumed. Theoretically, they have the great advantage that the projective motion $\mathbf{H}$ resulting from observations of an object in motion is linear in spatial coordinates.

After finding a homography $\gamma\mathbf{H}$, the next step is to normalise this translation homography. There are two equations (3.2) that calculate γ, where the advantage of the first one is to take into account all elements, with the inconvenience of the determinant's higher order terms. The second one is based on diagonal elements only, but is linear in those.

A homography estimate $\mathbf{H}$ will always deviate from the theoretical form of $\mathbf{H}_T$. Thus, the aim is now to find a homography closest to the estimate with respect to a certain distance measure.

3.2.1 SVD-based method

A first approach is to determine by means of (3.22) the Lie-algebra representation $\hat{\mathbf{H}}_T$ of the motion, which is in first instance neglecting the corrupted form. Consequently, a singular value decomposition must be used to compute the matrix of rank 1 which is closest to the estimate in terms of the matrix norm.

$$(\mathbf{U}, \mathbf{D}, \mathbf{V}^\top) = SVD(\mathbf{H} - \mathbf{I}), \qquad \mathbf{D} = \text{diag}\ (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \tag{3.32}$$

$$\hat{\mathbf{H}}_T = \mathbf{U} \cdot \text{diag}\ (\sigma_1, 0, 0, 0) \cdot \mathbf{V}^\top, \tag{3.33}$$

$$\boldsymbol{k}_3 = \mathbf{U}_{*1}, \qquad \boldsymbol{h}_4 = \mathbf{V}_{*1} \tag{3.34}$$

As long as interest is only in part of the geometrical information contained in one or many homographies, this methods can still be applied efficiently. To do so, the homographies are gathered onto a matrix stack $\mathcal{H}$, the singular value decomposition of which again yields a best approximation to the theoretical form, now with respect to the matrix norm of the entire stack, respectively. More precisely, the plane at infinity can be recovered from a number of arbitrary projective translations, even from additional projective displacements of other type. The direction at infinity common to a number of parallel projective translations can also be recovered in this way. In any cases, the rigid-stereo assumption always has to be valid.

(3.35)

3.2.2 Algorithm

In the following, a description of a practical implementation using trace-normalization and singular value decomposition is given. This method has the advantage of being numerically stable and naturally allowing for numerous input homographies to be accumulated. Suppose $\mathbf{H}^{(i)}$ to be a number of estimates of homographies resulting from translational motions, which have π_∞ as plane at infinity, and Δ_t as direction.

Step 1) $\hat{\mathbf{H}}_T^{(i)}$ from $\mathbf{H}^{(i)}, i \in \{1, \ldots, n\}$

- Normalize $\mathbf{H}^{(i)}$ by the trace (3.2) and compute $\hat{\mathbf{H}}^{(i)}$ by (3.22).

  Normalizing by the determinant or by directly calculated eigenvalues have turned out to yield less good results.

Step 2a) π_∞ and Δ_t from a single $\hat{\mathbf{H}}_T$:

- Compute the SVD, with $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \sigma_4$:

$$\hat{\mathbf{H}} = \mathbf{U}\ diag(\sigma_1, \sigma_2, \sigma_3, \sigma_4)\ \mathbf{V}^\top, \tag{3.36}$$

  To reject an outlier homography, check that the conditioning σ_1/σ_2 is large, i.e. check that the numerical rank of $\hat{\mathbf{H}}_T$ is not too far from 1. Then

$$\hat{\mathbf{H}}_T = \mathbf{U}\ diag(\sigma_1, 0, 0, 0)\ \mathbf{V}^\top = \sigma_1\ \mathbf{U}_{\bullet 1}\ \mathbf{V}_{1\bullet}$$

  is the matrix of rank 1 closest to $\hat{\mathbf{H}}$ with respect to matrix norms $||\cdot||_2$ and $||\cdot||_F$.
- Take the first column $\mathbf{U}_{\bullet 1}$=$\boldsymbol{k_3}$ as Δ_t and the first row $\mathbf{V}_{\bullet 1} = \boldsymbol{h}_4$ as π_∞.

Step 2b) Estimate π_∞ from accumulated $\mathbf{H}^{(i)}, i \in \{1, \ldots, n\}$

- Stack all n $\hat{\mathbf{H}}^{(i)}$ to

$$\mathcal{H}^{4n\times 4} = \begin{bmatrix} \hat{\mathbf{H}}^{(1)} \\ \hat{\mathbf{H}}^{(2)} \\ \ldots \\ \hat{\mathbf{H}}^{(n)} \end{bmatrix} \tag{3.37}$$

- Compute the SVD $\mathcal{H}^{4n\times 4} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$
- Take π_∞ as $\boldsymbol{h}_4 = \mathbf{V}_{\bullet 1}$

Step 2c) Estimate Δ_t from accumulated $\mathbf{H}^{(i)}, i \in \{1, \ldots, n\}$, with same direction

- Transpose and stack all n $\hat{\mathbf{H}}^{(i)^T}$ to

$$\mathcal{H}^{4n\times 4} = \begin{bmatrix} \hat{\mathbf{H}}^{(1)^\top} \\ \hat{\mathbf{H}}^{(2)^\top} \\ \cdots \\ \hat{\mathbf{H}}^{(n)^\top} \end{bmatrix} \tag{3.38}$$

- Compute the SVD $\mathcal{H}^{4n\times 4} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$
- Take Δ_t as $\boldsymbol{k}_3 = \mathbf{V}_{\bullet 1}$

3.2.3 Optimization-based method

A major drawback of the SVD-based approach is that the matrix norm implicit to the SVD is not related to image measurements and respective error statistics. A more sensible criterion would be the maximum a-posteriori likelihood with respect to noise in the image measurements. Since the problem is non-linear in nature, such a solution generally can only be found using numerical optimization. Still, for finding an initial guess for $\boldsymbol{k}_3$ and $\boldsymbol{h}_4$, the matrix norm remains an efficient criterion.

For a parameterization of $\mathbf{H}_T$ to be minimal, it has to take into account all underlying constraints. Here, let's formalize them in terms of the rank 1 matrix $\hat{\mathbf{H}}_T = \boldsymbol{k}_3\boldsymbol{h}_4^\top$, and a single constraint. The latter is easily eliminated by substituting one among the eight parameters in $\boldsymbol{k}_3$ and $\boldsymbol{h}_4$. Choose for instance k_{31} and assure h_{41} not to vanish:

$$\boldsymbol{h}_4^\top \cdot \boldsymbol{k}_3 = 0, \qquad k_{31} = -\tfrac{1}{h_{41}}(k_{32}h_{42} + k_{33}h_{43} + k_{34}h_{44}) \tag{3.39}$$

As a result, a projective translation is minimally parameterized by the vector $\boldsymbol{x}_t$ of 7-parameters:

$$\boldsymbol{x}_T = (h_{41}, k_{32}, h_{42}, k_{33}, h_{43}, k_{34}, h_{44}).$$

Now, the optimization problem is a minimization over these 7 variables of the image-error d between projective image points $\boldsymbol{m}_i, \boldsymbol{m}'_i$, appearing in the stereo images, and respective backprojected points $\boldsymbol{M}_i$, which depend on $\hat{\mathbf{H}}_T$, only, i.e. on $\boldsymbol{x}_T$. (compare section 4.4)

$$\min_{\boldsymbol{x}_T} \sum_i d\left(\boldsymbol{s}_i, \boldsymbol{P}(\boldsymbol{M}_i + \tau\hat{\mathbf{H}}_T\boldsymbol{M}_i)\right) + d\left(\boldsymbol{s}'_i, \boldsymbol{P}'(\boldsymbol{M}_i + \tau\hat{\mathbf{H}}_T\boldsymbol{M}_i)\right). \tag{3.40}$$

Note, the parameters $h_{41}, \ldots, h_{44}$ denote the plane at infinity, and can so be used for a consistent affine stratification of the model.

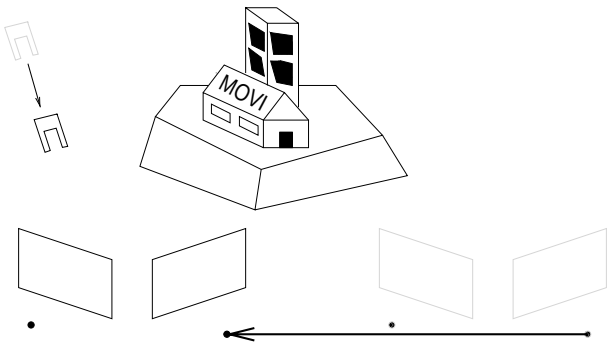

Figure 3.4: Translation of gripper in front of a rigid stereo camera (left), or of the rig relative to a static scene (right).

3.3 Experiments

This section presents experiments on simulated and real image data, designed to validate and evaluate the above introduced methods. It is taken with few modifications from [58], and it is referred to this paper for the complete context. Three different experiments have been conducted in order to study the performance of the method and the influence of several factors, such as: measurement noise, mismatches, number of homographies and estimation algorithm used (linear or non-linear). The name of the experiments refer to the objects under consideration as gripper-, grid-, and house-experiment.

It is worthwhile discussing first the evaluation criterion used, especially in the case of affine calibration. The basic idea is to evaluate three-dimensional affine structure $\boldsymbol{N}$ obtained in the experiment against affine ground-truth $\boldsymbol{X}$, which is here also the Euclidean one.

For that purpose, two error measures are defined:

- The first one calculates the average error in the length ratios of all triplets of collinear points $\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{X}_2$, where the factor $||\boldsymbol{X}_1|| - ||\boldsymbol{X}_0||$ is to rescale them to coherent length unit.

$$e_Q = \left(\frac{||\boldsymbol{N}_2 - \boldsymbol{N}_0||}{||\boldsymbol{N}_1 - \boldsymbol{N}_0||} - \frac{||\boldsymbol{X}_2 - \boldsymbol{X}_0||}{||\boldsymbol{X}_1 - \boldsymbol{X}_0||} \right) ||\boldsymbol{X}_1 - \boldsymbol{X}_0|| \tag{3.41}$$

- The second one calculates the average Euclidean distance between pairs of corresponding 3D points. For that purpose, the affine structure $\boldsymbol{N}$ in question is aligned with ground-truth Euclidean structure $\boldsymbol{X}$ using precisely the affinity $\mathbf{A}_{fit}$ which assures the best Euclidean fit between them:

$$\mathbf{A}_{fit} = \min_{A} ||\boldsymbol{X} - \mathbf{A}s\boldsymbol{N}|| \tag{3.42}$$

$$e_D = ||\boldsymbol{X} - \mathbf{A}_{fit}\boldsymbol{N}||, \tag{3.43}$$

Please note, the evaluation is based on projectively reconstructed structure which is upgraded a-posteriori to affine, i.e. after the infinity plane has been determined. This differs from a "real' affine reconstruction, which can already solve the triangulation problem in an affine space. So, the results are assured to reflect the quality of affine calibration and to be independent of a particular reconstruction method.

3.3.1 Gripper experiment

The motivation for the gripper experiment is a system for stereo self - calibration and simultaneous robot self-calibration from trial motions of prismatic joints. The simulated setup consists of a stereo rig with 20cm baseline observing a gripper of 10 cm in size from a distance of 90cm. The gripper translations are observed in terms of 18 marker points attached to it. The exact 3D structure of these markers serve as ground-truth, but it is not used by the estimation algorithm. The data results from 12 small translational gripper motions of $2cm$ each. The two diagrams compare the performance of affine calibration at increasing levels of artificially added image noise, which has been chosen Gaussian with standard deviations from $0.5px$ to $3.0px$. Figure 3.5 shows that 4 motions suffice to achieve stability. At $\sigma < 1px$, the error e_D is acceptable after only one motion and decreases below $0.2mm$. With $1 \leq \sigma \leq 2$, e_D is still below $0.5mm$, but for $\sigma \approx 3$ if $\mathbf{H}_T$ is estimated just linearly, the error increases rapidly.

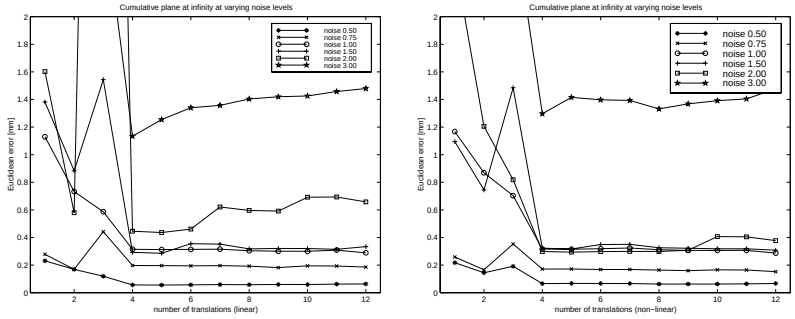

Figure 3.5: Accuracy at increasing noise levels quantified by e_D for linear and non-linear estimates of $\mathbf{H}_T$. The error mesure e_Q shows very similar results.

3.3.2 Grid experiment

The motivation for the grid-experiment is a system for off-line self-calibration of a stereo camera based on high precision image data. Another interest of this experiment is to acquire ground-truth data for the house experiment. In contrast to the gripper experiment, it is now the camera and no longer the

observed object which is being moved, and the data is acquired from a real image sequence. It consists of about 100 points per image, extracted with a precision of about $0.05px$. Ground-truth is a precise CAD model of the calibration grid. For the sequence itself, the stereo camera is moved along the three linear axis of a Cartesian robot and stereo images are grabbed at 6 equally spaced stops on each axis. Figure 3.6 shows that the results from real data are consistent with the simulations. The segments [1:5], [6:11], and [12:17] correspond to the three different direction of the motions. The accuracy is lower when the translation is aligned with the optical axis [6:11]. The type of estimation method, linear or non-linear, shows no real difference here, which could be explained by the larbe number of measurements in use. The cumulative estimate is robust against corrupted data or outliers due to a mismatch, but estimates improve from any additional data, even if it is rather noisy. The absolute error of $0.2 - 0.3mm$ compares favorably with the precision usually obtained by triangulation.

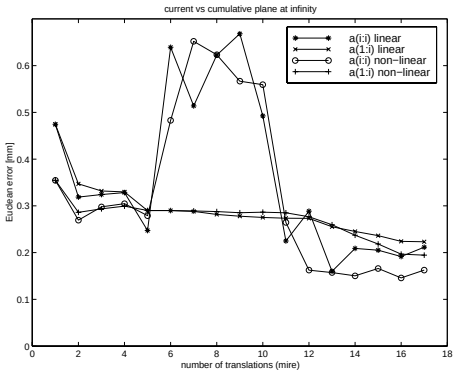

Figure 3.6: Comparison of single and cumulative estimates from linear and non-linear $\mathbf{H}_T$.

3.3.3 House experiment

The motivation for the house experiment is again a system for self-calibration of a stereo camera and of a robot's prismatic joints. Now, the focus however is on a system working on-line on arbitrary interest-point, i.e. it is able to cope with unknown objects and less precise measurements. For that purpose, the grid used above is simply replaced by a model house, and interest points are extracted and matched in an image sequence recorded for the same motion and the same rigid stereo camera as in the grid experiment. Although no ground-truth is available for the house itself, the affine data obtained in this experiment can be evaluated directly on the grid data, since the same rig has been imployed. More precisely, the affine calibration from the house is used to upgrade a perspective reconstruction of the calibration

grid (see Figure 3.7), and is evaluated as in section 3.3.2.

For the motion itself, the encoder values and the factory given kinematic model are taken as ground-truth. Qualitatively, stability and robustness are similar to the grid experiment. Quantitatively, the error e_D is slightly higher but below $1mm$ if $\mathbf{H}_T$ is estimated non-linearly.

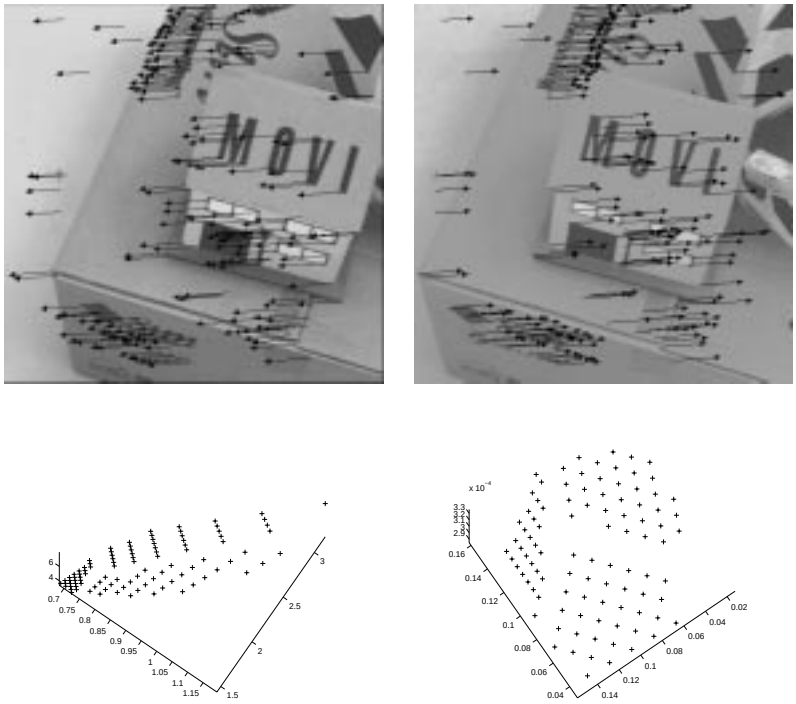

Figure 3.7: Motion field in house-scene and the resulting rectification applied to a calibration grid: projective (left), affine (right).

Chapter 4

Projective Rotations

This chapter is devoted to the theory and practice of purely rotational motion, and especially to such projective motions. It develops in formal detail their matrix representation as well as respective numerical methods. Rotations as a motion are of practical importance since they occur often and naturally in real-world and production environments, e.g. planar motions of vehicles, but especially robot revolute joints. In the context of this thesis, they have given rise to various practical methods, one for self-calibrating, pointing, and homing of a pan-tilt mounted stereo camera, one for self-calibration of the projective kinematics of a six axis robot with additional Euclidean calibration of the stereo camera, and one for visual servoing of pure rotations or planar motions.

4.1 Metric representations

4.1.1 Orthogonal matrix

Rotations of three-dimensional space are revolutions around an axis passing through the origin $\boldsymbol{o} = (0,0,0)^\top$. They are usually represented by 3×3 matrices $\mathbf{R}$ out of the "*special orthogonal group*" $SO(3)$

$$\mathbf{R} \in SO(3): \qquad det(\mathbf{R}) = 1, \qquad \mathbf{R}\mathbf{R}^T = \mathbf{I}. \tag{4.1}$$

For a unit direction vector $\boldsymbol{w}$ and for an angle θ, the corresponding rotation matrices are determined by the classical "*Rodriguez equation*"

$$\mathbf{R}_{\boldsymbol{w}}(\theta) = \exp(\theta[\boldsymbol{w}]_\times) = \mathbf{I} + \sin\theta[\boldsymbol{w}]_\times + (1-\cos\theta)[\boldsymbol{w}]_\times^2, \tag{4.2}$$

where $[\boldsymbol{w}]_\times$ stands for the anti-symmetric matrix corresponding to $\boldsymbol{w}$. They form a three-dimensional Lie algebra $so(3)$ in the components of $\boldsymbol{w}$, and the Rodriguez equation maps such an operator to an element of SO(3), the corresponding Lie group.

4.1.2 Pure rotation

Rotations inside three-dimensional space are a much larger class of motions where the rotations are no longer restricted to be around the origin. In this sense, any point $\mathbf{C}$ in three-space gives rise to a three-dimensional rotation group, denoted $\mathbf{T}_{R,C}$. It is a subgroup of the displacement group $SE(3)$ but it is in general not a subgroup of the standard rotation group $SO(3)$. In addition, a spatial line through $\boldsymbol{C}$ with fix direction $\boldsymbol{w}$ gives rise to a one-dimensional subgroup $\mathbf{T}_{R,\boldsymbol{C},\boldsymbol{w}}$ of $SE(3)$ which describes revolutions around this axis.

$$\mathbf{T}_{R,\boldsymbol{C},\boldsymbol{w}} = \begin{bmatrix} \mathbf{R}_{\boldsymbol{w}} & \mathbf{R}_{\boldsymbol{w}}\boldsymbol{C} - \boldsymbol{C} \\ \mathbf{o}^\top & 1 \end{bmatrix} \tag{4.3}$$

Although commonly used in literature, the definition of a pure rotations as the motions in $SO(3)$ fails to be a coordinate-free criteria. However, as soon as this criteria is weakened to the motions which are just similar to $SO(3)$ by an arbitrary displacement $\mathbf{T}$, it becomes a coordinate-free one. For a given displacement $\mathbf{T}$, a three-dimensional subgroup of $SE(3)$ is so defined, and additionally for a given direction $\boldsymbol{w}$, a one-dimensional subgroup is obtained.

Definition 9 (pure rotation) *A "pure rotation" is any rigid displacement $\mathbf{T}_R(\theta)$ which is similar to a rotational motion in $SO(3)$. The similarity transform must be an arbitrary displacement $\mathbf{T} \in SE(3)$. In terms of screw motions, a pure rotation is a screw with vanishing pitch.*

$$\mathbf{T}_R = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{R} & \mathbf{o} \\ \mathbf{o}^T & 1 \end{bmatrix} \mathbf{T}. \tag{4.4}$$

Clearly, a displacement $\mathbf{T}_w$ that moves the rotation axis onto the z-axis always exists. For $\boldsymbol{w}$ and two unit vectors $\boldsymbol{u}$, $\boldsymbol{v}$ forming an orthogonal triad, this rotation is explicitly

$$\mathbf{T} = \begin{bmatrix} \boldsymbol{u} & \boldsymbol{v} & \boldsymbol{w} & 0 \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \tag{4.5}$$

A conjugation by $\mathbf{T}_w$ is used to obtain a formal rotation around the z-axis, which already corresponds to a real Jordan decomposition of the rotation

$$\mathbf{T}_R = \mathbf{T}^{-1}\mathbf{T}_w^{-1} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{T}_w\mathbf{T} \tag{4.6}$$

4.2 Projective representations

4.2.1 Definition

Definition 10 (projective rotation) *Under the rigid-stereo assumption $\mathbf{H}_{PE}$ (2.18), a projective rotation is defined as the projective motion $\gamma\mathbf{H}_R$*

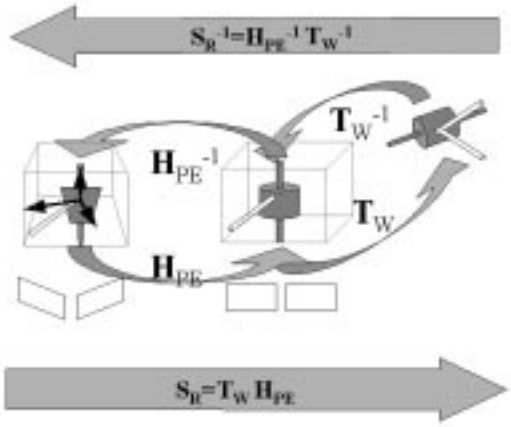

Figure 4.1: Geometric interpretation of the Jordan decomposition of a projective rotation.

resulting from a pure rotation $\mathbf{T}_R$ *in three-space. Let* θ *be the angle of rotation and let a point* $\boldsymbol{C}$ *and a unit direction* $\boldsymbol{w}$ *describe the axis of rotation, where* $\boldsymbol{w}$ *defines also the* 3×3 *rotation matrix* $\mathbf{R}_{\boldsymbol{w}}$ *(4.2).*

Now, the homography $\mathbf{H}_R$ *writes as (4.7). It is called "projective rotation" in the normalized form with unit determinant or equivalently with unit scale* γ*.*

$$\mathbf{T}_R(\theta) = \begin{bmatrix} \mathbf{R}_{\boldsymbol{w}}(\theta) & \mathbf{R}_{\boldsymbol{w}}\boldsymbol{C}-\boldsymbol{C} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

$$\gamma\mathbf{H}_R(\theta) = \mathbf{H}_{PE}^{-1} \quad \cdot \quad \mathbf{T}_R(\theta) \quad \cdot \quad \mathbf{H}_{PE}, \qquad \gamma = 1. \tag{4.7}$$

4.2.2 Jordan form

Theorem 5 (Jordan normal form of projective rotations)
A projective rotation always has a Jordan matrix of form $\mathbf{J}_R$ *(4.8). The six-parameter regular matrix form* $\mathbf{C}_R$ *(4.10) defines the matrix family* $\mathbf{S}_{CR} = \mathbf{C}_R\mathbf{S}_R$*, which characterizes all possible Jordan decompositions (4.9) of a projective rotation, where* $\mathbf{S}_R$ *is an arbitrary member of this family taken as seed (4.11).*

$$\mathbf{J}_R = \begin{bmatrix} \cos\theta & -sin\theta & 0 & 0 \\ \sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.8}$$

$$\mathbf{H}_R = \quad \mathbf{S}_{CR}^{-1}. \qquad \mathbf{J}_R \qquad \mathbf{S}_{CR}. \tag{4.9}$$

$$\mathbf{C}_R = \qquad \mu \begin{bmatrix} a & -b & 0 & 0 \\ b & a & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & e & f \end{bmatrix}, \tag{4.10}$$

$$\mathbf{H}_R = \quad \mathbf{S}_R^{-1} \quad \mathbf{C}_R^{-1} \qquad \mathbf{J}_R \qquad \mathbf{C}_R \quad \mathbf{S}_R. \tag{4.11}$$

Proof:

Independent of the outer similarity form, a pure rotation has the characteristic polynomial $p(\lambda) = (1-\lambda^2)(\lambda^2 - 2\lambda(\cos\theta) + 1)$, and thus a complex eigenvalue pair $\lambda_\theta = \cos\theta \pm \mathrm{i}\sin\theta$ and a double eigenvalue $\lambda_1 = 1$. Hence, the complex Jordan matrix is

$$\mathbf{J}_R^* = \begin{bmatrix} \exp i & 0 & 0 & 0 \\ 0 & \exp -i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Consequently, the real Jordan matrix has a rotation block $\mathbf{J}(\theta) = \left[\begin{smallmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{smallmatrix}\right]$ and a Jordan-Block $\mathbf{J}_2(1) = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ of order two. Among the possible permutations of the blocks, the one corresponding to a rigid displacement is chosen. Here, this is a rotation by θ around the z-axis. So, the Jordan matrix of a projective rotation can be written as $\mathbf{J}_R$ (4.8).

Given an initial decomposition with $\mathbf{S}_R$, the commutator $\mathbf{C}_R$ (4.10) of the Jordan matrix $\mathbf{J}_R$ – verify $\mathbf{J}_R\mathbf{C}_R - \mathbf{C}_R\mathbf{J}_R = \mathbf{O}$ – is a necessary and sufficient characterization for all possible real Jordan decompositions (see appendix A.5), where full rank has to be assumed. q.e.d. (4.9)

4.2.3 Eigenspaces and geometric interpretation

In this section, geometrical interpretations of the general Jordan decomposition (4.9) are given, namely of the Jordan matrix, the commutator, and the outer similarity. The reasoning is based on the invariant (sub-) spaces of the transformation matrices, and the relation between Euclidean and projective coordinates of three-dimensional space. Finally, also the generator $\hat{\mathbf{H}}_R$ is interpreted in terms of its action on a general point.

Using the construction leading to (4.4) and (4.7), the rectifying transformation

$$\mathbf{S}_{ER} = \mathbf{T}_w\mathbf{T}\mathbf{H}_{PE}, \qquad \mathbf{H}_R = \mathbf{S}_{ER}^{-1}\mathbf{J}_R\mathbf{S}_{ER} \tag{4.12}$$

can always be found so that $\mathbf{J}_R$ is a rotation of a Euclidean frame, more precisely, is a rotation of this frame around the z-axis. In this special case, the standard base-vectors have the usual Euclidean semantics: $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$ are x, y, z directions at infinity, $\boldsymbol{e}_4$ is the origin $\boldsymbol{o}$, and $\boldsymbol{e}_5$ is the unity point.

In the dual basis, $\boldsymbol{e}_1^\top$, $\boldsymbol{e}_2^\top$, $\boldsymbol{e}_3^\top$ represent the coordinate planes π_{yz}, π_{xz},π_{xy}, and $\boldsymbol{e}_4^T$ the plane at infinity π_∞.

The form of the matrix $\mathbf{J}_R$ is rather convenient for characterizing eigenvectors and eigenspaces. It reveals at a glance the two-dimensional eigenspace $\mathbf{E}_A = [\boldsymbol{e}_3, \boldsymbol{e}_4]$ corresponding to the double eigenvalue λ_1, and furthermore an invariant two-dimensional space $\mathbf{E}_\perp = [\boldsymbol{e}_1, \boldsymbol{e}_2]$ corresponding to the complex eigenvalue pair λ_θ:

$$\begin{bmatrix} \cos & -\sin & 0 & 0 \\ \sin & \cos & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} x\cos - y\sin \\ x\sin + y\cos \\ z \\ w \end{pmatrix}. \tag{4.13}$$

Geometrically, $\mathbf{E}_A$ is a line of fix points which goes through the origin and points in z-direction. In the context of the rotation represented by $\mathbf{J}_R$, this line $\mathbf{E}_A$ is the rotation axis. Also, $\mathbf{E}_\perp$ represents a fix line at infinity $l_\perp$, but not a line of fix points. It consists of all directions in the x-y plane, i.e. all directions normal to the rotation axis.

Dually, there is the right-eigenspace $\mathbf{F}_A = [\boldsymbol{e}_3, \boldsymbol{e}_4]^\top$ and the invariant space $\mathbf{F}_\perp = [\boldsymbol{e}_1, \boldsymbol{e}_2]^\top$.

$$(x, y, z, d) \cdot \begin{bmatrix} \cos & -\sin & 0 & 0 \\ \sin & \cos & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = (x\cos - y\sin, x\sin + y\cos, z, d). \tag{4.14}$$

Geometrically, $\mathbf{F}_A$ is a pencil of fix planes consisting of all planes parallel to the x-y plane, i.e. all planes normal to the rotation axis, including the plane at infinity. The pole of this pencil is the above introduced line at infinity l_∞. Also, $\mathbf{F}_\perp$ is a fix pencil of planes with the z-axis as pole, i.e. a pencil intersecting in the rotation axis, but it is not a pencil of fix planes.

There is more than one possible decomposing matrix $\mathbf{S}_R$, and this ambiguity is captured by the matrix family $\mathbf{S}_{CR}$ (4.9). It is the commutator $\mathbf{C}_R$ that introduces this ambiguity.

$$\begin{bmatrix} a & -b & 0 & 0 \\ b & a & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & e & f \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} ax - by \\ bx + ay \\ cz + dw \\ ez + fw \end{pmatrix}. \tag{4.15}$$

Applying $\mathbf{C}_R$ to a given basis of $\mathbf{E}_\perp$, which is a-priori an orthogonal pair of directions, shows that the resulting basis still consists of two orthogonal directions.

$$(a\boldsymbol{e}_1 - b\boldsymbol{e}_2) \cdot (b\boldsymbol{e}_1 + a\boldsymbol{e}_2) = ab\boldsymbol{e}_1\boldsymbol{e}_1 + a^2\boldsymbol{e}_1\boldsymbol{e}_2 - b^2\boldsymbol{e}_1\boldsymbol{e}_2 - ab\boldsymbol{e}_2\boldsymbol{e}_2 = ab - ab = 0$$

Applying $\mathbf{C}_R$ to $\mathbf{E}_A$ shows that an arbitrary pair of points on the axis yields an admissible decomposition. This means that origin, finite points, and the vanishing point are indistinguishable, even for points on the rotation axis.

The dual case (4.16) is analogous:

$$(x, y, z, w) \begin{bmatrix} a & -b & 0 & 0 \\ b & a & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & e & f \end{bmatrix} = (ax + by, -bx + ay, cz + ew, dz + fw)\,. \qquad (4.16)$$

The orthogonal pair of planes $\mathbf{F}_\perp$ is just rotated around its pole to another such an orthogonal pair, whereas a pair of planes $\mathbf{F}_A$ can be mapped onto an arbitrary pair in this pencil. Again, no distinction is possible between a finite plane and the plane at infinity. This information cannot be recovered.

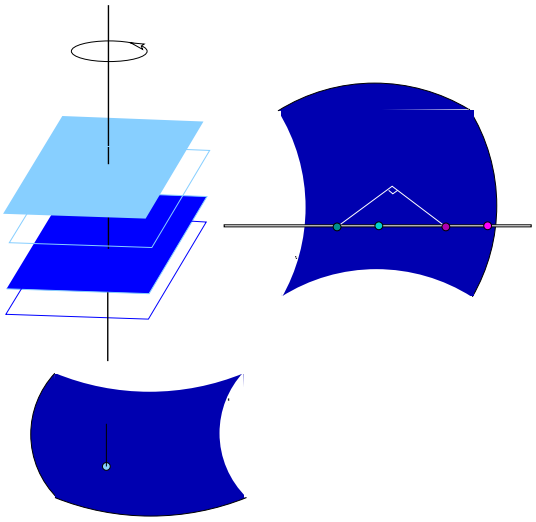

Figure 4.2: Geometric interpretation of rotation invariants and their bases.

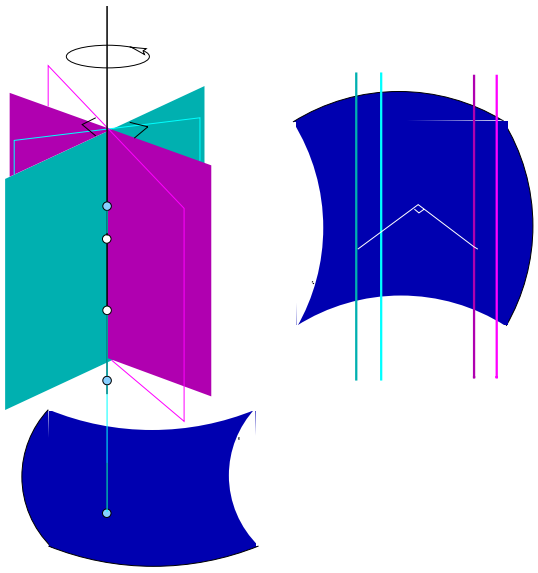

Figure 4.3: Geometric interpretation of rotation invariants and their bases.

Finally, the transition from $\mathbf{J}_R$ to a general $\mathbf{H}_R$ is made, using an $\mathbf{S}_{CR}$, which amounts to assigning projective coordinates to the above identified geometric locii while keeping their corresponding properties. Hence, $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ is a pair of normal directions with perpendicular Euclidean orientations, whereas $\boldsymbol{h}_3$ and $\boldsymbol{h}_4$ is any pair of points that span the axis. Dually, $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ are a pair of planes with perpendicular Euclidean orientations that intersect on the axis, whereas $\boldsymbol{k}_3$ and $\boldsymbol{k}_4$ are any pair of planes perpendicular to the axis.

4.2.4 Rodriguez equation and generator

Theorem 6 (Rodriguez equation)
A projective rotation is characterized by an angle θ and the rank two matrix $\hat{\mathbf{H}}_R$ (4.17), called its generator or (tangent) operator. Their product $\theta\hat{\mathbf{H}}_R$ is unique. The two column vectors $\boldsymbol{k}_1$,$\boldsymbol{k}_2$ in the operator represent two orthogonal directions, the two row vectors $\boldsymbol{h}_1^\top$, $\boldsymbol{h}_2^\top$ represent two orthogonal planes. The intersection of the two planes is the rotation axis, the span of the directions contain all normal directions of the axis. The rotation homography $\mathbf{H}_R$ is related to the operator matrix by equation (4.18), which is a formulation of the classical Rodriguez equation (4.2) in terms of 4×4 homographies.

$$\hat{\mathbf{H}}_R = \boldsymbol{k}_2\boldsymbol{h}_1^\top - \boldsymbol{k}_1\boldsymbol{h}_2^\top. \tag{4.17}$$

$$\mathbf{H}_R(\theta) = \mathbf{I} + \sin\theta\hat{\mathbf{H}}_R + (1-\cos\theta)\hat{\mathbf{H}}_R^2. \tag{4.18}$$

Proof:

To prove this, consider a seed matrix $\mathbf{S}_R$ arising in (4.9) and denote its rows by $\boldsymbol{h}_i^\top$ and its inverse's columns by $\boldsymbol{k}_j$, where obviously

$$\boldsymbol{h}_i^\top \boldsymbol{k}_j = \boldsymbol{k}_j^\top \boldsymbol{h}_i = \delta_{ij}, \text{ using Kronecker's } \delta_{ij} \ . \tag{4.19}$$

Subtracting the identity matrix from $\mathbf{J}_R$ and pulling it out of the similarity relation yields:

$$\begin{aligned}
\mathbf{H}_R &= \begin{bmatrix} & & & \\ & & & \\ \boldsymbol{k}_1 & \boldsymbol{k}_2 & \boldsymbol{k}_3 & \boldsymbol{k}_4 \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} \cos\theta\text{-}1 & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta\text{-}1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_1^\top \\ \boldsymbol{h}_2^\top \\ \boldsymbol{h}_3^\top \\ \boldsymbol{h}_4^\top \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= (\cos\theta - 1)\left(\boldsymbol{k}_1\boldsymbol{h}_1^\top + \boldsymbol{k}_2\boldsymbol{h}_2^\top\right) + \sin\theta\left(\boldsymbol{k}_2\boldsymbol{h}_1^\top - \boldsymbol{k}_1\boldsymbol{h}_2^\top\right) + \mathbf{I}
\end{aligned}$$

Expressing $\mathbf{H}_R$ as the sum of the identity and the two rank 2 matrices $\hat{\mathbf{H}}_C$, $\hat{\mathbf{H}}_R$ yields:

$$\mathbf{H}_R(\theta) = \mathbf{I} + \sin\theta\hat{\mathbf{H}}_R + (\cos\theta - 1)\hat{\mathbf{H}}_C, \tag{4.20}$$

$$\hat{\mathbf{H}}_C = \boldsymbol{k}_1\boldsymbol{h}_1^T + \boldsymbol{k}_2\boldsymbol{h}_2^T, \tag{4.21}$$

$$\hat{\mathbf{H}}_R = \boldsymbol{k}_2\boldsymbol{h}_1^T - \boldsymbol{k}_1\boldsymbol{h}_2^T. \tag{4.22}$$

Due to (4.19), these two matrices are closely related,

$$\hat{\mathbf{H}}_R^2 = -\hat{\mathbf{H}}_C, \tag{4.23}$$

so that the "*Rodriguez form of a projective rotation*" follows, q.e.d. (4.18).

$$\mathbf{H}_R = \mathbf{I} + \sin\theta\ \hat{\mathbf{H}}_R + (1 - \cos\theta)\ \hat{\mathbf{H}}_R^2. \tag{4.24}$$

As soon as the operator is written as a conjugate form of a matrix $\hat{\mathbf{J}}_R$,

$$\hat{\mathbf{J}}_R = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.25}$$

$$\hat{\mathbf{H}}_R = \mathbf{S}_{CR}^{-1} \cdot \hat{\mathbf{J}}_R \cdot \mathbf{S}_{CR}, \tag{4.26}$$

it is sufficient to verify that $\mathbf{C}_R$ commutes as well with this matrix to finally show that $\hat{\mathbf{H}}_R$ is one and the same matrix, no matter what Jordan decomposition $\mathbf{S}_{CR}$ it has been derived from.

$$\hat{\mathbf{H}}_R = \mathbf{S}_R^{-1}\mathbf{C}_R^{-1}\hat{\mathbf{J}}_R\mathbf{C}_R\mathbf{S}_R = \mathbf{S}_R^{-1}\hat{\mathbf{J}}_R\mathbf{S}_R = \boldsymbol{k}_2\boldsymbol{h}_1^\top - \boldsymbol{k}_1\boldsymbol{h}_2^\top.$$

q.e.d. (4.17)

4.2.5 Exponential and logarithm

Theorem 7 (exponential form and logarithm)
The matrix exponential of the operator exists, and it has the closed form solution (4.27). It equals the Rodriguez form (4.18), and thus maps a generator to a projective rotation. For $\sin\theta \neq 0$*, the matrix logarithm of a projective rotation exists in closed form (4.28). It maps the homography to its operator.*

$$\exp\left(\theta\hat{\mathbf{H}}_R\right) = \mathbf{I} + \sin\theta\hat{\mathbf{H}}_R + (1-\cos\theta)\hat{\mathbf{H}}_R^2 = \mathbf{H}_R. \tag{4.27}$$

$$\log\left(\mathbf{H}_R\right) = \tfrac{\theta}{2\sin\theta}\left(\mathbf{H}_R - \mathbf{H}_R^{-1}\right) = \theta\hat{\mathbf{H}}_R. \tag{4.28}$$

Proof:

First, verify the recurrences (4.29) and (4.30) using (4.19) and a simple induction argument.

$$\hat{\mathbf{H}}_R^{2n+2} = (-1)^n \hat{\mathbf{H}}_R^2 \tag{4.29}$$

$$\hat{\mathbf{H}}_R^{2n+1} = (-1)^n \hat{\mathbf{H}}_R \tag{4.30}$$

Then, consider the formal matrix exponential and separate the identity from the odd and even powers

$$\exp(\theta\hat{\mathbf{H}}_R) = \sum_{n=0}^{\infty} \frac{\theta^n \hat{\mathbf{H}}_R^n}{n!}, \tag{4.31}$$

$$\exp(\theta\hat{\mathbf{H}}_R) = \mathbf{I} + \sum_{n=0}^{\infty} \frac{\theta^{2n+2}\hat{\mathbf{H}}_R^{2n+2}}{(2n+2)!} + \sum_{n=0}^{\infty} \frac{\theta^{2n+1}\hat{\mathbf{H}}_R^{2n+1}}{(2n+1)!}.$$

Finally, an analytic solution is found after identifying the coefficients in the power series with those of sin and cos.

$$\cos\theta = 1 - \sum_{n=0}^{\infty}(-1)^n \frac{\theta^{2n+2}}{(2n+2)!} \quad , \tag{4.32}$$

$$\sin\theta = \sum_{n=0}^{\infty}(-1)^n \frac{\theta^{2n+1}}{(2n+1)!} \quad . \tag{4.33}$$

The resulting equation indeed equals $\mathbf{H}_R$ (4.18), q.e.d. (4.27)

$$\exp(\theta\hat{\mathbf{H}}_R) = \mathbf{I} + \sin\theta\hat{\mathbf{H}}_R + (1-cos\theta)\hat{\mathbf{H}}_R^2 \ .$$

To show the logarithm, it is sufficient to verify by direct calculation that (4.28) yields $\theta\hat{\mathbf{H}}_R$, which is already proven to have the correct exponential, q.e.d. (4.28),

$$\mathbf{H}_R - \mathbf{H}_R^{-1} = \mathbf{H}_R(\theta) - \mathbf{H}_R(-\theta) = \sin\theta\hat{\mathbf{H}}_R - \sin\text{-}\theta\hat{\mathbf{H}}_R = 2sin\theta\hat{\mathbf{H}}_R \ .$$

Let's briefly regard how $\hat{\mathbf{H}}_R$ acts as a matrix operator on points and let's give a simplifying geometric interpretation. The "circular trace" of a point M with projective coordinates $\boldsymbol{M}$ undergoing a projective rotation $\mathbf{H}_R(\theta)$ is

$$\begin{aligned} \boldsymbol{M}(\theta) &= \boldsymbol{M} + (1-\cos\theta)\ \boldsymbol{M}_r + \sin\theta\ \ \boldsymbol{M}_t & (4.34)\\ \mathbf{M}_r &= \hat{\mathbf{H}}_R^2 \boldsymbol{M}, \qquad \mathbf{M}_t = \hat{\mathbf{H}}_R \boldsymbol{M}, & (4.35) \end{aligned}$$

where $\mathbf{M}_r$ is the radial direction in M and $\mathbf{M}_t$ is the tangential direction in M. In this sense, multiplication by $\hat{\mathbf{H}}_R$ is analog to taking the vector product with the rotation axis $\boldsymbol{w}$ in the sense that it calculates a direction perpendicular to both, the axis and the radius. Additionally, the scale of these vectors has a well-defined value, depending on the orbital height of the initial point $\boldsymbol{M}$.

