



Oriented filters for feature extraction in digital Images : Application to corners detection, Contours evaluation and color Steganalysis

Hasan Abdulrahman

► To cite this version:

Hasan Abdulrahman. Oriented filters for feature extraction in digital Images : Application to corners detection, Contours evaluation and color Steganalysis. Image Processing [eess.IV]. Université Montpellier, 2017. English. NNT : 2017MONT077 . tel-01948473

HAL Id: tel-01948473

<https://theses.hal.science/tel-01948473>

Submitted on 7 Dec 2018

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

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

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Informatique

École doctorale I2S

Unité de recherche LGI2P

Oriented Filters for Feature Extraction in Digital Images: Application to Corners Detection, Contours Evaluation and Color Steganalysis

Présentée par Hasan Kareem ABDULRAHMAN

Le 17 Novembre 2017

Sous la direction de Philippe MONTESINOS

Devant le jury composé de:

M. William PUECH, Professeur, Université de Montpellier, Lab. LIRMM.

M. Vincent CHARVILLAT, Professeur, Université de Toulouse, Lab. IPAL.

M. Philippe CARRE, Professeur, Université de Poitiers, Lab. XLIM.

M. Krzysztof SZCZYPIORSKI, Professeur, Université de Warsaw, Pologne.

M. Philippe MONTESINOS, HDR, MCF, EMA, Lab. LGI2P.

M. Baptiste MAGNIER, MCF, EMA, Lab. LGI2P.

Examineur, Président de jury

Rapporteur

Rapporteur

Examineur

Directeur de thèse

Co-Encadrant



**UNIVERSITÉ
DE MONTPELLIER**

ABSTRACT

Interpretation of image contents is a very important objective in image processing and computer vision. Wherefore, it has received much attention of researchers. Digital image contains a lot of information such as color, shapes, edges, corners, size, orientation and etc. Moreover, contours include the most important structures in the image. In order to extract features contour of an object, we must detect the edges of that object. Edge detection, remains a key point and very important step in wide range of applications such as: image restoration, enhancement, steganography, watermarking, image retrieval, recognition, compression, and etc. An efficient boundary detection method should create a contour image containing edges at their correct locations with a minimum of misclassified pixels. However, the performance evaluation of the edge detection results is still a challenging problem. The digital images are sometimes modified by a legal or illegal data in order to send special or secret data. These changes modify slight values of this image. In order to be less visible, most of the steganography methods modify the pixel values in the edge/texture image areas. Therefore, it is important to detect the presence of hidden data in digital images. This thesis is divided into two main parts. The first part, deals with filtering edge detection, contours evaluation and corners detection methods. More deeply, there are five contributions are presented in this part: first, proposed a new normalized supervised edge map quality measure. The strategy to normalize the evaluation enables to consider a score close to 0 as a good edge map, whereas a score 1 translates a poor segmentation. Second, proposed a new technique to evaluate filtering edge detection methods involving the minimum score of the considerate measures. Moreover, build a new ground truth edge map labelled in semi-automatic way in real images. Third, proposed a new measure takes into account the distances of false positive points to evaluate an edge detector in an objective way. Finally, proposed a new approach for corner detection based on the combination of directional derivative and homogeneity kernels. The proposed approach remains more stable and robust to noise than ten famous corner detection methods. The second part, deals with color image steganalysis, based on a machine learning classification. More deeply, there are three contributions are presented in this part: first, proposed a new color image steganalysis method based on extract color features from correlations between the gradients of red, green and blue channels. Since these features give the cosine of angles between gradients. Second, proposed a new color steganalysis method based on geometric measures obtained by the sine and cosine of gradient angles between all the color channels. Finally, proposed a new approach for color image steganalysis based on steerable Gaussian filters bank. All the three proposed methods in this part, provide interesting and promising results by outperforming the state-of-art color image steganalysis.

Keywords : Oriented filters, edge detection, corners detection, ground truth, supervised evaluation, steganography, steganalysis.

RESUMÉ

L'interprétation du contenu de l'image est un objectif très important dans le traitement de l'image et la vision par ordinateur. Par conséquent, plusieurs chercheurs y sont intéressés. Une image contient des informations multiples qui peuvent être étudiées, telles que la couleur, les formes, les arêtes, les angles, la taille et l'orientation. En outre, les contours contiennent les structures les plus importantes de l'image. Afin d'extraire les caractéristiques du contour d'un objet, nous devons détecter les bords de cet objet. La détection de bords est un point clé dans plusieurs applications, telles que : la restauration, l'amélioration de l'image, la stéganographie, le filigrane, la récupération, la reconnaissance et la compression de l'image, etc. Toutefois, l'évaluation de la performance de la méthode de détection de bords reste un grand défi. Les images numériques sont parfois modifiées par une procédure légale ou illégale afin d'envoyer des données secrètes ou spéciales. Afin d'être moins visibles, la plupart des méthodes stéganographiques modifient les valeurs de pixels dans les bords/textures de parties de l'image. Par conséquent, il est important de détecter la présence de données cachées dans les images numériques. Cette thèse est divisée principalement en deux parties. La première partie discute l'évaluation des méthodes de détection des bords du filtrage, des contours et des angles. En effet, cinq contributions sont présentées dans cette partie : d'abord, nous avons proposé un nouveau plan de surveillance normalisée de mesure de la qualité. En second lieu, nous avons proposé une nouvelle technique pour évaluer les méthodes de détection des bords de filtrage impliquant le score minimal des mesures considérées. En plus, nous avons construit une nouvelle vérité terrain de la carte de bords étiquetée d'une manière semi-automatique pour des images réelles. En troisième lieu, nous avons proposé une nouvelle mesure prenant en compte les distances de faux points positifs pour évaluer un détecteur de bords d'une manière objective. Enfin, nous avons proposé une nouvelle approche de détection de bords qui combine la dérivée directionnelle et l'homogénéité des grains. Notre approche proposée est plus stable et robuste au bruit que dix autres méthodes célèbres de détection. La seconde partie discute la stéganalyse de l'image en couleurs, basée sur l'apprentissage automatique (machine learning). En effet, trois contributions sont présentées dans cette partie : d'abord, nous avons proposé une nouvelle méthode de stéganalyse de l'image en couleurs, basée sur l'extraction de caractéristiques de couleurs à partir de corrélations entre les gradients de canaux rouge, vert et bleu. En fait, ces caractéristiques donnent le cosinus des angles entre les gradients. En second lieu, nous avons proposé une nouvelle méthode de stéganalyse de l'image en couleurs, basée sur des mesures géométriques obtenues par le sinus et le cosinus des angles de gradients entre tous les canaux de couleurs. Enfin, nous avons proposé une nouvelle méthode de stéganalyse de l'image en couleurs, basée sur une banque de filtres gaussiens orientables. Toutes les trois méthodes proposées présentent des résultats intéressants et prometteux en avançant l'état de l'art de la stéganalyse en couleurs.

Mots clés : Filtres orientés, détection de contours, détection de coins, ground truth, évaluation supervisée, stéganographie, stéganalyse.

ACKNOWLEDGEMENTS

Completing the PhD and writing this thesis was an amazing journey that would not have been possible without supporting of the Iraqi Ministry of Higher Education and Scientific Research, my family and many people too.

First and Foremost, I would like to express my special appreciation and thanks to my supervisor Dr. Philippe Montesinos, I would like to thank you for offered me invaluable assistance, support, guidance, knowledge and for allowing me to grow as a research scientist throughout my thesis. I appreciate all his contributions of time and ideas to make my PhD experience productive and stimulating. Besides my supervisor, my gratitude is also extended to my co-supervisor Dr. Baptiste Magnier, for helpful guidance, knowledge, effort, and support during my PhD thesis. I feel privileged for being mentored by both my supervisors and look forward for future collaborations.

I would like to thank the jury members of my Ph.D thesis. Special thank goes to my PhD thesis reviewers, Prof. Vincent Charvillat and Prof. Philippe Carré, who accepted to read and evaluate my thesis and my research works. Their reports reflect the effort and time they have devoted to it, I am very grateful. Also, I would like to express my appreciation and thank my examination committee Prof. William Puech head of ICAR and Prof. Krzysztof Szczypiorski for their time and efforts.

I would like to express my great thanks to, prof. Jacky Montamain head of LGI2P lab, prof. Marianne Huchard, M. Laurent Champion, Mme. Edith Teychene, Mme. Valerie Roman and M. Nicolas Serrurier, for their help and support for making this research a possibility. My thanks also goes to all the members of the Ecole des Mines d'Ales and especially LGI2P lab. It has really been a pleasure to work with you all.

I would like to express my appreciation to Dr. Marc Chaumont for helpful discussions, valuable suggestions during my work in the steganography domain.

My greatest appreciation and gratitude goes to Iraqi Ministry of Higher Education and Scientific Research for their trust and financial support my PhD thesis.

I would like to thank all my friends, Ibrahim Najmaldeen, Nooraldeen Fathallah, Akram Goolam, Simko Shareef, Adel Azeez, Muaffaq Yahia, Mahmoud Khedr, Arkaan Mohamad, Fouad Jabaar, Asmahan Ahmed, Razzaw Kamal, Salahaldeen Najmaldeen, Ahmed Shareef, and Ali Adnan. For their confidence and for support me.

Last, but certainly not least, and most importantly, I would express my love and gratitude to my wife Amel, my beautiful daughter Dalya, my big family my mother, my brothers and my sisters for their distant support, care and encouragement me. Always you all stood by me and shared with me all the great moments of my life.

Hasan ABDULRAHMAN

LIST OF PUBLICATIONS

The contributions and results of this thesis have been published and presented in the (7) international conferences, workshops and (2) journals.

- Hasan Abdulrahman, Baptiste Magnier and Philippe Montesinos. ***Oriented half kernels for corner detection***. In Proceedings of the 25th European Signal Processing Conference (EUSIPCO 2017), Kos Island, Greek, 28 August - 2 September, ISBN 978-0-9928626-7-1, pages 808-812, **2017**.
- Hasan Abdulrahman, Baptiste Magnier and Philippe Montesinos. ***A New Objective Supervised Edge Detection Assessment using Hysteresis Thresholds***. In the First International Workshop on Brain-Inspired Computer Vision (WBICV 2017), held as part of the conference (ICIAP 2017), Catania, Sicily (Italy), 11 - 15 September, Proceedings, volume 10590, 11 pages, chapter 1, of the Lecture Notes in Computer Science Series. Springer, **2017**.
- Baptiste Magnier, Hasan Abdulrahman and Philippe Montesinos. ***Détection de coins par combinaison de filtres asyétriques orientés***. the 25th edition (GRETSI 2017), Juan-les-Pins, France, 5-8 September, **2017**.
- Hasan Abdulrahman, Baptiste Magnier and Philippe Montesinos. ***A new normalized supervised edge detection evaluation***. In the 8th Iberian International Conference on Pattern Recognition and Image Analysis (IbPRIA 2017), Faro, Portugal, published in the Lecture Notes in Computer Science Series, pages 203-213, 20-23 June, Springer, **2017**.
- Hasan Abdulrahman, Baptiste Magnier and Philippe Montesinos. ***From contours to ground truth: How to evaluate edge detectors by filtering***. In the 25th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2017), Pilsen, Czech Republic, 29 May - 2 June, 2017. **Selected as one of the best papers in the conference WSCG**, Published in the **Journal of WSCG**, Vol. 25, No.1-2, pages 133-142, ISSN 1213-6972, ISBN 978-80-86943-43-5, **2017**.
- Hasan Abdulrahman, Marc Chaumont, Philippe Montesinos, and Baptiste Magnier. ***Color image steganalysis based on steerable gaussian filters bank***. In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security ACM(IHMMSec 2016). Acceptance rate = 36.2 %, pages 109–114, Vigo, Galicia, Spain, 20-22 June, **2016**.
- Hasan Abdulrahman, Marc Chaumont, Philippe Montesinos, and Baptiste Magnier. ***Color images steganalysis using rgb channel geometric transformation measures***. Published in the **Wiley Journal** on Security and Communication Networks, DOI: 10.1002/sec.1427: 12 pages, Wiley, Feb. **2016**.
- Hasan Abdulrahman, Marc Chaumont, Philippe Montesinos, and Baptiste Magnier. ***Color image steganalysis using correlations between rgb channels***. In Proceedings of the 10th International Conference on Availability, Reliability and Security (ARES 2015), 4th International Workshop on Cyber Crime (IWCC), Toulouse, France, pages 448–454, IEEE, 24-28 August, **2015**.

CONTENTS

List of Tables	xiii
List of Figures	xv
1 Introduction	1
1.1 Introduction	2
1.2 Motivations	3
1.3 Key Contributions	5
1.4 Organization of this thesis	6
 I Overview on Digital Image Analysis Concepts	 7
2 Digital Image Analysis	9
2.1 Introduction	10
2.2 Digital Images	10
2.2.1 Color Images	12
2.2.2 CFA Bayer Demosaicing	12
2.2.3 Decomposition of Color Space	14
2.3 Noise in Digital Image	18
2.3.1 Models of Image Noise	20
2.4 Applications of Digital Image Analysis	21
2.5 Summary	23
 II Image Feature Detectors and Evaluation	 25
3 Edge Evaluation Methods and Corner Detection	27
3.1 Introduction	28
3.2 Low-level Feature Extraction	29
3.2.1 Global and Local Features	29
3.3 Edge Definition	30
3.3.1 Edge Labelling process	31
3.3.2 Non-maximum suppression technique	31
3.3.3 First and Second Order Derivative Edge Detection	32
3.3.4 Two-Dimensional Gaussian Derivative	34
3.4 Traditional edge operators	35
3.4.1 Sobel edge detector	35

3.4.2	Canny edge detector	36
3.5	Filter banks	37
3.5.1	Steerable Gaussian Filter	38
3.5.2	Anisotropic Gaussian kernels	40
3.5.3	Half Gaussian kernel	40
3.6	Summary	46
4	New Normalized Supervised Edge Detection Evaluation Method	47
4.1	Introduction	48
4.2	Supervised image contour evaluation	48
4.2.1	Error measures involving the confusion matrix only	49
4.2.2	Assessment involving distances of misplaced pixels	50
4.2.3	Normalization of the edge detection evaluation:	52
4.3	A new edge detection assessment measure	53
4.4	Experimental results	53
4.5	Summary	62
5	From Contours to Ground Truth: How to Evaluate Edge Detectors	63
5.1	Introduction	64
5.2	Supervised error measures	65
5.3	How to create precise ground truth images ?	70
5.3.1	Ground truth images	72
5.3.2	Minimum of the measure	74
5.4	Experimental results	75
5.4.1	Synthetic images	75
5.4.2	Real images	84
5.5	Summary	98
6	Objective Supervised Edge Detection using Hysteresis Thresholds	99
6.1	Introduction	100
6.2	Supervised Measures for Image Contour Evaluations	101
6.2.1	Distances of misplaced pixels	101
6.2.2	A new objective edge detection assessment measure:	103
6.2.3	Minimum of the measure and ground truth edge image	104
6.3	Experimental results	105
6.4	Summary	109
7	Oriented Half Kernels for Corner Detection	111
7.1	Introduction	112
7.1.1	Corner detection techniques	112
7.1.2	Oriented kernels and edges directions	114
7.2	A new method of corner extraction	115
7.2.1	Oriented filter of grayscale homogeneity	116
7.2.2	A combination of homogeneity and edge strength	117

7.2.3	Angle selection and corner extraction	117
7.3	Experiments and evaluation	118
7.4	Summary	125

III Detection of Hidden Data in Digital Images 127

8 Overview on Steganography and Steganalysis 129

8.1	Introduction	130
8.2	Steganography and steganalysis through these ages	131
8.3	Cryptography, Watermarking and Steganography	133
8.4	The prisoner's problem (steganography/steganalysis)	137
8.5	Digital Image Steganography	138
8.6	Digital Image Steganalysis	142
8.6.1	Color image Steganalysis	144
8.6.2	The ensemble classifiers	147
8.7	Summary	148

9 Color Channel Correlation and Geometric Steganalysis 151

9.1	Introduction	152
9.2	Methodology description	152
9.2.1	<i>RGB</i> Channel Correlation	153
9.2.2	Mirror transformations	154
9.2.3	Complete feature set	155
9.3	Building Image Dataset	156
9.4	Experimental results	158
9.4.1	Embedding methods	158
9.4.2	Results and Discussion	159
9.5	Summary	162

10 Steganalysis Based on Steerable Gaussian Filter 165

10.1	Introduction	166
10.2	Methodology description	166
10.2.1	Steerable Gaussian filters	167
10.2.2	Complete feature set	170
10.3	Experimental results	170
10.3.1	Performance of our proposed method	171
10.4	Summary	174

IV Conclusion and Perspectives	175
11 Conclusion and Perspectives	177
11.1 Conclusion	178
11.2 Perspectives	179
12 Résumé en Français	181
12.1 Introduction	182
12.2 Motivations	184
12.3 Contributions	185
12.4 Conclusion	186
12.5 Perspectives	186
Bibliographie	189

LIST OF TABLES

2.1	The relationship between the bit depth with available number of colors. . .	11
4.1	List of error measures, $k = 1$ or $k = 2$ are the most common values.	51
5.1	List of error measures involving only statistics.	65
5.2	List of normalized error measures compared in this work, with the parameter $\kappa \in]0; 1]$	66
5.3	List of non-normalized error measures. In the literature, the most common values are $k = 1$ or $k = 2$	67
5.4	Comparison of scores of dissimilarity measures using a ground truth from [MFTM01] (Fig. 5.3 (b)) image and a constructed ground truth by a semi-automatic way. Contour images and curves for all the measures are available in the supplementary material.	84
8.1	The difference between cryptography, watermarking and steganography. . .	136
9.1	Features description with their dimmensionalities corresponding to q and T . .	152
9.2	Numerical values of the average testing error \bar{P}_E and the detection rate $\mathcal{P}_D\%$ for three steganography methods. For easier navigation the dark gray background column presents our first method [ACMM15] and the light gray background column presents the our second proposed method [ACMM16b].	160
9.3	Our proposed method 2 detection rate of S-UNIWARD, WOW and Synch-HILL steganography methods at 0.2 bpc and 0.4 bpc payload embedding in the green channel compares with equal embedding in three channels. .	160
10.1	The probability of error \bar{P}_E to determine the efficient standard deviation (σ) and angle step ($\Delta\theta$) employed for steerable Gaussian filters using S-UNIWARD steganography method payload 0.3 bpc.	171
10.2	Error probability \bar{P}_E and the detection rate $\mathcal{P}_D\%$ for four steganalysis methods.	172

LIST OF FIGURES

2.1	Different version of the same image; (a) 24-bit color, (b) 8-bit color, (c) 8-bit grayscale, (d) 1-bit (black-and-white), and the histogram of each image.	11
2.2	Light beam splitting by a trichroic prism in a standard three CCD camera sensor.	13
2.3	The CFA Bayer pattern with red-green-red-green phase in the first row, and four corroborative CFA patterns.	13
2.4	Representation of the RGB color space as a three-dimensional unit cube.	15
2.5	Representation of the CMY color model as a three-dimensional unit cube.	15
2.6	Representation of the YCbCr color model.	16
2.7	Representation of the HSI color model.	17
2.8	Representation of the CIEL*a*b* color model.	18
2.9	Image acquisition using a typical digital camera.	19
2.10	Probability density function for: a) Gaussian noise model, b) Salt and pepper noise model.	20
3.1	The different types of edges and result of a convolution with a $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ mask (absolute value).	30
3.2	Examination if the pixel P in (b) is a local maxima of the magnitude of the gradient in the direction of the gradient is done by interpolating the gradient magnitude at pixels M and K in (b).	32
3.3	2D-dimensional Gaussian first and second order derivative operators give receptive of x and y directions.	34
3.4	The Sobel operator as a pair of 3×3 convolution kernels in X and Y directions.	35
3.5	All Sobel edge detector process steps.	36
3.6	All Canny edge detector process steps.	38
3.7	(a) Set of steerable Gaussian filters $\mathcal{G}_{\sigma,\theta}$ with $(\sigma=1.5, \Delta\theta = 10^\circ)$, starting from upper left $\theta = 0^\circ$. (b) The steerable filter banks block diagram. Figure obtained from [Kri14].	39
3.8	Anisotropic orientation kernel $G_{\sigma_x, \sigma_y}(x, y)$ and example of orientations $G'_{\sigma_x \sigma_y}(x, y)$. (a) AGK filter. (b) Anisotropic Oriented kernel by 0° , (b) Oriented kernel by 60° .	40
3.9	Address the problem on Anisotropic Gaussian kernel. This figure obtained from [MMD11a]. (a) Application of Anisotropic Gaussian kernel at the edge and corner of an object. (b) Application of Anisotropic half Gaussian kernel at at the edge and corner of the same object.	41
3.10	The figures (b) and (c) obtained from [MMD11a] show the HGK with full derivative and half smoothing. (b) example of HGK orientations. (d) example of discrete HGK.	41
3.11	Methodology of the Half Gaussian kernels process steps.	42

3.12	Comparison between Anisotropic kernel and Half filter derivatives at different orientations θ , (a)Anisotropic filter with $\sigma_x=1.5$, $\sigma_y=7.5$. (b) Half Gaussian filter with $\lambda=1.5$, $\mu=3.75$	43
3.13	Example for real image: differences edge detector methods result.	44
3.14	Comparison differences edge detector methods for blurry edge noise.	45
3.15	Comparison differences edge detector methods for blurry corner noise.	45
4.1	Evaluations issued of a confusion matrix can be the same for different D_c . For the two candidate edge images, number of FPs and number of FNs are the same.	49
4.2	The scores of the over-segmentation evaluations are higher for I_t whereas I_t is more closer visually to G_t than I_c	50
4.3	Measures scores in function of the over-segmentation in the contour area.	54
4.4	Evolution of the dissimilarity measures in function of the FPs addition.	54
4.5	Measure scores in function of the FNs addition and the edge translation.	55
4.6	Dissimilarity measure scores in function of addition of both the FNs and FPs.	56
4.7	Dissimilarity measure scores in function of the size of the original square.	56
4.8	Comparison measures of different edge detections involving different edge detectors. All the results are normalized and, compared to the ground truth, a score close to 0 indicates a good edge map whereas a score 1 translates a poor segmentation. Our proposed measure Ψ indicate a good measurement value.	58
4.9	Comparison measures of different edge detections involving different edge detectors. All the results are normalized and, compared to the ground truth, a score close to 0 indicates a good edge map whereas a score 1 translates a poor segmentation. Our proposed measure Ψ indicate a good measurement value.	59
4.10	Comparison measures of different edge detections involving different edge detectors. All the results are normalized and, compared to the ground truth, a score close to 0 indicates a good edge map whereas a score 1 translates a poor segmentation.	60
4.11	Comparison measures of different edge detections involving different edge detectors. All the results are normalized and, compared to the ground truth, a score close to 0 indicates a good edge map whereas a score 1 translates a poor segmentation. Our proposed measure Ψ indicate a good measurement value.	61
5.1	Ground truth vs. desired contour. In (b), D_c is contaminated with 6 FPs and 4 FNs.	64

5.2	Evolution of dissimilarity measures in function of the the distance of the false positive/negative points. A vertical line of false positive points (b) or false negative points (d) is shifted by a maximum distance of 16 pixels and the measure scores are plotted in function of the displacement of the desired/undesired contour.	69
5.3	Edge detection after the non-maximum suppression [RT71] and comparison with a ground truth image (green cells for the minimum measure and better edge detector filter).	71
5.4	Synthetic data with a 1 pixel width gray around each shape: value of white pixels = 1, values of black pixels = 0, values of gray pixels = 0.5.	72
5.5	Image of our database are built after an edge detection involving a $[-1 \ 0 \ 1]$ mask and concluded by hand.	73
5.6	The most relevant edge map for a dissimilarity measure is indicated by its minimum score.	74
5.7	Scores of the measures depending on the threshold of the thin gradient image [Can86a].	75
5.8	Synthetic images corrupted by a Gaussian noise with the associated ground truth edge map.	76
5.9	Images of our database are built in a semi-automatic way, after an edge detection involving $[-1 \ 0 \ 1]$ masks and concluded by hand.	77
5.10	Synthetic image SNR= 4 dB. (The original image is available in Fig. 5.8 (a) and the ground truth as shown in 5.8(c)). Best maps for the compared edge detection evaluations using our proposed measure Ψ	78
5.11	Synthetic image SNR= 3.3 dB (The original image is available in Fig. 5.8 (b) and the ground truth as shown in 5.8(c)). Best maps for the compared edge detection evaluations using our proposed measure Ψ	79
5.12	Synthetic image SNR= 4 dB: Best maps for the compared edge detection evaluations.	80
5.13	Synthetic image SNR= 4 dB: Best maps for the compared edge detection evaluations.	81
5.14	Synthetic image SNR= 3.3 dB: Best maps for the compared edge detection evaluations.	82
5.15	Synthetic image SNR= 3.3 dB: Best maps for the compared edge detection evaluations.	83
5.16	Real image: Best maps for the compared edge detection filters using our proposed measure Ψ . The ground truth edge image is computer by the consensus from human-labellers.	86
5.17	Real image: Best maps for the compared edge detection evaluations. The ground truth edge image is computer by the consensus from human-labellers.	87
5.18	Real image: Best maps for the compared edge detection evaluations. The ground truth edge image is computer by the consensus from human-labellers.	88

5.19	Real image: (The original image is available in Fig. 5.9 and the ground truth edge image comes from our database). Best maps for the compared edge detection filters using our proposed measure Ψ	89
5.20	Real image: Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.	90
5.21	Real image: Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.	91
5.22	Real image 36 (The original image is available in Fig. 5.9 and the ground truth edge image comes from our database). Best maps for the compared edge detection filters using our proposed measure Ψ	92
5.23	Real image 36 : Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.	93
5.24	Real image 36 : Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.	94
5.25	Real image parkingmeter (The original image is available in Fig. 5.9 and the ground truth edge image comes from our database). Best maps for the compared edge detection filters using our proposed measure Ψ	95
5.26	Real image parkingmeter : Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.	96
5.27	Real image parkingmeter : Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.	97
6.1	Gradient magnitude and orientation computation for a scalar image I and example of hysteresis threshold applied along a contour chain. I_θ represents the image derivative using a first-order filter at the θ orientation (in radians).	100
6.2	Different D_c : FPs and number of FNs are the same for D_1 and for D_2	102
6.3	Results of evaluation measures and images for the experiments.	103
6.4	Number of FNs penalizes λ and computation of a measure minimum score.	104
6.5	Synthetic image SNR= 3.3 dB. (The original image is available in Fig. 5.8 (b) and the ground truth as shown in 5.8(c)). Comparison of best maps and minimum scores for different evaluation measures using our proposed measure λ	106
6.6	Comparison of best maps and minimum scores for different evaluation measures. The bars legend is presented in Fig. 6.7. G_t and original image are available in Fig. 6.3.	107
6.7	Comparison of best maps and minimum scores for different evaluation measures. G_t and the original real image are presented in Fig. 6.3.	108
7.1	Corner detected involving several methods. The standard deviation used for the image derivatives is the same as for the structure tensor J_ρ : $\sigma = \rho = 1$. For the developed method: $\sigma_\eta = 1$ and $\sigma_\xi = 3$, $L = 3$ and $P = 5$	113

7.2	Different discretized 2D derivative Gaussian kernels and representation of the IRON filter. (b), (c) and (d) with $\sigma_\eta = 1$ and $\sigma_\xi = 3$. The Anisotropic endstop filter is equivalent to the derivative of the AGK along the Y direction.	115
7.3	Modulus of the energy of the different oriented kernels and their combinations (in degrees and normalized signals).	116
7.4	Direction field. (a) Cropped image of Fig. 7.1(a).	117
7.5	The minimum and maximum of the signal \mathcal{S} corresponds to the two directions of the edges and β to the angular sector between the θ_1 and θ_2 directions.	118
7.6	Comparison the <i>RMSE</i> as a function for the corners error detection between our method with several corner detection methods.	119
7.7	Corner detection using template or isotropic structure tensors obtains mistakes or corner displacement. The standard deviation σ for the image derivatives and for the structure tensor J_ρ are the same: $\sigma = \rho = 1$. For (h), the convolution masks a (for example $k = 0.004$). Note: [True corner = +, False corner = +, and Forget corner = +].	120
7.8	Corner detection using template or isotropic structure tensors obtains mistakes or corner displacement. The standard deviation σ for the image derivatives and for the structure tensor J_ρ are the same: $\sigma = \rho = 1$. For (h), the convolution masks a (for example $k = 0.004$). Note: [True corner = +, False corner = +, and Forget corner = +].	121
7.9	Comparison the Root Mean Square Error (RMSE) as a function for the corners error detection (a) our method with different $\Delta\theta$, (b) our method with different σ with $\Delta\theta = 5$	122
7.10	RMSE as a function of the noise level. (c)-(f): Corner detected involving several methods with $\sigma_\eta = 1$ et $\sigma_\xi = 3$	122
7.11	Corner detected involving several methods. The standard deviation used for the image derivatives and for the structure tensor J_ρ are: $\sigma = 1.5$ and $\rho = 1$. For the developed method: $\sigma_\eta = 1.5$ and $\sigma_\xi = 5$, $L = 3$ and $P = 5$	123
7.12	Corner detection using template or isotropic structure tensors obtains mistakes or corner displacement. The standard deviation σ for the image derivatives and for the structure tensor J_ρ are the same: $\sigma = \rho = 1$. For (h), the convolution masks a (for example $k = 0.004$)	124
8.1	The yearly count of research articles on the Google scholar the keywords steganography.	131
8.2	Examples of ancient steganography models.	132
8.3	Examples of ancient steganography using Germany Microdots.	132
8.4	The general classification of information hiding methods.	134
8.5	The general model for Simmons' "prisoners problem".	137
8.6	Types of models based on the behavior of the prison warden Eve.	138
8.7	The block diagram of steganography.	139

8.8	Architecture of color image steganography by embed messages into color images by decomposing the R , G and B channels with channel-dependent on the same payload partition.	142
8.9	Block diagram of universal steganalysis approach.	144
9.1	Features extraction: Cosine of the gradient angles [ACMM15].	154
9.2	Rotation angle between two channel gradients $\cos(\alpha_1) = \cos(\alpha_2)$ but $\sin(\alpha_1) = -\sin(\alpha_2)$ $\cos(\theta_1) = \cos(\theta_2)$ but $\sin(\theta_1) = -\sin(\theta_2)$. Sine is essential to determine the direction of the rotation.	155
9.3	Features of extraction: Sine of the gradients angles extracting information from the direction of the local rotation.	156
9.5	The preprocessing steps for building our database depending on the CFA idea.	157
9.6	Average testing error \bar{P}_E as a function of the payload for (a) S-UNIWARD, (b) WOW and (c) WOW steganography methods, comparison between the steganalysis methods (Color Rich Model, CFA-aware features steganalysis, our method 1 [ACMM15] and our method 2 [ACMM16b]).	161
9.7	ROC curves using our proposed method 2 feature set, for (a) S-UNIWARD, (b) WOW and (c) Synch-HILL steganography methods for payloads 0.2 bpc (up) and 0.4 bpc (down), to compare the detectability when embedding messages in only one channel with embedding messages spread in all channels.	161
9.8	This table obtained from [TLLH16] shown that our approach SCCRM [ACMM15] can better detect the synchronize inter-channel steganography than Color Rich Model CRM steganalysis method [GFC14].	162
10.1	Features extraction: the image derivatives are extracted at orientation $(\theta_m + 90^\circ)$ $[180^\circ]$ in each channel separately to compute a gradient and to estimate precise edges directions.	168
10.2	θ_m and η directions.	169
10.3	Positions of the steerable filters at the level of the edges to build the gradient image $\ \nabla I\ $ and image derivative $I_{\sigma, (\theta_m + 90^\circ) [180^\circ]}(x, y)$	170
10.4	Error probability \bar{P}_E as a function of the payload for three steganography methods.	173

1

INTRODUCTION

This chapter highlights briefly the motivation and the goals of this thesis. Furthermore, we form the main key contributions in the context of the computer vision and information security.

Contents

1.1	Introduction	2
1.2	Motivations	3
1.3	Key Contributions	5
1.4	Organization of this thesis	6

1.1 Introduction

In computer science, all digital image systems, especially automated information processing structures, must be evaluated before being developed, principally for industrial applications or medical data. Image processing and analysis are areas which have been widely investigated then there is a torrential literature covering different aspects of processing images. The digital image processing deals with developing a digital system that performs operations on a digital image by using a computer [GW02]. The interest in digital image processing methods is motivated by two major application tasks. The first is the improvement of pictorial information for the human interpretation, while the second is data processing for autonomous machine perception [Dou09].

A digital image contains different information from the scene, such as objects, color, and orientation. The discrimination of the objects from their background is the first problem that is performed before any further processing. In order to extract the contour of an object, edges forming that object must be detected, and this fact reveals the constitutional importance of edge detection in computer vision and image processing. Edge detection features support wide range of applications such as recognition, compression, image enhancement, restoration, registration, retrieval, watermarking, steganography, steganalysis and etc. [OH10].

In the literature, there is a considerable importance of research on various techniques for performing edge detection. In general, an edge detection method can be divided into three stages. First, a noise reduction process is performed in order to gain better performance of edge detection. This noise reduction is usually achieved by performing a low-pass filter because the additive noise is normally a high-frequency signal. Since, the edges are high-frequency signals, then it can possibly be removed during the noise reduction process. Hence, the selection of the appropriate filter with best parameter is important to preserve the edge information. Therefore, this thesis details the source of noise in digital image construction pipeline. In the second stage, a high-pass derivative filter such as a oriented filter is usually performed to find the edges. Finally, an edge localization process is performed to identify the real edges, which are distinguished from those similar responses caused by noise.

Part of this thesis focuses on edge detectors with evaluation measure methods and corners detection. On one hand, several reference based boundary detection evaluations are detailed, pointing their advantages and disadvantages through concrete examples of edge images, and applied toward an objective assessment. On the other hand, a new supervised edge map quality measure are proposed. Also, this thesis presents and demonstrate how to build a new ground truth edge map which is labeled as a semi-automatic way in real images. In order to evaluate the performance of filtering step/ramp edge detectors, a

technique is proposed to evaluate filtering edge detection methods involving the minimum score of the consideration measures.

In this thesis, we concentrate on oriented filters such as steerable Gaussian filter, Half Gaussian Kernel (HGK) or causal Gaussian filter. Furthermore, we proposed a new corner detection method by combining two filters: the Half Gaussian kernel with an asymmetric filter computing the homogeneity along edges. On one hand, the maxima responses of the HGK indicates the directions (2π periodic) of the edges. On the other hand, the oriented variance determines if the directions of the maxima of the HGK corresponds to edges or other types of pixels (texture, homogeneous region etc.).

Digital image steganalysis can thus be considered as a pattern recognition process and focuses on two class classifications: original image and stego image (image with hidden message), to decide which class a test image belongs to. The key issue for steganalysis just like for pattern recognition is feature extraction. The features should be sensitive to the data hiding process. In other words, the features should be rather different for the image without hidden message and for the stego image.

Furthermore, we developed three color image steganalysis methods based on feature extraction and machine learning, we devoted part of this thesis for these three methods. The first method [ACMM15], combined features obtained on channels independently (color rich model [GFC14]) with features obtained from correlation between the color channels. The second proposed method [ACMM16b], uses two types of features, computed between color image channels. The first type of features reflects local Euclidean transformations and the second features reflects mirror transformations. These geometric measures are obtained by the sine and cosine of gradient angles between all the color channels. The third proposed method [ACMM16a], uses orientation filter bank to detect hidden messages in color image which have been carried out within extracting features from color images and machine learning. More precisely , this method uses steerable Gaussian filter, the Gaussian filters are angled in different directions to precisely compute the tangent of the gradient vector. Then, the gradient magnitude and the derivative of this tangent direction are estimated. This refined method of estimation enables us to expose the minor changes that have occurred in the image when a message is embedded. We computed a *tangent vector* of contour for each pixel and for each channel. This *tangent vector* corresponds to the edge direction and is orthogonal to the gradient magnitude image of each color channel.

1.2 Motivations

In image processing tasks, edge detection remains a key point in many applications. Boundaries include the most important structures of the image, and an efficient boundary

detection method should create a contour image containing edges at their correct locations with a minimal of misclassified pixels. Edges are an example of a low-level feature, while high-level features are defined in terms of the object in the image. The contours are points in the projected image of an object where there is a discontinuity of depth or a discontinuity of surface orientation on the object.

In digital images, corners and junctions constitute important landmarks because they get affluent information like delimitation of objects. A pixel is considered as a corner or a junction when two or more edges meet each other and it refers to the point where several different contour orientations exist. Many successful feature extraction approaches are based on interest points [Lap05], such as corners, to allow features to be localised in two dimensions or in 3D dimensions [SB11]. There are different algorithms have been developed in the past, but a few of them give an objective performance comparison. wherefore, we presents in this thesis different comparison and proposed a new technique to detect the corners, more stable and robust to noise than the comparative methods.

Contour feature extraction remains a very useful preprocessing step in image segmentation, reconstruction, interpretation and tracking [BdKM⁺04, AMFM11]. An efficient boundary detection method should create a contour image containing edges at their correct locations with a minimum of misclassified pixels [BM98]. The evaluation process should produce a result that correlates with the perceived quality of the edge image, which is relied on human judgment. In other words, a reliable edge map should characterize all the relevant structures of an image. On the other hand, a minimum of spurious pixels or holes (oversights) must be created by the edge detector at the same time. Therefore, an efficient evaluation can be used to assess and improve an algorithm, or to optimize edge detector parameters [HSSB97].

The oriented filters have been used to analyse local feature orientation in imagery. Such filters are typically based on directional derivatives. Edge and corner detection methods using orientation filter bank are estimates the edges accurately. Therefore, oriented filter bank are used in many vision and image processing approaches, such as edge detection, texture analysis, image compression, object detection, steganography and image enhancement [NA12]. In order to be less visible, most of the steganographic methods modify the pixel values in the texture/edge areas [PFB10, HFD14b, LWHL14]...

For many years, information hiding has captured the imagination of researchers. Steganography is one of the techniques that can be used for hiding a secret [HH03]. As an example, in 2001, an international newspaper named USA Today, announced that steganography was used by terrorists [Kel01], in order to plan the September 11, 2001 attack. This have also been confined by other reports [KOL01, Tib02].

In the past, lot of people do not believe that steganography techniques was used by the criminals and terrorists. Now this idea is accepted and understood. For this reason, steganalysis techniques have to be developed to detect suspicious communications.

Steganography and Steganalysis has thrived in the digital era. Many interesting steganalysis techniques have been created dealing with grayscale images, and steganalysis is continuing its evolution. In particular, there is a need to detect hidden message in color images [KBB⁺13]. One of important motivation of this thesis is to support this situation by developing new steganalysis methods to detect hidden message in digital color image.

1.3 Key Contributions

The main key contributions of this thesis are based on feature extraction using oriented filter. On one hand, proposed new methods for contours and corners detection and evaluation. On the other hand, proposed steganalysis methods to detect hidden messages in image which have been carried out within extracting features from color images and machine learning. We can summarize contributions of this thesis in the list below:

- Proposed a novel method for corner detection based on the combination of directional derivative and homogeneity kernels [AMM17c].
- Proposed a new technique to evaluate filtering of edge detection methods involving the minimum score of the considered measures. Also, demonstrate how to label new ground truth data in a semi-automatic way, [AMM17a].
- Proposed new normalized supervised edge detection evaluation method, comparing a ground truth contour image, the candidate contour image and their associated spacial nearness [AMM17d].
- Several referenced-based boundary detection evaluations are detailed, pointing their advantages and disadvantages through concrete examples of edge images [AMM17d].
- Proposed a new supervised edge detection evaluation method. Theoretically, by varying the hysteresis thresholds of the thin edges [AMM17b].
- The development of a color image steganalysis based on color feature correlation and machine learning. Features are extracted from the channel correlation, and co-occurrences [ACMM15].
- Proposed a new technique to building big database for color image Portable Pixel Map (PPM) format of size 512×512 using a demosaicking algorithm [ACMM15].
- The development of new color image steganalysis based on geometric measures obtained by the sine and cosine of gradient angles between all the color channels [ACMM16b].

- The development of a new color steganalysis based on steerable Gaussian filter banks. This approach enriches the features from the Color Rich Model by adding new features obtained by applying steerable Gaussian filters and then computing the co-occurrence of pixel pairs [ACMM16b].
- Intensive experiments reveal that our steganalysis approaches is able to better detect the color adaptive steganographic approaches.

1.4 Organization of this thesis ---

This work has been structured in the following parts and chapters:

I. Overview on Digital Image Analysis concepts

- **Chapter 2:** *Digital image analysis.*

II. Image Feature Detectors and Evaluation

- **Chapter 3:** *Edge detection and evaluation methods.*
- **Chapter 4:** *New normalized supervised edge detection evaluation method.*
- **Chapter 5:** *From contours to ground truth: How to evaluation edge detectors by filtering*
- **Chapter 6:** *Objective supervised edge detection assessment using Hysteresis thresholds.*
- **Chapter 7:** *Orientation asymmetric kernels for corner detection.*

III. Detection of Hidden Data in Digital Images

- **Chapter 8:** *Overview on steganography and steganalysis*
- **Chapter 9:** *Color channel correlation and geometric steganalysis.*
- **Chapter 10:** *Steganalysis based on steerable Gaussian filter.*

IV. Conclusion and Future Work

- **Chapter 11:** *Conclusions and Future work*
- **Chapter 12:** *Résumé en Français*

Part I

OVERVIEW ON DIGITAL IMAGE
ANALYSIS CONCEPTS

2

DIGITAL IMAGE ANALYSIS

This chapter focus on CFA Bayer demosaicing, decomposition of color Space, and types of noise in digital image. Also, focus on digital image analysis that aims at detecting, identifying, measuring, and analysing features in digital images.

Contents

2.1	Introduction	10
2.2	Digital Images	10
2.2.1	Color Images	12
2.2.2	CFA Bayer Demosaicing	12
2.2.3	Decomposition of Color Space	14
2.3	Noise in Digital Image	18
2.3.1	Models of Image Noise	20
2.4	Applications of Digital Image Analysis	21
2.5	Summary	23

2.1 Introduction

Digital image processing can be thought of as a sequence of steps: image acquisition, pre-processing, segmentation, feature extraction, analysis and evaluation. These steps can also be expanded into more intermediate steps depending on the application. More deeply, digital image analysis is an area which has been widely investigated and there is a torrential literature covering different aspects of processing images. The digital image processing deals with developing a digital system that performs operations on an image by using a computer [GW02]. The interest in digital image processing methods is motivated by two major application tasks. The first is the improvement of pictorial information for the human interpretation, while the second is data processing for autonomous machine perception. Normally, digital image means collection of individual pixels, to which discrete brightness or color values are assigned. That's made it the essential core in many applications from our daily life.

One common example of a pre-processing step is noise reduction. Filtering in spatial domain, performed in a controlled manner will decrease the effect of noise on images. A meaningfully designed smoothing filter will even out the noise in an image without blurring important information such as edges. Median filtering, that determines filter output coefficients by the median of the neighboring pixels/voxels, is a non-linear filtering method that preserves edges. Also, image segmentation is the process that separates the content in an image into objects and background, and can be seen as an essential part of many image analysis tasks. Thresholding is a popular image segmentation method and there are a number of thresholding techniques that can be implemented to thin edge some methods use a signal, or use a double threshold and automated thresholding techniques.

2.2 Digital Images

A digital image is composed of a finite number of elements called pixels, each pixel is assigned a tonal value such as black, white, shades of gray or color and a particular location. These pixels are stored in a sequence of bits on a computer. We can define an image as a two-dimensional function $I(x, y)$, where x and y are the spatial (plane) coordinates, and I the amplitude at any pair of coordinates (x, y) is called the intensity of the image at this point. If x, y and the amplitude values of I are finite and discrete quantities, we call this image a digital image [GW02].

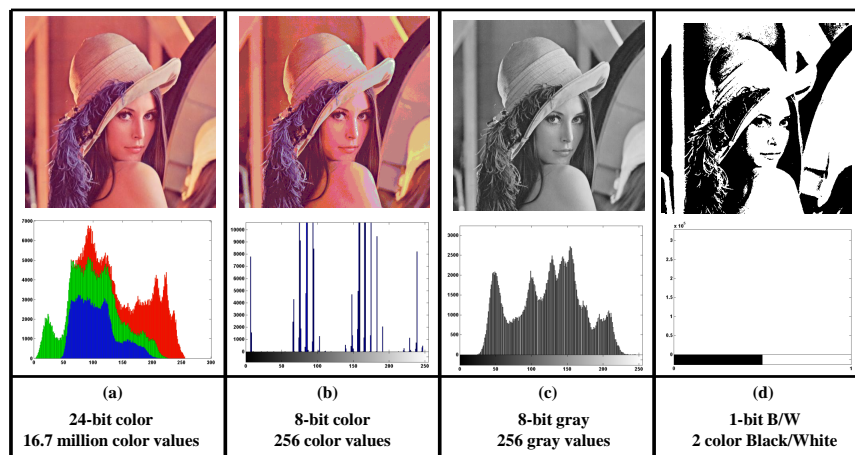


Figure 2.1: Different version of the same image; (a) 24-bit color, (b) 8-bit color, (c) 8-bit grayscale, (d) 1-bit (black-and-white), and the histogram of each image.

Each digital image have size which is the number of horizontal and vertical pixels. To complete the basic information about the digital image we should explain the bit depth which refers to the number of bits used to define a single pixel and determines the number of colors of a pixel. Increasing the bit depth leads to increase the number of tones for grayscale or color, to represent the image. Table 2.1 shows the relationship of bit depth to the number of available colors.

Table 2.1: The relationship between the bit depth with available number of colors.

Type	Bit-depth	Number of Colors/Shades
Standard	1-bit bitonal (binary).	Black or white.
	8-bit grayscale.	256 Shades of grayscale.
	24-bit RGB color (true color).	16.77 Million colors.
Professionals	10-bit grayscale.	1024 Shades.
	30-bit RGB color.	one Billion colors.
Professionals	12-bit grayscale.	1024 Shades.
	36-bit RGB color.	68.77 Billion colors.
Professionals	16-bit grayscale.	1024 Shades.
	48-bit RGB color.	281 Trillion colors.

Digital image may be produced in binary (black and white), grayscale and color. Fig. 2.1 illustrate the differences between a 1-bit (black-and-white), a 8-bit grayscale, a 8-bit color, and a 24-bit color with regard to representing Lena image. A binary image is represented by pixels consisting of 1 bit each, which can represent two tones (black and white), using the values 0 for black and 1 for white or vice versa. The grayscale image is composed of pixels represented by multiple bits of information, ranging from 2 to 8 bits or 16 bits. With 256 different gray levels, each pixel can be stored in a single byte (8 bits) of memory.

Color image is typically represented by a bit depth ranging from 8 to 48. With a 24-bit image, the bits are often divided into three groups: Red (R), Green (G), and Blue (B). A 24-bit color image offers 16.77 million color values, which provide significantly more color depth. This is more than enough to accurately portray true-to-life images. A 24-bit color is thus often referred to as "true color", if the vector components of the digitized color image represent spectral transmission of visible light.

2.2.1 Color Images

The most common way to model color in Computer Graphics is the RGB color model, this corresponds to the way both CRT monitors and LCD screens/projectors reproduce color. Each pixel is represented by three values, the amount of red, green and blue. Thus an RGB color image will use three times as much memory as a grayscale image of the same pixel dimensions.

2.2.2 CFA Bayer Demosaicing

In general, a camera capture the image by using a single Charge-Coupled Device (CCD) or Complementary Metal-Oxide Semiconductor (CMOS) sensor. This device is used to sample the light intensity into sensor. Virtually, every digital sensor works by capturing light in an array of photosites. In the same way, when we need to capture color image, the light intensity is measured in separate color bands, usually red (R), green (G), and blue (B). This require three separate sensors in conjunction with a beam splitter to accurately measure each of the three primary colors at each pixel. The three images in a 3 CCD camera can then be combined pixel by pixel to obtain a color image as shown in Fig. 2.2. This design is nevertheless very expensive and mechanically difficult to implement [MJYR⁺10]. To solve this problem and make economic and non-expensive camera price, the Color Filter Array (CFA) is introduced to capture a color image using only one sensor. This solution was patented in 1976 by Bryce Bayer from the Kodak company, and can be found in almost all digital camera sensors sold today [Bay76, LMY10].

A CFA pattern is a mosaic of color filters placed on top of the CCD sensor to filter out the red, green, and blue components of light spectrum that enter the camera. A digital image acquired by CFA filter, is stored in a single size raw image (a matrix), where each pixel only store one channel (Red, Green, and Blue). Note that the absent channel values are reconstructed latter, during the the CFA interpolation, also named demosaicking.

There are various CFA with different order of the mosaic of color filters. All greatly affect on the resolution, quality of the image, and processing times. One of the most popular

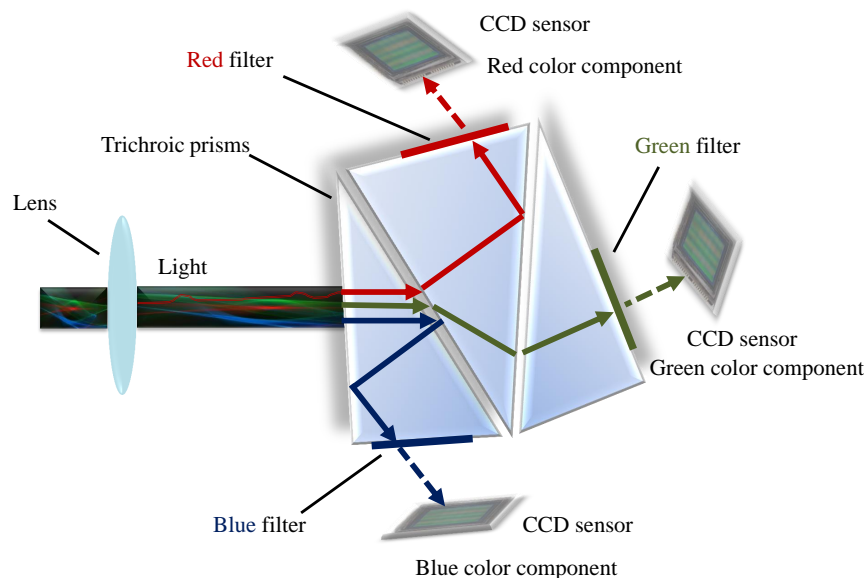


Figure 2.2: Light beam splitting by a trichroic prism in a standard three CCD camera sensor.

and successful CFA in the industry is the Bayer pattern which is a 2×2 repeating pattern using green, red and blue [Bay76]. Each 2×2 pattern uses half green, a quarter red and a quarter blue. When the output of the filter is read, it reads either RGGB, BGGR, GBRG and GRBG. Fig. 2.3 provides a schematic illustration of CFA Bayer design. The ratio of two green pixels to one red and one blue was chosen because the human eye is much more sensitive to green light [ASH05].