For two antipodal points, the angular terms are opposite and cancel out

$$\begin{aligned} \boldsymbol{M}(\theta) &= \boldsymbol{M} + \quad \sin(\theta)\ \boldsymbol{M}_t - \cos(\theta)\boldsymbol{M}_r \quad + \boldsymbol{M}_r,\\ \boldsymbol{M}(\theta+\pi) &= \boldsymbol{M} + \sin(\theta+\pi)\ \boldsymbol{M}_t - \cos(\theta+\pi)\boldsymbol{M}_r\ + \boldsymbol{M}_r\\ &= \boldsymbol{M} - \quad \sin(\theta)\ \boldsymbol{M}_t + \cos(\theta)\boldsymbol{M}_r \quad + \boldsymbol{M}_r, \end{aligned}$$

such that the center $\boldsymbol{C} = 1/2(\boldsymbol{M}(\theta) + \boldsymbol{M}(\theta+\pi))$ of the circle of M is

$$\boldsymbol{C} = \boldsymbol{M} + \boldsymbol{M}_r. \tag{4.36}$$

4.3 Lie group and Lie algebra

Theorem 8 (Lie group and Lie algebra of projective rotations) *The projective rotations $\mathbf{H}_R(\theta)$ are a Lie group and the operators $\theta\hat{\mathbf{H}}_R$ are a Lie algebra, both one dimensional, where the group is furthermore Abelian. They are isomorphic to the multiplicative Lie group $\mathbb{C}_1$ and its Lie algebra $\mathbb{R}$. Thus, they can be seen as 4-dimensional matrix representations of these.*

Proof:
A group manifold ϕ associated with projective rotations is for instance (4.27)

$$\phi : \mathbb{C}_1 \to \mathbb{R}^{4\times 4} : \mathbf{H}_R(\cos\theta + i\sin\theta) = \mathbf{I} + \sin\theta\hat{\mathbf{H}}_R + (1-\cos\theta)\hat{\mathbf{H}}_R^2, \tag{4.37}$$

where $\mathbb{C}_1$ is the complex unit circle. It is a sum of smooth functions and this is trivially in C^∞. Please note that $\mathbb{C}_1$ and the respective notions of continuity and differentiability have been used in order to get along with a single chart. In particular, this allows to correctly identify angles or rotations outside the 2π range.

$$(\cos\theta_1 + i\sin\theta_1)\cdot(\cos\theta_2 + i\sin\theta_2) = \cos(\theta_1+\theta_2) + i\sin(\theta_1+\theta_2) \tag{4.38}$$

$$\mathrm{e}^{\,i\theta_1}\cdot\mathrm{e}^{\,i\theta_2} = \mathrm{e}^{\,i\theta_1+i\theta_2} = \mathrm{e}^{\,i(\theta_1+\theta_2)} = \cos(\theta_1+\theta_2) + i\sin(\theta_1+\theta_2) \tag{4.39}$$

The group axioms are verified below. The equations on the multiplications and the inverse also demonstrate the smoothness of these operations.

$$\begin{aligned}
&\text{identity:}\\
\mathbf{I} &= \mathbf{S}_R^{-1}\mathbf{S}_R = \mathbf{S}_R^{-1}\mathbf{J}_R(0)\mathbf{S}_R\\
&\text{inverse:}\\
\mathbf{H}_R^{-1}(\theta) &= \left(\mathbf{S}_R^{-1}\mathbf{J}_R(\theta)\mathbf{S}_R\right)^{-1} = \mathbf{S}_R^{-1}\mathbf{J}_R^{-1}(\theta)\mathbf{S}_R\\
&= \mathbf{S}_R^{-1}\mathbf{J}_R(-\theta)\mathbf{S}_R\\
&\text{closure:}\\
\mathbf{H}_R(\theta_1)\cdot\mathbf{H}_R(\theta_2) &= \mathbf{S}_R^{-1}\mathbf{J}_R(\theta_1)\mathbf{S}_R\mathbf{S}_R^{-1}\mathbf{J}_R(\theta_2)\mathbf{S}_R\\
&= \mathbf{S}_R^{-1}\mathbf{J}_R(\theta_1)\mathbf{J}_R(\theta_2)\mathbf{S}_R = \mathbf{S}_R^{-1}\mathbf{J}_R(\theta_1+\theta_2)\mathbf{S}_R,\\
&\text{abelian}\\
\mathbf{H}_R(\theta_2)\cdot\mathbf{H}_R(\theta_1) &= \mathbf{S}_R^{-1}\mathbf{J}_R(\theta_2+\theta_1)\mathbf{S}_R.
\end{aligned}$$

A manifold ψ associated with the operator is for instance (4.40), which trivially assures them to form a linear space.

$$\psi : \mathbb{R} \to \mathbb{R}^{4\times4} : \hat{\mathbf{H}}_R(\theta) = \theta\hat{\mathbf{H}}_R. \tag{4.40}$$

The arguments of ψ and ϕ are classically related by the complex exponential as diffeomorphism

$$\cos\theta + i\sin\theta = \exp(i\theta).$$

The product operation is defined as the Lie bracket

$$[\theta_1\hat{\mathbf{H}}_R, \theta_2\hat{\mathbf{H}}_R] = \theta_1\theta_2\hat{\mathbf{H}}_R^2 - \theta_2\theta_1\hat{\mathbf{H}}_R^2 = \mathbf{O}, \tag{4.41}$$

and C^∞ follows directly from the product in $\mathbb{R}^{4\times4}$. Since the group is one-dimensional and Abelian, the product is necessarily an annihilation as shown above. So, the product axioms are trivially fulfilled. q.e.d.

The generalization of this treatment to rotation groups of higher dimension, e.g. spherical rotation groups, and respective projective representations is similar.

4.4 Numerical estimation

This section describes how to accurately estimate the tangent operator $\hat{\mathbf{H}}_R$ of an observed projective rotation from joint angle and image data, only. The overall goal is to use an uncalibrated stereo camera for recovering a projective kinematics from joint-wise trial motions of an articulated mechanism, here a pan-tilt mechanism with the cameras mounted onto.

4.4.1 Objective function

Consider a single revolute joint and denote its angle θ. Move it to n positions θ_i, take n stereo images (Fig. 4.7), and extract in the images the tracks[1] $\boldsymbol{m}_i$ and $\boldsymbol{m}'_i$ of points $\boldsymbol{M}_i$ on a rigid 3D structure.

The projective motion is described as $\boldsymbol{M}_i = \boldsymbol{H}_R(\theta_i)\boldsymbol{M}_0$, where the Rodriguez form (4.18) of $\boldsymbol{H}_R(\theta_i)$ is an explicit function in the tangent operator $\hat{\mathbf{H}}_R$. So, given a distance function $d(\cdot,\cdot)$ in the projective image plane, the total stereo reprojection-error along the sequence is an appropriate objective function f to be minimized in $\hat{\mathbf{H}}_R$

$$f = \sum_{i=1}^{n} d\left(\boldsymbol{m}_i, \mathbf{P}\mathbf{H}_R(\theta_i)\boldsymbol{M}_0\right) + d\left(\boldsymbol{m}'_i, \mathbf{P}'\mathbf{H}_R(\theta_i)\boldsymbol{M}_0\right). \tag{4.42}$$

The most commonly used distance function is the Euclidean distance in pixel between extracted and backprojected points, eventually weighted by the covariance matrix of measurement noise.

4.4.2 Minimal parameterization

An operator $\hat{\mathbf{H}}_R$ is characterizing the projective geometry of the respective revolute joint with respect to the observing stereo camera, where rigidity of the rig is assumed. Formally, it has 16 coefficients which are obviously constrained by the rank 2 condition. Now, the aim is to come up with a minimal vector of parameters, $\boldsymbol{x}_R$, corresponding to an operator matrix.

A projective rotation as naturally defined in equation (4.12) has 13 parameters: the upgrade matrix $\mathbf{H}_{PE}$ (2.16) holds 5 in $\mathbf{K}$ (intrinsic parameters), 3 in the plane at infinity $\boldsymbol{a}^T$, and 5 in the rotational displacement $\mathbf{T}_R$ (the angle plus the axis as 3D-line). It is known, that from a single rotation, the plane at infinity is not fully defined. Moreover, just a subset of the internal parameters can be calibrated and the respective constraints are of non-trivial orders [11]. The "physical" parameters are hence not suitable, nor are the vectors in (4.17), since they are subject to the constraints (4.19) and to the ambiguity (4.11). There, these "physical" parameters are omitted in favor of the algebraic parameters related to a Jordan decomposition.

Given an initial guess, a two-parameter family of solutions is introduced by the square $\left[\begin{smallmatrix} a & -b \\ b & a \end{smallmatrix}\right]$ (4.10). In order to resolve this ambiguity, a and b are fixed by setting $h_{11} = 1$, $h_{21} = 0$. In practice, this constraint is imposed by QR-decomposition $\mathbf{Q}\cdot\mathbf{W}$ with column-permutations $\mathbf{Y}$ to ensure dominant diagonal elements and a good numerical condition.

$$\begin{bmatrix} \boldsymbol{h}_1^T \\ \boldsymbol{h}_2^T \end{bmatrix} = \mathbf{Q}\cdot\mathbf{W}\cdot\mathbf{Y}, \quad \mathbf{W} = \begin{bmatrix} 1 & h'_{12} & h'_{13} & h'_{14} \\ 0 & h'_{22} & h'_{23} & h'_{24} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}'^T_1 \\ \boldsymbol{h}'^T_2 \end{bmatrix}.$$

[1] The vectors $\boldsymbol{m}_i$, $\boldsymbol{m}'_i$, $\boldsymbol{M}_i$ generically stand for a whole set of matched points and corresponding reconstructions.

Now, a well-conditioned form is $\hat{\mathbf{H}}'_R = \hat{\mathbf{H}}_R \mathbf{Y}^{-1}$

$$\hat{\mathbf{H}}'_R = \left[\, \boldsymbol{k}'_1 \; \boldsymbol{k}'_2 \,\right] \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right] \begin{bmatrix} \boldsymbol{h}'^T_1 \\ \boldsymbol{h}'^T_2 \end{bmatrix}, \quad \left[\, \boldsymbol{k}'_1 \; \boldsymbol{k}'_2 \,\right] = \left[\, \boldsymbol{k}_1 \; \boldsymbol{k}_2 \,\right] \mathbf{Q}. \tag{4.43}$$

Given a good initialization, an initial QR-step is sufficient; otherwise h'_{11}, h'_{21} should be monitored during the minimization. Besides h_{11} and h_{21}, further four parameters are eliminated by imposing the bilinear constraints (4.19), such that $\hat{\mathbf{H}}_R$ depends in the end only on a 10-parameter-vector $\boldsymbol{x}_R = (\, h'_{12}$, h'_{13}, h'_{14}, h'_{22}, h'_{23},h'_{24}, k'_{13},k'_{14}, k'_{23}, $k'_{24})$. Consequently, the objective f will be minimized over $\boldsymbol{x}_R$, only.

$$\min_{\boldsymbol{x}_R} f(\boldsymbol{x}_R) = f\left(\hat{\mathbf{H}}_R(\boldsymbol{x}_R)\right).$$

In order to further increase precision, the reconstructions $\boldsymbol{M}_0$ can be included in the minimization, which adds 3 parameters per point. Although computational costs increase considerably, such a "bundle-adjustment" is crucial if image noise is non-neglectable or if several different rotations have to be considered, as in case of the kinematics of a six-axes manipulator (chapter 5).

4.4.3 Initialization

After the homography has been estimated linearly, and possible refined non-linearly, it is straight-forward that any of its Jordan decompositions (4.9) gives an initial guess (4.17). An alternative is to calculate the tangent operator $\hat{\mathbf{H}}_R$ using (4.28) and to impose rank 2 using SVD. Then, one still has to calculate an eigen-decomposition

$$\begin{bmatrix} & & \\ \boldsymbol{f}_{-i} & \boldsymbol{f}_i & \\ & & \end{bmatrix} \begin{bmatrix} -i & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} \boldsymbol{e}^T_{-i} \\ \boldsymbol{e}^T_i \end{bmatrix},$$

where $\boldsymbol{f}_{-i}$, $\boldsymbol{f}_i$ and $\boldsymbol{e}^T_i$, $\boldsymbol{e}^T_{-i}$ are left- and right eigenvectors to the eigenvalues $-i$ and i. Then, they are recombined using $\Upsilon = \left[\begin{smallmatrix} 1 & -i \\ 1 & i \end{smallmatrix}\right]$ to obtain a real form like in (4.25), which determines immediately $\boldsymbol{x}_R$.

4.5 Experiments

This section presents experiments using a pan-tilt mechanism with a stereo rig mounted on top of it. The projective motion [58] of a 3D-point $\boldsymbol{M}$ can be described in function of the pan- and tilt angles, Ψ and Φ, using a corresponding operator $\hat{\mathbf{H}}_\Psi$, $\hat{\mathbf{H}}_\Phi$ for each joint:

$$\boldsymbol{M}(\Psi, \Phi) = \exp(\Psi \hat{\mathbf{H}}_\Psi) \exp(\Phi \hat{\mathbf{H}}_\Phi) \boldsymbol{M}. \tag{4.44}$$

The three experiments document the projective self-calibration of the system based on trial motions, the prediction of image-motion based on a forward projective kinematics, and the homing of the mechanism based on a numerical inversion of the projective kinematics. The methods are validated and evaluated on real image data of a complex, unknown object. The presentation is taken with few modifications from [58], and it is referred to this paper for the complete context.

4.5.1 Experimental setup

A stereo head consisting of two CCD-cameras with $12.5mm$ lenses, baseline $35cm$ and vergence-angle 75^o is mounted onto a robot end-effector. The two revolute axes of a 5-dof Cartesian robot by SINTERS constitutes the pan-tilt mechanism. It has a repeatability of 0.01^o, so encoder readings are taken as ground-truth for joint-angle differences (Ψ, Φ). The real image sequences show a small model of a mountain taken from $2m$ distance, and with tilt 20^o w.r.t. the ground-plane. We track about 200 Harris' interest-points using SSD-based cross-correlation and refine them to sub-pixel accuracy using paraboloid interpolation. The theoretically predicted accuracy of $0.3px$ is only achieved between consecutive images. Otherwise a noticeable drift of the features is introduced since errors sum up due to the sequential concatenation of correlations.

4.5.2 Non-metric calibration

In the calibration phase, two separate trajectories, the first stepping the pan-joint 20 times by 1^o, and the second stepping the tilt-joint 10 times also by 1^o, are driven. The corresponding image sequences show the mountain moving in the image horizontally (u) for pan and vertically (v) for tilt, along a path of $300px$ length. A subset of the steps and interest points is taken to establish a non-metric calibration of the system, i.e. to estimate $\hat{\mathbf{H}}_\Psi$ and $\hat{\mathbf{H}}_\Phi$ using the parameterization proposed in section 4.4. The result is evaluated against the steps and the points not used for calibration (Fig. 4.7). We evaluate backprojected against extracted points by the mean (signed) differences Δ_u, Δ_v of their u- and v-coordinates, and by the mean of the Euclidean distances in pixel calculated over the left or right image, Δ_L or Δ_R, or both, Δ_{LR}. The results show that the estimation is unbiased and accurate to sub-pixels (Fig. 4.8, Tab. 4.1). In order to eliminate a possible interference due to the drift, a reconstruction $\boldsymbol{M}_i$ at step i is evaluated just against the step $i + 1$.

4.5.3 Feed-forward prediction

In the prediction experiments, the mountain is observed from 9 general positions of the pan-tilt head, which are not on the calibration trajectories,

mean/sdv	$(\Psi, 0)$	$(0, \Phi)$	(Ψ, Φ)
Δ_u	-0.8e4/0.36	0.02/0.42	-0.04/0.73
Δ_v	-0.01/0.71	-0.04/0.50	-0.02/ 0.77
Δ_{LR}	0.66/ –	0.51/–	0.77/–

Table 4.1: Image errors in calibration and prediction:
The columns from left to right show result of the calibration of pan-joint, tilt-joint, and the feed-forward experiment. The rows give mean values and standard deviations of the error measures Δu, Δv, Δ_{LR}. All values are in pixel [px].

i.e. both joints are moved. For each pair of positions, the just established weak calibration is used to predict points (4.44) in the second image pair from the joint angles and the reconstruction done from the first image pair. Again, predicted points are evaluated against extracted points. The results show that the mean prediction error is below $1px$ (Fig. 4.9, Tab. 4.1).

Practical applications of the feed-forward prediction are in image-based target tracking, and trajectory planning to avoid occlusions, collisions, or targets out-of-view.

4.5.4 Homing

In the homing experiments, a stereo image or at least a single reconstructed point $\boldsymbol{M}_h$ defines a home position of the pan-tilt system. Suppose its projective kinematics (4.44) to be known. Given an unknown configuration of the pan-tilt head and the respective image $\boldsymbol{M}_c$ of the reference point: What is the joint-space displacement that moves the head to the home position? This amounts to solving the inverse kinematic map from the constraint

$$\boldsymbol{M}_h = \mathbf{H}_\Phi \mathbf{H}_\Psi \boldsymbol{M}_c \rightarrow (\Psi, \phi)\,. \tag{4.45}$$

In the experiments, we use again the objective function f, but minimize now over the two parameters (Ψ, ϕ) (4.45), [35], [43]. The results are evaluated against the joint-angles measured when the home image was taken. The results presented in Figure 4.10 show that a single point is sufficient to home a pan-tilt head with a precision better than 0.1^o for pan and 0.02^o for tilt. Practical applications are visual reset of the pan-tilt system to zero-reference during power-up, visual measurement of joint-angles, visual homing, or target tracking with the pan-tilt mechanism.

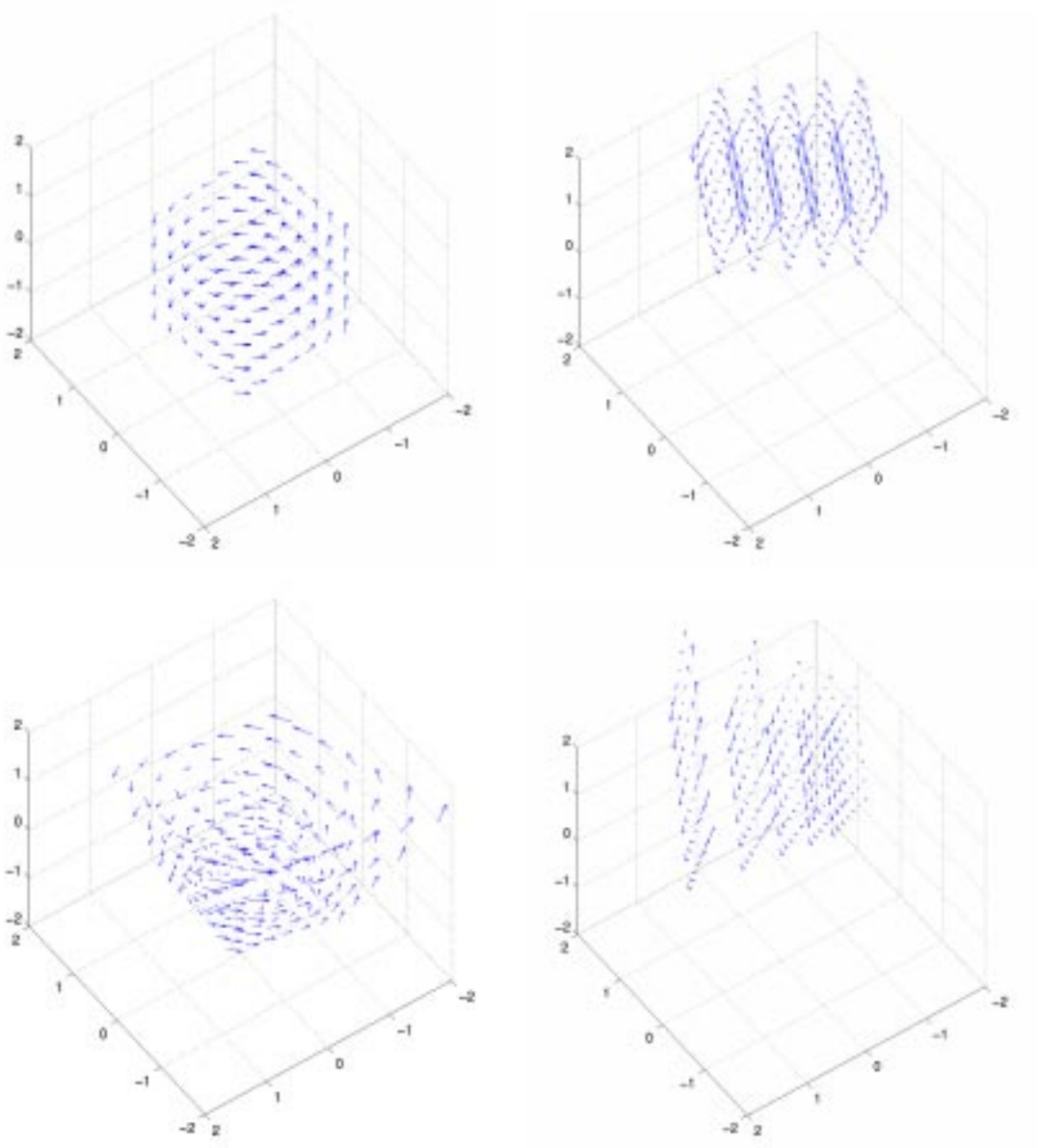

Figure 4.4: Vector field generated by an operator matrix $\hat{\mathbf{H}}_R$ seen in various positions and ambient spaces: z-axis (left), general axis (right). The Euclidean space (top), and projective space (bottom), which is plotted as if it were Euclidean, so that the projective skew is apparent in the vectors being misaligned with the tangents, and in their arbitrary lengths (scales).

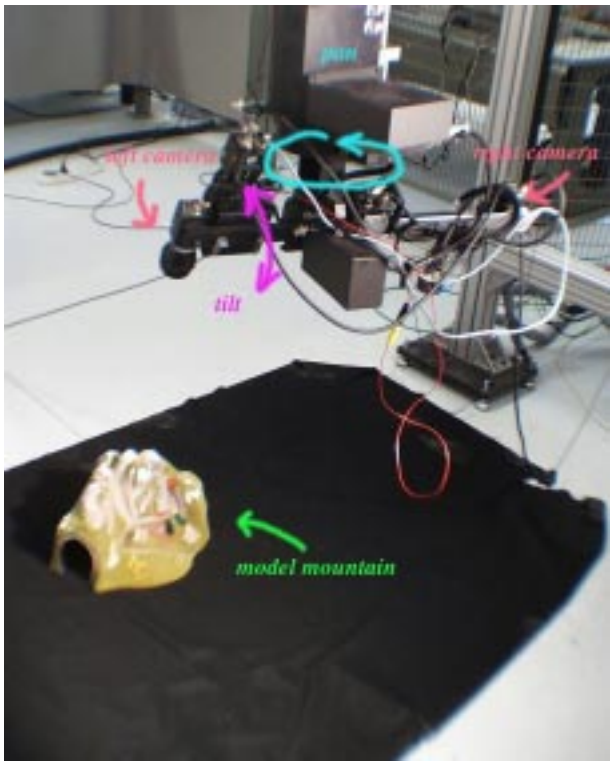

Figure 4.5: Photograph of experimental setup, showing pan-tilt mounted stereo camera observing a model-mountain.

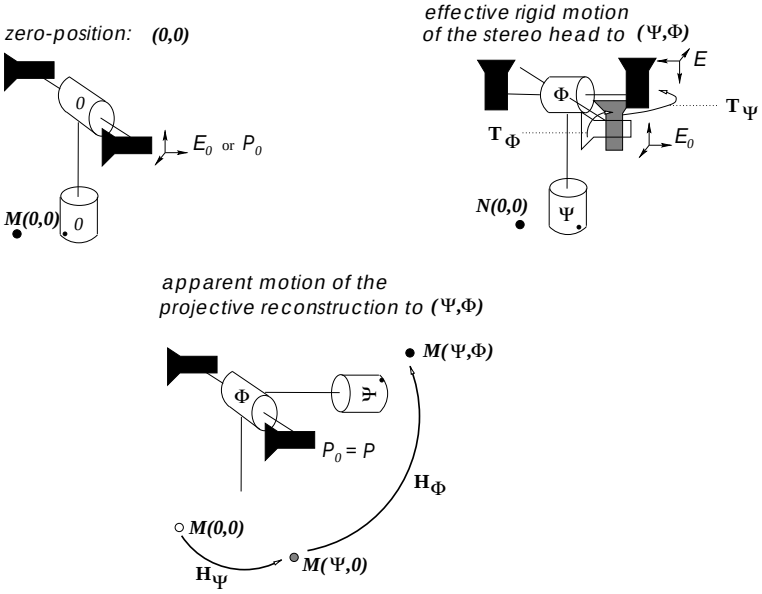

Figure 4.6: Sketch of pan-tilt mechanism, showing camera motion and apparent scene motion.

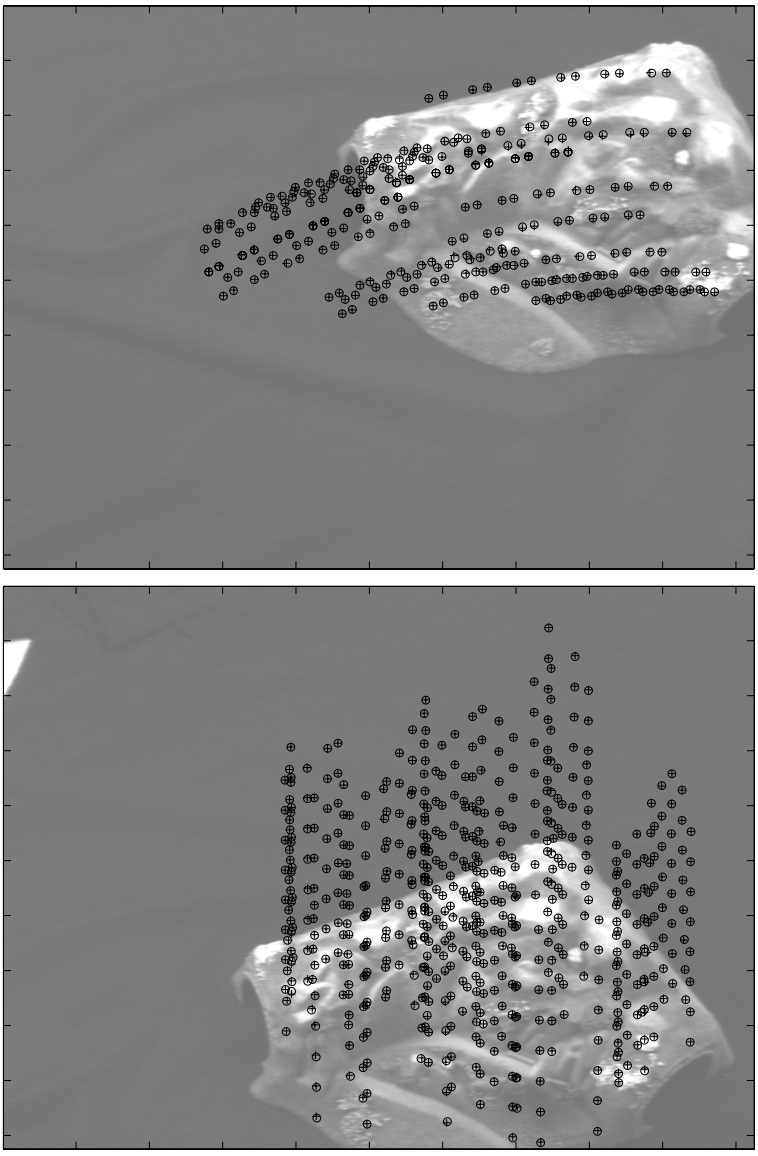

Figure 4.7: Projective Calibration of pan-(top) and tilt-joint(bottom):
The images show the mountain model and superimposed point trajectories of pan- and tilt-motion, respectively. The missing steps are those used to calibrate. The + marks the backprojected points, the *o* marks the actual extracted points.

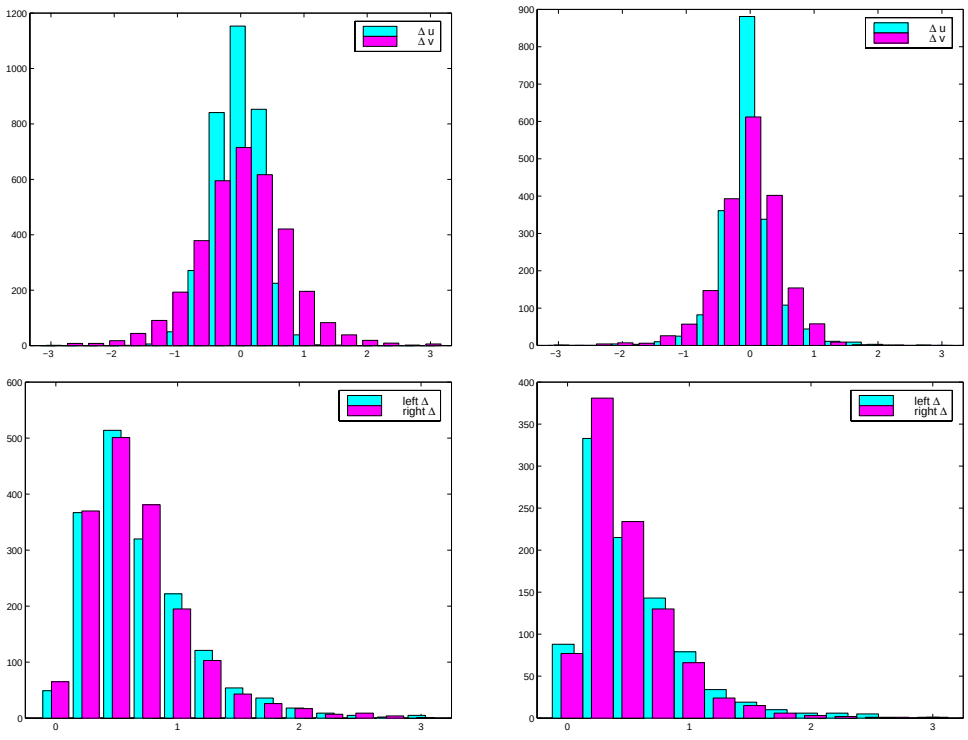

Figure 4.8: Calibration of pan- (left) and tilt-joint (right):
(top) Histogram of calibration errors Δ_u, Δ_v. They seem to be normally distributed with zero mean, which indicates that the estimation is unbiased. (bottom) Histogram of image distances Δ_L, Δ_R. Left and right images show equal error distributions which accord with an assumed additive Gaussian noise.

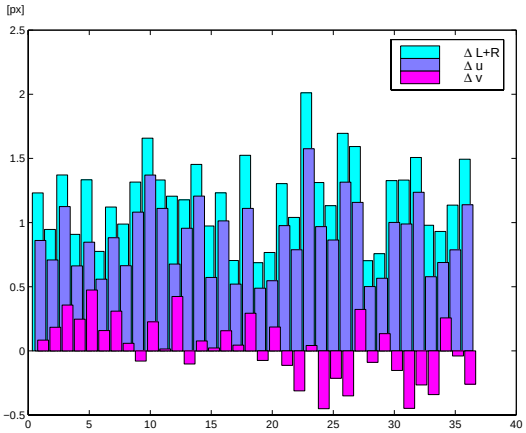

Figure 4.9: Feed-forward prediction over 35 trials:
The mean differences Δ_u, Δ_v indicates the lateral offset, whereas Δ_{LR} gives the mean distance between predicted and extracted points.

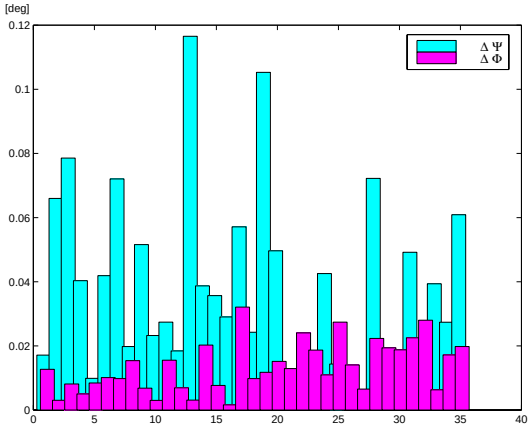

Figure 4.10: Visual homing using a single point:
Joint-space error over 35 trials.

Chapter 5

Projective Kinematics

This chapter is devoted to the theory and practice of articulated mechanisms, bodies, and articulated motion in general, but especially to projective representations of these. It develops in detail a product-of-exponentials representation for chains of lower-order pairs, and for their corresponding Jacobians. It is based on projective matrix representations of the underlying Lie groups and Lie algebras. Clearly, the importance of articulated motions and mechanisms lies particularly in the modeling and operation of robot manipulators.

The contribution here is the introduction of a methodology for modeling manipulators using projective transformations. It allows robot to be calibrated and controlled using an "uncalibrated" stereo camera, i.e. a triangulation device in projective space alone. As opposed to former approaches, a-priori metric knowledge of the camera calibration or of the scene structure are no longer required. Moreover, the visual servoing laws are formulated directly in the robot joint-space, and metric kinematic models as well as Cartesian robot controllers are no longer required.

In the context of this thesis, the introduction of projective kinematics gave rise to various practical methods, one for self-calibration of a six-axis robot from uncalibrated stereo images (section 5.4), one for non-metric image-based visual servoing (section 8.4), and one for trajectory-based direct joint-space control (section 7.3, section 8.5).

5.1 Kinematics of articulated mechanisms

Consider a serial robot manipulator with n joints. Generally speaking, this is an articulated mechanism consisting of a serial linkage of n lower-order kinematic pairs.

The "*configuration*" of the mechanism is represented by a n-vector $\boldsymbol{q}$ of joint variables q_i which stand for either the "*joint angle*" θ_j of a revolute joint, or the "*extension*" τ_j of a prismatic joint. The vectors $\boldsymbol{q} = (q_1, \ldots, q_n)^\top$

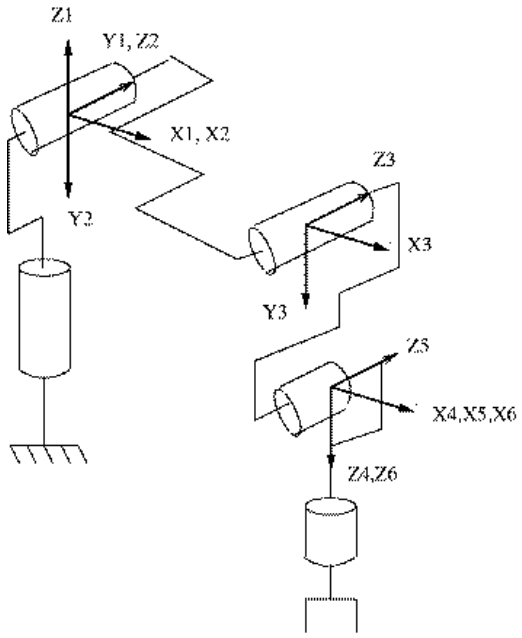

Figure 5.1: Schematic description of the kinematics of a robot manipulator.

form a compact joint-space bounded by the robot joint-limits, with origin $\boldsymbol{q} = \mathbf{o}$, called the "*zero*"(-configuration) or "*zero-reference*".

For us, the "*posture*" of the articulated chain is the posture of its hand with respect to its base, represented by the displacement between a Cartesian frame $\mathcal{H}$ on the tip-limb and a frame $\mathcal{B}$ on the base-limb. Such a mapping is called the "*forward kinematics*"

$$\boldsymbol{q} \rightarrow \mathbf{T}_{BH}(\boldsymbol{q})$$

Note that these frames can be chosen arbitrarily, so long as they are rigidly linked to the tip or the base. A common choice however is to allocate such frames according to the Denavit-Hartenberg conventions.

Denavit-Hartenberg kinematic model

The "*DH-conventions*" [16] are a standard method for almost[1] uniquely allocating frames on the elements of the chain, that allow the linkages to be parameterized minimally. This is done such that the geometry of the j^{th} element is represented by a "*link-transform*" $\mathbf{L}_j$ and a "*joint-transform*" $\mathbf{J}_j$. The so-called "*Denavit-Hartenberg*" parameters of an element are:

- α_j (link angle)

[1] ... Ambiguities arise in case of parallel axes or several prismatic joints.

- r_j (link offset)
- l_j (link length)

The product of the link- and joint-transforms is the overall *"joint action"* $\mathbf{Q}_j$ for a revolute joint subscripted by $_R$, and a prismatic one subscripted by $_P$.

$$\mathbf{Q}_{jR}(q_j = \theta_j) = \mathbf{L}_j \cdot \mathbf{J}_{jR}(\theta_j) = \begin{bmatrix} 1 & 0 & 0 & r_j \\ 0 & \cos\alpha_j & -\sin\alpha_j & 0 \\ 0 & \sin\alpha_j & \cos\alpha_j & l_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_j & -\sin\theta_j & 0 & 0 \\ \sin\theta_j & \cos\theta_j & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.1)$$

$$\mathbf{Q}_{jT}(q_j = \tau_j) = \mathbf{L}_j \cdot \mathbf{J}_{jT}(\tau_j) = \begin{bmatrix} 1 & 0 & 0 & r_j \\ 0 & \cos\alpha_j & -\sin\alpha_j & 0 \\ 0 & \sin\alpha_j & \cos\alpha_j & l_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau_j \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.2)$$

Given this, the posture $\mathbf{T}_{BH}$ regarded as a serial linkage between hand and base becomes the product of the transformations $\boldsymbol{Q}_j(q_j)$ between adjacent elements. This product is an example for the *"forward kinematic map"* of the manipulator:

$$\mathbf{T}_{BH}(\boldsymbol{q}) = \mathbf{Q}_1(q_1) \cdot \mathbf{Q}_2(q_2) \cdots \mathbf{Q}_n(q_n). \quad (5.3)$$

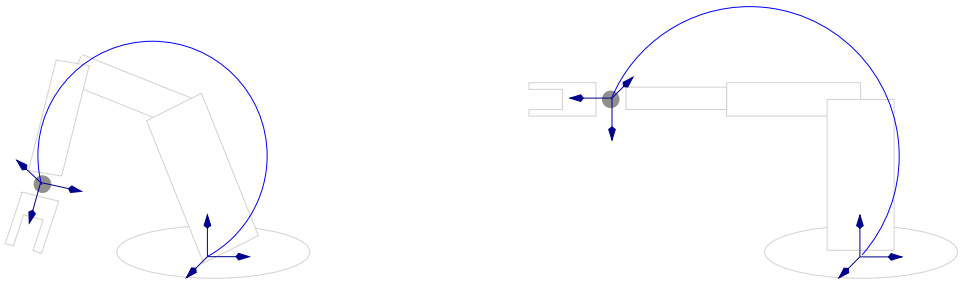

Figure 5.2: The kinematic mapping formulated for two frames, e.g. hand and base frame.

Two-frame relative kinematic model

As soon as one allows the hand or base frames to deviate from the DH-conventions, various alternative formulations of the kinematics are possible. They differ essentially in the frames that they connect, or more abstractly speaking, in which they are represented.

Firstly, replace the hand frame with a frame $\mathcal{O}$ allocated on a tool mounted on the hand or an object grasped by it.

Secondly, replace the base frame with a reference frame $\mathcal{C}$ rigidly installed in workspace. This could be e.g. the camera frame associated with an independently fixed stereo camera.

$$\mathbf{T}_{CO}(\boldsymbol{q}) = \mathbf{T}_{CB} \cdot \mathbf{Q}_1(q_1) \cdot \mathbf{Q}_2(q_2) \cdots \mathbf{Q}_n(q_n) \cdot \mathbf{T}_{HO}. \quad (5.4)$$

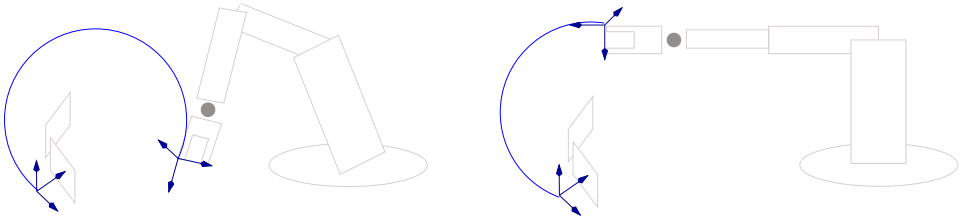

Figure 5.3: The kinematic mapping formulated for two frames, e.g. object and camera frame.

Single-frame relative kinematic model

Thirdly, and more confusingly, the same frame can be used for both, the base and the hand frames. Actually, it is one frame and a second coincident frame which are "virtually " linked to the base and the hand frames. In other words, changes in the robot posture, which are rigid motions, are expressed with respect to an externally defined reference frame, which may be linked to either the hand or the base, or to neither of them, as the application requires. In the present case, the frame will often be linked to the camera, wherever it may be mounted.

While one frame is considered as immobile (base), the other frame is considered as mobile (hand), indicated by a prime $'$. In consequence, the articulated motion can now be represented as the "*relative motion*" of an arbitrarily chosen frame. Two natural examples for such a single reference are the tool, or the camera frame. This results in the robot posture to represented by the tool motion $\mathbf{T}_O$ or by the apparent camera motion $\mathbf{T}_C^{-1}$.

$$\mathbf{T}_O(\boldsymbol{q}) = \mathbf{T}_{OB} \cdot \mathbf{Q}_1(q_1) \cdot \mathbf{Q}_2(q_2) \cdots \mathbf{Q}_n(q_n) \cdot \mathbf{T}_{HO'}, \tag{5.5}$$

$$\mathbf{T}_C(\boldsymbol{q}) = \mathbf{T}_{CB} \cdot \mathbf{Q}_1(q_1) \cdot \mathbf{Q}_2(q_2) \cdots \mathbf{Q}_n(q_n) \cdot \mathbf{T}_{HC'}. \tag{5.6}$$

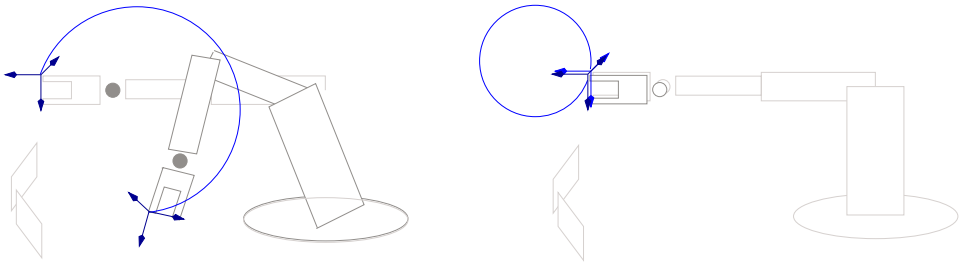

Figure 5.4: The kinematic mapping formulated for a single frame, e.g. the object-frame. Zero-reference and identical object-motion (right). Robot-motion and corresponding object-motion (left).

Here, $\mathbf{T}_{OB}$ stands for the virtual link "tool-base"and $\mathbf{T}_{HC'}$ stands for the virtual link "hand-camera", both established or "frozen" when the robot

was at zero. It is important to realize that these virtual links represent a certain posture of the robot. Since for this posture, the relative motion is vanishing, it is worthwhile to shift the origin of the joint-space to $\boldsymbol{q}-\boldsymbol{Q}=\mathbf{0}$, such that a null-motion in joint-space corresponds to the identical relative posture $\mathbf{T}_O(\mathbf{0})=\mathbf{T}_C(\mathbf{0})=\mathbf{I}$.

Then, the posture of the robot at zero equals:

$$\mathbf{T}_{OB}=\mathbf{T}_{HO'}^{-1}\cdot\mathbf{Q}_n^{-1}(q_n)\cdots\mathbf{Q}_1^{-1}(q_1), \tag{5.7}$$

$$\mathbf{T}_{HC'}=\mathbf{Q}_n^{-1}(q_n)\cdots\mathbf{Q}_1^{-1}(q_1)\cdot\mathbf{T}_{CB}^{-1}. \tag{5.8}$$

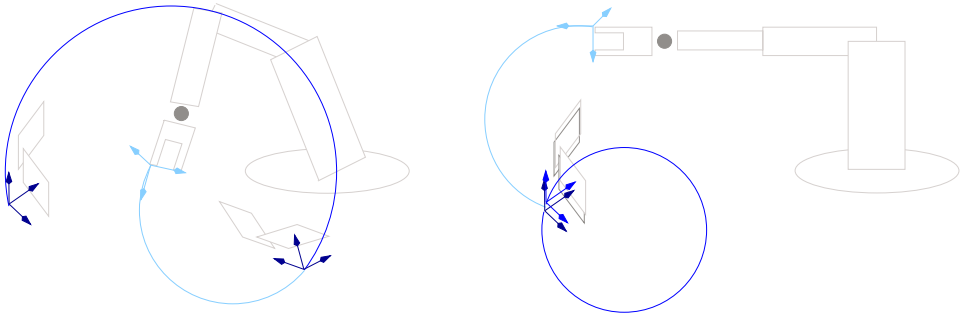

Figure 5.5: The kinematic mapping formulated for a single frame, e.g. the camera frame. Zero-reference and identical object-motion (right). Robot-motion and corresponding object-motion (left).

Zero-reference kinematic model

To prepare the ground for more general geometric models of an articulated mechanism (section 5.2), it is worthwhile omitting all intermediate frames on the elements, since they are somehow irrelevant, and to replace them with a single reference frame. For that purpose, consider separate motions of each of the mechanism's joints, and denote such elementary joint-space motions as

$$\boldsymbol{q}_j=(0,\ldots 0,q_j,0,\ldots,0). \tag{5.9}$$

Intuitively, it is clear that the relative motion arising from an elementary motion is either a pure rotation $\mathbf{T}_{jR}(\theta_j)=\mathbf{T}(\boldsymbol{q}_j)$, or a pure translation $\mathbf{T}_{jT}(\tau_j)=\mathbf{T}(\boldsymbol{q}_j)$, depending on the type of joint j. This continues to hold in the reference frame. So, denote the pure motions of each joint as

$$\mathbf{T}_j(q_j)=\mathbf{T}(\boldsymbol{q}_j). \tag{5.10}$$

where the subscript of the reference frame is omitted as the particular frame used is irrelevant.

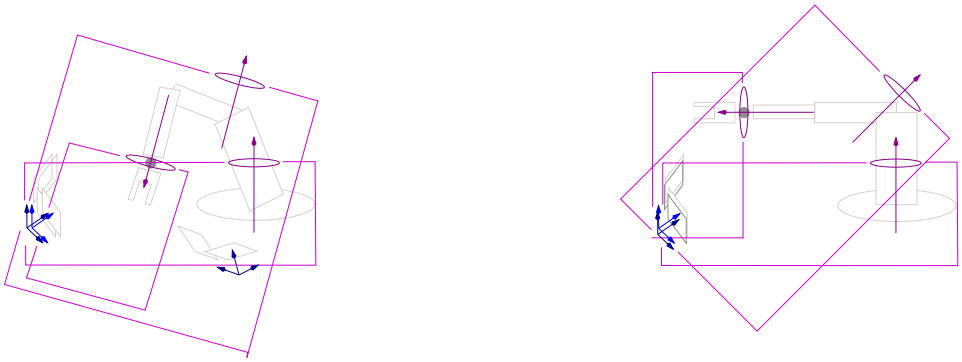

Figure 5.6: The kinematic mapping formulated as zero-reference model, here in the camera frame. The closed rectangles are to illustrate the action of a joint onto the camera-frame: $\mathbf{T}_j(q_j)$.