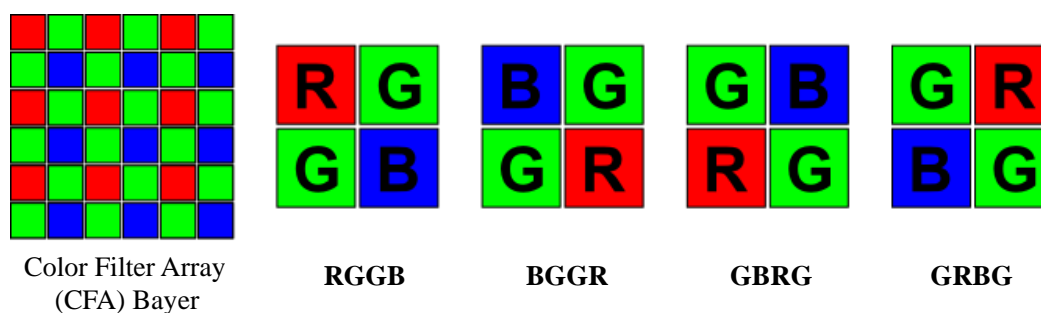


Figure 2.3: The CFA Bayer pattern with red-green-red-green phase in the first row, and four corroborative CFA patterns.

To resume, an image is captured into a sensor device with a Bayer filter array. It gives the raw image, this raw image is then interpolated (demosaicking) in order to give a RGB image. Then additional internal processing are applied in order to give the output image. Different CFA interpolation algorithms are proposed in the literature [All04, SM05]. Each digital camera applies one of these algorithms leading to different image quality. In general, CFA interpolation algorithms are categorized into two major branches as follows:

- Adaptive interpolation. With these type of demosaicking algorithm, the missing values are sectioned using feature of surrounding pixels. These algorithms requires lot of calculations. For example, there are : pattern matching based interpolation algorithms, interpolation using a threshold-based variable number of gradients, and edge sensing interpolation.
- Non-Adaptive interpolation. With these type of demosaicking algorithm, all missing values are interpolated by a fixed pattern. These algorithms requires to low calculation cost. For example, there are nearest neighbour replication, bilinear interpolation, median interpolation, and gradually color change interpolation.

2.2.3 Decomposition of Color Space

The color results from an interaction between light, object, and the viewer. All these three elements must be present for color as we know it to exist. A color space is a useful method for users to understand the color capabilities of a particular digital device or file. Each color in the system is represented by a single dot.

A color model is a method to describe a color, numerous color models have been proposed in the literature for various applications and purposes. We can classified the color models into two classes: the hardware oriented, and the user oriented [GW02].

- The hardware oriented models, includes the RGB color space model, which is used in color cameras and monitors, the CMYK(Cyan, Magenta, Yellow, Key) color model used in color printers, the YUV, YIQ, YCbCr (Luminance - Chrominance) color models which are used in the television transmission color spaces.
- The user oriented models, includes the Hue Saturation Intensity (HSI) color model, the CIEL^{*}a^{*}b^{*} which is based on the human perceptions of color and used for graphics designing and editing [GW02].

More details are briefly given below:

- RGB Color Model. The RGB (R=Red, G=Green, B=Blue) is a color space where each pixel of channel 24 bits an image is assigned a range of 0 to 255 intensity value. The RGB color solid has the shape of unit cube with the grayscale dimension going from 0,0,0 to 1,1,1. The colors red, green, blue, yellow, cyan, and purple are situated at corners of the cube as shown in Fig. 2.4.

The color components are laid out in separate arrays of identical dimensions. In this case, the color image

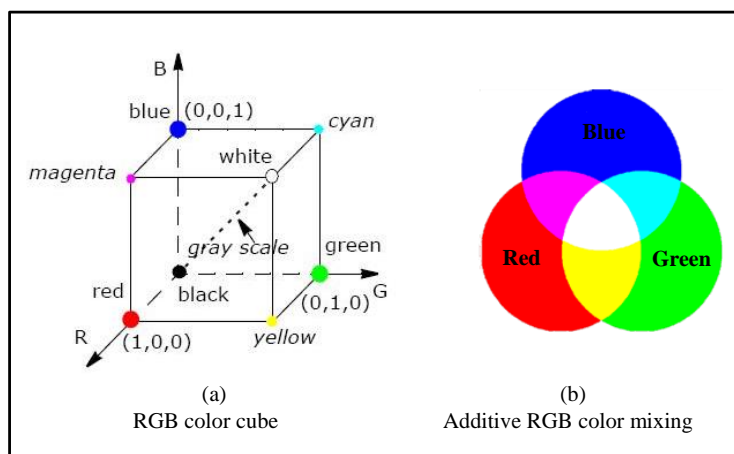


Figure 2.4: Representation of the RGB color space as a three-dimensional unit cube.

- **CMY(K) Color Model.** The (CMY) Cyan, Magenta, and Yellow color model is similar with the RGB color model and uses cyan, magenta, and yellow instead of red, green, and blue, respectively. When all the three color CMY are added together we introduce black color noted K and called key - plate. Fig 2.5 shows the CMY color model. This model is used for printer device. The transformation between

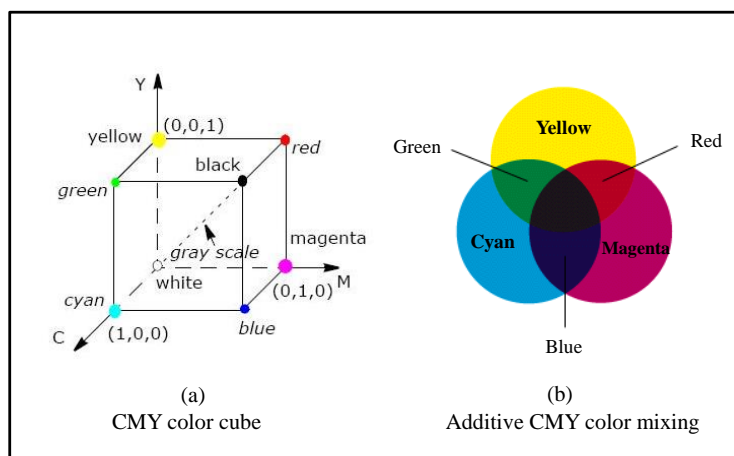


Figure 2.5: Representation of the CMY color model as a three-dimensional unit cube.

RGB color model and CYM color model is presented below:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- **YUV, YIQ, YCbCr (Luminance - Chrominance).** The Luminance - Chrominance systems are based on one luminance value and two chrominance values. All these models are used in video systems. One of the main advantages of these color systems

are that grayscale information is separated from color data, so the same signal can be used for both color and black and white sets.

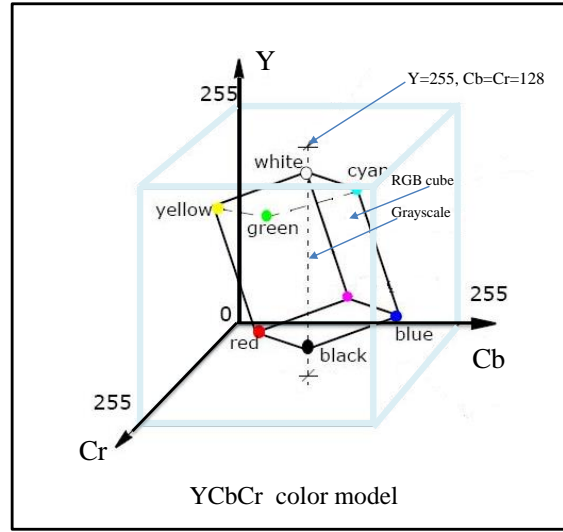


Figure 2.6: Representation of the YCbCr color model.

The (Luminance Component, Chroma Blue difference, Chroma Red difference) YCbCr color space model, came as a result of the evolution of a world-wide digital component video standard. Y is the luminance component, which is defined to have a nominal 8-bit range of 16–235 and (Cb and Cr) are the blue-difference and red-difference chroma components as shown in Fig. 2.6. The conversion equations from RGB to YCbCr color space model is below:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- **HSI Color Model.** The Hue Saturation Intensity (HSI) color model reflects the human vision. Hue defines the pure color tone out of the color spectrum, saturation defines the mixture of the color tone with gray and finally luminance defines the lightness of the resulting color. Fig. 2.7 illustrates how the HIS color space represents colors. The HSI color model is very popular for graphics designing and editing, because it gives the user a good impression about the resulting color for a certain color value. The following formula can be used to convert RGB values to HSI values:

- First normalizing RGB values:

$$r = \frac{R}{R + G + B}, \quad g = \frac{G}{R + G + B}, \quad b = \frac{B}{R + G + B}. \quad (2.1)$$

- The H, S, and I components are obtained by the following:

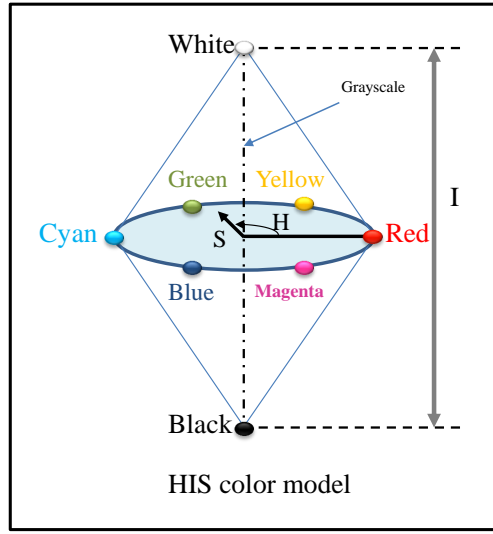


Figure 2.7: Representation of the HSI color model.

$$h = \begin{cases} \cos^{-1} \left\{ \frac{0.5 * [(r-g) + (r-b)]}{[(r-g)^2 + (r-b) * (g-b)]^{\frac{1}{2}}} \right\} & \text{if } b \leq g, \\ 2\pi - \cos^{-1} \left\{ \frac{0.5 * [(r-g) + (r-b)]}{[(r-g)^2 + (r-b) * (g-b)]^{\frac{1}{2}}} \right\} & \text{if } b > g. \end{cases} \quad (2.2)$$

$$s = 1 - 3 * \min(r, g, b). \quad (2.3)$$

$$i = \frac{(R + G + B)}{(3 * 255)}. \quad (2.4)$$

- Finally, The values h , s , and i are converted in the ranges of $[0, \dots, 360]$, $[0, \dots, 100]$, and $[0, \dots, 255]$, respectively to obtains H, S, and I by using:

$$H = h \times \frac{180}{\pi}; \quad S = s \times 100; \quad I = i \times 255.$$

- CIE $L^*a^*b^*$ Color Model. This model is compute from XYZ and hence is device independent. For the purpose of standardization the International Commission on Illumination (CIE) in 1931 defined a system of virtual primary three colors to represent all visible colors by three color values (X, Y, Z) positive [GGVdWG12]. The CIE $L^*a^*b^*$ models the human vision and is independent of the equipment. L^* denotes luminosity (values are always ranging from 0-100), a^* denotes red/green axis, and b^* yellow/blue axis. Both a^* and b^* describes the hue and saturation, they have no specific numerical limits. Fig. 2.8 illustrated this color model, positive a^* is red, negative a^* is green, positive b^* is yellow, negative b^* is blue.

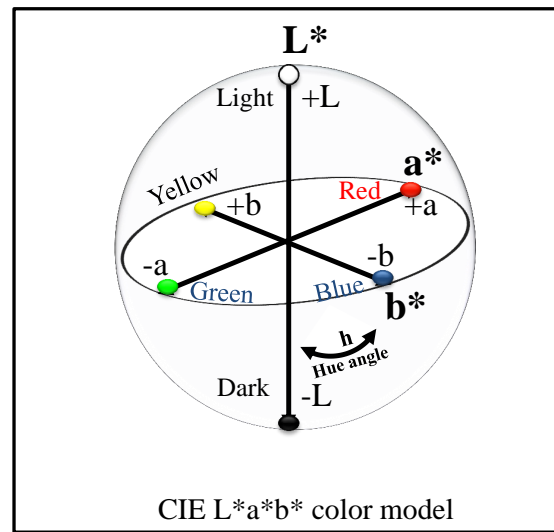


Figure 2.8: Representation of the CIE $L^*a^*b^*$ color model.

2.3 Noise in Digital Image

During image acquisition many noise sources are superimposed on the acquired image. It is possible to classify the sources of noise into two main families: the physical noise, like the photons of natural light, and the hardware noise, due to the components and the mechanical joint of the camera. This section focuses on the noise that is created through the different stages of the image formation process. Some of these noises are due to raw sensor [Hyt06], and others from additional processing such as color interpolation, denoising, color correction, and filtering. These combined noise sources are usually modelled as an independent signal of the image content. Through the image formation process, many types of noise are created and added to the image. Fig. 2.9 illustrates a typical camera with several different components. The different sources of noise are:

- **Readout noise.** This type of noise is present and inherent in every CCDs image, sometimes abbreviated (readnoise). When a CCD image is taken, noise will appear as well as the main CCD image, it is a result of the incomplete operation of physical electronic devices. More clearly, readout noise is the noise of the on-chip amplifier which converts the charge (i.e. the electrons) into a change in analogue voltage. This means the noise will produce fluctuations in the number of analog to digital units recorded [Jan01]. Amplifier noise is one of this type of noise, and resembles Gaussian noise.
- **Shot noise.** This type of noise is fundamentally connected to the way photons spatially come on a detector and associated with the particle nature of light. Light

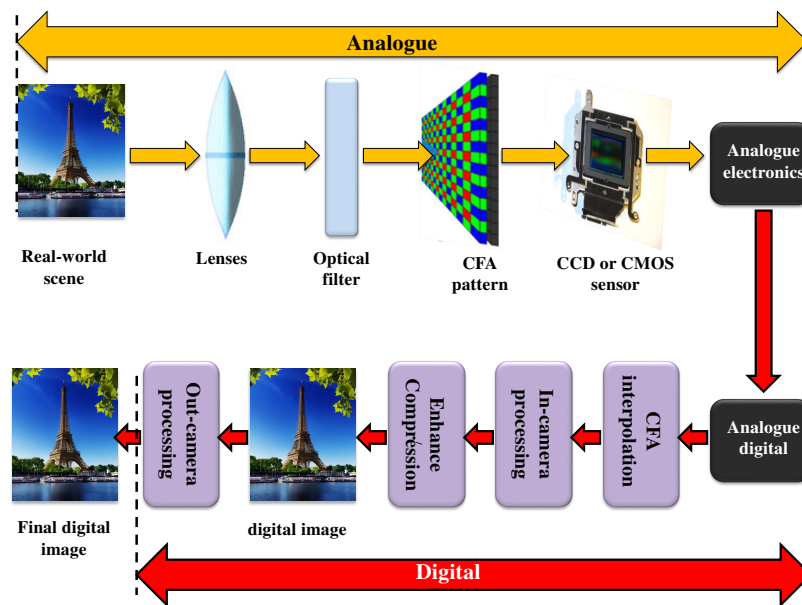


Figure 2.9: Image acquisition using a typical digital camera.

is made up of discrete bundles of energy called photons. The type of noise is caused by the arrival process of light photons on the sensor [Hol98].

When more light can be captured this, the shot noise will be higher than the readout noise in the image.

- **Quantization noise.** All digital devices produce quantization noise. This noise is introduced by quantization in the analogue to digital conversion. This quantization error is unavoidable and due to the limitations of the converting from an analog signal to a digital signal. It is independent of the signal and can be modelled as additive noise [LBMR10]. There are two types of quantization errors: the approximation error, and the clipping error. The approximation error is caused by the round-off effect of the analog signal due to the lack of quantization resolution. The clipping error is due to inadequate quantizer range limit [LBMR10].
- **Dark current.** This type is formed from the generated temperature inside the camera device or due to the external temperature. This noise is also called thermal noise.
- **Pattern noise.** There are two components of the pattern noise: the Fixed Pattern Noise (FPN) and the Photo-Response Non-Uniformity noise (PRNU) [LFG06]. The (FPN) noise pattern is refers to, pixel to pixel variation when the sensor does not take enough of light [Moo91, EGFML98]. Also this noise some times occurred by the temperature. The (PRNU) noise pattern is refers to, Pixel Non-Uniformity (PNU), that is caused by the physical properties of the sensor itself [Hol98]. This type of noise has been successfully used for source camera identification as described in [LFG05].

2.3.1 Models of Image Noise

Noise in images, is considered to be a stationary phenomenon and influenced by several factors, some of these factors can be controlled, and others are very difficult to control such as the hardware noise. It is very difficult to determine a universal model for the noise. Wherefore, there are many models for noise. In general terms, the most popular models are: Gaussian noise or white Gaussian, impulse noise (salt and pepper noise), Rayleigh noise, uniform noise, and gamma noise.

- Gaussian model. This model is important for modeling a natural processes, which present noise, such as the noise that result by the conversion of the optical signal into an electrical one a shot noise, and the noise during acquisition. The Gaussian noise has a normal (Gaussian) probability density function (PDF):

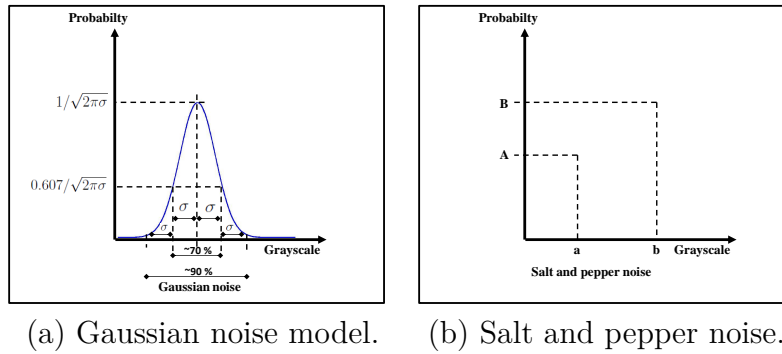


Figure 2.10: Probability density function for: a) Gaussian noise model, b) Salt and pepper noise model.

$$PDF_{Gaussian} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(g - \mu)^2}{2\sigma^2}}, \quad (2.5)$$

Where, g = gray scale; μ =mean; and σ =standard deviation. As shown in Fig. 2.10(a) we can observe an approximately 70% of the values are contained between $\mu \pm \sigma$, and 90% of the values are contained between $\mu \pm 2\sigma$.

- Salt and pepper model. In this model, only two possible values are conceivable, a and b. The probability of producing one of them is very small, and otherwise, the noise would vastly dominate the image. In general, the salt & pepper noise is result by malfunctioning camera's sensor cells, by synchronization errors, or by memory cell failure in the image digitizing or transmission. For an 8 bit/pixel image, the typical intensity value for pepper noise is close to 0 and for salt noise is close to 255 in Fig. 2.10 (b), the PDF is given the Eq. 2.6 below:

$$PDF_{Salt\&pepper} = \begin{cases} A & \text{for } g=a \text{ ("pepper")} \\ B & \text{for } g=b \text{ ("salt")}. \end{cases} \quad (2.6)$$

- Gamma model. The gamma model is generally seen in the laser based images. It obeys the Gamma distribution, which is given in the Eq. 2.7 below:

$$PDF_{Gamma} = \begin{cases} \frac{a^b g^{b-1} e^{-ag}}{(b-1)!} & \text{for } g \geq 0, \\ 0 & \text{for } g < 0, \end{cases} \quad (2.7)$$

Where, $\mu = \frac{b}{a}$ and variance $\sigma^2 = \frac{b}{a^2}$ are given.

- Rayleigh model. This model present in radar images. The probability density function is given in the Eq. 2.8 below:

$$PDF_{Rayleigh} = \begin{cases} \frac{2}{b} (g-a) \cdot e^{-\frac{(g-a)^2}{b}} & \text{for } g \geq 0, \\ 0 & \text{for } g < 0, \end{cases} \quad (2.8)$$

where, $\mu = a + \sqrt{\frac{\pi b}{4}}$ and variance $\sigma^2 = \frac{b(4-\pi)}{4}$ are given.

2.4 Applications of Digital Image Analysis

Image processing has become such a critical component in recent science and technology that many tasks would not be attempted without it. It is a truly interdisciplinary topic that draws from effected developments involving many disciplines and is used in medical imaging, microscopy, astronomy, computer vision, geology and many other fields [Dou09].

- Filtering. is a technique for modifying or enhancing an image by applying various effects on it. For example, you can filter an image to emphasize certain features or remove other features. Image filtering is useful for many applications, including smoothing, sharpening, removing noise, and edge detection. Linear filtering of an image is accomplished through an operation called convolution. Convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels. The matrix of weights is called the convolution kernel, also known as the filter which is a small array applied to each pixel and its neighbors within an image.

- Machine vision. In the computer vision we want to make computers emulate human vision in terms of high-level understanding, including learning and being able to make inferences and take actions based on visual inputs (digital images or videos). Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images. Sometimes uses the image processing tools and algorithms to accomplish its tasks. One other word, machine vision is the process of applying a range of technologies and methods to provide imaging based automatic inspection, process control and robot guidance in industrial applications. Machine vision tends to focus on applications, mainly in manufacturing, e.g., vision based autonomous robots and systems for vision based inspection or measurement. This implies that image sensor technologies and control theory often are integrated with the processing of image data to control a robot and that real-time processing is emphasized by means of efficient implementations in hardware and software. It also implies that the external conditions such as lighting can be and are often more controlled in machine vision than they are in general computer vision, which can enable the use of different algorithms.
- Medical imaging. The rapid and continuing progress in computerized medical image reconstruction, and the associated developments in analysis methods and computer-aided diagnosis, have induced medical imaging into one of the most important sub-fields in scientific imaging. Medical image processing refers to the applied technologies of image processing in medical disciplines. It represents the tools that are used to have an internal vision of any anatomical part of the human body which helps doctors to improve and speed up the diagnosis process. Different modes of medical imaging exist regarding the different types of energy which used in the image acquisition. These modes are called the modalities such that Radiography, magnetic resonance imaging, ultrasound imaging, nuclear medicine, computed tomography [Leo05].
- Remote sensing. is the acquisition of physical data of an object without touch or contact. Earth observation satellites have been used for many decades in a wide field of applications. Remote sensors can be either passive or active. Passive sensors respond to external stimuli. Active sensors use internal stimuli to collect data about Earth. Imaging spectrometers and thermal multi-spectral systems acquire detailed spectroscopic information of physical properties of the earth's surface. High-resolution imaging allows a very detailed, three-dimensional reconstruction of the earth surface. One particular application of digital image processing in the field of remote sensing is to detect infrastructure damages caused by an earthquake [FDvdM02]. Since the area effected by the earthquake is sometimes so wide, that it is not possible to examine it with human eye in order to estimate damages. Even if it is, then it is very hectic and time consuming procedure. So a solution to this is found in digital image processing. An image of the effected area is captured from the above ground and then it is analysed to detect the various types of damage done by the earthquake.

- **Robotics humanoid.** robots are machines that looks like a human being and does activities like walking, lifting, speech, cognition, and basically anything a human can do. Most of the robots today work by following the line and thus are called line follower robots. This help a robot to move on its path and perform some tasks. This has been achieved through image processing. Computer vision systems rely on image sensors which detect electromagnetic radiation which is typically in the form of either visible light or infra-red light. The process by which light propagates and reflects off surfaces is explained using optics. Sophisticated image sensors even require mechanics to provide a complete understanding of the image formation process. These robots can also be equipped with multiple vision sensors to be better able to compute the sense of depth in the environment. Like human eyes, robots eyes must also be able to focus on a particular area of interest, and also adjust to variations in light intensities. The developments of humanoid robots proceed from building individual robots to establishing societies of robots working alongside with humans.
- **Security.** Image processing plays an important role in the development of technologies for dealing with security issues especially with the increase of cyber crime. Surveillance cameras are widely diffused as means of crime reduction, and image analysis tools are used in forensic fields. Intelligent Gate and electronic lock are another examples of using image processing in security issues. Verifying a digital image integrity and authenticity is an important task in security especially considering that the images can be digitally modified by low-cost hardware and software tools that are widely available.

2.5 Summary

To summarize this chapter, here basic concepts and principles of digital image processing have been presented. More deeply, we started from the introduction of digital image analysis followed by CFA Bayer Demosaicing. A CFA pattern is a mosaic of color filters placed on top of the CCD sensor to filter out the red, green, and blue components of light spectrum that enter the camera. Afterwards, the fundamental decomposition of color space were presented. In addition, some commonly models of image noise have been summarized. Finally, commonly applications of digital image analysis techniques have been also presented.

Part II

IMAGE FEATURE DETECTORS
AND EVALUATION

3

EDGE EVALUATION METHODS AND CORNER DETECTION

This chapter begins with background of edge and corner detection, mainly including methods for extracting edge features and the performance evaluation for edge and corner detection.

Contents

3.1	Introduction	28
3.2	Low-level Feature Extraction	29
3.2.1	Global and Local Features	29
3.3	Edge Definition	30
3.3.1	Edge Labelling process	31
3.3.2	Non-maximum suppression technique	31
3.3.3	First and Second Order Derivative Edge Detection	32
3.3.4	Two-Dimensional Gaussian Derivative	34
3.4	Traditional edge operators	35
3.4.1	Sobel edge detector	35
3.4.2	Canny edge detector	36
3.5	Filter banks	37
3.5.1	Steerable Gaussian Filter	38
3.5.2	Anisotropic Gaussian kernels	40
3.5.3	Half Gaussian kernel	40
3.6	Summary	46

3.1 Introduction

During the last years, image features detectors have become popular tools and one of the most active topics for image representation in computer vision community, they are widely used in many applications [Sze10]. Furthermore, extraction feature is an essential pre-processing step in pattern recognition [AT13]. Image feature, is referred as the identification of interested image primitives such as points, lines, curves, and regions. Edges and contours are much related to object boundaries. They are required and applied in image interpretation, such object recognition, image segmentation, visual tracking, action analysis. Whereas, corner detection is much related to gradient orientations. traditional corner detection based the calculation of second order derivatives. Edge refers to pixels where the image intensities change abruptly. Image pixels are discontinuous at different sides of edges. Contour/boundary can be viewed as the generalized definition of edge which indicates the intersection of different regions. Corner refers, to the point at which two different edge directions occur in the local neighborhood. Generally, the difficulties in feature detection are caused by the changes viewpoint [HS88], illumination, scale [Lin98, Low04], image quality [WBSS04], etc.

In computer vision and image processing, edges, corners and junctions detection are provides very important visual information about the object or image. Extracting edge features is the main and necessary process in edge and corner detection. Edge detection allows us to observe those features of an image where there is a more or less abrupt change in gray level or texture indicating the end of one region in the image and the beginning of another. However, edge detection techniques for color image presents some challenge, the simple technique is to look for discontinuities in luminance component [ACG10]. Edges typically occur on the boundary between two different regions in an image. The most common types of image intensity variations are steps, lines, and junctions [RFM06]. All these types results from different original seen, for example when one object hide another or when there is a shadow on surface. Edge detector methods are briefly divided into two classes differentiation based class and learning based class. Whilst, corner detector methods are divided into three classes gradient based, contour based and template based methods. Many edge detection techniques have been developed for extracting edges from digital images. Differentiation based filters are convolved to identify edge points. The edge detector techniques can be classified into two main categories, first one is gradient methods, finds edge by gradient maxima and minima Intensity. and the second is Laplacian methods, finds zero crossing in the second derivative of the image. More deeply, first-order derivative based gradient kernels appear in pairs such as Sobel, Prewitt, and Roberts. The gradients are computed at different orientation, by convolution these kernels with an image, and finally the local maxima of the gradient magnitudes are recorded as edges. Second-order derivative filters such as Laplacian of Gaussian (LoG) will be find the zero-crossing as the edge positions [Har84]. In a grey level image the edge is a local feature that, with in a

neighborhood separates regions in each of which the gray level is more or less uniform with in different values on the two sides of the edge. For a noisy image it is difficult to detect edges as both edge and noise contains high frequency contents which results in blurred and distorted result. Evaluation techniques for edge detection methods are studied in this thesis. This evaluation is difficult because of difficulties in determining the best parameters associated with each edge techniques. We noticed in the recent years there has been considerable interest in techniques for evaluation of computer vision applications and methods. All systems, especially automated information processing structures, must be evaluated before being developed, principally for industrial applications or medical data.

The majority require ground truth, which makes the evaluation methods impractical. Some alternatives exist; for instance, assessing the quality of image segmentation without reference to a ground truth image, but by measuring for each region its internal homogeneity and its contrast along its boundaries. The quality of thresholded edge maps using edge continuity and thinness. While appealing, the problem is that these criteria do not always reflect good results, most cases ground truth is essential for performing a quantitative analysis of an algorithm results.

3.2 Low-level Feature Extraction

Low-level features are defined as basic features that can be extracted automatically from an image without any shape information. As such, thresholding is actually a form of low-level feature extraction performed as a point operation. Difference methods find features by calculating the derivative values of an image and then analysing them. In general, several computer vision systems are used two types of features global image features, which is describe an image as a whole or used the local image features representing local image patches [NA12]. Global texture features and local features provide different information about the image because the support over which texture is computed varies.

3.2.1 Global and Local Features

These features concerning the search on the whole image like the color histograms, contour representations, texture, shape form, ... etc. The system based solely on global features can not give the desired results. Either an image composed of several objects with very different characteristics, colors and textures, the global feature vector extracted from the entire image loses local information (objects) and produces only a rough average of the content of this image [LMB⁺05]. Local features concerning the precise search on a part of

the image using regional segmentation, detection of points of interest, etc. Analysis based solely on local characteristics risks losing the overall sense of the image by submerging it in a stream of unnecessary little details.

3.3 Edge Definition

Typically, edges occur on the boundary between two different regions in an image. In other words, an edge is the boundary between an object and the background. In one dimension, ideal edge may be viewed as a step change in intensity. In real signals, the step change is likely to be mixed with noise caused by sensor as explained in chapter two, surface, or light variations. Detection ideal step edges with noise is difficult and many times is deficiency since both the noise and the edges contain high-frequency content, while ideal edges without noise detected very easy and simple. Therefore, in the literature on techniques for edge detection there are many methods uses ground truth image or synthetic image to measure the error detection rate to evaluate the efficiency of the method [SMB00, AMFM09, LMDBB16].

There exist different types of edges in images such as: peak, roof, ramp, step, small blur and step with blur edges, Fig. 3.1 illustrated the signal of these types of edges after convolution the image with $[-1 \ 0 \ 1]$ mask. In real images, step edges are localized at the inflection points of the image. Accordingly, the step edges are localized as positive maxima or negative minima of the first-order derivative. While, the zero-crossings of the second-order derivative are used to find an image edges. In the gray image the first-order derivative is defined by the gradient filter while the second-order derivative is defined by the second derivative along the gradient direction or by Laplacian filter. There are many

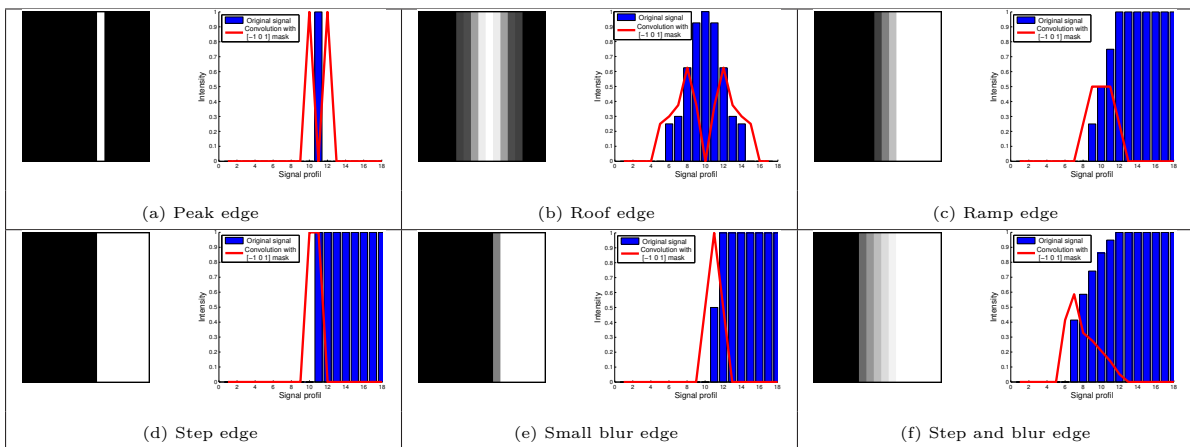


Figure 3.1: The different types of edges and result of a convolution with a $[-1 \ 0 \ 1]$ mask (absolute value).

methods to make edge detection, but the most common method for edge detection is to calculate the differentiation of an image.

3.3.1 Edge Labelling process

Edge labelling process include localizing the edges and increasing signal-to-noise ratio by suppressing the false edges. The localization step is based on the applied differentiation operator. The edge detector methods based on used the gradient, the edges are localized by applying a threshold to the gradient magnitude. The edges resulting in these type of methods are dispersed. Therefore, needs additional process to lead up to final results with uniform edges [ZT⁺98b]. Non-maximum suppression (NMS) procedure is used to improve the performance of threshold based edge detection. The basic idea for Non-maximum suppression is to extract local maxima of the gradient magnitude along the direction of the gradient vector, then a specific pixel is a local maximum, if the gradient magnitude at this pixel is greater than the gradient of two neighbouring points situated at the distance on either side of the given pixel along the gradient direction. Whilst, edge labelling in zero-crossing methods compare the result of a second-order derivative at a given pixel with the neighbour pixels to the left and down, if these three pixels do not have the same signs, than the horizontal and vertical direction improves the localization [ZT⁺98b].

3.3.2 Non-maximum suppression technique

Non-maximum suppression (NMS), has been widely used in several computer vision applications, it is considered as an intermediate part of many approaches in detection, might it be edge, corner or object detection [RT71, DT05, GDDM14]. Furthermore, this post-preprocessing is named an edge thinning technique, applied to thin the edges after compute gradient. More deeply, the image is scanned along the image gradient direction, and if pixels are not part of the local maxima they are set to zero. This has the effect of suppressing all image information that is not part of local maxima. Non-maximum suppression consists of the following:

- a) Let the location of a point (x, y) , where x and y are integers and $f(x, y)$ the intensity of pixel at the location (x, y) .
- b) Compute the gradient of image intensity with he magnitude in location (x, y) .
- c) Assessment the magnitude of the gradient over the direction of the gradient in some neighborhood around (x, y) .

- d) If pixel in the location (x, y) is not a local maximum of the magnitude of the gradient along the direction of the gradient then we can know that it is not an edge point. In other word, if a pixel is not part of the local maximum we set this pixel to zero.

In General, for step (d) the neighborhood is taken to be 3×3 and the values of the magnitude are linearly interpolated between the closest points in the neighborhood, Fig. 3.2 shown that the value at pixel K is interpolated between the values at the pixels P7 and P8 and the values at the pixel M between those at the pixels P3 and P4. In other word,

- If the rounded gradient angle is 0° then the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the east and west directions.
- If the rounded gradient angle is 90° the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the north and south directions.
- If the rounded gradient angle is 135° the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the north west and south east directions.
- If the rounded gradient angle is 45° the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the north east and south west directions.

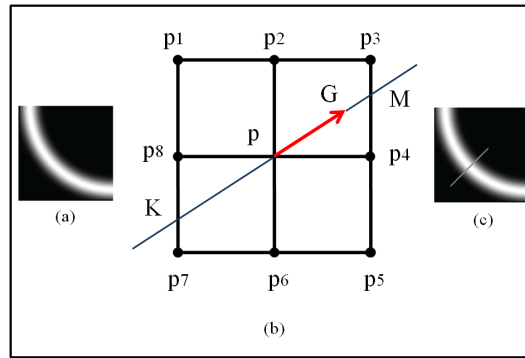


Figure 3.2: Examination if the pixel P in (b) is a local maxima of the magnitude of the gradient in the direction of the gradient is done by interpolating the gradient magnitude at pixels M and K in (b).

3.3.3 First and Second Order Derivative Edge Detection

The first derivative of the function $f(x)$, is the slope of the tangent line to the function at the point x . The first derivative tells us whether a function is increasing or decreasing, and by how much it is increasing or decreasing. This information is reflected in the graph of

a function by the slope of the tangent line to a point on the graph, which is sometimes describe as the slope of the function. Positive slope tells us that, as x increases, $f(x)$ also increases. Negative slope tells us that, as x increases, $f(x)$ decrease. Zero slope does not tell us anything in particular: the function may be increasing, decreasing, or at a local maximum or a local minimum at that point [Pra13]. The first-order derivative in image processing is the gradient defined, and computation of the gradient of an image is based on obtaining the partial derivatives of $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ at every pixel location.

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}. \quad (3.1)$$

An important quantity in edge detection is the magnitude of this vector, denoted $|\nabla f|$. As with any vector, we can compute its magnitude and orientation $\Theta(\nabla f)$ where:

$$|\nabla f| = \sqrt{G_x^2 + G_y^2}. \quad (3.2)$$

The gradient magnitude gives the amount of the difference between pixels in the neighborhood (the strength of the edge), while the gradient orientation gives the direction of the greatest change, which presumably is the direction across the edge normal. Another important quantity is angle of ∇f the direction of the gradient vector:

$$\text{angle of } \nabla f \text{ with } x\text{-axis} = \tan^{-1} \left(\frac{G_y}{G_x} \right). \quad (3.3)$$

The second order derivative of a function is the derivative of the derivative of that function. Graphically, the first order derivative represents the slope of the function at a point and second order derivative describes how the slope changes over the independent variable in the graph. If the second derivative is positive, then the first derivative is increasing, so that the slope of the tangent line to the function is increasing as x increases. A basic definition of the second-order derivative as the difference $\frac{\partial^2 f}{\partial^2 x}$ [Pra13]. Second derivative have a stronger response to fine detail, such as thin lines and isolated points, its response is stronger to a line than to a step, and to a point than to a line and produce a double edge response at ramp and step transition in intensity. One of popular edge detection method of second-order derivative used the zero-crossings of the Laplacian Δf which is correspond to the contours given by Eq. 3.4. The Laplacian filter highlights edges in all directions, it supports estimation of the edge strength regardless of edge orientation.

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3.4)$$

Most edge detecting operators can be thought of as gradient calculation. Since derivatives are linear and shift invariant, gradient calculation is most often done by convolution. Numerous kernels have been proposed to find the edges and contours, some of these kernels are basis of our work and will be explained in this chapter more deeply.

3.3.4 Two-Dimensional Gaussian Derivative

Gaussian filters are designed to give no overshoot to a step function input while minimizing the rise and reduce time. This behavior is closely connected to the fact that the Gaussian filter has the minimum possible group delay. It is considered the ideal time domain filter. The two-dimensional Gaussian filter has an impulse response. Fig. 3.3 (a) illustrates the Gaussian distribution and given by Eq.3.5.

$$f(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.5)$$

Where x is the distance from the origin of horizontal axis, y is the distance from the origin of vertical axis, and σ is the standard deviation of the Gaussian distribution representing the width of the Gaussian. In two dimensional Gaussian first order derivative, it is the product of two such Gaussian is one per direction Gaussian distribution are shown in the Fig. 3.3(b) and given by Eq. 3.6 and Eq. 3.7 .

$$\frac{\partial f(x, y, \sigma)}{\partial x} = -\frac{x}{2\pi\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.6)$$

$$\frac{\partial f(x, y, \sigma)}{\partial y} = -\frac{y}{2\pi\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.7)$$

In two dimensional Gaussian second order derivative, it is the product of two such Gaussians is one per direction given by Eq. 3.8 and Eq.3.9 and Gaussian distribution are shown in the Fig. 3.3(c).

$$\frac{\partial^2 f(x, y, \sigma)}{\partial^2 x} = \left(-1 + \frac{x^2}{\sigma^2}\right) \frac{e^{-\frac{(x^2+y^2)}{2\sigma^2}}}{2\pi\sigma^4} \quad (3.8)$$

$$\frac{\partial^2 f(x, y, \sigma)}{\partial^2 y} = \left(-1 + \frac{y^2}{\sigma^2}\right) \frac{e^{-\frac{(x^2+y^2)}{2\sigma^2}}}{2\pi\sigma^4} \quad (3.9)$$

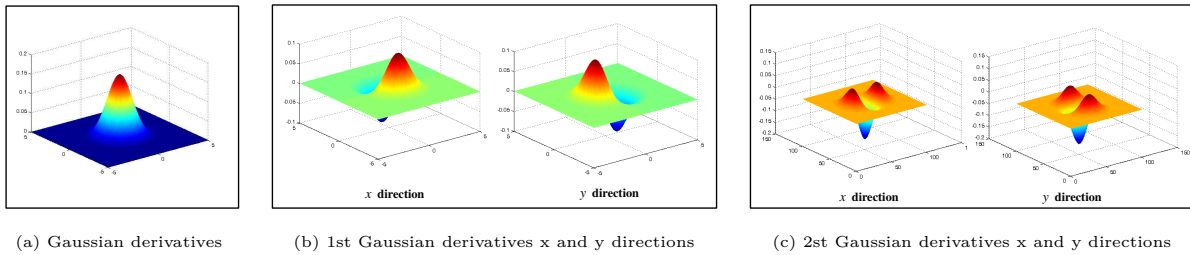


Figure 3.3: 2D-dimensional Gaussian first and second order derivative operators give receptive of x and y directions.

There are many properties and advantages of Gaussian filters like removes the noise from an image, blurs the edges, also with using larger σ more details maybe are removed and the good advantage is separable for example Gaussian can be separated into x and y components, thus increasing the speed of convolution the image with Gaussian filter [Low04, BETVG08].

3.4 Traditional edge operators

In the literature, there are numerous edge detection operators available for image segmentation and object boundary to extract the edges features of digital color or gray-scale images. The traditional popular works deals with this group such as: the methods developed by Sobel [Sob70], Prewitt [Pre70], Canny [Can86b] and Shen-Castan [SC92]. These methods compute gradient and look for local maxima to find step edges. In general, all these methods are simple in computation and capable to detect the edges and their orientation. In this section, a study is made to review some of the traditional edge detection techniques that are based on discontinuity intensity levels. In general, each filter is designed to be sensitive to certain types of edges.

3.4.1 Sobel edge detector

The Sobel operator [Sob70] is one of classic first order edge detector operator. In general, it is used as a standard gradient computation method to retrieve the image gradient and edges. More deeply, this operator only considers the two orientations which are 0° and 90° degrees convolution kernels uses the masks as shown in Fig. 3.4. The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the first derivatives for horizontal changes (G_x), and another for vertical (G_y), means the gradient component in each orientation.

-1	0	1
-2	0	2
-1	0	1

(a) G_x for the derivative in x

-1	-2	-1
0	0	0
1	2	1

(b) G_y for the derivative in y

Figure 3.4: The Sobel operator as a pair of 3×3 convolution kernels in X and Y directions.

Then combined these component together to find the absolute magnitude of the gradient at each point and the orientation of this gradient.

$$|G| = \sqrt{G_x^2 + G_y^2}, \quad (3.10)$$

$$|G| = |G_x| + |G_y|, \quad (3.11)$$

While the angle of orientation of the edge can be computed by the Eq. 3.12

$$\Theta = \arctan(G_y/G_x). \quad (3.12)$$

This operator detects edges are where the gradient magnitude is high, this makes the Sobel edge detector more sensitive to diagonal edge than horizontal and vertical edges. Fig. 3.5 illustrates the process steps for Sobel edge detection.

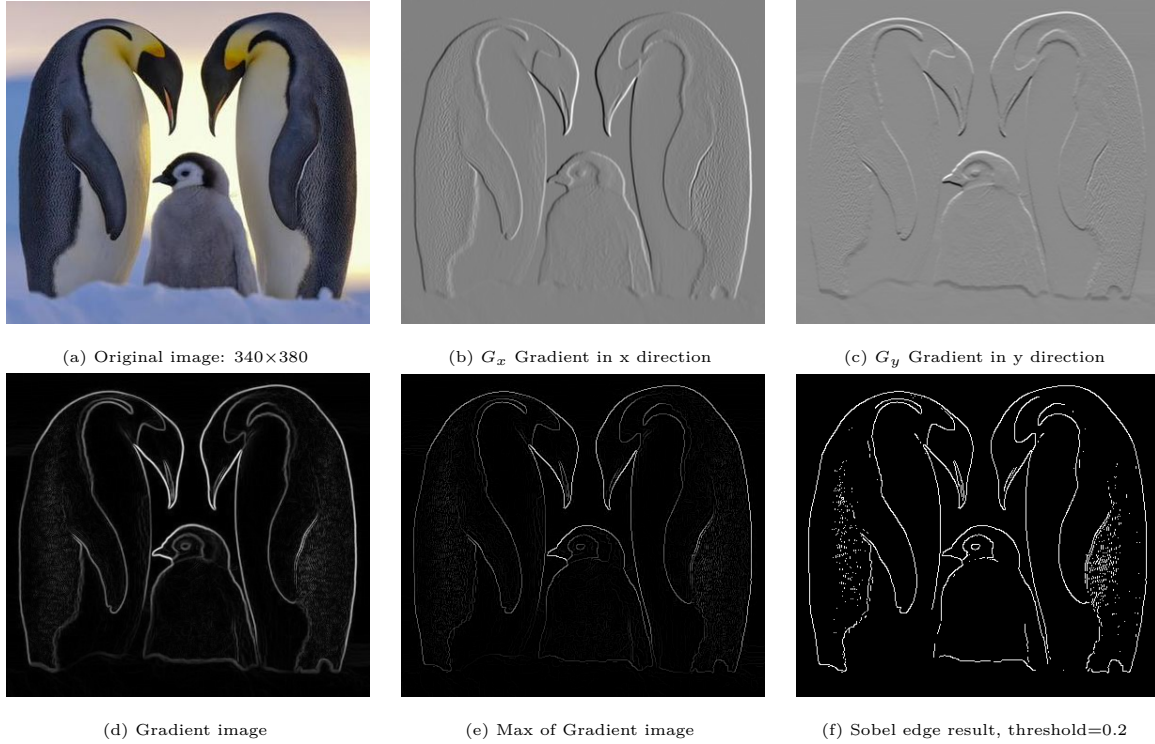


Figure 3.5: All Sobel edge detector process steps.

3.4.2 Canny edge detector

The Canny edge detector [Can86b], is one of the optimal edge detection method. This filter works in a multi-step process. First step, smooths the image with Gaussian filter to

reduce and eliminate the image noise.

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.13)$$

The second steps computes the horizontal and vertical gradient. the gradient is the first-order derivative of the image for each direction to find the regions with high spatial derivatives. The next step, is non-maximal suppression, this process results in a one pixel wide. More deeply, passes along all the regions and suppresses any pixel that is not at the maximum value, thin edges by applying(non-maximum suppression described in previous section 3.3.2) to the gradient magnitude. If the pixel value is not greater than its neighbourhood pixels, then this pixel is not the edge. Important point, this process needs to know the direction of the gradient vectors in order to find two neighbour pixels on the same direction. The edge direction is grouped into eight directions. Finally, edge tracking by hysteresis threshold. The gradient array reduced by using hysteresis thresholds. This hysteresis helps to remove the weak edges and set them to zero. To construct or connect edges, starting from the pixel which is greater than high threshold and search all the eight adjacent neighbors pixels. Also, if the neighbour pixel is greater than low threshold, then it will be an edge too. The range of the hysteresis threshold is ($0 < T_{low} < T_{high} < 1$). Fig. 3.6 illustrates the complete Canny edge detection process on the Lena image including all the intermediate results with $\sigma=3$, $T_{high}=0.3$ and $T_{low}=0.1$.

3.5 Filter banks

In the domain of image analysis, the estimation of a precise gradient is crucial, and is often based on the computation of local derivatives. When using filters with image derivatives in only two directions, the x and y (i.e 0° and 90°), the gradient estimation is not accurate enough to describe the geometrical structures in the image. Using an orientation filter bank can improve the quality of the gradient estimation; indeed, its orientation and its magnitude are far more accurate. Filter banks, are a general framework where the output results from the combination of a set of filters sharing some properties. In general, the filter banks designed for curvilinear structures are oriented, i.e. each filter of the bank is sensitive to a different orientation. By combining all the oriented filters, filter banks are able to detect structures in arbitrary orientations. Even if some filter banks are derivative based, the unifying idea remains the combination of oriented filters. the goal in choosing the various filter bank parameters is to minimize the error. Due to multiple orientations, a filter bank allows to detect several image features such as contours, edges, corners. There are popular filters bank used in successful approaches for processing images, such as steerable Gaussian filter [FA91] and Gabor filter bank [MM96] Filter orientation banks, can be used in many applications such as biometric systems, image retrieval and medical imaging.