Formally, (5.5) and (5.10) allow this to be shown as follows:

$$\begin{aligned}
&\mathbf{T}(\boldsymbol{q}_j) = \\
&= \mathbf{T}_{OB} \cdot \mathbf{Q}_1(0) \cdots \mathbf{Q}_{j-1}(0) \cdot \mathbf{Q}_j(q_j) \cdot \mathbf{Q}_{j+1}(0) \cdots \mathbf{Q}_n(0) \cdot \mathbf{T}_{HO'} && (5.11)\\
&= \mathbf{T}_{HO'}^{-1} \cdot \mathbf{Q}_n^{-1}(0) \cdots \mathbf{Q}_{j+1}^{-1}(0) \cdot \mathbf{Q}_j^{-1}(0) \mathbf{Q}_j(q_j) \cdot \mathbf{Q}_{j+1}(0) \cdots \mathbf{Q}_n(0) \cdot \mathbf{T}_{HO'} \\
&= \mathbf{S}^{-1} \cdot \mathbf{Q}_j^{-1}(0) \mathbf{Q}_j(q_j) \cdot \mathbf{S} = \mathbf{S}^{-1} \cdot \mathbf{L}_j^{-1} \mathbf{J}_j^{-1}(0) \cdot \mathbf{J}_j(q_j) \mathbf{L}_j \cdot \mathbf{S} \\
&= \left(\mathbf{S}^{-1} \mathbf{L}_j^{-1} \right) \cdot \mathbf{J}_j(q_j) \cdot (\mathbf{L}_j \mathbf{S}) \,, && (5.12)\\
&\text{where } \mathbf{S} \text{ abbreviates a respective conjugate form.} && (5.13)
\end{aligned}$$

In this expression for $\mathbf{T}(q_j)$, the index referring to frame $\mathcal{O}$ has intentionally been omitted as all such frames are equivalent, i.e. the formulation is "*coordinate-free*". To help intuition however, one can just think of the relative tool-motion $\mathbf{T}_O$.

The fact that each joint-action is expressed relative to one and the same reference frame has the consequence that the transforms $\mathbf{L}_i$, previously required to represent the links, become the identity. The serial linkage is thus expressed directly as a simple product of joint-actions, where the geometry of the eliminated links is captured now by the joint-actions $\mathbf{T}_j(q_j)$:

$$\mathbf{T}(\boldsymbol{q}) = \mathbf{T}_1(q_1) \cdots \mathbf{T}_n(q_n). \tag{5.14}$$

Intuitively, multiplication from right-to-left in tip-to-base order can be interpreted as actuating the joints in just this order. This means that each joint spatially occupies before being operated the position that it occupies for the robot at zero. For this reason, the formulation (5.14) of the forward kinematics is valid in accordance with the present definition of the elementary motions. The formal validation is also straight-forward, it suffices to substitute (5.11) and to simplify.

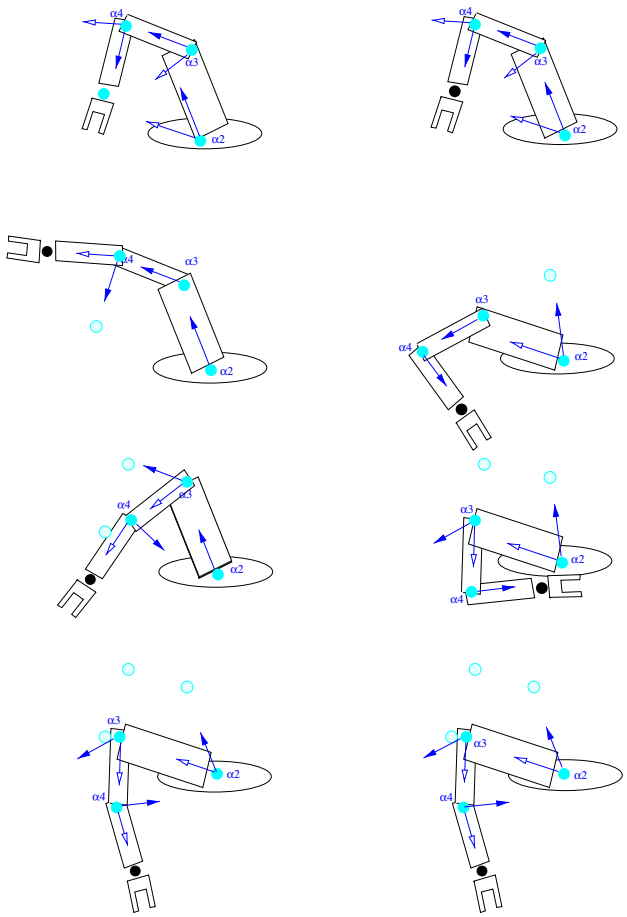

Figure 5.7: Actuation of joints in order "tip-to-base" (left) or "base-to-tip" (right). The first assures each joint to be actuated in its initial position, whereas the second causes a displacement of the joints before its actuation. This illustrates that a zero-reference model, which encodes fix 3D positions of all joint-axis, is a valide model if the product-of-exponentials is calculates in tip-to-base order.

5.2 Projective Kinematics

The idea of projective kinematics is to by-pass both, the Euclidean modeling of the camera system and the Cartesian modeling of the robot system, by replacing and unifying them with a projective model. Both models refer to a single projective ambient space, which is in fact an abstraction of stereo image-space. This allows for a direct relationship between joint-space and image-space to be established.

Classical kinematic modeling is based on Euclidean geometry, so it makes heavy use of notions like lengths, angles, perpendicularity. Hence, at first glance, the generalization of these models to projective space seems rather implausible. For example, try to think of a projective equivalent of the Denavit-Hartenberg conventions: How to allocate a projective frame and in particular the fifth point on the axis of a revolute joint, without a normal to the axis being well-defined.

5.2.1 Projective kinematics: zero-reference

Consequently, the projective formulation has to be based on a frame which "naturally" or "by definition" allows the projective and Euclidean representations to be related. In this sense, the camera frame, and especially a kinematic model based in this frame, is an appropriate formalism to investigate. Certainly, this is a zero-reference model formulated in the camera frame, as already described for the Euclidean case.

Theorem 9 (projective forward kinematic map)
Under the rigid-stereo assumption (2.18), an articulated mechanism has a projective model consisting of operators $\hat{\mathbf{H}}_j$, each of which represents a joint j. They relate to the zero of joint-space $\boldsymbol{q} = 0$, and describe there the projective motion $\mathbf{H}_j = \exp(q_j \hat{\mathbf{H}}_j)$ of a single joint $\boldsymbol{q}_j$. The projective motion $\mathbf{H}(\boldsymbol{q})$ corresponding to a general joint-space motion $\boldsymbol{q}$ results from these as the product-of-exponentials in the operators (5.18).

Proof:
Starting from (5.14), the projective displacement (5.15) corresponding to $\mathbf{T}(\boldsymbol{q})$ is $\mathbf{H}_{PE}^{-1}\mathbf{T}(\boldsymbol{q})\mathbf{H}_{PE}$ (2.33). Additional terms $\mathbf{H}_{PE}\mathbf{H}_{PE}^{-1} = \mathbf{I}$ can be inserted in (5.16) without the motion itself being affected. So, the product indeed consists of projective joint-actions $\boldsymbol{H}_j(q_j)$ (5.17), only, which are either projective rotations $\mathbf{H}_j(q_j) = \mathbf{H}_{jR}(\theta_j)$ or projective translations $\mathbf{H}_j(q_j) = \mathbf{H}_{jT}(\tau_j)$.

$$\begin{aligned}
\mathbf{H}(\boldsymbol{q}) &= \mathbf{H}_{PE}^{-1}\mathbf{T}_1(q_1) \quad \cdots \quad \mathbf{T}_n(q_n)\mathbf{H}_{PE} && (5.15)\\
&= \mathbf{H}_{PE}^{-1}\mathbf{T}_1(q_1)\ \mathbf{H}_{PE}\ \mathbf{H}_{PE}^{-1}\ \cdots\ \mathbf{H}_{PE}\ \mathbf{H}_{PE}^{-1}\mathbf{T}_n(q_n)\mathbf{H}_{PE} && (5.16)\\
&= \mathbf{H}_1(q_1) \quad \cdots \quad \mathbf{H}_n(q_n). && (5.17)
\end{aligned}$$

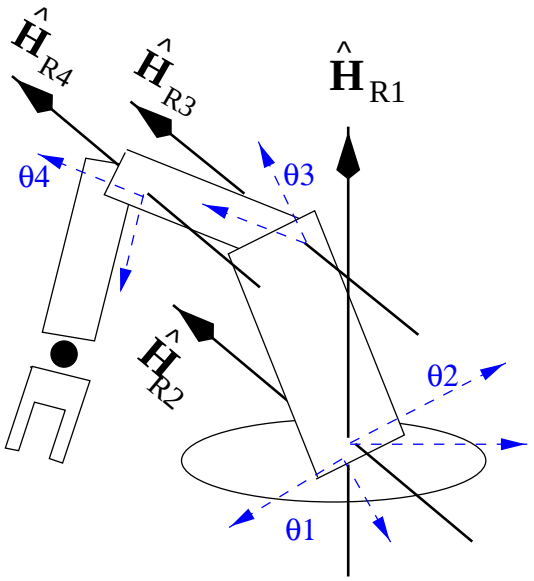

Figure 5.8: The kinematics of the manipulator is represented by a series of projective rotations, i.e. their generators, associated with the revolute joints, and their position at zero.

As soon as the exponential forms (4.27) or (3.21) are substituted, a product-of-exponentials in projective form is obtained. In the most generic formulation, it looks like (5.18), where $\hat{\mathbf{H}}_j$ are generic operators without the type of joints being made explicit.

$$\mathbf{H}(\boldsymbol{q}) = \exp(q_1\hat{\mathbf{H}}_1)\cdots\exp(q_n\hat{\mathbf{H}}_n). \tag{5.18}$$

The motion of a single joint $\boldsymbol{q}_j$ amounts to the one-parameter groups investigated in chapters 3 and 4.

$$\begin{aligned}\mathbf{H}(\boldsymbol{q}_j) &= \exp(0\hat{\mathbf{H}}_1)\cdots\exp(0\hat{\mathbf{H}}_{j-1})\cdot\exp(q_j\hat{\mathbf{H}}_j)\cdot\exp(0\hat{\mathbf{H}}_{j+1})\cdots\exp(0\hat{\mathbf{H}}_n),\\ \mathbf{H}(\boldsymbol{q}_j) &= \exp(q_j\hat{\mathbf{H}}_j) = \mathbf{H}_j(q_j).\end{aligned} \tag{5.19}$$

Alternatively, a proof is given now, which is entirely based on the concept of projective motion, and which is free of any explicit reference to an underlying Cartesian model, i.e. "*calibration-free*". The basic argument is an induction, based on a "truncated" kinematic model $\bar{\mathbf{H}}_j(\boldsymbol{q})$, representing the chain up to joint j:

$$\bar{\mathbf{H}}_j(\boldsymbol{q}) = \mathbf{H}_1(q_1)\cdots\mathbf{H}_j(q_j) \tag{5.20}$$

Proof:
Points $\boldsymbol{M}_1$ in rigid linkage with the 1^{st} element are affected only by the

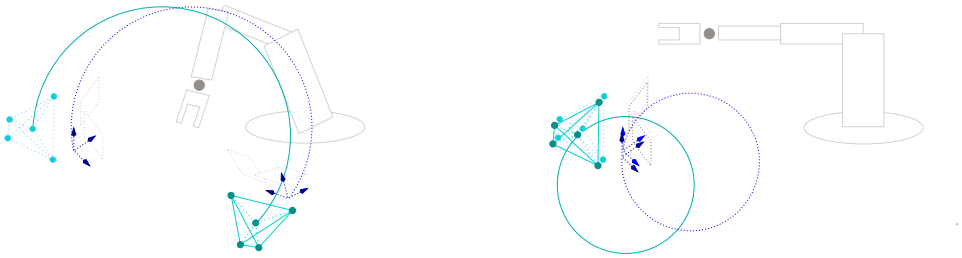

Figure 5.9: The projective kinematic mapping in a single projective frame, where the tetrahedron illustrates the projective camera frame associated with the stereo camera. Zero-reference and identical camera-motion (right). Robot-motion and apparent camera-motion, both projective (left).

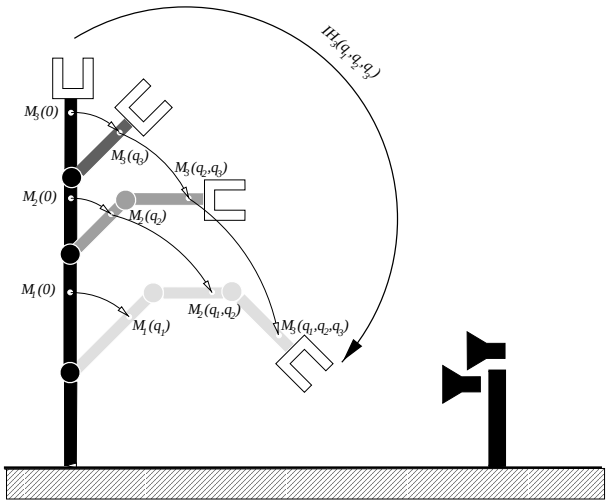

Figure 5.10: Proof by induction over the number of joints in the chain.

motion of the 1^{st} joint: $\boldsymbol{M}_1(\boldsymbol{q}) = \mathbf{H}_1(q_1)\boldsymbol{M}_1$. Thus, the hypothesis is that the elements preceding the j^{th} joint are moving like (5.20). Now, points $\boldsymbol{M}_j$ associated with the j^{th} element are moving like $\boldsymbol{M}_j(q_j) = \mathbf{H}_j(q_j)\boldsymbol{M}_j$ if only the j^{th} joint is actuated. As soon as the j^{th} joint has been locked to q_j, the points $\boldsymbol{M}_j(q_j)$ are in effect rigidly linked with the element $j-1$. In conclusion, the induction hypothesis for $j-1$ can be applied, which shows that elementary joint motions have to be left-multiplied in tip-to-base order to obtain (5.17). q.e.d.

5.2.2 Projective kinematics: general reference

It is worthwhile to investigate formulations of the forward kinematics for arbitrary choices of the joint-space origin, and to describe how they relate to one another. Actually, this amounts to updating the forward kinematics

during an articulated motion in order to account for the current configuration of the robot having varied. Similar calculations are done in the control loops of a robot's Cartesian velocity mode.

As the projective kinematic model is less intuitive, and because it has some peculiarities, the projective update formalism is described in detail below. It is essential for the derivations in later chapters, that for any configuration of a robot – not just for the zero – a sound analytic expression for the projective kinematics is available.

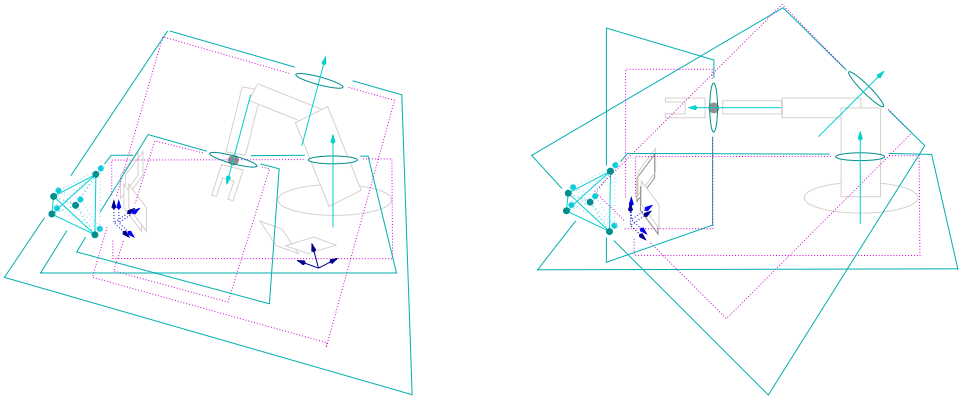

Figure 5.11: The projective kinematic mapping generalized around an arbitrary zero of joint-space. The skewed rectangles illustrate that joint-action are now projective transformations: $\hat{\mathbf{H}}_j$. They depend on the current configuration $\boldsymbol{Q}$ as soon as the generalized model is considered: $\hat{\mathbf{H}}_j s\boldsymbol{Q}$ (left).

Theorem 10 (reconfiguration of projective model)
Given a projective kinematic model consisting of operators $\hat{\mathbf{H}}_j$ which represent the joints of the mechanism at an initial configuration $\boldsymbol{q} = \mathbf{o}$, the zero, a reconfiguration of the mechanism to $\boldsymbol{Q}$ implies a displacement of the joint j by $\bar{\mathbf{H}}_j(Q)$ (5.20), and hence requires the corresponding operator to be updated to (5.21). The operators $\hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}}$, expressed as a function of the current configuration $\boldsymbol{Q} = \boldsymbol{q}(t)$ form a general model of the projective kinematics, that is equivalent to the original one. In the shifted joint-space the model has the form (5.22):

$$\hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}} = \bar{\mathbf{H}}_j(\boldsymbol{Q}) \cdot \hat{\mathbf{H}}_j \cdot \bar{\mathbf{H}}_j^{-1}(\boldsymbol{Q}), \tag{5.21}$$

$$\mathbf{H}(\boldsymbol{q} - \boldsymbol{Q}) = \exp\left((\boldsymbol{q}_1 - \boldsymbol{Q}_1)\hat{\mathbf{H}}_1\big|_{\boldsymbol{Q}}\right) \cdots \exp\left((\boldsymbol{q}_n - \boldsymbol{Q}_n)\hat{\mathbf{H}}_n\big|_{\boldsymbol{Q}}\right). \tag{5.22}$$

Proof:
For the robot in a general configuration $\boldsymbol{Q}$, shifting the origin of joint-space to $\boldsymbol{Q}$ gives

$$\boldsymbol{q}' = \boldsymbol{q} - \boldsymbol{Q}, \tag{5.23}$$

where $\boldsymbol{q}' = (q'_1, \ldots, q'_n)^\top$ denotes a configuration in the shifted space. Let's define $\mathbf{H}'$ so that a motion in the shifted joint-space and the original one multiply as follows

$$\mathbf{H}(\boldsymbol{q}) = \mathbf{H}'(\boldsymbol{q} - \boldsymbol{Q})\mathbf{H}(\boldsymbol{Q}) = \mathbf{H}'(\boldsymbol{q}')\mathbf{H}(\boldsymbol{Q}). \tag{5.24}$$

Using the matrix equality (4.27) for the exponential of a similarity, the original operators can be introduced as detailed in (5.25), where $\hat{\mathbf{H}}'_j$ is an abbreviation for $\hat{\mathbf{H}}'_j\big|_{\boldsymbol{Q}}$.

$$\exp(q'_j\hat{\mathbf{H}}'_j) = \bar{\mathbf{H}}_j(\boldsymbol{Q})\exp(q'_j\hat{\mathbf{H}}_j)\bar{\mathbf{H}}_j^{-1}(\boldsymbol{Q}) \tag{5.25}$$

In the product (5.26), the similarity transforms $\bar{\mathbf{H}}_j$ in two subsequent factors cancel out, as shown in (5.28), leaving the shift $\mathbf{H}_{j+1}(Q_{j+1})$ appearing in (5.27). The original joint-actions $\mathbf{H}_{j+1}(\boldsymbol{q}_{j+1})$ can then be substituted into (5.28), having (5.20) in mind.

$$\bar{\mathbf{H}}_j(\boldsymbol{Q})\exp(q'_j\hat{\mathbf{H}}_j)\bar{\mathbf{H}}_j^{-1}(\boldsymbol{Q}) \cdot \bar{\mathbf{H}}_{j+1}(\boldsymbol{Q})\exp(q'_{j+1}\hat{\mathbf{H}}_{j+1})\bar{\mathbf{H}}_{j+1}^{-1}(\boldsymbol{Q}), \tag{5.26}$$

$$\bar{\mathbf{H}}_j(\boldsymbol{Q})\exp(q'_j\hat{\mathbf{H}}_j)\mathbf{H}_{j+1}(Q_{j+1})\exp(q'_{j+1}\hat{\mathbf{H}}_{j+1})\bar{\mathbf{H}}_{j+1}^{-1}(\boldsymbol{Q}), \tag{5.27}$$

$$\mathbf{H}_j(Q_j)\exp(q'_j\hat{\mathbf{H}}_j) = \exp(Q_j\hat{\mathbf{H}}_j)\exp(q'_j\hat{\mathbf{H}}_j) = \exp(q_j\hat{\mathbf{H}}_j) \tag{5.28}$$

Although the first (5.29) and last element (5.30) have to be treated separately, they equally cancel out.

$$\bar{\mathbf{H}}_1(\boldsymbol{Q})\exp(q'_1\hat{\mathbf{H}}_1) = \mathbf{H}_1(\boldsymbol{Q})\exp(q'_1\hat{\mathbf{H}}_1), \tag{5.29}$$

$$\bar{\mathbf{H}}_n^{-1}(\boldsymbol{Q})\mathbf{H}(\boldsymbol{Q}) = \mathbf{I}. \text{ q.e.d.} \tag{5.30}$$

When putting all this together, the shifted model has been shown to correspond exactly to the original one in (5.24)

Intuitively, the equation (5.21) can be understood as follows: For each joint j, first its displacement $\bar{\mathbf{H}}_j^{-1}(\boldsymbol{Q})$ in the chain must be undone, using the partial forward kinematics (5.20). Then, the original operator $\hat{\mathbf{H}}_j$ can be applied as before. After that, applying $\bar{\mathbf{H}}_j(\boldsymbol{Q})$ returns the displacement to $\boldsymbol{Q}$. More abstractly, applying the adjoint maps of $\bar{\mathbf{H}}_j(\boldsymbol{Q})$ resets each operator to the reference frame it was originally expressed in, which accounts for the motion the corresponding joint has undergone.

5.3 Projective Jacobian

Now, the relationship between joint-space velocities $\dot{\boldsymbol{q}}$ and velocities $\dot{\mathbf{H}}(\boldsymbol{q})$ of the chain's elements is examined. Formally, the results will be 4×4 matrices lying in the tangent space of the projective displacements. The geometric interpretation is as velocity-screws of rigid bodies, here the screws of each

limb element. They can be applied to a rigid point-structure in order to derive point-velocities. Below, two versions of the Jacobian are derived, the differences between them are pointed-out, and their formal equivalence is shown.

5.3.1 Projective Jacobian: zero-reference

Theorem 11
Assume an articulated mechanism and its projective kinematic geometry given in terms of operators $\hat{\mathbf{H}}_j$, representing the action of joint j for the configuration $\boldsymbol{q} = 0$, the zero-reference. Consider further a rigid structure of points linked with the chain's tip element. Let us denote one of these points as $\boldsymbol{M}$, expressed in the zero-reference coordinate system. Then, for the mechanism in configuration $\boldsymbol{q}$, an articulated motion with velocity $\dot{\boldsymbol{q}}$ causes a corresponding projective motion with tangent

$$\dot{\mathbf{H}} = \sum_{j=1}^{n} \dot{\boldsymbol{q}}_j \Big(\mathbf{H}_1(q_1) \cdots \hat{\mathbf{H}}_j \mathbf{H}_j(q_j) \cdots \mathbf{H}_n(q_n) \Big), \tag{5.31}$$

and a point $\boldsymbol{M}$ initially on the terminal element to move with velocity

$$\dot{\boldsymbol{M}} = \dot{\mathbf{H}} \boldsymbol{M}. \tag{5.32}$$

The spatial and body velocity of this element are

$$\hat{\mathbf{H}} = \sum_{j=1}^{n} \dot{q}_j \cdot \Big(\bar{\mathbf{H}}_{j-1}(\boldsymbol{q}) \; \hat{\mathbf{H}}_j \; \bar{\mathbf{H}}_{j-1}^{-1}(\boldsymbol{q}) \Big) \tag{5.33}$$

$$\check{\mathbf{H}} = \sum_{j=1}^{n} \dot{q}_j \cdot \Big(\mathbf{H}^{-1}(\boldsymbol{q}) \; \hat{\mathbf{H}}_j \; \mathbf{H}(\boldsymbol{q}) \Big). \tag{5.34}$$

Proof:
Consider a time instant t, at which the chain instantaneously has the configuration $\boldsymbol{q} = \boldsymbol{q}(t)$ and the joint-space velocity $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}(t)$. The partial derivatives of (5.17) allow the tangent to the motion to be written as a function of $\dot{\boldsymbol{q}}$.

$$\dot{\mathbf{H}}(\boldsymbol{q}) = \frac{d\mathbf{H}(\boldsymbol{q})}{d\boldsymbol{q}} \;\; \dot{\boldsymbol{q}} = \sum_{j=1}^{n} \dot{q}_j \;\; \frac{\partial \mathbf{H}(\boldsymbol{q})}{\partial q_j}, \quad \text{where} \tag{5.35}$$

$$\frac{\partial \mathbf{H}}{\partial q_j} = \mathbf{H}_1(q_1) \cdots \hat{\mathbf{H}}_j \exp(q_j \hat{\mathbf{H}}_j) \cdots \mathbf{H}_n(q_n). \tag{5.36}$$

For $\boldsymbol{M}$ on the tip element, its projective point-velocity is then

$$\dot{\boldsymbol{M}} = \dot{\mathbf{H}}(\boldsymbol{q}) \boldsymbol{M} = \sum_{j=1}^{n} \dot{q}_j \frac{\partial \mathbf{H}(\boldsymbol{q})}{\partial q_j} \boldsymbol{M}. \tag{5.37}$$

This can be written in terms of a $4 \times n$ matrix $\mathbf{J}(\boldsymbol{q}, \boldsymbol{M})$, the projective Jacobian for this point.

$$\mathbf{J}(\boldsymbol{q}, \boldsymbol{M}) = \left[\frac{\partial \mathbf{H}(\boldsymbol{q})}{\partial q_1} \boldsymbol{M} \middle| \quad \dots \quad \middle| \frac{\partial \mathbf{H}(\boldsymbol{q})}{\partial q_n} \boldsymbol{M} \right], \tag{5.38}$$

$$\dot{\boldsymbol{M}} = \mathbf{J}(\boldsymbol{q}, \boldsymbol{M}) \quad \dot{\boldsymbol{q}}. \tag{5.39}$$

The spatial velocity $\hat{\mathbf{H}}$ (2.42) of the tip element is

$$\begin{aligned} \hat{\mathbf{H}} &= \dot{\mathbf{H}}(\boldsymbol{q}) \cdot \mathbf{H}^{-1}(\boldsymbol{q}) = \\ &= \sum_{j=1}^{n} \dot{q}_j \cdot \left(\mathbf{H}_1(q_1) \cdots \hat{\mathbf{H}}_j \exp(q_j \hat{\mathbf{H}}_j) \cdots \mathbf{H}_6(q_6) \right) \mathbf{H}^{-1}(\boldsymbol{q}) \\ \hat{\mathbf{H}} &= \sum_{j=1}^{n} \dot{q}_j \cdot \left(\bar{\mathbf{H}}_{j-1}(\boldsymbol{q}) \; \hat{\mathbf{H}}_j \; \bar{\mathbf{H}}_{j-1}^{-1}(\boldsymbol{q}) \right). \end{aligned} \tag{5.40}$$

Also, the body velocity $\check{\mathbf{H}}$ (2.44) of the tip element is:

$$\check{\mathbf{H}} = \mathbf{H}^{-1}(\boldsymbol{q}) \cdot \dot{\mathbf{H}}(\boldsymbol{q}). \tag{5.41}$$

5.3.2 Projective Jacobian: general reference

Now, a second version of the Jacobian is presented. It differs from the first one mainly in the fact that it applies no longer to point-coordinates valid at zero, but to point-coordinates currently valid at $\boldsymbol{Q} = \boldsymbol{q}(t)$. Although the proof and the formalism are somewhat different, the resulting projective velocities necessarily correspond.

Theorem 12
Assume an articulated mechanism and its projective kinematic geometry given in terms of operators $\hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}}$, that represent the action of joint j for the configuration $\boldsymbol{Q} = \boldsymbol{q}(\boldsymbol{t})$, i.e. the current configuration. Consider further an arbitrary point, and its current coordinates $\boldsymbol{M}(t)$. Then, an articulated motion with velocity $\dot{\boldsymbol{q}}$ results in the tip moving with

$$\hat{\mathbf{H}}\big|_{\boldsymbol{Q}} = \sum_{j=1}^{6} \dot{q}_j \quad \hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}} \quad , \tag{5.42}$$

$$\hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}} = \bar{\mathbf{H}}_{j-1}(\boldsymbol{Q}) \; \hat{\mathbf{H}}_j \; \bar{\mathbf{H}}_{j-1}^{-1}(\boldsymbol{Q}), \tag{5.43}$$

and a point $\boldsymbol{M}(t)$ on the tip moving with

$$\dot{\boldsymbol{M}} = \hat{\mathbf{H}}\big|_{\boldsymbol{Q}} \cdot \boldsymbol{M}(t). \tag{5.44}$$

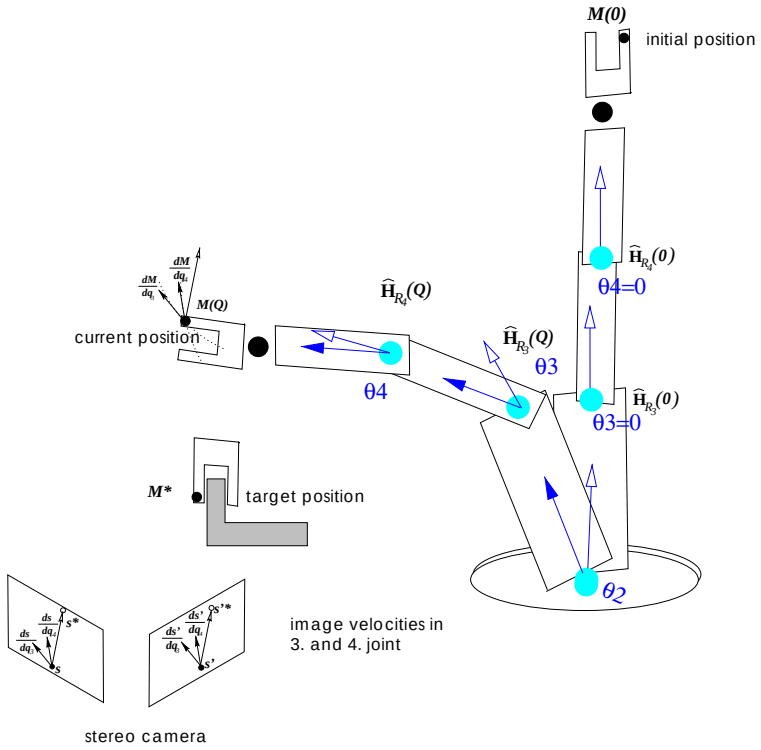

Figure 5.12: The robot-image Jacobian in the general form for the robot currently at $\boldsymbol{Q}$. The joint operators relate the 3D-velocities of $\boldsymbol{M}$, and respective image-velocities of $\boldsymbol{s}$ and $\boldsymbol{s}'$ in pixel coordinates.

Proof:
At a time instant t, the chain is in configuration $\boldsymbol{Q}$ while moving instantaneously with joint-space velocity $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}(t)$. The update of the projective kinematic model around $\boldsymbol{Q}$ – being now the effective origin of a shifted joint-space – is given in (5.21). At the new zero, the joint-action $\mathbf{H}_j\big|_{\boldsymbol{Q}}(\boldsymbol{Q}) = \mathbf{I}$ of each joint vanishes. Thus, the partial derivatives of (5.22) are strictly equal to the shifted operators $\hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}}$, as all the factors $\exp\left((\boldsymbol{q}_j - \boldsymbol{Q}_j)\hat{\mathbf{H}}_{jR}\right)$ evaluate to the identity $\mathbf{I}$:

$$\frac{\partial \mathbf{H}\big|_{\boldsymbol{Q}}}{\partial q_j} = \hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}}. \tag{5.45}$$

As the projective kinematics refers to the current configuration $\boldsymbol{Q}$, it can be applied directly to arbitrary points in three-space, i.e. to their current coordinates $\boldsymbol{M}(t)$. The result is a projective point velocity, as if this point

were moving rigidly with the mechanism's tip.

$$\dot{\boldsymbol{M}} = \dot{\mathbf{H}}(\boldsymbol{q})\boldsymbol{M} = \sum_{j=1}^{n} \dot{q}_j \hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}} \boldsymbol{M}. \tag{5.46}$$

This can be written in terms of a $4 \times n$ matrix $\mathbf{J}(\boldsymbol{Q}, \boldsymbol{M})$, the Jacobian of this point

$$\mathbf{J}(\boldsymbol{Q}, \boldsymbol{M}(t)) = \left[\hat{\mathbf{H}}_1\big|_{\boldsymbol{Q}} \boldsymbol{M}(t), \quad \ldots \quad , \hat{\mathbf{H}}_n\big|_{\boldsymbol{Q}} \boldsymbol{M}(t)\right], \tag{5.47}$$

$$\dot{\boldsymbol{M}} = \mathbf{J}(\boldsymbol{Q}, \boldsymbol{M}(t)) \, \dot{\boldsymbol{q}}. \tag{5.48}$$

As $\mathbf{H}\big|_{\boldsymbol{Q}}(\boldsymbol{Q}) = \mathbf{I}$, the spatial velocity (2.42) of the tip element is simply the weighted sum

$$\hat{\mathbf{H}} = \sum_{j=1}^{n} \dot{q}_j \hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}}. \tag{5.49}$$

In this particular case, thanks to the updating of the projective kinematics, the spatial and moving reference frame coincide at any instant, such that the body velocity $\check{\mathbf{H}}$ (2.44) is equal to the spatial one:

$$\check{\mathbf{H}} = \sum_{j=1}^{n} \dot{q}_j \hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}}(\boldsymbol{q}). \tag{5.50}$$

q.e.d.

The Jacobian in (5.47) is essential to sections 7.3 and 7.4, where an abstraction is made from the Jacobian of some physical point on a robot's tip to motion fields in projective three-space under the action of an articulated actuator, which generate a manifold representing projective rigid motion.

5.4 Numerical estimation

Once again (see sections 3.2,4.4), the basic principle is to define an objective function that takes model parameters and physical measurements while at the same time expressing a physically meaningful error-measure. Minimizing the objective then determines a best fit in the model parameters. A-priori, projective quantities, such as reconstructions $\boldsymbol{M}$, allow only an "algebraic" error to be defined, which lacks in the first instance any physical or statistical meaning.

5.4.1 Objective function

In the present case, the input data is robot joint-angles and image measurements in pixel-coordinates (Fig. 5.14). The angles are considered as perfectly

accurate, whereas the image measurements are assumed to be corrupted by additive Gaussian noise. Therefore, the main objective to be minimized is the Euclidean pixel distance $d(\boldsymbol{s}, \boldsymbol{m})$ between actually measured image points $\boldsymbol{s} = (s_u, s_v)$ and points $\boldsymbol{m}$ in the projective image plane, expressed as functions ((3.40), (4.42)) of the model parameters.

$$d(\boldsymbol{s}, \boldsymbol{m}) = \left(\left(s_u - \frac{m_1}{m_3} \right)^2 + \left(s_v - \frac{m_2}{m_3} \right)^2 \right)^{1/2}. \tag{5.51}$$

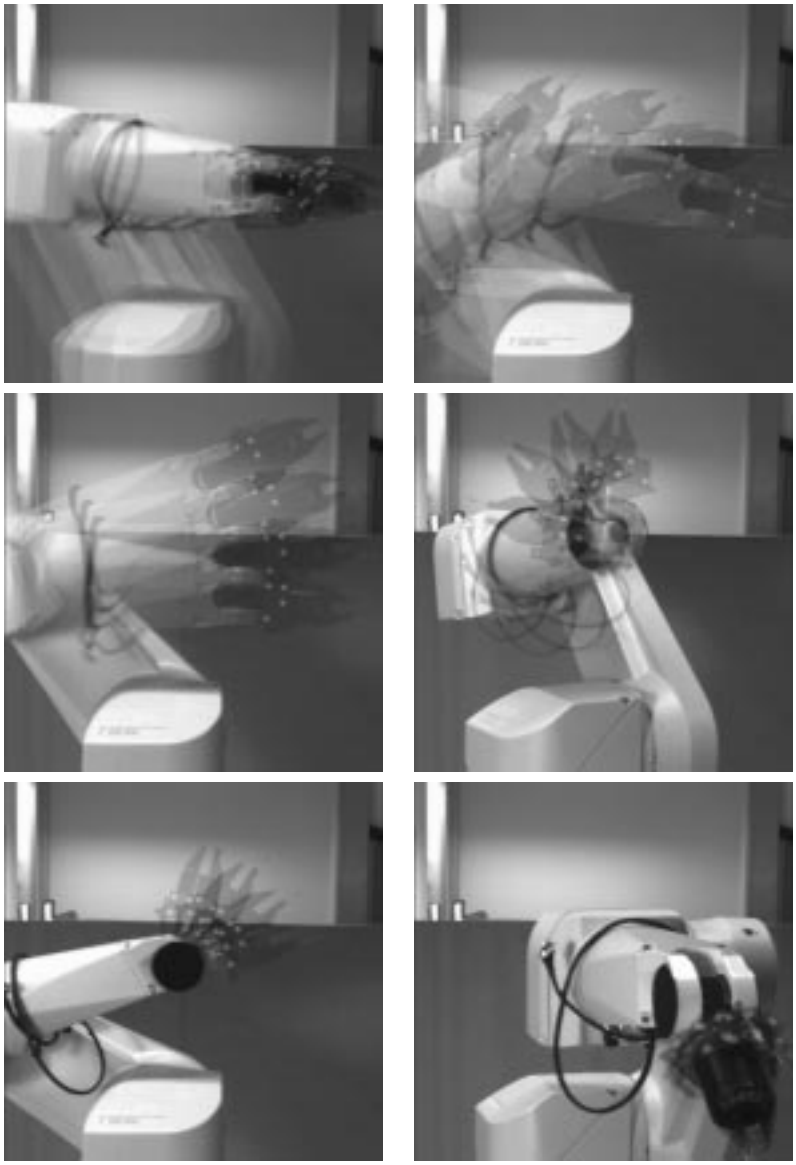

Figure 5.13: The six trial-motions.

For concreteness, consider $j = 1 \ldots 6$ joints: $\hat{\mathbf{H}}_{jR}$. The $l = 1 \ldots 6$ trial motions (Fig. 5.13) are driven from configurations $\boldsymbol{q}^{l1}$ to configurations $\boldsymbol{q}^{lk}$, where all joints but the l^{th} remain locked, while this one moves from angle θ^{l1} to angles θ^{lk}. For the sake of accurate image detection, the choice of positions $\boldsymbol{q}^{l1}$ is made to have image trajectories that are close to circles (Fig. 5.14. Correspondingly, the reconstructed markers in the first position are indexed as $\boldsymbol{M}^{l1}$, their subsequent 3D-positions as $\boldsymbol{M}^{lk}$, and their projective images as $\boldsymbol{m}^{lk}$ and $\boldsymbol{m}'^{lk}$. Hence, given $\boldsymbol{M}^{l1}$, all the subsequent positions are fully determined by equations (5.22) and (2.12), while the operators $\hat{\mathbf{H}}_{jR} = \hat{\mathbf{H}}_{jR}(\boldsymbol{q}^{l1})$ depend on the unknown model parameters. Supposing the image measurements to be indexed as $\boldsymbol{s}^{lk}$ and $\boldsymbol{s}'^{lk}$, the objective function (5.51) takes the form given in (5.52), below. An initial guess is found by applying (4.28) to a single rotation.

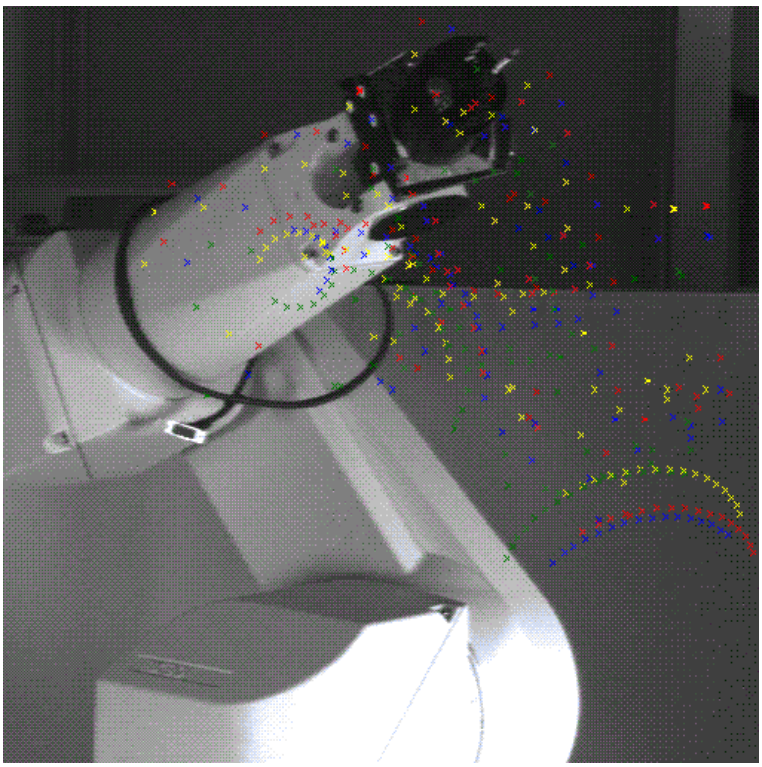

Figure 5.14: Total distribution of data-points in image-space.

5.4.2 Projective parameterization

The implementation of this minimization is non-trivial, as the 16 coefficients of $\hat{\mathbf{H}}_{jR}$ have only 10 degrees of freedom, due to algebraic constraints and free "gauge parameters". In section 4.4, a minimal parameterization is devised, together with explicit solutions of the resulting constraints. Note,

the minimizations (5.52)-(5.55) are implemented using numerical non-linear optimization over e.g. a 10-parameter-vector $\boldsymbol{x}_{jR}$ for each joint j. They are expanded in terms of the operators $\hat{\mathbf{H}}_{jR}$ for the computation of the objective function.

A three-step estimation process is devised:

Separate joints

The joints $\hat{\mathbf{H}}_{jR}$ are treated separately, and the markers $\boldsymbol{M}^{l1}$ are reconstructed a-priori, separately for each joint.

Linked joints

The initially independent operators are gathered into an explicit and complete projective kinematic model, expressed in terms of operators at zero $\hat{\mathbf{H}}_{jR} = \hat{\mathbf{H}}_{jR}\big|_{(\boldsymbol{Q}=\mathbf{0})}$. An initial guess can be obtained from the results of the first step using the formalism in (5.21). Additionally, multiple markers, which up to now have been reconstructed independently for each joint, are also replaced by a single reconstruction, e.g. the one done a-priori at zero $\boldsymbol{M}_Q = \boldsymbol{M}^{l1}$. Thus, step two reflects the fact that the joints are part of a single articulated chain, with the four markers rigidly linked to its end. As a result, one obtains an initial estimate for all 60 parameters after minimizing equation (5.53).

Linked joints and rigid bundle

The 3D-coordinates of the points $\boldsymbol{M}_Q$ are considered as free parameters, and refined a-posteriori together with the kinematic parameters (5.55). This removes the bias that results from privileging a-priori the points $\boldsymbol{M}_Q$, as this is done in (5.53) and (5.52), and significantly improves the precision (Table 5.1).

$$for\ each\ j\ \min_{\hat{\mathbf{H}}_{jR}} \sum_k d(\boldsymbol{s}^{lk}, \mathbf{P}\exp(\theta^{lk}\hat{\mathbf{H}}_{Rl})\boldsymbol{M}^{l1}) + d(\boldsymbol{s}'^{lk}, \mathbf{P}'\exp(\theta^{lk}\hat{\mathbf{H}}_{jR})\boldsymbol{M}^{l1}). \tag{5.52}$$

$$\min_{all\ \hat{\mathbf{H}}_{jR}} \sum_l \sum_k d\left(\boldsymbol{s}^{lk}, \mathbf{P}\ \mathbf{H}_{RT}(\boldsymbol{q}^{lk})\ \boldsymbol{M}_Q\right) + d\left(\boldsymbol{s}'^{lk}, \mathbf{P}'\ \mathbf{H}_{RT}(\boldsymbol{q}^{lk})\ \boldsymbol{M}_Q\right). \tag{5.53}$$

$$\text{where } \mathbf{H}_{RT}(\boldsymbol{q}^{lk}) = \prod_i \exp(\boldsymbol{q}_i^{lk})\hat{\mathbf{H}}_{jR}. \tag{5.54}$$

$$\min_{\substack{all\ four\ \boldsymbol{M}_Q \\ all\ \hat{\mathbf{H}}_{jR}}} \sum_l \sum_k d\left(\boldsymbol{s}^{lk}, \mathbf{P}\ \mathbf{H}_{RT}(\boldsymbol{q}^{lk})\ \boldsymbol{M}_Q\right) + d\left(\boldsymbol{s}'^{lk}, \mathbf{P}'\ \mathbf{H}_{RT}(\boldsymbol{q}^{lk})\ \boldsymbol{M}_Q\right). \tag{5.55}$$

5.4.3 Metric parameterization

From a projective point of view, the projective representation of a revolute joint has the advantage that it minimally but fully characterizes the relationship between image-space and one dimension of joint-space, whereas a "physical" or metric representation is obviously over-parameterized. Counting variables in (4.7) yields 5 parameters for the camera's CCD, 3 for the plane at infinity, 4 for the 3D-location of the axis, plus the angle. In contrast, $\hat{\mathbf{H}}_{jR}$ has just 10 independent parameters [59]. In addition, work on stereo self-calibration of mobile robots undergoing ground-plane motions [11], which are basically pure rotations around vertical axes called "pole", has shown that in case of a single fixed plane, at least two trial motions determine the infinity plane [4]. Moreover, in case of a single fixed pole, any number of trial motions determine a line at infinity but not the full plane [36].

From a numeric point of view, the purely projective recovery of the kinematic model fails to exactly impose the constraints arising from the fact that the upgrade $\mathbf{H}_{PE}$ is constant for all generators $\hat{\mathbf{H}}_{jR}$. In order to incorporate this, an initial guess for $\mathbf{H}_{PE}$ is used for upgrading a twist from the projective to the metric parameterization

$$\hat{\mathbf{T}}_{jR} = \mathbf{H}_{PE}\hat{\mathbf{H}}_{jR}\mathbf{H}_{PE}^{-1}.$$

Then, the minimization is run once again over the complete kinematic model. However, this is now over a set of metric parameters: six twists $\hat{\mathbf{T}}_{jR}$ and the camera upgrade $\mathbf{H}_{PE}$. In fact, this amounts to strong camera calibration in addition to classical robot calibration, albeit up to an unknown global scale, however.

$$\min_{\substack{all\ four\ \boldsymbol{M}_Q \\ all\ \hat{\mathbf{T}}_{jR} \\ \mathbf{H}_{PE}}} \sum_l \sum_k \begin{pmatrix} d\left(\boldsymbol{s}^{lk}, \mathbf{P}\ \mathbf{H}_{PE}^{-1}\mathbf{T}(\boldsymbol{q}^{lk})\mathbf{H}_{PE}\ \boldsymbol{M}_Q\right) + \\ d\left(\boldsymbol{s}'^{lk}, \mathbf{P}'\ \mathbf{H}_{PE}^{-1}\mathbf{T}(\boldsymbol{q}^{lk})\mathbf{H}_{PE}\ \boldsymbol{M}_Q\right) \end{pmatrix}, \tag{5.56}$$

$$\text{where}\quad \mathbf{T}(\boldsymbol{q}^{lk}) = \prod_j \exp\left(\boldsymbol{q}_j^{lk}\hat{\mathbf{T}}_{jR}\right). \tag{5.57}$$

5.4.4 Initialization

The initial guess for $\mathbf{H}_{PE}$ is the critical issue. Guessing $\mathbf{K}$ requires factory-given values. The infinity plane can be guessed from an initial algebraic estimate. Since it is the only plane invariant to several rotations around differently oriented axes, it can be directly calculated as the common right eigenvector with eigenvalue $\mu = 1$ of a number of observed projective rotations $\mathbf{H}_R$ [59]. In practice, the numerical implementation is by singular value decomposition of the matrix stack build-up from the plane homographies $\mathbf{H}_R^{-T}$ minus a unit matrix.

$$SVD\begin{bmatrix} \mathbf{H}_R^{-\top} - \mu\mathbf{I} \\ \vdots \end{bmatrix} \tag{5.58}$$

Alternatively, the common null-space for a stack of dual operators $\hat{\mathbf{H}}^\top$ can be computed

$$\ker\begin{bmatrix} -\mathbf{H}_{R1}^\top \\ -\mathbf{H}_{R2}^\top \\ \vdots \end{bmatrix}. \tag{5.59}$$

5.5 Experiments

This section briefly discusses the experimental results published in [62]. A number of experiments including self-calibration of a robot that moves its articulations, one by one, in front of an uncalibrated, and the subsequent recovery of additional metric parameters for robot and cameras are discussed here.

Throughout all the experiments, the input data consists of the image-points of four white markers (Fig. 5.15) on a black plate rigidly attached to the end-effector. They are extracted from the raw images by first thresholding the intensity image and then locating the centers of gravity with subpixel accuracy. Additionally, the robot's joint-configuration $\boldsymbol{q}(t)$ is read each time t a stereo image is taken. For the sake of simplicity, we use single letters to generically denote all four points: $\boldsymbol{M}$ for the reconstruction, $\boldsymbol{m}$, $\boldsymbol{m}'$ for its

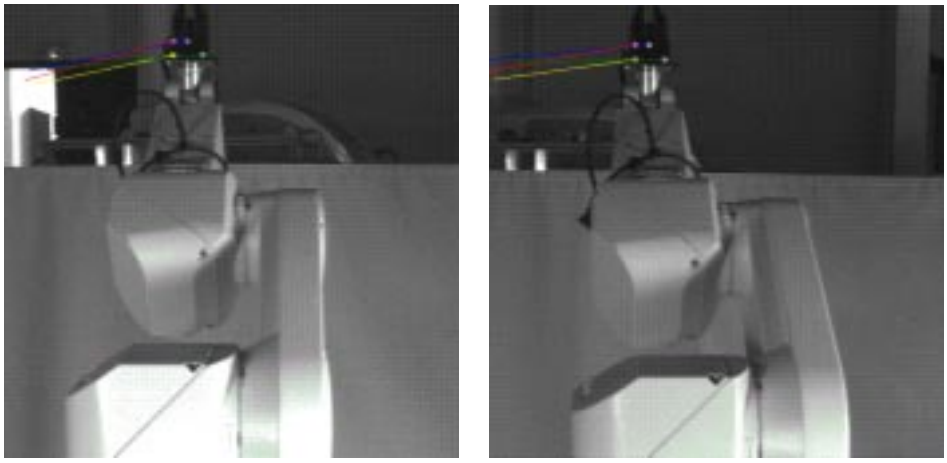

Figure 5.15: Track of four marker-points in image-space.

theoretical, and $\boldsymbol{s}$, $\boldsymbol{s}'$ for its actual, image-projections. Since a homography requires at least 5 rigid points, but the plate shows just 4 markers, their number is multiplied to 16 by turning the plate four times, similar to the last image.

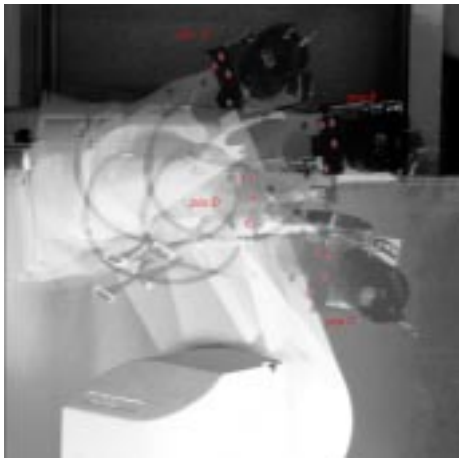

Figure 5.16: A rigid structure of $16 = 4 \cdot 4$ points is generated by moving the plate on the end-effector in four rigidly linked position.

5.5.1 Evaluation of image error

This section quotes the results of an experiment on projective self-calibration of a Stäubli RX90 robot having a PUMA-like geometry [54]. Unfortunately, besides a factory given kinematic model and a prior off-line calibration of the cameras, no really precise ground-truth of the kinematic parameters

was available. However, the particular strength of the proposed method is that the usual requirements for a precisely calibrated sensor and a precisely known calibration tool can be dropped, while at the same time the results obtained compare favorably with state-of-the-art methods for stereo self-calibration or grid-based camera calibration.

calibration:	fixed 3D-points image-error [px]	free 3D-points image-error [px]
axis 1	0.55	0.13
axis 2	0.14	0.10
axis 3	0.15	0.09
axis 4	0.16	0.12
axis 5	0.16	0.10
axis 6	0.24	0.16

Table 5.1: Comparison of reprojection-errors obtained with privileged or free varying 3D-points. The given values are mean image-errors over the entire trial motion of the respective joint (Cf. eqs. (5.53) and (5.55))

5.5.2 Evaluation of metric error

In order to evaluate the metric dimensions "hidden" by the projective model, the results of an experiment for stereo self-calibration from robot motions are quoted. In this case, the kinematic parameters are separated from the camera ones, so that the robot's Denavit–Hartenberg parameters can be identified, which is possible only up to a global scale. So, first a projective calibration of the robot is fitted to the measured date at maximal precision (Table 5.2), i.e. subpixel-accuracy of marker extraction. Based on this, a metric self-calibration is recovered, that shows very good agreement, even for the principal point, although it is known to be inherently unstable. Similarly, the kinematics now expressed in Denavit-Hartenberg parameters (Tables 5.3, 5.4) agrees with the factory-given values at an accuracy that compares favourably with off-the-shelf robot calibration.

calibration of:	off-line	experiment
right camera		
$\alpha_u[mm/px]$	-1516.1	-1505.9
$\alpha_v[mm/px]$	1521.5	1501.9
$u_0[px]$	260.5	260.6
$v_0[px]$	252.6	245.8
left camera		
$\alpha_u[mm/px]$	-1534.3	-1512.7
$\alpha_v[mm/px]$	1539.6	1507.5
$u_0[px]$	255.5	250.4
$v_0[px]$	266.8	254.6

Table 5.2: Comparison of intrinsic parameters, grid-based calibration vs. a-posteriori robot self-calibration.

axis angles	Factory $[^o]$	Experiment $[^o]$
axis 1, 2	90	89.72
axis 2, 3	0	0.28
axis 3, 4	90	89.71
axis 4, 5	90	90.002
axis 5, 6	90	90.10

Table 5.3: Comparison of angles between successive axis (unaffected by offsets), factory-given vs. measured values.

link lengths	Factory [mm]	Experiment [mm]
link 1	0	0.5522
link 2 (scale)	400	400
link 3	800	800.5457
link 4	0	0.0633
link 5	0	0.0347
link axis 4, 6	0.3	0.1895

Table 5.4: Comparison of link lengths, factory-given values with measured values. Here, the length of link 2 has been used to adjust the free scale factor.

Chapter 6

Projective Inverse Kinematics

This chapter is devoted to the inverse kinematic problem and in particular to the inversion of projective kinematic models. For such models, detailed solutions are derived for prismatic joints, revolute joints, and serial linkages of up to three revolute joints having a configuration of either an arm or a wrist. Finally, these subproblems allow modular solutions for the inverse of common industrial manipulators to be derived. The classical solutions to these problems make heavy use of Euclidean geometry, e.g. metric distances and trigonometry, but the projective model lacks such explicit metric concepts. The main contribution here is thus to derive inverse solutions for projective kinematics based only on incidence constrains between planes and points, i.e. their positions under the action of the mechanism.

6.1 Prismatic joints

Consider a mechanism consisting of a number of prismatic joints represented by matrix operators $\hat{\mathbf{H}}_{1T}, \hat{\mathbf{H}}_{2T}, \hat{\mathbf{H}}_{3T}$, which can be projective translations in the most general case. Consider further a point $\boldsymbol{A}$ representing the tip of the mechanism. It has $n \leq 3$ degrees of freedom, depending on the number n of joints, and can be moved respectively along a line in space, on a plane in space, or arbitrarily within three-space. The spatial position of $\boldsymbol{A}$ as a function of the joint configurations τ_1, τ_2, τ_3 can thus be written as (6.1), modelling for instance the three linear axes of a Cartesian robot, where τ_i

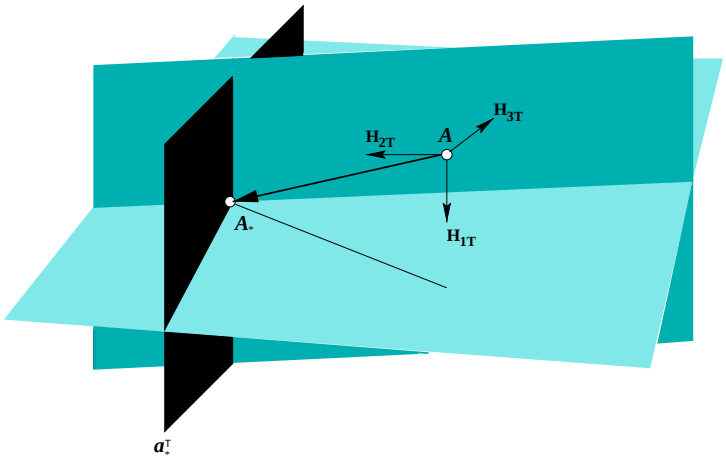

Figure 6.1: Three prismatic joints.

stands for encoder values in the most general case:

$$\boldsymbol{A}(\tau_1,\tau_2,\tau_3) = \left(\mathbf{I}+\tau_1\hat{\mathbf{H}}_{1T}\right)\cdot\left(\mathbf{I}+\tau_2\hat{\mathbf{H}}_{2T}\right)\cdot\left(\mathbf{I}+\tau_3\hat{\mathbf{H}}_{3T}\right)\cdot\boldsymbol{A}, \tag{6.1}$$

$$= \left(\mathbf{I}+\tau_1\hat{\mathbf{H}}_{1T}+\tau_2\hat{\mathbf{H}}_{2T}+\tau_3\hat{\mathbf{H}}_{3T}\right)\cdot\boldsymbol{A}, \tag{6.2}$$

$$= \left(\mathbf{I}\boldsymbol{A}+\tau_1\hat{\mathbf{H}}_{1T}\boldsymbol{A}+\tau_2\hat{\mathbf{H}}_{2T}\boldsymbol{A}+\tau_3\hat{\mathbf{H}}_{3T}\boldsymbol{A}\right), \tag{6.3}$$

$$= \tau_1\boldsymbol{A}_{1T}+\tau_2\boldsymbol{A}_{2T}+\tau_3\boldsymbol{A}_{3T}+\boldsymbol{A}, \tag{6.4}$$

$$= \begin{bmatrix} \boldsymbol{A}_{1T} & \boldsymbol{A}_{2T} & \boldsymbol{A}_{3T} & \boldsymbol{A} \end{bmatrix}\cdot\begin{pmatrix}\tau_1\\ \tau_2\\ \tau_3\\ 1\end{pmatrix}. \tag{6.5}$$

Consider now a goal position $\boldsymbol{A}^*$ for this point to attain and its dual in form of the orthogonal complement $\boldsymbol{A}_*^\perp$, a 3×4 linear form with $\boldsymbol{A}_*^\perp\cdot\boldsymbol{A}^* = \mathbf{o}$. The goal condition $\boldsymbol{A} = \boldsymbol{A}_*$ gives rise to a homogeneous system of four linear equations in τ_1,τ_2,τ_3

$$\boldsymbol{A}_*^\perp\begin{bmatrix} \boldsymbol{A}_{1T} & \boldsymbol{A}_{2T} & \boldsymbol{A}_{3T} & \boldsymbol{A} \end{bmatrix}\cdot\nu\begin{pmatrix}\tau_1\\ \tau_2\\ \tau_3\\ 1\end{pmatrix} = \mathbf{o} \tag{6.6}$$

Geometrically, each row of the dual $\boldsymbol{A}_*^\perp$ represents a plane in three-space, the three planes intersecting in $\boldsymbol{A}_*$. Apart from this, the choice of such three planes is arbitrary. For a plane $\boldsymbol{a}_*^\top$ to constrain a degree of freedom of the mechanism it has to have an orientation transverse to that of the corresponding prismatic joint (Fig. 6.1). A choice arises as soon as the number n of joints is less than 3. Assuming that $\boldsymbol{A}_*$ is attainable, either a pseudo-inverse solution of (6.6) is determined, or, neglecting unactuated

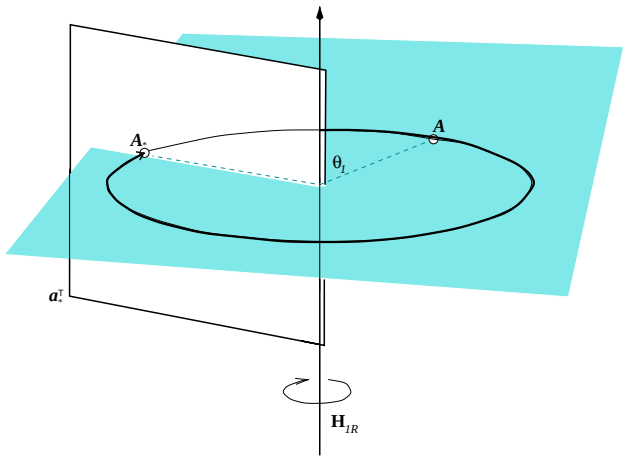

Figure 6.2: Single revolute joint.

degrees of freedom, an exact solution is determined by voluntarily selecting just n transversal planes through $\boldsymbol{A}_*$. Additional parallel planes, which fail to constrain certain degrees of freedom of the mechanism, can also be chosen so that the system becomes upper triagonal, or diagonal (see also section 7.3.1). The case of a single joint $\hat{\mathbf{H}}_{1T}$ and a single plane $\boldsymbol{a}_*^\top$ for instance simplifies to:

$$\boldsymbol{a}_*^\top \left(\mathbf{I} + \tau_1\ \hat{\mathbf{H}}_{1T}\right) \boldsymbol{A} = \mathbf{o}, \qquad \tau_1 = -\frac{\boldsymbol{a}_*^\top \boldsymbol{A}}{\boldsymbol{a}_*^\top \hat{\mathbf{H}}_{1T} \boldsymbol{A}}. \tag{6.7}$$

6.2 Single revolute joint

Consider a mechanism whose only degree of freedom is a revolute joint represented by a respective matrix operator $\hat{\mathbf{H}}_{1R}$, generating a projective rotation in the most general case. Consider further a point $\boldsymbol{A}$ rigidly linked to the joint, so that it moves on a circular trajectory $\boldsymbol{A}(\theta_1)$ as a function of the joint angle θ_1. Again, consider a goal position $\boldsymbol{A}_*$ and its dual complement $\boldsymbol{A}_*^\perp$, each row of which provides a constraint. One such is furthermore sufficient if its plane is transversally oriented and passes through the axis while $\boldsymbol{A}_*$ has to be attainable. Under such a necessary and sufficient condition, the joint angle θ_1 is determined up to the twofold ambiguity in $\theta_1 - \pi$, which however is trivially resolvable by direct comparison.

Therefore, the inversion of a projective revolute joint reduces to solving the intersection of the circle $\boldsymbol{A}(\theta_1)$ with a plane $\boldsymbol{a}_*^\top$ for θ_1 (Fig. 6.2):

$$\begin{array}{ccccccc}
\boldsymbol{a}_*^\top\Big(\mathbf{I} & + \sin\theta_1 & \hat{\mathbf{H}}_R & + (1-\cos\theta_1) & \hat{\mathbf{H}}_R^2\Big)\boldsymbol{A} & = & 0, \\
\Big(\boldsymbol{a}_*^\top \boldsymbol{A} + \boldsymbol{a}_*^\top \hat{\mathbf{H}}_R^2 \boldsymbol{A}\Big) & + \sin\theta_1 & \Big(\boldsymbol{a}_*^\top \hat{\mathbf{H}}_R \boldsymbol{A}\Big) & - \cos\theta_1 & \Big(\boldsymbol{a}_*^\top \hat{\mathbf{H}}_R^2 \boldsymbol{A}\Big) & = & 0, \\
p_0 & + \sin\theta_1 & p_s & - \cos\theta_1 & p_c & = & 0.
\end{array}$$

(6.8)

The result is a first-order trigonometric equation in θ_1 with coefficients p_0, p_s, p_c (**??**). Half-angle substitution yields a quadratic equation in $\tan\frac{\theta_1}{2}$. So, the tangent's analytic inverse atan2 yields a pair of solutions due to the squareroot's ambiguouity in the argument $\alpha_{1/2}$ [1]. Geometrically, the problem has the trivial ambiguity in the sense of the rotation (6.10). The total number of distinct solution can be determined by simple comparison.

$$\theta^+_{1/2} = \text{atan2}(\alpha_{1/2}(p_0, p_s, p_c), \beta(p_0, p_s, p_c)), \tag{6.9}$$

$$\theta^-_{1/2} = \pi - \theta^+_{1/2}. \tag{6.10}$$

It is worthwhile to discuss the choice of the transversal plane $\boldsymbol{a}^\top_*$: Unless the goal lies exactly on the circle, the best approximation for the joint angle is the one that moves $\boldsymbol{A}(\theta_1)$ closest to the goal. Again, although "closest" cannot be explicitly expressed in the projective case, moving $\boldsymbol{A}$ onto the "axial" plane between the axis of $\hat{\mathbf{H}}_R$ and $\boldsymbol{A}_*$ yields in fact the best position under the mobility restriction imposed by the joint (Fig. 6.2). Such a plane is easily computed as follows

$$\boldsymbol{a}^\top \cdot \begin{bmatrix} \boldsymbol{A}_* & \hat{\mathbf{H}}_{1R}\boldsymbol{A} & \hat{\mathbf{H}}^2_{1R}\boldsymbol{A} \end{bmatrix} = 0 \tag{6.11}$$

6.3 Two revolute joints: general configuration

Consider now a mechanism consisting of two non-parallel revolute joints represented by the two matrix operators $\hat{\mathbf{H}}_{1R}, \hat{\mathbf{H}}_{1R}$ and two joint angles θ_1, θ_2. Again, a point $\boldsymbol{A}$ is rigidly linked with the tip and its goal position $\boldsymbol{A}_*$ sufficiently constrain the mechanism's two degrees of freedom. Inverting such a serial linkage is more involved than in case of prismatic joints (section 6.1), as the simultaneous solution of two trigonometric equations like (**??**) – especially in a least-squares sense, as is desirable in the presence of round-off errors or measurement noise – is far from being a trivial problem.

Therefore, the inversion will be reduced geometrically to solving twice the one-joint case (section 6.2), while at the same aiming for graceful degradation in the presence of corrupted inputs.

Geometrically, circular trajectories can be associated with both the forward motion of $\boldsymbol{A}$ (6.13) under $\hat{\mathbf{H}}_{R2}$ and the "backward motion" of $\boldsymbol{A}_*$ under $\hat{\mathbf{H}}_{R1}$ (6.14), thus defining two circles $\boldsymbol{A}(\theta_2)$ and $\boldsymbol{A}_*(\theta_1)$ as functions of the

[1] Exact algebraic solution calculated using Maple.

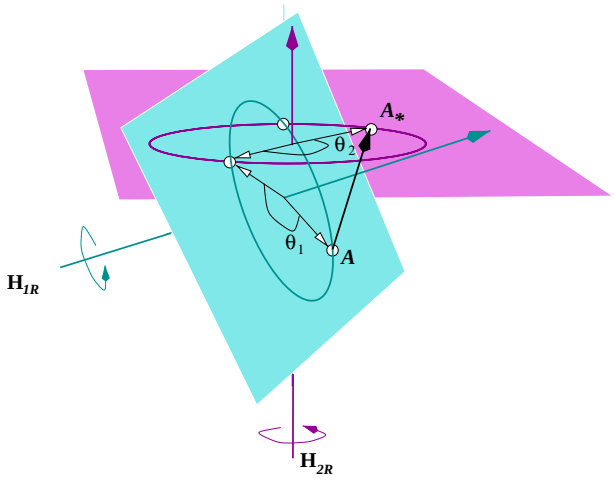

Figure 6.3: A general pair of revolute joints.

joint angles.

$$\boldsymbol{A}_* = \mathbf{H}_{R1}(\theta_1) \cdot \mathbf{H}_{R2}(\theta_2) \cdot \boldsymbol{A},$$

$$\mathbf{H}_{R1}^{-1}(\theta_1) \cdot \boldsymbol{A}_* = \mathbf{H}_{R2}(\theta_2) \cdot \boldsymbol{A}, \tag{6.12}$$

$$\text{where} \quad \boldsymbol{A}_*(\theta_1) = \mathbf{H}_{R1}(-\theta_1) \cdot \boldsymbol{A}_*, \tag{6.13}$$

$$\boldsymbol{A}(\theta_2) = \mathbf{H}_{R2}(\theta_2) \cdot \boldsymbol{A}. \tag{6.14}$$

In this sense, the overall solution can be determined by the intersection (6.12) of these two circles. A total of zero, one, two, or infinitely many solutions are possible. However, intersecting two quadrics is cumbersome and not very robust, and ultimately it is the joint angles and not the points of intersection that one is looking for. For this reason, it is worthwhile to observe that each of the circles lies entirely within a plane, $\boldsymbol{a}^\top$ or $\boldsymbol{a}_*^\top$, defined as the span of three points

$$\boldsymbol{a}^\top \cdot \left[\boldsymbol{A} \; \hat{\mathbf{H}}_{1R} \boldsymbol{A} \; \hat{\mathbf{H}}_{1R}^2 \boldsymbol{A} \right] = 0 \qquad \boldsymbol{a}_*^\top \cdot \left[\boldsymbol{A}_* \; \hat{\mathbf{H}}_{2R} \boldsymbol{A}_* \; \hat{\mathbf{H}}_{2R}^2 \boldsymbol{A}_* \right] = 0$$

Consequently, two necessary conditions on θ_1, θ_2 are

$$\boldsymbol{a}^\top \mathbf{H}_{R2}(\theta_2) = 0, \tag{6.15}$$

$$\boldsymbol{a}_*^\top \mathbf{H}_{R1}(-\theta_1) = 0. \tag{6.16}$$

They can be solved as two independent one-joint problems, e.g. using a transveral plane as proposed in (6.11), resulting in two pairs of solutions in total. Among them, the ones corresponding to feasible articulated motions (θ_1, θ_2) have to be selected, where up to four such tuples are possible, not mentioning the trivial ambiguities in $\pi \pm \theta_1$.

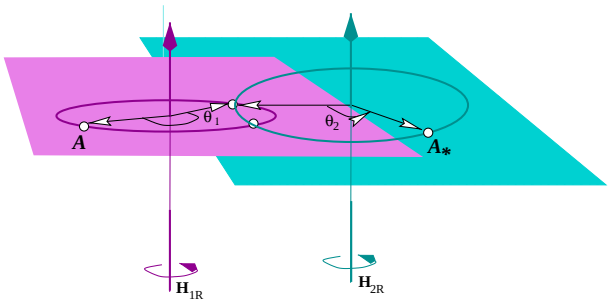

Figure 6.4: Two parallel revolute joint.

In practice, if the goal is attainable but the inputs are slightly corrupted, a more robust computation solves first for θ_1, to establish $\boldsymbol{A}' = \boldsymbol{A}(\theta_1)$ and then determines θ_2 by solving for θ_2 in $\boldsymbol{a}_*^\top \cdot \mathbf{H}_{R2}(\theta_2)\boldsymbol{A}'$, which has to be done for both solutions for $\boldsymbol{A}'$ (see section 6.2). In this way, the disambiguation is done incrementally "step-by-step".

6.4 Two revolute joints: parallel configuration

Consider a mechanism with two revolute joints having – in contrast to the previous section 6.3 – two parallel axis. Again, the two operators $\hat{\mathbf{H}}_{1R}$, $\hat{\mathbf{H}}_{2R}$ in conjunction with a point $\boldsymbol{A}$ and its goal position $\boldsymbol{A}_*$ give rise to two circles $\boldsymbol{A}_*(\theta_1)$, $\boldsymbol{A}(\theta_2)$ (6.14), (6.13).

Theoretically, they are either coplanar, if the goal is attainable, or they are parallel with a slight offset, if the goal position is slightly corrupted due to numerical artefacts or noise. In both cases, an intersection of the first circle with a plane containing the second circle is not well-defined. Additionally, even in the case of exact coplanar alignment, where quadric intersection is possible, the angles corresponding to the intersection remain to be recovered. Therefore, this work gives preference to a trigonometric solution aiming at graceful degradation in presence of corrupted inputs and numeric artifacts. It is based on the cosine-theorem and particularly on an affine line in the assumed common plane.

Compare Fig. 6.4 with Fig. 6.5, and focus on the triangle between the centers $\boldsymbol{C}_2$, $\boldsymbol{C}_3$ of the two circles and their intersection $\boldsymbol{Q}$. The edges a and b are the two radii, while the edge c is the distance between the two centers. Assuming that these three distances are known, the cosine theorem determines the two angles σ_2 and σ_3, which describe the intersection in terms of an angle above the centerline as in figure 6.5:

$$\cos\sigma_2 = \frac{b^2 + c^2 - a^2}{2bc}, \qquad \cos\sigma_3 = \frac{a^2 + c^2 - b^2}{2ac} \tag{6.17}$$

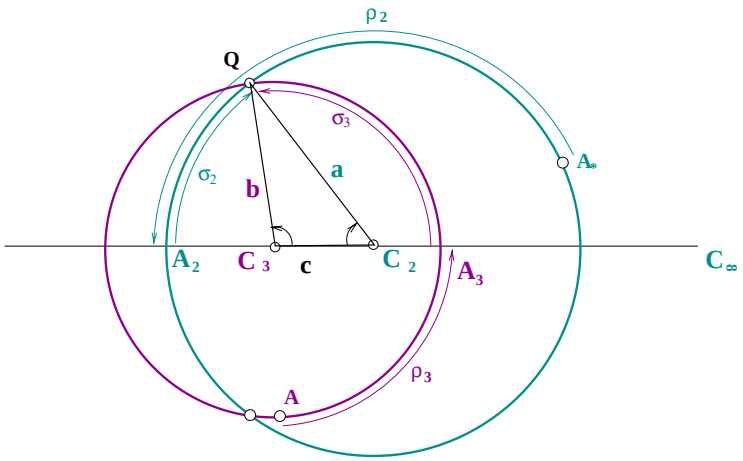

Figure 6.5: Cosine theorem in a plane for two parallel revolute joints.

The fact that lengths are known only up to a common scale factor does not affect the results.

The method is as follows:

- Firstly, determine for two given points $\boldsymbol{A}$, $\boldsymbol{A}_*$ the intersection of their circles with a transversal plane $\boldsymbol{c}_\top$ through the line joining the corresponding centers. Such a plane is defined e.g. by the parallel rotation axes. The intersections and angles follow from solving two a one-joint problem, one for $\boldsymbol{A}(\rho_3)$, and one for $\boldsymbol{A}_*(\rho_2)$.

$$\boldsymbol{c}^\top \mathbf{H}_{2R}(-\rho_2)\cdot\boldsymbol{A}_* \;=\; 0 \tag{6.18}$$

$$\boldsymbol{c}^\top \mathbf{H}_{3R}(\rho_3)\cdot\boldsymbol{A} \;=\; 0 \tag{6.19}$$

  The result is four points $\boldsymbol{A}(\rho_3)$, $\boldsymbol{A}(\rho_3-\pi)$, $\boldsymbol{A}_*(\rho_2)$, $\boldsymbol{A}_*(\rho_2-\pi)$ and their respective angles ρ_2, ρ_3 beneath the centerline. In the present case, the inner points are selected and denoted $\boldsymbol{A}_3$, $\boldsymbol{A}_2$.

- Secondly, define affine coordinates on this line, where for instance $\boldsymbol{C}_2$ and $\boldsymbol{C}_\infty = \hat{\mathbf{H}}^2_{R2}\boldsymbol{C}_3$ serve as origin and infinity point, and associate to each of the points its respective affine coordinates

$$\alpha_i = \frac{x_i}{y_i}, \qquad \text{with} \quad \begin{bmatrix} \boldsymbol{C}_2 & \boldsymbol{C}_\infty \end{bmatrix} \begin{pmatrix} y_i \\ x_i \end{pmatrix} = \boldsymbol{A}_i \tag{6.20}$$

$$\beta_i = \frac{x_i}{y_i}, \qquad \text{with} \quad \begin{bmatrix} \boldsymbol{C}_2 & \boldsymbol{C}_\infty \end{bmatrix} \begin{pmatrix} y_i \\ x_i \end{pmatrix} = \boldsymbol{C}_i \tag{6.21}$$

  Since all points have been put on the same line, affine ratios can be

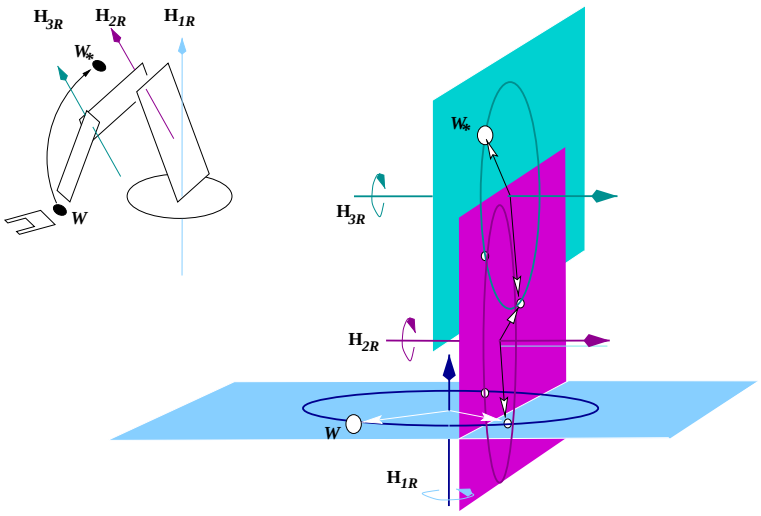

Figure 6.6: Three revolute joints in arm configuration.

used to determine the unknown distances

$$\begin{array}{lll} a & = \alpha_2 - \beta_2, \\ b & = \alpha_3 - \beta_3, \\ c & = \beta_3 - \beta_2 \end{array} \quad . \tag{6.22}$$

Although there is no guarantee that the points will be exactly colinear, the linear equations (6.20), (6.21) should give a reasonable result even in presence of slightly corrupted inputs.

- Thirdly, apply the cosine theorem to a,b, c, and determine a valid combination of angles ρ and σ.

6.5 Three revolute joints: arm configuration

Consider three revolute joints in arm configuration represented by matrix operators $\hat{\mathbf{H}}_{1R}$, $\hat{\mathbf{H}}_{2R}$, $\hat{\mathbf{H}}_{3R}$, i.e. the axes of $\hat{\mathbf{H}}_{2R}$, $\hat{\mathbf{H}}_{3R}$ are parallel and that of $\hat{\mathbf{H}}_{1R}$ has general orientation. Consider further a point $\boldsymbol{A}$ rigidly linked to the mechanism's tip, and its goal position $\boldsymbol{A}_*$. Clearly, under action of joints 2 and 3 there is a single plane $\boldsymbol{b}^\top$ that holds all trajectories of $\boldsymbol{A}$.

$$\boldsymbol{b}^\top = \begin{bmatrix} \boldsymbol{A} & \hat{\mathbf{H}}_{R2}\boldsymbol{A} & \hat{\mathbf{H}}_{R2}^2\boldsymbol{A} \end{bmatrix}^\perp , \qquad \boldsymbol{b}^\top = \begin{bmatrix} \boldsymbol{A} & \hat{\mathbf{H}}_{R3}\boldsymbol{A} & \hat{\mathbf{H}}_{R3}^2\boldsymbol{A} \end{bmatrix}^\perp . \tag{6.23}$$

This plane allows θ_2, θ_3 to be eliminated (6.24), and the problem simpli-

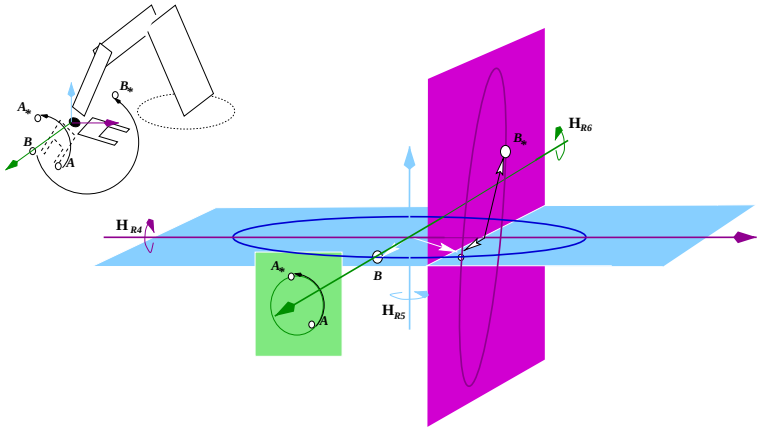

Figure 6.7: Three revolute joints in wrist configuration.

fies to the one-joint case (section 6.2) for θ_1 (6.25).

$$\begin{aligned} \boldsymbol{A}_* &= \mathbf{H}_{1R}(\theta_1)\mathbf{H}_{2R}(\theta_2)\mathbf{H}_{3R}(\theta_3)\boldsymbol{A} \\ \mathbf{H}_{1R}(-\theta_1)\boldsymbol{A}_* &= \mathbf{H}_{2R}(\theta_2)\mathbf{H}_{3R}(\theta_3)\boldsymbol{A} \\ \boldsymbol{b}^\top\mathbf{H}_{1R}(-\theta_1)\boldsymbol{A}_* &= \boldsymbol{b}^\top\mathbf{H}_{2R}(\theta_2)\mathbf{H}_{3R}(\theta_3)\boldsymbol{A} \qquad (6.24) \\ \boldsymbol{b}^\top\mathbf{H}_{1R}(-\theta_1)\boldsymbol{A}_* &= \boldsymbol{b}^\top\boldsymbol{A} = 0 \qquad (6.25) \end{aligned}$$

Since the first angle $-\theta_1$ can now be determined, the residual problem reduces to the two-parallel-joints case (section 6.4) for point $\boldsymbol{A}'_*$, the position of $\boldsymbol{A}_*$ after the action of the first joint:

$$\begin{aligned} \boldsymbol{A}'_* &= \mathbf{H}_{1R}(-\theta_1)\boldsymbol{A}_*, \\ \boldsymbol{A}'_* &= \mathbf{H}_{2R}(\theta_2)\mathbf{H}_{3R}(\theta_3)\boldsymbol{A}. \qquad (6.26) \end{aligned}$$

6.6 Three revolute joints: wrist configuration

Consider again three revolute joints $\hat{\mathbf{H}}_{4R}$, $\hat{\mathbf{H}}_{5R}$, $\hat{\mathbf{H}}_{6R}$, now in wrist configuration (Fig. 6.7), for instance the spherical one. Even though a single point imposes three plane constraints, they fail to allow for an elegant inverse solution, which furthermore fails to constrain all the mechanism's joints. So, in contrast to the previous cases, the goal position has here to be given in terms of a projective displacement, which, if joints have a spherical wrist configuration, is necessarily a projective rotation $\mathbf{H}_{*R}$ around the wrist-center $\boldsymbol{W}$ (Fig. 6.7).

Choosing a general point $\boldsymbol{A}$ on the axis of $\hat{\mathbf{H}}_{6R}$, for instance, and determining its goal position $\boldsymbol{A}_*$ allows θ_6 to be ignored, and reduces the problem

to the two-joint case (section 6.3):

$$\begin{aligned} \boldsymbol{A}_* &= \mathbf{H}_{4R}(\theta_4)\mathbf{H}_{5R}(\theta_5)\mathbf{H}_{6R}(\theta_6)\boldsymbol{A} \\ \boldsymbol{A}_* &= \mathbf{H}_{4R}(\theta_4)\mathbf{H}_{5R}(\theta_5)\boldsymbol{A} \\ \mathbf{H}_{4R}(-\theta_4)\boldsymbol{A}_* &= \mathbf{H}_{5R}(\theta_5)\boldsymbol{A} \end{aligned} \tag{6.27}$$

Since the first two angles θ_4,θ_5 can now be determined, the residual problem amounts to the one-joint case formulated for some point $\boldsymbol{B}$ not on axis 6.

$$\begin{aligned} \mathbf{H}_{6R}(\theta_6) &= \mathbf{H}_{5R}^{-1}(\theta_5)\mathbf{H}_{4R}^{-1}(\theta_4)\mathbf{H}_{*R} \\ \mathbf{H}_{*R}\boldsymbol{B} &= \mathbf{H}_{6R}(\theta_6)\boldsymbol{B} \end{aligned} \tag{6.28}$$

6.7 Six axis manipulator: PUMA geometry

Consider a six-axis manipulator having a PUMA-like geometry, i.e. the first three axes have arm configuration and the final three axes have a spherical wrist configuration. In particular, this means that there exists a single point, the wrist-center $\boldsymbol{W}$, which is common to all three wrist axes, and which consequently is not affected by $\mathbf{H}_{4R}, \dots \mathbf{H}_{6R}$. Consider further the goal position of the end-effector given as a projective displacement $\mathbf{H}_*$. The goal position $\boldsymbol{W}_* = \mathbf{H}_*\boldsymbol{W}$ of the wrist-center allows the unknown joint-angles $\theta_4 \cdots \theta_6$ to be eliminated, which amounts to simplifying the problem to the arm-case (section 6.5).

$$\begin{aligned} \mathbf{H}_*\boldsymbol{W} &= \mathbf{H}_{1R}(\theta_1)\mathbf{H}_{2R}(\theta_2)\mathbf{H}_{3R}(\theta_3)\mathbf{H}_{4R}(\theta_4)\mathbf{H}_{5R}(\theta_5)\mathbf{H}_{6R}(\theta_6)\boldsymbol{W}, \\ \boldsymbol{W}_* &= \mathbf{H}_{4R}(\theta_4)\mathbf{H}_{5R}(\theta_5)\mathbf{H}_{6R}(\theta_6)\boldsymbol{W}. \end{aligned} \tag{6.29}$$

Then, the residual inverse problem amounts to the wrist case (section 6.6) in $\theta_4, \dots \theta_6$, alone, as the arm angles $\theta_1, \dots, \theta_3$ are now known:

$$\begin{aligned} \mathbf{H}_{*R} &= \mathbf{H}_{4R}(\theta_4)\mathbf{H}_{5R}(\theta_5)\mathbf{H}_{6R}(\theta_6), \\ &= \left(\mathbf{H}_{1R}(\theta_1)\mathbf{H}_{2R}(\theta_2)\mathbf{H}_{3R}(\theta_3)\right)^{-1}\mathbf{H}_*. \end{aligned} \tag{6.30}$$

6.8 Numerical estimation

The main interest of the algebraic solution is to calculate a number of initial guesses for an exact numerical solution. Then, they can be used to initialize a non-linear bundle-adjustment where date-points are image-measurements, alone, and the free parameters are the joint angles.

$$\min_{q_1,\dots,q_6} \sum_{i=1}^{n} d(\boldsymbol{m}_i^*, \mathbf{P}\mathbf{H}(q_1, \dots, q_6)\boldsymbol{A}_i) + d(\boldsymbol{m'}_i^*, \mathbf{P}'\mathbf{H}(q_1, \dots, q_6)\boldsymbol{A}_i)$$

Such a solution should no longer be biased by the especial computational geometry used in the algorithms for modular inverse kinematics and projective reconstruction.

Chapter 7

Projective Trajectories

In this chapter, two approaches to trajectory generation for rigid projective motion are presented, with applications in visual servoing in mind. The fundamental difference between them lies in the geometrical concepts employed. The first is based on finite projective displacements – e.g. as captured by the 5 points in a homography – and their geometric relationship with rigid motion. The second is based on the differential geometry of projective motion – e.g. as captured by the Jacobian of a projective kinematic model – and its relationship with tangent spaces and velocities of rigid motion. Both methods decompose tasks into some primitive components and subsequently recombine them to formally describe a trajectory or a behavior of a control loop, for instance as Cartesian motion: rotation of a rigid body about a simultaneously translating center-point. It turns out that such decompositions and the constraints underlying them can be formulated directly in the projective task space associated with a stereo camera. In particular, conditions on the visual and physical feasibility of the trajectories can be formulated and monitored. In practice, as soon as a feasible trajectory is found, it can also be tracked easily with basic visual servoing, so that problems of local minima and convergence can be overcome.

7.1 Cartesian trajectories

With such scenarios in mind, visual servoing task will be augmented to a guided motion along a trajectory which realizes the desired alignment. As one choice among many, this thesis considers "*Cartesian*" trajectories, i.e. ones characterized by a straight-line motion of a particular point, called the "*center*", and a simultaneous rotation of the moving body about the translating center (Fig. 7.2).

Definition 11 (Cartesian motion) *A continuous motion in σ is called "Cartesian", if there is a center-point that moves along a straight line so*

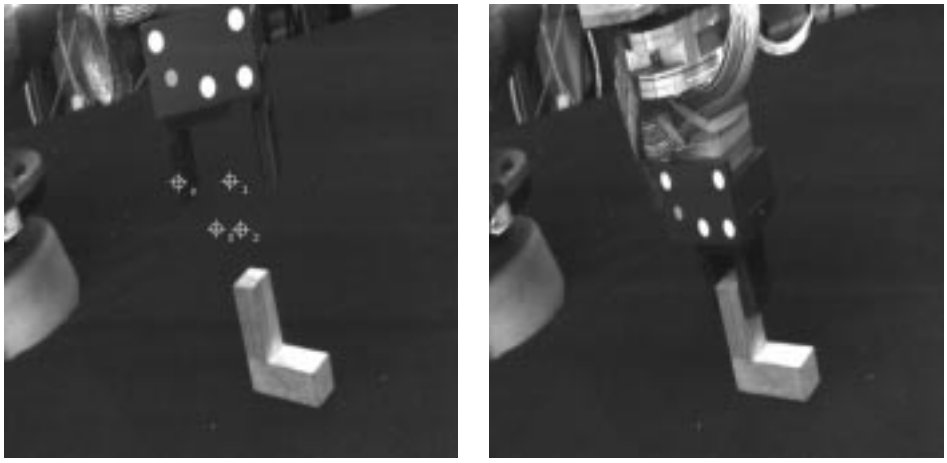

Figure 7.1: Alignment task encoded by means of a projective displacement, expressed relative to a basis of five points on the L-shaped object.

that the rotation at each instant is around this point. If the origin $\mathbf{0}$ *is chosen as center, a Cartesian motion* $T_c(\sigma)$ *of a point* $\boldsymbol{S}$ *has the general form*

$$\boldsymbol{S}(\sigma) = \mathbf{R}\left(\theta(\sigma)\right) \cdot \boldsymbol{S}(0) + \tau(\sigma) \cdot \boldsymbol{t}, \tag{7.1}$$

$$\mathbf{T}_c(\sigma) = \mathbf{T}_T\left(\tau(\sigma)\right) \cdot \mathbf{T}_R\left(\theta(\sigma)\right). \tag{7.2}$$

Such a trajectory can be regarded as "shortest way" in the respective configuration space associated with a rigid object.