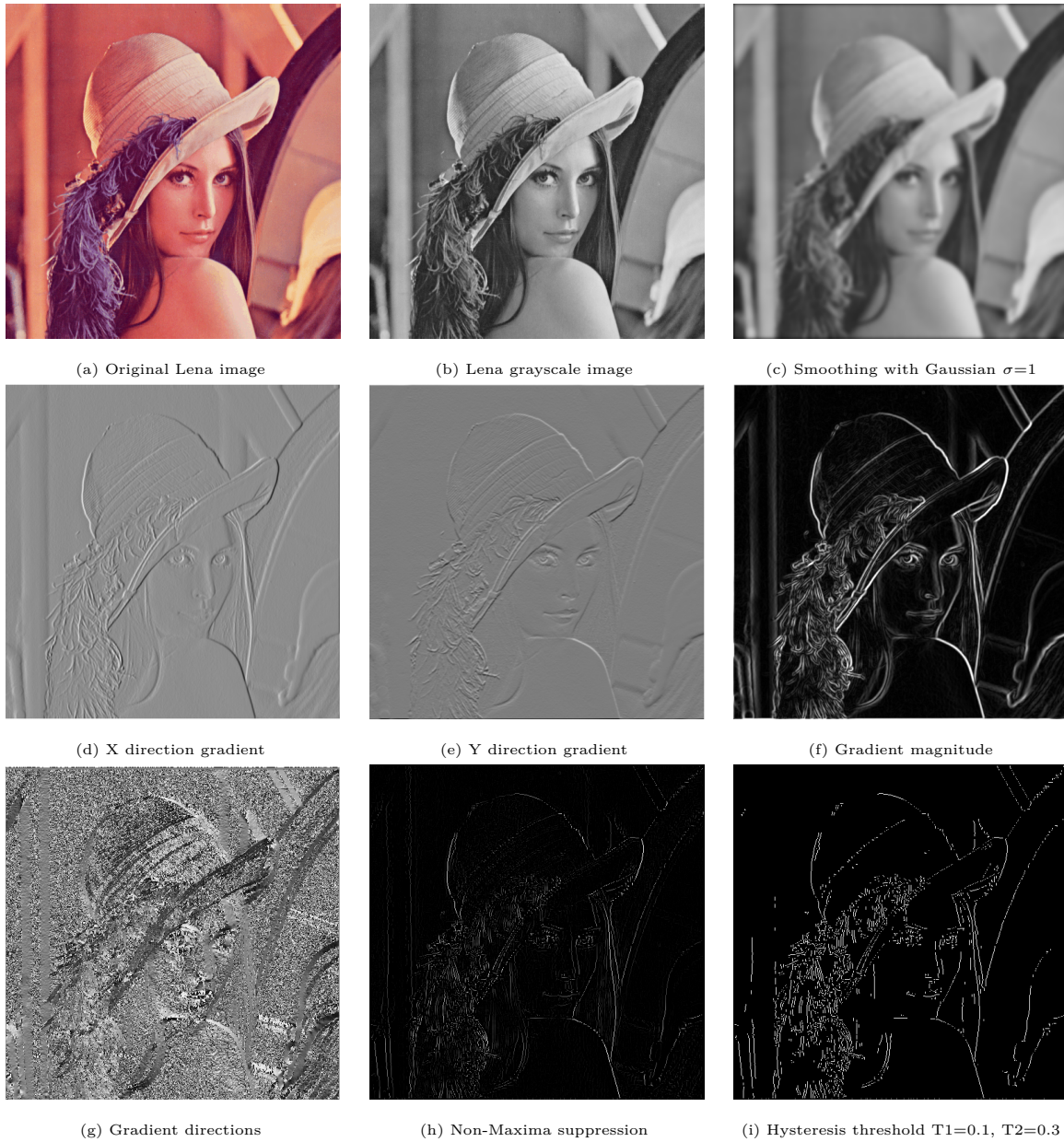


Figure 3.6: All Canny edge detector process steps.

3.5.1 Steerable Gaussian Filter

One of the most popular filter banks is the steerable filters. As a solution to the above stated problem, Freeman and Adelson [FA91] introduced an elegant way for steerable filters that can be directed at specific angles using a linear combination of isotropic filters like Gaussian derivatives. Let us note the basic derivatives of Gaussian filters $\partial \mathcal{G}_\sigma / \partial x$ and

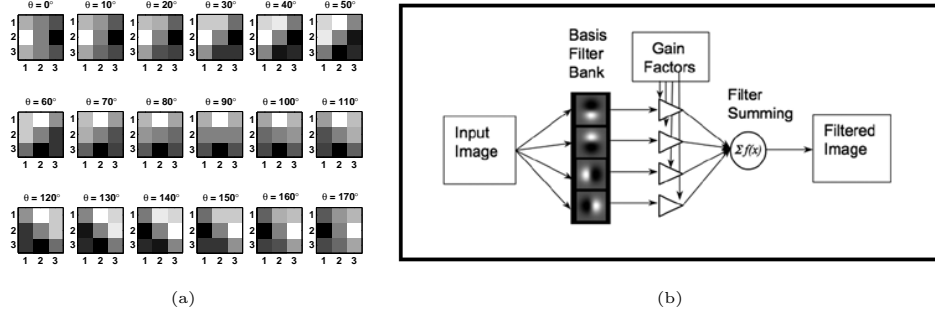


Figure 3.7: (a) Set of steerable Gaussian filters $\mathcal{G}_{\sigma, \theta}$ with $(\sigma=1.5, \Delta\theta = 10^\circ)$, starting from upper left $\theta = 0^\circ$. (b) The steerable filter banks block diagram. Figure obtained from [Kri14].

$\partial\mathcal{G}_\sigma/\partial y$ along the x -axis and y -axis respectively, for example:

$$\begin{cases} \frac{\partial\mathcal{G}_\sigma(x, y)}{\partial x} = \frac{-x}{2\pi\sigma^4} \cdot e^{-\frac{x^2 + y^2}{2\sigma^2}} \\ \frac{\partial\mathcal{G}_\sigma(x, y)}{\partial y} = \frac{-y}{2\pi\sigma^4} \cdot e^{-\frac{x^2 + y^2}{2\sigma^2}}, \end{cases} \quad (3.14)$$

with σ the standard-deviation of the Gaussian filter. Freeman and Adelson have shown that the first order directional Gaussian derivative $\mathcal{G}_{\sigma, \theta}$ at an angle θ can be generated by a linear combination of a rotation of the basic derivatives of isotropic Gaussian filters illustrated in Fig. 3.7 and Fig. 10.1 shown how features extraction by using steerable filter banks [ACMM16a]:

$$\mathcal{G}_{\sigma, \theta}(x, y, \sigma) = \cos(\theta) \cdot \frac{\partial\mathcal{G}_\sigma}{\partial x}(x, y) + \sin(\theta) \cdot \frac{\partial\mathcal{G}_\sigma}{\partial y}(x, y). \quad (3.15)$$

The image derivative $I_{\sigma, \theta}$ is obtained by convolving the original grayscale image I with the oriented Gaussian kernels $\mathcal{G}_{\sigma, \theta}$:

$$I_{\sigma, \theta}(x, y) = (I * \mathcal{G}_{\sigma, \theta})(x, y). \quad (3.16)$$

Finally, the gradient magnitude $\|\nabla I(x, y)\|$ is calculated as the maximum absolute value response to the oriented operator $\mathcal{G}_{\sigma, \theta}$:

$$\|\nabla I(x, y)\| = \max_{\theta \in [0, 180[} (|I_{\sigma, \theta}(x, y)|). \quad (3.17)$$

3.5.2 Anisotropic Gaussian kernels

Anisotropic Gaussian kernels (AGK), can be used in many applications, one of its most important use is image smoothing and for edge detection. The anisotropic Gaussian kernel $G_{\sigma_x, \sigma_y}(x, y)$ can given by:

$$G_{\sigma_x \sigma_y}(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{\frac{-x^2}{2\sigma_x^2}} e^{\frac{-y^2}{2\sigma_y^2}}. \quad (3.18)$$

Perona *et al.* [GSvdW02] used the function $G_{\sigma_x, \sigma_y}(x, y)$ to obtained higher smoothing in the direction of the filter, by use the ratio $\frac{\sigma_x}{\sigma_y}=3$. Fig. 3.8 show the anisotropic oriented kernel with sample example of orientated kernel. More deeply, the edges are detected by computing the 1st derivative of the anisotropic kernels as in Eq. 3.19 and by rotating the filter in different orientations and retaining the orientation which produces the maximum of energy. Fig. 3.12 (a) illustrates the (AGK) derivatives at different orientations θ .

$$G'_{\sigma_x \sigma_y}(x, y) = \frac{\partial G_{\sigma_x \sigma_y}(x, y)}{\partial x} = \frac{-x}{\pi\sigma_x\sigma_y} e^{\frac{-x^2}{2\sigma_x^2}} e^{\frac{-y^2}{2\sigma_y^2}}, \quad (3.19)$$

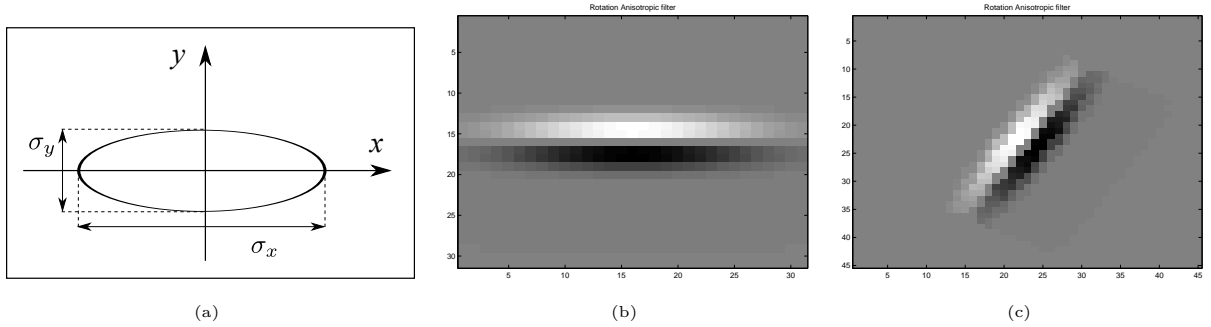


Figure 3.8: Anisotropic orientation kernel $G_{\sigma_x, \sigma_y}(x, y)$ and example of orientations $G'_{\sigma_x \sigma_y}(x, y)$. (a) AGK filter. (b) Anisotropic Oriented kernel by 0° , (c) Oriented kernel by 60° .

3.5.3 Half Gaussian kernel

We can describe problems with the anisotropic filter. Suppose we have application of anisotropic kernel with the outline of an object as shown in Fig.3.9(a) In general, the filter

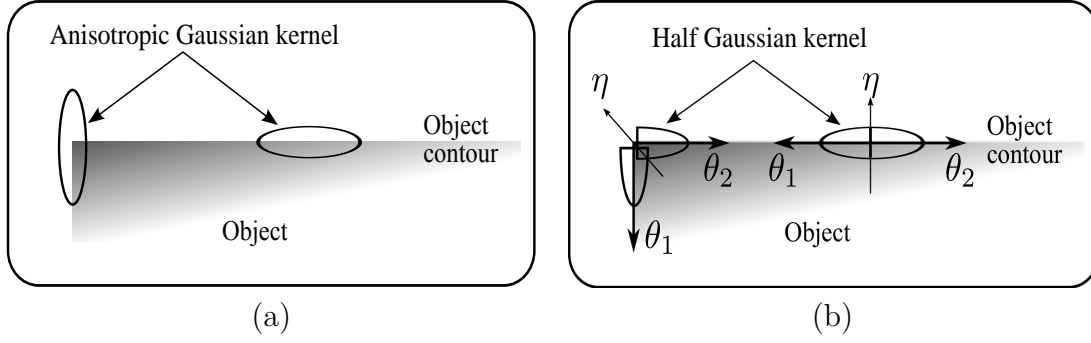


Figure 3.9: Address the problem on Anisotropic Gaussian kernel. This figure obtained from [MMD11a]. (a) Application of Anisotropic Gaussian kernel at the edge and corner of an object. (b) Application of Anisotropic half Gaussian kernel at at the edge and corner of the same object.

is applied at two locations, one on the horizontal line of the edge where the response of the anisotropic filter is high and work very good and the second location, is applied at the corner of the an object. Here, we address the problem as shown in the Fig.3.9 (a) only small part of the anisotropic filter takes into account the information. Therefore, the filter response is greatly reduced where the results are particularly affected by noise.

Wherefore, the solution for anisotropic kernel problem is solve by the Half Gaussian kernel (HGK) as shown in the Fig.3.9 (b), which is one of anisotropic filters family able to detecting the difficult edges, corners and junctions in the image. This new filter produced by Magnier et al. [MMD11a]. The equation of the Half Gaussian Kernels (HGK) derivative is given by the Eq. 3.20:

$$\text{HGK}_{\sigma_\xi, \sigma_\eta}(x, y) = -H(y) \cdot \frac{x}{\sigma_\eta} \cdot G_{\sigma_\xi, \sigma_\eta}(x, y), \quad (3.20)$$

Where H corresponds to the Heaviside function, it helps in the recursive implementation of the (HGK), $(x, y) \in \mathbb{R}^2$ represent the pixel coordinates, σ_η and σ_ξ are referred to the Gaussian scale. Fig. 3.10 show the important parameters that controls the size of the HGK filter.

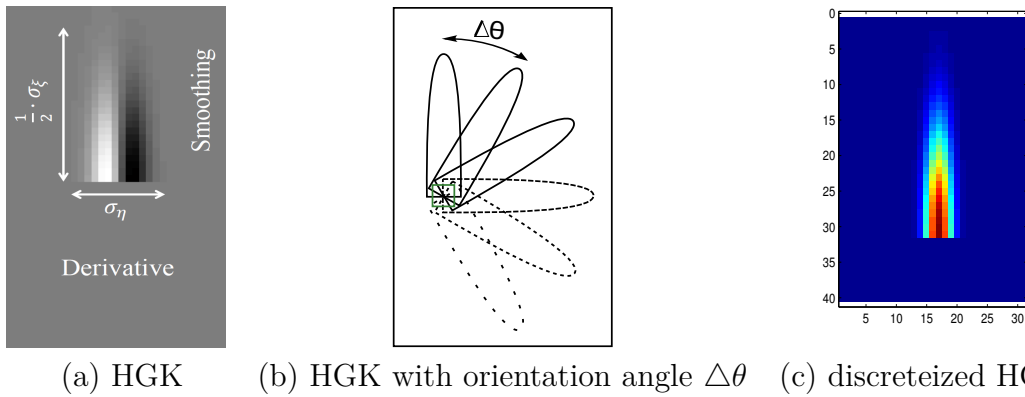


Figure 3.10: The figures (b) and (c) obtained from [MMD11a] show the HGK with full derivative and half smoothing. (b) example of HGK orientations. (d) example of discrete HGK.

First step rotating the image at some discretized orientations from 0 to 360 degrees and convolving the image with the oriented kernels. In a second step, they apply an inverse rotation of the smoothed image and obtain a bank of $360/\Delta\theta$ images. maxima of the filter responses indicate the directions of the edges from 0 to 2π [MMD11a, MM14a]. The methodology is shown in Fig. 3.11 to clearly all these process. The Half Gaussian

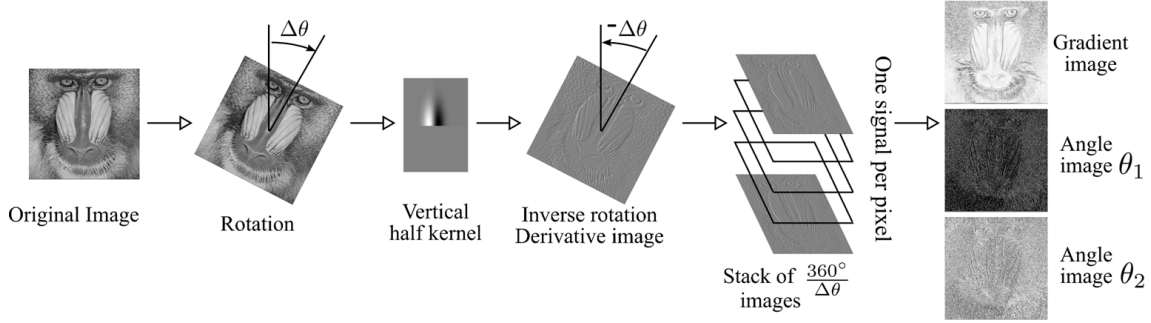


Figure 3.11: Methodology of the Half Gaussian kernels process steps.

kernels have been used in many applications such as edges detection in color images [MM10], anisotropic diffusion [MM14b] enhancement image or texture removal in color image by pixel classification [MMD11b, MMD11c], medical imaging approach [MMD13] and for Image deblurring and Regularization [MXM13]. Moreover, have been used in the new approaches of image descriptors [VMDM15a, VMDM15b] and in corner detection approach [AMM17c]. Fig. 3.12 (a) and (b) illustrates the comparison between anisotropic kernel and Half Gaussian kernel derivatives at different orientations θ , as shown in this figure the edges in each θ is more clear and good localization, while the detector used an anisotropic kernel the edges are more blurring for each orientations. As an example, for real image, illustrated in Fig. 3.13 to compare eight edge detectors still has scope and achieved that Half Gaussian kernel is best detector than other detector methods. Furthermore experiments, illustrates in Fig. 3.14 and Fig. 3.15 Half Gaussian kernel (HGK) remains more stable and robust to blur and noise than the comparative methods.

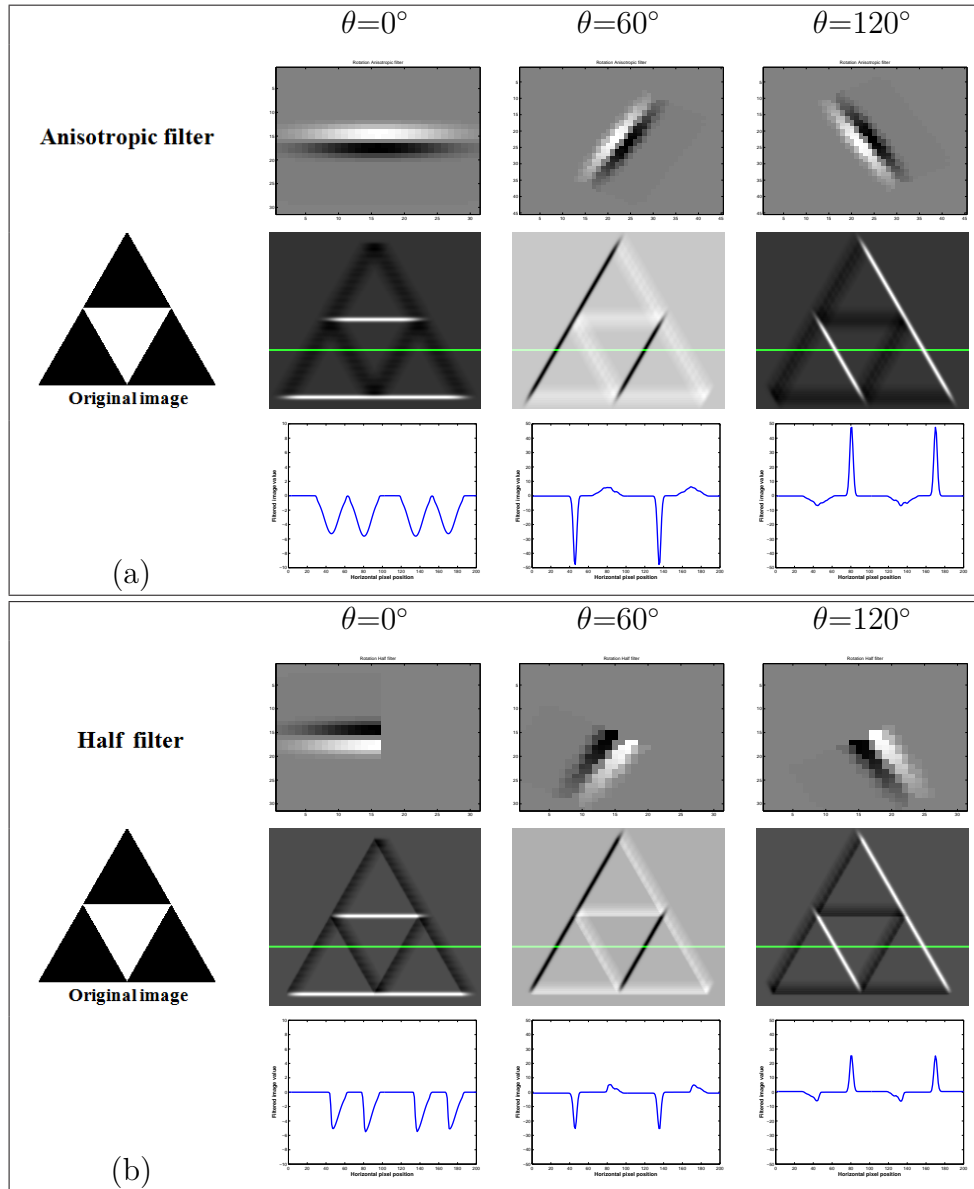


Figure 3.12: Comparison between Anisotropic kernel and Half filter derivatives at different orientations θ , (a) Anisotropic filter with $\sigma_x=1.5$, $\sigma_y=7.5$. (b) Half Gaussian filter with $\lambda=1.5$, $\mu=3.75$.



Figure 3.13: Example for real image: differences edge detector methods result.

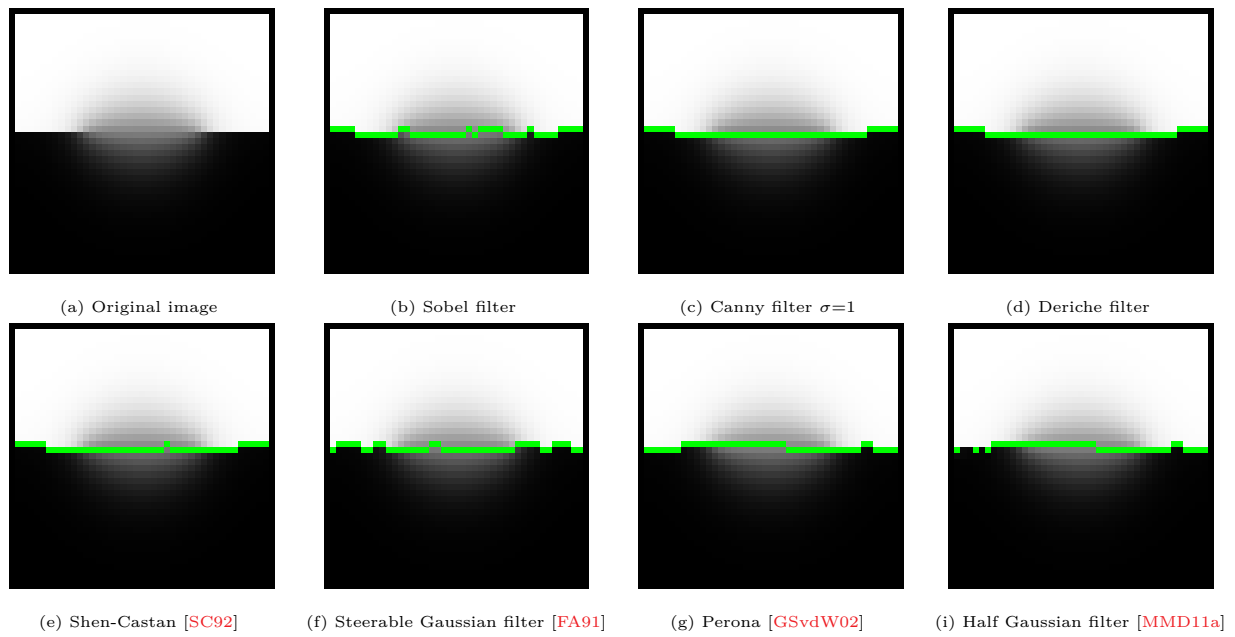


Figure 3.14: Comparison differences edge detector methods for blurry edge noise.

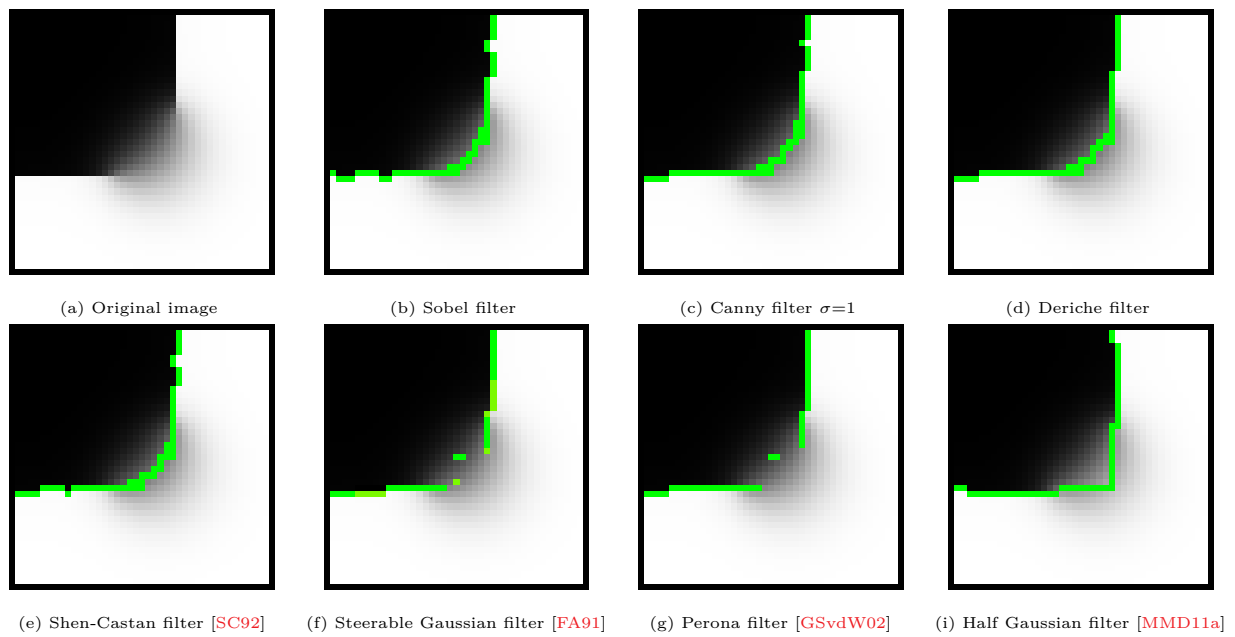


Figure 3.15: Comparison differences edge detector methods for blurry corner noise.

3.6 Summary

In this chapter, we begin with background of edges and corners detection, mainly including methods for extracting edge features and the performance evaluation for edge and corner detection methods. Here different traditional edge operators based on gradient group, intensity and filter response are presented. We explained about the post processing steps for each operator. More deeply, we focused more on the filters in the particular about the filter banks, steerable Gaussian filters, Anisotropic kernels, Half Gaussian kernels, which forms the basis of this thesis. In the rest of the chapter we refer to Anisotropic Gaussian kernels as (AGK), Half Gaussian Kernel as (HGK). Here, we have also presented the advantage of splitting the Anisotropic Gaussian kernel to constructing the (HGK) filter. Finally, in this chapter different experiments have been presented to highlight the results of the edge detector filters with real and synthetic images, also with noise and without noise.

NEW NORMALIZED SUPERVISED EDGE DETECTION EVALUATION METHOD

This chapter describes the proposed method for normalized supervised edge detection evaluation.

Contents

4.1	Introduction	48
4.2	Supervised image contour evaluation	48
4.2.1	Error measures involving the confusion matrix only	49
4.2.2	Assessment involving distances of misplaced pixels	50
4.2.3	Normalization of the edge detection evaluation:	52
4.3	A new edge detection assessment measure	53
4.4	Experimental results	53
4.5	Summary	62

4.1 Introduction

In image processing tasks, edge detection remains a key point in many applications. Boundaries include the most important structures of the image, and an efficient boundary detection method should create a contour image containing edges at their correct locations with a minimal of misclassified pixels. Different algorithms have been developed in the past, but few of them give an objective performance comparison. The evaluation process should produce a result that correlates with the perceived quality of the edge image, which is relied on human judgment. In other words, a reliable edge map should characterize all the relevant structures of an image. On the other hand, a minimum of spurious pixels or holes (oversights) must be created by the edge detector at the same time. Therefore, an efficient evaluation can be used to assess and improve an algorithm, or to optimize edge detector parameters [HSSB97].

The measurement process can be classified into either supervised or unsupervised evaluation criteria. The first class of methods exploits only the input contour image and gives a score of coherence that qualifies the result given by the algorithm [HSSB97]. The second class computes a dissimilarity measure between a segmentation result and a ground truth obtained from synthetic data or an expert judgment (i.e. manual segmentation) [DJ94, CLRE08, LMDBB13]. This work focusses on comparisons of supervised assessment of edge detection evaluations. Furthermore, a new supervised edge map quality measure based on the distances of misplaced pixels is presented and compared to the others, using synthetic and real images. In digital images, edges characterize object boundaries, then their detection remains a crucial stage in numerous applications. To achieve this task, many edge detectors have been designed, producing different results, with different qualities. Evaluating the response obtained by these detectors has become a crucial task.

In this chapter, several referenced-based boundary detection evaluations are detailed, pointing their advantages and disadvantages through concrete examples of edge images. Then, a new supervised edge map quality measure is proposed, comparing to a ground truth contour image, the candidate contour image and their associated spacial nearness. Compared to other boundary detection assessments, this new method has the advantage to be normalized and remains more reliable edge map quality measure.

4.2 Supervised image contour evaluation

As introduced above, a supervised evaluation process estimates scores between a ground truth and a candidate edge map. In image processing, the Structural Similarity Index

$\text{Measure}(SSIM)$ corresponds to an image quality evaluation, which estimates the visual impact of gray scale shifts in an image [WBSS04]. Otherwise, contours (binary images) could be evaluated counting the number of erroneous pixels, but also throughout spatial distances of misplaced or oversights contours.

4.2.1 Error measures involving the confusion matrix only

Let G_t be the reference contour map corresponding to ground truth and D_c the detected contour map of an image I . Comparing pixel per pixel G_t and D_c , common positive or negative presence of points is the first criterion to be assessed. A basic evaluation is compounded of statistics issued of a confusion matrix. To that effect, G_t and D_c are combined. Afterward, denoting $|\cdot|$ the cardinality of a set, all points are partitioned into four sets:

- True Positive points (TPs), common points of G_t and D_c : $TP = |D_c \cap G_t|$,
- False Positive points (FPs), spurious detected edges of D_c : $FP = |D_c \cap \neg G_t|$,
- False Negative points (FNs), missing boundary points of D_c : $FN = |\neg D_c \cap G_t|$,
- True Negative points (TNs), common non-edge points: $TN = |\neg D_c \cap \neg G_t|$.

In one hand, let us consider boundary detection of images, FPs appear in the presence of noise, texture or other contours influencing the filter used by the edge detection operator. In the other hand, FNs represent holes in a contour of D_c . Finally, a wrong threshold of the segmentation could generate both FPs and FNs. Computing only FPs and FNs enables a segmentation assessment [MMD11a, MAB⁺14], and a reliable edge detection should minimize the following indicators [CLRE08]:

$$\begin{cases} \text{Over-detection error} : \text{Over}(G_t, D_c) = \frac{FP}{|I| + |G_t|}, \\ \text{Under-detection error} : \text{Under}(G_t, D_c) = \frac{FN}{|G_t|}, \\ \text{Localization-error} : \text{Loc}(G_t, D_c) = \frac{FP + FN}{|I|}. \end{cases}$$

Additionally, the *Performance measure* (P_m^*) presented in Table 4.1 considers directly at the same time the three entities TP , FP and FN to assess a binary image. The obtained score reflects the percentage of statistical errors.

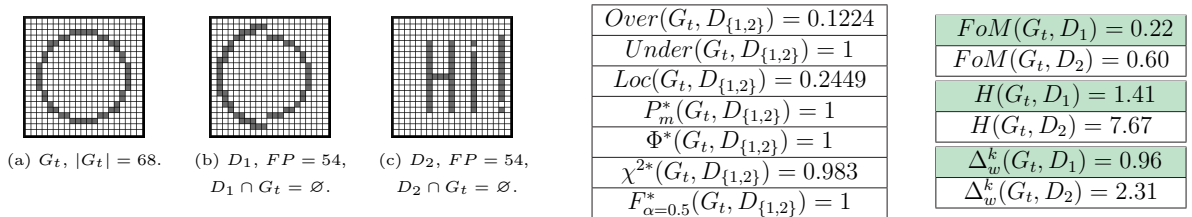


Figure 4.1: Evaluations issued of a confusion matrix can be the same for different D_c . For the two candidate edge images, number of FPs and number of FNs are the same.

Another way to display evaluations are Receiver Operating Characteristic (ROC) [BKD01] curves or Precision-Recall (PR) [MFM04], involving *True Positive Rates* ($TPR = \frac{TP}{TP+FN}$) and *False Positive Rates* ($FPR = \frac{FP}{FP+TN}$). Derived from TPR and FPR , the three measures Φ , χ^2 and F_α (see Table 4.1) are frequently used in edge detection assessment.

P_m^* , Φ , χ^2 and F_α measures are normalized and decrease with the quality of the detection; a score equal to 0 qualifies a perfect segmentation. These measures evaluate the comparison of two edge images, pixel per pixel, tending to penalize severely a misplaced contour (even weak). So they do not indicate significant variations of the desired contour shapes through an evaluation (as illustrated in Fig. 4.1). As this penalization tends to be too severe, some evaluations issued from the confusion matrix recommend a spatial tolerance, particularly for assimilation of TPs [BKD01, MFM04]. This inclusion could be carried by a distance threshold or a dilation of D_c and/or G_t . A such strategy of assimilation leads to counting several near contours as parallel stripes to the desired boundary. Tolerating a distance from the true contour and integrating several TPs for one detected contour are opposite to the principle of unicity in edge detection expressed by the 3rd Canny criteria: an optimal edge detector must produce a single response for one contour [Can86a]. Finally, to perform an edge evaluation, the assessment should penalize a misplaced edge point proportionally to the distance to its true location.

4.2.2 Assessment involving distances of misplaced pixels

A reference-based edge map quality measure requires that a displaced edge expects to be penalized in function not only of the FPs and/or FNs but also in function of the distance to the position they should be located at. Table 4.1 reviews the most relevant existing measures. The common feature between these evaluators corresponds to the error distance $d_{G_t}(p)$ or $d_{D_c}(p)$. Indeed, for a pixel $p \in D_c$, $d_{G_t}(p)$ represents the minimal distance between p and G_t , whereas if $p \in G_t$, $d_{D_c}(p)$ corresponds to the minimal distance between p and D_c . This distance refers to the Euclidean distance, even though some authors involve others, see [LMDBB13]. Thus, a measure computing an error distance only in function of d_{G_t} estimates the divergence of FPs, which corresponds to an over-segmentation (cases Υ , D^k ,

(a) G_t (b) I_c (b) I_t

Measure	I_t	I_c
FoM	0.63	0.17
F	0.72	0.18
d_4	0.62	0.15
$SFoM$	0.64	0.17
$MFoM$	0.82	0.58
D_P	0.33	0.007
Υ	1.14	3.30
H	109	130
$H_{5\%}$	10.39	11.01

Measure	I_t	I_c
$D_{k=2}^k$	0.39	0.47
$\theta_{\delta_{TH}=1}$	2.74	9.07
$\theta_{\delta_{TH}=5}$	0.37	3.35
$\Omega_{\delta_{TH}=1}$	7.91	0
Δ^k	6.05	1.20
$f2d6$	5.79	1.60
$S_{k=2}^k$	5.97	2.84
Γ	0.19	0.12
Ψ	0.94	0.12

Figure 4.2: The scores of the over-segmentation evaluations are higher for I_t whereas I_t is more closer visually to G_t than I_c .

Table 4.1: List of error measures, $k = 1$ or $k = 2$ are the most common values.

Error measure name	Formulation	Parameters
<i>Performance measure</i> [GPW03]	$P_m^*(G_t, D_c) = 1 - \frac{TP}{TP + FP + FN}$	None
Complemented Φ measure [VR95]	$\Phi^*(G_t, D_c) = 1 - \frac{TPR \cdot TN}{TN + FP}$	None
Complemented χ^2 measure [YP03]	$\chi^{2*}(G_t, D_c) = 1 - \frac{TPR - TP - FP}{1 - TP - FP} \cdot \frac{TP + FP + FPR}{TP + FP}$	None
Complemented F_α measure [MFM04]	$F_\alpha^*(G_t, D_c) = 1 - \frac{PREC \cdot TPR}{\alpha \cdot TPR + (1 - \alpha) \cdot PREC}$, with $PREC = \frac{TP}{TP + FP}$	$\alpha \in]0; 1]$
Pratt's FoM [AP79]	$FoM(G_t, D_c) = 1 - \frac{1}{\max(G_t , D_c)} \cdot \sum_{p \in D_c} \frac{1}{1 + \kappa \cdot d_{G_t}^2(p)}$	$\kappa \in]0; 1]$
FoM revisited [PA95]	$F(G_t, D_c) = 1 - \frac{1}{ G_t + \beta \cdot FP} \cdot \sum_{p \in G_t} \frac{1}{1 + \kappa \cdot d_{D_c}^2(p)}$	$\kappa \in]0; 1]$ and $\beta \in \mathbb{R}^+$
Combination of FoM and statistics [BG09]	$d_4(G_t, D_c) = \frac{1}{2} \cdot \sqrt{\frac{(TP - \max(G_t , D_c))^2 + FN^2 + FP^2}{(\max(G_t , D_c))^2} + FoM(G_t, D_c)}$	$\kappa \in]0; 1]$ and $\beta \in \mathbb{R}^+$
Yasnoff measure [YGB78]	$\Upsilon(G_t, D_c) = \frac{100}{ I } \cdot \sqrt{\sum_{p \in D_c} d_{G_t}^2(p)}$	None
Hausdorff distance [HR93]	$H(G_t, D_c) = \max\left(\max_{p \in D_c}(d_{G_t}(p)), \max_{p \in G_t}(d_{D_c}(p))\right)$	None
Maximum distance [DJ94]	$f_2 d_6(G_t, D_c) = \max\left(\frac{1}{ D_c } \cdot \sum_{p \in D_c} d_{G_t}(p), \frac{1}{ G_t } \cdot \sum_{p \in G_t} d_{D_c}(p)\right)$	None
Distance to G_t [PM82, LMDBB13]	$D^k(G_t, D_c) = \frac{1}{ D_c } \cdot \sqrt[k]{\sum_{p \in D_c} d_{G_t}^k(p)}$, $k = 1$ for [PM82]	$k \in \mathbb{R}^+$
Oversegmentation [Har84, OBBC02]	$\Theta(G_t, D_c) = \frac{1}{FP} \cdot \sum_{p \in D_c} \left(\frac{d_{G_t}(p)}{\delta_{TH}}\right)^k$, $k = \delta_{TH} = 1$ for [Har84]	for [OBBC02]: $k \in \mathbb{R}^+$ and $\delta_{TH} \in \mathbb{R}_*^+$
Undersegmentation [Har84, OBBC02]	$\Omega(G_t, D_c) = \frac{1}{FN} \cdot \sum_{p \in G_t} \left(\frac{d_{D_c}(p)}{\delta_{TH}}\right)^k$, $k = \delta_{TH} = 1$ for [Har84]	for [OBBC02]: $k \in \mathbb{R}^+$ and $\delta_{TH} \in \mathbb{R}_*^+$
Symmetric distance [DJ94, LMDBB13]	$S^k(G_t, D_c) = \sqrt[k]{\frac{\sum_{p \in D_c} d_{G_t}^k(p) + \sum_{p \in G_t} d_{D_c}^k(p)}{ D_c \cup G_t }}$, $k = 1$ for [DJ94]	$k \in \mathbb{R}^+$
Baddeley's Delta Metric [Bad92]	$\Delta^k(G_t, D_c) = \sqrt[k]{\frac{1}{ I } \cdot \sum_{p \in I} w(d_{G_t}(p)) - w(d_{D_c}(p)) ^k}$	$k \in \mathbb{R}^+$ and a convex function $w : \mathbb{R} \mapsto \mathbb{R}$
Edge map quality measure [PGAN16]	$D_p(G_t, D_c) = \frac{1/2}{ I - G_t } \cdot \sum_{p \in D_c} \left(1 - \frac{1}{1 + \alpha \cdot d_{G_t}^2(p)}\right) + \frac{1/2}{ G_t } \cdot \sum_{p \in G_t} \left(1 - \frac{1}{1 + \alpha \cdot d_{G_t \cap D_c}^2(p)}\right)$	$\alpha \in]0; 1]$
Magnier <i>et al.</i> measure [MLZ16]	$\Gamma(G_t, D_c) = \frac{FP + FN}{ G_t ^2} \cdot \sqrt{\sum_{p \in D_c} d_{G_t}^2(p)}$	None

Θ , FoM and Γ). On the contrary, the sole use of a distance d_{D_c} enables an estimation of the FNs divergence, representing an under-segmentation, as Ω distance measure. A measure widely computed in matching techniques is represented by the Hausdorff distance H which estimates the mismatch of two sets of points [HR93]. This max-min distance

could be strongly deviated by only one pixel which can be positioned sufficiently far from the pattern (illustrated in Fig. 4.2); so the measured distance becomes that between the pattern and the (erroneous) point, disturbing in that case the score of H . To improve H such that it becomes less sensitive to outliers, an idea is to compute H with a proportion of the maximum distances (for example 5% of the values [HR93]); let us note $H_{n\%}$ this measurement for $n\%$ of values ($n \in \mathbb{R}_*^+$). One of the most popular descriptor corresponds to Figure of Merit (FoM). This distance measure ranges from 0 to 1, where 0 corresponds to a perfect segmentation [AP79], but computes only distances of the FPs [Bad92]. Thus, some improvements have been developed as F and d_4 . Furthermore, as concluded in [CLRE08], a complete and optimum edge detection evaluation measure should combine assessments of both over- and under-segmentation, as in S^k , Δ_w^k and D_p . As an example, inspired by f_2d_6 [DJ94], another way is to consider the combination of both $FoM(G_t, D_c)$ and $FoM(D_c, G_t)$, as the two following formulas:

$$\text{Symmetric } FoM: SFoM(G_t, D_c) = \frac{1}{2} \cdot FoM(G_t, D_c) + \frac{1}{2} \cdot FoM(D_c, G_t) \quad (4.1)$$

$$\text{Maximum } FoM: MFoM(G_t, D_c) = \max(FoM(G_t, D_c), FoM(D_c, G_t)). \quad (4.2)$$

Finally, $SFoM$ and $MFoM$ take into account both distances of FNs (i.e. d_{D_c}) and FPs (i.e. d_{G_t}), so they can compute a global evaluation of a contour image.

Another way to compute a global measure is presented in [PGAN16] with the normalized edge map quality measure D_p . In fact, this distance measure seems similar to $SFoM$ with different coefficients. However, both the left and the right terms are composed of a $\frac{1}{2}$ coefficient, so in the presence of only a pure under- or over-segmentation, the score of D_p does not attain over $\frac{1}{2}$.

4.2.3 Normalization of the edge detection evaluation:

In order to compare each boundary detection assessments, all measures must be normalized, but also indicate the same information: an error measure close to 1 means a poor segmentation whereas a value close to 0 indicates a good segmentation. Thereby, the values of FoM , F , d_4 and D_p belongs to $[0, 1]$. However, concerning other distance measures of table 4.1, a normalization is required. Introduced in [MLZ16], a formula called Key Performance Indicator (KPI) gives value close to 1 for a poor segmentation; alternatively, a KPI value close to 0 translates a good segmentation:

$$\begin{aligned} KPI_u : [0; \infty[&\mapsto [0; 1[\\ u &\rightarrow 1 - \frac{1}{1 + u^h}. \end{aligned} \quad (4.3)$$

where the parameter u represents a distance error and h a constant ($h \in \mathbb{R}_*^+$).

An undeniable parameter of KPI formula is the power of the denominator term called h . Inasmuch as KPI depends on its value, it evolves more or less quickly around 0.5 and

embodies a range of observable cases. The advice to choose values between 1 and 2 can be easily checked. Otherwise, the more *KPI* evolution will be abrupt, the less the transition between 0.5 and 1 will be marked. As a compromise, fixing the power at the *golden ratio* $\phi \simeq 1.6180339887$, the measurement becomes not too strong in the presence a small measure score, but increases to 1 for a high score of the distance measure [MLZ16].

4.3 A new edge detection assessment measure

In [MLZ16] is developed a normalized measure of the edge detection assessment, denoted Γ . This function represents an over-segmentation measure which depends also of FN and FP . As this measure is not sufficiently efficient concerning FNs because it does not consider d_{D_c} for false negative points (see Fig. 4.7). Thus, inspired by S^k , the new measure Ψ holds different coefficients changing the behavior of the measure:

$$\Psi(G_t, D_c) = \frac{FP + FN}{|G_t|^2} \cdot \sqrt{\sum_{p \in G_t} d_{D_c}^2(p) + \sum_{p \in D_c} d_{G_t}^2(p)} \quad (4.4)$$

Compared to Γ , Ψ improves the measurement by combining both d_{G_t} and d_{D_c} (illustrated in Fig. 4.7). Authors of Γ have studied the influence of the coefficient in different concrete cases [MLZ16]. They concluded that such a formulation must take into consideration all observable cases and theoretically observable. In fact, a performing measure has to take into account all the following input parameters $|G_t|$, FN and FP whereas the image dimensions should not be considered. Thus, the parameter $\frac{FP+FN}{|G_t|^2}$ seems a good compromise and has been introduced to the new formula of assessment Ψ .

4.4 Experimental results

Experiments realized in this part aim to be the most accomplished, thus the more close and realistic of the reality. In respect of these directives, in a first time, considering a synthetic edge model (i.e. ground truth) the edge detection evaluation measures are subject to the following studies:

- Addition of false positive points close to the true contour,
- Addition of false negative points (under-segmentation),
- Addition of false positive points (over-segmentation),
- Addition of both false negatives and false positive points,

- Translation of the boundary.

Thus, 24 measures and the new proposed method are tested and compared together. The KPI in eq. 4.3 is computed for the non-normalized algorithms in Table 4.1.

The first experiment is to create an over-segmentation at a maximal distance of 5 pixels, as illustrated in Fig. 4.3 (100% of over-segmentation represents a dilatation of the vertical line with a structural element of size 1×6 , corresponding of a total saturation of the contour, see Fig. 4.3(d)). Curves presented here show that F_α , d_4 , F , $MFoM$, H , $H_{5\%}$, Δ_k , D^k , f_2d_6 , and S^k are very sensitive to FPs, whereas $SSIM$, D_p , Φ and D^k (which is not homogeneous) do not penalize enough D_c . Ω remains constant at 0 because it corresponds to an under-segmentation measure. Moreover, Υ and FoM are not too abrupt, even though they stagnate, like D^k , $SFoM$, f_2d_6 , and S^k . Finally, Γ and Ψ are not too abrupt and penalize strongly D_c in the presence of many FPs.

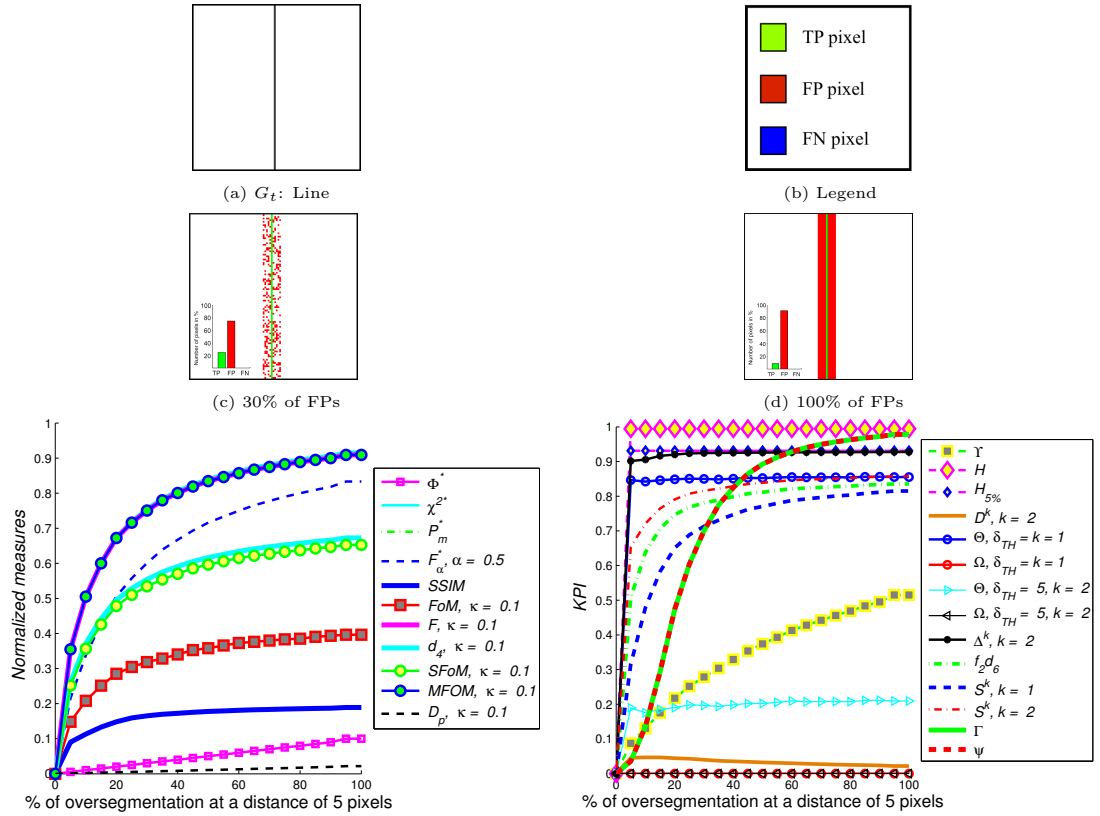


Figure 4.3: Measures scores in function of the over-segmentation in the contour area.

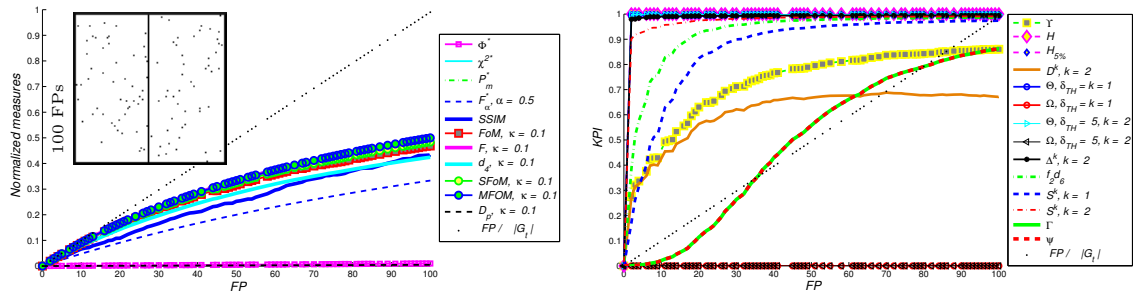


Figure 4.4: Evolution of the dissimilarity measures in function of the FPs addition.

The second test is to add random undesirable pixels to G_t until 100 FPs, as represented in Fig. 4.4 top left. Globally, the curves in Fig. 4.4 illustrates that the measures using KPI behave like the previous experiment; only Γ and Ψ are not too sensitive to FPs. The normalized evaluation measures increase correctly, but seem stagnant, excepted Φ and D_p which stay close to 0.

Concerning the addition of FNs, Fig. 4.5 (left) illustrates that H , $H_{5\%}$ and $\Omega_{\delta_{TH}=1}$ are very sensitive to the presence of FNs. Also, D_p attains only the score of $\frac{1}{2}$. The over-segmentation methods Υ , D^k , Ω and Γ remain constant at 0. On the other hand, the score of the KPI of Ψ attains 0.5 for 50% of FNs. Afterward, contrary to the addition of FPs or FNs, error measures without distance measures obtains a score of 1 after one pixel of translation and the score of D_p stays constant at $\frac{1}{2}$ (Fig. 4.5 (right)). Only FoM , $SFoM$, $MFoM$, the KPI of $\Omega_{\delta_{TH}=5}$, the KPI of Γ and the KPI of Ψ behave correctly.

Concerning the line, the last experiment corresponds to an addition of both FPs and FNs. Thus, Fig. 4.6 shows that the normalized measures, excepted D_p and $SSIM$ behave correctly. Concerning other measures, the KPI scores of $\Omega_{\delta_{TH}=5}$, Γ and Ψ are not too abrupt for few number of errors and penalize strongly D_c in the presence of many FPs and FNs (but $\Omega_{\delta_{TH}=5}$ is not homogeneous). For example, Fig. 4.6 (bottom right) illustrates the line where both 50% of the pixels are missing and 50 FPs are added, corresponding to 33% of TPs. In this precise case, the KPI score of new measure Ψ is close to 0.7, thus, reflecting the reality.

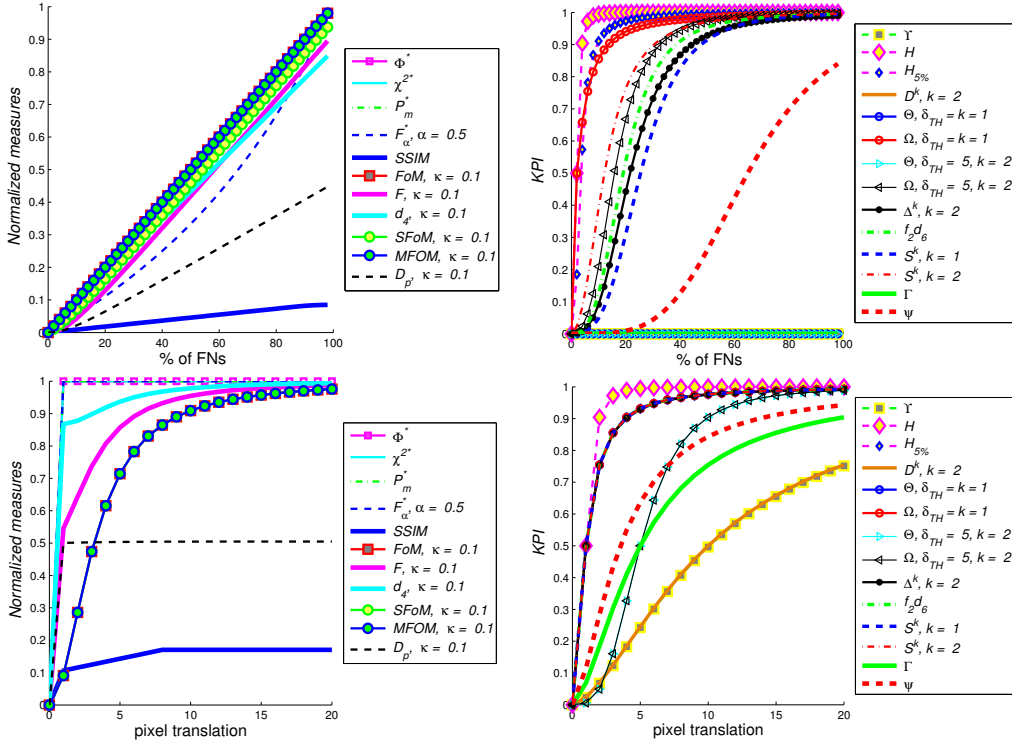


Figure 4.5: Measure scores in function of the FNs addition and the edge translation.

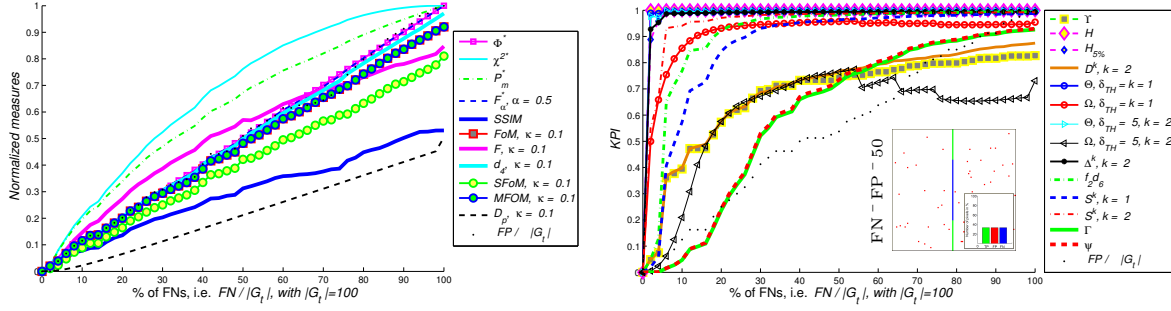


Figure 4.6: Dissimilarity measure scores in function of addition of both the FNs and FPs.

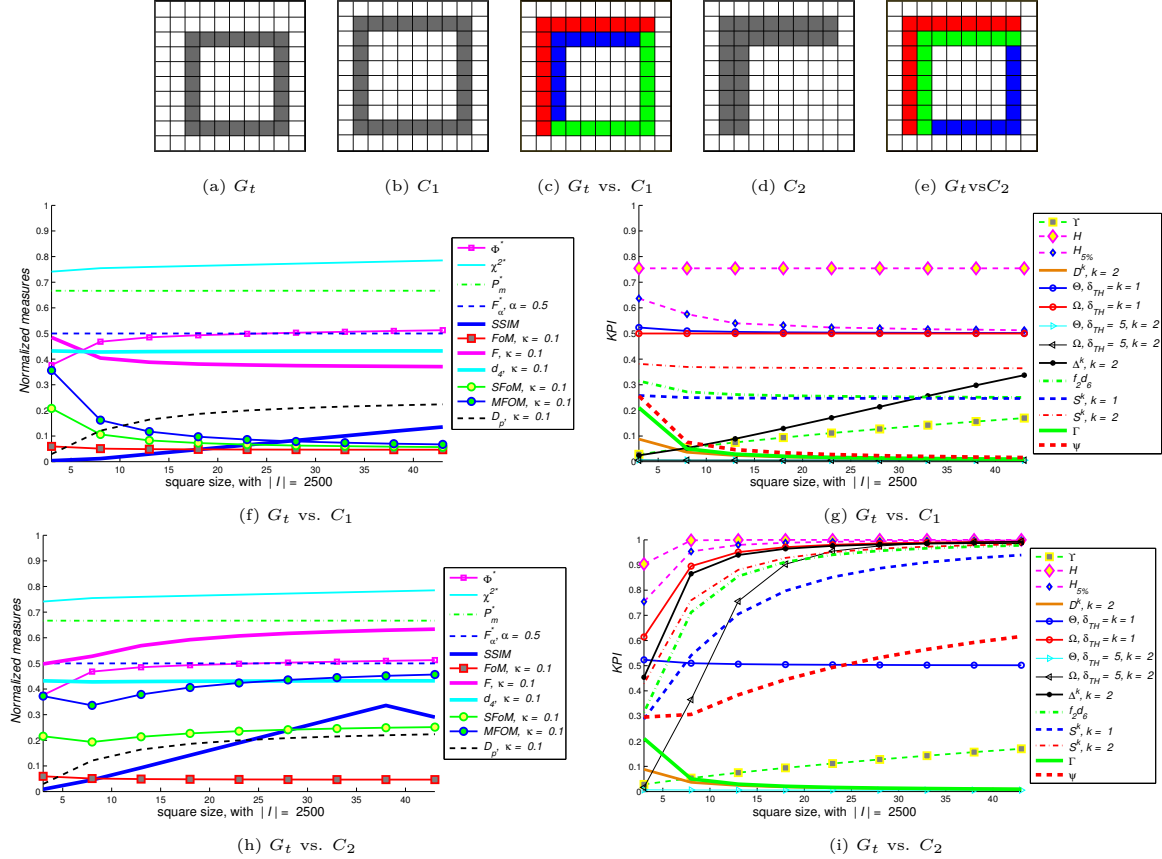


Figure 4.7: Dissimilarity measure scores in function of the size of the original square.

Another experiment in Fig. 4.7, two different shapes are compared to a square (G_t), illustrating the importance to consider both d_{D_c} and d_{G_t} . Furthermore, all the shapes are growing at the same time, keeping the same percentage of FPs and FNs with G_t . The more G_t grows, the more C_1 is visually closer to G_t whereas FNs deviate strongly in the case of C_2 . Despite these two different evolutions, statistical measures, FoM , F , d_4 , D_p , Υ and Γ obtain close the same measurements for C_1 and C_2 . On the contrary, the KPI of Ψ grows around 0.5 for C_2 , whereas it converges towards 0 for C_1 , since C_1 becomes visually closer to G_t with the enlargement (note that $MFoM$ behaves identically).

To conclude experimental evaluations, Fig. 4.8 reports different assessments for five edge detection methods on a real image: Sobel, Canny [Can86a], Steerable Filters of order 1 (SF_1) [FA91], Steerable Filters of order 5 (SF_5) [JU04a] and Half Gaussian Kernels (HK) [MMD11a]. Even though the problem of hand-made ground truth on real images is

discussed by some researchers [HYK13], only the comparison of D_c with a G_t is studied here. Compared to G_t (Fig. 4.8 (a)), the well known Sobel edge detector generates more FPs than the other three methods while SF_5 and HK are less sensitive to noise or texture. Furthermore, HK captures easier straight contours and corners closest to their true positions [MMD11a]. So, the measurements in tables of Fig. 4.8 must be close to 1 for Sobel and a little less for Canny, but decrease with reasonable error for HK (scores involving KPI for non-normalized algorithms, eq. 4.3). Thus, Γ and Ψ respect this evolution and indicate a good measurement value. FoM , F , d_4 , $SFoM$, $MFoM$ and Υ evolve similarly, but the score for the HK remains too elevated. Also, Φ^* , χ^{2*} , P_m^* , F_α^* and D_p do not indicate a significant difference between all the segmentations. Further, other non normalized methods are not adapted to give a score between 0 and 1 using a KPI . Eventually, given the segmented images, Γ and Ψ indicate a good measurement value. Other results involving other edge images Figs. 4.9 to 4.11: Assessment of edge detection evaluations concerning real images, shows that our proposed measure Ψ indicate a good measurement value.

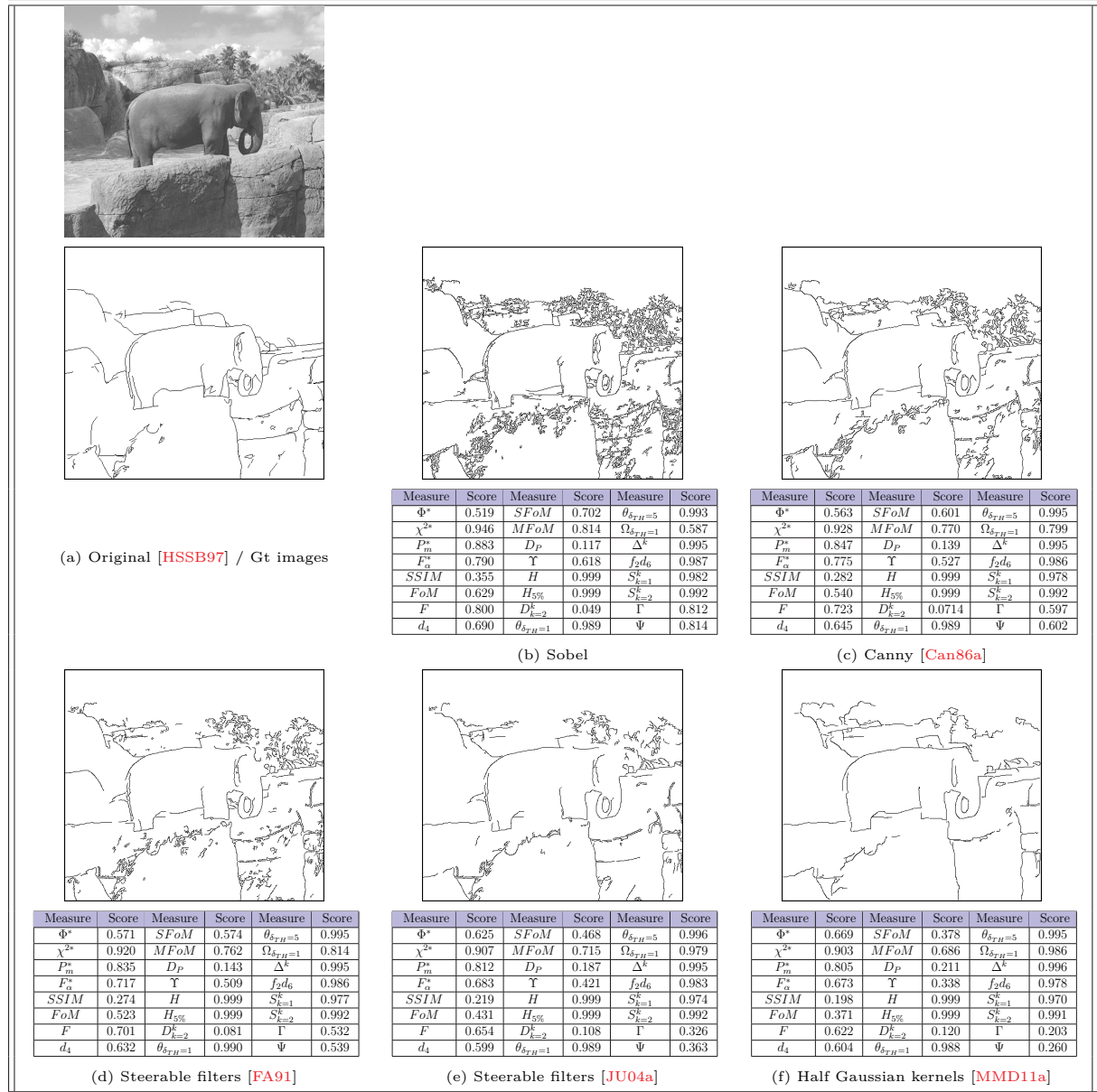


Figure 4.8: Comparison measures of different edge detections involving different edge detectors. All the results are normalized and, compared to the ground truth, a score close to 0 indicates a good edge map whereas a score 1 translates a poor segmentation. Our proposed measure Ψ indicate a good measurement value.

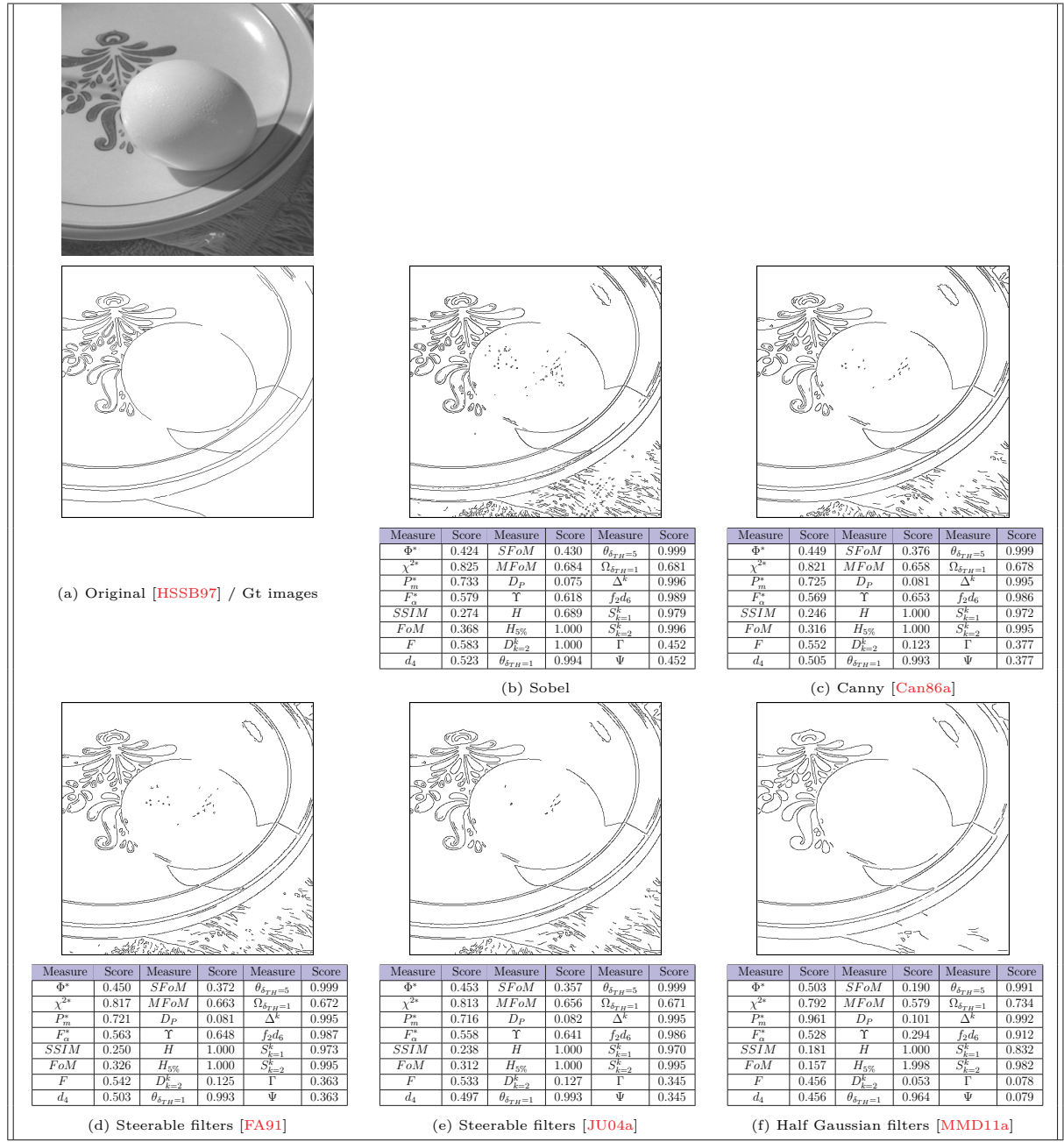


Figure 4.9: Comparison measures of different edge detections involving different edge detectors. All the results are normalized and, compared to the ground truth, a score close to 0 indicates a good edge map whereas a score 1 translates a poor segmentation. Our proposed measure Ψ indicate a good measurement value.

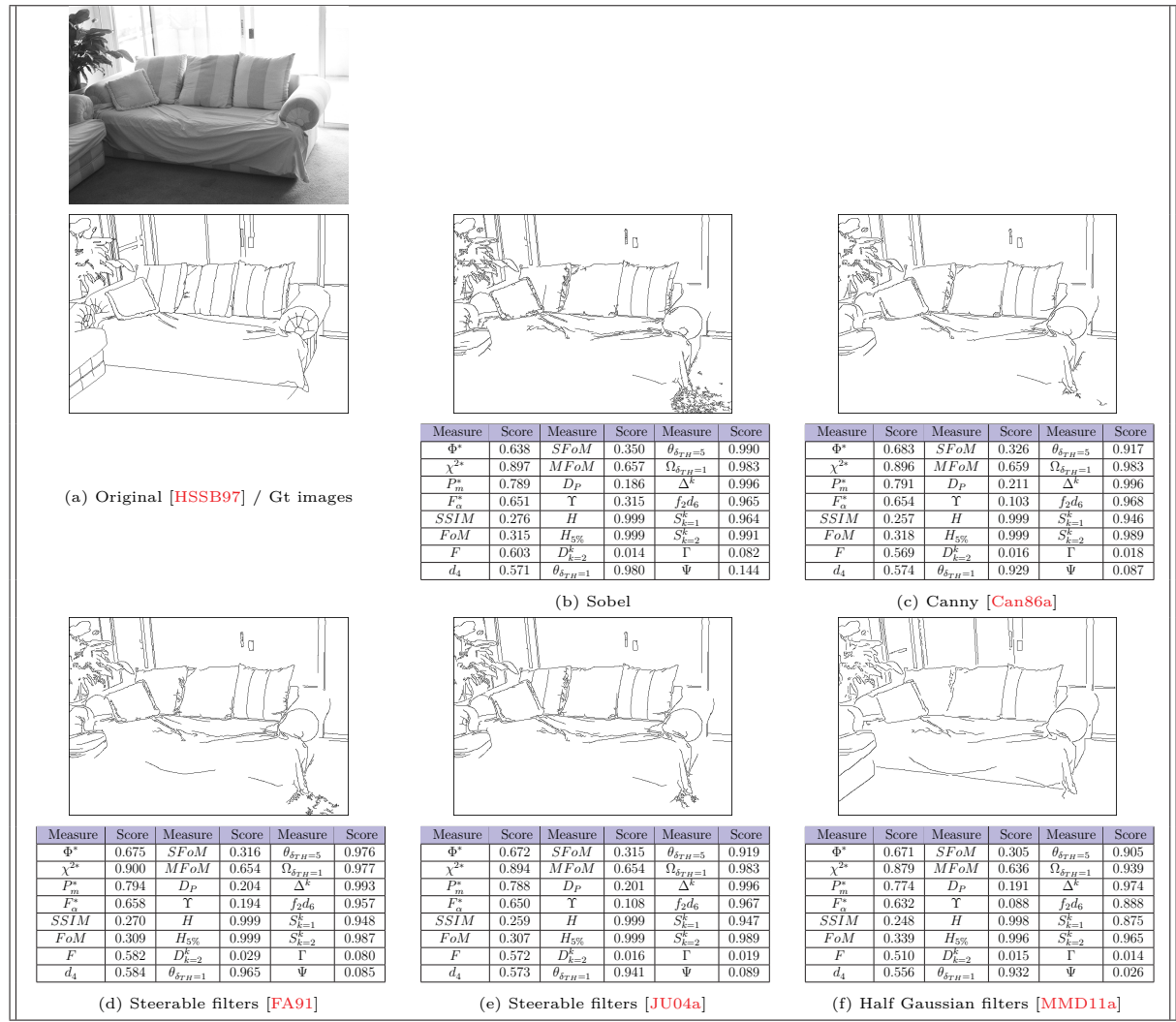


Figure 4.10: Comparison measures of different edge detections involving different edge detectors. All the results are normalized and, compared to the ground truth, a score close to 0 indicates a good edge map whereas a score 1 translates a poor segmentation.

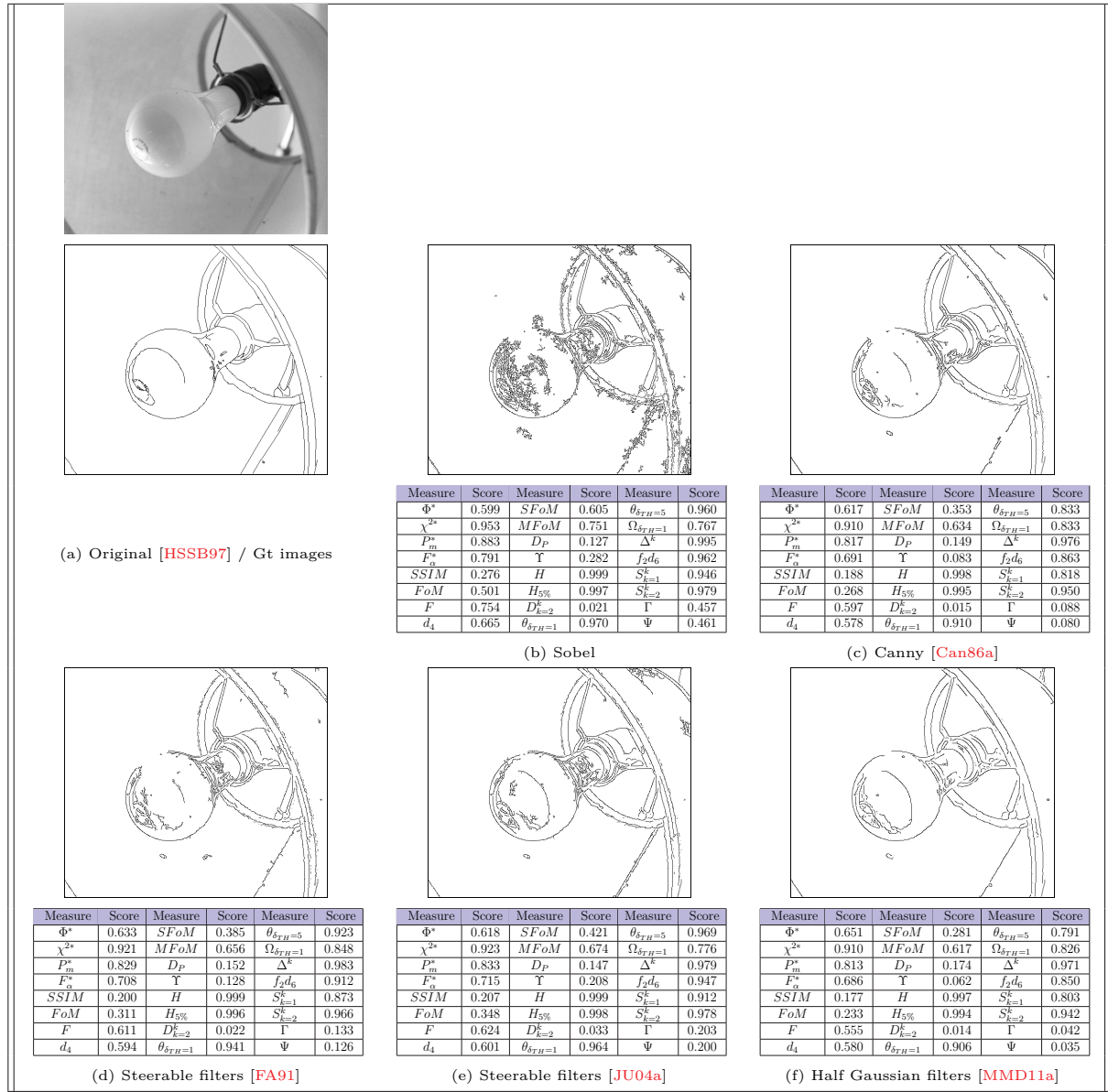


Figure 4.11: Comparison measures of different edge detections involving different edge detectors. All the results are normalized and, compared to the ground truth, a score close to 0 indicates a good edge map whereas a score 1 translates a poor segmentation. Our proposed measure Ψ indicate a good measurement value.

4.5 Summary

In this chapter, several referenced-based boundary detection evaluations are detailed, pointing their advantages and disadvantages through concrete examples of edge images. A new normalized supervised edge map quality measure is proposed, comparing a ground truth contour image, the candidate contour image and their associated spacial nearness. This methodology forms our contribution and resulted in the publication [AMM17d]. The strategy to normalize the evaluation enables to consider a score close to 0 as a good edge map, whereas a score 1 translates a poor segmentation. Eventually, compared to other edge evaluation assessments, the score of the new evaluation indicates confidently the quality of a segmentation.

5

FROM CONTOURS TO GROUND TRUTH: HOW TO EVALUATE EDGE DETECTORS

This chapter focus on several edge dissimilarity measures and present how to evaluate filtering edge detection technique involving these considerate measures.

Contents

5.1	Introduction	64
5.2	Supervised error measures	65
5.3	How to create precise ground truth images ?	70
5.3.1	Ground truth images	72
5.3.2	Minimum of the measure	74
5.4	Experimental results	75
5.4.1	Synthetic images	75
5.4.2	Real images	84
5.5	Summary	98

5.1 Introduction

Over the last decades, edge detection remains a crucial role in the computer vision community [ZT98a]. This segmentation is considered as a fundamental step in many image processing applications or analysis, pattern recognition, as well as in human vision. Moreover, contours include the most important structures in the image. Typically, edges occur on the boundary between two different regions in an image. In other words, an edge is the boundary between an object and the background or between two different objects. There exist many different edge detection methods. Nevertheless, an important problem in image processing remains an efficient edge detector comparison and which parameter(s) correspond(s) to the best setting to obtain an accurate edge detection results. Indeed, a robust boundary detection method should create a contour image containing edges at their correct locations with a minimum of misclassified pixels. In order to objectively quantify the performance of an edge detector, a supervised measure computes a similarity/dissimilarity between a segmentation result and a ground truth obtained from synthetic data or a human judgment [MFTM01]. In this chapter, we detail several edge dissimilarity measures and present how to evaluate filtering edge detection technique involving these considerate measures. In a second time, we demonstrated how to build a new ground truth database which can be used in supervised contour detection evaluation. Indeed, results presented show the importance of the choice of the ground truth. Finally, considering these new ground truth images, results obtained by the measures are exposed.

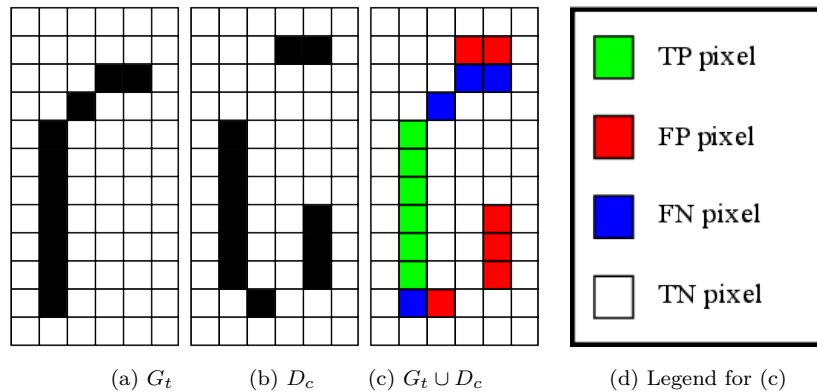


Figure 5.1: Ground truth vs. desired contour. In (b), D_c is contaminated with 6 FPs and 4 FNs.

5.2 Supervised error measures

To assess an edge detector, the confusion matrix remains a cornerstone in boundary detection evaluation methods. Let G_t be the reference contour map corresponding to ground truth and D_c the detected contour map of an original image I . Comparing pixel per pixel G_t and D_c , the first criterion to be assessed is the common presence of edge/non-edge points, as illustrated in Fig. 5.1. A basic evaluation is compounded from statistics; to that effect, G_t and D_c are combined. Afterwards, denoting $|\cdot|$ as the cardinality of a set, all points are divided into four sets: TPs, FPs, FNs and TNs (as described in previous chapter 4). Computing only FPs and FNs enables a segmentation assessment to be performed [MMD11a]. The complemented *Performance measure* P_m^* presented in Table 5.1 considers directly and simultaneously the three entities TP , FP and FN to assess a binary image [GPW03]. The measure is normalized and decreases with improved quality of detection, with $P_m^* = 0$ qualifying perfect segmentation.

By combining FP , FN , TP and TN , another way to display evaluations is to create Receiver Operating Characteristic (ROC) [BKD01] curves or Precision-Recall (PR) [MFM04], involving *True Positive Rates* (TPR) and *False Positive Rates* (FPR): $TPR = \frac{TP}{TP+FN}$ and $FPR = \frac{FP}{FP+TN}$.

Derived from TPR and FPR , the three measures Φ , χ^2 and F_α (detailed in Table 5.1) are frequently used in edge detection assessment. Using the complement of these measures, a score close to 1 indicates a poor segmentation, whereas a value close to 0 a good segmentation. Among these three measures, F_α remains the most stable because it does not consider the TNs, which are dominant in edge maps. Indeed, taking into consideration TN in Φ and χ^2 influences solely the measurement (as is the case in huge images).

Table 5.1: List of error measures involving only statistics.

Complemented <i>Performance measure</i> [GPW03]
$P_m^*(G_t, D_c) = 1 - \frac{TP}{TP + FP + FN}$
Complemented Φ measure [VR95]
$\Phi^*(G_t, D_c) = 1 - \frac{TPR \cdot TN}{TN + FP}$
Complemented χ^2 measure [YP03]
$\chi^{2*}(G_t, D_c) = 1 - \frac{TPR - TP - FP}{1 - TP - FP} \cdot \frac{TP + FP + FPR}{TP + FP}$
Complemented F_α measure [MFM04]
$F_\alpha^*(G_t, D_c) = 1 - \frac{PREC \cdot TPR}{\alpha \cdot TPR + (1 - \alpha) \cdot PREC},$
with $PREC = \frac{TP}{TP + FP}$ and $\alpha \in]0; 1]$

Table 5.2: List of normalized error measures compared in this work, with the parameter $\kappa \in]0; 1]$.

Figure of Merit (<i>FoM</i>) [AP79]
$FoM(G_t, D_c) = 1 - \frac{1}{\max(G_t , D_c)} \cdot \sum_{p \in D_c} \frac{1}{1 + \kappa \cdot d_{G_t}^2(p)}$
<i>FoM</i> of over-segmentation [SG91]
$FoM_e(G_t, D_c) = 1 - \frac{1}{\max(e^{-FP}, FP)} \cdot \sum_{p \in D_c \cap \neg G_t} \frac{1}{1 + \kappa \cdot d_{G_t}^2(p)}$
<i>FoM</i> revisited [PA95]
$F(G_t, D_c) = 1 - \frac{1}{ G_t \cup D_c } \cdot \sum_{p \in G_t} \frac{1}{1 + \kappa \cdot d_{D_c}^2(p)}$
Combination of <i>FoM</i> and statistics [BG09]
$d_4(G_t, D_c) = \frac{1}{2} \cdot \sqrt{S + FoM(G_t, D_c)}$ with $S = \frac{(TP - \max(G_t , D_c))^2 + FN^2 + FP^2}{(\max(G_t , D_c))^2}$
Symmetric Figure of Merit [AMM17d]
$SFoM(G_t, D_c) = \frac{1}{2} \cdot FoM(G_t, D_c) + \frac{1}{2} \cdot FoM(D_c, G_t)$
Maximum Figure of Merit [AMM17d]
$MFoM(G_t, D_c) = \max(FoM(G_t, D_c), FoM(D_c, G_t))$
Edge map quality measure [PGAN16]
$D_p(G_t, D_c) = \frac{1/2}{ I - G_t } \cdot L + \frac{1/2}{ G_t } \cdot R$ $L = \sum_{p \in D_c} 1 - \frac{1}{1 + \kappa \cdot d_{G_t}^2(p)} \quad \text{and} \quad R = \sum_{p \in G_t} 1 - \frac{1}{1 + \kappa \cdot d_{G_t \cap D_c}^2(p)}$

These measures evaluate the comparison of two edge images, pixel per pixel, tending to severely penalize a (even slightly) misplaced contour, as illustrated in Fig. 5.2 (g) and (h). Thus, to perform an edge evaluation, the assessment should penalize a misplaced edge point proportionally to the distance from its true location. A reference-based edge map quality measure requires that a displaced edge should be penalized in function not only of FPs and/or FNs but also of the distance from the position where it should be located. Tables 5.2 and 5.3 review the most relevant measures in the literature. The common feature between these evaluators corresponds to the error distance $d_{G_t}(p)$ or/and $d_{D_c}(p)$. Indeed, for a pixel belonging to the desired contour $p \in D_c$, $d_{G_t}(p)$ represents the minimal euclidian distance between p and G_t . On the contrary, if a pixel p belongs to the ground truth G_t , $d_{D_c}(p)$ is the minimal euclidian distance between p and D_c . On the one hand, some distance measures are specified in the evaluation of over-segmentation (i.e. presence of FPs), like: FoM_e , Υ , D^k , Θ and Γ . On the other hand, Ω measure assesses an edge detection by computing only an under segmentation (i.e. missing ground truth points). Other edge detection evaluation measures consider both FPs and FNs. First, to achieve a quantitative index of edge detector performance, one of the most popular descriptors is the Figure of Merit (*FoM*). This distance measure ranges from 0 to 1, where 0 corresponds to a perfect segmentation [AP79]. Widely utilized for comparing several different segmentation methods, in particular thanks to its normalization criterion, this assessment approach nonetheless suffers from a main drawback. Whenever FNs are

Table 5.3: List of non-normalized error measures. In the literature, the most common values are $k = 1$ or $k = 2$.

Yasnoff measure [YGB78]
$\Upsilon(G_t, D_c) = \frac{100}{ I } \cdot \sqrt{\sum_{p \in D_c} d_{G_t}^2(p)}$
Hausdorff distance [HR93]
$H(G_t, D_c) = \max \left(\max_{p \in D_c} d_{G_t}(p), \max_{p \in G_t} d_{D_c}(p) \right)$
Distance to G_t [PM82] [HR93][DJ94][LMDBB13]
$D^k(G_t, D_c) = \frac{1}{ D_c } \cdot \sqrt[k]{\sum_{p \in D_c} d_{G_t}^k(p)},$ $k \in \mathbb{R}^+, \quad k = 1 \text{ for [PM82]}$
Maximum distance [DJ94]
$f_{2d_6}(G_t, D_c) = \max \left(\frac{1}{ D_c } \cdot \sum_{p \in D_c} d_{G_t}(p), \frac{1}{ G_t } \cdot \sum_{p \in G_t} d_{D_c}(p) \right)$
Oversegmentation [Har84][OBBC02]
$\Theta(G_t, D_c) = \frac{1}{FP} \cdot \sum_{p \in D_c} \left(\frac{d_{G_t}(p)}{\delta_{TH}} \right)^k,$ $k \in \mathbb{R}^+ \text{ and } \delta_{TH} \in \mathbb{R}_*^+ \text{ [OBBC02]}, k = \delta_{TH} = 1 \text{ for [Har84]}$
Undersegmentation [Har84][OBBC02]
$\Omega(G_t, D_c) = \frac{1}{FN} \cdot \sum_{p \in G_t} \left(\frac{d_{D_c}(p)}{\delta_{TH}} \right)^k,$ $k \in \mathbb{R}^+ \text{ and } \delta_{TH} \in \mathbb{R}_*^+ \text{ [OBBC02]}, k = \delta_{TH} = 1 \text{ for [Har84]}$
Baddeley's Delta Metric [Bad92]
$\Delta^k(G_t, D_c) = \sqrt[k]{\frac{1}{ I } \cdot \sum_{p \in I} w(d_{G_t}(p)) - w(d_{D_c}(p)) ^k},$ $k \in \mathbb{R}^+ \text{ and a convex function } w : \mathbb{R} \mapsto \mathbb{R}$
Symmetric distance [DJ94][LMDBB13]
$S^k(G_t, D_c) = \sqrt[k]{\frac{\sum_{p \in D_c} d_{G_t}^k(p) + \sum_{p \in G_t} d_{D_c}^k(p)}{ D_c \cup G_t }},$ $k \in \mathbb{R}^+, \quad k = 1 \text{ for [DJ94]}$
Magnier <i>et al.</i> measure [MLZ16]
$\Gamma(G_t, D_c) = \frac{FP + FN}{ G_t ^2} \cdot \sqrt{\sum_{p \in D_c} d_{G_t}^2(p)}$
Symmetric distance measure [AMM17d]
$\Psi(G_t, D_c) = \frac{FP + FN}{ G_t ^2} \cdot \sqrt{\sum_{p \in G_t} d_{D_c}^2(p) + \sum_{p \in D_c} d_{G_t}^2(p)}$

created, the distance of FNs ($d_{D_c}(p)$) are not recorded. Indeed, FoM can be rewritten as:

$$\begin{aligned}
FoM(G_t, D_c) &= 1 - \frac{\sum_{p \in D_c \cap G_t} \frac{1}{1 + \kappa \cdot d_{G_t}^2(p)} + \sum_{p \in D_c \cap \neg G_t} \frac{1}{1 + \kappa \cdot d_{G_t}^2(p)}}{\max(|G_t|, |D_c|)} \\
&= 1 - \frac{TP + \sum_{p \in D_c \cap \neg G_t} \frac{1}{1 + \kappa \cdot d_{G_t}^2(p)}}{\max(|G_t|, |D_c|)},
\end{aligned} \tag{5.1}$$

because, for $p \in D_c \cap G_t$, $d_{G_t}^2(p) = 0$ and $\frac{1}{1+\kappa \cdot d_{G_t}^2(p)} = 1$. Knowing that $TP = |G_t| - FN$, for the extreme cases, the FoM measures takes the following values:

$$\begin{cases} \text{if } \hat{E}FP = 0 : FoM(G_t, D_c) = 1 - \frac{TP}{|G_t|}, \\ \text{if } \hat{E}\hat{E}FN = 0 : FoM(G_t, D_c) = 1 - \frac{1}{\max(|G_t|, |D_c|)} \cdot \sum_{p \in D_c \cap \neg G_t} \frac{1}{1 + \kappa \cdot d_{G_t}^2(p)}. \end{cases} \quad (5.2)$$

When $FP = 0$, FoM behaves like matrix-based error assessments. Moreover, for $FP > 0$, as $\frac{1}{1+\kappa \cdot d_{G_t}^2(p)} < 1$, the FoM measure penalizes the over-detection very low compared to the under-detection. The curve in Fig. 5.2 shows that the penalization of missing points (FNs) becomes higher whereas it is weaker concerning FP . On the contrary, the F measure computes the distances of FNs:

$$F(G_t, D_c) = 1 - \frac{TP + \sum_{p \in \neg D_c \cap G_t} \frac{1}{1 + \kappa \cdot d_{D_c}^2(p)}}{|G_t \cup D_c|}. \quad (5.3)$$

F behaves inversely to FoM :

$$\begin{cases} \text{if } \hat{E}FP = 0 : F(G_t, D_c) = 1 - \frac{\hat{E}|D_c| + \sum_{p \in \neg D_c \cap G_t} \frac{1}{1 + \kappa \cdot d_{D_c}^2(p)}}{|G_t|}, \\ \text{if } \hat{E}\hat{E}FN = 0 : F(G_t, D_c) = 1 - \frac{|G_t|}{|D_c|}. \end{cases} \quad (5.4)$$

Also, d_4 r measure depends particularly on TP , FP , FN and FoM . Nonetheless, this measure penalizes FNs like the FoM measure, as shown in Fig. 5.2 (j). $SFoM$ and $MFoM$ take into account both distances of FNs and FPs, so they can compute a global evaluation of a contour image, but as illustrated in Figs. 5.2 (i) and (j), $MFoM$ does not considers FPs and FNs at the same time, contrary to $SFoM$. Another way to compute a global measure is presented in [PGAN16] with the edge map quality measure D_p . The over-segmentation measure (left term) evaluates d_{D_c} , the distances between the FPs and G_t . The under-segmentation measure (right term) computes the distances of the FNs between the closest correctly detected edge pixel, i.e. $G_t \cap D_c$. That means that FNs and their distances are not counted without the presence of TP(s), and D_p is more sensitive to FNs than FPs, see Figs. 5.2 (i) and (j).

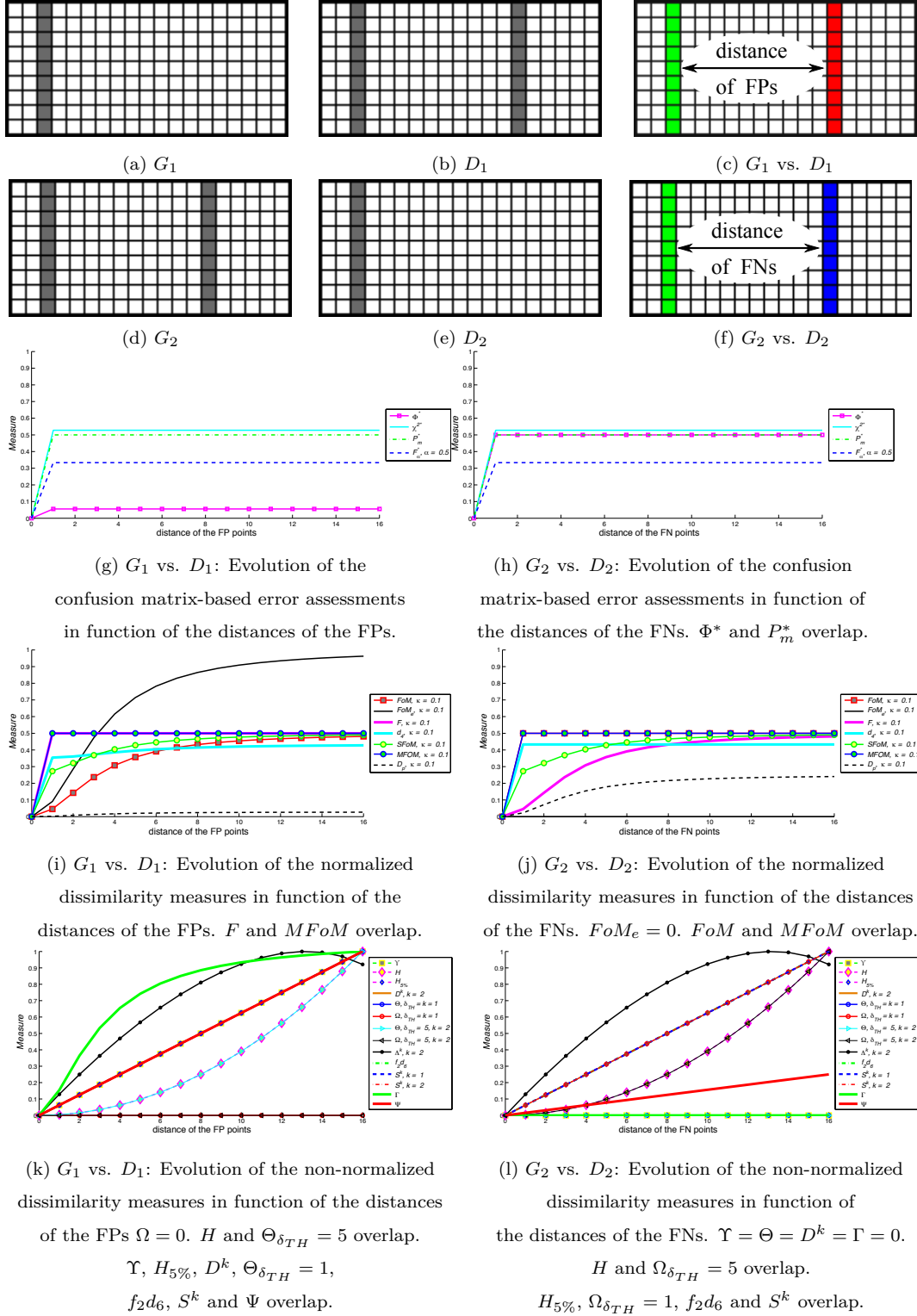


Figure 5.2: Evolution of dissimilarity measures in function of the the distance of the false positive/negative points. A vertical line of false positive points (b) or false negative points (d) is shifted by a maximum distance of 16 pixels and the measure scores are plotted in function of the displacement of the desired/undesired contour.