7.2 Trajectories from a projective displacement

The task of alignment of all six degrees-of-freedom of a rigid body is unambiguously described by a projective displacement. Consider for instance the alignment of the tool with the object to be manipulated. It can be represented by means of the projective displacement $\mathbf{H}_{RT}$ between the current and the target position of the tool (Fig. 7.1). If the representation is based on a projective basis fixed rigidly to the tool, e.g. five corner points [39], it becomes calibration-free.

7.2.1 Decomposition

A general projective displacement can always be expressed as a rotation followed by a translation, both written projectively as

$$\mathbf{H}_{RT} = \mathbf{H}_T \cdot \mathbf{H}_R. \tag{7.3}$$

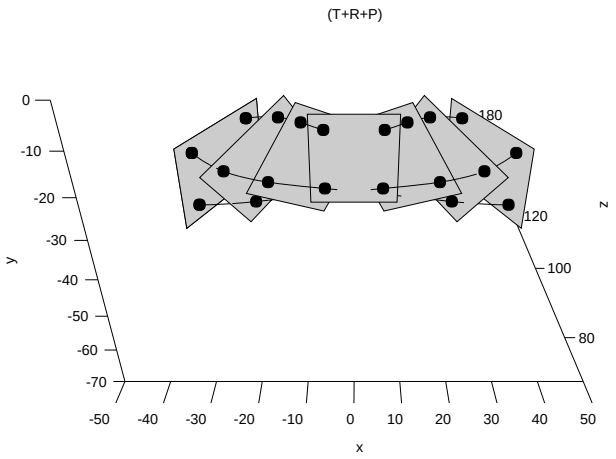

Figure 7.2: Cartesian motion: Simultaneous rotation and translation of a rigid body about a center point.

Assuming that the displacement of the center point from $\boldsymbol{A}_c$ to

$$\boldsymbol{A}_c^* = \mathbf{H}_{RT}\boldsymbol{A}_c \tag{7.4}$$

is realized along a straight-line by the projective translation $\mathbf{H}_T$,

$$\boldsymbol{A}_c^* = \mathbf{H}_T\boldsymbol{A}_c \tag{7.5}$$

the residual displacement is necessarily a projective rotation about the initial position of the center $\boldsymbol{A}_c$.

$$\mathbf{H}_R = \mathbf{H}_T^{-1}\mathbf{H}_{RT}. \tag{7.6}$$

Clearly, the choice of the center affects the decomposition that is obtained, including the rotation (Fig. 7.3). The order of the homographies in equation (7.3) defines $\mathbf{H}_R$ to rotate around the initial position of the center. It would be around the final position if the order were reversed.

For (7.6) to be applied, a translation homography must still be calculated beforehand, which is not as trivial in the projective case as in the Euclidean one. Any projective translation $\mathbf{H}_T$ can be expressed (7.7) in terms of the outer product of the plane at infinity $\boldsymbol{a}_\infty^\top$ and a 4-vector $\boldsymbol{k}$ which represents "direction-and-distance'.

$$\mathbf{H}_T = \mathbf{I} + \boldsymbol{k}\cdot\boldsymbol{a}_\infty^\top. \tag{7.7}$$

It can be determined from the motion of one point, and $\boldsymbol{a}_\infty^\top$ is determined by the eigenvector of the homography $\mathbf{H}_{RT}^{-\top}$ dual to a projective displacement $\mathbf{H}_{RT}$ of general type. From (7.5) it follows that $\boldsymbol{k}$ has the form

$$\boldsymbol{k} = \frac{\boldsymbol{A}_c^* - \boldsymbol{A}_c}{\boldsymbol{a}_\infty^\top\boldsymbol{A}_c}. \tag{7.8}$$

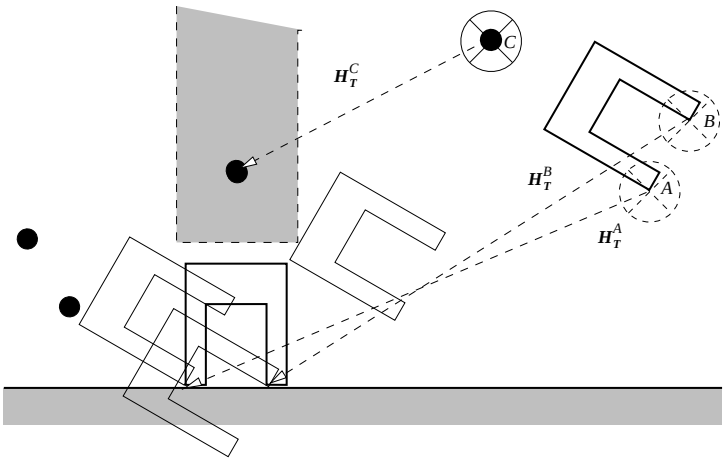

Figure 7.3: The choice of center-point $\boldsymbol{A}$, $\boldsymbol{B}$, or $\boldsymbol{C}$ for the Cartesian motion determines the decomposition into rotational and translational components.

Actually, the main interest is in generating trajectories that drive these two motions simultaneously rather than sequentially, as explained in section 7.2.2. It also seems worthwhile to note that a projective displacement, which is ultimately not tied to a particular object, allows trajectories to be generated for an arbitrary rigid structure, even if the latter was not involved in defining the displacement homography. For instance, a desired projective displacement of an object allows to generate also the corresponding trajectory for a rigid grasp of a tool (Fig. 7.6). The rigidity assumption allows in fact to generate the projective motion from a small number point-features, which could not define a displacement homography themself (Fig. 7.6).

7.2.2 Generation of Cartesian motions

Now, write $\mathbf{H}_R(\theta)$ as function of an angle θ, and $\mathbf{H}_T(\tau)$ as function of a distance τ, and let θ^* be the total angle, $\tau^* = 1$ the total distance traversed. This implies that the rotation has a fixed axis direction. Relating these variables and (3.21), (4.27) to a common parameter σ, the family of Cartesian trajectories around $\boldsymbol{A}_c$ becomes

$$\mathbf{H}_C(\sigma) = \mathbf{H}_T\left(\tau(\sigma)\right) \cdot \mathbf{H}_R\left(\theta(\sigma)\right), \tag{7.9}$$

$$\tau(\sigma) = \sigma\tau^*, \quad \theta(\sigma) = \sigma\theta^*, \quad 0 < \sigma < 1. \tag{7.10}$$

Intuitively, the compound motion in σ has the desired characteristics because $\mathbf{H}_R(\sigma)$ affects only orientation and leaves $\boldsymbol{A}_c$ invariant, while the latter is translated by $\mathbf{H}_T(\sigma)$.

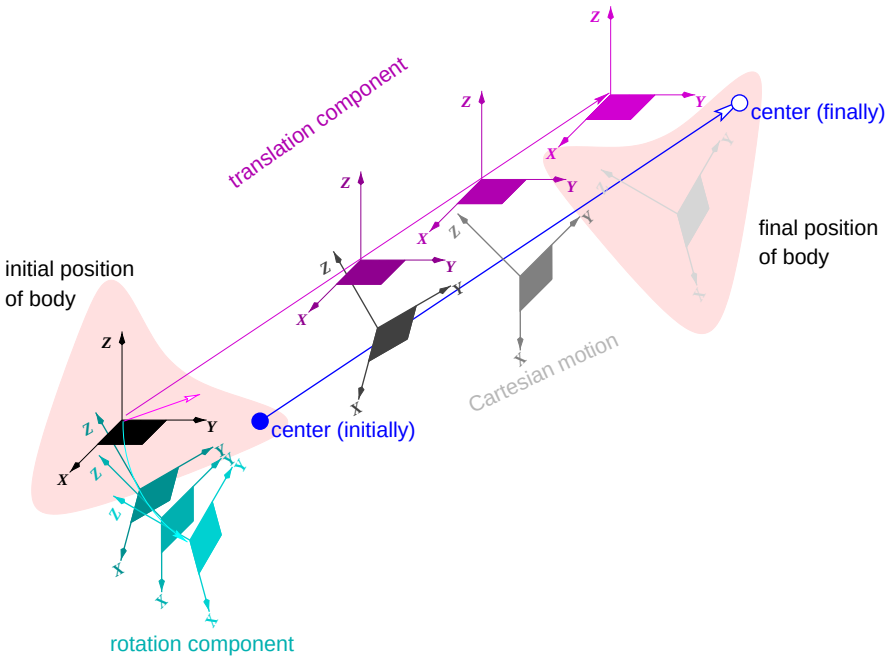

Figure 7.4: Cartesian motion with two components.

Spatial velocity of Cartesian motion

The derivative $\dot{\mathbf{H}}_C$ of the trajectory function with respect to σ allows a differential characterization of Cartesian trajectories in terms of their tangent operators $\hat{\mathbf{H}}_C$ of a Cartesian motion:

$$\dot{\mathbf{H}}_C = \dot{\mathbf{H}}_T\mathbf{H}_R + \mathbf{H}_T\dot{\mathbf{H}}_R \tag{7.11}$$

$$\begin{aligned}\hat{\mathbf{H}}_C(\sigma) &= \dot{\mathbf{H}}_C\mathbf{H}_C^{-1} \\ &= \left(\dot{\mathbf{H}}_T\mathbf{H}_R + \mathbf{H}_T\dot{\mathbf{H}}_R\right)\mathbf{H}_R^{-1}\mathbf{H}_T^{-1}, \\ &= \dot{\mathbf{H}}_T\mathbf{H}_T^{-1} + \mathbf{H}_T\dot{\mathbf{H}}_R\mathbf{H}_R^{-1}\mathbf{H}_T^{-1}, \\ &= \hat{\mathbf{H}}_T + \mathbf{H}_T(\sigma)\hat{\mathbf{H}}_R\mathbf{H}_T^{-1}(\sigma).\end{aligned} \tag{7.12, 7.13}$$

The operator $\hat{\mathbf{H}}_C$ reflects the geometry of the Cartesian motion: the translation has a fixed direction, as well as the rotation axis. The latter is at each instant σ passing through the center-point (Fig. 7.5), i.e. it is continuously translated by the center's motion $\mathbf{H}_T(\sigma)\boldsymbol{A}_c$ (7.13). This becomes more evident when the tangent operator is applied to the moving center-point $\boldsymbol{A}_c(\sigma)$. First of all, $\boldsymbol{A}_c(\sigma) = \mathbf{H}_C(\sigma)\boldsymbol{A}_c = \mathbf{H}_T(\sigma)\boldsymbol{A}_c$, since the latter is invariant under the rotational component. Secondly, at an arbitrary location σ along the trajectory, applying the tangent operator to $\boldsymbol{A}_c(\sigma)$ yields the translation component (7.15), only, since $\hat{\mathbf{H}}_R\boldsymbol{A}_c$ equals zero. This illustrates that the center is driven at any instant along one and the same pure

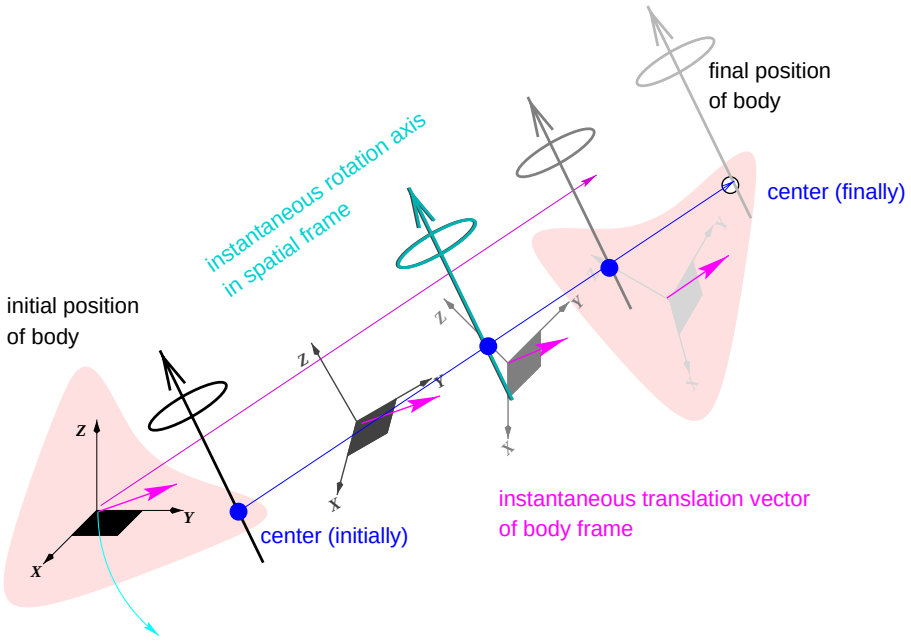

Figure 7.5: Cartesian motion characterized by its instantaneous velocity. Not the motion of the spatial rotation axis along the translation, and the fact that the direction of translation w.r.t. the body frame changes at each instant.

translational motion (Fig. 7.5).

$$\begin{aligned}\dot{\boldsymbol{A}}_c &= \hat{\mathbf{H}}_C \boldsymbol{A}_c(\sigma) && (7.14)\\ &= \left(\hat{\mathbf{H}}_T + \mathbf{H}_T(\sigma)\hat{\mathbf{H}}_R\mathbf{H}_T^{-1}(\sigma)\right)\mathbf{H}_T(\sigma)\,\boldsymbol{A}_c,\\ &= \hat{\mathbf{H}}_T\boldsymbol{A}_c + \mathbf{H}_T(\sigma)\hat{\mathbf{H}}_R\boldsymbol{A}_c,\\ &= \hat{\mathbf{H}}_T\boldsymbol{A}_c, && (7.15)\end{aligned}$$

Body velocity of Cartesian motion

The moving frame's velocity operator is another way to illustrate the nature of Cartesian motion. It is derived as follows:

$$\begin{aligned}\check{\mathbf{H}}_C(\sigma) &= \mathbf{H}_C^{-1}\dot{\mathbf{H}}_C && (7.16)\\ &= \mathbf{H}_R^{-1}\mathbf{H}_T^{-1}\left(\dot{\mathbf{H}}_T\mathbf{H}_R + \mathbf{H}_T\dot{\mathbf{H}}_R\right)\\ &= \mathbf{H}_R^{-1}\mathbf{H}_T^{-1}\dot{\mathbf{H}}_T\mathbf{H}_R + \mathbf{H}_R^{-1}\mathbf{H}_T^{-1}\mathbf{H}_T\dot{\mathbf{H}}_R\\ \check{\mathbf{H}}_C(\sigma) &= \mathbf{H}_R^{-1}(\sigma)\check{\mathbf{H}}_T\mathbf{H}_R(\sigma) + \mathbf{H}_R^{-1}\dot{\mathbf{H}}_R, && (7.17)\end{aligned}$$

The operator $\check{\mathbf{H}}_c$ reflects the characteristics of Cartesian motions as seen from the moving frame, i.e. by an observer on the moving object. The rota-

tion axis is fixed in the moving frame, whereas the direction of translation, which was in the spatial frame, has to be adapted continuously to account for the varying orientation of the moving frame (Fig. 7.5).

It is important to realize that the homographies of a Cartesian motion form a $1D$ family but in general not a group. To see this, consider the inverse motion $\mathbf{H}_C^{-1} = \mathbf{H}_R^{-1}\mathbf{H}_T^{-1}$, and verify that equation (7.9) fails to describe the inverse, since the conjugation of $\mathbf{H}_R$ with $\mathbf{H}_T$ results in a displaced rotation axis.

$$\mathbf{H}_C^{-1} = \mathbf{H}_R^{-1}\mathbf{H}_T^{-1} = \mathbf{H}_T^{-1}(\mathbf{H}_T\mathbf{H}_R^{-1}\mathbf{H}_T^{-1})$$

7.2.3 Translation-first and translation-last motions

The above scheme (7.9) using the common abscissa σ defines a Cartesian motion with equal weighting of translation and rotation. However, any sort of combination between the two components can be expressed in terms of continuous functions $\theta(\sigma), \tau(\sigma)$ – preferably differentiable and monotonic – respecting the boundary condition (7.10). In the extreme case, they can be translation-first or translation-last trajectories (Fig. 7.6), where the first is useful for turn-a-key tasks, for instance, whereas the latter is useful for a grasping or insertion task.

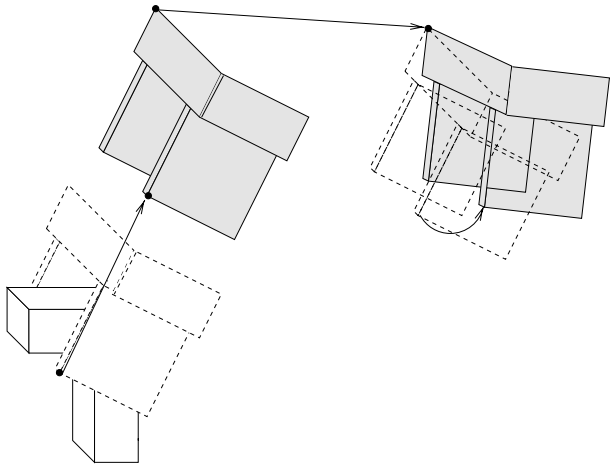

Figure 7.6: Trajectory for a grasp with rotation-first characteristics. The final translation can be constructed in a point-to-point manner based on the edges of the gripper's claw.

7.3 Trajectories of primitive motions from mobility constraints

The above *R-T* decomposition of projective displacements is an example of how a general motion relates to a pair of primitive motions. In particular, the construction of the translation (7.8) shows how a minimal amount of geometric information, here the motion of one point and the infinity plane, allows primitive motions to be determined from task-space information. In both cases, however, a rigid motion $\mathbf{H}_{RT}$, and the respective locus of the plane at infinity are required a-priori.

In contrast, the differential characterization of Cartesian trajectories developed above suggests already that differential information should also allow rigid and robot motions to be constrained, e.g. from the differential motion or "*mobility*" of some well-chosen geometric features, such as the center.

On the one hand, the mobility of points and planes can be constrained in projective three-space in order to obtain a number of primitive projective motions of a rigid structure. On the other hand, the mobility of this structure, which is moving under action of the actuated mechanism, is equally constrained by the Jacobian of projective articulated motion. Then, relating the two of them allows the constraint-equations to be solved in an explicit representation of the primitive motions. This representation is either a joint-screw $\hat{\boldsymbol{q}}$, or a projective screw in form of the operator $\hat{\mathbf{H}}$.

In addition, formulating the problem in the visual domain allows the constraints to reflect the geometry of a given task, and also visibility conditions (section 7.4.2) as well as mechanically feasible trajectories (section 7.4.3) to be described (section 7.4.1). Moreover, it allows the joint-velocities that drive and control the robot along the trajectories to be computed directly from the observations (section 8.5).

7.3.1 Translation along an axis

Suppose that the direction of translation is given in terms of an axis through two points $\boldsymbol{A}_1, \boldsymbol{A}_2$. Their dual is a pencil of planes spanned by a general pair of planes $\boldsymbol{a}_1^\top, \boldsymbol{a}_2^\top$ that intersect in the axis (2.59). A rigid motion that maps each plane into itself is a pure translation along the given axis, and it is the only such rigid motion (Fig. 7.7). The projective kinematic model, here the plane-operators $-\hat{\mathbf{H}}_j^\top$ for each joint j, characterizes all possible rigid motions and their plane-velocities, so the translation corresponds to the only joint-space motion $\hat{\boldsymbol{q}}_T$ for which both plane-velocities vanish. Formally, this requires $\hat{\boldsymbol{q}}_T = \dot{\boldsymbol{q}}$ to be in the kernel of (7.18), where ν is an arbitrary scalar. The generator $\hat{\mathbf{H}}_T$ of the corresponding one-dimensional group of projective

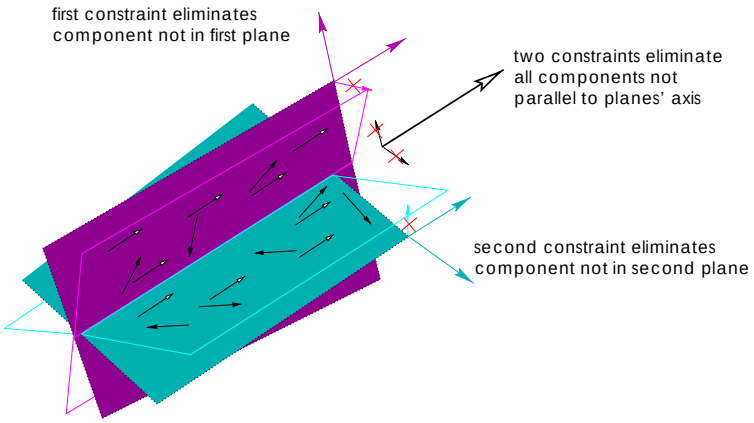

Figure 7.7: Two plane-velocities constrained to zero eliminate one velocity component (crossed out), each, such that the two resolve into a translation along the planes' common axis.

translations is a linear combination in this $\hat{\boldsymbol{q}}_T$:

$$\nu\ \hat{\boldsymbol{q}}_T = \ker \begin{bmatrix} -\hat{\mathbf{H}}_1^\top \boldsymbol{a}_1, \ \ldots, \ -\hat{\mathbf{H}}_6^\top \boldsymbol{a}_1 \\ -\hat{\mathbf{H}}_1^\top \boldsymbol{a}_2, \ \ldots, \ -\hat{\mathbf{H}}_6^\top \boldsymbol{a}_2 \end{bmatrix}_{8\times 6}, \qquad \hat{\mathbf{H}}_T = \hat{\boldsymbol{q}}_{T1}\hat{\mathbf{H}}_1 + \cdots + \hat{\boldsymbol{q}}_{T6}\hat{\mathbf{H}}_6. \tag{7.18}$$

7.3.2 Revolution around an axis

Suppose two points $\boldsymbol{A}_1, \boldsymbol{A}_2$ defining an axis and the point-operators $\hat{\mathbf{H}}_j$ for each joint j. Among all rigid motions, here expressed as joint-space motions, the ones $\hat{\boldsymbol{q}}_R$ for which the velocities of both points vanish are rotations $\hat{\mathbf{H}}_R$ around the axis connecting $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ (Fig. 7.8). This is formalized by the kernel in (7.19), with ν being a free scalar. The corresponding one-dimensional projective rotation group has the generator $\hat{\mathbf{H}}_R$ (7.19), which follows directly from the kernel's joint-space motion $\hat{\boldsymbol{q}}_R$,

$$\nu\ \hat{\boldsymbol{q}}_R = \ker \begin{bmatrix} \hat{\mathbf{H}}_1\boldsymbol{A}_1, \ \ldots, \ \hat{\mathbf{H}}_6\boldsymbol{A}_1, \\ \hat{\mathbf{H}}_1\boldsymbol{A}_2, \ \ldots, \ \hat{\mathbf{H}}_6\boldsymbol{A}_2 \end{bmatrix}_{8\times 6}, \qquad \hat{\mathbf{H}}_R = (\hat{\boldsymbol{q}}_{R1}\hat{\mathbf{H}}_1 + \cdots + \hat{\boldsymbol{q}}_{R6}\hat{\mathbf{H}}_6), \tag{7.19}$$

where $\hat{\mathbf{H}}_R$ is normalized to have eigenvalues $i, -i$, such that ν measures angles in radians.

7.3.3 Revolution around a point in a plane

Another way for constraining a transformation to a rotation is as follows: suppose the transformations to act on a given plane $\boldsymbol{a}_1^\top$ as a "planar" rotations, i.e. the plane turns "in-place", and suppose in addition that the point

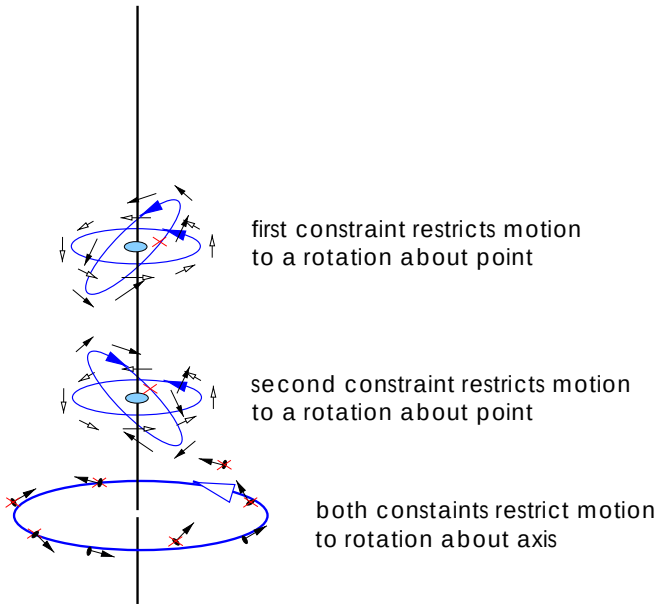

Figure 7.8: Two point-velocities constrained to zero eliminate one rotational velocity-component (crossed out), each, such that the two resolve to a rotation around the points' common axis.

$\boldsymbol{A}_1$ is on the axis (Fig. 7.9). The axis then is perpendicular to the plane and goes through the given point. Determine among all joint-space motion – among all mechanically feasible motions – the one $\hat{\boldsymbol{q}}_P$ for which both, the point-velocity of $\boldsymbol{A}_1$ and the plane-velocity of $\boldsymbol{a}_1^\top$ vanish. The point-operators $\hat{\mathbf{H}}_j$ and the plane-operators $-\hat{\mathbf{H}}_j^\top$ of the projective kinematics allow this to be formally express as the kernel in (7.20). The resulting joint-space motion $\hat{\boldsymbol{q}}_P$ allows an operator $\hat{\mathbf{H}}_P$ that generates the corresponding one-dimensional projective rotation group to be synthesized:

$$\nu\ \hat{\boldsymbol{q}}_P = \ker \begin{bmatrix} -\hat{\mathbf{H}}_1^\top \boldsymbol{a}_1, & \ldots, & -\hat{\mathbf{H}}_6^\top \boldsymbol{a}_1 \\ \hat{\mathbf{H}}_1 \boldsymbol{A}_1, & \ldots, & \hat{\mathbf{H}}_6 \boldsymbol{A}_1 \end{bmatrix}_{8\times 6}, \qquad \hat{\mathbf{H}}_P = (\hat{\boldsymbol{q}}_{P1}\hat{\mathbf{H}}_1 + \cdots + \hat{\boldsymbol{q}}_{P6}\hat{\mathbf{H}}_6), \tag{7.20}$$

Again, $\hat{\mathbf{H}}_p$ is normalized to have eigenvalues $i, -i$, so that ν measures angles in radians.

The above postulated equivalence between joint-space motions and rigid (projective) motions holds only when the manipulator configuration is fully actuated. For an under-actuated configuration, the kernel becomes empty if the projective motion requires a missing degree-of-freedom. Conversely, at kinematic singularities, the kernel may have a higher dimension as it contains internal fixing motions that fix all end-effector points and planes, as well as possibly the desired projective motion itself. These cases can be

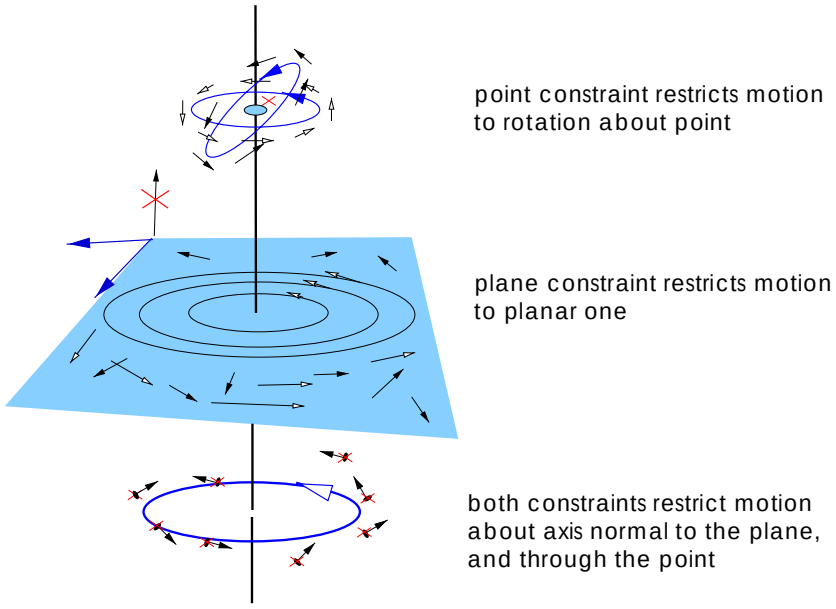

Figure 7.9: A point- and a plane-velocity constrained to zero eliminate a translation and a rotational velocity-component (crossed out), each, such that the two resolve into a rotation in a plane about the given point.

detected directly from the rank of the stacked 8×6 constraint-matrices in (7.18), (7.19), (7.20). So, either there may be at an instant zero, one or many joint-space motions corresponding to the demanded motion. This explicit projective relationship between robot singularities and the mobility of rigid structures can be highly useful for singularity avoidance in visual servoing.

7.4 Trajectories based on three points

In this section, the aim is to augment the displacement between the initial and target positions of only three 3D-points $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3$ to a continuous reaching trajectory for these points. These trajectories are moreover constructed to be mechanically and visually feasible in the sense of rigid motion and back-face culling. Note that although three points in general suffice to define a unique rigid displacement, they fail to define a unique projective one, which requires a homography of five points. This suggests that section 7.2 cannot be applied in this case.

To overcome this, the underlying but unknown projective displacement is decomposed into three primitive motions: a translation and two rotations (Fig. 7.10), using the results of section 7.4.1. Although the decomposition is formulated as a sequence of three primitive motions, they are constructed in a way that allows them to be driven simultaneously, thus ensuring that the

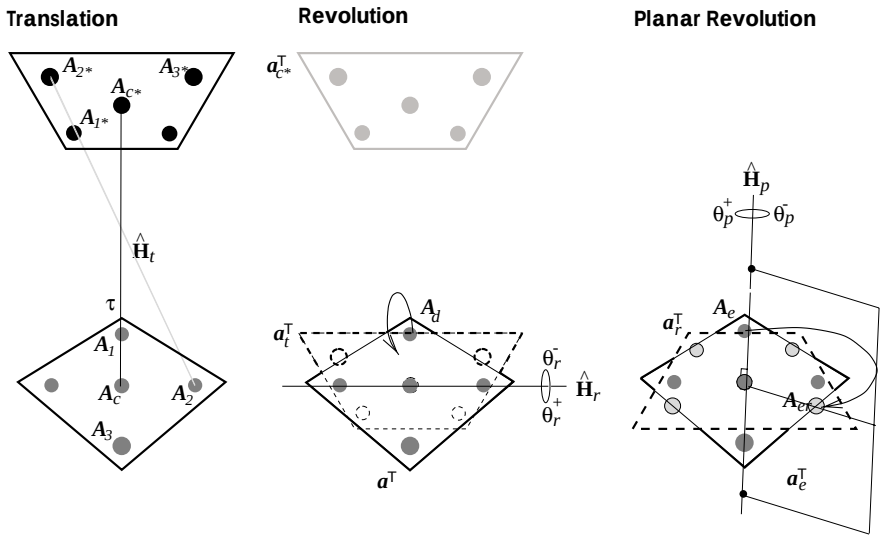

Figure 7.10: Three partitions of a task.

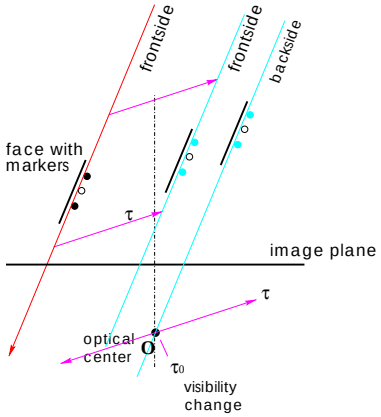

Figure 7.11: Translation to τ_0 where visibility changes.

desired trajectories emerge. These so-called "*Cartesian motions*" consist of straight-line trajectories of a center point with superposed rotations that are at any instant about this center (section 7.4.3). The superposition is possible because "subsequent" component motions do not affect the "alignments" achieved by the preceding ones. For instance, as both rotations are about the center point, the translation of this point remains unaffected. Similarly, as the final rotation axis is perpendicular to the face spanned by the three points, the plane containing the face remains invariant.

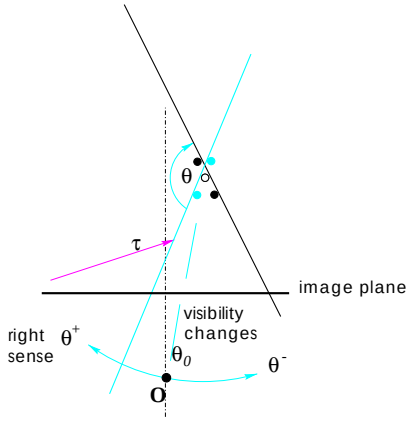

Figure 7.12: Rotation of the face in the sense which preserves visibility.

7.4.1 Decomposition

It is now shown how three points allow constraints on primitive motions (section 7.3) to be computed, and how they decompose the points' projective motion. The computational geometry detailed below eliminates step by step the various components of the task. In the first instance, the latter is given in terms of the initial and final positions, $\boldsymbol{A}$ and $\boldsymbol{A}_*$, of three points on a face of a rigid object. The face also defines a plane with initial and target positions, $\boldsymbol{a}^\top$ and $\boldsymbol{a}_*^\top$. The result of the decomposition is three primitive motions: a translation of the center, a hinge of the face onto the target plane, and a rotation within the target plane. These motions are obtained as projective operators $\hat{\mathbf{H}}_t$, $\hat{\mathbf{H}}_r$, $\hat{\mathbf{H}}_p$ represented with respect to the current spatial configuration of the task's components. Joint-screws $\hat{\boldsymbol{q}}_t$, $\hat{\boldsymbol{q}}_r$,$\hat{\boldsymbol{q}}_p$ can also be obtained directly with respect to the configuration of the actuator.

Translation

The **first component (translation of center)** (Fig. 7.10, left) chooses one marker or (less projectively) the marker's midpoint as a center point $\boldsymbol{A}_c$, and decomposes the task into a translation $\mathbf{H}_t$, modulo $\mathbf{H}_s$, a rotation of the face around the center. The translation operator $\hat{\mathbf{H}}_t$ is obtained by applying (7.18) to the center's current and target position, $\boldsymbol{A}_c$ and $\boldsymbol{A}_c^*$. The translation amplitude τ amounts to a scalar "distance-to-target". It is obtained by solving (7.21) for τ, which corresponds to intersecting $\boldsymbol{A}_c$'s straight-line path with a transversal plane ${\boldsymbol{a}^\top}_c^*$ in $\boldsymbol{A}_c^*$ (Fig. 7.10, left).

$$\boldsymbol{a}_c^{*\top}\left(\mathbf{I}+\tau\,\hat{\mathbf{H}}_t\right)\boldsymbol{A}_c=0,\qquad \tau=-\frac{\boldsymbol{a}_c^{*\top}\boldsymbol{A}_c}{\boldsymbol{a}_c^{*\top}\hat{\mathbf{H}}_t\boldsymbol{A}_c}. \tag{7.21}$$

Given $\hat{\mathbf{H}}_t$, the translational component can be eliminated from the task by applying the projective translation $\mathbf{H}_t(\tau) = \mathbf{I} + \tau\hat{\mathbf{H}}_t$ "backwards" onto the target primitives, which will be then subscripted with $_t$ to indicate this. (Fig. 7.10, middle, dashed)

$$\boldsymbol{A}_{it} = \mathbf{H}_t(-\tau)\boldsymbol{A}_{i*}, \qquad \boldsymbol{a}_t = \mathbf{H}_t^{-\top}(-\tau)\boldsymbol{a}_* \ . \tag{7.22}$$

So, the residual task can be written in terms of a new set of target primitives, $\boldsymbol{A}_{it}$ and $\boldsymbol{a}_t^\top$, with the translational component now being eliminated. The elimination is done "backwards" so that the residual task will be expressed around the current position of the center and other points. Similarly, the joint-space representation $\hat{\boldsymbol{q}}_t$ of the translation, as arising in (7.18), is expressed with respect to the current configuration of the robot. Hence, for the joint screw $\hat{\boldsymbol{q}}_t$ to be a valid joint-velocity at any instant, it is crucial to use in (7.18) the operators $\hat{\mathbf{H}}_j|_{\boldsymbol{Q}}$ (5.21). Then, the velocities can be sent immediately to a robot controller, as done in section 8.5.

Revolution

The **second component (hinge between two planes)**, (Fig. 7.10, center) is the rotation $\mathbf{H}_r$ around the axis between two planes: the initial face and the translated target face, $\boldsymbol{a}^\top$ and $\boldsymbol{a}_t^\top$. In this way, the rotational part $\mathbf{H}_s$ of the task is split into two, a rotation $\mathbf{H}_r(\theta_r)$ onto the target plane modulo a residual rotation $\mathbf{H}_p$ within this plane (Fig. 7.10 right, dashed).

The corresponding operator $\hat{\mathbf{H}}_r$ is obtained by applying (7.19) to a pair of arbitrary points on the intersection axis of the planes. Based on this, an "angle-to-target" θ_r can be defined using the intersection of the new target plane $\boldsymbol{a}_t^\top$ with the circular path of a point $\boldsymbol{A}_d$ on the face (Fig. 6.2).

$$\boldsymbol{a}_t^\top \left(\mathbf{I} + \sin\theta_r\hat{\mathbf{H}}_r + (1 - cos\theta_r)\hat{\mathbf{H}}_r^2\right)\boldsymbol{A}_d = 0 \tag{7.23}$$

This amounts to solving (7.23) for θ_r, which has already been detailed in section 6.2. There is a two-fold ambiguity in the sense of rotation. In practice, the solution that turns the back-face onto the front-face is eliminated using a simple visibility argument (section 7.4.2).

As for the translation, the rotation is eliminated by applying a projective rotation $\mathbf{H}_r(-\theta_r)$ backwards to the $_t$-primitives, to give a new set of target primitives which are subscripted with $_r$, to indicate this:

$$\boldsymbol{A}_{ir} = \mathbf{H}_r(-\theta_r)\boldsymbol{A}_{it}, \qquad \boldsymbol{a}_r = \mathbf{H}_r^{-\top}(-\theta_r)\boldsymbol{a}_t \tag{7.24}$$

Planar rotation

The **third component (rotation within plane)** (Fig. 7.10, right) is a planar rotation $\mathbf{H}_p(\theta_p)$ of the face about the center $\boldsymbol{A}_c$ in order to finally

move the points $\boldsymbol{A}_i$ onto the previously rotated target positions $\boldsymbol{A}_{ir}$. The corresponding projective operator $\hat{\mathbf{H}}_p$ is obtained by applying (7.20) to $\boldsymbol{a}^\top$ and $\boldsymbol{A}_c$. The angle θ_p requires intersecting the circular path of a point $\boldsymbol{A}_e \in \boldsymbol{A}_i$ with a transversal plane $\boldsymbol{a}_e^\top$ chosen to go through the corresponding target point $\boldsymbol{A}_{er}$. The calculations are analogous to (7.23).

7.4.2 Visibility

In classical projective geometry, homogeneous coordinates are "unoriented", in contrast to oriented projective geometry. For example, homogeneous plane coordinates such as $\boldsymbol{a}^\top$ lack a distinction between the front- and the back-side, based on a signed normal direction. This makes classical backface-culling impossible as the corresponding notion of visibility has not been encoded. However, as projective displacements preserve the scalar ϕ on a plane orbit (section 2.4.3) – and consequently also the sign of the projective coordinates of both, points and planes – it is at least decidable whether a projective displacement has altered the sign of the dot product between a plane and a point. This actually would indicate that the point has changed sides with respect to the plane, or vice versa. In particular, applying this to the face-plane $\boldsymbol{a}^\top$ and the optical center $\boldsymbol{O}$ amounts to detecting changes in the visibility of the face from the sign of its product with the optical center, $\boldsymbol{a}^\top \cdot \boldsymbol{O}$. More precisely, for the one-parameter motions introduced above, such events are characterized by the amplitude τ_0 (Fig. 7.11), or by the pair of angles θ_0^+, θ_0^- (Fig. 7.12), that are obtained by applying equations (7.21) or (7.23) to the optical center.

Concerning the three components of the trajectories, this means the following:

- If the translation is towards and beyond τ_0, a respective reorientation of the face is required before a feed-forward along the trajectory reaches τ_0 (Fig. 7.11).

- If the rotation either leaves or enters the interval $\left[\theta_0^-, \theta_0^+\right]$, the visibility changes and a corresponding translation of the face is required before the feed-forward reaches the required θ_0 (Fig. 7.12).

- This concerns only the rotation $\hat{\mathbf{H}}_r$, but not the rotation $\hat{\mathbf{H}}_p$, since visibility remains unaltered under planar rotations of the face.

- Additionally, heavy use of such "side-of-plane" reasoning is made in the implementation for determining the "right" sense of the rotation θ_r (**??**). For instance, it is used to avoid the back-face being turned towards the camera (Fig. 7.12) or to prevent that the face is moved backside-up onto the target position (Fig. 7.10).

- In presence of a second camera, the above arguments are independently to both of the optical centers, and the most conservative among the left and right thresholds is taken. The field-of-view problem is not covered by the above, but is straight-forward. It basically amounts to intersecting the trajectories with two viewing cones. Interestingly, besides the planar rotation of the face, only translations along the optical axis can be considered, under some restrictions, to be a motion that leaves visibility unaffected. Although inside the viewing cone, such a translation can hardly traverse the optical center without destroying a camera, the optical axis of a stereo camera pair, as they are generally non parallel, prevent this motion criterion to be exploitable in practice.

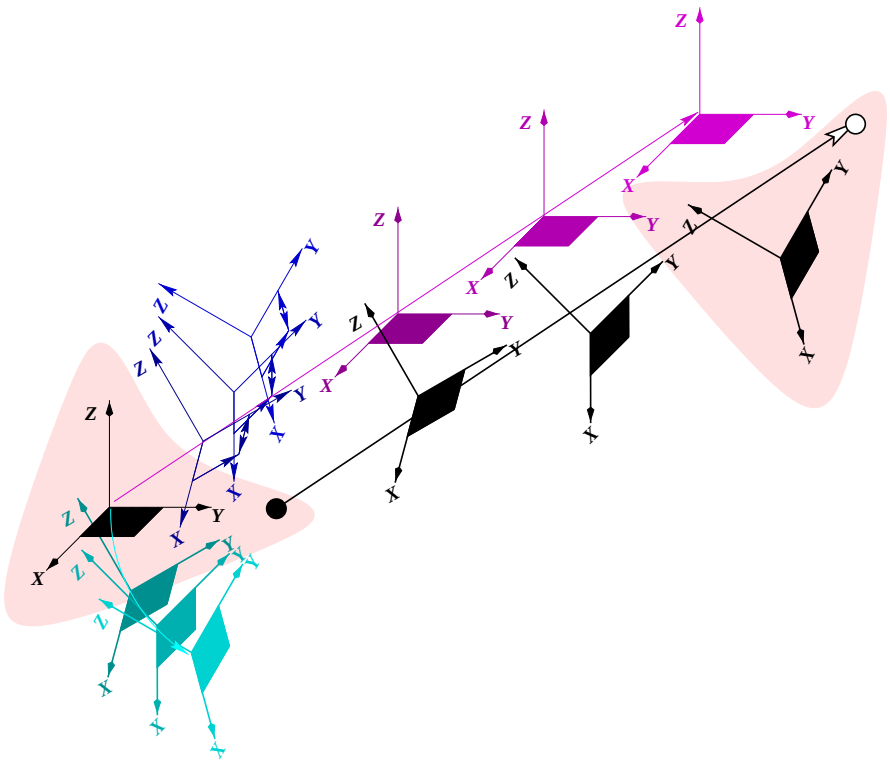

Figure 7.13: Cartesian motion of three components.

7.4.3 Generation of Cartesian trajectories

Now, the expression for the family of Cartesian trajectories $\mathbf{H}_c(\sigma)$ are given. They allow the three components of the task to be executed independently (Fig. 8.15). These are three scalar functions $\mu_t(\sigma)$, $\mu_r(\sigma)$, $\mu_p(\sigma)$ with a common abscissa σ (7.26) that allow the characteristics of the trajectories

to be modified to incorporate constraints such as visibility or reachability. In the same way as for the product-of-exponentials (5.18), the 1-dof primitive motions have to be multiplied in the "reverse" order (7.25) for the desired trajectories to emerge. Intuitively, the translation must be the left-most one, as the rotations must not affect either direction or the position of the translating center. The hinge is the second one, as the planar rotation must not affect the face as a plane, etc:

$$\begin{aligned}\mathbf{H}_c(\sigma) &= \mathbf{H}_c\left(\tau(\sigma), \theta_r(\sigma), \theta_p(\sigma)\right) \\ &= \exp(\tau(\sigma)\hat{\mathbf{H}}_t)\ \exp(\theta_r(\sigma)\hat{\mathbf{H}}_r)\ \exp(\theta_p(\sigma)\hat{\mathbf{H}}_p), \qquad (7.25)\\ \text{where } \tau(\sigma) &= \mu_t(\sigma)\tau, \qquad \theta_r(\sigma) = \mu_r(\sigma)\theta_r, \qquad \theta_p(\sigma) = \mu_p(\sigma)\theta_p. \qquad (7.26)\end{aligned}$$

The various μ functions should be monotonically growing functions $[0, t_*] \to [0, 1]$ subject to visibility constraints between μ_r and μ_t. For the center-point, these are:

- if $\mathbf{H}_r^{-\top}(\theta_r(\sigma))\boldsymbol{a}$ is visible then $\mu_t(\sigma) < \tau_0$, otherwise $\mu_t(\sigma) > \tau_0$.