A second measure widely computed in matching techniques is represented by the Hausdorff distance H , which measures the mismatch of two sets of points [HR93]. This max-min distance could be strongly deviated by only one pixel which can be positioned sufficiently far from the pattern. To improve the measure, one idea is to compute H with a proportion of the maximum distances (for example 5% of the values [HR93]); let us note $H_{5\%}$ this measure. Nevertheless, as pointed out in [DJ94], an average distance from the edge pixels in the candidate image to those in the ground truth is more appropriate for matching purposes than H and $H_{n\%}$. To achieve this task, D^k , Υ , Θ and Γ which represent errors of distance only in function of d_{G_t} , they correspond to a measure of over-segmentation (only FPs), as indicated by the curves in Figs. 5.2 (l) where the curves stagnate at 0. On the contrary, the sole use of a distance d_{D_c} instead of d_{G_t} enables an estimation of the FN divergences, representing an under-segmentation (as in Ω). Nevertheless, as concluded in [CLRE08], a complete and optimum edge detection evaluation measure should combine assessments of both over- and under-segmentation, as f_2d_6 , S^k and Ψ . Also, combining both d_{D_c} and d_{G_t} , Baddeley's Delta Metric (Δ^k) [Bad92] is a measure derived from the Hausdorff distance which is intended to estimate the dissimilarity between each element of two binary images. Finally, curves in Figs. 5.2 (k) and (l) illustrate that H , $H_{5\%}$, Δ^k , f_2d_6 and S^k behave similarly in function of the FPs or FNs distances. Note that the Ψ measure is more sensitive to the distance of the FPs. The scores of the non-normalized measures in Figs. 5.2 (k) and (l) are normalized using the following equation for easy visual comparison. Denoting by $f \in [0; +\infty[$ the score vectors of a distance measure such that:

$$\begin{cases} m &= \min(\min(f(G_1, D_1)), \min(f(G_2, D_2))), \\ M &= \max(\max(f(G_1, D_1)), \max(f(G_2, D_2))); \end{cases}$$

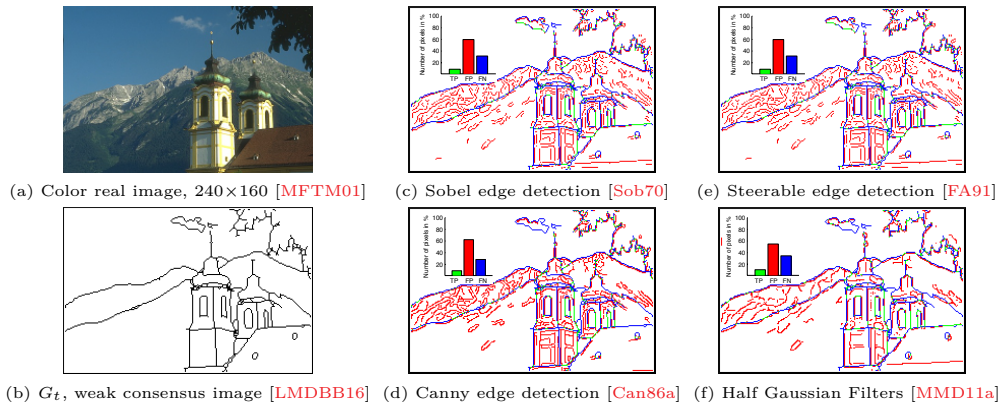
then the normalization \mathcal{N} of a measure is computed by:

$$\mathcal{N}(f) = \begin{cases} 0 & \text{if } M = m = 0 \\ 1 & \text{if } M = m \neq 0 \\ \frac{f - m}{M - m} & \text{if } M > 1 \text{ and } m \neq 0 \\ f & \text{otherwise.} \end{cases} \quad (5.5)$$

5.3 How to create precise ground truth images ?

An edge detector is considered as robust when the evaluation score of the dissimilarity with a given G_t is close to 0. Table in Fig.5.3 reports different assessments for four edge detection methods on a real image (color): Sobel [Sob70], Canny [Can86a], Steerable Filters (S-F) [FA91] and Half Gaussian Kernels (H-K) [MMD11a]. Only the comparison of D_c with a G_t is studied here. Segmentations are classified together by comparing the scores of the dissimilarity measures and the smallest score for a given measure indicates the

best segmentation. Indeed, for example, Sobel corresponds to the best edge detector for Υ , Canny for Δ^k , S-F for Ω and H-F for FoM . However, this assessment suffers from two main drawbacks. Firstly, segmentations are compared using the threshold (voluntary) chosen by the user, this evaluation is very subjective and not reproducible [AMM17d]. Secondly, some deficiencies appear in real ground truth contour maps, which could disturb the evaluation of efficient segmentation methods, or, on the contrary, advantage weak/biased edge detectors. Thus, according to the used measure or threshold any detector is classified the first one or the last one.



Measure	Sobel	Canny [Can86b]	S-F [FA91]	H-K [MMD11a]
P_m^*	0.908	0.908	0.988	0.893
Φ^*	0.766	0.777	0.799	0.779
χ^{2*}	0.987	0.986	0.988	0.979
F_α^*	0.831	0.831	0.838	0.807
FoM	0.399	0.381	0.408	0.344
F	0.705	0.679	0.649	0.6148
Υ	1.007	1.041	1.058	1.025
$\Theta_{\delta_{TH}=1}$	4.323	4.369	4.852	4.577
$\Omega_{\delta_{TH}=1}$	2.783	1.721	1.410	1.640
$H_{5\%}$	19.76	18.45	19.56	20.89
Δ^k	9.454	7.019	8.517	11.13
$S_{k=2}^k$	5.799	5.421	5.626	5.776
Γ	0.499	0.486	0.465	0.393
Ψ	0.584	0.499	0.471	0.402

Figure 5.3: Edge detection after the non-maximum suppression [RT71] and comparison with a ground truth image (green cells for the minimum measure and better edge detector filter).

5.3.1 Ground truth images

In edge detection assessment, the ground truth is considered as a perfect segmentation. The most common method for ground-truth definition in natural images remains manual labeling by humans [MFTM01] [HSSB97]. These data sets are not optimal in the context of the definition of low-level segmentation. Firstly, labelers have marked mainly edges of salient objects, whereas equally strong edges in the background or around less important objects are missing. Moreover, errors may be created by human labels (oversights or supplements); indeed, an inaccurate ground truth contour map in terms of localization penalizes precise edge detectors and/or advantages the rough algorithms. Finally, in [HYK13], the question is raised concerning the reliability of the datasets regarded as ground truths for novel edge detection methods. Thus, an incomplete ground truth penalizes an algorithm detecting true boundaries and efficient edge detection algorithms obtain between 30% and 40% of errors. Furthermore, when G_t maps are built from a consensus which consists in the combination of several human-labelled images [FGCPMCMC08][CLRE08][LMDBB16], the deficiencies recalled above remain present. These reasons accentuate the importance of the relevant development of the ground truth labeling.

In real digital images, various profile edge types determine contours such as: step, ramp, roof of peaks. Pure step edges are seldom present in real image scenes, but they can be created in synthetic data. For a 2D signal, pixels of contours are measured having the higher slope and are localized in the perpendicular direction of the slope of the image function. Considering synthetic data, true edges are positioned between two different colors/gray levels. Nevertheless, the edge position of an object could be interpreted in different ways: for a vertical step edge, an edge can be located either on the left, or on the right. In Fig. 5.4, several white shapes are immersed in a black background, creating step edges. To avoid the problem of edge pixel placements, a blur must be voluntary created by adding a 1 pixel width of gray around each shape. Thus, the ground truth corresponds to the points where the slope of the image surface is maximum, i.e. to this

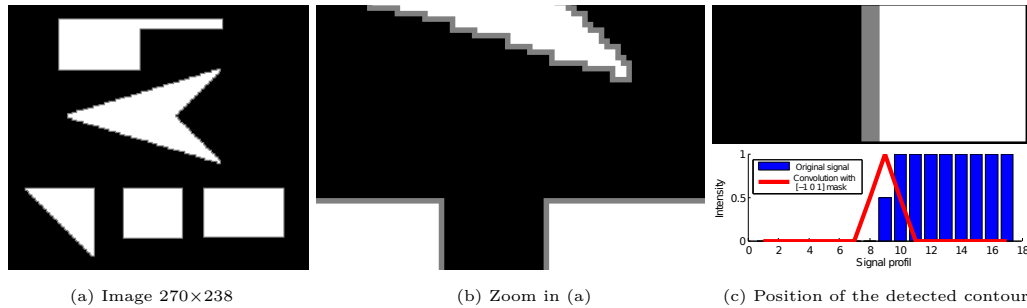


Figure 5.4: Synthetic data with a 1 pixel width gray around each shape: value of white pixels = 1, values of black pixels = 0, values of gray pixels = 0.5.

gray. These points could be extracted involving odd filters (derivative filters of order 1). In the one hand, a $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ mask allows to extract the edges at the correct position, i.e. the gray pixels in Fig. 5.4, contrary to edge detector involving smoothing parameters which delocalize edge positions (especially corners and small objects [ZT98a]). The new database of contour images issued of real images takes into account all these properties. This section presents ground truth edge maps which are labeled in a semi-automatic way in order to evaluate the performance of filtering step/ramp edge detectors. Therefore, the motivations to create new ground truth edge images are:

1. To obtain contours accurately localized,
2. To extract edges of the secondary objects or in the background,
3. To exclude boundaries inside noisy/textured regions.

In fact, this new label processes in return to hand made ground truth. Indeed, in a first time, the contours are detected involving the convolution of the image with $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ vertical and horizontal masks followed by a computation of a gradient magnitude and a suppression of local non-maxima in the gradient direction [RT71]. Concerning color images, $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ vertical and horizontal masks are applied to each channel of the image followed by a structure tensor [DZ86]. In a second time, undesirable edges are deleted while missing points are added both by hand.

Fig. 5.5 illustrates the steps to obtain new ground truth images. Using the $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ mask enables to capture the majority of edge points and corners without deforming small objects, contrary to edge detectors involving Gaussian filters (see for example Fig. 6 in

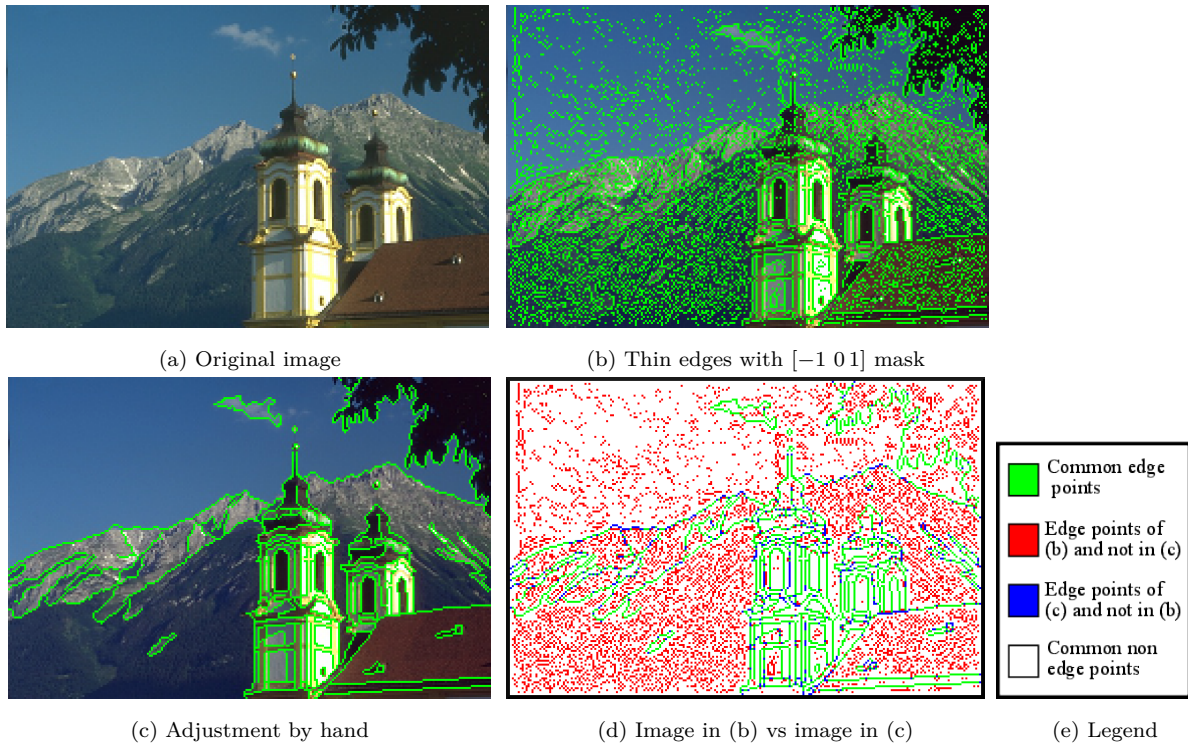


Figure 5.5: Image of our database are built after an edge detection involving a $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ mask and concluded by hand.

[PM90]). Moreover, this process enables to detect the good positions of the contours while avoiding the addition of too much imprecise ground truth points, as shown in Fig. 5.4.

5.3.2 Minimum of the measure

Instead of thresholding manually or automatically [Ots75][Ros01] and then comparing the segmentation of several edge detectors, as in Fig. 5.7 (c) and (d), the dissimilarity measures are used for an objective assessment. Indeed, the purpose is to compute the minimal value of a dissimilarity measure by varying the threshold Th of the thin edges computed by an edge detector (thin edges are created after the non-maximum suppression of the absolute gradient [RT71]). Indeed, compared to a ground truth contour map, the ideal edge map for a measure corresponds to the desired contour at which the evaluation obtains the minimum score for the considered measure among the thresholded gradient images. Theoretically, this score corresponds to the threshold at which the edge detection represents the best edge map, compared to the ground truth contour map [FGMCCP⁺04][CLRE08]. Fig. 5.6 illustrates all this process. Since a small threshold leads to heavy over-segmentation and a strong threshold may create numerous false negative pixels, the minimum score of an edge detection evaluation should be a compromise between under- and over-segmentation. As illustrated in Fig. 5.7 (e) the best score for the under-segmentation evaluation corresponds to $Th = 0$, because false negative points penalize the Ω measure. On the contrary, false positive points penalize over-segmentation dissimilarity measures, as $FoMe$, Υ , D^k , Θ and Γ measures, see Fig. 5.7 (g). Consequently, the best score concerning an over-segmentation measure corresponds to $Th \approx 1$. As G_t are not the same for the evaluation in Fig. 5.7 (g) and (h), the two curves are different.

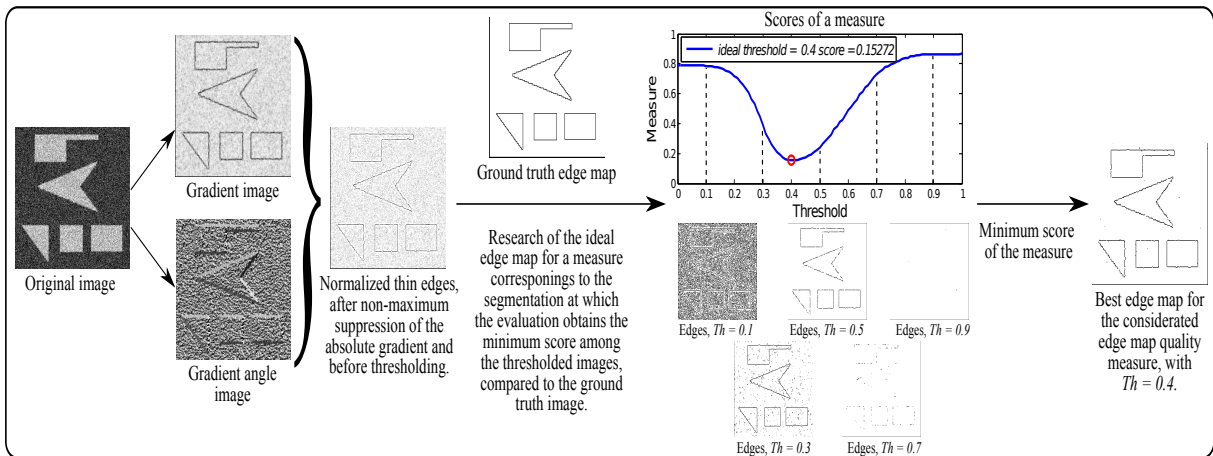


Figure 5.6: The most relevant edge map for a dissimilarity measure is indicated by its minimum score.

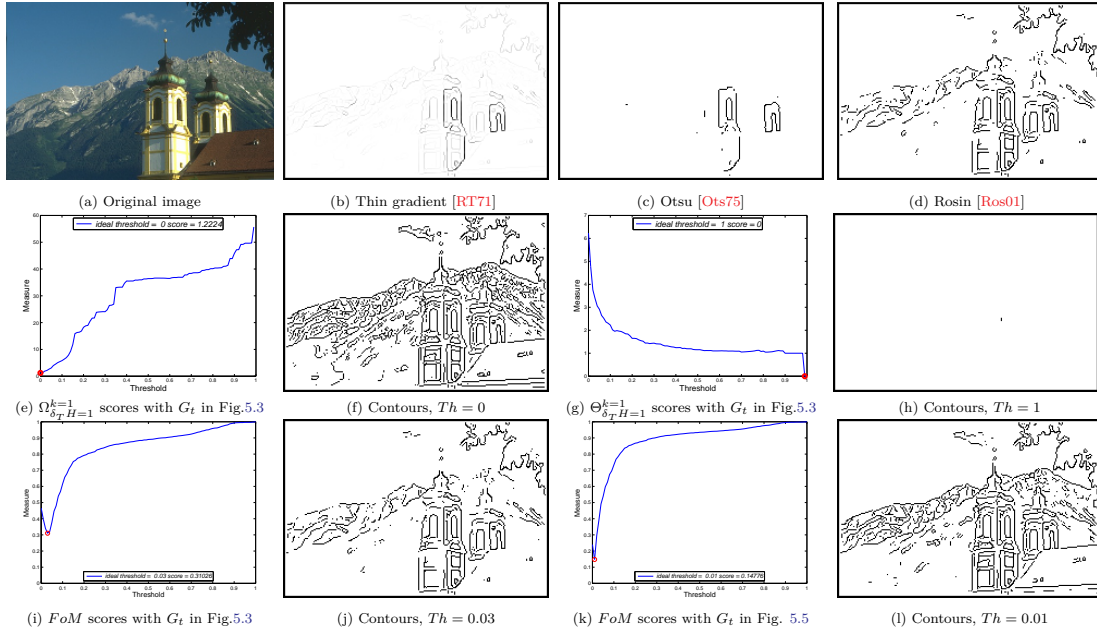


Figure 5.7: Scores of the measures depending on the threshold of the thin gradient image [Can86a].

5.4 Experimental results

The purpose of the experiments presented here is to obtain the best edge map in a supervised way. In order to study the performance of the contour detection evaluation measures, each measure is compared by varying the threshold of the thin edges computed by until six edge detectors: Sobel [Sob70], Canny [Can86a], Steerable Filters of order 1 (SF_1) [FA91], Steerable Filters of order 5 (SF_5) [JU04a], Anisotropic Gaussian Kernels (AGK) [GSvdW02] and Half Gaussian Kernels (HK) [MMD11a]. In one hand, experiments are led on two synthetic noisy images. In other hand, contour detection evaluations are compared on seven real images where G_t edge maps are labelled by a semi-automatic way (section 5.3.1). Finally, compared to a ground truth contour map, the ideal edge map for a measure corresponds to the desired contour at which the evaluation obtains the minimum score for the considered measure among the thresholded gradient images.

5.4.1 Synthetic images

To evaluate the performances of the dissimilarity measures, the original image in Fig. 5.4(a) is disturbed with random Gaussian noise and edges are extracted from the noisy images (4 dB and 3.3 dB):

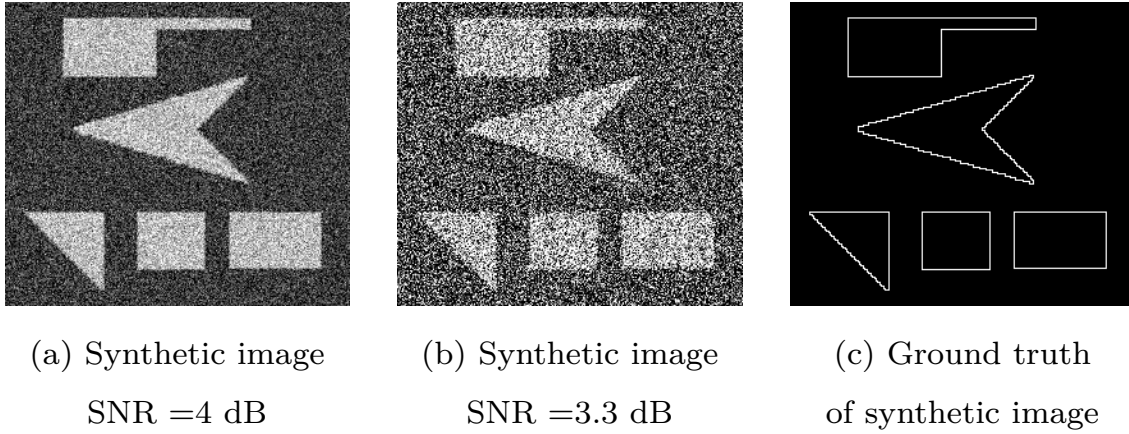


Figure 5.8: Synthetic images corrupted by a Gaussian noise with the associated ground truth edge map.

- The minimum score for several dissimilarity measures and several edge detection methods for a noisy synthetic image (SNR= 4 dB) are presented for our proposed measure Ψ [AMM17d] in Fig. 5.10 and in Fig. 5.12 and Fig. 5.13 for all the state-of-art measures methods. The original image is available in Fig. 5.8 (a) and the ground truth as shown in 5.8(c).
- The minimum score for several dissimilarity measures and several edge detection methods for a noisy synthetic image (SNR= 3.3 dB) are presented for our proposed measure Ψ [AMM17d] in Fig. 5.11 and in Fig. 5.14 and Fig. 5.15. The original image is available in Fig. 5.8 (b) and the ground truth as shown in Fig. 5.8(c).

Generally, the scores of Φ^* , d_4 and D_p measures allow to correctly extract the edges at the price of numerous FPs. Moreover, Δ^k is more sensitive to FPs than the other dissimilarity measures and the best score corresponds to a contour edge map with many discontinuous contours. As pointed out in section 5.2, concerning the image corrupted by a noise at a level of 4dB, FoM penalizes strongly FNs to the detriment of FPs apparitions, and it considers that anisotropic edge detectors are less performant than the Canny edge detector [Can86b]. Other measures classify the Sobel method [Sob70] as the less efficient one and the H-K as the best one.

Images of our database are built after an edge detection involving $[-1 \ 0 \ 1]$ masks and concluded by hand. Fig. 5.9 presents some examples of ground truth contour images.

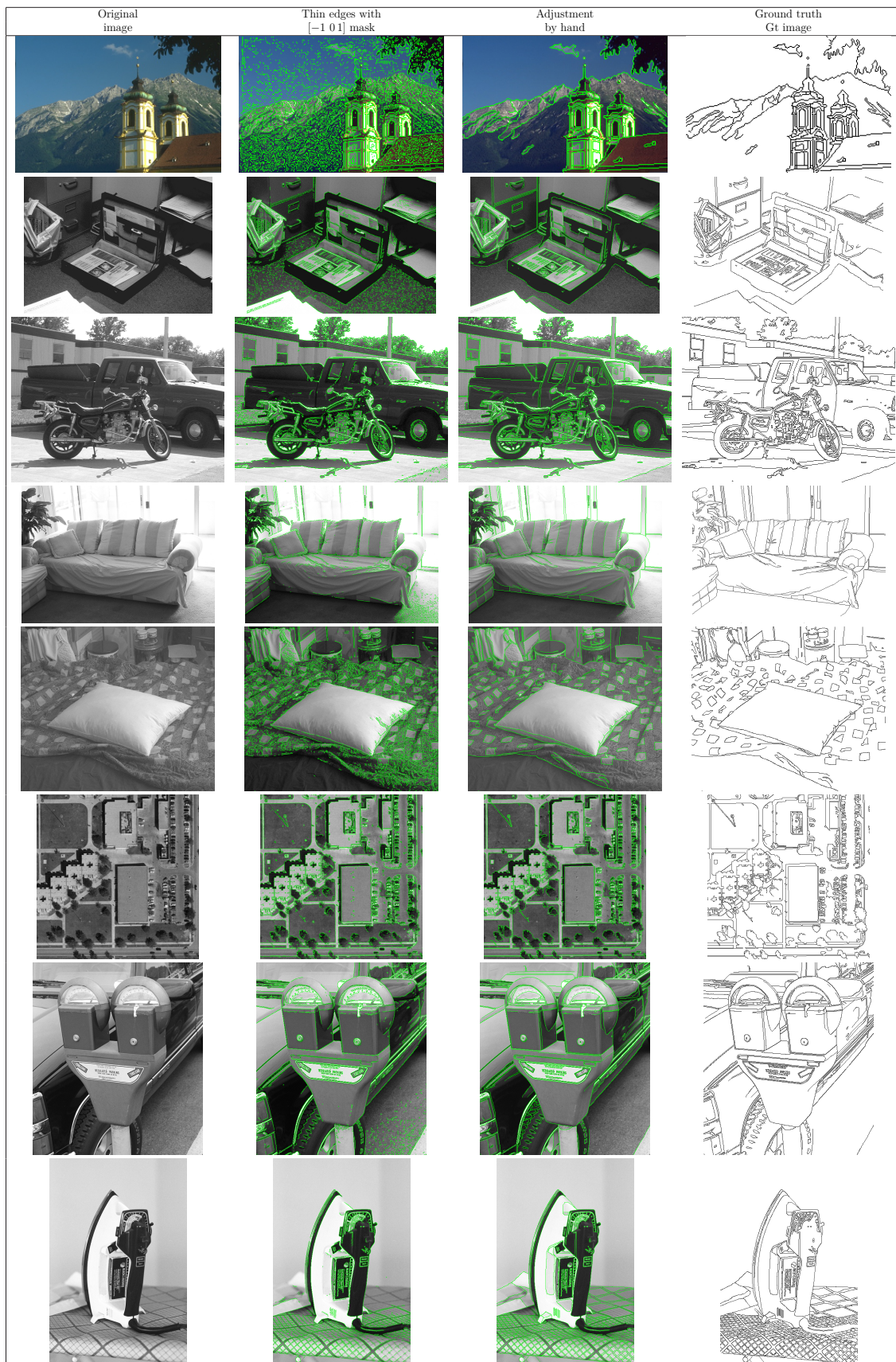


Figure 5.9: Images of our database are built in a semi-automatic way, after an edge detection involving $[-1\ 0\ 1]$ masks and concluded by hand.

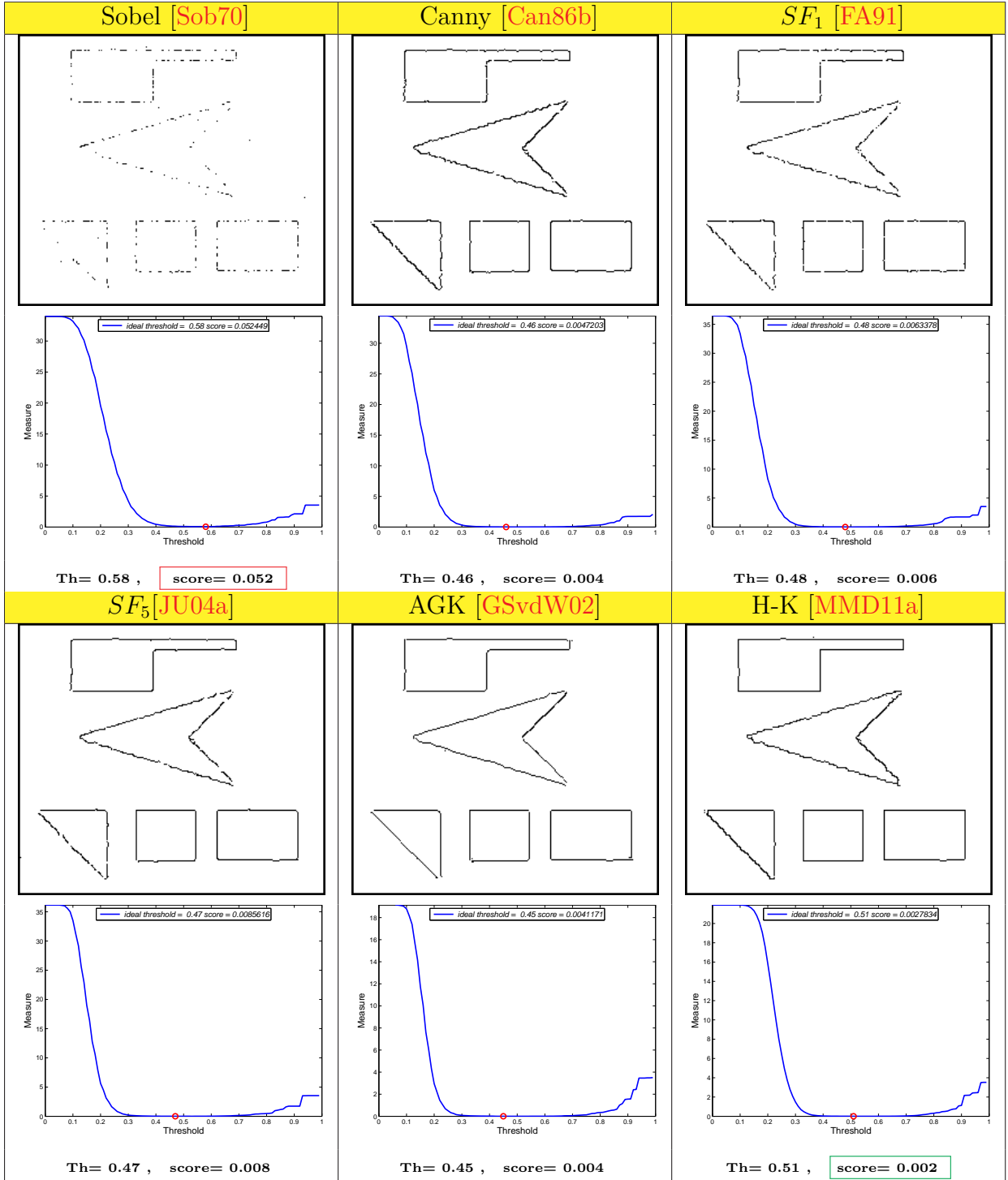


Figure 5.10: Synthetic image SNR= 4 dB. (The original image is available in Fig. 5.8 (a) and the ground truth as shown in 5.8(c)). Best maps for the compared edge detection evaluations using our proposed measure Ψ .

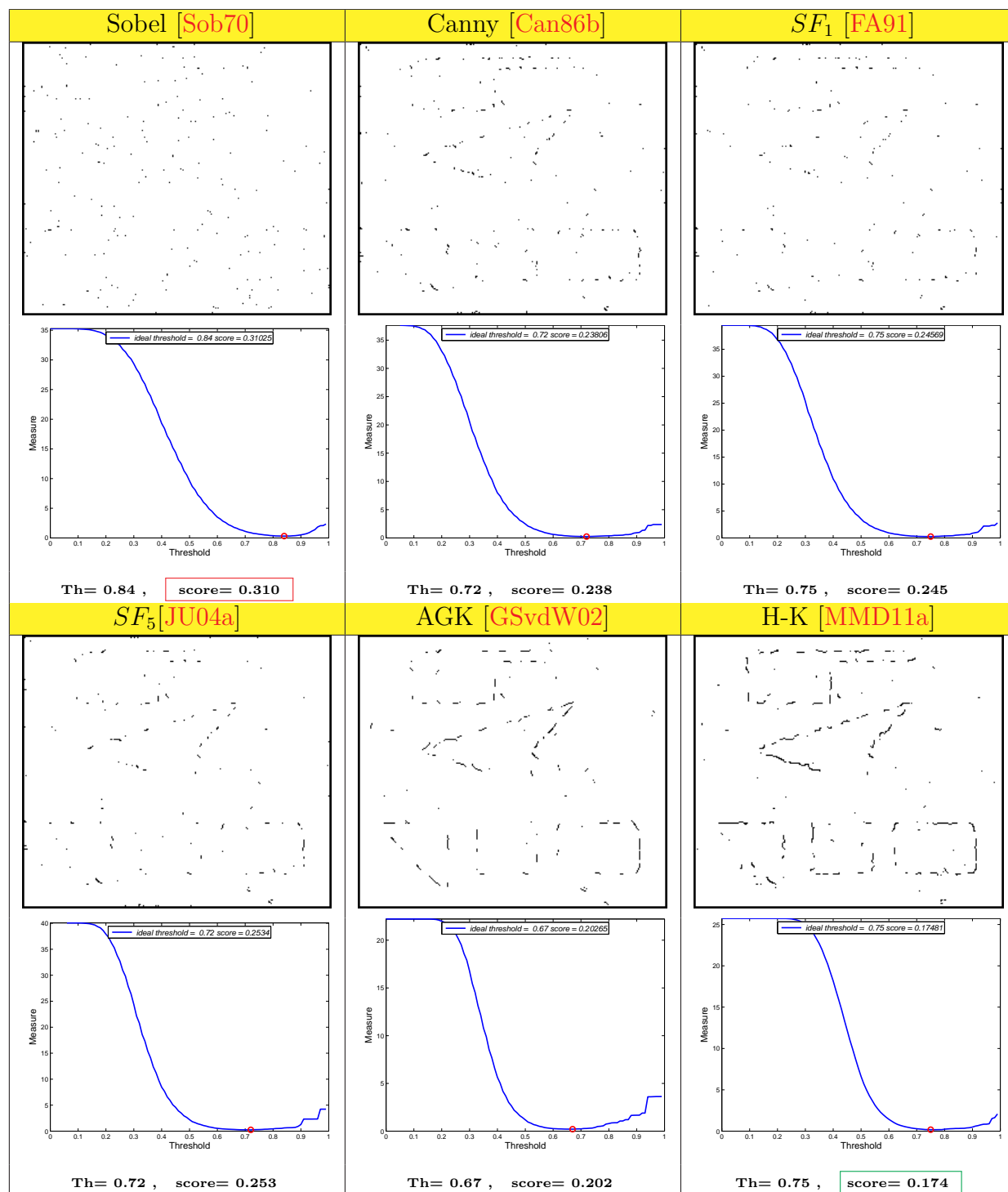


Figure 5.11: Synthetic image SNR= 3.3 dB (The original image is available in Fig. 5.8 (b) and the ground truth as shown in 5.8(c)). Best maps for the compared edge detection evaluations using our proposed measure Ψ .

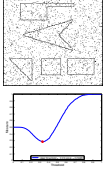
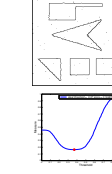
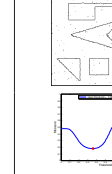
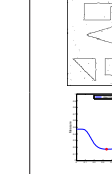
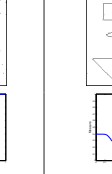
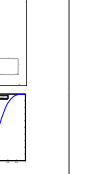
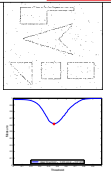
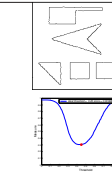
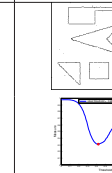
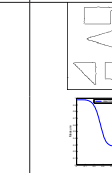
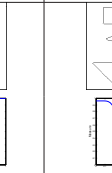
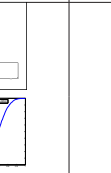
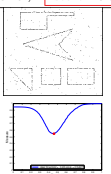
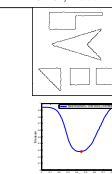
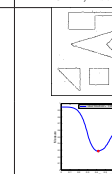
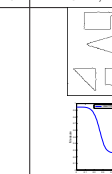
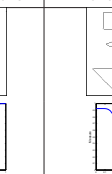
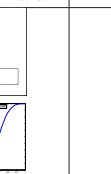
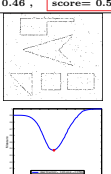
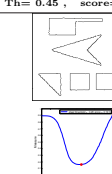
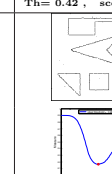
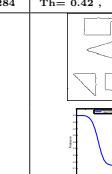

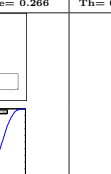
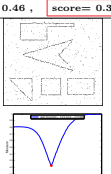
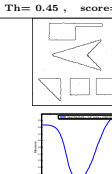




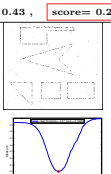
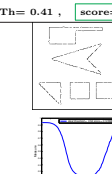

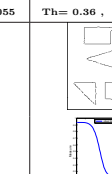
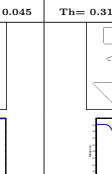
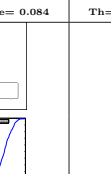
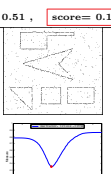


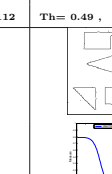
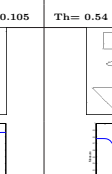
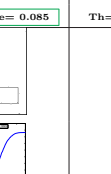
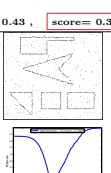

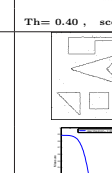
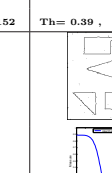
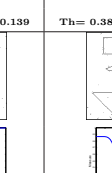

Measure	Sobel [Sob70]	Canny [Can86b]	SF_1 [FA91]	SF_5 [JU04a]	AGK [GSvdW02]	H-K [MMD11a]
Φ^*	 Th= 0.33 , score= 0.284	 Th= 0.37 , score= 0.164	 Th= 0.36 , score= 0.180	 Th= 0.34 , score= 0.170	 Th= 0.38 , score= 0.207	 Th= 0.51 , score= 0.134
χ^{2*}	 Th= 0.46 , score= 0.614	 Th= 0.45 , score= 0.303	 Th= 0.42 , score= 0.310	 Th= 0.42 , score= 0.284	 Th= 0.45 , score= 0.285	 Th= 0.51 , score= 0.220
P_m^*	 Th= 0.46 , score= 0.546	 Th= 0.45 , score= 0.278	 Th= 0.42 , score= 0.284	 Th= 0.42 , score= 0.261	 Th= 0.45 , score= 0.266	 Th= 0.51 , score= 0.205
F_α^*	 Th= 0.46 , score= 0.376	 Th= 0.45 , score= 0.161	 Th= 0.42 , score= 0.166	 Th= 0.42 , score= 0.150	 Th= 0.45 , score= 0.153	 Th= 0.51 , score= 0.114
FoM	 Th= 0.43 , score= 0.224	 Th= 0.41 , score= 0.022	 Th= 0.38 , score= 0.055	 Th= 0.36 , score= 0.045	 Th= 0.31 , score= 0.084	 Th= 0.41 , score= 0.038
F	 Th= 0.51 , score= 0.199	 Th= 0.54 , score= 0.125	 Th= 0.49 , score= 0.112	 Th= 0.49 , score= 0.105	 Th= 0.54 , score= 0.085	 Th= 0.56 , score= 0.093
d_4	 Th= 0.43 , score= 0.351	 Th= 0.42 , score= 0.141	 Th= 0.40 , score= 0.152	 Th= 0.39 , score= 0.139	 Th= 0.38 , score= 0.165	 Th= 0.51 , score= 0.107
$SFoM$	 Th= 0.43 , score= 0.148	 Th= 0.41 , score= 0.019	 Th= 0.39 , score= 0.037	 Th= 0.36 , score= 0.030	 Th= 0.31 , score= 0.052	 Th= 0.41 , score= 0.025

Figure 5.12: Synthetic image SNR= 4 dB: Best maps for the compared edge detection evaluations.

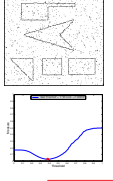
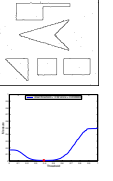


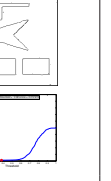

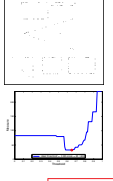
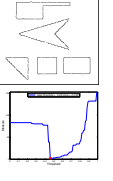




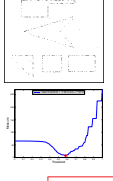
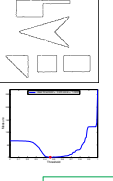




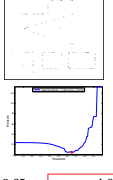
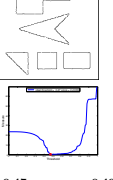




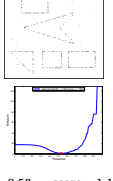
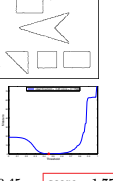

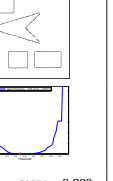

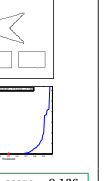
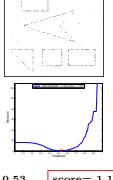
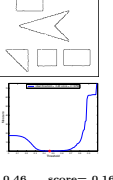


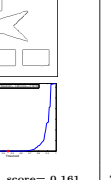
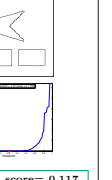
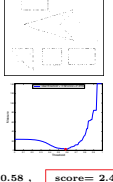
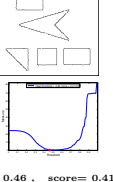



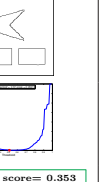
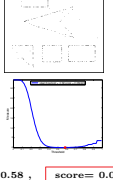
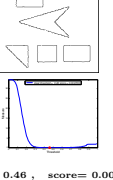
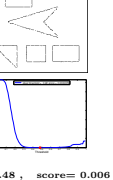
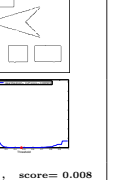
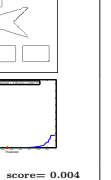
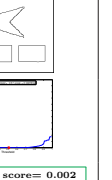
Measure	Sobel [Sob70]	Canny [Can86b]	SF_1 [FA91]	SF_5 [JU04a]	AGK [GSvdW02]	H-K [MMD11a]
D_p	 Th= 0.38 , score= 0.028	 Th= 0.39 , score= 0.009	 Th= 0.38 , score= 0.011	 Th= 0.39 , score= 0.009	 Th= 0.38 , score= 0.012	 Th= 0.51 , score= 0.008
H	 Th= 0.66 , score= 32.14	 Th= 0.49 , score= 2.236	 Th= 0.50 , score= 4.242	 Th= 0.54 , score= 5.385	 Th= 0.45 , score= 2.236	 Th= 0.51 , score= 2.828
$H_{5\%}$	 Th= 0.58 , score= 7.601	 Th= 0.46 , score= 1.083	 Th= 0.48 , score= 1.910	 Th= 0.46 , score= 1.807	 Th= 0.45 , score= 1.283	 Th= 0.51 , score= 1.128
Δ^k	 Th= 0.65 , score= 4.618	 Th= 0.47 , score= 0.403	 Th= 0.48 , score= 0.525	 Th= 0.53 , score= 0.780	 Th= 0.47 , score= 0.510	 Th= 0.51 , score= 0.327
f_2d_6	 Th= 0.53 , score= 1.183	 Th= 0.45 , score= 1.758	 Th= 0.46 , score= 0.256	 Th= 0.45 , score= 0.200	 Th= 0.41 , score= 0.216	 Th= 0.50 , score= 0.136
$S_{k=1}^k$	 Th= 0.53 , score= 1.130	 Th= 0.46 , score= 0.167	 Th= 0.48 , score= 0.219	 Th= 0.46 , score= 0.179	 Th= 0.45 , score= 0.161	 Th= 0.51 , score= 0.117
$S_{k=2}^k$	 Th= 0.58 , score= 2.470	 Th= 0.46 , score= 0.414	 Th= 0.48 , score= 0.531	 Th= 0.53 , score= 0.751	 Th= 0.45 , score= 0.419	 Th= 0.51 , score= 0.353
Ψ proposed measure	 Th= 0.58 , score= 0.052	 Th= 0.46 , score= 0.004	 Th= 0.48 , score= 0.006	 Th= 0.47 , score= 0.008	 Th= 0.45 , score= 0.004	 Th= 0.51 , score= 0.002

Figure 5.13: Synthetic image SNR= 4 dB: Best maps for the compared edge detection evaluations.

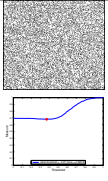
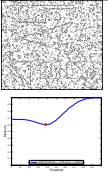
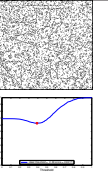
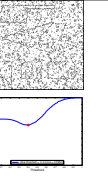
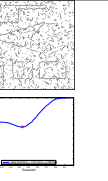
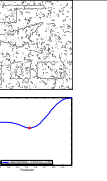
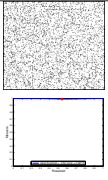
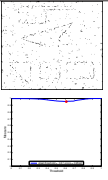
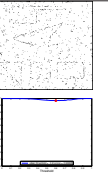
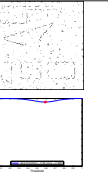
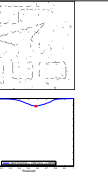
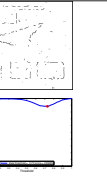
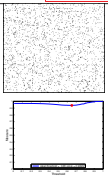
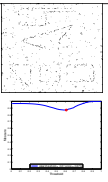
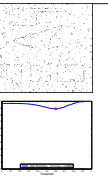
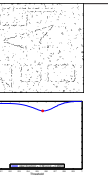
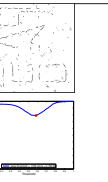
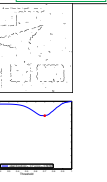
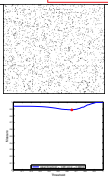
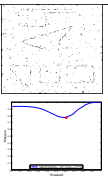
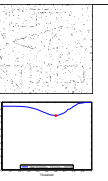
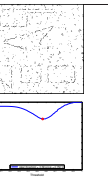
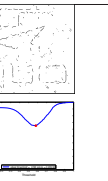
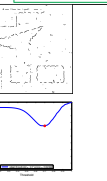
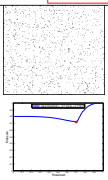
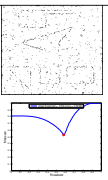
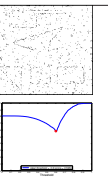
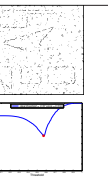
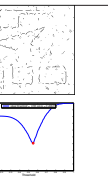
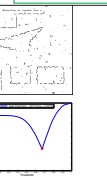
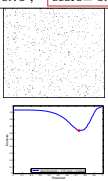
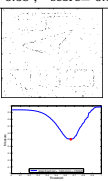
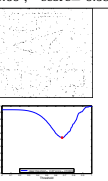
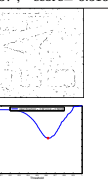
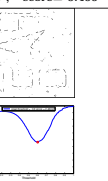
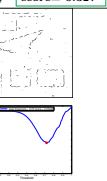
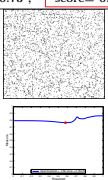
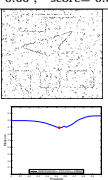
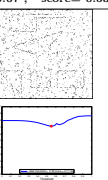
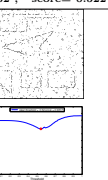
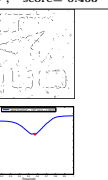
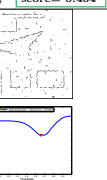
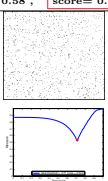
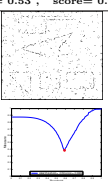
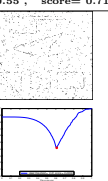
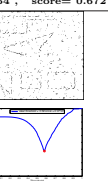
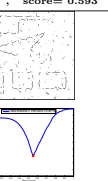
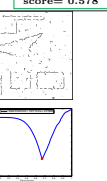
Measure	Sobel [Sob70]	Canny [Can86b]	SF_1 [FA91]	SF_5 [JU04a]	AGK [GSvdW02]	H-K [MMD11a]
Φ^*	 Th= 0.37 , score= 0.680	 Th= 0.38 , score= 0.602	 Th= 0.39 , score= 0.621	 Th= 0.40 , score= 0.595	 Th= 0.43 , score= 0.567	 Th= 0.53 , score= 0.551
χ^{2*}	 Th= 0.54 , score= 0.991	 Th= 0.61 , score= 0.954	 Th= 0.60 , score= 0.968	 Th= 0.59 , score= 0.946	 Th= 0.58 , score= 0.885	 Th= 0.73 , score= 0.884
P_m^*	 Th= 0.65 , score= 0.939	 Th= 0.61 , score= 0.871	 Th= 0.60 , score= 0.889	 Th= 0.56 , score= 0.858	 Th= 0.58 , score= 0.790	 Th= 0.70 , score= 0.787
F_α^*	 Th= 0.65 , score= 0.886	 Th= 0.61 , score= 0.771	 Th= 0.60 , score= 0.801	 Th= 0.56 , score= 0.752	 Th= 0.58 , score= 0.653	 Th= 0.70 , score= 0.649
FoM	 Th= 0.70 , score= 0.718	 Th= 0.58 , score= 0.528	 Th= 0.60 , score= 0.585	 Th= 0.57 , score= 0.516	 Th= 0.55 , score= 0.406	 Th= 0.67 , score= 0.327
F	 Th= 0.73 , score= 0.634	 Th= 0.66 , score= 0.506	 Th= 0.67 , score= 0.530	 Th= 0.62 , score= 0.522	 Th= 0.60 , score= 0.463	 Th= 0.72 , score= 0.454
d_4	 Th= 0.58 , score= 0.763	 Th= 0.53 , score= 0.689	 Th= 0.55 , score= 0.711	 Th= 0.54 , score= 0.672	 Th= 0.57 , score= 0.593	 Th= 0.66 , score= 0.578
$SFoM$	 Th= 0.71 , score= 0.534	 Th= 0.59 , score= 0.382	 Th= 0.61 , score= 0.420	 Th= 0.58 , score= 0.371	 Th= 0.55 , score= 0.301	 Th= 0.67 , score= 0.258

Figure 5.14: Synthetic image SNR= 3.3 dB: Best maps for the compared edge detection evaluations.