- if $\mathbf{H}_t^{-\top}(\tau(\sigma))\boldsymbol{a}_*$ is visible then $\mu_r(\sigma) \in [\theta_0^-, \theta_0^+]$, otherwise $\mu_r(\sigma) \notin [\theta_0^-, \theta_0^+]$.

- $\mu_p(\sigma)$ is always unconstrained.

Either of these cases can be used to drive a feed-forward control either in τ or in θ_r while constraining the other one correspondingly. Further modifications of the overall behavior are possible. For instance, a linear decay of time-to-goal arises for $\mu(\sigma) = \sigma$, whereas an exponential decay like as in classical feed-back loops arises for $\mu(\sigma) = 1 - \exp(-\frac{\sigma}{t_*})$. An initial very flat plateau of the respective μ-function allows trajectories like rotation-first, translation-first, planar-first to be implemented.

Although feed-back laws for robot guidance and visual trajectory tracking are now feasible, chapter 8 will further exploit the results established in this chapter. In particular, the direct relationships of velocities along Cartesian trajectories and primitive projective motions to the corresponding joint-space motions give rise in the end to directly computed control law (section 8.5).

Chapter 8

Projective Visual Control

In the seminal work [19], the visual servoing problem was thoroughly formalized in terms of the task function approach [64]. The latter can be understood as a mathematical formalism allowing a control law that implements a task to be derived if the corresponding mathematical function obeys a number of properties. This chapter adopts this approach to identify the various geometric variations of the task functions for visual servo control. Each of the proposed formulations relates the perception domain to the action domain, but they differ in the way these domains are represented. In particular, the state of the art has been extended by introducing projective representations of the action as well as the perception domain while at the same time assuring a sound analytical expression for their mutual interaction.

In the classical approaches, the perception domain is the image-plane, represented as image-points that are actually the projections of 3D scene-points. The action domain is a robot operating in velocity mode, and it is represented as a kinematic screw $(\boldsymbol{v}, \boldsymbol{w}) \in se(3)$ associated with the end-effector. Since it is points in three-space that are both observed and acted on, the presentation of this chapter is centered around three-dimensional point-spaces as the fundamental link between the two domains. Based on this, it formally develops the relationships ruling their mutual interaction. The first consists in image-based visual servoing based on a projective kinematic model. It is presented in great generality where the numerous varieties of this approach are shown as special cases of a coordinate-free formulation of the system. In particular, the equivalence of the hand-eye and independent-eye systems is shown. Similarly, a calibration-free formulation is possible, assuming nothing more than a rigid structure associated with the tool, while camera and robot configurations now may vary.

Second, the results of the trajectory chapter gave rise to a directly computed feed-forward for trajectory control of Cartesian motion. The computations essentially correspond to those done in trajectory generation, while

the degrees-of-freedom unconstrained by the trajectory are considered as a feedback error, which can then be used for instance to avoid singularities or constrained visibility. A numerical criterion on the Jacobian could allow to determine what degree-of-freedom the robot is actually able to drive, and what joints do have to be blocked.

The basic results are validated on a real visual servoing system, and the advanced techniques are tested against simulated data.

8.1 Perception domain

8.1.1 Single camera

The perception domain is most often represented as the image-plane of a video camera, and image-points as 2-vectors in pixel-coordinates $\boldsymbol{s} = (u, v)^\top$. Central to visual servoing is the relation between a 3D point $\boldsymbol{S} = (X, Y, Z)^\top$, its spatial velocity $\dot{\boldsymbol{S}}$, and the velocity $\dot{\boldsymbol{s}}$ of its projection in the image. For the moment, let's write this as an abstract analytical function G, that encapsulates the perspective projection model and its Jacobian matrix $\mathbf{J}_G$

$$G : \boldsymbol{S} \to \boldsymbol{s} = G(\boldsymbol{S}), \qquad \dot{\boldsymbol{s}} = \mathbf{J}_G \dot{\boldsymbol{S}}. \tag{8.1}$$

Specific forms of this Jacobian arise from the specific frames in which the three-space and the image-planes are represented. Along these lines, the classical case is as follows, where $\boldsymbol{s}$ is in pixel coordinates, $\boldsymbol{S}$ is in metric Euclidean coordinates, and the relation between the two is the camera intrinsic parameters $\mathbf{K}$.

$$\boldsymbol{s} = G(\boldsymbol{m}) = \begin{pmatrix} m_1/m_3 \\ m_2/m_3 \end{pmatrix}, \qquad \boldsymbol{m} = \mathbf{K} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{8.2}$$

$$\dot{\boldsymbol{s}} = \mathbf{J}_G\Big|_{\boldsymbol{m}} \cdot \dot{\boldsymbol{m}}, \qquad \dot{\boldsymbol{m}} = \mathbf{K} \cdot \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix}, \tag{8.3}$$

$$\mathbf{J}_G|_{\boldsymbol{m}} = \begin{bmatrix} \frac{1}{m_3} & 0 & -\frac{m_1}{m_3^2} \\ 0 & \frac{1}{m_3} & -\frac{m_2}{m_3^2} \end{bmatrix}_{2\times 3}. \tag{8.4}$$

Note that the Jacobian depends on the 3D information in $\boldsymbol{S}$, i.e. its depth.

$$\dot{\boldsymbol{s}} = \left[\mathbf{J}_G\Big|_{\mathbf{K}\boldsymbol{S}} \cdot \mathbf{K} \right]_{2\times 3} \cdot \dot{\boldsymbol{S}}. \tag{8.5}$$

In detail, the Jacobian matrix $\mathbf{J}_G$ is developed "around" a given 3-vector $\boldsymbol{m} = \mathbf{K}\boldsymbol{S}$, which gives the full 3D-coordinates of points with respect to the affine camera frame $\mathcal{A}$. It is equivalent to $\boldsymbol{S}$, since $\mathbf{K}$ is a regular 3×3 affinity (see section 2.1). Therefore, $\boldsymbol{m}$ has to be interpreted in this context as 3D-coordinates rather than as homogeneous coordinates in the image.

In order to approximate the depth of $\boldsymbol{S}$, a number of solutions are possible (see also section 8.1.2):

- In the "variable point" case, a varying estimate of $\boldsymbol{S}$ is used [39], e.g. resulting from reconstruction or pose.
- In the "target point" case, a constant value for the point's goal position $\boldsymbol{S}^*$ is substituted into the Jacobian.
- The "fixed depth" case is an elegant intermediate solution. There, the X and Y coordinates are deduced from the current image of $\boldsymbol{S}$ but a constant value for its depth Z^* is substituted into the Jacobian.

An equivalent formulation of the interaction between 3D- and 2D-motion can be based on homogeneous coordinates $\boldsymbol{M}$ in three-space, which also allows a projective ambient space to be represented. [1]

$$\boldsymbol{s} = G(\boldsymbol{m}), \qquad \boldsymbol{m} = \mathbf{P}\boldsymbol{M} \tag{8.6}$$

$$\dot{\boldsymbol{s}} = \mathbf{J}_G\Big|_{\boldsymbol{m}} \cdot \dot{\boldsymbol{m}}, \qquad \dot{\boldsymbol{m}} = \mathbf{P} \cdot \dot{\boldsymbol{M}} \tag{8.7}$$

$$\dot{\boldsymbol{s}} = \left[\mathbf{J}_G\Big|_{\mathbf{P}\boldsymbol{M}} \mathbf{P}\right]_{2\times 4} \cdot \dot{\boldsymbol{M}} \tag{8.8}$$

In case of a Euclidean ambient space, the coordinates of a point are set to $\boldsymbol{M} = (\boldsymbol{S}, 1)^\top$, and the camera matrix to $\mathbf{P} = [\mathbf{K}|\mathbf{o}]$. So, the 3-vector $\boldsymbol{m} = \mathbf{P}\boldsymbol{M}$ still encodes depth and can be considered as 3D affine, despite the cancellation produced by the fourth column of $\mathbf{P}$ being zero. In case of a projective ambient space, however, this is no longer the case since the fourth row of $\boldsymbol{M}$ is now non-trivial, and the projection onto a 3-vector $\mathbf{m} = \mathbf{P}\boldsymbol{M}$ no longer holds complete depth information.

8.1.2 Stereo camera pair

In case of a stereo camera, the second camera is represented by the projection matrix $\mathbf{P}'$, and points on the right image-plane are represented by 2-vectors $\boldsymbol{s}'$ or by points $\boldsymbol{m}'$ on a respective projective plane.

$$\boldsymbol{s}' = G'(\boldsymbol{m}'), \qquad \boldsymbol{m}' = \mathbf{P}'\boldsymbol{M}, \tag{8.9}$$

$$\dot{\boldsymbol{s}}' = \mathbf{J}_{G'}\Big|_{\boldsymbol{m}'} \cdot \dot{\boldsymbol{m}}', \qquad \dot{\boldsymbol{m}}' = \mathbf{P}'\dot{\boldsymbol{M}}, \tag{8.10}$$

$$\dot{\boldsymbol{s}}' = \left[\mathbf{J}_G\Big|_{\mathbf{P}'\boldsymbol{M}} \mathbf{P}'\right]_{2\times 4} \cdot \dot{\boldsymbol{M}}. \tag{8.11}$$

Regarding depth, the same can be said about right image-vectors, $\boldsymbol{s}'$ and $\boldsymbol{m}'$, as about left ones. In particular, a vector $\boldsymbol{m} = \boldsymbol{P}'\boldsymbol{M}$ holds depth

[1] To help intuition, let's assume homogeneous Euclidean coordinates. A similar but far more general formulation will be given later on.

information as long as $\boldsymbol{M}$ is denoted in a Euclidean or affine frame, but no longer if it is a general projective one.

In contrast, a stereo pair of such image-vectors almost always contains full depth information. It is most commonly written as a stacked 4-vector $(\boldsymbol{s}, \boldsymbol{s'})^\top$. This has to be interpreted as a measurement vector, in contrast to a pair of ideal image-points $\boldsymbol{m}$, $\boldsymbol{m'}$, that can be interpreted as algebraic variables obeying the epipolar constraint (2.10). In [47], the implications of imposing this constraint within a visual servoing loop are investigated.

$$\begin{pmatrix} \dot{\boldsymbol{s}} \\ \dot{\boldsymbol{s}}' \end{pmatrix} = \begin{bmatrix} \mathbf{J}_G \big|_{\mathbf{P}\boldsymbol{M}} & \mathbf{P} \\ \mathbf{J}_{G'} \big|_{\mathbf{P}'\boldsymbol{M}} & \mathbf{P}' \end{bmatrix} \dot{\boldsymbol{M}}. \tag{8.12}$$

8.1.3 3D triangulation device

A stereo camera together with a computational method for 3D-reconstruction can be viewed as a logical triangulation device. The "measurements" of this device are the 3D-coordinates of points observed in both images. Depending on the degree of a-priori calibration, the three-dimensional scene will be reconstructed in either a Euclidean, affine, or projective camera frame. In the first two cases, a representation of the measurements directly as 3-vectors is possible, $(X, Y, Z)^\top$ or $(U, V, W)^\top$. They can be extended trivially to homogeneous 4-vectors, $\boldsymbol{X}$ or $\boldsymbol{M}$, by adding a 1 in the fourth row.

In the projective case, they have to be extended, and the homogeneous coordinates $\boldsymbol{M}$ of points are non-trivial, i.e. finite points may have a 0 on the fourth row. In other words, the plane of points at infinity is a regular part of projective three-space, which furthermore cannot be identified in an uncalibrated setup. Therefore, it is not obvious whether a strictly 3-dimensional representation can be found for the projective case at all, since normalizing the coordinates on this manifold becomes more difficult. Barring unit 4-vectors $||\boldsymbol{M}|| = 1$, which still need more than a single patch in $\mathbb{R}^3$, normalizing the third row to one $\boldsymbol{M} = (u, v, 1, \rho)^\top$ seems a sensible choice, since the 3-vectors (u, v, ρ) form a single region in $\mathbb{R}^3$ that covers all points "visible" in the left image. The only points excluded are those on the left focal plane, which are indeed invisible.

Independent of the choice of normalization and embedding, the respective projections onto 3-vectors are most often non-linear, and the resulting coordinate-patches for the $\mathbb{P}^3$-manifold have a differential geometry that is rather difficult to handle. It is essentially the minimal representation of the perception and action domains that prevents their interaction from having an efficient linear form. The present formulations of projective kinematic models, in connection with ρ-orbits in $\mathbb{R}^4$ (see sections 5.2, 2.4.3), illustrate the feasibility and advantages of linear forms.

For these reasons, let's assume the output of the triangulation device to be represented, depending on the calibration present, either as trivial homogeneous 4-vectors, $\boldsymbol{X}$ or $\boldsymbol{N}$, with a homogeneous 1, or as unnormalized projective 4-vectors $\boldsymbol{M}_\rho$. Remember that ρ indicates that each such vector implicitly holds an unknown but well-defined scalar, its orbital height ρ (2.47).

8.2 Action domain

It is the actuator that characterizes the action domain and induces the corresponding representations. Here, it usually will be a robot manipulator with six fully actuated degrees-of-freedom: either a six-axis arm with a $6R$ structure, or a Cartesian robot with $3P3R$ structure[2]. Two different types of representation are possible; one based on the configuration of all of the joints, and one that abstracts this to a robot end-effector in terms of a posture in the Euclidean sense. The tangent space of each representation allow differential motions of the robot to be represented, either as "*physical velocities*" at which the robot is moving, or as "*screws*" that the robot is commanded to perform. This difference is also made in the notations either as $\dot{}$ or as $\hat{}$.

8.2.1 Kinematic screws

In both cases, a kinematic screw, and in particular the vector notation in terms of Cartesian velocities $(\boldsymbol{v}, \boldsymbol{w})^\top \in se(3)$ is an appropriate representation, which matches the actuator's 6 degrees of freedom. In the $3P3R$-case, the kinematic screw refers to the velocity of the hand frame which – owing to the Cartesian structure of the mechanism – is also the joint-velocity screw of the mechanism. In the $6R$-case, the kinematic screws still refer to the hand frame, and no longer to joint-velocities. In contrast, they serve as input to the Cartesian velocity mode of the robot controller, which instantaneously generates a corresponding joint-velocity screw for the low-level controllers to perform.

The addition of further linkages between robot, camera, and tool, be they mechanical or virtual, allows further kinematic screws to be defined. Each additional frame introduced by such a linkage gives rise to a corresponding representation of the action domain, referring in consequence to this new frame. Based on this idea, the action domains for various setups of visual servoing can be described and calculated by means of one and the same formal language.

[2]In robotics, the structure of a mechanism is often denoted with letters "P" for a prismatic joint, and "R" for a revolute joint, listed in base-to-tip order.

Camera screw

No matter what frame the actions are described in, the perception domain relates in the first instance to the frame of the cameras. Therefore, the interaction between perception and action is most naturally and most immediately formulated for 3D-points in the camera frame, and their image projections (8.14). This interaction is introduced by means of constraints between the linear velocity of a number of 3D points and the velocity screw of the camera, seen as a rigid body. Equation (8.14) states this for the arbitrary point $\boldsymbol{S} = (X, Y, Z)^\top$ in terms of a Jacobian relating its velocity $\dot{\boldsymbol{S}}$ to the camera's kinematic screw $(\boldsymbol{v}, \boldsymbol{w})^\top$. For a sufficient number of point-velocities $\dot{\boldsymbol{S}}_i$, the inversion of this relation allows the corresponding camera motion to be computed.

$$\dot{\boldsymbol{S}} = \mathbf{J}|_{\boldsymbol{S}} \cdot \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}, \tag{8.13}$$

$$\mathbf{J}|_{\boldsymbol{S}} = \begin{bmatrix} 1 & 0 & 0 & 0 & -X & Y \\ 0 & 1 & 0 & X & 0 & -Z \\ 0 & 0 & 1 & -Y & Z & 0 \end{bmatrix}, \tag{8.14}$$

$$\begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}_C = \begin{bmatrix} \mathbf{J}|_{\boldsymbol{S}_1} \\ \mathbf{J}|_{\boldsymbol{S}_2} \\ \mathbf{J}|_{\boldsymbol{S}_3} \\ \vdots \end{bmatrix}^+ \begin{pmatrix} \dot{\boldsymbol{S}}_1 \\ \dot{\boldsymbol{S}}_2 \\ \dot{\boldsymbol{S}}_3 \\ \vdots \end{pmatrix}. \tag{8.15}$$

Necessarily, the resulting screw refers to the same frame as the point-velocities, which in the first instance is the camera frame, denoted $(\boldsymbol{v}, \boldsymbol{w})^\top_C$.

Hand screw

The implementation of a robot controller always refers to a reference frame based on the kinematics of the mechanism, alone. For that purpose, a hand frame $\mathcal{H}$ is allocated onto the tip of the mechanism, e.g. onto the intersection of axes 4,5,6, which defines the origin of $\mathcal{H}$. A camera screw expressed in $\mathcal{C}$ has to be transferred into the controlled frame $\mathcal{H}$ before it can be used as a command. This transfer of screws is defined by the rigid linkage $\mathbf{T}_{HC} = (\mathbf{R}, \boldsymbol{t})$ relating camera and hand, or alternatively by the current state of the corresponding virtual link.

$$\boldsymbol{S}_H = \mathbf{R}\boldsymbol{S}_C + \boldsymbol{t}, \qquad \boldsymbol{X}_H = \mathbf{T}_{HC}\boldsymbol{X}_C.$$

This can be written for vectors $(\boldsymbol{v}, \boldsymbol{w})^\top_{C,H}$ as well as for the matrix representations $\hat{\mathbf{T}}_H, \hat{\mathbf{T}}_C$ of screws. In a projective ambient space only matrix operators are well-defined.

$$\begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}_H = \begin{bmatrix} \mathbf{R} & [\boldsymbol{t}]_\times \mathbf{R} \\ \mathbf{O} & \mathbf{R} \end{bmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}_C, \qquad \hat{\mathbf{T}}_H = \mathbf{T}_{HC} \cdot \hat{\mathbf{T}}_C \cdot \mathbf{T}_{HC}^{-1}. \tag{8.16}$$

For the classical eye-in-hand configuration, $\mathbf{T}_{HC}$ denotes the mechanical hand-eye link, and is hence constant. It can be directly combined with the camera Jacobian 8.14, as is done in the classical approach [19]. Equations 8.15 and 8.16 amount in fact to 3D-visual servoing, using a triangulation sensor.

$$\begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}_H = \begin{bmatrix} \mathbf{R} & [\boldsymbol{t}]_\times \mathbf{R} \\ \mathbf{O} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{J}|_{\boldsymbol{s}_1} \\ \mathbf{J}|_{\boldsymbol{s}_2} \\ \mathbf{J}|_{\boldsymbol{s}_3} \\ \vdots \end{bmatrix}^+ \begin{pmatrix} \dot{\boldsymbol{s}}_1 \\ \dot{\boldsymbol{s}}_2 \\ \dot{\boldsymbol{s}}_3 \\ \vdots \end{pmatrix}. \tag{8.17}$$

The calibration of this link has been treated exhaustively in literature [38], In contrast, the independent-eye configuration, where the robot's motion causes $\mathbf{T}_{HC}$ to vary, requires $\mathbf{T}_{HC}$ to be continuously updated, e.g. using an estimate of the current camera pose, since it is just a virtual link. A more elegant solution is to deal with a "base screw", as proposed below.

Tool screw

The variation of the camera pose due to robot motion is apparent only as varying pose of a tool rigidly mounted onto the hand, which itself cannot be physically observed. For a tool-frame $\mathcal{L}$, the current estimate of the tool-camera pose $\mathbf{T}_{LC}$ provides a virtual camera-tool link defining the current tool-screw $\hat{\mathbf{T}}_L$. The latter can be transferred into the hand-screw $\hat{\mathbf{T}}_H$ from the rigid tool-hand link $\mathbf{T}_{HL}$, which is constant, but which still has to be calibrated a-priori, e.g. using the hand-eye formalism.

$$\hat{\mathbf{T}}_H = \mathbf{T}_{HL} \left(\underbrace{\mathbf{T}_{LC}(t) \hat{\mathbf{T}}_C \mathbf{T}_{LC}^{-1}(t)}_{\hat{\mathbf{T}}_L} \right) \mathbf{T}_{HL}^{-1} \tag{8.18}$$

Obviously, it is due to the robot controller's need for a hand screw that this rather artificial and cumbersome relation between the camera screw and the action domain has to be established. A less redundant representation of the action domain is explained below.

Base screw, or "camera-centered" formulation

In the literature [42], a strict distinction is made between the independent-eye case (IE), where the robot and camera are independently but rigidly installed in the workspace, and the hand-eye case (HE), where the camera is rigidly located on the robot hand. However, the difference is only apparent, and the two cases are just different formulations of fundamentally the same problem. They are easily unified as soon as a little abstraction is made.

The classical implementation of robot control privileges by referring to the hand frame the tip end of the mechanism. However, from a more abstract point of view, an articulated chain is strictly symmetric in the sense that the hand and the base frame are both terminal elements of the chain. Hence, they are equally valid frames for a robot controller implementation. Thus, a relation also exists between base-screws and joint-screws and can be established from the mechanism's geometry, alone.

For the purpose of visual servoing, the only "natural" reference is the camera, which generates the image signal and which gives rise immediately to a camera screw (8.14). In the HE-case, this variable signal is associated with the object to be grasped, while in the IE-case, a variable signal is associated with the tool used for grasping. In this sense, the varying position is the camera-object one in the HE-case, but the camera-tool one in the IE-case. Additionally, there is an exact correspondence between camera-and-hand in the HE-case, and a rigid setup of camera-and-base in the IE-case. Therefore, the camera screw defines in the first instance a hand-screw in the HE-case, but a base-screw $\hat{\mathbf{T}}_B$ in the IE-case.

While the hand-screw can be driven mechanically, the base is mechanically immobile, so the base-screw is only a virtual one. The practical consequences are,

- Either, the base-screw is first transferred into a corresponding hand-screw, which the low-level robot controller then "automatically" transfers into a joint-screw. This intermediate transformation is determined by the current robot posture $\mathbf{T}_{HB}(t) = \mathbf{T}_{HB}(\boldsymbol{Q}), \boldsymbol{Q} = \boldsymbol{q}(t)$, e.g. by the forward kinematics, alone, so that estimating the tool pose can be omitted.

$$\hat{\mathbf{T}}_H = \mathbf{T}_{HB}(\boldsymbol{Q})\hat{\mathbf{T}}_B\mathbf{T}_{HB}^{-1}(\boldsymbol{Q}) \tag{8.19}$$

- Or, the "virtual" base-screw is directly transformed into corresponding joint-velocities (section 5.3), which amounts to a reformulation of low-level robot control. To see this, it suffices to absorb the similarity in (8.19) into the joint-wise sum of screws $\hat{\mathbf{T}}_j$ (5.49) that governs the interaction model. To this end, convince yourself that each joint-operator $\hat{\mathbf{T}}_j$ can effectively be transferred from the hand to the base frame, from $\mathcal{H}$ to $\mathcal{B}$:

$$\hat{\mathbf{T}}_B = \mathbf{T}_{HB} \cdot \left(\sum_j \hat{\mathbf{T}}_{jH} \right) \cdot \mathbf{T}_{HB}^{-1} = \sum_j \left(\mathbf{T}_{HB}^{-1} \cdot \hat{\mathbf{T}}_{jH} \cdot \mathbf{T}_{HB} \right), \tag{8.20}$$

$$= \sum_j \hat{\mathbf{T}}_{jB}, \tag{8.21}$$

	base-eye	hand-eye
variable	camera to tool-in-hand	camera to object-on-base
fixed	camera-and-object-on-base	camera-and-tool-in-hand
goal	image-of-tool	image-of-object
implicit	tool-object (ECL)	camera-tool (EOL)

Table 8.1: Abstract comparison of hand-eye and base-eye setup.

Now, without the hand frame intervening at all, a given base-screw immediately results in a respective joint-screw $\hat{\boldsymbol{q}}$.

$$\hat{\boldsymbol{q}} = \left[\begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}_{1B} \cdots \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}_{6B} \right]^{-1} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}_{B}, \tag{8.22}$$

where $\begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}_{jB}$ is the equivalent of $\hat{\mathbf{T}}_{jB}$.

In summary, the Cartesian formulation of the visual servoing paradigm relates a sensor-screw to an actuator-screw. They are related by the sensor-actuator geometry, which is the rigid link between the sensor and the "*sensor-end*" of the actuator. This link is the classical hand-eye link, or the camera-base link in a "base-eye" case, i.e. the IE-case. In both cases, it is this fixed link that transforms a sensor screw to an actuator one, which translates directly to a robot command.

8.2.2 Projective screws

Towards a projective representation of the action domain, let us briefly discuss the screw representations introduced above. There are a number of notions whose projective generalizations are straight-forward. The analogue of the point-velocity $\dot{\boldsymbol{S}}$ is $\dot{\boldsymbol{X}}$ or $\dot{\boldsymbol{N}}$, as long as ambient space is respectively Euclidean or affine, with a trivial fourth row, and $\dot{\boldsymbol{M}}_\rho$ is projective. The analogue of a kinematic screw is obtained by conjugating the operator matrices $\hat{\mathbf{T}}$ to obtain the corresponding $\hat{\mathbf{H}}$. In addition, under the corresponding rigid-stereo assumption, an similar screw transfer, now of projective screws, can be calculated using $\mathbf{H}_R T$ as the projective displacement group. However, there is no calibration-free projective representation with only six parameters, so the operator matrices $\hat{\mathbf{H}}$ are highly over-parameterized.

Thus, although a projective screw in the form of a 4×4 matrix can be calculated linearly, this requires more than three point-velocities, and fails to guarantee the Euclidean-like algebraic structure underlying these matrices. Also, it still seems unreasonable to artificially allocate frames, now projective ones, onto tool, robot, hand, etc. However, the action domain is

ultimately still a joint-space associated with the mechanical structure of the actuator. An approach that represents the action domain as a joint-space in the projective case is discussed in the following section.

As already mentioned, the joint-screws and kinematic screws coincide in case of a Cartesian robot, whereas they are determined by means of an intermediate kinematic screw in case of a general robot, e.g. a $6R$ structure.

8.2.3 Joint-space and joint-screws

The most natural representation of the action domain is the joint-space of the robot mechanism and the corresponding joint-velocities. For a 6-dof mechanism, they can always be represented by a 6-vector in $\mathbb{R}^6$, which are called "*joint-screws*" in this thesis. A joint-configuration $\boldsymbol{q}$ is always relative to an arbitrary origin $\boldsymbol{q} = \mathbf{0}$ in joint-space, the "*zero-reference*", but the joint-velocities $\dot{\boldsymbol{q}}$, or equivalently the joint-screws $\hat{\boldsymbol{q}}$, do not depend on that.

In the Euclidean case, for a kinematic screw $(\boldsymbol{v}, \boldsymbol{w})^\top$ to define a joint screw, it is sufficient to linearize the dynamic kinematic model around $\boldsymbol{Q}$ and solve the respective linear equations in $\hat{\boldsymbol{q}}$. In detail, each joint j is associated at instant t with a kinematic screw. Depending on the current robot configuration $\boldsymbol{Q} = \boldsymbol{q}(t)$, this can be written in vector notation as $\left(\boldsymbol{v}_j|_{\boldsymbol{Q}}, \boldsymbol{w}_j|_{\boldsymbol{Q}}\right)^\top$.

$$\hat{\boldsymbol{q}} = \left[\begin{pmatrix} \boldsymbol{v}_1|_{\boldsymbol{Q}} \\ \boldsymbol{w}_1|_{\boldsymbol{Q}} \end{pmatrix}, \quad \cdots \quad , \begin{pmatrix} \boldsymbol{v}_6|_{\boldsymbol{Q}} \\ \boldsymbol{w}_6|_{\boldsymbol{Q}} \end{pmatrix} \right]^{-1} \cdot \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}_H$$

In a projective ambient space, however, such a concise and in fact minimal 6×6 matrix system can no longer be formulated, because the matrix representation of the joint-operators $\hat{\mathbf{H}}_j$ has to be used. Although the respective linear system of 4×4 matrices, or equivalently of 16-vectors of coefficients, can still be formulated, a unique solution of this system can not be guaranteed due to the high redundancy.

$$\hat{\mathbf{T}} = \sum_{j=1}^{n} \hat{\boldsymbol{q}}_j \cdot \hat{\mathbf{T}}_j|_{\boldsymbol{Q}}, \qquad \hat{\mathbf{H}} = \sum_{j=1}^{n} \hat{\boldsymbol{q}}_j \cdot \hat{\mathbf{H}}_j|_{\boldsymbol{Q}}.$$

In addition, numerics would require the operators to perfectly fit the underlying algebraic structure, including constraints, which is not guaranteed given an $\hat{\mathbf{H}}$ obtained from unconstrained linear calculation.

For these reasons, determining explicitly a projective screw seems not worthwhile. A more appropriate approach, however, is to eliminate all intermediate frames, and all representations in form of kinematic screws, aiming at a direct relationship between the perception domain and the joint-space representation of the action domain. This can be done, and the solution is similar to the one in 8.2.1, where the various screw transfers, camera to

tool to hand to base, have been contracted into joint-operators that directly refer to the base frame. In addition, a further contraction can be made by conjugation of the joint-operators with the camera-base link in order to give a camera-based kinematic model, as introduced in chapter 5.

Finally, a linear system in matrix form is obtained as soon as the projective kinematic model is contracted against the 3D-points used for visual servoing. Possible numerical problems in the operator matrices can be avoided in this way. Generically, the respective Jacobian matrices look like (5.47), where various specific formulations arise, depending on the way 3D-information is incorporated.

$$\dot{\boldsymbol{M}} = \begin{bmatrix} \frac{\partial \boldsymbol{M}(\boldsymbol{q})}{\partial q1} & \cdots & \frac{\partial \boldsymbol{M}(\boldsymbol{q})}{\partial q6} \end{bmatrix} \cdot \dot{\boldsymbol{q}}$$
$$\text{where e.g. } \frac{\partial \boldsymbol{M}(\boldsymbol{q})}{\partial q1} = \hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}} \boldsymbol{M}(\boldsymbol{q}). \tag{8.23}$$

So, the following section explicitly derives laws for visual joint-space control using one or two cameras. The respective Jacobian matrices can be regarded as a more or less faithful representation of the action domain in terms of its co-screws in the respective perception domain.

8.3 Projective control in an image plane

In the original approach [19], visual servoing is formulated as a task-function in a single image-plane. The error-signal $\boldsymbol{e}$ is a stack of difference vectors between the current image-points $\boldsymbol{s}(t)$ and the respective set-points $\boldsymbol{s}^*$ in the image, where m is the number of such points.

$$\boldsymbol{e} = \begin{pmatrix} \boldsymbol{s}_1 - \boldsymbol{s}_1^* \\ \vdots \\ \boldsymbol{s}_m - \boldsymbol{s}_m^* \end{pmatrix}. \tag{8.24}$$

The perception and action domains can then be related to a $2m \times 6$ stack $\mathbf{J}$ of Jacobian matrices $\mathbf{J}_{1\cdots m}$ of the form $\mathbf{J}_G \mathbf{P} \mathbf{J}_H$, as in (8.8) etc.

$$\dot{\boldsymbol{e}} = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_m \end{bmatrix} \cdot \dot{\boldsymbol{q}} \tag{8.25}$$
$$\mathbf{J} \quad \cdot \dot{\boldsymbol{q}}$$

It was shown that $p = 3$ points almost always suffice for the corresponding Jacobians to have full rank, although singularities and local minima can still occur. Then, the pseudo-inverse calculates a control reference $\hat{\boldsymbol{q}}$, with Λ a diagonal gain matrix.

$$\hat{\boldsymbol{q}} = -\Lambda \mathbf{J}^+ \cdot \boldsymbol{e} \tag{8.26}$$

Besides the approximate solutions described in section 8.1.1, the main problem in the monocular case, in contrast to the stereo one, is to recover 3D-information about the current scene, i.e. depth of the controlled 3D-points. This is even more cumbersome in a projective scene model. Below, two solutions are sketched that propose to calculate the analogue of "pose" for a rigid object for which only projective structure is known. In detail, the arguments are based on a projection matrix expressed with respect to at least six rigid points whose projective coordinates $\boldsymbol{E}_{1\ldots6}$ are considered constant. Given their current projections $\boldsymbol{m}_{E_1\ldots E_6}$ onto the image-plane, a linear estimate of $\mathbf{P}$ can always be found from the six projection constraints $\boldsymbol{m}_{E_1\ldots E_6} = \mathbf{P}\boldsymbol{E}_{1\cdots6}$. This is similar but not equivalent to the "virtual" rigid stereo assumption (2.21), in the sense that six rigid points also fixed a projective basis, but in contrast to just five, allow the projection matrix to be recovered. Thus, the second camera in the stereo pair, or their epipolar geometry are no longer required.

The following two cases are practically the most important:

- six points $\boldsymbol{E}$ on the tool (end-effector) are known, not necessarily in Euclidean coordinates, which would yield a simple calibration grid. Additionally, the projective kinematic model of the manipulator is expressed with respect to this fixed projective basis. This actually amounts to a camera-independent and self-contained representation of the action domain. Its relationship with the perception domain is a projective camera matrix $\mathbf{P}$, as introduced above. It suffices to substitute the current estimate of $\mathbf{P}(t)$ at each instant t in (8.8) for a faithful Jacobian-relation to be established.

$$\mathbf{J}_H = \begin{bmatrix} \partial\boldsymbol{E}_{q_1}, \cdots, \partial\boldsymbol{E}_{q_6} \end{bmatrix}, \qquad \partial\boldsymbol{E}_{q_j} = \hat{\mathbf{H}}_j\Big|_{\boldsymbol{Q}} \boldsymbol{E}. \tag{8.27}$$

$$\mathbf{J}_G = \mathbf{J}_G\Big|_{\mathbf{P}(t)\boldsymbol{E}}. \tag{8.28}$$

- for at least three-points on the tool (end-effector), their initial projective coordinates $\boldsymbol{M}(0)$ are known, in addition to a projective kinematic model and the initial camera matrix, all consistent with each other. Then, it is basically this initially known 3D-information on the points that is propagated by continuously updating their positions using the current robot configuration $\boldsymbol{Q}$, i.e. the projective displacement $\mathbf{H}(\boldsymbol{Q})$ the robot has already driven. This case is practically relevant, for instance to be able to continue a projective stereo law (see section 8.4)

in the presence of camera drop-outs.

$$\mathbf{J}_H = \left[\partial \boldsymbol{M}_{q_1}, \cdots, \partial \boldsymbol{M}_{q_6} \right], \qquad \partial \boldsymbol{M}_{q_j} = \hat{\mathbf{H}}_j \Big|_{\boldsymbol{Q}} \boldsymbol{M}(t). \tag{8.29}$$

$$\mathbf{J}_G = \mathbf{J}_G \Big|_{\mathbf{P}\boldsymbol{M}(t)}, \qquad \boldsymbol{M}(t) = \mathbf{H}(\boldsymbol{Q})\boldsymbol{M}(0). \tag{8.30}$$

8.4 Projective control in stereo images

In the various approaches using stereo [26], visual servoing is formulated as a task function on vectors of the direct sum of a stereo pair of images, thereby neglecting the epipolar constraint. The error-signal is a stack of left and right difference vectors between the current points $\boldsymbol{s}, \boldsymbol{s}'$ and their respective set-points $\boldsymbol{s}^*, \boldsymbol{s}'^*$

$$\boldsymbol{e} = \begin{pmatrix} \boldsymbol{s}_1 - \boldsymbol{s}_1^* \\ \vdots \\ \boldsymbol{s}_m - \boldsymbol{s}_m^* \end{pmatrix}, \qquad \boldsymbol{e}' = \begin{pmatrix} \boldsymbol{s}'_1 - \boldsymbol{s}'^*_1 \\ \vdots \\ \boldsymbol{s}'_m - \boldsymbol{s}'^*_m \end{pmatrix}. \tag{8.31}$$

These error-vectors are considered as a first order approximation to image-velocities $\dot{\boldsymbol{s}}, \dot{\boldsymbol{s}}'$. So, the corresponding Jacobian is a stack of size $4m \times 6$, containing the left camera and right camera blocks of Jacobians (8.25), $\mathbf{J}$ and $\mathbf{J}'$, of size $2m \times 6$.

$$\begin{pmatrix} \dot{\boldsymbol{e}} \\ \dot{\boldsymbol{e}}' \end{pmatrix} = \begin{bmatrix} \mathbf{J} \\ \mathbf{J}' \end{bmatrix}_{(4m \times 6)} \dot{\boldsymbol{q}}. \tag{8.32}$$

Each block holds two rows for each point, which are in fact a concatenation of the Jacobian for the projection mapping, and the one for the kinematic mapping, which most generally is the projective kinematics. Their generic form looks like

$$\mathbf{J} = \mathbf{J}_G \cdot \mathbf{P} \cdot \mathbf{J}_H, \qquad \mathbf{J}' = \mathbf{J}_G \cdot \mathbf{P}' \cdot \mathbf{J}_H. \tag{8.33}$$

Different formulations of this product of Jacobians are possible, depending on the way the 3D information in the point-coordinates $\boldsymbol{M}$ is incorporated.

- continuous triangulation of points over time, denoted $\boldsymbol{M}(t)$, and update of the kinematics around $\boldsymbol{Q} = \boldsymbol{q}(t)$ to $\hat{\mathbf{H}}_j|_{\boldsymbol{Q}}$

$$\mathbf{J}_H = \Big[\partial \boldsymbol{M}_{q_1} \big| \quad \cdots \quad \big| \partial \boldsymbol{M}_{q_6} \Big], \qquad \partial \boldsymbol{M}_{q_j} = \hat{\mathbf{H}}_j \Big|_{\boldsymbol{Q}} \boldsymbol{M}(t), \tag{8.34}$$

$$\mathbf{J}_G = \mathbf{J}_G \Big|_{\mathbf{P}\boldsymbol{M}(t)}, \qquad \mathbf{J}'_G = \mathbf{J}'_G \Big|_{\mathbf{P}'\boldsymbol{M}(t)}, \qquad \boldsymbol{M}(t) \text{ triangulated.}$$

- no continuous triangulation, but an a-priori triangulation of points $\boldsymbol{M}(0)$ for the zero $\boldsymbol{q} = \mathbf{o}$ defining the kinematic model. The kinematics are used to develop the projection Jacobian around $\mathbf{M}(t) = \mathbf{H}(\boldsymbol{q}) \cdot \boldsymbol{M}(0)$:

$$\mathbf{J}_H = \left[\partial \boldsymbol{M}_{q_1} \middle| \quad \cdots \quad \middle| \partial \boldsymbol{M}_{q_6}\right], \qquad \partial \boldsymbol{M}_{q_j} = \hat{\mathbf{H}}_j \boldsymbol{M}(0), \tag{8.35}$$

$$\mathbf{J}_G = \mathbf{J}_G\Big|_{\mathbf{P}\boldsymbol{M}(t)}, \qquad \mathbf{J}'_G = \mathbf{J}'_G\Big|_{\mathbf{P}'\boldsymbol{M}(t)}, \qquad \boldsymbol{M}(t) = \mathbf{H}(\boldsymbol{q}(t))\boldsymbol{M}(0).$$

Although two points give rise to a Jacobian of size 8×6, the inverse of this matrix analytically fails to constrain all six degrees of freedom of a rigid motion. The residual degree is a rotation about the axis between the two points. For the Jacobian matrix to have full analytical rank, at least three points are required in general. To see this, a geometric argument is given, as this holds in any sort of ambient space. Additionally, for numerical or systematic artifacts to be ruled out, which could possibly cause the numerical conditioning of the Jacobian to become poor, an exact system-model and geometrically consistent measurements are assumed.

Basically, for a single point, its pair of image-velocities $\dot{\boldsymbol{s}}$, $\dot{\boldsymbol{s}}'$ results through $\mathbf{J}_P, \mathbf{J}_{P'}$ in a 3D point-velocity. This constrains the solution through $\mathbf{J}_H$ to a 3-"*dof*" linear form in the joint-velocities. A second pair and the respective 3D-point are related to the first one by a single rigidity constraint: the points' mutual distance. Thus, the second point adds only two constraints, leaving a 1-"*dof*" joint-space solution. The third pair has two rigidity constraints, so its 3D-point has only 1-dof, which fixes the last "*dof*" in joint-space.

In practice, there are several methods to invert the over-constrained Jacobian.

- The pseudo-inverse is most commonly used. It yields a command which best fits the desired image-velocities in a least-squares sense.

$$\hat{\boldsymbol{q}} = -\Lambda \begin{bmatrix} \mathbf{J} \\ \mathbf{J}' \end{bmatrix}^{+} \begin{pmatrix} \boldsymbol{e} \\ \boldsymbol{e}' \end{pmatrix}_{(4m)}. \tag{8.36}$$

- The block-wise pseudo-inverse, with one block for the left, and one block for the right camera. Essentially, this corresponds to a sum of two parallel monocular laws.

$$\hat{\boldsymbol{q}} = -\Lambda \begin{bmatrix} \mathbf{J}^{+} & \mathbf{J}'^{+} \end{bmatrix}_{(n \times 4m)} \begin{pmatrix} \boldsymbol{e} \\ \boldsymbol{e}' \end{pmatrix}_{(4m)}. \tag{8.37}$$

8.5 Projective Cartesian control

This section describes a directly computed control, consisting of a feed-forward part that guides the motion along a globally valid and visually feasible trajectory (see also chapter 7), and a feed-back part that drives a Cartesian configuration-error to zero.

Classical visual servoing laws compute the robot control by applying the inverse Jacobian (8.36) to an error-vector in point-coordinates, which most commonly is in image-coordinates $\boldsymbol{s}_i(t) - \boldsymbol{s}_i^*$ (8.4). Geometrically, this determines the instantaneous rigid motion that drives each point with a velocity-vector opposite to its respective error-vector [60]. As a result of this local linear approximation, the convergence and stability depends highly on the conditioning of the Jacobian matrix. Moreover, the dimensionality of the problem, i.e. the six degrees-of-freedom of a rigid motion is rarely matched by practical error-vectors, especially in the stereo case, so that often a pseudo-inverse is used to calculate a least-squares approximation to the desired point-velocities.

In contrast, the task is represented here in terms of a "*guided motion*" towards the target, as chapter 7 already suggests. In particular, the computation of primitive components in the form of their joint-space representations suggests this specific formalism to allow for a direct computation of a corresponding robot control, at least if the control-law is to be in joint-space. Additionally, the use of a stereo camera as a triangulation device allows pairs of image-points $\boldsymbol{s}_i$,$\boldsymbol{s}'_i$ to be replaced by reconstructed point-coordinates $\boldsymbol{A}_i$ in three-space, most generally by a projective reconstruction. In this context, a task is assumed to be achieved as soon as the points' current positions $\boldsymbol{A}_i(t)$ coincide with the target ones $\boldsymbol{A}_i^*$.

8.5.1 Projective control in $(\tau, \theta_r, \theta_p)$-space

The general idea underlying projective Cartesian control can be described as follows. On the one hand, the constraints on the motion are formulated at the current work-space position $\boldsymbol{A}_i(t)$, and globally ensure that the motion follows a Cartesian trajectory (7.9) in $\hat{\mathbf{H}}_t$, $\hat{\mathbf{H}}_r$, $\hat{\mathbf{H}}_p$. On the other hand, the constraints on the motion arising from a projective kinematic model (see $\hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}}$ in (5.46)) are formulated (5.21) at the robot's current position $\boldsymbol{Q}$ in joint-space, and allow instantaneous joint-space motions $\hat{\boldsymbol{q}}_t, \hat{\boldsymbol{q}}_r, \hat{\boldsymbol{q}}_p$ to be calculated. Since they are equivalent to the underlying trajectory components, they can immediately be sent to the joint-level controllers. In addition, the respective calculations provide a corresponding "*distance-to-target*" along the trajectory, which amounts to a lower-dimensional feed-back error. It has an unambiguous Cartesian interpretation, and corresponds to the degrees-of-freedom the feed-forward leaves unconstrained.