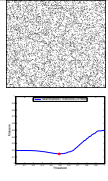
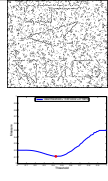
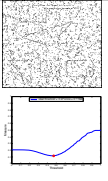
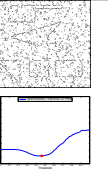
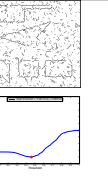

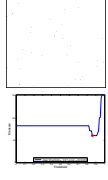
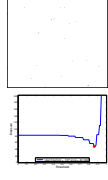
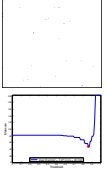
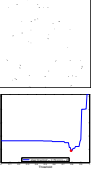
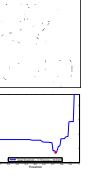
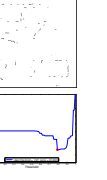
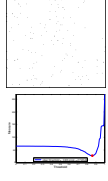
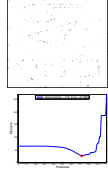
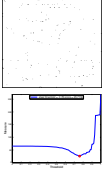

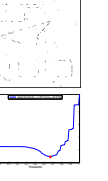
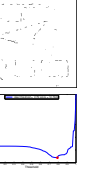
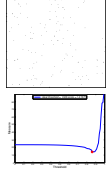
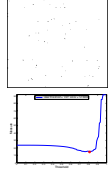
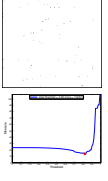

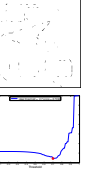
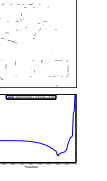
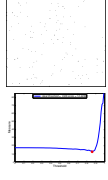
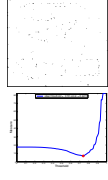
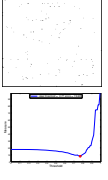
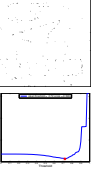
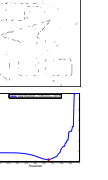
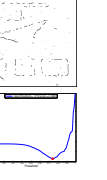
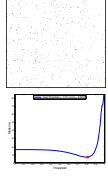
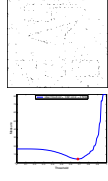
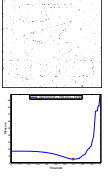
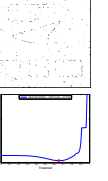
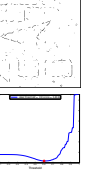

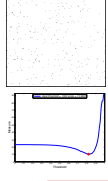
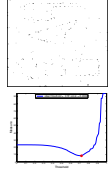
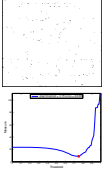
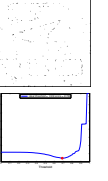
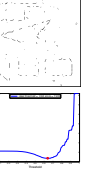

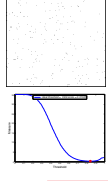
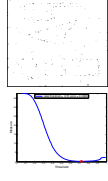
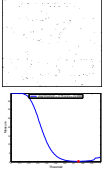
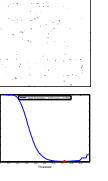


Measure	Sobel [Sob70]	Canny [Can86b]	SF_1 [FA91]	SF_5 [JU04a]	AGK [GSvdW02]	H-K [MMD11a]
D_P	 Th= 0.49 , score= 0.145	 Th= 0.43 , score= 0.108	 Th= 0.47 , score= 0.115	 Th= 0.46 , score= 0.115	 Th= 0.46 , score= 0.098	 Th= 0.55 , score= 0.092
H	 Th= 0.91 , score= 60.63	 Th= 0.87 , score= 48.33	 Th= 0.87 , score= 48.33	 Th= 0.79 , score= 45.00	 Th= 0.74 , score= 35.69	 Th= 0.81 , score= 33.60
$H_{5\%}$	 Th= 0.86 , score= 27.65	 Th= 0.72 , score= 26.89	 Th= 0.76 , score= 25.77	 Th= 0.72 , score= 25.87	 Th= 0.67 , score= 22.67	 Th= 0.79 , score= 19.72
Δ^k	 Th= 0.86 , score= 14.16	 Th= 0.81 , score= 14.52	 Th= 0.82 , score= 13.86	 Th= 0.79 , score= 13.04	 Th= 0.70 , score= 8.739	 Th= 0.80 , score= 8.261
f_2d_6	 Th= 0.86 , score= 12.44	 Th= 0.74 , score= 6.941	 Th= 0.77 , score= 7.840	 Th= 0.72 , score= 6.526	 Th= 0.65 , score= 3.917	 Th= 0.74 , score= 2.982
$S_{k=1}^k$	 Th= 0.81 , score= 6.993	 Th= 0.68 , score= 4.503	 Th= 0.69 , score= 5.208	 Th= 0.65 , score= 5.015	 Th= 0.60 , score= 3.424	 Th= 0.73 , score= 2.734
$S_{k=2}^k$	 Th= 0.82 , score= 10.38	 Th= 0.72 , score= 8.260	 Th= 0.75 , score= 8.835	 Th= 0.69 , score= 8.743	 Th= 0.64 , score= 7.403	 Th= 0.75 , score= 5.842
Ψ proposed measure	 Th= 0.84 , score= 0.310	 Th= 0.72 , score= 0.238	 Th= 0.75 , score= 0.245	 Th= 0.72 , score= 0.253	 Th= 0.67 , score= 0.202	 Th= 0.75 , score= 0.174

Figure 5.15: Synthetic image SNR= 3.3 dB: Best maps for the compared edge detection evaluations.

5.4.2 Real images

To evaluate the performances of the dissimilarity measures, by using first the ground truth image built with human-labelled and second by using the ground truth images of our database. Table 5.4 mentions scores involving the two different G_t : by human-labelled, and our G_t a semi-automatic. It is important to note that the scores for each measure is smaller concerning G_t built in a semi-automatic way (our G_t), than G_t built with human-labelled.

5.4.2.1 Real images results with Gt built with human-labelled

This experiment concerns a real image presented in Fig. 5.5(a); G_t is available in Fig. 5.5(c).

- Edge images associated to the minimum score for several dissimilarity measures and several edge detection methods concerning a real image are presented in Fig. 5.17 and Fig. 5.18. To compare with our ground truth, here, the ground truth is computed by a consensus from human labellers with the method [set11]. The consensus image is build involving the human-labelled contour images 1105_126007, 1113_126007, 1114_126007, 1115_126007 and 1119_126007 from a real image. With the method in [LMDBB16].

Table 5.4: Comparison of scores of dissimilarity measures using a ground truth from [MFTM01] (Fig. 5.3 (b)) image and a constructed ground truth by a semi-automatic way. Contour images and curves for all the measures are available in the supplementary material.

Meas.	Sobel [Sob70]		Canny [Can86b]		SF_1 [FA91]		AGK [GSvdW02]		H-K [MMD11a]	
	Berkeley G_t	Our G_t	Berkeley G_t	Our G_t	Berkeley G_t	Our G_t	Berkeley G_t	Our G_t	Berkeley G_t	Our G_t
Φ^*	0.738	0.298	0.757	0.430	0.971	0.447	0.813	0.496	0.761	0.504
χ^{2*}	0.979	0.635	0.975	0.725	0.983	0.712	0.982	0.759	0.973	0.502
P_m^*	0.901	0.530	0.901	0.603	0.909	0.594	0.917	0.637	0.893	0.778
F_α^*	0.820	0.360	0.819	0.432	0.834	0.422	0.847	0.468	0.808	0.483
FoM	0.303	0.168	0.310	0.147	0.309	0.164	0.299	0.154	0.277	0.146
F	0.592	0.346	0.579	0.352	0.572	0.310	0.589	0.337	0.589	0.367
d_4	0.675	0.333	0.671	0.379	0.687	0.375	0.695	0.412	0.667	0.424
$SFoM$	0.297	0.145	0.289	0.134	0.270	0.111	0.271	0.119	0.268	0.128
D_P	0.173	0.036	0.184	0.058	0.193	0.056	0.208	0.065	0.183	0.072
H	40.02	29.52	19.41	15.175	18.97	18.02	35.35	14.76	36.87	15.03
$H_{5\%}$	13.72	9.406	11.89	9.142	11.53	6.781	14.18	6.048	14.56	7.165
Δ^k	6.632	4.094	5.039	3.000	4.844	2.462	6.044	2.040	6.562	2.576
$f_2 d_6$	2.851	1.066	2.498	1.294	2.467	0.900	2.625	0.895	2.582	0.983
$S_{k=1}^k$	2.584	1.005	2.315	0.990	2.316	0.877	2.471	0.866	2.432	0.966
$S_{k=2}^k$	4.270	2.323	3.725	2.361	3.690	1.819	4.172	1.667	4.281	2.029
Ψ	0.213	0.041	0.181	0.044	0.173	0.032	0.224	0.032	0.222	0.038

Our G_t and G_t of Berkeley segmentation image (Fig. 5.3(b)). Excepted for Φ^* , d_4 and D_p measures, the best edge map for all the other measures contains many holes in the contour chains and it is clearly impossible to conclude which edge detector is the most efficient.

Statistical measures and d_4 consider that Sobel is the best edge detector for this image because edges are well localized. Even though edge maps are different, the scores obtained by FoM and F are similar for the different filtering techniques. Oriented kernels, however, are qualified as reliable by distance measures and edge maps corresponding to the minimum scores are less noisy.

5.4.2.2 Real images results with Gt from our new database

- Figs. 5.19, 5.22 and 5.25 shows a comparison of best maps and minimum scores for different edge detector filters using our proposed measure ψ . We can clearly determine which filter is better than the other filter. Plus, our measurement gave a good result with HGK filter. Figs. 5.20 to 5.27: Assessment of edge detection evaluations concerning real images. Processing steps are the same as in Fig. 5.17 and Fig. 5.18. Ground truth edge maps are available in the database presented in Fig. 5.9.

When objects appear clear, like in image 56 and buildings, most of the measure scores indicate that the edge detectors are equivalent. By contrast, as soon as images contain blur or/and noise, as in image 109 and parkingmeter, the evaluation measures involving error distances considerate that oriented and anisotropic filters produce better-defined contours. Finally, image 109 is a noisy image, however Δ^k and D_p evaluate that Sobel detects better edge, whereas it creates many undesirable contour points, contrary to filtering techniques involving smoothing effects.

Numerous experiments show that $S_{k=1 \text{ or } k=2}^k$ and Ψ dissimilarity measures are best suited for the problem of supervised edge evaluation. Indeed, the minimum evaluation scores are coherent and the edge detectors are qualified as best when the filtering technique is adapted to the image structure (blur, noise, small objects). Moreover, the edge map corresponding to the minimum score delimit correctly the object with a majority of continuous contours points without much undesirable points.

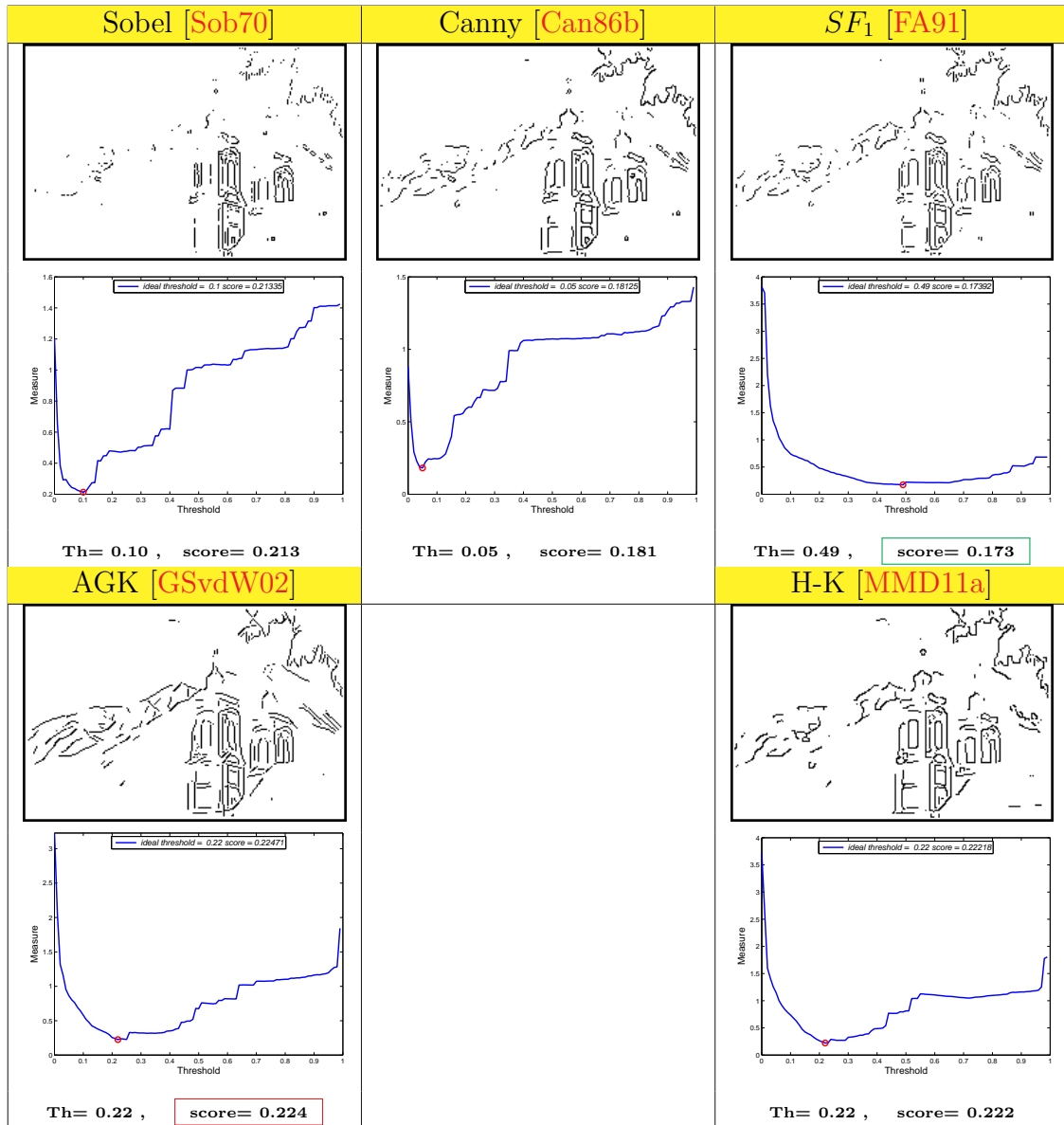


Figure 5.16: Real image: Best maps for the compared edge detection filters using our proposed measure Ψ . The ground truth edge image is computer by the consensus from human-labellers.



Figure 5.17: Real image: Best maps for the compared edge detection evaluations. The ground truth edge image is computer by the consensus from human-labellers.

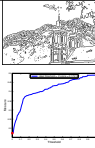
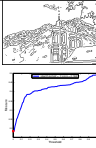
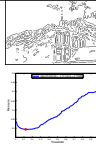
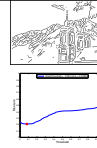
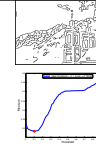
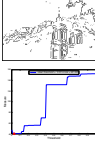
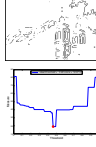
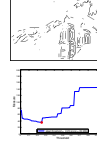
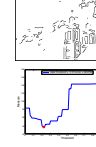
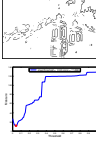
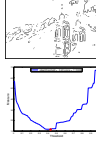
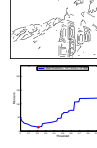
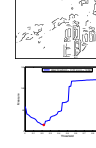
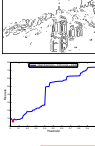
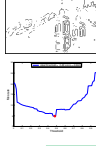
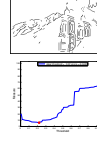
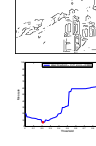
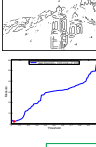
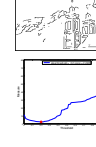
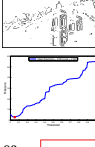
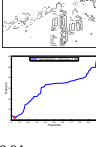
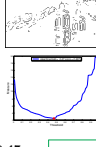
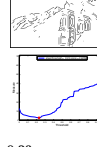
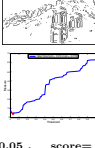
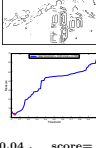
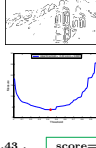
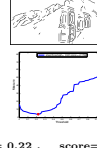
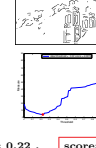
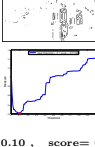
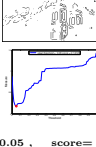
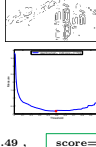
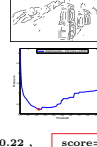
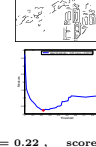
Measure	Sobel [Sob70]	Canny [Can86b]	SF_1 [FA91]	AGK [GSvdW02]	H-K [MMD11a]
D_P	 Th= 0.00 , score= 0.173	 Th= 0.00 , score= 0.184	 Th= 0.12 , score= 0.193	 Th= 0.08 , score= 0.208	 Th= 0.08 , score= 0.183
H	 Th= 0.03 , score= 40.02	 Th= 0.05 , score= 19.41	 Th= 0.49 , score= 18.97	 Th= 0.25 , score= 35.35	 Th= 0.25 , score= 36.87
$H_{5\%}$	 Th= 0.03 , score= 13.72	 Th= 0.04 , score= 11.89	 Th= 0.43 , score= 11.53	 Th= 0.21 , score= 14.18	 Th= 0.21 , score= 14.56
Δ^k	 Th= 0.03 , score= 6.632	 Th= 0.05 , score= 5.039	 Th= 0.48 , score= 4.844	 Th= 0.22 , score= 6.044	 Th= 0.22 , score= 6.562
$S_{k=1}^k$	 Th= 0.05 , score= 2.584	 Th= 0.04 , score= 2.315	 Th= 0.37 , score= 2.316	 Th= 0.21 , score= 2.471	 Th= 0.21 , score= 2.432
f_2d_6	 Th= 0.03 , score= 2.851	 Th= 0.04 , score= 2.498	 Th= 0.47 , score= 2.467	 Th= 0.23 , score= 2.625	 Th= 0.23 , score= 2.589
$S_{k=2}^k$	 Th= 0.05 , score= 4.270	 Th= 0.04 , score= 3.725	 Th= 0.43 , score= 3.690	 Th= 0.22 , score= 4.172	 Th= 0.22 , score= 4.281
Ψ proposed measure	 Th= 0.10 , score= 0.213	 Th= 0.05 , score= 0.181	 Th= 0.49 , score= 0.173	 Th= 0.22 , score= 0.224	 Th= 0.22 , score= 0.222

Figure 5.18: Real image: Best maps for the compared edge detection evaluations. The ground truth edge image is computer by the consensus from human-labellers.

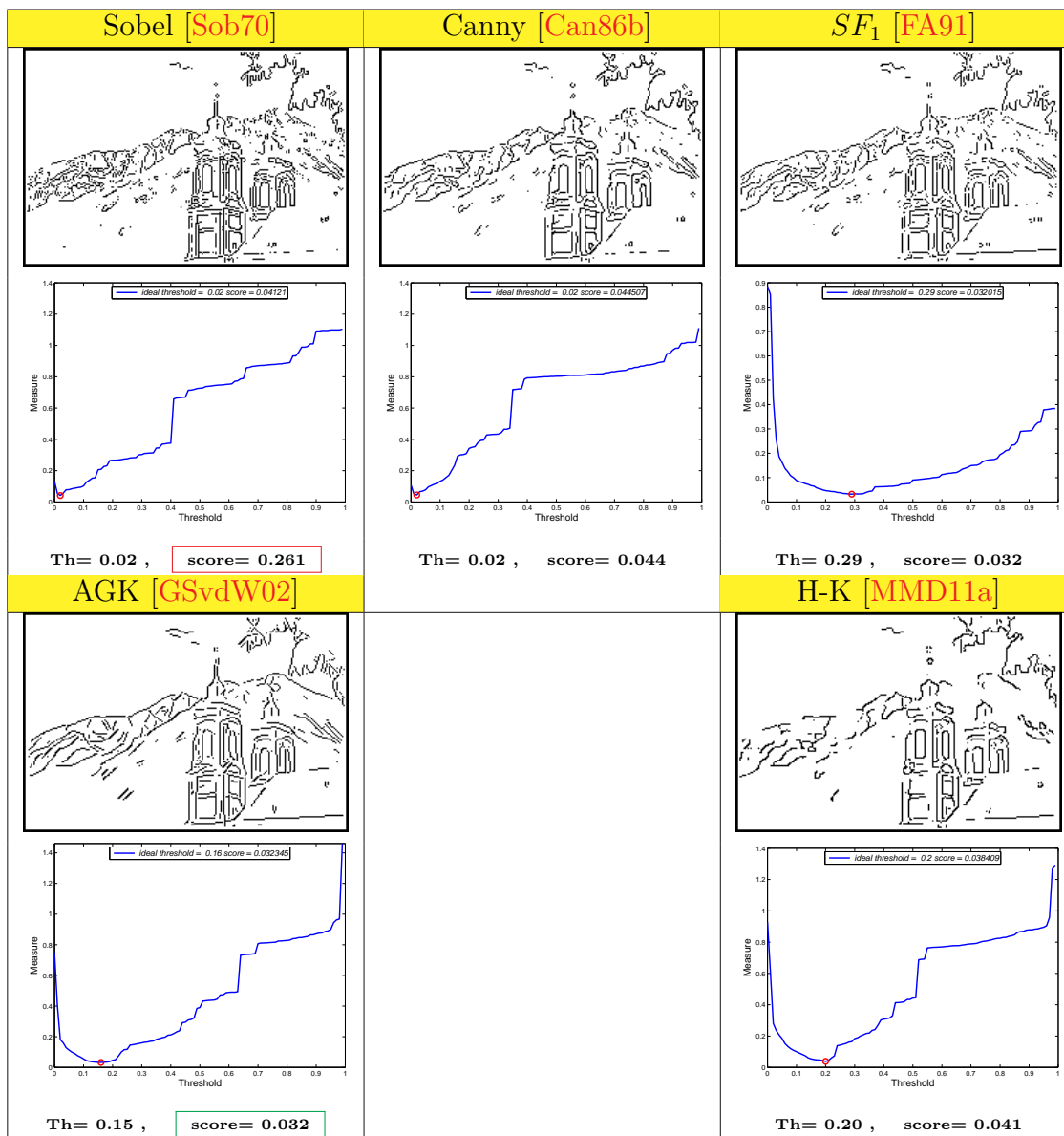


Figure 5.19: Real image: (The original image is available in Fig. 5.9 and the ground truth edge image comes from our database). Best maps for the compared edge detection filters using our proposed measure Ψ .

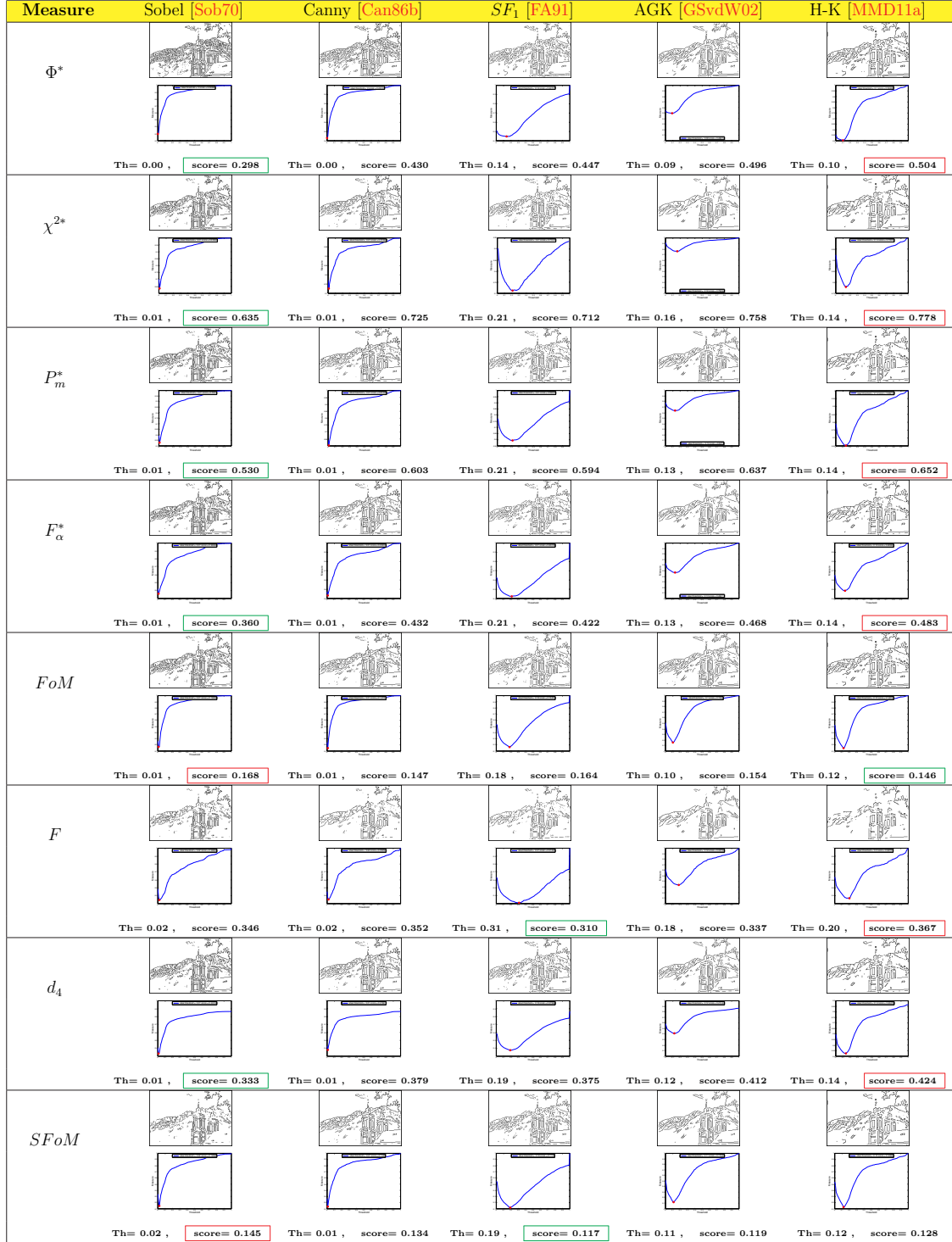


Figure 5.20: Real image: Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.

Measure	Sobel [Sob70]	Canny [Can86b]	SF_1 [FA91]	AGK [GSvdW02]	H-K [MMD11a]
D_P	 Th= 0.01 , score= 0.036	 Th= 0.00 , score= 0.058	 Th= 0.18 , score= 0.056	 Th= 0.10 , score= 0.065	 Th= 0.11 , score= 0.072
H	 Th= 0.03 , score= 29.52	 Th= 0.02 , score= 25.17	 Th= 0.33 , score= 18.02	 Th= 0.18 , score= 14.76	 Th= 0.20 , score= 15.03
$H_{5\%}$	 Th= 0.02 , score= 9.406	 Th= 0.02 , score= 9.142	 Th= 0.29 , score= 6.781	 Th= 0.16 , score= 6.048	 Th= 0.20 , score= 7.165
Δ^k	 Th= 0.03 , score= 4.094	 Th= 0.02 , score= 3.000	 Th= 0.28 , score= 2.462	 Th= 0.14 , score= 2.040	 Th= 0.20 , score= 2.576
$S_{k=1}^k$	 Th= 0.01 , score= 1.005	 Th= 0.01 , score= 0.990	 Th= 0.26 , score= 0.877	 Th= 0.15 , score= 0.866	 Th= 0.16 , score= 0.966
$f_2 d_6$	 Th= 0.02 , score= 1.066	 Th= 0.02 , score= 1.294	 Th= 0.29 , score= 0.900	 Th= 0.14 , score= 0.895	 Th= 0.15 , score= 0.983
$S_{k=2}^k$	 Th= 0.02 , score= 2.323	 Th= 0.01 , score= 2.361	 Th= 0.29 , score= 1.819	 Th= 0.15 , score= 1.667	 Th= 0.20 , score= 2.029
Ψ proposed measure	 Th= 0.02 , score= 0.261	 Th= 0.02 , score= 0.044	 Th= 0.29 , score= 0.032	 Th= 0.15 , score= 0.032	 Th= 0.20 , score= 0.041

Figure 5.21: Real image: Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.

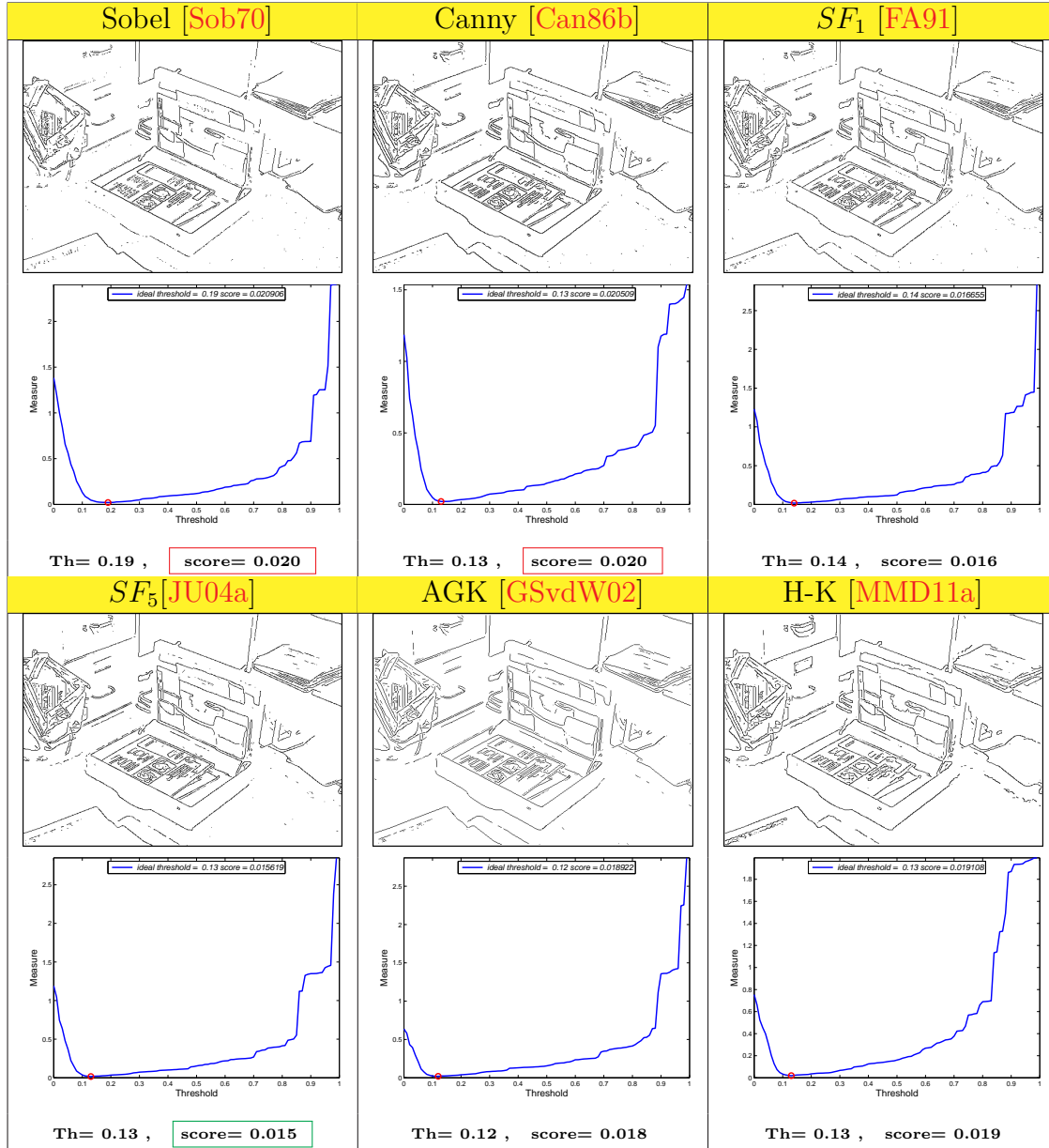


Figure 5.22: Real image 36 (The original image is available in Fig. 5.9 and the ground truth edge image comes from our database). Best maps for the compared edge detection filters using our proposed measure Ψ .

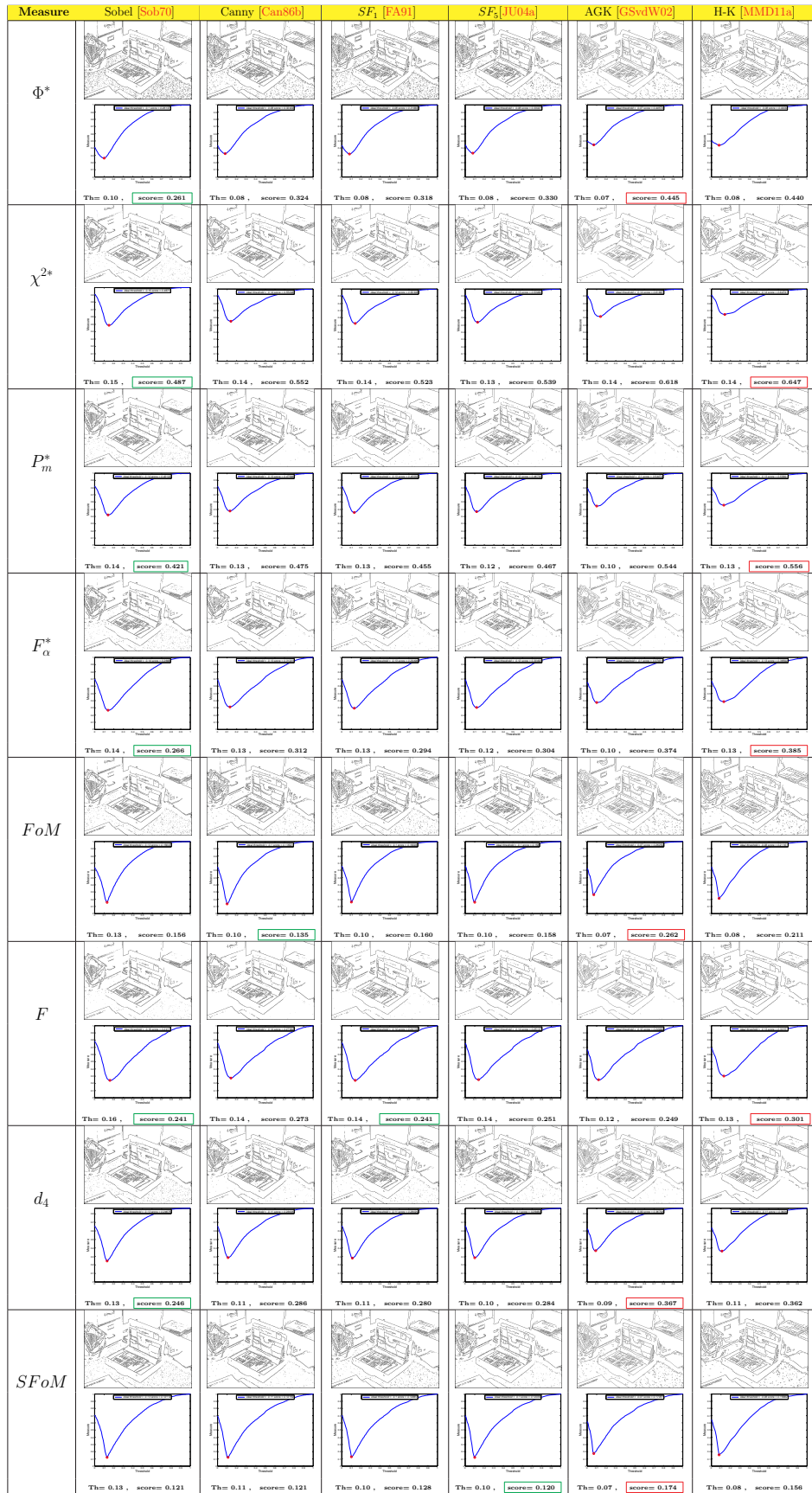


Figure 5.23: Real image 36: Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.

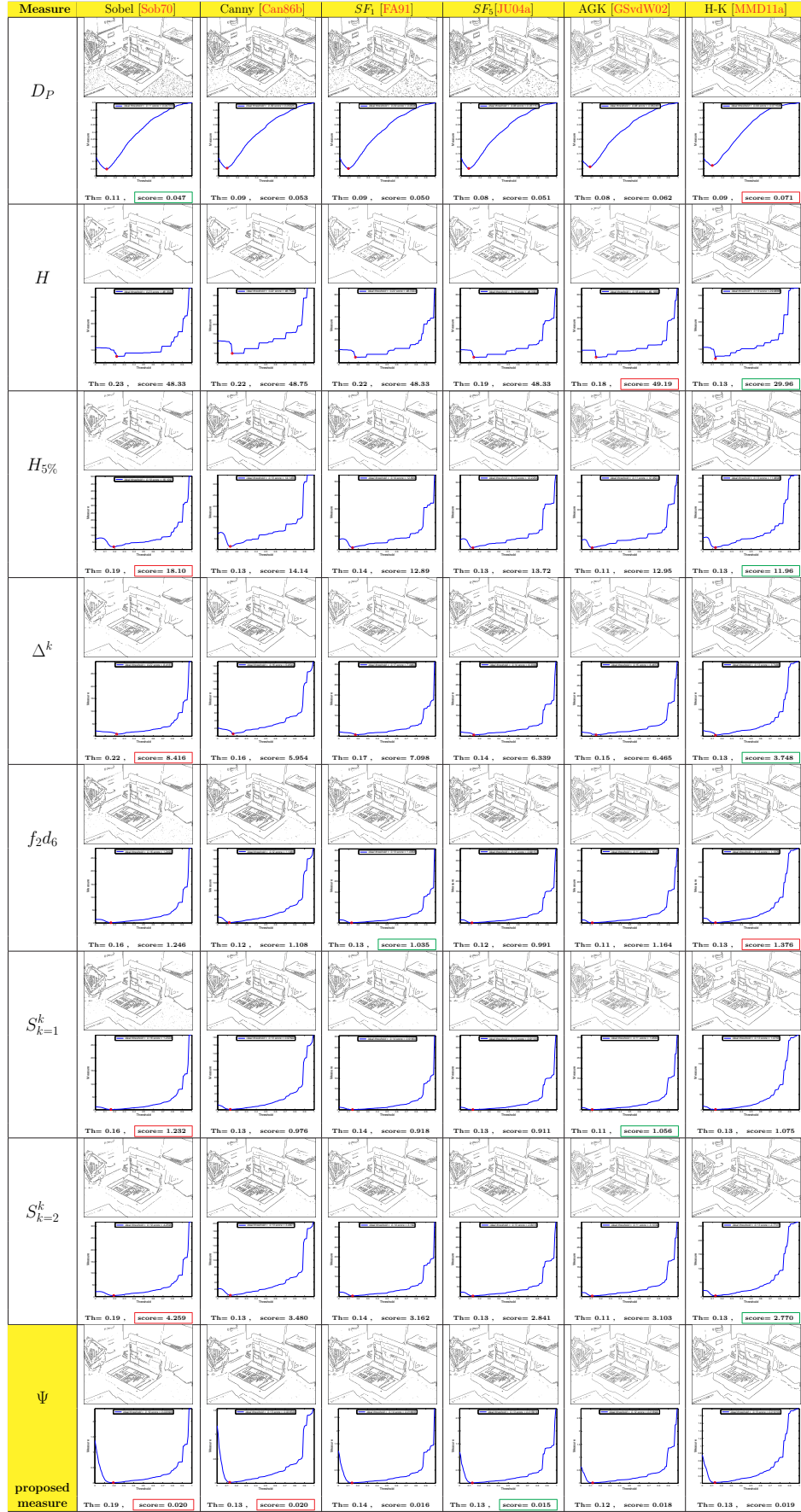


Figure 5.24: Real image 36: Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.

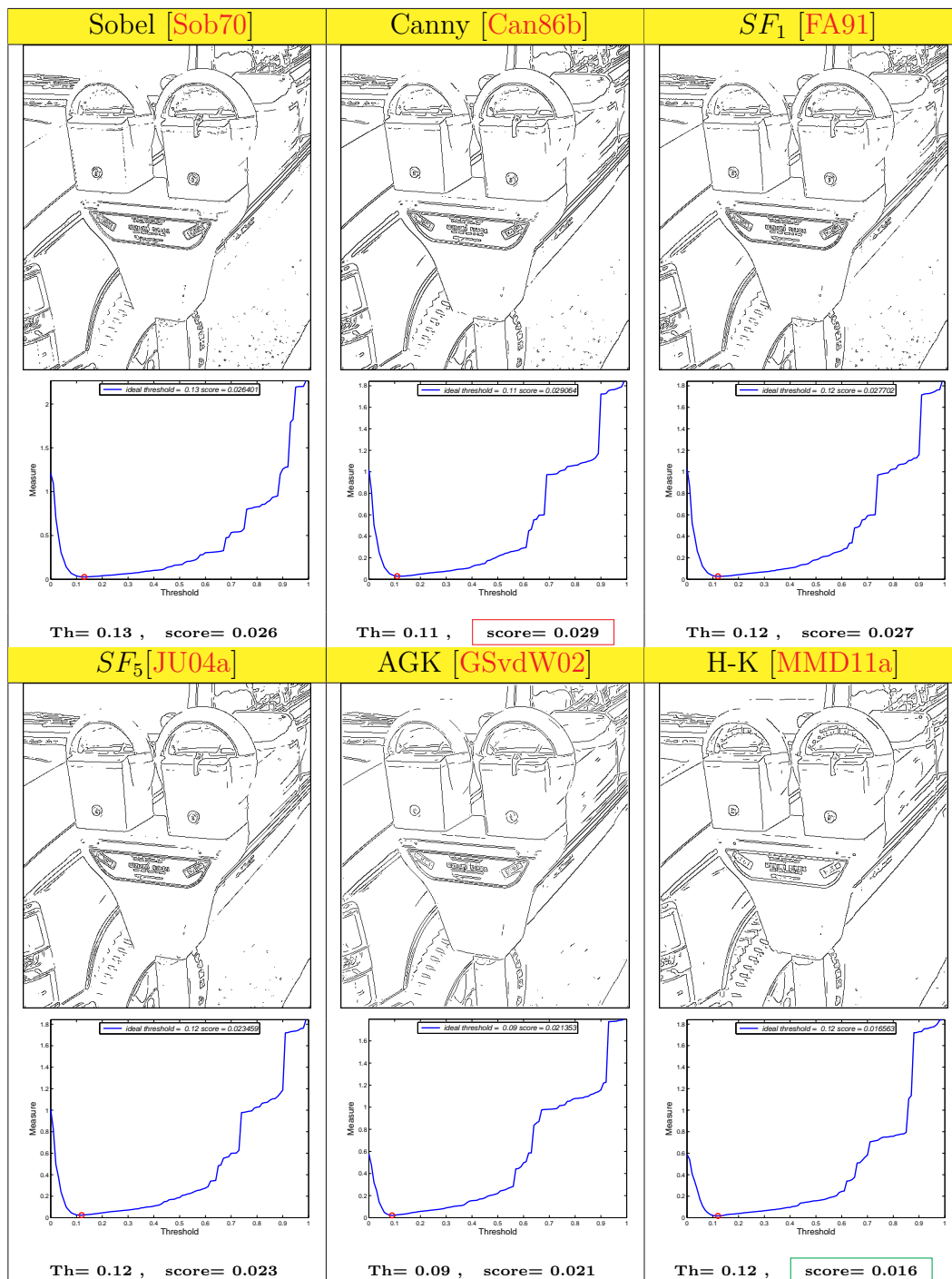


Figure 5.25: Real image **parkingmeter** (The original image is available in Fig. 5.9 and the ground truth edge image comes from our database). Best maps for the compared edge detection filters using our proposed measure Ψ .

















































Measure	Sobel [Sob70]	Canny [Can86b]	SF_1 [FA91]	SF_3 [JU04a]	AGK [GSvdW02]	H-K [MMD11a]
Φ^*	 Th= 0.06 , score= 0.255	 Th= 0.05 , score= 0.355	 Th= 0.06 , score= 0.332	 Th= 0.06 , score= 0.346	 Th= 0.05 , score= 0.424	 Th= 0.07 , score= 0.454
χ^{2*}	 Th= 0.10 , score= 0.489	 Th= 0.10 , score= 0.597	 Th= 0.11 , score= 0.560	 Th= 0.10 , score= 0.568	 Th= 0.09 , score= 0.619	 Th= 0.11 , score= 0.659
P_m^*	 Th= 0.10 , score= 0.417	 Th= 0.10 , score= 0.508	 Th= 0.10 , score= 0.478	 Th= 0.10 , score= 0.486	 Th= 0.08 , score= 0.533	 Th= 0.11 , score= 0.563
F_α^*	 Th= 0.10 , score= 0.264	 Th= 0.10 , score= 0.340	 Th= 0.10 , score= 0.314	 Th= 0.10 , score= 0.321	 Th= 0.08 , score= 0.363	 Th= 0.11 , score= 0.391
FoM	 Th= 0.09 , score= 0.146	 Th= 0.08 , score= 0.133	 Th= 0.08 , score= 0.151	 Th= 0.08 , score= 0.161	 Th= 0.05 , score= 0.214	 Th= 0.07 , score= 0.206
F	 Th= 0.11 , score= 0.261	 Th= 0.11 , score= 0.304	 Th= 0.11 , score= 0.277	 Th= 0.10 , score= 0.280	 Th= 0.09 , score= 0.277	 Th= 0.11 , score= 0.305
d_4	 Th= 0.09 , score= 0.246	 Th= 0.08 , score= 0.309	 Th= 0.09 , score= 0.291	 Th= 0.08 , score= 0.297	 Th= 0.07 , score= 0.348	 Th= 0.10 , score= 0.366
$SFoM$	 Th= 0.09 , score= 0.131	 Th= 0.08 , score= 0.116	 Th= 0.08 , score= 0.123	 Th= 0.08 , score= 0.128	 Th= 0.06 , score= 0.160	 Th= 0.07 , score= 0.157

Figure 5.26: Real image **parkingmeter**: Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.

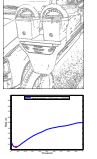
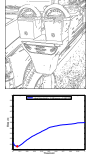
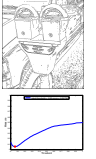
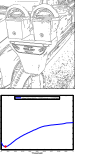
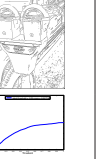

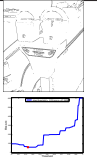
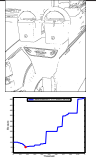
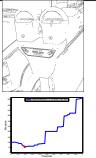
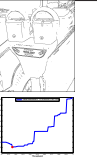
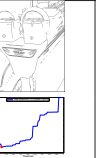

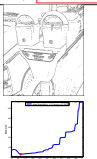
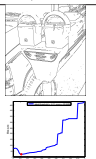
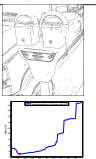



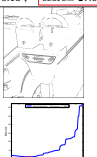
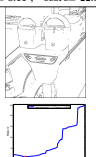
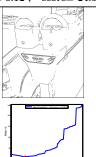
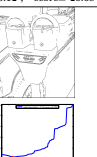








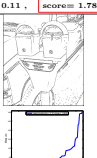

















Measure	Sobel [Sob70]	Canny [Can86b]	SF_1 [FA91]	SF_2 [JU04a]	AGK [GSvdW02]	H-K [MMD11a]
D_P	 Th= 0.07 , score= 0.050	 Th= 0.06 , score= 0.061	 Th= 0.06 , score= 0.057	 Th= 0.06 , score= 0.057	 Th= 0.05 , score= 0.067	 Th= 0.07 , score= 0.078
H	 Th= 0.25 , score= 61.84	 Th= 0.17 , score= 54.33	 Th= 0.19 , score= 59.08	 Th= 0.15 , score= 54.14	 Th= 0.13 , score= 54.03	 Th= 0.14 , score= 59.35
$H_{5\%}$	 Th= 0.13 , score= 27.08	 Th= 0.11 , score= 22.88	 Th= 0.12 , score= 24.52	 Th= 0.12 , score= 20.68	 Th= 0.09 , score= 16.29	 Th= 0.12 , score= 11.18
Δ^k	 Th= 0.24 , score= 11.89	 Th= 0.17 , score= 9.943	 Th= 0.20 , score= 10.73	 Th= 0.15 , score= 8.739	 Th= 0.13 , score= 7.135	 Th= 0.14 , score= 4.522
f_2d_6	 Th= 0.11 , score= 1.780	 Th= 0.10 , score= 1.428	 Th= 0.11 , score= 1.679	 Th= 0.10 , score= 1.393	 Th= 0.08 , score= 1.325	 Th= 0.10 , score= 1.121
$S_{k=1}^k$	 Th= 0.12 , score= 1.608	 Th= 0.10 , score= 1.396	 Th= 0.11 , score= 1.470	 Th= 0.10 , score= 1.272	 Th= 0.08 , score= 1.153	 Th= 0.11 , score= 0.994
$S_{k=2}^k$	 Th= 0.13 , score= 5.826	 Th= 0.11 , score= 5.181	 Th= 0.12 , score= 5.422	 Th= 0.12 , score= 4.553	 Th= 0.09 , score= 3.836	 Th= 0.12 , score= 2.686
Ψ proposed measure	 Th= 0.13 , score= 0.026	 Th= 0.11 , score= 0.029	 Th= 0.12 , score= 0.027	 Th= 0.12 , score= 0.023	 Th= 0.09 , score= 0.021	 Th= 0.12 , score= 0.016

Figure 5.27: Real image **parkingmeter**: Best maps for the compared edge detection evaluations. The ground truth edge image comes from our database.

5.5 Summary

To summarize this chapter, here a study presents a review of supervised edge detection assessment methods in details. Moreover, based on the theory of these dissimilarity evaluations, a technique is proposed to evaluate filtering edge detection methods involving the minimum score of the considerate measures. Indeed, to evaluate an edge detection technique, the result which obtains the minimum score of a measure is considerate as the best one and represents an objective evaluation. Theoretically and with the backing of many experiments is demonstrated that the minimum score of the $S_{k=1 \text{ or } k=2}^k$ and Ψ dissimilarity measures correspond to the best edge quality map evaluations. These two measures take into account both the distances of false positive and false negative points. Many experiments of edge detection on synthetic and real images involving several edge detectors illustrate this conclusion. Experiments show the significance of the ground truth map choice: an inaccurate ground truth contour map in terms of localization penalizes precise edge detectors and/or advantages the rough algorithms. That is the reason why is described in this conversation how to build a new ground truth edge map labelled in semi-automatic way in real images. Firstly, the contours are detected involving the convolution of the image with $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ masks. Secondly, undesirable edges are removed while missing points are added both by hand, thus a more accuracy ground truth edge map image is built and can be used for supervised contour detection evaluation. By comparison with a real image where contours points are not precisely labelled, experiments illustrate that the new ground truth database allows to evaluate the performance of edge detectors by filtering. Finally, the advantage to compute the minimum score of a measure involving this new ground truth database is that it does not require tuning parameters. This methodology forms one of our contribution and resulted in the Journal publication [AMM17a].

6

OBJECTIVE SUPERVISED EDGE DETECTION USING HYSTERESIS THRESHOLDS

This chapter describes a novel technique to compare edge detection techniques by using hysteresis thresholds in a supervised way.

Contents

6.1	Introduction	100
6.2	Supervised Measures for Image Contour Evaluations	101
6.2.1	Distances of misplaced pixels	101
6.2.2	A new objective edge detection assessment measure:	103
6.2.3	Minimum of the measure and ground truth edge image	104
6.3	Experimental results	105
6.4	Summary	109

6.1 Introduction

Edge detection is an important field in image processing because this process frequently attempts to capture the most important structures in the image. Hence, edge detection represents a fundamental step concerning computer vision approaches. Furthermore, edge detection itself could be used to qualify a region segmentation technique. Additionally, the edge detection assessment remains very useful in image segmentation, registration, reconstruction or interpretation. Hence, it is hard to design an edge detector which is able to extract the exact edge with good localization and orientation from an image. In the literature, different techniques have emerged and, due to its importance, edge detection continues to be an active research area [AMFM11]. The best-known and useful edge detection methods are based on gradient computing first-order fixed operators [Sob70, Can86a]. Oriented operators compute the maximum energy in an orientation [FA91, JU04a, GSvdW02] or two directions [MMD11a]. Typically, these methods are composed of three steps:

1. Computation of the gradient magnitude and its orientation η , see Fig. 6.1.
2. Non-maximum suppression to obtain thin edges: the selected pixels are those having gradient magnitude at a local maximum along the gradient direction η which is perpendicular to the edge orientation.
3. Thresholding of the thin contours to obtain an edge map.