In detail, linking the primitive motions $\sigma\hat{\mathbf{H}}_t$, $\sigma\hat{\mathbf{H}}_r$, $\sigma\hat{\mathbf{H}}_p$ are essentially

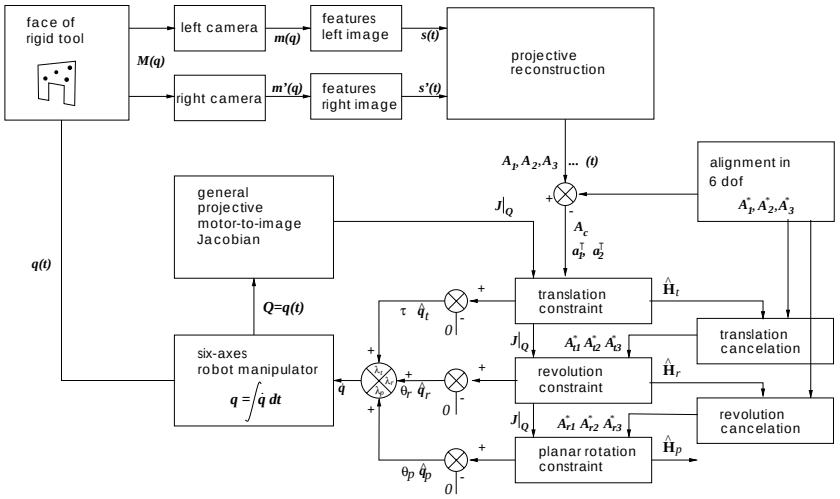

Figure 8.1: Block-schema illustrating the various steps in projective Cartesian control in three components.

linear translations and rotations about a center-point each governed by a single parameter σ. They combine to a one-parameter Cartesian motion $\mathbf{H}_c(\sigma)$, which has the form of a product-of-exponentials (8.38). The geometrical construction of the three generators, and their order in the product ensures that the primitive motions are superposed as desired, without an undesirable coupling becoming effective (7.9).

$$\mathbf{H}_c(\sigma) = \exp\left(\sigma\hat{\mathbf{H}}_t\right)\exp\left(\sigma\hat{\mathbf{H}}_r\right)\exp\left(\sigma\hat{\mathbf{H}}_p\right). \tag{8.38}$$

The formulation as one-parameter motion suggests that Cartesian motions could form a group, and that consequently a single matrix could generate a Cartesian motion. However, this is not the case, since e.g. the inverse $\mathbf{H}_c^{-1}$ cannot be expressed as (8.38). However, the definition of tangent operators $\dot{\mathbf{H}}(\sigma)\mathbf{H}_c^{-1}(\sigma)$ results in a continuously varying, analytical expression $\hat{\mathbf{H}}_c(\sigma)$ (7.11) in σ, which geometrically amounts to the velocity of a Cartesian motion at instant σ.

As long as the origin $\sigma = 0$ of the Cartesian motion is considered, its instantaneous description as tangent operator $\hat{\mathbf{H}}_c(\sigma)$ has a very convenient form due to $\mathbf{H}_c(0) = \mathbf{I}$.

$$\hat{\mathbf{H}}_c = \lim_{\sigma\to 0}\frac{\mathbf{H}_c(\sigma)-\mathbf{I}}{\sigma} = \hat{\mathbf{H}}_t + \hat{\mathbf{H}}_r + \hat{\mathbf{H}}_p \tag{8.39}$$

$$= \frac{d}{d\sigma}\dot{\mathbf{H}}_c(0)\mathbf{H}_C^{-1}(0) \tag{8.40}$$

The velocity screw $\hat{\mathbf{H}}_c$ being the sum of the operator matrices allows in particular the corresponding joint-screw $\hat{\boldsymbol{Q}}_c$ to be calculated directly as the

sum of component-wise joint-screws (8.41).

$$\hat{\boldsymbol{q}}_c = \hat{\boldsymbol{q}}_t + \hat{\boldsymbol{q}}_r + \hat{\boldsymbol{q}}_p. \tag{8.41}$$

Remember, for this equivalence to be valid, the joint-screws as well as the operators have to be developed around the current state of the system, which comprises both, the joint-operators $\hat{\mathbf{H}}_j\big|_{\boldsymbol{Q}}$ (5.21) at the current joint-space configuration $\boldsymbol{Q}$, as well as the Cartesian operators $\hat{\mathbf{H}}_t$, $\hat{\mathbf{H}}_r$, $\hat{\mathbf{H}}_p$ at the current work-space configurations $\boldsymbol{A}(t), \boldsymbol{A}^*$. Thus each iteration of the control-loop amount to newly developing a Cartesian trajectory (see section 7.4.

Therefore, a gain-weighted proportional control in the 3-dof feed-back error $(\tau, \theta_r, \theta_p)$ corresponding to a "distance-to-target" in the Cartesian configuration is a valid control law:

$$\boldsymbol{e} = (\tau, \theta_r, \theta_p)^\top, \quad -\dot{\boldsymbol{e}} = (\lambda_t, \lambda_r, \lambda_p)\ \boldsymbol{e}, \quad \dot{\boldsymbol{q}} = \left[\hat{\boldsymbol{q}}_t,\ \hat{\boldsymbol{q}}_r,\ \hat{\boldsymbol{q}}_p\right] \dot{\boldsymbol{e}}, \tag{8.42}$$

$$\mathbf{H}_e(\dot{\boldsymbol{e}}) \approx \exp\left(\lambda_t \tau \hat{\mathbf{H}}_t + \lambda_r \theta_r \hat{\mathbf{H}}_r + \lambda_p \theta_p \hat{\mathbf{H}}_p\right), \quad \text{for } \tau,\ \theta_r, \theta_p \text{ small.} \tag{8.43}$$

8.5.2 Discussion

The question is now, to what extent and under what assumptions will this direct control produce a Cartesian trajectory? When will it deviate or degenerate? Two aspects have to be considered: firstly, the general relationship between joint-screws and projective screws, $\hat{\boldsymbol{q}}$ and $\hat{\mathbf{H}}$, and secondly, the relationship between a Cartesian trajectory $\hat{\mathbf{H}}_c$ and its components, either represented as $\hat{\mathbf{H}}_t$, $\hat{\mathbf{H}}_r$, $\hat{\mathbf{H}}_p$, or as $\hat{\boldsymbol{q}}_t$, $\hat{\boldsymbol{q}}_r$, $\hat{\boldsymbol{q}}_p$. Generally, this equivalence holds only instantaneously at a configuration $\boldsymbol{Q}$, which implies that high joint-velocities require in any case a high frequency of the control loop, i.e. require a recalculation of joint-screws at a rate which accounts for the rapid variations in $\boldsymbol{Q}$. A similar argument holds in the second case. The t, r, p operators are representing the Cartesian motion only at the origin. As long as the Cartesian error $\boldsymbol{e}$ or the respective gains remain small, the approximations are valid. In the presence of larger errors $\boldsymbol{e}$, however, a higher loop frequency, or alternatively small gains, can ensure that such a piece-wise linearized control is technically feasible. Additionally, imposing limits on Cartesian or joint-velocities, $\dot{\boldsymbol{e}}$ or $\dot{\boldsymbol{q}}$, can restrict a command locally, according to the local validity of the linear model. Alternatively, the update of joint-screws could be calculated at a higher frequency than the Cartesian ones, in analogy to a low-level Cartesian robot control.

Interpolation

In contrast, high-performance systems with corresponding high gains would require higher-order solutions,providing either an algebraic inversion (8.45)

of the various products-of-exponentials involved, or at least take into account the coupling terms (8.46), (8.47) up to higher orders until they get numerically insignificant.

On the kinematic level, for example, most industrial robot controllers interpolate the desired work-space trajectory and generate a respective joint-space trajectory by means of the inverse kinematics. An analogous solution based on the inverse projective kinematics (see chapter 6) would calculate set-points $\boldsymbol{q}^*$ in joint-space from intermediate projective displacement $\mathbf{H}_c(\sigma)$, so that a constant joint-velocity $\hat{\boldsymbol{q}} = \Lambda(\boldsymbol{q}^* - \boldsymbol{q})$ integrates to the desired Cartesian motion $\mathbf{H}_c$, with gain matrix Λ.

$$\mathbf{H}_c(t^*) = \int_t^{t^*} \left(\hat{\boldsymbol{q}}_1 \hat{\mathbf{H}}_1 \big|_{\boldsymbol{Q}}(t) + \quad \cdots \quad + \hat{\boldsymbol{q}}_n \hat{\mathbf{H}}_n \big|_{\boldsymbol{Q}}(t) \right) dt$$

On the work-space level – which could be a projective one in the general case – a similar interpolation-based inverse solution is feasible, and is most conveniently be expressed in terms of a fixed projective screw $\hat{\mathbf{H}}_x$, whose integral over an entire cycle, σ to σ^*, produces the desired motion. In contrast to the instantaneous solution $\dot{\mathbf{H}}_c(\sigma)\mathbf{H}_c^{-1}(\sigma)$, it could be applied constantly during an entire cycle.

$$\mathbf{H}_c(\sigma^*) = \int_\sigma^{\sigma^*} \hat{\mathbf{H}}_x d\sigma = \exp \hat{\mathbf{H}}_x \tag{8.44}$$

The solution $\hat{\mathbf{H}}_x$ is in general the twist corresponding to $\mathbf{H}_c(\sigma)\mathbf{H}_c^{-1}(\sigma)$, and can be calculated in first instance as shown in the appendix (A.10). However, since interpolation usually is concerned with small motions, a direct solution for $\hat{\mathbf{H}}_x$ from $\hat{\mathbf{H}}_t$,$\hat{\mathbf{H}}_r$,$\hat{\mathbf{H}}_p$ would be preferable, also from a numerical point of view. Such a solution would take into account higher-order coupling and does generically look like

$$\exp(\hat{\mathbf{H}}_x) = \prod_{j=1,2,\ldots} \exp(\hat{\mathbf{H}}_j) \tag{8.45}$$

$$\begin{aligned} \hat{\mathbf{H}}_x = {} & a'\hat{\mathbf{H}}_1 + b'\hat{\mathbf{H}}_2 + c'\hat{\mathbf{H}}_1\hat{\mathbf{H}}_2 + d'\hat{\mathbf{H}}_2\hat{\mathbf{H}}_1 \\ & + e'\hat{\mathbf{H}}_1^2\hat{\mathbf{H}}_2 + f'\hat{\mathbf{H}}_1\hat{\mathbf{H}}_2^2 + g'\hat{\mathbf{H}}_2\hat{\mathbf{H}}_1\hat{\mathbf{H}}_2 + \cdots \end{aligned} \tag{8.46}$$

$$\begin{aligned} \hat{\mathbf{H}}_x = {} & a\hat{\mathbf{H}}_1 + b\hat{\mathbf{H}}_2 + c[\hat{\mathbf{H}}_1, \hat{\mathbf{H}}_2] + \\ & d\left[\hat{\mathbf{H}}_1, [\hat{\mathbf{H}}_1, \hat{\mathbf{H}}_2]\right] + e\left[\hat{\mathbf{H}}_2 [\hat{\mathbf{H}}_2, \hat{\mathbf{H}}_1]\right] + \cdots , \end{aligned} \tag{8.47}$$

where in contrast to the logarithmic solution, explicit relationships between the error-variables θ, τ, and the coefficients $a, b, \cdots$ of the compensating velocities in $\hat{\mathbf{H}}_c$ are apparent.

Campbell-Baker-Hausdorff equation

The underlying expansion is known as the "Campbell-Baker-Hausdorff" equation in the context of Lie-group theory. The general case is best cap-

tured in terms of "*Campbell-Baker-Hausdorff*" power series, and respective polynomials of non-commutating variables. It has efficient iterative solutions, which allow to calculate the coefficients up to numerically significant orders.

Depending on the groups involved, closed-form solutions exist for a number of simple cases, which in case of a Cartesian motion are a translation in connection with two rotations about the same center. While the two rotations can be combined to a single element of their instance of $SO(3)$, a solution for the translation has turned out to be rather difficult to find.

In detail, the product-of-exponentials for two given operators, $\hat{\mathbf{H}}_r$ and $\hat{\mathbf{H}}_p$ can be written as the exponential of a possibly infinite sum of higher-order Lie brackets.

$$\hat{\mathbf{H}}_s = \log\left(\exp(\theta_r\hat{\mathbf{H}}_r)\exp(\theta_p\hat{\mathbf{H}}_p)\right) \tag{8.48}$$

$$= a\hat{\mathbf{H}}_r + b\hat{\mathbf{H}}_r + c[\hat{\mathbf{H}}_r, \hat{\mathbf{H}}_p] + d[\hat{\mathbf{H}}_r, [\hat{\mathbf{H}}_r, \hat{\mathbf{H}}_p]] + e\cdots \tag{8.49}$$

However, since the group and its algebra are only three-dimensional, all brackets of order higher than one can be written in terms of the first three (8.51). The coefficients of the sum can be collected, and rewritten as trigonometric expressions (8.52) by comparison with the trigonometric power series [66].

$$\exp(\theta_s\hat{\mathbf{H}}_s) = \exp(\theta_r\hat{\mathbf{H}}_r)\exp(\theta_p\hat{\mathbf{H}}_p) \tag{8.50}$$

$$\hat{\mathbf{H}}_s = (\sin\tfrac{\theta_s}{2})^{-1}\left(a\hat{\mathbf{H}}_r + b\hat{\mathbf{H}}_p + c\Big[\hat{\mathbf{H}}_r, \hat{\mathbf{H}}_p\Big]\right), \tag{8.51}$$

$$a = \sin\tfrac{\theta_r}{2}\cos\tfrac{\theta_p}{2}, \quad b = \cos\tfrac{\theta_r}{2}\sin\tfrac{\theta_p}{2}, \quad c = \sin\tfrac{\theta_p}{2}\sin\tfrac{\theta_r}{2}. \tag{8.52}$$

Once again, as the operators $\hat{\mathbf{H}}_r$, $\hat{\mathbf{H}}_p$ are just conjugate forms of $so(3)$, the above solutions can be calculated directly from the projective operators. The Lie-bracket can be calculated, also in the projective case, using the matrix commutation, analogous to the classical product of the $so(3)$ algebra:

$$\begin{aligned}[\hat{\mathbf{H}}_r, \hat{\mathbf{H}}_p] &= \hat{\mathbf{H}}_r\hat{\mathbf{H}}_p - \hat{\mathbf{H}}_p\hat{\mathbf{H}}_r = \mathbf{H}_{PE}^{-1}\hat{\mathbf{T}}_r\hat{\mathbf{T}}_p\mathbf{H}_{PE} - \mathbf{H}_{PE}^{-1}\hat{\mathbf{T}}_p\hat{\mathbf{T}}_r\mathbf{H}_{PE} \\ &= \mathbf{H}_{PE}^{-1}\left(\hat{\mathbf{T}}_r\hat{\mathbf{T}}_p - \hat{\mathbf{T}}_p\hat{\mathbf{T}}_r\right)\mathbf{H}_{PE} \\ &= \mathbf{H}_{PE}^{-1}[\hat{\mathbf{T}}_r, \hat{\mathbf{T}}_p]\mathbf{H}_{PE}\end{aligned} \tag{8.53}$$

while at the same time

$$\begin{aligned}[\hat{\mathbf{T}}_r, \hat{\mathbf{T}}_p] &= \hat{\mathbf{T}}_r\hat{\mathbf{T}}_p - \hat{\mathbf{T}}_p\hat{\mathbf{T}}_r = \mathbf{S}^{-1}\left(\hat{\mathbf{W}}_r\hat{\mathbf{W}}_p - \hat{\mathbf{W}}_p\hat{\mathbf{W}}_r\right)\mathbf{S} \\ &= \mathbf{S}^{-1}[\hat{\mathbf{W}}_r, \hat{\mathbf{W}}_p]\mathbf{S},\end{aligned} \tag{8.54}$$

where a matrix $\mathbf{S}$ always can be chosen such that $\hat{\mathbf{W}}_r$, $\hat{\mathbf{W}}_p$ hold $[\boldsymbol{w}]_\times$, the anti-symmetric representation $[\boldsymbol{w}]_\times$ of $so(3)$ in their upper 3×3 blocks.

In addition, the calculation of most of the required angles (see [66]) makes us of the trace, which likewise is independent of the particular representation in use.

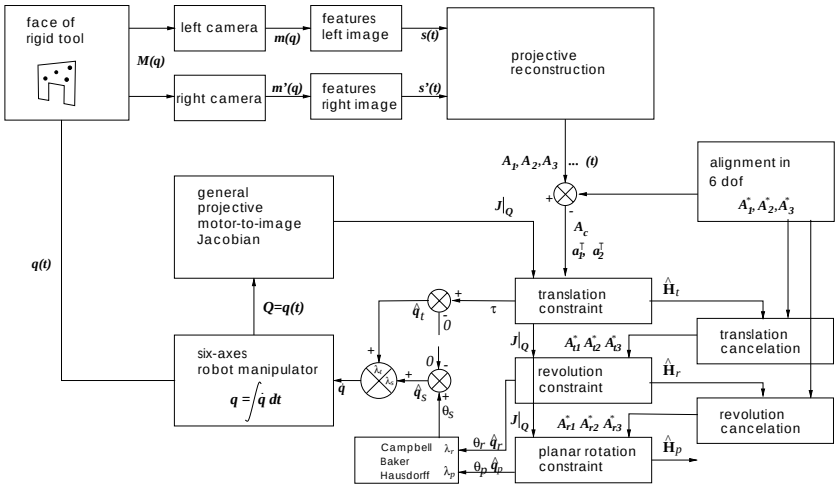

Figure 8.2: Block-schema illustrating the various steps in projective Cartesian control of a single translation and single rotation.

8.5.3 Projective control in (τ, θ_s)-space

Based on this, a new formulation "**directTWO**" of projective Cartesian control can be derived, consisting of a single effective rotation $\hat{\boldsymbol{q}}_s$, and controlling only a 2-dof feed-back error $(\tau, \theta_s)^\top$

$$\boldsymbol{e} = (\tau, \theta_s)^\top, \quad -\dot{\boldsymbol{e}} = (\lambda_t, \lambda_s)\ \boldsymbol{e}, \quad \dot{\boldsymbol{q}} = \left[\hat{\boldsymbol{q}}_t\ \hat{\boldsymbol{q}}_s\right] \dot{\boldsymbol{e}}. \tag{8.55}$$

Although a closed-form solution for $\hat{\mathbf{H}}_s$ exists, based on applying the product of the operator matrices within the CHB-formula, the joint-space representation of actions lacks such a "natural" matrix-product. Thus, for the joint-screw $\hat{\mathbf{q}}_s$ corresponding to $\hat{\mathbf{H}}_s$ to be found, a linear system has to be solved numerically:

$$\hat{\mathbf{H}}_s = \Sigma_{i=1}^{k} \hat{\boldsymbol{q}}_{si} \hat{\mathbf{H}}_i. \tag{8.56}$$

Besides this inconvenience, direct control by $\hat{\boldsymbol{q}}_t + \hat{\boldsymbol{q}}_s$ still fails to take into account the coupling of rotation and translation. This approximation is therefore only valid if the translations remain small.

8.6 Experiments I

Now, a number of experiments with a real implementation of non-metric visual servoing are presented. Since the metric geometry of the independent-eye setup remains unknown, the below given dimensions have to be understood as coarse values, only. The stereo system has approximately $20cm$ baseline, 20^o vergence angle, a $3/4''$ CCD, and $12.5mm$ lenses. The robot

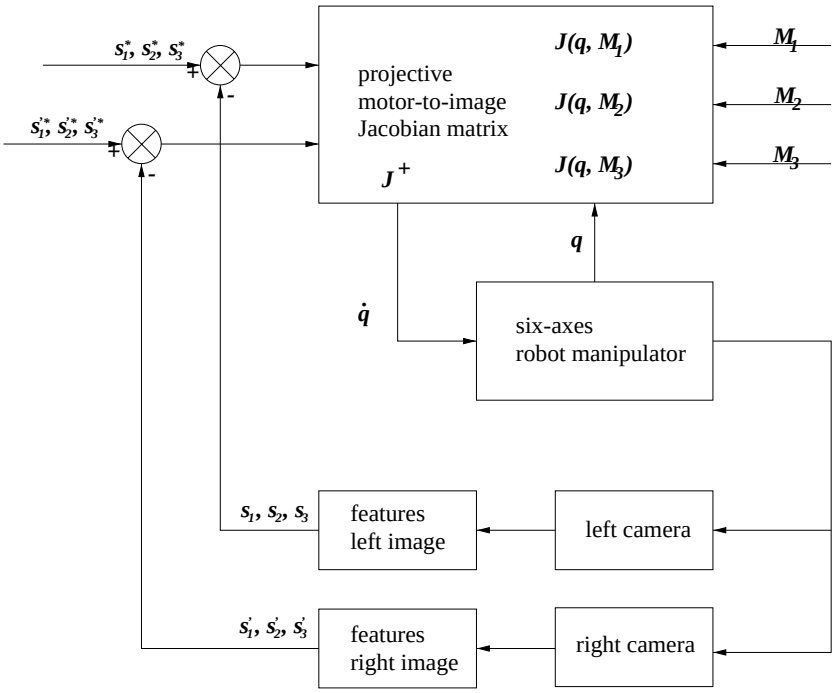

Figure 8.3: Block-schema illustrating the various steps in image-based projective control.

is the PUMA-alike manipulator $RX90$ by Stäubli, the dimension of which are given in tables 5.3, 5.4. The camera observes an end-effector mounted gripper of $10cm$ in size from a distance of $1m$ distance.

Throughout the experiments, the image data consists of stereo projections of four white markers on a black plate rigidly attached to the end-effector. They are extracted from the raw images by first thresholding the intensity image and then localizing the centers of gravity with sub-pixel accuracy. Additionally, the robot's joint-configuration $\boldsymbol{q}(t)$ is read each time t a stereo image is taken.

The workspace-task investigated here is a general 6-dof alignment of a tool with respect to a workpiece. It is encoded [39] by a stereo-image showing markers and features on these two objects in the position of alignment. In practice, they have been acquired and recorded a-priori, or are synthesized from CAD-data. The fact that it is an image pair allows the target-markers to be reprojected onto workspace-images actually taken at task time, even if the cameras have changed.

The visual servoing loop is based on a precise projective kinematic model of the system, which has been recovered using the methods in section 5.5. Preference has been given to the formulation about a fixed zero $\hat{\mathbf{H}}_j$, since it allows the present implementation to easily cope with occasional loss of

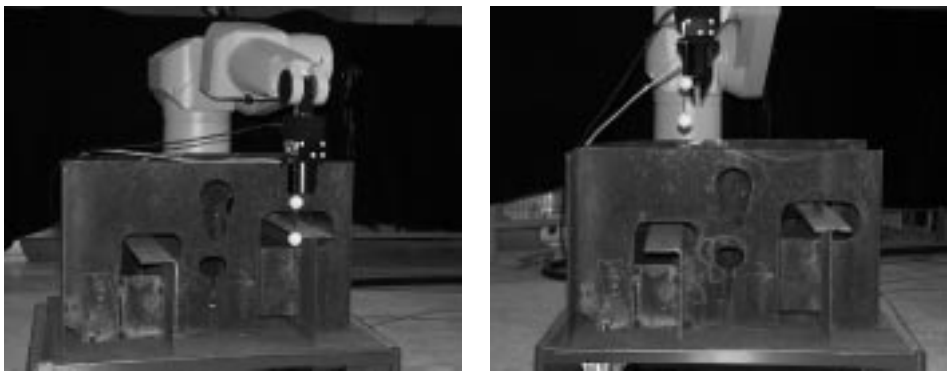

Figure 8.4: Projective encoding of set-points of alignment task: goal position (left), initial position (right)

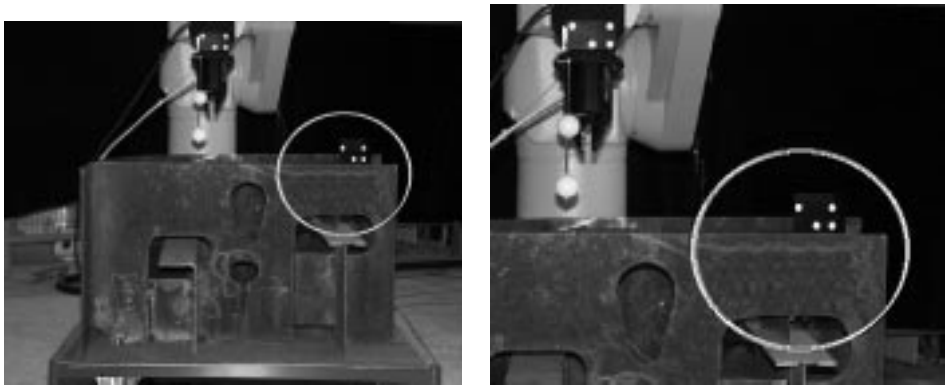

Figure 8.5: Reprojection of visual set-point (left), close-up view (right).

features. Besides that, a precise estimate of the epipolar geometry is fixed and used to model the stereo geometry. A beneficial consequence of precise projective instead of coarse metric modelling is that extraction, and tracking of markers is highly facilitated, since joint-angles read during the operation of the servoing allow for a prediction that reduces the search areas, and hence increases robustness and tracking range.

The below example photographs (Fig. 8.6) show the initial and target position of the manipulator for three successfully realized alignment tasks. Especially task B is a difficult one, since an important rotation of the plate has to be driven. A purely image-based law causes the inclination of the face to become so strong, that some of the markers suffer a strong distortion which makes the extraction algorithm fail. The kinematics based prediction scheme however allows the tracking to tolerate this, and to recover seamlessly. In detail, the accuracy of prediction is far better than a pixel in a large range of the workspace, such that efficient and reliable operation of the image-processing can be assured with no major efforts.

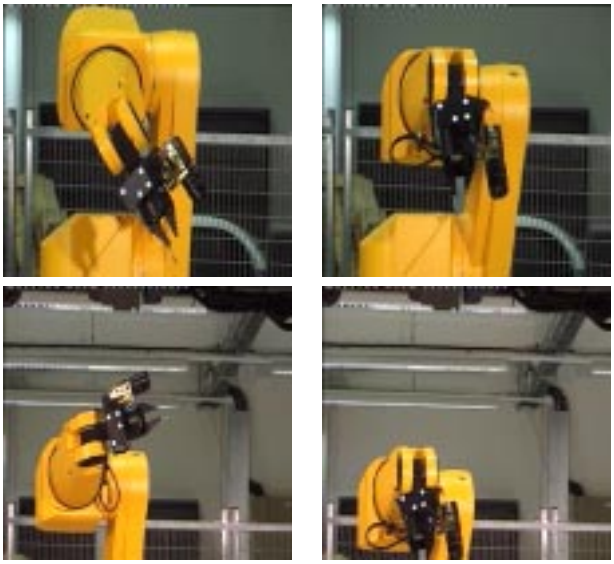

Figure 8.6: Examples of difficult but feasible visual servoing tasks. Initial position (left), target position (right).

8.7 Experiments II

This section recapitulates the experiments presented in [63], where projective Cartesian control was studied. The aim was to validate and evaluate the theoretical results of chapters 7, 8 on a rather hard benchmark configuration, containing the classical problems of large rotations and corresponding self-occlusions. The benchmark considered is a rotation of 180^o around the

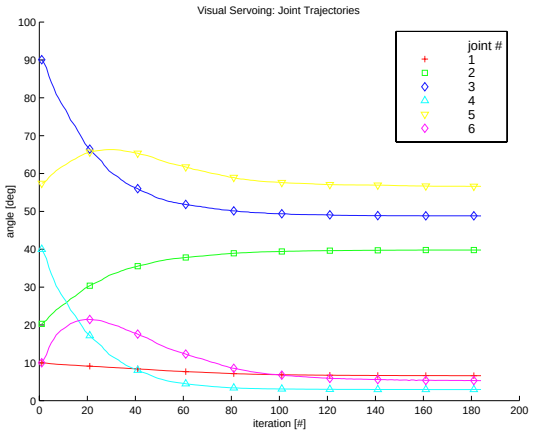

Figure 8.7: Joint-trajectories.

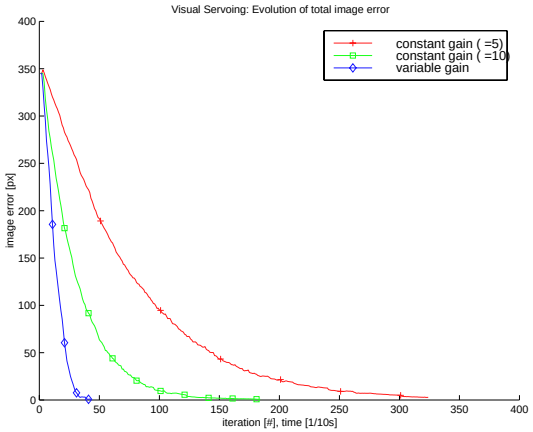

Figure 8.8: Exponential decay of total image-error for several gain-values.

optical axis – the stereo rig's roll axis in our case (Fig. 8.13) – which is known to be degenerate for points in the case of monocular points [8]. The potential self-occlusion is produced by a face of the object being transversal to the image-planes. Besides that, the chosen dimensions correspond to those of the experimental system at INRIA, the projective kinematic data of which was taken from a former self-calibration experiment [62].

Firstly, three classical stereo servoing laws were tried (Fig. 8.14): pseudo-inverse of the stacked Jacobians [26], their block-wise pseudo-inverse [39], and a Jacobian for 3D Euclidean points [7] (like (8.36) but in Euclidean three-space). The second law, which basically sums two independent monocular controls, diverges while moving towards infinity. The other two laws run into the self-occlusion while more less translating towards the target and finally get trapped in a local minimum. Due to small perturbations, both eventually manage to escape, then turn the face almost in-place, again

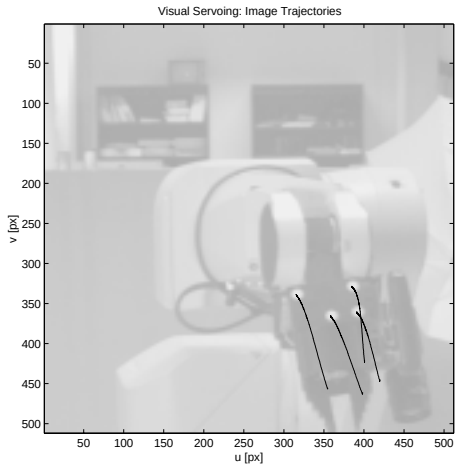

Figure 8.9: Image motion of simple task. Both, projective and coarse calibration manage to servo this task.

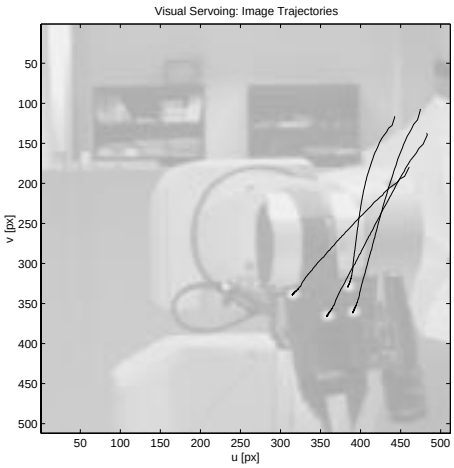

Figure 8.10: Image motion of hard task. A run using coarse calibration failed in this case.

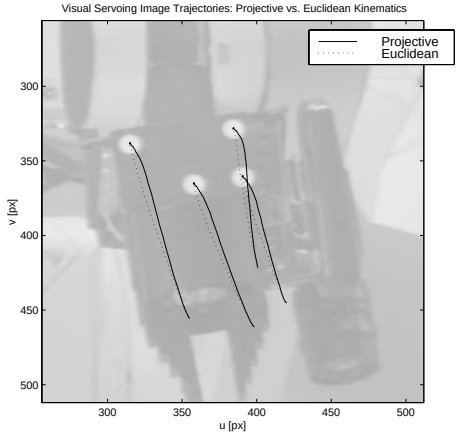

Figure 8.11: Image motion: projective vs. Euclidean.

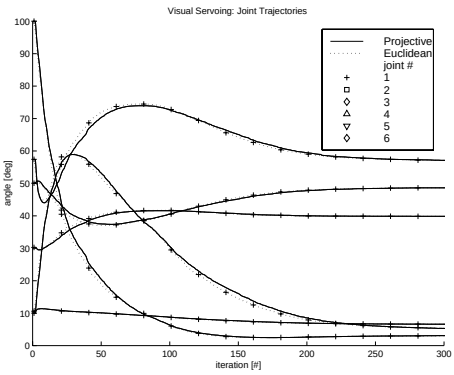

Figure 8.12: Joint motion: projective vs. Euclidean.

through a self-occlusion, before finally converging correctly.

Secondly, trajectory generation from chapter 7 was tested. Figure 8.15 illustrates the solutions found using (7.9), where all the $\mu(\sigma)$ are linear and μ_t is rather steep in Fig. 8.16. The figures are rendered from a central view-point close to the one of the stereo rig. Obviously, they show that the potential self-occlusion has been successfully avoided. Besides this illustrative example, each iteration of the direct control can be interpreted as a newly generated feed-forward trajectory. So, the control experiments likewise validate the reliability and precision of the trajectories (Figs. (8.20), (8.19)).

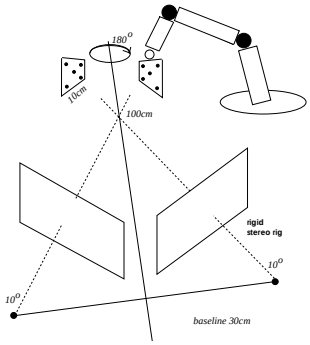

Figure 8.13: Benchmark setup.

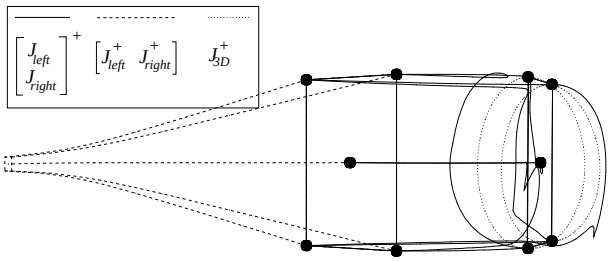

Figure 8.14: Failure of classical stereo visual servoing

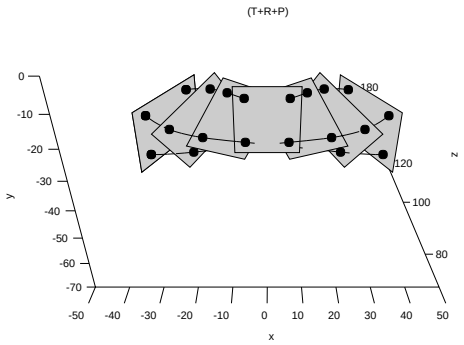

Figure 8.15: Trajectories: visibility is preserved.

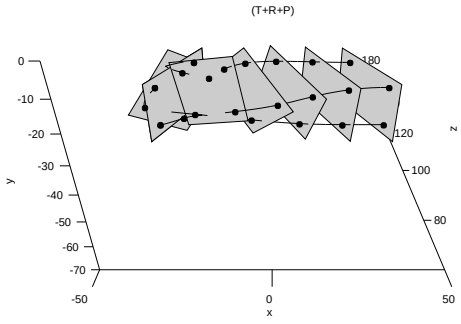

Figure 8.16: Trajectories: early translation.

Thirdly, the two control laws were compared. Figure 8.17 shows the very small deviation of their image trajectories. More remarkable is the center-point's deviation from the desired straight-line trajectory. Figure 8.18 also shows this deviation decreasing with progressively smaller gains. A first conjecture is that this behavior reflects the integration error between the desired "Cartesian"-velocities $\hat{\mathbf{H}}_c$ and the actually driven joint-velocities $\hat{\mathbf{q}}$. This is confirmed by the innermost trajectory for which the joint-velocities were limited to $5^o/s$.

Fourthly, both control-errors (8.42), (8.55) were confirmed to have exponential convergence rate (Figs. 8.19, 8.20). In the case of directTHREE, the translation error τ, which is ambiguous due to the center's scale ρ, is normalized to 100, and its decay is compared to Euclidean ground-truth. The evident difference for τ is only an apparent one, as it is cancelled by a reciprocal scaling of $\hat{\boldsymbol{q}}_t$. The effect of using a projective model rather than a Euclidean one was also studied. For both, the direct control was run, resulting in an almost perfect overlap of the curves for θ_r and θ_p (Fig. 8.20), while τ shows the above described scale-dependency. The results of the directTWO law, once with and once without limited joint-speed, is depicted

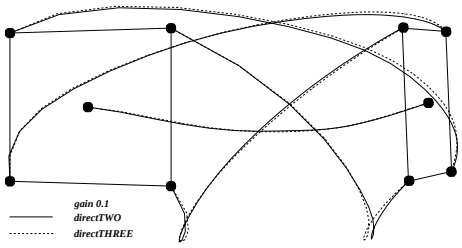

Figure 8.17: directTHREE versus directTWO.

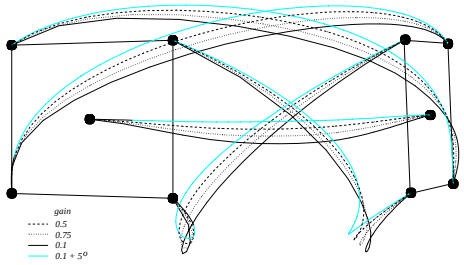

Figure 8.18: Various gains + joint-speed limit.

in figure 8.19. The curve of θ_s clearly reflects the task's overall rotation of 180^o, which was beforehand split between two rotational motions.

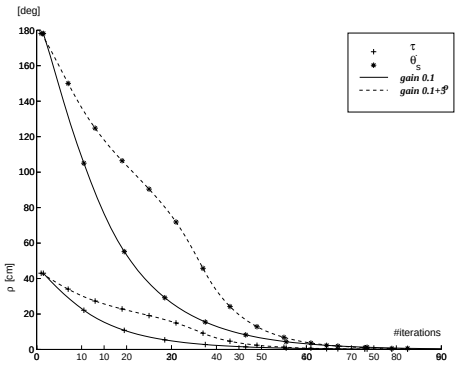

Figure 8.19: directTWO: control-error.

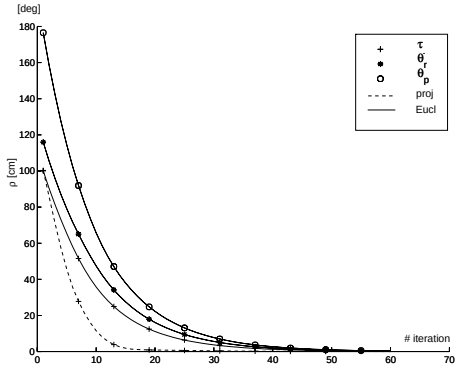

Figure 8.20: directTHREE: control-error.

Fifthly, the errors of the markers image projections as plotted in figure 8.21 clearly shows that they no longer have an exponential decays, nor even a monotonic one. The zero-line actually reflects the center's straight horizontal trajectory. Finally, the corresponding trajectories in joint-space are given in figures 8.22, once without and once with the 5^o limit. Apparently it is the initially high velocities of q_1 and q_3 that are the cause of the above mentioned integration errors, resulting in a drift away from the straight line.

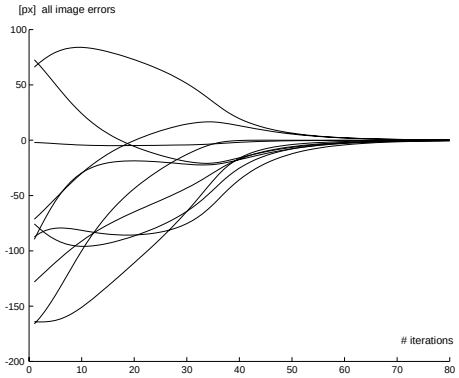

Figure 8.21: Image-errors.

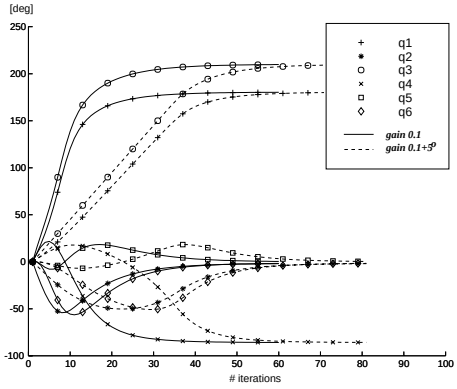

Figure 8.22: Joint-space motion.

Chapter 9

Summary

In this thesis, the **object of study** are robot-vision systems consisting of robot manipulators and stereo cameras. It investigates how to model, calibrate, and operate such systems, where operation refers especially to visual servoing of a tool with respect to an object to be manipulated.

The main **contributions** of this thesis are to introduce a novel, "projective" geometrical formalization of these three problems, and to develop novel coordinate- and calibration-free solutions to them.

The **impact in practice**: the "projective" systems can operate with less priori knowledge. Only metrically unstratified but projectively unified parameters are needed to "projectively calibrate" a fully operational model of the system. This can be done automatically with a self-calibration method. As far as visual servoing is concerned, the projective models are equivalent to classical ones. Also, their particular properties allow new answers to open questions in visual servoing to be found, such as how to control trajectories that are at the same time visually and mechanically feasible over the entire work-space.

The **impact on theory**: a novel and complete projective framework has been introduced, that allows calibration-free representations for all basic geometrical objects relevant to visual servoing, or to camera-based perception-action cycles in general. Most important are the projective representations of rigid- and articulated motion, as well as the respective notion of projective velocities. Effective computational methods for the numerical treatment of these representations were also developed.

9.1 Practice

As far as the **modeling part** is concerned, all parameters of the geometric model refer in full generality to a projective frame, which is defined either by a stereo camera pair or by a rigid structure of five points. As long as this frame remains fixed, the remaining components of the system can vary dy-

namically, while being dynamically covered by the model. The stratification of the geometric model into kinematic, hand-eye, and camera parameters – affine and projective ones – is given up in favor of an integrated, fully projective sensor-actuator model based on the elementary projective motions associated with each of the robot joints. Owing to this, the dynamic model in the form of the Jacobian matrix becomes an analytic function referring to the kinematics and the current configuration of the robot, and its tool. The unified modeling has the advantages that it is no longer coarse and approximate one, but accurate and sound, and that intermediate frames on links, hand, base, and camera have been eliminated in favor of coordinate- and calibration-free representations. The drawbacks are that metric coordinates and error-measures are no longer available, and that the new projective parameters are less intuitive, technically less convenient, and that numerical scaling is more difficult to handle.

As far as **calibration** is concerned, the technical and numerical procedures developed are based on elementary motions of the robot joints, which exploit the strongest available constraints on the system. Other self-calibration techniques use general motions in work-space, and thus exploit only weaker constraints. Earlier manual calibration still relied on prior knowledge, kinematics for hand-eye calibration, and calibration grids or jigs. Besides these advantages, projective calibration is less prone to the instabilities inherent in stratified camera calibration, its numerical solution in a single optimization loop, is free from bias due to fixed priors, and the maximization of the posterior probability in the measurements yields high accuracy as expected from a bundle-adjustment type method. The algebraic closed-form solutions allow for autonomous initialization, good convergence, and global validity of the procedure. The major drawback is that unstratified parameters might become incoherent, requiring their consistency to be enforced numerically, and that priori knowledge can be incorporated only in terms of stratified parameters. However, numerical inconsistency is not essential for the system to operate, and performance remains unaffected by small inconsistencies.

As far as **visual servo control** is concerned, two different approaches are proposed: one that is a Jacobian-based linear control of the feedback-error in the position coordinates of point-features, and the other is a Jacobian-based direct control of feed-forward trajectories that reduce a feedback-error in the configuration of a Cartesian alignment.

Linear feedback-loops can be derived for various formulations of hand-eye interaction. The action domain can be represented either in Cartesian or in fully-coupled joint-space. The perception domain can be represented by an image, an image pair, or a triangulated stereo image. The parameterized, general form of the Jacobian is an accurate and sound local linear model of hand-eye interaction, depending on current actuator states and work-space images. This allows for convergent, precise, and efficient feed-back control,

which is moreover able to handle and incorporate dynamic variations of the system.

The **direct feed-forward control** is feasible and valid thanks to the soundness of the local linear model. The presented novel solution is based on well chosen constraints on the projective motion and dual motion of projective planes, and remains therefore calibration-free. Basically, a reaching trajectory is parameterized in terms of three primitive reaching components, with a corresponding configuration-error in each component. Each component directly relates to a command in joint-space, and these are combined to first-order to obtain a feed-forward control that fixes 3-dof, while feedback controls the remaining 3-dof. This more elaborate method has the advantages that local minima no longer affect convergence, that the trajectories are under control, and that singularities – in fact only manipulator singularities remain – can be related to each component of the trajectory. The drawback is that non-linear calculations are more involved and might be sensitive to measurement noise. Also, the chosen trajectory might have to be adapted to various user needs, and the described algorithm is only designed to handle a single face of an object.

9.2 Theory

The **scientific approach** consists in revisiting the problem of robot vision and investigating a coordinate-free approach to formalize and solve this problem. While robot geometry is naturally Euclidean, vision geometry is intrinsically projective, so that robot vision requires an integration of projective coordinate systems in the coordinate-free approach, which thus becomes essentially calibration-free. For instance, formulating the independent-eye case without referring to a hand-frame produces a base-eye case, which is analogous to and so unified with the hand-eye case.

Matrix transformations have been chosen as a representational and computational tool for the mathematical developments. Linear algebra provides solutions to most of the algebraic questions concerned. First, the relationship between a frame of reference and a motion matrix is covered by conjugation of the matrix by an adjoint map. For the resulting families of **similar matrices**, the Jordan decomposition is the appropriate analytical tool to characterize angle, invariance properties, and generic type. Thanks to this form, each class of motions important for hand-eye coordination has been given a coordinate-free parameterization. The corresponding solutions for the exponential and logarithmic matrix functions, which have been found in closed form, are crucial for further analytical developments.

As these parameterizations form a constraint manifold in a linear space, the topology, geometry, and differential geometry of this manifold are of interest. Since it is displacement groups that are concerned, the appropriate

algebraic tools are **Lie groups** and their Lie algebras, particularly the special Euclidean group $SE(3)$. Homographies of projective three-space suffice to represent faithful and homomorphic representations of this group, and its subgroups, defined within a projective coordinate frame. In order to relate these representations to a robot vision system, a rigid-stereo assumption is made, which amounts to fixing a projective basis of the ambient space. Algebraically, such a choice gives rise to a corresponding embedding of $SE(3)$ in the homography group, which is named a "**projective motion**".