Thus, Fig. 6.1 exposes the different possibilities of gradient and its associated orientations involving several edge detection algorithms compared in this chapter.

The final step remains a difficult stage in image processing, however it represents a crucial operation to compare several segmentation algorithms. In edge detection, the hysteresis process uses the connectivity information of the pixels belonging to thin contours and thus remains a more elaborated method than binary thresholding. Simply, this technique determines a contour image that has been thresholded at different levels (low: τ_L and high: τ_H). The low threshold τ_L determines which pixels are considered as edge points if at least one point higher than τ_H exists in a contour chain where all the pixel values are also higher than τ_L , as represented with a signal in Fig. 6.1. Thus, the lower the thresholds are, the more the undesirable pixels are preserved. Usually, in order to compare several

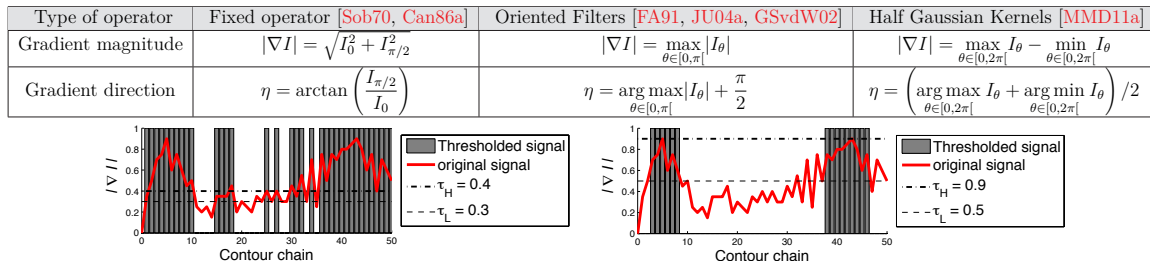


Figure 6.1: Gradient magnitude and orientation computation for a scalar image I and example of hysteresis threshold applied along a contour chain. I_θ represents the image derivative using a first-order filter at the θ orientation (in radians).

edge detection methods, the user has to try some thresholds to select the ones that appear visually as the best edge maps in quality. However, this assessment suffers from a main drawback: segmentations are compared using the threshold (deliberately) chosen by the user, this evaluation is very subjective and not reproducible. Hence, the purpose is to use the dissimilarity measures without any user intervention for an objective assessment. Finally, to consider a valuable edge detection assessment, the evaluation process should produce a result that correlates with the perceived quality of the edge image, which relies on human judgment [HSSB97, MFTM01, AMM17a]. In other words, a reliable edge map should characterize all the relevant structures of an image as closely as possible, without any disappearance of desired contours. Nevertheless, a minimum of spurious pixels can be created by the edge detector, disturbing at the same time the visibility of the main/desired objects to detect.

In this chapter, a novel technique is presented to compare edge detection techniques by using hysteresis thresholds in a supervised way, being consistent with the visual perception of a human [AMM17b]. Indeed, by comparing a ground truth contour map with an ideal edge map, several assessments can be compared by varying the parameters of the hysteresis thresholds. This study shows the importance to penalize stronger the false negative points, compared to the false positive points, leading to a new edge detection evaluation algorithm. The experiment using synthetic and real images demonstrated that the proposed method [AMM17b] obtains contours maps closer to the ground truth without requiring tuning parameters and outperforms other assessment methods in an objective way.

6.2 Supervised Measures for Image Contour Evaluations

A supervised evaluation criterion computes a dissimilarity measure between a segmentation result and a ground truth obtained from synthetic data or an expert judgment (i.e. manual segmentation) [DJ94, CLRE08, LMDBB13, AMM17d]. In this chapter, the closer to 0 the score of the evaluation is, the more the segmentation is qualified as good. This work focusses on comparisons of supervised edge detection evaluations and proposes a new measure, aiming at an objective assessment.

6.2.1 Distances of misplaced pixels

A reference-based edge map quality measure requires that a displaced edge should be penalized in function not only of FPs and/or FNs but also of the distance from the position where it should be located. In the chapter 4, Table 4.1 reviews the most relevant measures involving distances. Thus, for a pixel p belonging to the desired contour D_c ,

$d_{G_t}(p)$ represents the minimal Euclidian distance between p and G_t . If p belongs to the ground truth G_t , $d_{D_c}(p)$ is the minimal distance between p and D_c . On the one hand, some distance measures are specified in the evaluation of over-segmentation (i.e. presence of FPs), like: Υ , D^k , Θ and Γ . On the other hand, Ω measure assesses an edge detection by computing only an under segmentation (FNs). Other edge detection evaluation measures consider both distances of FPs and FNs [AMM17a]. A perfect segmentation using an over-segmentation measure could be an image including no edge points and an image having most undesirable edge points (FPs) concerning under-segmentation evaluations (see Fig. 6.3). Also, another limitation of only over and under-segmentation evaluations are that several binary images can produce the same result (Fig. 6.2). Therefore, as demonstrated in [AMM17a], a complete and optimum edge detection evaluation measure should combine assessments of both over- and under-segmentation.

Among the distance measures between two contours, one of the most popular descriptors is named the Figure of Merit (FoM). Nonetheless, for FoM , the distance of the FNs is not recorded and are strongly penalized as statistic measures (see above). For example, in Fig.3, $FoM(G_t, C) > FoM(G_t, M)$, whereas M contains both FPs and FNs and C only FNs. Further, for the extreme cases:

- if $\hat{E}FP = 0$: $FoM(G_t, D_c) = 1 - TP/|G_t| = 1 - (|G_t| - FN)/|G_t|$,
- if $FN = 0$: $FoM(G_t, D_c) = 1 - \frac{1}{\max(|G_t|, |D_c|)} \cdot \sum_{p \in D_c \cap \neg G_t} \frac{1}{1 + \kappa \cdot d_{G_t}^2(p)}$.

When $FN > 0$ and FP constant, it behaves like matrix-based error assessments (Fig.6.2). Moreover, for $FP > 0$, the FoM penalizes the over-detection very low compared to the under-detection. On the contrary, the F measure computes the distances of FNs but not of the FPs, so F behaves inversely to FoM . Also, d_4 measure depends particularly on TP , FP , FN and FoM but penalizes FNs like the FoM measure. $SFoM$ and $MFoM$ take into account both distances of FNs and FPs, so they can compute a global evaluation of a contour image. However, $MFoM$ does not consider FPs and FNs at the same time, contrary to $SFoM$. Another way to compute a global measure is presented in [PGAN16] with the edge map quality measure D_p . The right term computes the distances of the FNs between the closest correctly detected edge pixel, i.e. $G_t \cap D_c$. Finally, D_p is more sensitive to FNs than FPs because of the coefficient $\frac{1}{|I| - |G_t|}$.

A second measure widely computed in matching techniques is represented by the Hausdorff distance H , which measures the mismatch of two sets of points [HR93]. This max-min distance could be strongly deviated by only one pixel which can be positioned sufficiently far from the pattern (Fig. 6.3). To improve the measure, one idea is to compute H with a proportion of the maximum distances; let us note $H_{15\%}$ this measure for 15% of the

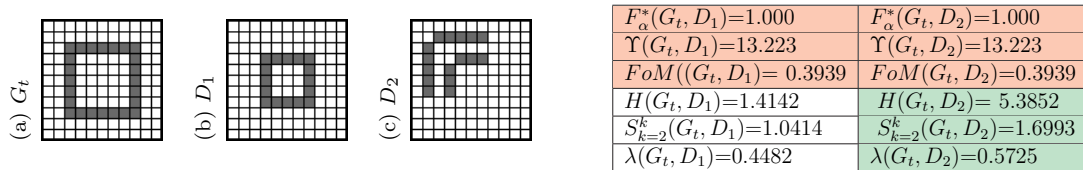


Figure 6.2: Different D_c : FPs and number of FNs are the same for D_1 and for D_2 .

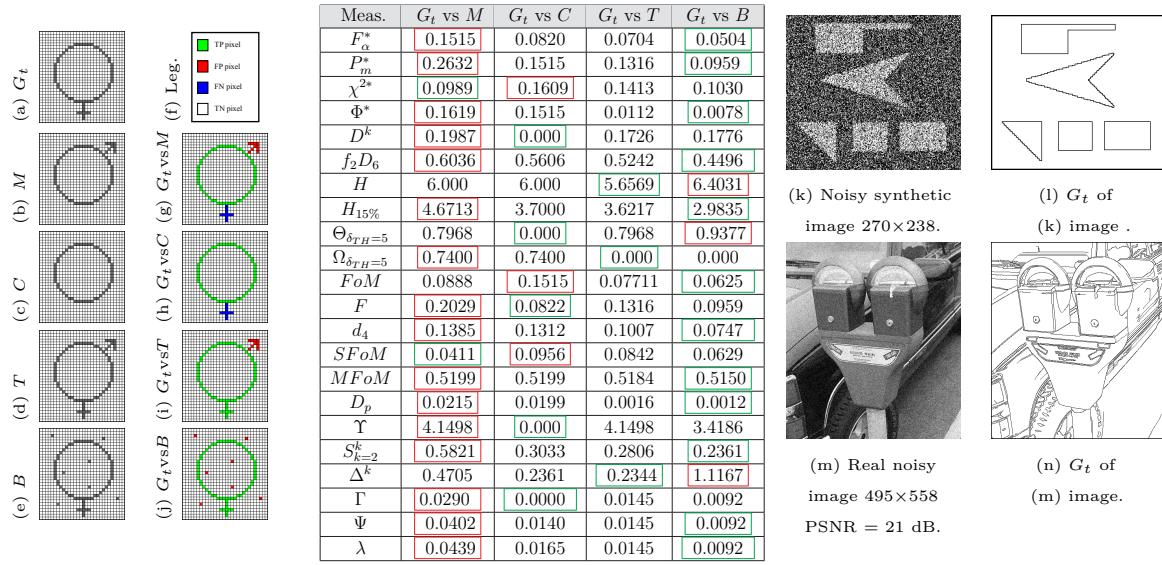


Figure 6.3: Results of evaluation measures and images for the experiments.

values [HR93]. Nevertheless, as pointed out in [DJ94], an average distance from the edge pixels in the candidate image to those in the ground truth is more appropriate, like S^k or Ψ . Eventually, Delta Metric (Δ^k) [Bad92] intends to estimate the dissimilarity between each element of two binary images, but is highly sensitive to distances of misplaced points [AMM17d, AMM17a].

6.2.2 A new objective edge detection assessment measure:

In [AMM17d] a measure of the edge detection assessment is developed: it is denoted Ψ and improves the over-segmentation measure Γ , by combining both d_{G_t} and d_{D_c} , see Fig. 6.3. Ψ gives the same weight for d_{G_t} and d_{D_c} in its assessment of errors. Thus, using Ψ , a missing edge remains not enough penalized contrary to the distance of FPs which could be too important. Another example, in Fig. 6.3, $\Psi(G_t, C) < \Psi(G_t, T)$ whereas C must be more penalized because of FNs which does not allow to identify the object (also Fig. 6.6). The solution proposed here is to penalize stronger the distances of the FNs depending on the number of TPs:

$$\lambda(G_t, D_c) = \frac{FP + FN}{|G_t|^2} \cdot \sqrt{\sum_{p \in D_c} d_{G_t}^2(p) + \min\left(|G_t|^2, \frac{|G_t|^2}{TP^2}\right) \cdot \sum_{p \in G_t} d_{D_c}^2(p)} \quad (6.1)$$

The term influencing the penalization of FN distances can be rewritten as: $\frac{|G_t|^2}{TP^2} = \left(\frac{FN+TP}{TP}\right)^2 = \left(1 + \frac{FN}{TP}\right)^2 \geq 1$, ensuring a stronger penalty for $d_{D_c}^2$, compared to $d_{G_t}^2$. When $TP = 0$, the min function avoids the multiplication by infinity; moreover, the number of FNs is large, corresponding to a strong penalty with the weight term $|G_t|^2$ (see Fig.

6.4 left). When $|G_t| = TP$, λ is equivalent to Ψ and Γ (see Fig. 6.3, image T). Also, compared to Ψ , λ penalizes more D_c having FNs, than D_c with only FPs, as illustrated in Fig. 6.3 (images C and T). Finally, the weight $\frac{|G_t|^2}{TP^2}$ tunes the λ measure by considering an edge map of better quality when FNs points are localized close to the desired contours D_c .

The next subsection details the way to evaluate an edge detector in an objective way. Results presented in this communication show the importance to penalize stronger the false negative points, compared to the false positive points because the desired objects are not always completely visible by using ill-suited evaluation measure, and, λ provides a reliable edge detection assessment.

6.2.3 Minimum of the measure and ground truth edge image

Dissimilarity measures are used for an objective assessment using binary images. Instead of choosing manually a threshold to obtain a binary image (see Fig. 3 in [AMM17a]), the purpose is to compute the minimal value of a dissimilarity measure by varying the thresholds (double loop: loop over τ_L and loop over τ_H) of the thin edges (see the table in Fig. 6.1). Thus, compared to a ground truth contour map, the ideal edge map for a measure corresponds to the desired contour at which the evaluation obtains the minimum score for the considered measure among the thresholded (binary) images. Theoretically, this score corresponds to the thresholds at which the edge detection represents the best edge map, compared to the ground truth contour map [FGMCCP+04, CLRE08, AMM17a]. Fig. 6.4 right illustrates the choice of a contour map in function of τ_L and τ_H . Since small thresholds lead to heavy over-segmentation and strong thresholds may create numerous false negative pixels, the minimum score of an edge detection evaluation should be a compromise between under- and over-segmentation (detailed and illustrated in [AMM17a]). As demonstrated in [AMM17a], the significance of the ground truth map choice influences on the dissimilarity evaluations. Indeed, if not reliable [HYK13], an inaccurate ground truth contour map

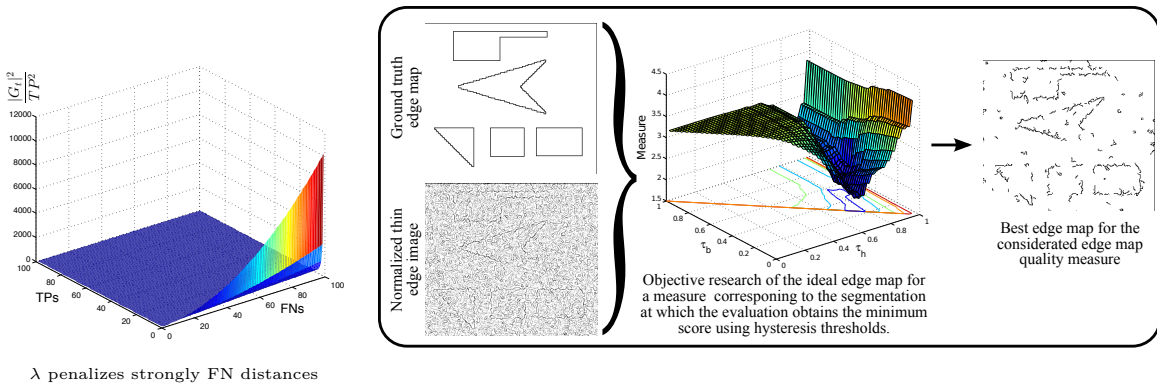


Figure 6.4: Number of FNs penalizes λ and computation of a measure minimum score.

in terms of localization penalizes precise edge detectors and/or advantages the rough algorithms as edge maps presented in [HSSB97, MFTM01]. For these reasons, the ground truth edge map concerning the real image in our experiments is built in a semi-automatic way detailed in [AMM17a].

6.3 Experimental results

In these experiments, the importance of an assessment to penalize stronger the false negative points is enlightened, compared to the false positive points. In order to study the performance of the contour detection evaluation measures, the hysteresis thresholds vary and the minimum score of the studied measure corresponds to the best edge map. The thin edges of both synthetic and real noisy images are computed by five or six edge detectors: Sobel [Sob70], Canny [Can86a], Steerable Filters of order 1 (SF_1) [FA91] or order 5 (SF_5) [JU04a], Anisotropic Gaussian Kernels (AGK) [GSvdW02] and Half Gaussian Kernels (H-K) [MMD11a]. Fig. 6.5 shows a comparison of best maps and minimum scores for different evaluation measures using our proposed measure λ . We can clearly determine which filter is better than the other filter. Plus, our measure gave a good result with HGK filter. Fig. 6.6 presents the results for 14 measures with their associated scores (bars) according to the hysteresis parameters. In the one hand, we must take into account the obtained edge map, and on the other hand the measure score. Generally, the optimal edge map for FoM , $SFoM$, f_2d_6 , Ψ and λ measures allows to distinct the majority of the desired edges for each contour detection operator (except Sobel), whereas for the other assessments, contours are too disturbed by undesirable points or distinguished with high difficulty (especially Ψ which does not penalizes enough FNs). Note that $SFoM$ measure does not classify the Sobel algorithm as less efficient. Concerning the experiment with a real image in Fig. 6.7, 8 measures are compared together. For FoM , H , Δ^k and S^k , the ideal edge maps concerning Sobel edge detector are highly corrupted by undesirable contours, the main objects are not recognizable. The other segmentations are also disturbed by undesirable pixels for FoM , H and Δ^k . Moreover, the higher score for Δ^k (AGK) does not represent the more disturbed map. Ultimately, using λ , the essential structures are visible in the optimal contour map for each edge detector (objects are easily recognizable). Moreover, contrary to H , FoM , d_4 , Δ^k and S^k measures, the scores of λ are coherent, in relation to the obtained segmentations (Sobel and H-K results).

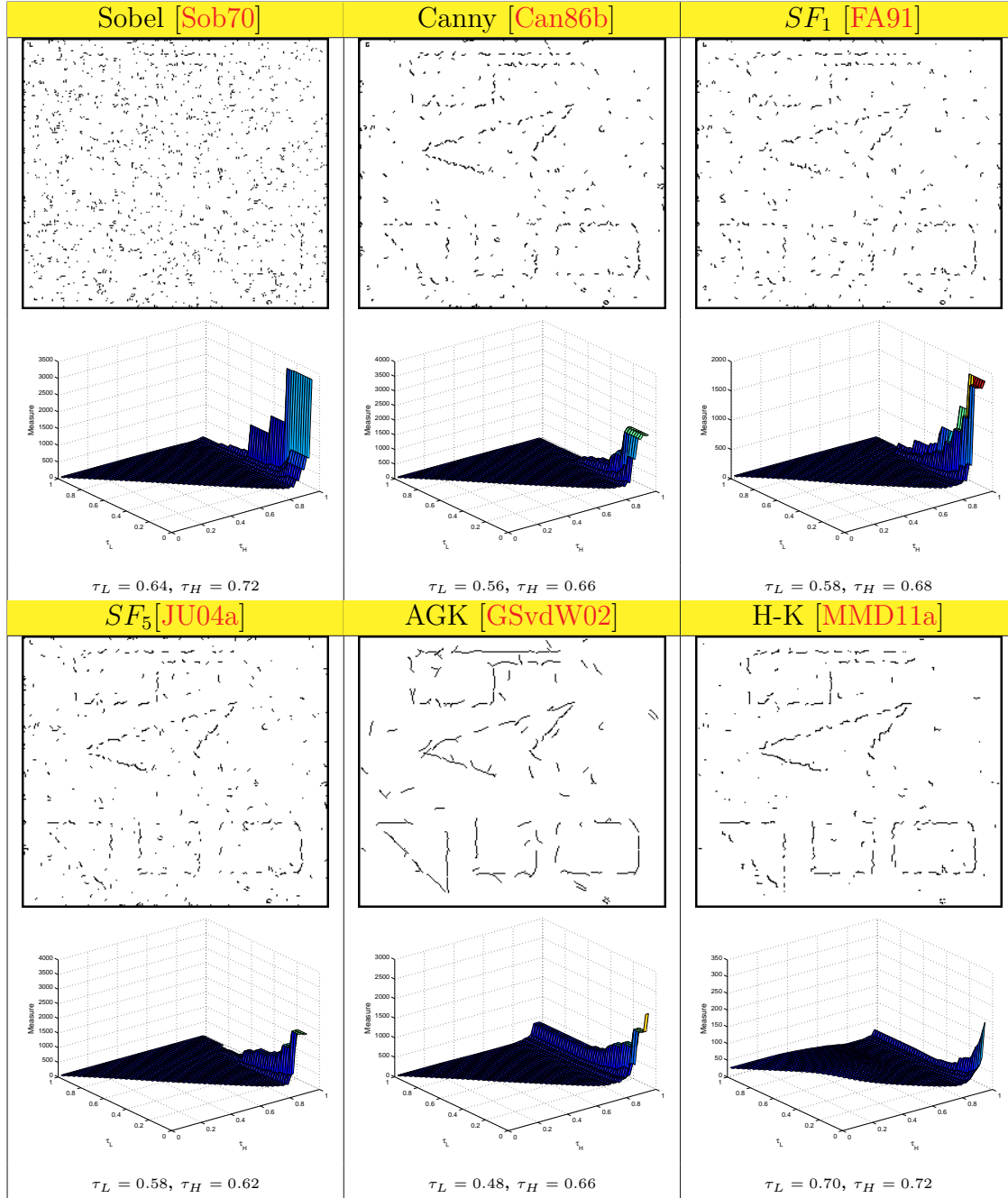


Figure 6.5: Synthetic image SNR= 3.3 dB. (The original image is available in Fig. 5.8 (b) and the ground truth as shown in 5.8(c)). Comparison of best maps and minimum scores for different evaluation measures using our proposed measure λ .

Measure	Sobel [Sob70]	Canny [Can86a]	SF_1 [FA91]	SF_5 [JU04a]	AGK [GSvdW02]	H-K [MMD11a]	
F_α^*	 $\tau_L = 0.44, \tau_H = 0.82$	 $\tau_L = 0.40, \tau_H = 0.72$	 $\tau_L = 0.42, \tau_H = 0.76$	 $\tau_L = 0.48, \tau_H = 0.68$	 $\tau_L = 0.46, \tau_H = 0.68$	 $\tau_L = 0.60, \tau_H = 0.80$	
χ^{2*}	 $\tau_L = 0.44, \tau_H = 0.82$	 $\tau_L = 0.40, \tau_H = 0.72$	 $\tau_L = 0.42, \tau_H = 0.76$	 $\tau_L = 0.48, \tau_H = 0.68$	 $\tau_L = 0.46, \tau_H = 0.68$	 $\tau_L = 0.60, \tau_H = 0.80$	
f_2d_6	 $\tau_L = 0.50, \tau_H = 0.86$	 $\tau_L = 0.34, \tau_H = 0.80$	 $\tau_L = 0.32, \tau_H = 0.82$	 $\tau_L = 0.36, \tau_H = 0.78$	 $\tau_L = 0.44, \tau_H = 0.70$	 $\tau_L = 0.50, \tau_H = 0.80$	
H	 $\tau_L = 0.00, \tau_H = 0.86$	 $\tau_L = 0.20, \tau_H = 0.86$	 $\tau_L = 0.26, \tau_H = 0.86$	 $\tau_L = 0.34, \tau_H = 0.80$	 $\tau_L = 0.50, \tau_H = 0.74$	 $\tau_L = 0.70, \tau_H = 0.80$	
$H_{15\%}$	 $\tau_L = 0.84, \tau_H = 0.84$	 $\tau_L = 0.72, \tau_H = 0.72$	 $\tau_L = 0.75, \tau_H = 0.76$	 $\tau_L = 0.70, \tau_H = 0.70$	 $\tau_L = 0.64, \tau_H = 0.64$	 $\tau_L = 0.74, \tau_H = 0.74$	
FoM	 $\tau_L = 0.40, \tau_H = 0.86$	 $\tau_L = 0.36, \tau_H = 0.80$	 $\tau_L = 0.36, \tau_H = 0.82$	 $\tau_L = 0.36, \tau_H = 0.82$	 $\tau_L = 0.40, \tau_H = 0.68$	 $\tau_L = 0.56, \tau_H = 0.80$	
F	 $\tau_L = 0.72, \tau_H = 0.72$	 $\tau_L = 0.66, \tau_H = 0.66$	 $\tau_L = 0.68, \tau_H = 0.68$	 $\tau_L = 0.62, \tau_H = 0.62$	 $\tau_L = 0.60, \tau_H = 0.60$	 $\tau_L = 0.72, \tau_H = 0.72$	
d_4	 $\tau_L = 0.36, \tau_H = 0.84$	 $\tau_L = 0.40, \tau_H = 0.72$	 $\tau_L = 0.36, \tau_H = 0.76$	 $\tau_L = 0.42, \tau_H = 0.70$	 $\tau_L = 0.60, \tau_H = 0.60$	 $\tau_L = 0.60, \tau_H = 0.80$	
$SFoM$	 $\tau_L = 0.70, \tau_H = 0.72$	 $\tau_L = 0.46, \tau_H = 0.72$	 $\tau_L = 0.36, \tau_H = 0.76$	 $\tau_L = 0.48, \tau_H = 0.68$	 $\tau_L = 0.40, \tau_H = 0.68$	 $\tau_L = 0.56, \tau_H = 0.80$	
$MFoM$	 $\tau_L = 0.70, \tau_H = 0.72$	 $\tau_L = 0.56, \tau_H = 0.62$	 $\tau_L = 0.60, \tau_H = 0.62$	 $\tau_L = 0.58, \tau_H = 0.58$	 $\tau_L = 0.56, \tau_H = 0.56$	 $\tau_L = 0.68, \tau_H = 0.68$	
$S_{k=2}^k$	 $\tau_L = 0.82, \tau_H = 0.82$	 $\tau_L = 0.72, \tau_H = 0.72$	 $\tau_L = 0.72, \tau_H = 0.76$	 $\tau_L = 0.70, \tau_H = 0.70$	 $\tau_L = 0.44, \tau_H = 0.70$	 $\tau_L = 0.52, \tau_H = 0.80$	
Δ^k	 $\tau_L = 0.84, \tau_H = 0.86$	 $\tau_L = 0.20, \tau_H = 0.86$	 $\tau_L = 0.30, \tau_H = 0.86$	 $\tau_L = 0.32, \tau_H = 0.86$	 $\tau_L = 0.50, \tau_H = 0.74$	 $\tau_L = 0.70, \tau_H = 0.80$	
Ψ	 $\tau_L = 0.84, \tau_H = 0.84$	 $\tau_L = 0.72, \tau_H = 0.72$	 $\tau_L = 0.76, \tau_H = 0.76$	 $\tau_L = 0.72, \tau_H = 0.72$	 $\tau_L = 0.66, \tau_H = 0.66$	 $\tau_L = 0.76, \tau_H = 0.76$	
λ	 $\tau_L = 0.64, \tau_H = 0.72$	 $\tau_L = 0.56, \tau_H = 0.66$	 $\tau_L = 0.58, \tau_H = 0.68$	 $\tau_L = 0.58, \tau_H = 0.62$	 $\tau_L = 0.48, \tau_H = 0.66$	 $\tau_L = 0.70, \tau_H = 0.72$	

Figure 6.6: Comparison of best maps and minimum scores for different evaluation measures. The bars legend is presented in Fig. 6.7. G_t and original image are available in Fig. 6.3.

Legend for the bars in Figs. 6.5 and 6.6.

The standard deviation for Canny, SFs, AGK and HK is equal to 1.

In Fig.6.6, the contour images and scores of Canny are similar to SF_7 .



Measure	Sobel [Sob70]	Canny [Can86a]	SF_5 [JU04a]	AGK [GSvdW02]	H-K [MMD11a]	
F^*_α	$\tau_L = 0.34$ $\tau_H = 0.16$ 	$\tau_L = 0.20$ $\tau_H = 0.12$ 	$\tau_L = 0.22$ $\tau_H = 0.14$ 	$\tau_L = 0.16$ $\tau_H = 0.10$ 	$\tau_L = 0.24$ $\tau_H = 0.16$ 	
H	$\tau_L = 0.38$ $\tau_H = 0.0$ 	$\tau_L = 0.26$ $\tau_H = 0.06$ 	$\tau_L = 0.5866$ $\tau_H = 0.08$ 	$\tau_L = 0.26$ $\tau_H = 0.0$ 	$\tau_L = 0.24$ $\tau_H = 0.0$ 	
FoM	$\tau_L = 0.34$ $\tau_H = 0.10$ 	$\tau_L = 0.20$ $\tau_H = 0.08$ 	$\tau_L = 0.22$ $\tau_H = 0.08$ 	$\tau_L = 0.12$ $\tau_H = 0.08$ 	$\tau_L = 0.20$ $\tau_H = 0.10$ 	
d_4	$\tau_L = 0.28$ $\tau_H = 0.14$ 	$\tau_L = 0.20$ $\tau_H = 0.10$ 	$\tau_L = 0.20$ $\tau_H = 0.10$ 	$\tau_L = 0.14$ $\tau_H = 0.08$ 	$\tau_L = 0.20$ $\tau_H = 0.14$ 	
Δ^k	$\tau_L = 0.38$ $\tau_H = 0.06$ 	$\tau_L = 0.26$ $\tau_H = 0.08$ 	$\tau_L = 0.28$ $\tau_H = 0.08$ 	$\tau_L = 0.20$ $\tau_H = 0.06$ 	$\tau_L = 0.24$ $\tau_H = 0.12$ 	
$S^k_{k=2}$	$\tau_L = 0.38$ $\tau_H = 0.06$ 	$\tau_L = 0.18$ $\tau_H = 0.18$ 	$\tau_L = 0.18$ $\tau_H = 0.18$ 	$\tau_L = 0.14$ $\tau_H = 0.14$ 	$\tau_L = 0.18$ $\tau_H = 0.18$ 	
Ψ	$\tau_L = 0.28$ $\tau_H = 0.28$ 	$\tau_L = 0.18$ $\tau_H = 0.18$ 	$\tau_L = 0.18$ $\tau_H = 0.18$ 	$\tau_L = 0.14$ $\tau_H = 0.14$ 	$\tau_L = 0.18$ $\tau_H = 0.18$ 	
λ	$\tau_L = 0.24$ $\tau_H = 0.24$ 	$\tau_L = 0.16$ $\tau_H = 0.16$ 	$\tau_L = 0.16$ $\tau_H = 0.16$ 	$\tau_L = 0.12$ $\tau_H = 0.12$ 	$\tau_L = 0.18$ $\tau_H = 0.18$ 	

Figure 6.7: Comparison of best maps and minimum scores for different evaluation measures. G_t and the original real image are presented in Fig. 6.3.

6.4 Summary

This chapter, presents a new supervised edge detection assessment method λ which enables to assess a contour map in an objective way. Based on the theory of the dissimilarity evaluation measures, the objective evaluation allows to evaluate 1st-order edge detectors. Indeed, the segmentation which obtains the minimum score of a measure is considered as the best one. Theory and experiments prove that the minimum score of the new dissimilarity measure λ corresponds to the best edge quality map evaluations, which is similarly closer to the ground truth, compared to the other methods. On the one hand, this new measure takes into account the distances of false positive points, in the other hand, it considers the distance of false negative points tuned by a weight. This methodology forms our contribution and resulted in the publication [AMM17b] This weight depends on the number of false negative points: the more it is elevated, the more the segmentation is penalized. Thus, this enables to obtain objectively an edge map containing the main structures, similar to the ground truth, concerning a reliable edge detector. Finally, the computation of the minimum score of a measure does not require tuning parameters, which represents a huge advantage.

ORIENTED HALF KERNELS FOR CORNER DETECTION

This chapter describes our proposed method using oriented half kernels for corner detection.

Contents

7.1	Introduction	112
7.1.1	Corner detection techniques	112
7.1.2	Oriented kernels and edges directions	114
7.2	A new method of corner extraction	115
7.2.1	Oriented filter of grayscale homogeneity	116
7.2.2	A combination of homogeneity and edge strength	117
7.2.3	Angle selection and corner extraction	117
7.3	Experiments and evaluation	118
7.4	Summary	125

7.1 Introduction

Corners and junctions play an important role in many image analysis applications. However, these features extracted by the majority of the proposed algorithms in the literature do not correspond to the exact position of the corners. In this chapter, an approach for corner detection based on the combination of different asymmetric kernels is proposed. Informations captured by the directional kernels enable to describe precisely all the grayscale variations and the directions of the crossing edges around the considered pixel. Comparing to other corner detection algorithms on synthetic and real images, our proposed approach remains more stable and robust to noise than the comparative methods.

In digital images, corners and junctions constitute important landmarks because they get affluent information like delimitation of objects. A pixel is considered as a corner or a junction when two or more edges meet each other and it refers to the point where several different contour orientations exist. These orientations could be estimated using structure tensors, voting tensors or oriented kernels. In this chapter, a non exhaustive review of corner detection methods is presented. Then, oriented kernels are described. Finally, a new corner detection method based only on oriented kernels is described.

7.1.1 Corner detection techniques

In the literature, several approaches have been developed to detect corners and junctions: (i) involving contour chains, (ii) using templates or, (iii) by image filtering techniques.

Traditional contour based corners methods focus on the processing of binary edges, by searching points having curvature in contour chains or in line segment intersections [XDG14, SZ13]. This approach might be insufficient for applications in natural images because the accuracy of corner localization relies on edge detection performance (filtering, threshold(s)).

Another way to extract these feature points involves templates by compared all pixel values with the of center pixel(s) of the shape template. In the well known SUSAN (Smallest Univalued Segment Assimilating Nucleus [SB97]), a corner is extracted in comparing every pixel inside a circular mask with the central pixel. On the contrary, for the FAST (Features from Accelerated Segment Test [RPD10]) method, a point is considered as a corner only if there are several pixels in the path of a circular template which are higher or smaller than the value determined by the central pixel. These corner detection methods do not require any spatial derivatives; however, they do not lead directly to the position of the corner (i.e. the pixel having several local orientations as in Fig. 7.1 (d) and (e)). Concerning image

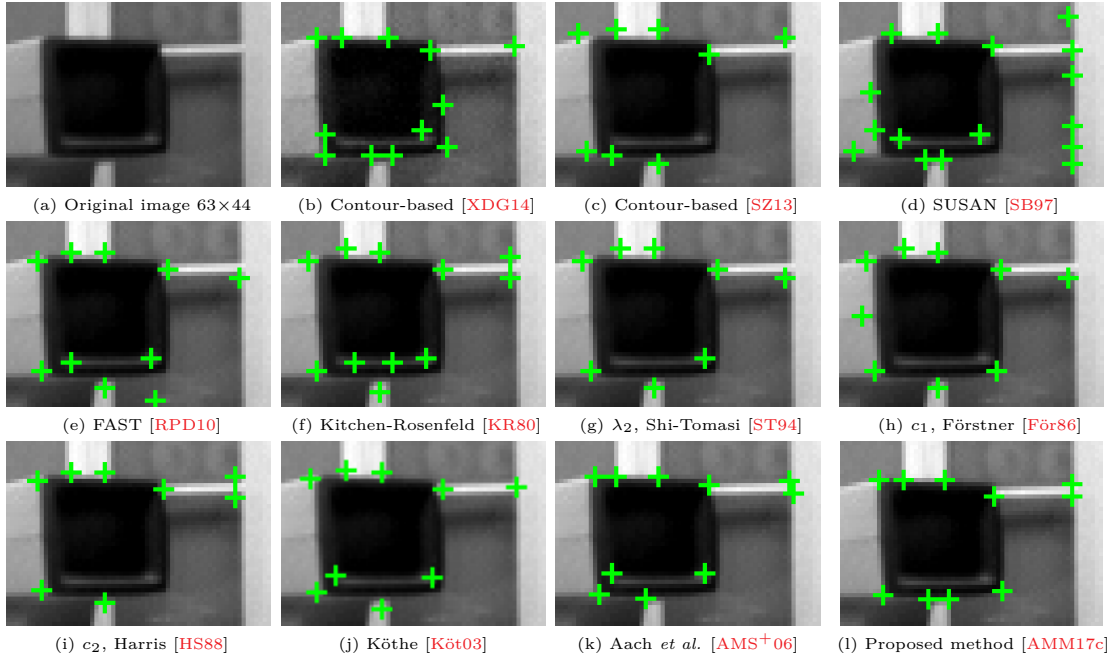


Figure 7.1: Corner detected involving several methods. The standard deviation used for the image derivatives is the same as for the structure tensor J_ρ : $\sigma = \rho = 1$. For the developed method: $\sigma_\eta = 1$ and $\sigma_\xi = 3$, $L = 3$ and $P = 5$.

filtering, the feature detectors operate directly on image intensities. Corners are defined by the combination of the gradient magnitude and points having maximum curvature of the image surface. A pioneer work in this domain remains Kitchen-Rosenfeld algorithm which involves first and second order image derivatives in the corneriness computation [KR80]. Methods using the second derivative are more sensitive to noise then introduces false. Indeed, the computation of only the first image derivatives informs on the local structure at a pixel by examining also the data in the neighborhood. Moreover, integrating the gradient information in the neighborhood of the pixel (i.e. the correlation) brings indications about whether the pixel must be considered as an edge or as a corner. This selection is given by involving a 2×2 symmetrical structure tensor. The derivation of a scalar image I is called the image gradient and is noted by $\nabla I = (I_x, I_y)^t$ in which I_x and I_y represent the image derivatives in x and y directions respectively (usually calculated by means of Gaussian derivative filters with a standard deviation of $\sigma \in \mathbb{R}_+$, see Fig. 7.2(a)). Involving a smoothing Gaussian kernel G_ρ of standard deviation ρ , the first-order structure tensor J_ρ is given by $J_\rho(\nabla I) = G_\rho * \nabla I \cdot \nabla I^t$. The scale of the neighborhood information is given by G_ρ . When J_ρ possesses two positive eigenvalues, then the pixel is considered having at least two distinguished orientations, therefore a corner or a junction. This isotropic corneriness measurement is given by the second eigenvalue denoted λ_2 [ST94, BWBM06] or could be estimated by $c_1 = \frac{\det(J_\rho)}{\text{tr}(J_\rho) + k}$ [För86] or $c_2 = \det(J_\rho) - k \cdot \text{tr}(J_\rho)$ [HS88], with $k > 0$. Feature detection using the linear structure tensor J_ρ is insufficient in the presence of more than one dominant direction. Depending on its smoothing parameter ρ , this tensor representation is robust under noise, but generally the localization of the detected corner

misses precision. Indeed, according to the scale ρ and the image derivatives estimated by convolution with Gaussian kernels, the detected location of a corner tends to shift as ρ increases [DG93], as illustrated Figs. 7.1(g), (h) and (i). To bypass this weakness, several solutions have been proposed in the literature as nonlinear structure tensors [BWBM06] or tensor voting [MPB⁺12]. In [Köt03], the cornerness measurement λ_2 is propagated using oriented filters that are shaped like hour-glass instead of the Gaussian mask for data-adaptive smoothing at the crossing edge position.

Finally, in [AMS⁺06], double local orientations for corners and junctions are extracted by: (1) involving J_ρ to detect regions containing double orientations and (2) the computation and combinations of the eigenvalues of a 3×3 tensor. This method is more precise than previous mentioned approaches, as shown in Fig. 7.1(k).

In this work, corners and junctions are directly extracted involving only a combination of asymmetric oriented kernels.

7.1.2 Oriented kernels and edges directions

Oriented filters were designed to capture multidirectional gray intensity variations [FA91, JU04b]. Indeed, they consist in finding the orientation where the derivative corresponds to the maximum response. The concept was generalized in [Per92] by decomposing a given filter kernel optimally in a set of basis filters which approximates an Anisotropic Gaussian Kernel (AGK): Anisotropic Gaussian kernel for (smoothing) Eq. 7.1:

$$G_{(x,y,\sigma_\eta,\sigma_\xi)} = \frac{1}{2\pi\sigma_\eta\sigma_\xi} \cdot e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_\eta^2} + \frac{y^2}{\sigma_\xi^2}\right)}. \quad (7.1)$$

$$\begin{aligned} AGK_{(x,y,\sigma_\eta,\sigma_\xi)} &= -\frac{1}{2} \cdot \left(\frac{2x}{\sigma^2}\right) \cdot G_{(x,y,\sigma_\eta,\sigma_\xi)} = \frac{-x}{\sigma^2} \cdot G_{(x,y,\sigma_\eta,\sigma_\xi)} \\ &= \frac{-x}{2\pi\sigma_\eta^3\sigma_\xi} \cdot e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_\eta^2} + \frac{y^2}{\sigma_\xi^2}\right)} \end{aligned} \quad (7.2)$$

where $(x,y) \in \mathbb{R}^2$ represent the pixel coordinates, σ_η and σ_ξ are referred to the Gaussian scale and to the anisotropic factor respectively (Fig. 7.2(b)). The AGK can be oriented [SZ13] but, possesses a common shortcoming, as a matter of fact, only one π -periodic orientation is extracted efficiently [Per92], so the impossibility of these filters to estimate in a relevant way several coexisting orientations at the same pixel (see Fig. 7.1 (c)).

Contrary to the templates mentioned above, wedge [SF96][MDA07] or asymmetric oriented filters [MMD11a][MDCL⁺07] sound better suited for a purpose like multiple edge directions detection or modelling a template. Thus, corner analysis requires finding maxima in filter

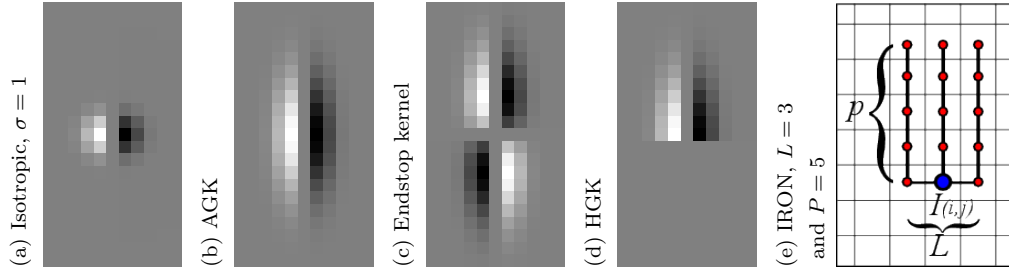


Figure 7.2: Different discretized 2D derivative Gaussian kernels and representation of the IRON filter. (b), (c) and (d) with $\sigma_\eta = 1$ and $\sigma_\xi = 3$. The Anisotropic endstop filter is equivalent to the derivative of the AGK along the Y direction.

responses when these 2π -periodic filters are successively steered in different directions. The equation of the anisotropic edge detector based on Half Gaussian Kernels (HGK) derivative is given by:

$$\text{HGK}_{\sigma_\eta, \sigma_\xi}(x, y) = H(y) \cdot \text{AGK}_{\sigma_\eta, \sigma_\xi}(x, y). \quad (7.3)$$

where H corresponds to the Heaviside function, illustrated in Fig. 7.2(d). By rotating the image [MMD11a], maxima of the filter responses indicate the directions of the edges from 0 to 2π [MMD11a, MM14a]. In addition, HGK and wedge steerable filters responses characterize easily corners and junctions. Nevertheless, inside homogeneous/noisy regions, due to the , isophotes i.e. curves of the image surface of constant intensity, these directions become unpredictable. Perona shows in [Per92] that the combination of the endstop kernel (see Fig.7.2 (c)) with the HGK enables a junction and corner characterization. Indeed, the response of the combination is null along a straight contour whereas the response has maxima along the directions of the contours forming a junction or a corner (illustrated in Fig. 7.3). In order to avoid false corners or junctions, another solution proposed in [MM14a] is to align the directions of the HGK when the gradient value is weak. However, near edges, the gradient generated by the HGK remains not so weak and disturbs the corner detection by creating a halo of acute angles around the contours (note that this remains the same problem using the endstop kernel). Fig. 7.3 (bottom) and Fig. 7.4 (c) illustrate this phenomenon, corners are correctly localized, however, close to the edge, HGK creates an angle which can be considered as a corner.

7.2 A new method of corner extraction

The main idea of this new approach is to combine the HGK with an asymmetric filter computing the homogeneity along edges. On one hand, the maxima responses of the HGK indicate the directions (2π periodic) of the edges. On the other hand, the oriented variance determines if the directions of the maxima of the HGK corresponds to edges or other types of pixels (texture, homogeneous region etc.).

7.2.1 Oriented filter of grayscale homogeneity

The asymmetric IRON (Isotropic and Recursive Oriented Network) filter estimates the homogeneity in multiple local directions [MDCL⁺07]. This filter consists of a network of several parallel lines in which a homogeneity is computed and enables an estimation of edge directions modulo 2π . The variance for a pixel located at (x, y) on the network is computed by:

$$\text{IRON}(x, y) = \frac{1}{L} \sum_{j=0}^L \left(\frac{1}{P} \sum_{i=0}^P (I(i, j)^2) - \left(\frac{1}{P} \sum_{i=0}^P I(i, j) \right)^2 \right). \quad (7.4)$$

Here, L represents the number of lines where the variance is computed and P the number of points per line. Fig. 7.2(e) represents an example of an asymmetric IRON filter. Computationally, the rotation of the image is applied at some discretized orientations from 0 to 2π before applying the IRON filters. Some examples of IRON filter signals are available in Fig. 7.3 fifth column, values of the IRON are close to 0 in the edges directions and it is shown in [MDCL⁺07] that the detections of edges directions stay precise in the presence of noise.

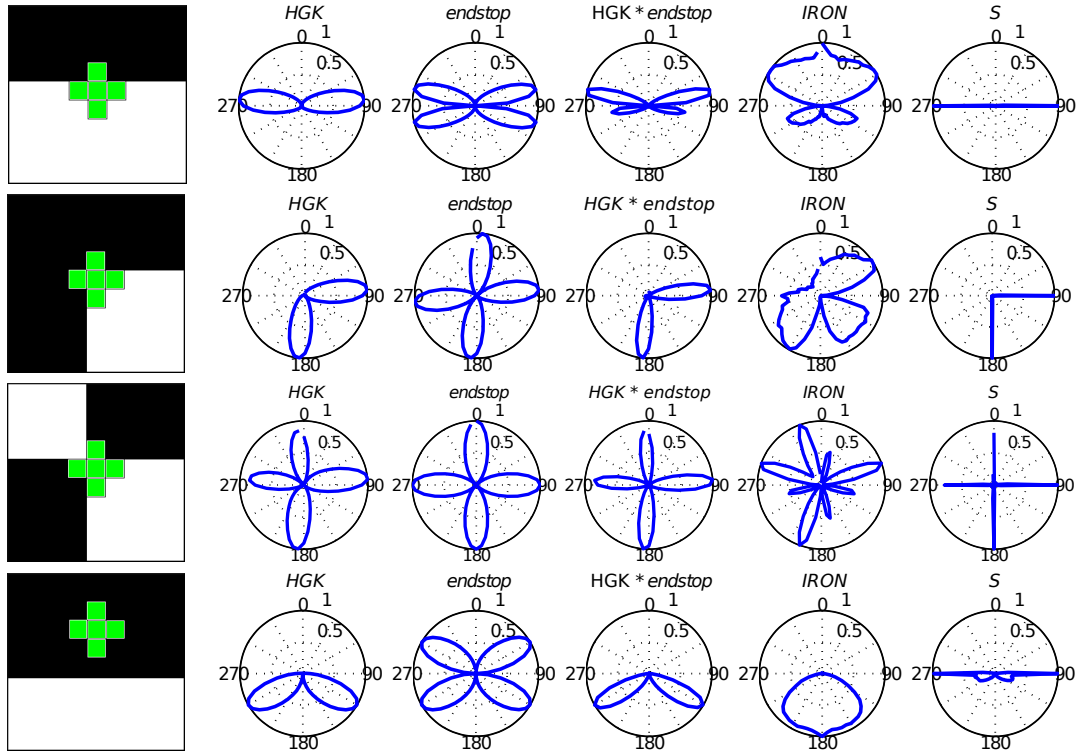


Figure 7.3: Modulus of the energy of the different oriented kernels and their combinations (in degrees and normalized signals).

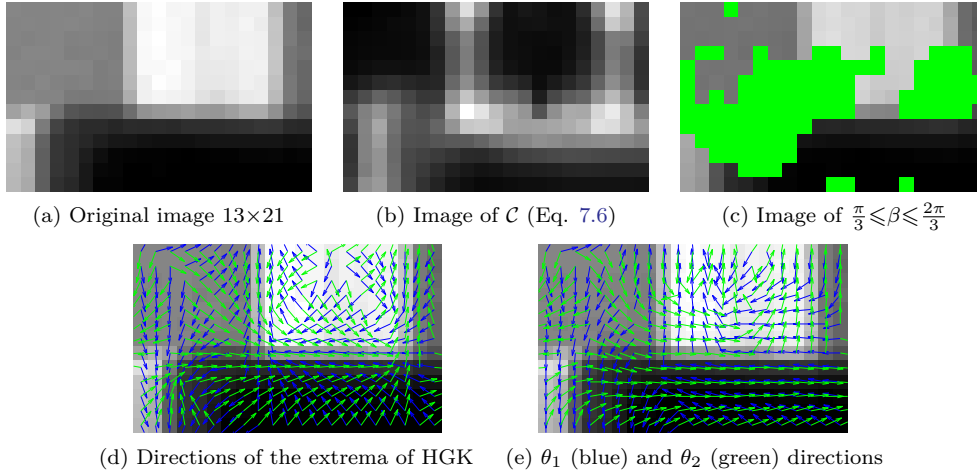


Figure 7.4: Direction field. (a) Cropped image of Fig. 7.1(a).