This notion has been developed in detail to produce a projective framework containing projective counterparts for all components in robot vision and all classical formalisms for hand-eye coordination. It covers hence general rigid motions, and elementary subgroups representing robot joints, for instance, which can be combined to represent articulated motion in the form of **products-of-exponentials**. Furthermore, the corresponding notions of velocity have also been derived for rigid body and articulated body motion, including the corresponding velocities of point-, line-, and plane-features both in space and in the images. Since they are analytically defined as temporal derivatives of a motion, **velocities** are situated geometrically in the **tangent space** of the Lie group manifold, which algebraically is represented by a **Lie algebra**. Projective representations of Lie algebra elements were defined, and their actions on projective space, i.e. a scale-free point-space, were investigated.

Besides identifying the orbits in this space under the action of projective displacements, the projective tangent space of points and its cotangent space of planes have been used to define elementary motions by means of their differential geometry, i.e. by constraints on their tangent spaces. These are the basic components for trajectory tracking using **Cartesian motions**. The use of the **dual space** is an interesting example on how the projective approach given rise to a deeper understanding of the underlying problems. While fixating two points is a natural way to enforce a rotation about their axis, fixating two (motion-) planes is no longer an intuitive way to enforce a translation along their axis. In this context, rotation and translation have been interpreted as being dual, meaning that they have dualized constraints. Besides representing velocities, Lie algebras and especially their product operation can be used to relate the various components of a trajectory and their respective joint-commands to the expanded motion trajectory. More abstractly speaking, the use of the Lie bracket allows multiplication in the group to be replaced by addition in the algebra, which can highly facilitate the solution of inverse problems. The above theoretical contributions of the thesis are actually a practical application of the mathematical field of **representation theory**. The projectively introduced matrices are matrix representations of $SE(3)$ and $se(3)$ induced by the arbitrary position of the Euclidean structure in the ambient projective space. These matrices and their properties allow the numerical and computational treatment of

projective motion by linear, linear-iterative, and non-linear techniques. It is essentially the invariance of angles under similarities that allows joint-angles and joint-velocities to be directly related to projective motions and their image projections. In additionally, projective inverse kinematics and the Cartesian trajectory-error also rely on this property. Other quantities, such as distance of translation, or point-to-point distances etc. can still only be defined up to an overall rescaling, however, projective rigid motion allows this scale to be fixed individually so that a scaled-distances eventually become well-defined.

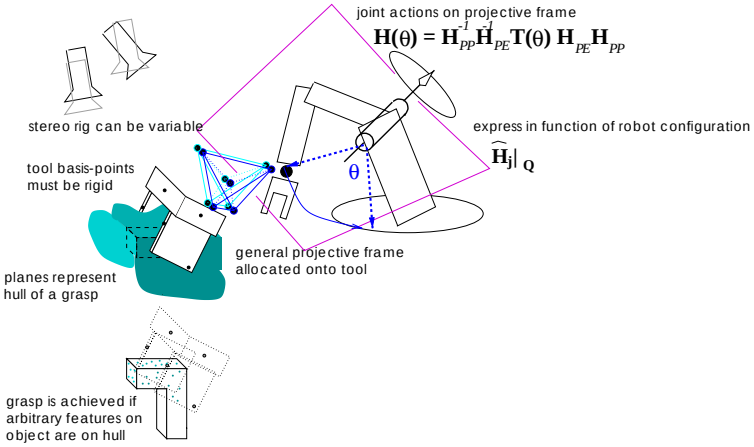

Figure 9.1: Visual servoing system and task are described with respect to an especially designed "visual-servoing-tool". All components, except for the tool, are allowed to dynamically vary on-line.

9.3 Conclusion and Perspectives

In conclusion, a comprehensive projective framework for **coordinate- and calibration - free hand-eye coordination** has been proposed. The solution is very flexible, as no priori knowledge is required and all representations refer to a single but arbitrary reference frame, that is generally projective, e.g. an uncalibrated stereo rig can be used. Stratified metric parameters have successfully been eliminated – their formal occurrence is incidental rather than inherent to image-based servoing – yet, the non-metric approach to visual servoing retains almost the same properties as the metric one, both qualitatively and quantitatively. Many problems that are difficult even in the metric case can be solved projectively, as well, shown for instance for globally valid direct control of visually feasible Cartesian trajectories.

However many things remain to be done, and immediate perspectives of the presented scientific and technical methodology are manifold. First

are the control theoretical questions need to be studied, such as the robustness, stability, and convergence properties of the proposed projective laws, especially of projective Cartesian control. An in-depth comparison of performances under Euclidean and projective modeling should be interesting, especially from a practical point-of-view. To further exploit the computational projective geometry involved in trajectory generation, extensions to alternative and more complex trajectories and real-time interaction seem likely to prove useful. Also, trajectory-tracking using a PD-type law is a very promising approach. In this context, control laws directly based on a triangulation device, i.e. on 3D-points, together with a 6-dimensional representation of the set-points are promising approaches. Especially in the 3D case, a solution under projective modeling has not yet been proposed. As already sketched in the control section, the flexibility of projective modeling has not yet been fully exploited, and remains an open field of application for the presented theories. A next generation of visual servoing system (Fig. 9.1 could consist for instance of a tool, designed for visual servoing requirements, a robot under kinematic self-calibration with respect to this tool, and a larger number of mobile cameras, that are adjusted on-line with respect to this tool. The task could be represented relative to the tool features, and expressed in terms of a minimal number of point-on-plane constraints. The integration of filtering and robust techniques may also be necessary, to ensure reliable and accurate operation.

Perspectives beyond the field of robotics are numerous, and should generally be expected in domains in which actions and visual perceptions, and especially their geometry and dynamics, have to be coupled at high rates. Human motion seems to become the most promising and most import field in the future. Besides the seminal problem of visual servoing, several motion and action domains other than robotic ones are becoming more and more important. Firstly, **animation** and synthetic motion pictures for computer graphics, photo- and video-synthesis, visual avatars, and humanoid robots. Secondly, pre-recorded motion pictures, and new solutions to the cycle of **video** encoding, compression, transmission, streaming, and decoding. Lastly, a very promising direction seems may be instantaneous human motion, actions, and acting for human-computer interfacing beyond gestures. This could lead to high performance, **high dimensional, high dynamics man-machine coupling** for complex, large-scale applications of graphics and vision in engineering, multi-media, entertainment, teaching, and the arts.

Appendix A

Mathematics

A.1 Matrix exponential and logarithm

The exponential function of a square matrix can be defined in terms of a power series, which converges uniformely [81].

$$\exp(\tau \mathbf{A}) = \sum_{n=0}^{\infty} \frac{\tau^n \mathbf{A}^n}{n!}. \tag{A.1}$$

The matrix exponential, in contrast to the scalar one, is generally non-commutative, due to the matrix product

$$\exp(\mathbf{A}) \exp(\mathbf{B}) \neq \exp(\mathbf{A} + \mathbf{B}), \quad \text{unless} \quad \mathbf{AB} = \mathbf{BA}. \tag{A.2}$$

Nevertheless, a number of useful relationships can be derived:

$$\mathbf{S}^{-1} \exp(\mathbf{A})\ \mathbf{S} = \exp\left(\mathbf{S}^{-1}\mathbf{A}\mathbf{S}\right), \tag{A.3}$$

$$\det\left(\exp(\mathbf{A})\right) = \exp(\text{tr } \mathbf{A}), \tag{A.4}$$

$$\exp(\mathbf{A})^{-1} = \exp(-\mathbf{A}), \tag{A.5}$$

$$\exp(\mathbf{A}^\top) = \left(\exp(\mathbf{A})\right)^\top. \tag{A.6}$$

Similarly, a matrix logarithm can be defined in terms of a power series

$$\log\left(\mathbf{I} + \mathbf{A}\right) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{\mathbf{A}^n}{n}. \tag{A.7}$$

If it converges, this is a formal inverse of the matrix exponential in the sense that $\exp(\log(\mathbf{I} + \mathbf{A})) = \mathbf{I} + \mathbf{A}$, but as in the scalar case the general log may be multivalued (e.g. for rotations).

The exp **and** log **of a twist**

Chapters 3 and 4 derive closed-form solutions for exponentials and logarithms of 4×4 matrices that are in particular conjugate forms of pure translations and pure rotations. For the remaining class of conjugated rigid motions, general screws $\mathbf{H}_{RT}$ and their respective "twists" $\hat{\mathbf{H}}_{RT}$, the expansions are as follows:[1]

$$\exp(\theta\hat{\mathbf{H}}_{RT}) = \mathbf{H}_{RT}(\theta), \tag{A.8}$$
$$= \mathbf{I} - \sin\theta\hat{\mathbf{H}}_{RT}^3 + (1-\cos\theta)\hat{\mathbf{H}}_{RT}^2 + \theta(\hat{\mathbf{H}}_{RT}^3 + \hat{\mathbf{H}}_{RT}). \tag{A.9}$$
$$\log(\mathbf{H}_{RT}(\theta)) = \theta\hat{\mathbf{H}}_{RT}. \tag{A.10}$$
$$\mathbf{H}_{RT}^- = \tfrac{1}{2}(\mathbf{H}_{RT} - \mathbf{H}_{RT}^{-1}), \tag{A.11}$$
$$\mathbf{H}_{RT}^+ = \tfrac{1}{2}(\mathbf{H}_{RT} + \mathbf{H}_{RT}^{-1}), \tag{A.12}$$
$$\hat{\mathbf{H}}_{RT}^2 = \tfrac{1}{1-\cos\theta}(\hat{\mathbf{H}}_{RT}^+ - \mathbf{I}), \tag{A.13}$$
$$\hat{\mathbf{H}}_{RT} = \mathbf{H}_{RT}^- \left(\theta\mathbf{I} + (\theta - \sin\theta)\hat{\mathbf{H}}_{RT}^2\right)^{-1}. \tag{A.14}$$

Alternatively, an expression can be found in terms of a concatenation $\mathbf{H}_{RT} = \mathbf{H}_T\mathbf{H}_R$ of rotation and translation, as for Cartesian motions:

$$\exp(\tau\hat{\mathbf{H}}_T)\exp(\theta\hat{\mathbf{H}}_R) = \mathbf{H}_{RT}(\tau,\theta), \tag{A.15}$$
$$= \mathbf{I} + \sin\theta\hat{\mathbf{H}}_R + (1-\cos\theta)\hat{\mathbf{H}}_R^2 + \tau\hat{\mathbf{H}}_T. \tag{A.16}$$

This generally differs from the exponential of the sum $\tau\hat{\mathbf{H}}_T + \theta\hat{\mathbf{H}}_R$

$$\exp(\tau\hat{\mathbf{H}}_T + \theta\hat{\mathbf{H}}_R) = \mathbf{I} + (\tau\hat{\mathbf{H}}_T + \theta\hat{\mathbf{H}}_R^2\hat{\mathbf{H}}_T) +$$
$$+ \sin\theta(\hat{\mathbf{H}}_R - \tfrac{\tau}{\theta}\hat{\mathbf{H}}_R^2\hat{\mathbf{H}}_T) + (1-\cos\theta)(\hat{\mathbf{H}}_R^2 + \tfrac{\tau}{\theta}\hat{\mathbf{H}}_R\hat{\mathbf{H}}_T).$$

In general, translation and rotation homographies are unaligned, i.e. the rotation axis has a direction different from the translation one. As soon as they are aligned, the pair $\hat{\mathbf{H}}_T$ and $\hat{\mathbf{H}}_R$ faithfully represents a screw motion, the two components of which commute, and $\exp(\tau\hat{\mathbf{H}}_T + \theta\hat{\mathbf{H}}_R) = \mathbf{H}_{RT}(\tau,\theta)$.

A.2 Lie groups and Lie algebras

A group is an n-dimensional **Lie group** if the set of its elements can be represented as a continuously differentiable manifold of dimension n, on which the group product and inverse are continuously differentiable functions, as well.

$$(G,H) \to G \cdot H^{-1} \in C^\infty.$$

[1] $\mathbf{H}_{RT}^- = \hat{\mathbf{H}}_{RT}\left(-\sin\theta\hat{\mathbf{H}}_{RT}^2 + \theta(\mathbf{I} + \hat{\mathbf{H}}_{RT}^2)\right)$

For matrix representations, the manifold is a submanifold of $\mathbb{R}^{n\times n}$., and the multiplications and inverse are the usual matrix ones. Any element of the group is represented by a point on the manifold, the neighborhood of which is necessarily diffeomorphic to the neighborhood of the identity matrix $\mathbf{I}$, e.g. by right or left multiplication by the inverse of the element.

A linear space is a **Lie algebra** if a product, the "*Lie-bracket*", is associated with each pair of elements $\hat{\mathbf{G}}$,$\hat{\mathbf{H}}$, such that

$$\begin{aligned}
\text{antisymmetric:} \quad & [\hat{\mathbf{G}},\hat{\mathbf{H}}] = -[\hat{\mathbf{H}},\hat{\mathbf{G}}].\\
\text{bilinear:} \quad & [\alpha\hat{\mathbf{G}},\beta\hat{\mathbf{H}}] = \alpha\beta[\hat{\mathbf{G}},\hat{\mathbf{H}}].\\
\text{Jacobi identity:} \quad & [\hat{\mathbf{F}},[\hat{\mathbf{G}},\hat{\mathbf{H}}]] + [\hat{\mathbf{G}},[\hat{\mathbf{H}},\hat{\mathbf{F}}]] + [\hat{\mathbf{H}},[\hat{\mathbf{F}},\hat{\mathbf{G}}]] = \mathbf{O}.
\end{aligned}$$

For a matrix representation of a Lie group, the tangent-space in $\mathbb{R}^{n\times n}$ of the group manifold at the identity defines a Lie algebra and a corresponding matrix representation if the Lie bracket can be associated with a pair of infinitesimal group elements (tangent-operators) as follows:

$$\begin{aligned}
&\mathbf{G}(0,0) = \mathbf{I},\\
&\hat{\mathbf{G}}_1 = \lim_{\theta_1\to 0} \tfrac{1}{\theta_1}(\mathbf{G}(\theta_1,0) - \mathbf{I}), \qquad \hat{\mathbf{G}}_2 = \lim_{\theta_2\to 0} \tfrac{1}{\theta_2}(\mathbf{G}(0,\theta_2) - \mathbf{I}),\\
&[\hat{\mathbf{G}}_1,\hat{\mathbf{G}}_2] = \hat{\mathbf{G}}_1\hat{\mathbf{G}}_2 - \hat{\mathbf{G}}_2\hat{\mathbf{G}}_1.
\end{aligned}$$

Conversely, a n-dimensional Lie group is generated by applying the **matrix exponential** to the elements of an associated n-dimensional Lie algebra with n independent **generators** $\hat{\mathbf{G}}_{1\cdots n}$ as basis-vectors.

$$\mathbf{G}(\theta_1,\cdots,\theta_n) = \exp\left(\theta_1\hat{\mathbf{G}}_1 + \cdots + \theta_n\hat{\mathbf{G}}_n\right).$$

Locally around the identity, the exponential map is invertible, and the computation of the **matrix logarithm** is convergent.

$$\theta\hat{\mathbf{G}} = \log\mathbf{G}(\theta).$$

The above objects fully characterize the differential geometry of the group as the neighbourhood of each element $\mathbf{G}$ can be mapped diffeomorphically onto the identity. For example, a path $\mathbf{G}(\theta)$ through $\mathbf{G}$ can be associated with the generator of the instantaneous motion at θ:

$$\tfrac{d}{d\theta}\left(\mathbf{G}^{-1}\mathbf{G}(\theta)\right) = \hat{\mathbf{G}}(\theta).$$

.

A.3 Adjoint map

An invertible matrix defines an adjoint mapping on a general matrix, briefly called "*adjoint map*".

Definition 12 (adjoint map) *The adjoint map associated with an invertible matrix* **S** *is the conjugation or similarity transform* $\mathbf{S}^{-1}\quad\mathbf{J}\quad\mathbf{S}$ *of the argument* **J**. *Each adjoint map is an action and Lie group isomorphism of* **S** *onto the general matrix group.*

$$\mathbf{J} \to \mathrm{ad}_{\mathbf{S}}(\mathbf{J}) = \mathbf{S}^{-1} \cdot \mathbf{J} \cdot \mathbf{S}. \tag{A.17}$$

In the context of this thesis, **J** often represents the action of an actuator within its proper frame, or its standard action on the ambient space, and **S** a transform taking this frame to some other one. Hence, two geometrical interpretations may help to illustrate this.

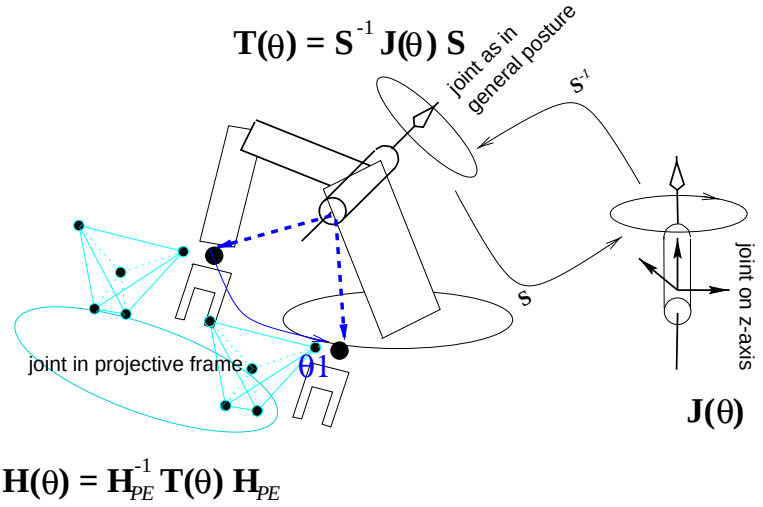

Figure A.1: Adjoint map allows to represent the action of a (revolute-)joint displaced to a general position as well as its action on various reference frames, for instance a projective (tetrahedron) instead of a Euclidean one.

On the one hand, if you consider **J** as a mechanical implementation of a subgroup, e.g. a robot joint, the adjoint map describes this joint after it has been physically moved through $\mathbf{S}^{-1}$.

On the other hand, if you consider **J** as a representation of the joint with respect to a standard frame of reference – here the z-axis is always aligned with the joint axis – the adjoint map produces a description of the same joint in a new reference frame related to the original one by **S**.

Only some parts of the displacement **S** affect the geometry of the subgroup that **J** is representing. The other parts of the adjoint map simply "cancel out". In the appendix on Jordan decompositions A.5, it is shown rigorously that the cancelling parts of **S** are precisely those that commutate with the Jordan matrix. For instance, translation of a translational subgroup always leaves its action unaffected, as the direction remains unaf-

fected. Also, rotation of a rotational subgroup about its own axis leaves its action unaffected.

The key point is that $\mathbf{S}$ can be taken from a more general group than $\mathbf{J}$, e.g. the projective group. In this case, the adjoint map develops the action of $\mathbf{J}$ in a new, more general reference frame or ambient space.

A.4 Representation theory

The definition of rigid rojective motion studied in this thesis can be interpreted as a new representation of the displacement group within the three-dimensional projective group. It is therefore worthwhile to draw a line between $SE(3)$ as an abstract group, with both its formal elements, operations, and properties and its concrete representations as matrix groups acting on various vector spaces. The mathematical theory covering matrix representations is called "*representation theory*". Some basic notions of this are briefly stated below. For an introduction related to the context of robotics, refer to [66]. For concise definitions, consult [81]

Definition 13 (Representation theory at a glance) *A "homomorphism" is a mapping from one group into another group which preserves the group structure. It is an "isomorphism" if it is injective, it is an "endomorphism" if it is surjective. If the two groups are identical, it is an "automorphism".*

An "action" is an operation of a group on a manifold which has the following properties:

- *each operation is a differential map,*
- *the identity element acts as the identity map,*
- *the action preserves the group operation.*

An action on a vector space becomes a "representation" of the underlying group, if its operation is also **linear**.

A "matrix representation" is a representation in matrix form, where a linear action is often a left multiplication with this matrix. Two matrix representations $\mathbf{G}_1$, $\mathbf{G}_2$ of the same group are equivalent, if a matrix $\mathbf{S}$ exists such that its adjoint map $\mathrm{ad}_{\mathbf{S}}$ maps the representations onto each other:

$$\mathbf{S}^{-1}\mathbf{G}_1(g)\mathbf{S} = \mathbf{G}_2(g)$$

The most prominent case of the manifold in question is the group manifold itself, where the action is often left multiplication in the group. Another manifold is a real vector-space, where the action is often the matrix-vector multiplication. These operations are always diffeomorphisms.

A.5 Jordan canonical form

One of the fundamental tools of linear algebra, which is very useful for the analysis of matrix representations, is the Jordan canonical form: an almost-diagonal form associated with similarity classes of regular matrices.

Definition 14 *Two square matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *are "conjugate" or "similar", if a non-singular matrix* $\mathbf{S}$ *exists, such that*

$$\mathbf{B} = \mathbf{S}^{-1}\mathbf{A}\mathbf{S}, \tag{A.18}$$

where $\mathbf{S}$ *is often called a "similarity", and (A.18) a "similarity decomposition".*

The maximal set of mutually similar matrices constitute a "similarity class" of matrices. The algebraic properties that they have in common are called "similarity invariants". The principal invariant is the characteristic polynomial $p_A(\lambda) = 0$ *from* $\det(\mathbf{A} - \lambda\mathbf{I})$*, which induces the following associated invariants*[2]

$$\begin{aligned} \text{trace } (\mathbf{A}) &= \text{trace } (\mathbf{B}), & \text{(A.19)} \\ \det(\mathbf{A}) &= \det(\mathbf{B}), & \text{(A.20)} \\ \text{spec } (\mathbf{A}) &= \text{spec } (\mathbf{B}). & \text{(A.21)} \end{aligned}$$

Hence, similarity preserves not only the eigenvalues but it preserves their multiplicities, as well.

The "*Jordan matrices*" $\mathbf{J}$ introduced below are canonically defined representatives of similarity classes, which display all the similarity properties at a glance. The two matrices are similar if they have the same Jordan matrix $\mathbf{J}$ [41]. The Jordan normal form of a matrix is computed by applying a series of similarity transforms $\mathbf{S}_i$ to the original matrix $\mathbf{A}$ in order to obtain the "*Jordan decomposition*" of the matrix:

$$\mathbf{A} = \mathbf{S}^{-1}\mathbf{J}\mathbf{S}, \quad \text{where} \quad \mathbf{S} = \mathbf{S}_1\mathbf{S}_2, \cdots . \tag{A.22}$$

The Jordan matrix is quasi-diagonal, i.e. it is block-diagonal with Jordan blocks $\mathbf{J}_k(\lambda)$ that are upper-tridiagonal having the eigenvalues of $\mathbf{A}$ on the diagonal, counting multiplicities, possibly having ones on the super-diagonal, and zeros elsewhere. Hence, a Jordan block has the form:

$$\mathbf{J}_k(\lambda) = \begin{bmatrix} \lambda & 1 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & 1 \\ 0 & 0 & 0 & \lambda \end{bmatrix}. \tag{A.23}$$

[2] spec designates the spectrum, i.e. the set of eigenvalues of the argument matrix

In the general case, complex-conjugate pairs of eigenvalues may occur. However, if $\mathbf{A}$ is real, a "*real Jordan decomposition*" can be calculated that replaces complex conjugate pairs by plane rotations, e.g.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

such that both $\mathbf{J}$ and $\mathbf{S}$ become real matrices.

Although the Jordan form is canonical, the corresponding real Jordan decomposition is not. There are usually many similarities $\mathbf{S}$ that decompose $\mathbf{A}$ into $\mathbf{J}$. The class of matrices $\mathbf{C}$ that commute with $\mathbf{J}$ completely characterizes this ambiguity. In fact, $\mathbf{CJ} = \mathbf{JC}$ gives all possible real Jordan decompositions:

$$\begin{aligned} \mathbf{S}^{-1}\mathbf{JS}, &= \mathbf{S}^{-1}\mathbf{JC}^{-1}\mathbf{CS}, \\ &= \mathbf{S}^{-1}\mathbf{C}^{-1}\mathbf{JCS}, \\ &= (\mathbf{CS})^{-1}\,\mathbf{J}\,(\mathbf{CS})\,. \end{aligned} \tag{A.24}$$

Appendix B

Notations

$\mathbb{R}$	real number
$\mathbb{C}$	complex numbers
$\mathbb{R}^2, \mathbb{R}^3, etc.$	real vector spaces of dimensions 2,3, etc. with associated l_2 vector norm (Euclidean)
$\mathbb{P}^2$	projective plane associated with an uncalibrated pinhole camera
$\mathbb{P}^3, \mathcal{P}$	projective three-space associated with a stereo camera pair with known epipolor calibration
$\mathbb{A}^3, \mathcal{A}$	affine three-space associated with a stereo camera pair with known affine calibration
$\mathbb{E}^3, \mathcal{E}$	Euclidean three-space, associated with a stereo camera pair with known strong (metric) calibration
$SE(3)$	special Euclidean Group, Lie group of displacements in $\mathbb{E}^3$
$SO(3)$	special orthogonal group, Lie group of rotations in $\mathbb{E}^3$
$SO(2)$	special orthogonal group, Lie group of rotations in $\mathbb{E}^2$
$se(3)$	Lie algebra to $SE(3)$
$so(3)$	Lie algebra to $SO(3)$
$so(2)$	Lie algebra to $SO(2)$
$\mathbf{I}$	identity matrix, usually 4×4
$\mathbf{H}$	transformation matrix (4×4) of three-dimensional projective space
$\mathbf{A}$	transformation matrix (4×4) of three-dimensional affine space
$\mathbf{T}$	transformation matrix (4×4) of three-dimensional Euclidean space
	$\vdots$

⋮

$\hat{\mathbf{H}}$	tangent operator, generator, Lie algebra element associated with $\mathbf{H}$
$\hat{\mathbf{T}}$	tangent operator, generator, Lie algebra element associated with $\mathbf{T}$
$\hat{\ }$	accent indicating the spatial frame as reference for instantaneous rigid motin
$\check{\ }$	accent indicating the body frame as reference for instantaneous rigid motion
$\mathbf{J}$	Jordan matrix of a transformation matrix
$\mathbf{C}$	commutator of the Jordan form
$\square_{RT}$	subscript indicating a general screw motion, e.g. a general displacement or a cylindrical joint
$\square_{R}$	subscript indicating a zero-pitch screw, e.g. pure rotational motion or a revolute joint
$\square_{T}$	subscript indicating a zero-angle screw, i.e. pure translational motion or a prismatic joint
$\boldsymbol{h}_1 \dots \boldsymbol{h}_4$	column vectors of $\boldsymbol{H}$
$\boldsymbol{k}_1^\top \dots \boldsymbol{k}_4^\top$	row vectors of $\boldsymbol{H}^{-1}$
$\boldsymbol{M}$	column vector (4) of homogeneous point-coordinates in projective three-space
$\boldsymbol{N}$	column vector (4) of homogeneous point-coordinates in affine three-space
$\boldsymbol{X}$	column vector (4) of homogeneous point-coordinates in Euclidean three-space
$(U, V, W, T)^\top$	homogeneous point-coordinates in projective three-space
$(U, V, W, 1)^\top$	homogeneous point-coordinates in affine three-space
$(X, Y, Z, 1)^\top$	homogeneous point-coordinates in Euclidean three-space
$\boldsymbol{A}$	column vector of homogeneous point-coordinates in $\mathbb{P}^3$
$\boldsymbol{a}^\top$	row vector of homogeneous plane-coordinates in $\mathbb{P}^3$
$[[\boldsymbol{A}]]$	multi-coluumn
$[[\boldsymbol{a}^\top]]$	multi-row
$\boldsymbol{m}$	column vector (3) of homogeneous point-coordinates in projective image-plane $[px]$
$\boldsymbol{s}$	column vector (2) of coordinates in standard image-plane [px]

⋮

⋮

$(u, v, 1)^\top$	image coordinates in $[px]$
$(x, y, 1)^\top$	image coordinates in $[mm]$
$(0, 0, 0, 1)^\top$	row vector (1×4) for plane at infinity in Euclidean three-space
$(\boldsymbol{a}, 1)^\top$	row vector (1×4) for plane-at-inifinty in projective three-space
$\mathbf{R}$	rotation matrix (3×3) of three-dimensional Euclidean space
$\boldsymbol{t}$	translation vector (3) of three-dimensional Euclidean space
$\mathbf{K}, \mathbf{K}'$	matrix (3) of intrinsic camera parameters, left and right camera
$\mathbf{H}_{PE}$	homography linking projective and Euclidean ambient spaces, spanned by respective camera frames
$\mathbf{P}, \mathbf{P}'$	projection matrix (3×4) for pinhole cameras, left and right camera
$\mathbf{F}$	fundamental matrix
$\mathbf{E}$	essential matrix
θ, θ_j	angle of general or j^{th} joint, being a revolute joint, in $[rad]$
τ, τ_j	deflection of a general or j^{th} joint, being a prismatic joint, in $[mm]$
q, q_j	generic joint variable of general or of j^{th} joint, being either a prismatic or a revolute joint
$\boldsymbol{q}, \boldsymbol{Q}$	column vector (6) of joint variables: posture or configuration of robot
$\hat{\mathbf{H}}_j$	projective operator of j^{th} joint
$\hat{\mathbf{T}}_j$	Euclidean operator of j^{th} joint
$\mathbf{L}_j$	link transformation of j^{th} joint in Denavit-Hartenberg model
$\square_0$	subscript indicating a datum valid in zero-reference posture
$\square\vert_{\boldsymbol{Q}}$	subscript indicating a datum valid at current posture $\boldsymbol{Q}$
$\square^*$	superscript indicating a datum valid in goal posture
$\mathbf{J}_H$	Jacobian-matrix of forward kinematic model
$\mathbf{J}_G$	Jacobian-matrix of perspective projection
γ	scale of a projective displacement
λ	scale of a projective point in $\mathbb{P}^2$
ρ	scale or "orbital height" of a projective point in $\mathbb{P}^3$
φ	scale or "orbital height" of a projective plane $\mathbb{P}^3$

⋮

$\vdots$

$\square^\top$	matrix transpose
$\square^\perp$	dual of matrix
$\square^{-1}$	matrix inverse
$[\ \ ]_\times$	antisymmetric matrix form of a vector
$\square^+$	matrix pseudo-inverse
$\dot{\square}$	temporal derivative of $\square$, often indicating a velocity
exp	exponential function, for matrices defined as convergent power-series
log	logarithmic function, for matrices defined as convergent power-series
sin	sine function
cos	cosine function
tr	trace of a matrix
det	determinant of a matrix
spec	spectrum (set of eigenvalues) of a matrix
ker	kernel of a matrix
poly	characteristic polynomial of a matrix

Bibliography

[1] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipoles. In *Proceedings of the International Conference on Computer Vision, (ICCV'98)*, pages 863–869, Bombay, India, January 1998.

[2] E. Bayro-Corrochano and J. Lasenby. A unified language for computer vision and robotics. In G. Sommer and J.J. Koenderink, editors, *Algebraic Frames for the Perception–Action Cycle*, volume 1315 of *Lecture Notes in Computer Science*, pages 219–234. Int. Workshop AFPAC'97, Kiel, Springer-Verlag, Heidelberg, 1997.

[3] P. A. Beardsley, I. D. Reid, Andrew Zisserman, and D. W. Murray. Active visual navigation using non-metric structure. In *Proceedings of 5th International Conference on Computer Vision*, pages 58 – 64, Cambridge-MA, June 1995. IEEE Computer Society Press, Los Alamitos – CA.

[4] P. A. Beardsley and Andrew Zisserman. Affine calibration of mobile vehicles. In *Proceedings of Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision*, pages 214 – 221, Xi'an, China, April 1995. Xidan University Press.

[5] O. Bottema and B. Roth. *Theoretical Kinematics*. Dover Publications, 1990.

[6] B. Boufama and Roger Mohr. Epipole and fundamental matrix estimation using the virtual parallax property. In *Proceedings of the International Conference on Computer Vision (ICCV'95)*, pages 1030–1036, Cambridge-MA, 1995.

[7] Enrice Cervera and Philippe Martinet. Combining pixel and depth information in image-based visual servoing. In *Proceedings of the* 9^{th} *International Conference on Advanced Robotics (ICAR 99)*, Tokyo, Japan, October 1999.

[8] Francois Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In David Kriegman,

Greg Hager, and Morse A.S., editors, *The Confluence of Vision and Control*, number 237 in LNCIS Series, pages 66–78. Springer-Verlag, 1998.

[9] Peter Corke. Visual control of robot manipulators – a review. series in robotics and automated systems. World Scientific, 1993.

[10] John J. Craig. *Introduction to robotics: mechanics and control*. Series in electrical and computer engineering. Control engineering. Addison Wesley, Reading-MA, 1989.

[11] Gabriela Csurka, David Demirdjian, Andreas Ruf, and Radu Horaud. Closed-form solutions for the euclidean calibration of a stereo rig. In Burkhardt Hans and Neumann Bernd, editors, *Proceedings of the 5th European Conference on Computer Vision, (ECCV'98)*, volume 1, pages 426 – 442, Freiburg, Germany, June 1998. Springer Verlag.

[12] Gabriella Csurka, David Demirdjian, and Radu Horaud. Finding the collineation between two projective reconstructions. *Computer Vision and Image Understanding*, 75(3):260–269, September 1999.

[13] Kostas Daniilidis and E. Bayro-Corrochano. The dual quaternion approach to hand–eye calibration. In *Proceedings of the International Conference on Pattern Recognition*, volume A, pages 318–322, Vienna, 1996.

[14] David Demirdjian, Gabriella Csurka, and Radu Horaud. Autocalibration in the presence of critical motions. In *Proceedings of the British Machine Vision Conference*, pages 751–759, University of Southampton, UK, September 1998.

[15] David Demirdjian and Radu Horaud. Motion-egomotion discrimination and motion segmentation from image-pair streams. *Computer Vision and Image Understanding*, 78(1):53–68, April 2000.

[16] J. Denavit and R.S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Transactions of ASME — Journal of Applied Mechanics*, 22(2):215–221, June 1955. The historical paper on DH-notation.

[17] Frédérick Devernay and Olivier D. Faugeras. From projective to Euclidean reconstruction. In *Proceedings of Computer Vision and Pattern Recognition, (CVPR'96)*, pages 264 – 269, San Francisco – CA, June 1996.

[18] Tom Drummond and Roberto Cipolla. Application of lie algebras to visual servoing. *International Journal of Computer Vision*, 37(1):21–41, June 2000.

[19] Bernard Espiau, Francois Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. *IEEE Transaction on Robotics and Automation*, 8(3):313–326, June 1992.

[20] O. D. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Boston, 1993.

[21] O. D. Faugeras. Stratification of three-dimensional vision: Projective, affine and metric representations. *Journal of Optical Society of America*, 12:465 – 484, 1995.

[22] Olivier D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In G Sandini, editor, *Proceedings of European Conference on Computer Vision (ECCV'92)*, volume 588 of *Lecture notes in Computer Science*, pages 563–578. Springer Verlag, 1992.

[23] Olivier D. Faugeras and G. Toscani. The calibration problem for stereo. In *Proc. Computer Vision and Pattern Recognition*, pages 15 – 20, Miami Beach-FL, June 1986.

[24] Greg Hager. Calibration-free visual control using projective invariance. In *Proceedings of the International Conference on Computer Vision*, pages 1009–1015, 1995.

[25] Greg Hager, W. Chang, and A.S. Morse. Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination. *IEEE Control Systems Magazine*, 15(1):30–39, 1995.

[26] Gregory Hager. A modular system for robust hand-eye coordination using feedback from stereo vision. *IEEE Transactions on Robotics and Automation*, 13(4):582–595, 1997.

[27] R.M. Haralick, H. Joo, C. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(19):1426–1446, November/December 1989.

[28] Richard Hartley. Projective reconstruction from line correspondences. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 903–907, 1994.

[29] Richard Hartley and Andrew Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[30] Richard I. Hartley. Euclidean reconstruction from uncalibrated views. In Mundy Zisserman Forsyth, editor, *Applications of Invariance in Computer Vision*, pages 237 – 256. Springer Verlag, Berlin Heidelberg, 1994.

[31] Richard I. Hartley. In defence of the 8-point algorithm. In *Proceedings Fifth International Conference on Computer Vision (ICCV'95)*, pages 1064 – 1070, Cambridge-MA, June 1995. IEEE Computer Society Press, Los Alamitos-CA.

[32] Richard I. Hartley and Peter F. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146 – 157, November 1997.

[33] Koichi Hashimoto. *Visual Servoing: real-time control of robot manipulators based on visual sensory feedback*. series in robotics and automated systems. Word Scientific, 1993.

[34] A. Heyden. Projective structure and motion from image sequences using subspace methods. In M. Frydrych, J. Parkkinen, and A. Visa, editors, *Proceedings of the 10th Scandinavian Conference on Image Analysis, Lappeenranta, Finland*, volume II, pages 963–968, June 1997.

[35] Nick Hollinghurst and Roberto Cipolla. Uncalibrated stereo hand-eye coordination. *Image and Vision Computing*, 12(3):187–192, 1994.

[36] R. Horaud and G. Csurka. Self-calibration and euclidean reconstruction using motions of a stereo rig. In *Proceedings of 6th International Conference on Computer Vision*, pages 96 – 103, Bombay, India, January 1998. IEEE Computer Society Press, Los Alamitos - CA.

[37] R. Horaud and O. Monga. *Vision par ordinateur: outils fondamentaux*. Editions Hermès, Paris, 1993.

[38] Radu Horaud and Fadi Dornaika. Hand-eye calibration. *International Journal of Robotics Research*, 14(3):195–210, June 1995.

[39] Radu Horaud, Fadi Dornaika, and Bernard. Espiau. Visually guided object grasping. *IEEE Transactions on Robotics and Automation*, 14(4):525–532, August 1998.

[40] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge-MA, 1986.

[41] R. A. Horn and C. A. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge-MA, 1994.

[42] Seth Hutchinson, Greg D. Hager, and Peter I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–669, October 1996.

[43] Martin Jaegersand. On-line estimation of visual-motor models using active vision. In *Proceedings of the ARPA Image Understanding Workshop*, 1996.

[44] Martin Jaegersand. Visual servoing using trust region methods and estimation of the full coupled visual-motor jacobian. In *Proceedings of IASTED Applications of Control and Robotics*, pages 105–108, 1996.

[45] Kenichi Kanatani. *Group-theoretical methods in image understanding.* Springer series in information sciences. Springer Verlag, Berlin New-York Paris, 1990.

[46] K. Kinoshita. Visual servoing with uncalibrated cameras. In *The Forth International Conference on Control, Automation, Robotics and Vision*, pages 1705–1709, December 1996.

[47] Bart Lamiroy, Bernard Espiau, Nicolas Andreff, and Radu Horaud. Controlling robots with two cameras: How to do it properly. In *Proceedings of Int. Conference on Robotics and Automation*, pages 2100–2105, San Fransisco - CA, April 2000.

[48] H.C. Longuet-Higgins. A computer program for reconstructing a scene from two projections. *Nature*, 293:133 – 135, September 1981.

[49] Q-T. Luong and Olivier D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43 – 75, 1996.

[50] Yi Ma, Stefano Soatto, Jana Kosecka, and Shankar S. Sastry. Euclidean reconstruction and reprojection up to subgroups. *International Journal of Computer Vision, special issue for David Marr's prize papers*, 38(3):217–227, 2000.

[51] Ezio Malis and Francois Chaumette. 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37((1)):79–97, juin 2000.

[52] Ezio Malis, Francois Chaumette, and Sylvie Boudet. 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, 15((2)):238–250, Avril 1999.

[53] Eric Marchand, A. Rizzo, and Francois Chaumette. Avoiding robot joint limits and kinematic singularities in visual servoing. In *Proceedings of 13th IARP/IEEE International Conference in Pattern Recognition*, volume A, pages 297 – 301, 1996.

[54] Frédérick Martin. Identification numérique du modèle métrique et non-métrique d'un robot par vision stéréo-scopique. Master's thesis, ENSIMAG, UJF, June 1999.

[55] J.M. McCarthy. *An Introduction to Theoretical Kinematics*. MIT Press, Cambridge-MA, 1990.

[56] B.W. Mooring, Z.S. Roth, and M.R. Driels. *Fundamentals of Manipulator Calibration*. John Wiley & Sons, New York, 1991.

[57] Richard M. Murray, Z. Li, and S. Shakar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton; Ann Arbor; London; Tokyo, 1994.

[58] Andreas Ruf and Radu Horaud. Projective translations and affine stereo calibration. In *Proceedings of the Intl. Conf. on Computer Vision and Pattern Recognition, (CVPR'98)*, pages 475 – 481, Santa Barbara, CA, June 1998. IEEE Computer Society.

[59] Andreas Ruf and Radu Horaud. Projective rotations applied to non-metric pan-tilt head. In *Proceedings of the Intl. Conf. on Computer Vision and Pattern Recognition, (CVPR'99)*, volume I, pages 144 – 150, Fort Collins, CO, June 1999. IEEE Computer Society.

[60] Andreas Ruf and Radu Horaud. Rigid and articulated motion seen with an uncalibrated stereo rig. In Tsotsos John, editor, *Proceedings of the 7th International Conference on Computer Vision (ICCV'99)*, volume II, pages 789 – 796, Kerkyra, Greece, September 1999. IEEE Computer Society.

[61] Andreas Ruf and Radu Horaud. Visual servoing of robot manipulators, part "i" : Projective kinematics. *International Journal on Robotics Research*, 18(11):1101 – 1118, November 1999.

[62] Andreas Ruf and Radu Horaud. Visual servoing using projective kinematics. In John J. Hollerbach and Daniel E. Koditscheck, editors, *Proceeding of the International Symposium on Robotics Research, (ISRR'99)*, Lecture Notes in Computer Science, pages 97–104, Salt Lake City - UT, October 1999. Springer.

[63] Andreas Ruf and Radu Horaud. Vision-based guidance and control of robots in projective space. In Vernon David, editor, *Proceedings of the 6th European Conference on Computer Vision (ECCV'00)*, volume II of *Lecture Notes in Computer Science*, pages 50–83, Dublin, Ireland, June 2000. Springer.

[64] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control, the Task Function Approach*. Oxford Science Publications, Oxford, 1991.

[65] Claude Samson, Bernard Espiau, and M. Le Borgne. *Robot Control: The Task Function Approach*. Clarendon Press, Oxford, 1991.

[66] J.M. Selig. *Geometrical Methods in Robotics*. Monographs in Computer Science. Springer, New York, 1996.

[67] S.B Skaar, W.H. Brockman, and R. Hanson. Camera space manipulation. *International Journal of Robotics Research*, 6:20–32, Winter 1987.

[68] S.B. Skaar, W.H. Brockman, and W.S. Jang. Three-dimensional camera space manipulation. *International Journal of Robotics Research*, 9(4):22–39, 1990.

[69] Gerald Sommer, editor. *Geometric Computing with Clifford Algebra: Theory and Applications in Vision and Robotics*. Springer Verlag, November 2000.

[70] Mike Spratling and Robert Cipolla. Uncalibrated visual servoing. In E. Trucco R. B. Fisher, editor, *British Machine Vision Conference*, pages 545–554, 1996.

[71] Peter Sturm and Bill Triggs. A factorization based algorithm for multi-image projective structure and motion. In B. Buxton and R. Cipolla, editors, *Proceedings of European Conference on Computer Vision (Eccv'96)*, volume 1065 of *Lecture Notes on Computer Science*, pages 709 – 720. Springer Verlag, April 1996.

[72] Peter F. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 1100 – 1105, Puerto-Rico, June 1997.

[73] C. Taylor, J. Ostrowski, and S. Jung. Robust visual servoing based on relative orientation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 99)*, pages 574–580, June 1999.

[74] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal on Computer Vision*, 9(2):137–154, 1992.

[75] Martin Tonko and Hans-Hellmut Nagel. Model-based stereo-tracking of non-polyhedral objects for automatic disassembly experiments. *International Journal of Computer Vision*, 37(1):99–118, 2000.

[76] R.Y. Tsai and R.K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Journal of Robotics and Automation*, 5(3):345–358, June 1989.

[77] Thierry Viéville and Olivier D. Faugeras. The first order expansion of motion equations in the uncalibrated case. *Computer Vision and Image Understanding*, 64(1):128 – 146, 1996.

[78] Thierry Viéville and D. Lingrand. Using singular displacements for uncalibrated monocular visual systems. In B. Buxton and R. Cipolla, editors, *Proceedings of European Conference on Computer Vision (ECCV'96)*, volume 1065 of *Lecture Notes in Computer Science*, pages 207 – 216. Springer Verlag, April 1996.

[79] G.-Q. Wei, K Arbter, and Gerd Hirzinger. Active self-calibration of robotic eyes and hand-eye relationships with model identification. *IEEE Transactions on Robotics and Automation*, 14(1):158–166, 1998.

[80] W.J. Wilson, C.C.W. Hulls, and G.S. Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Transaction on Robotics and Automation*, 12(5):684–696, October 1996.

[81] E. Zeidler, editor. *Teubner-Taschenbuch der Mathematik*, volume I. B.G. Teubner, Stuttgart; Leipzig, 1996.

[82] X. Zhuang and Y. Huang. Robust 3D – 3D estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):818–824, August 1994.

[83] Andrew Zisserman, P.A. Beardsley, and I.D. Reid. Metric calibration of a stereo rig. In *Proceedings of IEEE Workshop on Representation of Visual Scences*, pages 93 – 100, Cambridge-MA, June 1995.