7.2.2 A combination of homogeneity and edge strength

The HGK and the asymmetric IRON are 1-side kernels, so they are steered in 2π directions. Moreover, the response of the HGK corresponds to maxima along the directions of the edges while the response of the IRON has minima along these same directions. Hence, the combination at the orientation $\theta \in [0; 2\pi[$ between the HGK and IRON is straightforward:

$$\mathcal{S}(x, y, \theta) = \frac{\text{HGK}(x, y, \theta)}{\varepsilon + \text{IRON}(x, y, \theta)}, \quad \text{with } \varepsilon \in \mathbb{R}_*^+, \quad (7.5)$$

where $\text{HGK}(x, y, \theta)$ and $\text{IRON}(x, y, \theta)$ represent, respectively the HGK and IRON responses in a rotated image of angle θ . Finally, ε corresponds to a small constant avoiding a division by 0, generally is taken as $\varepsilon = 0.001$. The corner detection is equivalent to analyze the resulting signal \mathcal{S} for each pixel. The polar curves in Fig. 7.3 indicate the modulus of the different kernels responses. Moreover, the extrema of \mathcal{S} indicate the precise directions of the edges for contour and corner points. As illustrated in Fig. 7.4 (d), at a distance between 1 and $2\sigma_\eta$ pixels, the extrema of \mathcal{S} correspond to directions which are parallel to the edge directions. Finally, in order to compute the variance and the oriented derivative on the same neighborhood, the spacial influence of the IRON filter is inserted in the support of the HGK, i.e. $3P \leq \sigma_\xi$ and $3L \leq \sigma_\eta$.

7.2.3 Angle selection and corner extraction

The IRON energy is always positive while the HGK filter corresponds to an oriented derivative, so its responses are either positive, or negative. Consequently, the signal \mathcal{S}

possesses positive/negative values when HGK is positive/negative (see Fig. 7.5). To obtain the cornerness measure \mathcal{C} as in Fig. 7.4 (b), the global extrema of \mathcal{S} are combined:

$$\begin{cases} \mathcal{C}(x, y) = \max_{\theta \in [0, 2\pi[} \mathcal{S}(x, y, \theta) - \min_{\theta \in [0, 2\pi[} \mathcal{S}(x, y, \theta) \\ \theta_1(x, y) = \arg \max_{\theta \in [0, 2\pi[} (\mathcal{S}(x, y, \theta)) \\ \theta_2(x, y) = \arg \min_{\theta \in [0, 2\pi[} (\mathcal{S}(x, y, \theta)) \\ \beta(x, y) = \begin{cases} |\theta_1(x, y) - \theta_2(x, y)|, & \text{if } |\theta_1(x, y) - \theta_2(x, y)| \leq \pi \\ 2\pi - |\theta_1(x, y) - \theta_2(x, y)| & \text{elsewhere.} \end{cases} \end{cases} \quad (7.6)$$

Once \mathcal{C} , θ_1 and θ_2 have been obtained, the corners can be easily extracted in two steps: (i) by selecting pixels where β , the angle formed by θ_1 and θ_2 , corresponds to a desired angular sector followed by (ii) thresholding the local maxima of \mathcal{C} . Finally, for a pixel belonging to a straight contour, the value of \mathcal{C} is high (see Fig. 7.4 (b)) while β corresponds to an open angle $\approx \pi$ ((illustrated in Fig. 7.5 and 7.4(c)); so it is not considered as a corner point.

7.3 Experiments and evaluation

The experiments are carried out on synthetic images and real images to compare the corner detectors methods with our proposed corner detection method. First, the 31 "best" corners are extracted from the synthetic image by corner detectors. These corners are composed of acute and obtuse angles. Then, the Root-Mean-Square Error (RMSE) is

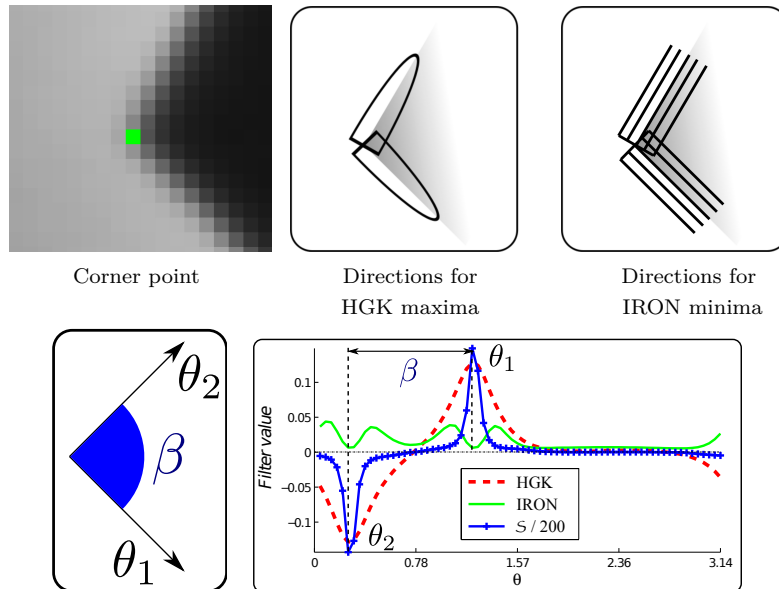


Figure 7.5: The minimum and maximum of the signal \mathcal{S} corresponds to the two directions of the edges and β to the angular sector between the θ_1 and θ_2 directions.

computed between the 31 true corners and the extracted features. Considering G_t and D_c the set of true and detected corners respectively:

$$\text{RMSE} = \sqrt{\frac{1}{\text{card}(G_t) + \text{card}(D_c)} \cdot \left(\sum_{p \in D_c} d_{G_t}^2(p) + \sum_{p \in G_t} d_{D_c}^2(p) \right)}.$$

For a pixel $p \in D_c$, $d_{G_t}(p)$ represents the minimal Euclidean distance between p and G_t , whereas if $p \in G_t$, $d_{D_c}(p)$ corresponds to the minimal distance between p and D_c . Thus, ten corner detection approaches are compared in terms of the noise level, which is indicated by the Signal-to-Noise Ratio (SNR), as illustrated in Figs. 7.6 (a), (b), (c) and (d). Another experiment illustrated in Figs. 7.10 (c), (d), (e) and (f), show that our approach is more robust to noise since the RMSE error is always better than the others methods.

The curve in Fig. 7.6 (e) shows the RMSE in function of the SNR when the standard deviation for the Gaussian derivative is the same for all methods excepted FAST ($\sigma_\eta=1$ in our case). Indeed, the standard deviation is the same for all detectors in order to compare them together, even for the contour based methods [XDG14][SZ13]. Our method achieves the best results in term of RMSE for all the noise levels. The shape of the considered filters enables to locate the corners at the correct position even though the noise is strong (see Fig. 7.6(d) with SNR= 4dB). The same evaluation is led by changing the standard deviation for the Gaussian derivative ($\sigma = \sigma_\eta = 2$) for all tested methods. The curve in Fig. 7.6 (f) illustrates the error measures. As pointed out in Section 7.1.1, the scale of isotropic detectors affects the localization of detected corners. Note that the contour based on anisotropic Gaussian kernels is not robust to the detection of acute corners because these kernels delocalize strongly the corner points in the edge detection stage (see [MMD11a]). On the contrary, our proposed method [AMM17c] remains stable (less than 1 pixel RMSE measure compared to $\sigma_\eta=1$ in the previous case). Concerning acute angles, the half kernels are able to select the two directions of the edges and then qualify these pixels as corners. Finally, the performance of the new method is due to the

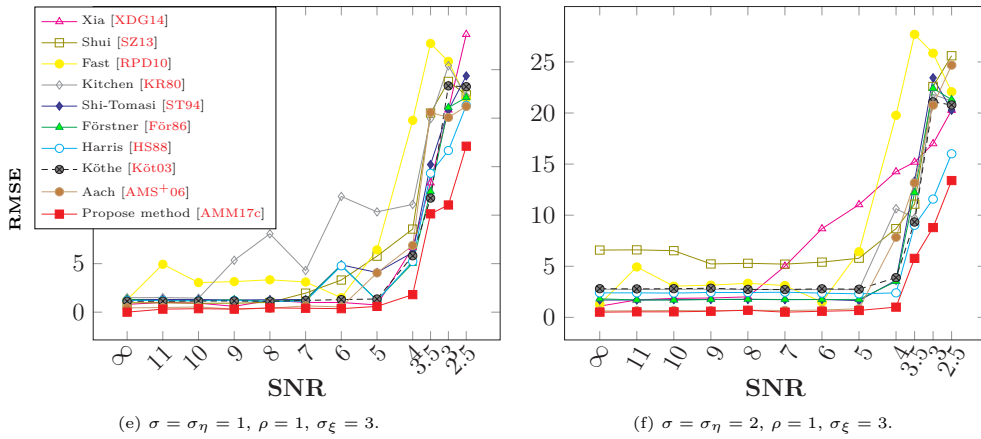


Figure 7.6: Comparison the *RMSE* as a function for the corners error detection between our method with several corner detection methods.

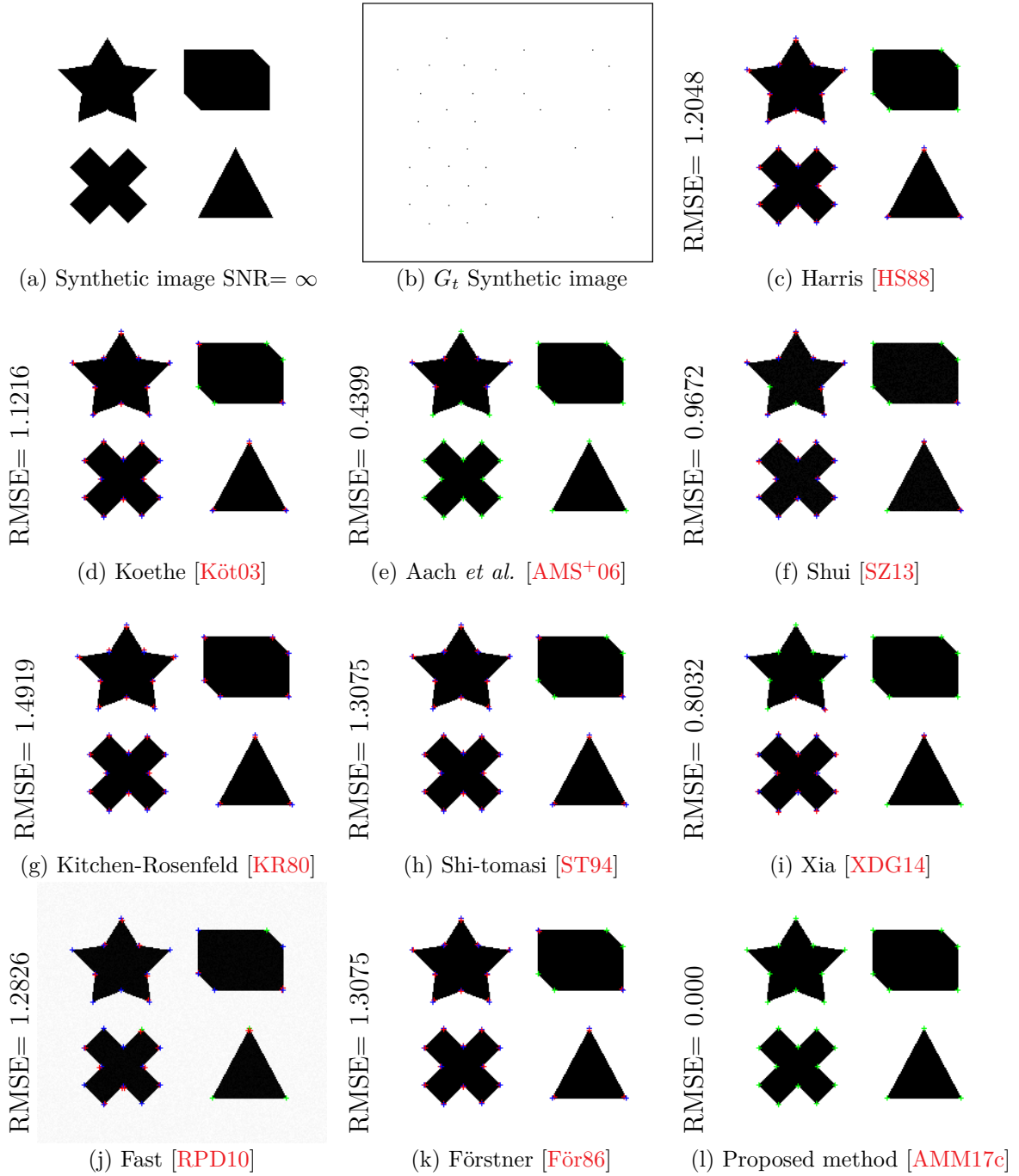


Figure 7.7: Corner detection using template or isotropic structure tensors obtains mistakes or corner displacement. The standard deviation σ for the image derivatives and for the structure tensor J_ρ are the same: $\sigma = \rho = 1$. For (h), the convolution masks a (for example $k = 0.004$). Note: [True corner = +, False corner = +, and Forget corner = +].

HGK combined with the IRON filter which corresponds to thin filters, allowing a precise direction of contours, and thus of corners.

The first real image in Fig. 7.1(a) is composed of thin structures with a blur. Despite

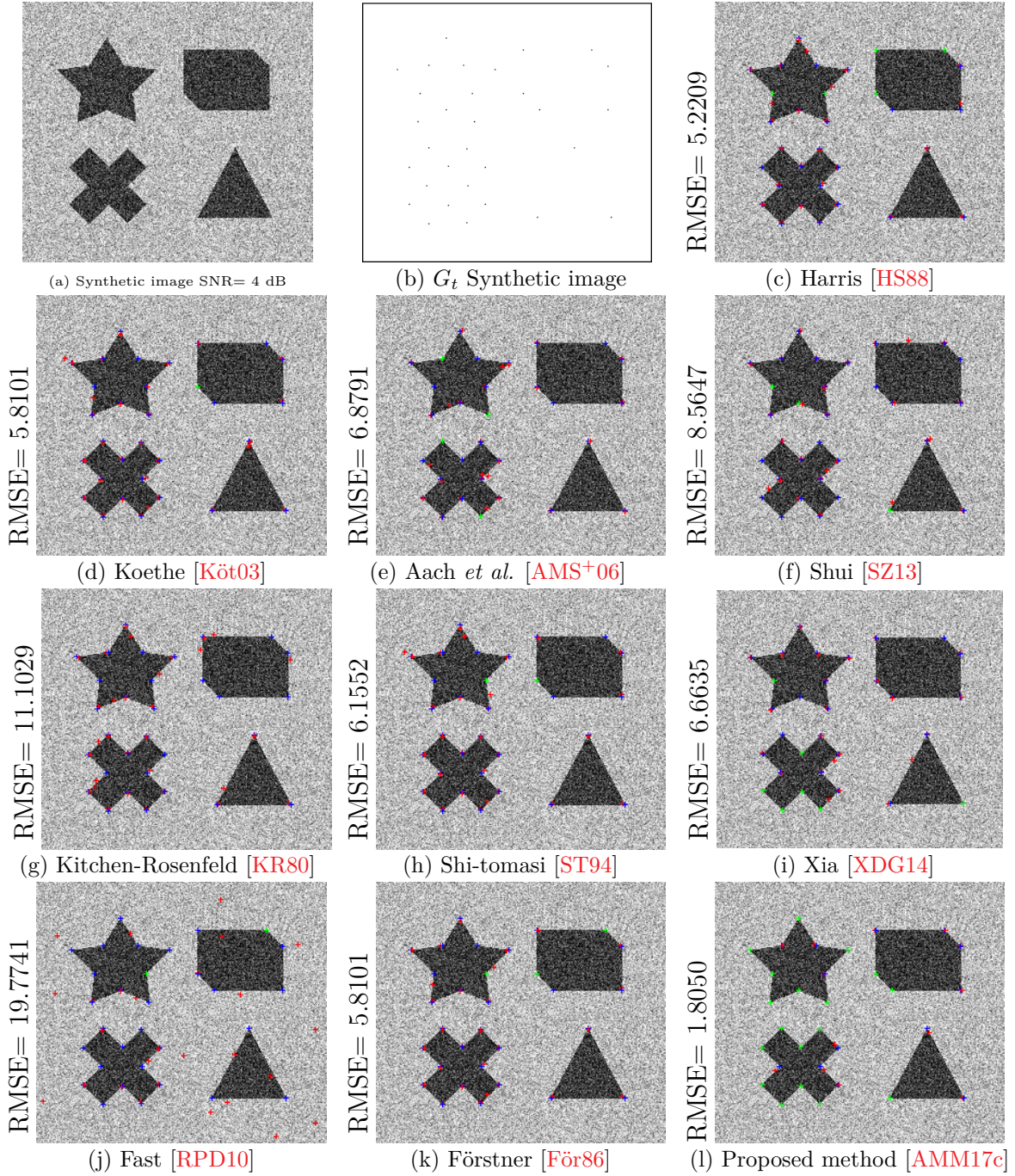


Figure 7.8: Corner detection using template or isotropic structure tensors obtains mistakes or corner displacement. The standard deviation σ for the image derivatives and for the structure tensor J_ρ are the same: $\sigma = \rho = 1$. For (h), the convolution masks a (for example $k = 0.004$). Note: [True corner = +, False corner = +, and Forget corner = +].

that, Fig. 7.1(l) illustrates very clearly that the new corner detection method has better accuracy than the ten other corner detectors. The last experiment presented in this study concerns the 'lab' image in Fig. 7.11(a). This image contains several corners of

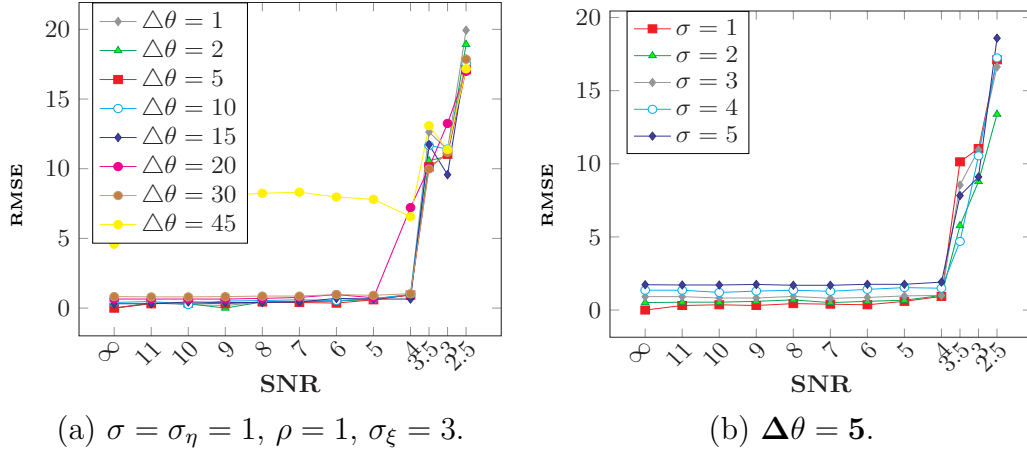


Figure 7.9: Comparison the Root Mean Square Error (RMSE) as a function for the corners error detection (a) our method with different $\Delta\theta$, (b) our method with different σ with $\Delta\theta = 5$.

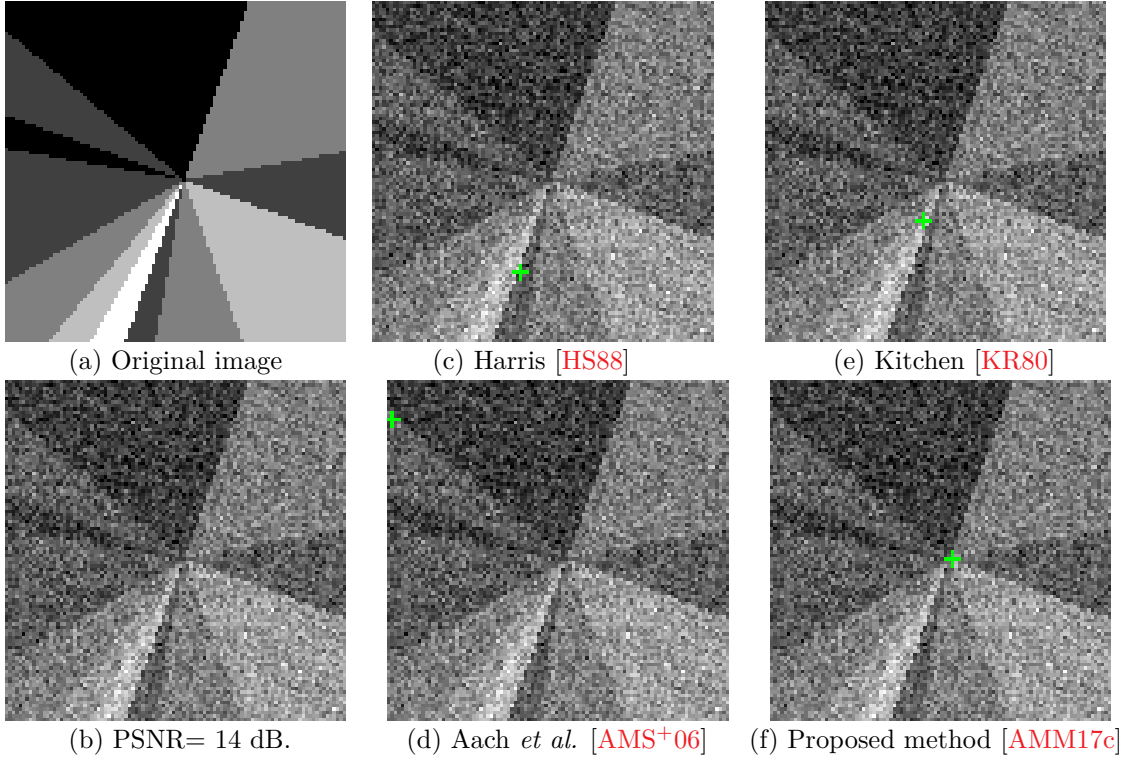


Figure 7.10: RMSE as a function of the noise level. (c)-(f): Corner detected involving several methods with $\sigma_\eta = 1$ et $\sigma_\xi = 3$.

different type of angles and blurred edges. To make the comparison easier, the 280 best interest points are extracted for each algorithm. The contour-based methods [XDG14] and the tensorial approaches [HS88][Köt03][AMS+06] fail to detect the majority of corners of obtuse angle (e.g. carpets on the floor) while they detect a lot of features concerning small objects as in the top right part of the image. On the other hand, Fig. 7.11 (f) shows that the performance of our proposed method [AMM17c] is more efficient to detect

the features. In order to capture acute and obtuse corner, the angular sector belongs to $\left[\frac{\pi}{6}, \frac{5\pi}{6}\right]$. Such an angular sector is enough to detect only desired features. These results using the combination of oriented half kernels and IRON filter illustrate reliable and promising results, even when dealing with blurred images. Other experiments are carried out real image to compare corner detectors. Fig. 7.12 shows that the performance of the our method is more efficient to detect the corners.

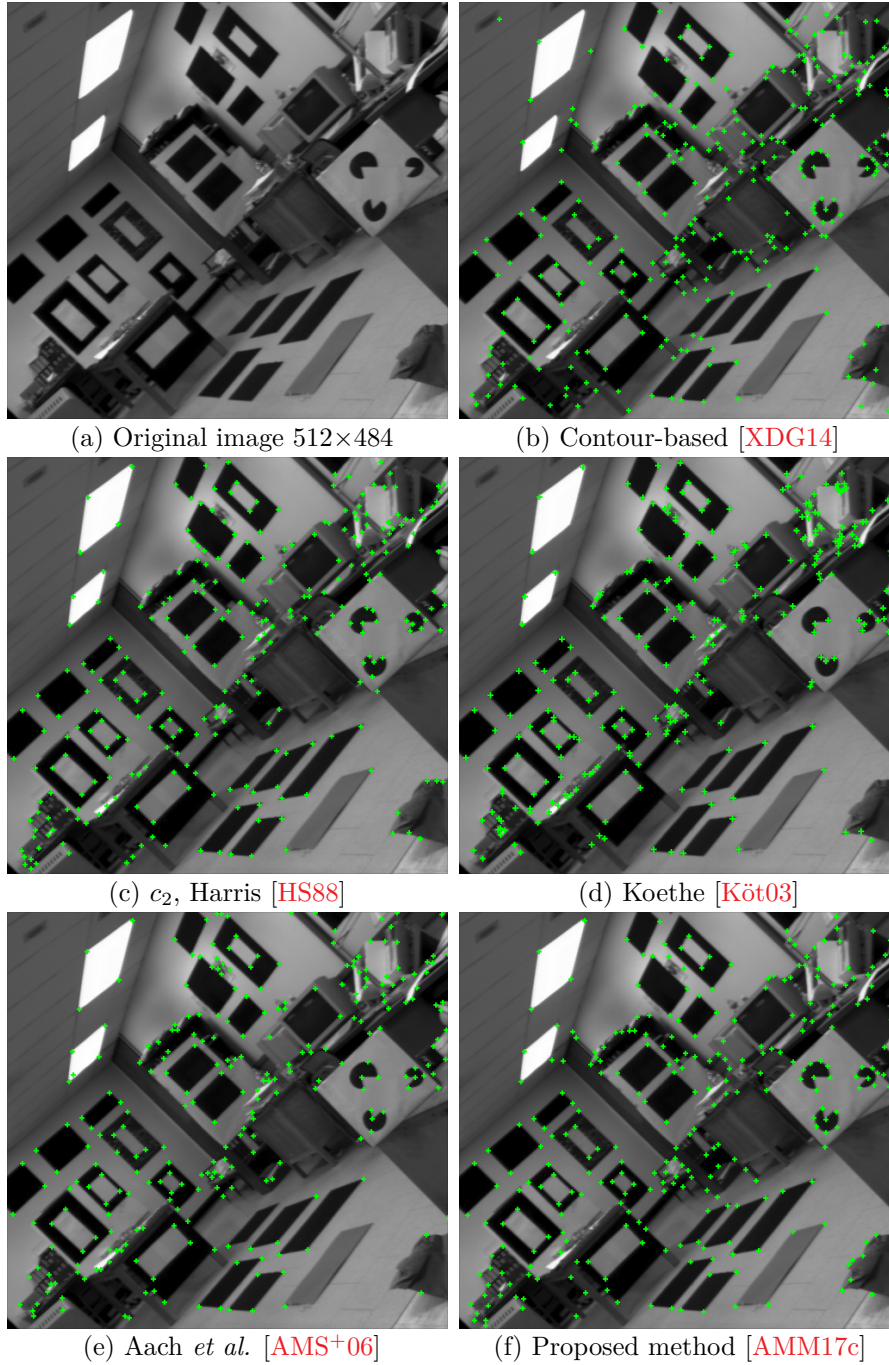


Figure 7.11: Corner detected involving several methods. The standard deviation used for the image derivatives and for the structure tensor J_ρ are: $\sigma = 1.5$ and $\rho = 1$. For the developed method: $\sigma_\eta = 1.5$ and $\sigma_\xi = 5$, $L = 3$ and $P = 5$.

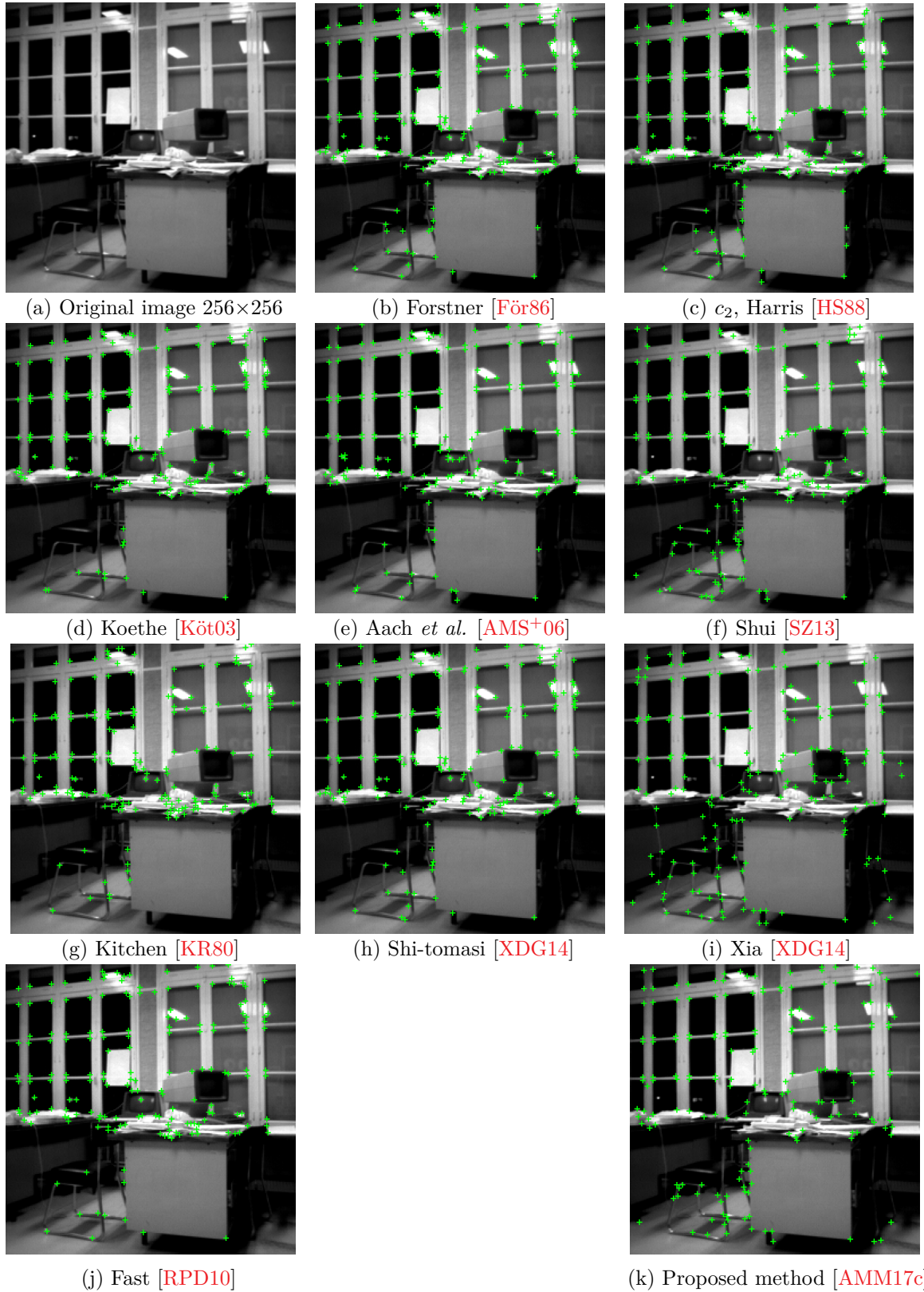


Figure 7.12: Corner detection using template or isotropic structure tensors obtains mistakes or corner displacement. The standard deviation σ for the image derivatives and for the structure tensor J_ρ are the same: $\sigma = \rho = 1$. For (h), the convolution masks a (for example $k = 0.004$)

7.4 Summary

In this chapter an approach for corner detection based on the combination of different asymmetric kernels has been presents. Informations captured by the directional kernels enable to describe precisely all the grayscale variations and the directions of the crossing edges around the considered pixel. This methodology forms our contribution and resulted in the publication [AMM17c]. The half Gaussian kernels (HGK) allow to detect a relevant strength of a corner while the IRON indicates the more homogeneous directions. A such combination enables to remove undesirable corners near contour areas which could be considered as a ideal feature by the HGK. Moreover, due to the shape and the functionality of the considered filters, corners are located at the correct position of the corners, even in the presence of a strong noise. Compared to other corner detection methods on synthetic and real images, our proposed approach remains more stable and robust to blur and noise than the comparative methods.

Part III

DETECTION OF HIDDEN DATA
IN DIGITAL IMAGES

8

OVERVIEW ON STEGANOGRAPHY AND STEGANALYSIS

This chapter focus on digital image steganography and steganalysis methods.

Contents

8.1	Introduction	130
8.2	Steganography and steganalysis through these ages	131
8.3	Cryptography, Watermarking and Steganography	133
8.4	The prisoner's problem (steganography/steganalysis)	137
8.5	Digital Image Steganography	138
8.6	Digital Image Steganalysis	142
8.6.1	Color image Steganalysis	144
8.6.2	The ensemble classifiers	147
8.7	Summary	148

8.1 Introduction

Since long time, steganography techniques are known and widely used [Fri09]. Steganography is a technique to hide a secret message in a digital media and communicate secretly. In modern-day digital steganography works by modifying bits within various digital media files (for example text, image, video, audio) to avoid the detection of the secret message and doing that a manner that it is difficult for an adversary to detect and at the same time difficult for an adversary to remove too. The message remains secret such that it is only known by the sender and receiver and an adversary does not recognize its existence visually. Based on this objective, there are important principles must be used to measure the performance of a given steganography method:

- Quantity of data: relates to more data you can hide, the better the method.
- Hardness of detection: suggests to how easy it is, for anyone to detect the hidden message. Indeed, there is great relationship between quantity of data that can be hidden and how much it is easy for anyone. Increase the hidden data, greater the probability of change the image quality with
- Difficulty of removal: involves the principle that someone intercepting your file should not be able to remove the hidden data easily.

Today, the challenge is to design a digital steganography method which has reliable security of the hidden communication, along with minimum size of the payload and provides good robustness against attacks. On the other hand, detecting steganography methods belongs to the field of steganalysis. This field is constantly evolving its techniques, methods and theories in parallel line with the development of steganography techniques, to detect hidden communication.

In reality, the steganography can be used for legal and illegal purpose, but in recently it has received a great attention from national security seasons.

Eventually, modern steganography emerged with the advent of digital media, personal computers, communication networks, and the Internet.

We carried out a sample search query with keyword **steganography** at scholar.google.fr, combining conference and journals papers from 1990 to 2015 year. Fig. 8.1 shows the yearly count of research articles on the steganography domain founded by Google Scholar. This bring to light, the importance of steganography domain and how the researchers proposed different methods during the period (1990-2015), certainly, the request for using steganography methods increasing bewildering, during 25 last years and this raises several questions about this domain.

8.2 Steganography and steganalysis through these ages

The use of information hiding methods dates back to several millennia. Considered the steganography method is one of the most ancient and famous method. Steganography literally means, "Covered writing" which is derived from the Greek language steganos-graphy, steganos means (covered or secret) and graphy means (writing or drawing).

We can define steganography as the art and science of covert communications between two parties. The secret messages are communicated by hiding them in a cover.

The first known steganography technique was implemented in ancient Greece around 440 B.C. Herodotus [Her96] reports an interesting story of Histaeus the Greek ruler, that decided to send a secret message to his son in law Aristagoras in Miletus, to tell him that it is the time to start a revolt. He shaved the head of a slave, then tattooed a message on the slave scalp, he waited for the growth of hair to disclose the message, and sent the slave on his way to deliver the message as shown in Figure 8.2 (a). Indeed, the message was arrived to his son and the revolt was successful.

Another story reported by Herodotus, from Ancient Greece is to write a secret message on a tablet underlying wood and then cover the tablet with wax as shown in Figure 8.2 (b). The tablet appeared to be blank and unused so they can passed on inspectors. Ancient Romans used to write between the lines of text the invisible inks and to display the hidden messages required heat. Aeneas the Tactician [Tac90] proposed many steganography

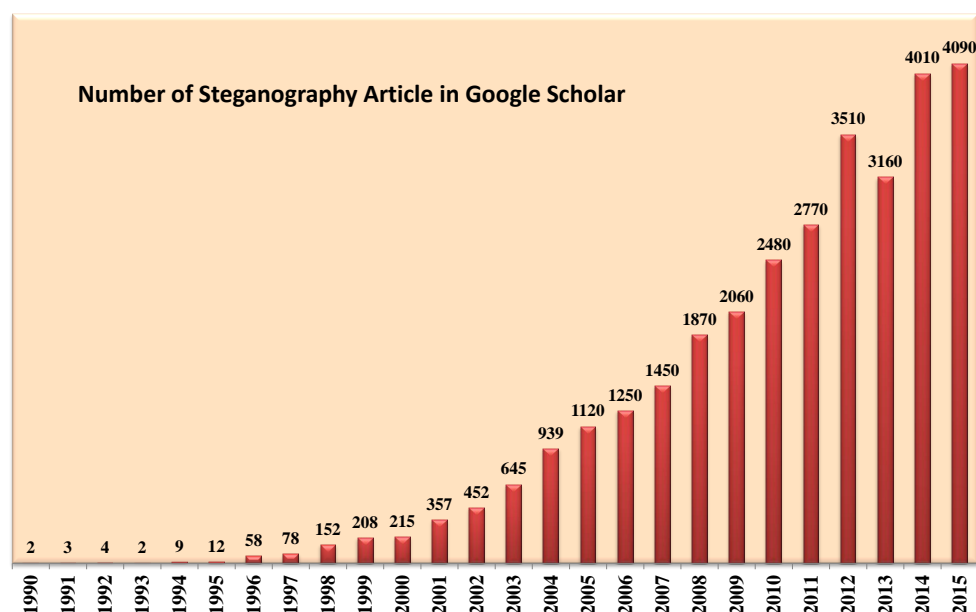


Figure 8.1: The yearly count of research articles on the Google scholar the keywords steganography.



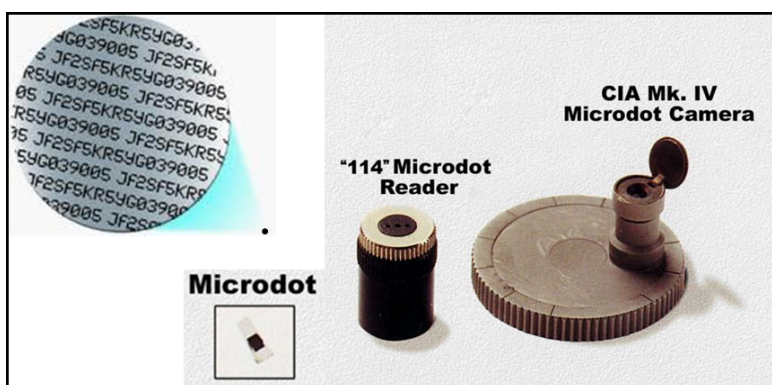
(a) Hidden message on the head of a slave.



(b) Hidden message in wax tablet.

Figure 8.2: Examples of ancient steganography models.

techniques by hiding messages in women's earrings or using pigeons to deliver a messages. He also described many simple methods for hiding in text, by modifying the height of letter strokes and sometimes he marked the letters in a text by using small holes, or by signs, or symbols. Throughout the history, there has always been an urgent need to use of some steganography. As an example, the text steganography proposed by Brassil *et al.* [BLMO95] use the shifting lines of the text up or down by $1/300$ of an inch to encode zeros and ones, which is not visually perceptible and robust against multi-generation photocopying. This small changes are very strong to survive photocopying. During World War I, Brewster's idea became real by the Germans introduced microdots as shown in Figure 8.3. The microdots hidden in corners of postcard were complete documents, photo, and plans reduced in size to the size of a period and attached to common paperwork. The invisible inks have always been a popular method of steganography. Used in the

**Figure 8.3:** Examples of ancient steganography using Germany Microdots.

World War II, the sources for invisible inks are fruit juices, milk, vinegar. With the growth of technology, more advanced inks were developed which react to diverse chemicals [Kah96].

8.3 Cryptography, Watermarking and Steganography

Many techniques have been developed to hidden the secret messages within the image like digital watermarking, visual cryptography, and image steganography [PTC⁺12, KL14, AMY10]. The fields of detecting hidden information in digital medium is called steganalysis methods.

Cryptography, watermarking and steganography are popular techniques for available security. Each technique is different from the other one and are used depending on the purpose. Cryptography is the science of protecting information by transforming it into an unreadable format. Only the use of a secret key can convert the cipher text back into a human readable-text. The main goal is to ensure privacy of data in communication and the ability to send information between participants in a way that prevents others from reading it.

Predominantly, when the cryptographic made the cipher, he depended on the adversary have no any knowledge about the the encryption algorithm. With the great increasing power of computer systems and mathematical theories, some applications become less protection as time passes. Wherefore, some popular algorithms are now considerably less effective, and other methods are developed to use very complex encryption algorithms such as encryption system in banks [Sma03], selective encryption [PBR13] and hash function [SNMM15].

We can divided encryption into two classes based on the key type as shown in Figure 8.4. The first class is symmetric algorithms, also called secret key application where the same key for encryption and decryption is used. This class allows very rapidly quick, it can be use to encrypt small or large data and does not need intensive resource. The problem is that the key must be send to the receiver. The more popular algorithm from this class are the Data Encryption Algorithm (DEA) and Data Encryption Standard (DES) [BS12].

The second class corresponds to asymmetric methods also called public key encryption is a special class, a pair of related by independent keys can be used to perform asymmetric encryption and decryption. This means not need to send over the key to receiver, can be use for encryption and validation. However, it is use for small data and very intensive resource [TG03]. Some of this type of algorithms provide for digital signatures and some others provide key distribution, and some provide both such as the Rivest Shamir Adleman (RSA) algorithm [RSA78], which is the most popular used in electronic commerce.

In fact, steganography methods can be a good way for secret communication when the use of cryptography was forbidden. There is a possibility of using cryptography method as a preprocessing for steganography, by encrypting the message before embedding it inside the cover [AOAKOA12, PTP16].

Digital watermarking scenario is the act of hiding an information called watermark, signature, or a tag into a digital media content. The hidden information is often related to the digital media content. For example, we can embed an image [Ber97, WPD99], an audio [LI03] a video [LLG05, SCP13, KJZC14] and a text [HW04] within the signal itself.

There are four brand applications that can use digital watermarking:

- Copyright protection: This objective is to protect the content of the digital media [BLM99, NMCHGU⁺15].
- Authentication: This objective is to prove the authenticity of printed materials, such as personal checks, identity cards, passports and proof of identity [WL98, NMCHGU⁺15].
- Broadcast monitoring: The objective is to put a unique watermark in each video or sound clip prior to broadcast. Automated monitoring stations can then receive broadcasts and look for these watermarks, identify when and where each clip appears [KDHM99].
- Media enrichment: This objective is to enrichment the data by embedding additional information to the data [MP10].

There are two main types of watermarking: visible watermarking and invisible watermarking, as shown in Figure 8.4.

- Visible watermarking. The information is visible on the digital media. For example, visible image watermarking is a secondary translucent image overlaid into the primary

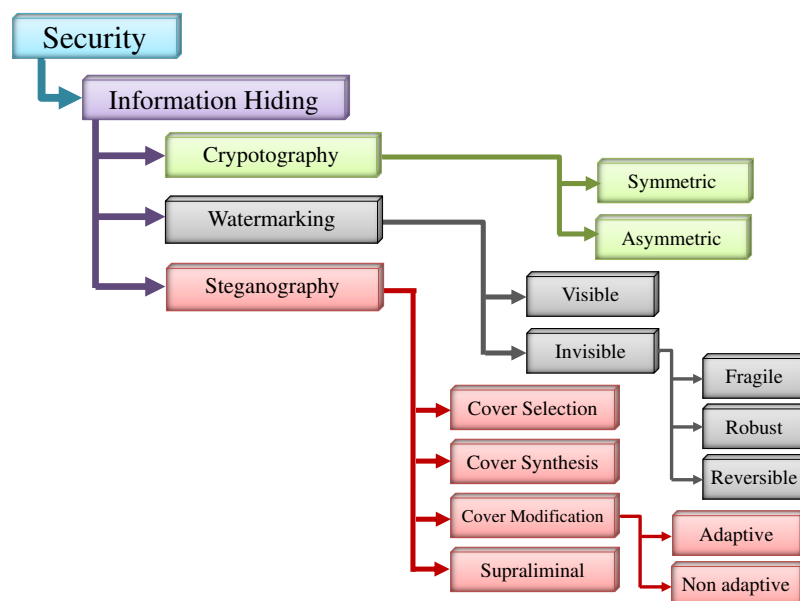


Figure 8.4: The general classification of information hiding methods.

image and appears visible to a viewer on a careful inspection. However, these type of watermarking can be easily corrupted or removed using simple image processing techniques.

- Invisible watermarking. A message is hidden in digital media in imperceptible way. The main goal for using this type of watermarking is to copyright protection systems, which is intended to prevent or deter unauthorized copying of digital media. Invisible watermarks can be classified into robust, fragile and reversible watermarking. Robust watermarks are those which are difficult to remove from the object in which they are embedded, despite various attacks they might be subjected to. Fragile watermarks are those that are easily destroyed by any attempt to tamper with them [CMB⁺07]. Reversible watermarking this techniques was appeared in literature during the last ten years approximately, and are also named as invertible or lossless [Tia02]. These techniques are applied mainly in a way where the authenticity of a digital image has to be granted and the original content is peremptorily needed at the decoding side [CFB10].

Digital steganography is the art and science of hiding messages inside a digital media in a way such that no one, except the sender and receiver, can suspect the presence of the hidden message.

There are four main classifications for steganography techniques according to the construction rules, as shown in Figure 8.4:

- Steganography by cover selection. In this type of steganography the sender selects an image from a large set of available images and applies to it a message. These images always a signification that can be selected by the receiver. Furthermore, the sender can be use a hash function, with a shared secret key between him and the receiver, to transmit his message. In this case, the sender runs its imagery base until it falls on an image whose fingerprint coincides with the desired message. [KJ05, KSM06, SJ09, SL10]. Selection of a suitable cover plays an important role to achieve the goals of steganography. For example, increasing the payload and decreasing detectability. The advantage of this type is that the cover is always seems very natural. On other hand, the disadvantage of this methods are dealing with low payload [Fri09].
- Steganography by cover synthesis, In this type of steganography the sender creates the cover that communicates the desired message [WM08]. We can combined this type of steganography with steganography by cover selection to relieve the exponential complexity of embedding [PKSM05]. The advantage of this type is good security, and the disadvantage is low payload.
- Steganography by cover modification, In this type of steganography the sender modifies an existing cover in order to embed a message, then convey the required message [Fri09]. This method can be used to embed payloads that are large enough in order to make the steganography system very practical [FK13]. There are

two main branches for this technique: adaptive and non-adaptive steganography. Adaptive steganography allows to embed a message into a cover by selecting the most secure embedding positions mostly selecting texture pixels.[PFB10, KCP13, HFD14b, SCF16]. Non-Adaptive steganography allows to embed a message into cover by randomly select the pixels . In other word, the modifications to the cover image are uncorrelated with image features. We can consider that adaptive steganography is an interesting problem of steganography and steganalysis. In this thesis, we used steganography methods by cover modification.

- **Steganography by cover supraliminal.** In this type of steganography, the sender encodes a message in the sementic content of the digital cover. The basic tenets of this types are that, they must be robust in that making small changes to the cover will not destroy the message [MSC⁺13].

Digital watermarking techniques are very close to steganography techniques. However, we can distinguishes between steganography from digital watermarking, by their major goal. As explain before the main goal of watermarking is to hide a message in digital media, in such a way that an eavesdropper cannot remove or replace the hidden message. While the main goal of steganography is to hide a message in digital media, in such a way that an eavesdropper cannot detect the presence of the hidden message. We summarized important issue in Table 8.1.

Table 8.1: The difference between cryptography, watermarking and steganography.

Cryptography	Watermarking	Steganography
• Encrypted message can be seen look like unmeaning mix of characters.	• Hides a message inside a media with or not.	• Hide a message inside media with undetectability.
• Not require cover objects, message is ciphered.	• Require cover objects, message is ciphered and embed into a host.	• Require cover objects, message is ciphered and embed into a cover.
• The cipher is the important information.	• The host media is the important information.	• The hidden message is the important information.
• Can encrypt any size of data.	• Can hide small amount of data.	• Can hide a large amount of data.
• Encrypt data. No channel concept.	• Communication over a noisy channel. Concept of robustness.	• Communication on a secret channel. Concept of undetectability.
• There are some laws that ban cryptography.	• There are some laws that ban watermarking.	• There are no laws associated with steganography.

8.4 The prisoner's problem (steganography/steganalysis)

In 1983, Simmons' proposed the first formulation of the steganography [Sim83] for secret (straightforward) communication. Three characters are necessary to explain this formulation, Alice and Bob two criminal imprisoned in separate cells, far at distance from each other. They can communicate together and they know that this communication is monitored by Eve, the prison warden. They decide to put an escape plan from jail and communicate by covered their messages without getting noticed by Eve. Indeed, if Eve notice about the secret messages, she can decide to remove communication, therefore the escape plan will fail. Figure 8.5 illustrate the general model for the Simmons' prisoners' problem.

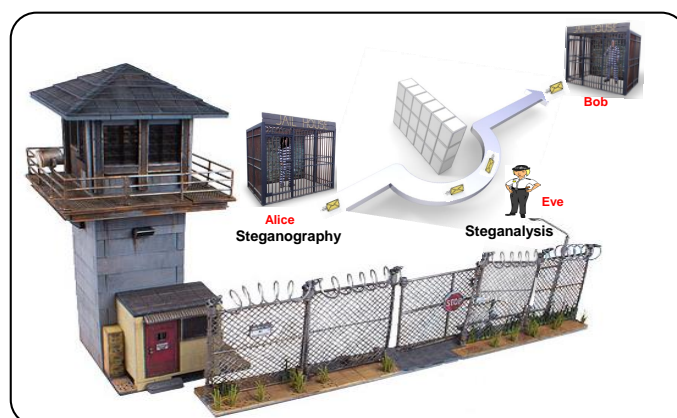


Figure 8.5: The general model for Simmons' "prisoners problem".

From Simmons' model, we can start to explain the roles and the goals for each player. Firstly, Alice and Bob have to use a steganography method. Their goal is communicate together in a highly confidential way and without be suspected by Eve. The goal of steganography is to hide messages inside a cover medium in a way such that no one, except the sender and the receiver, can suspect the presence of a hidden messages. Eve have to use a steganalysis method, his main goal is to monitor the communication. There are three types of models based on the behavior of Eve, the prison warden:

- The passive warden model: In this case, the warden can only prevent or authorize to deliver the message. Eve is restricted from modifying the messages sent by Alice or Bob as shown in Figure 8.6 (a). Eve will analyze the messages. If the test is negative, the message will be delivery and if she detect a secret message, she will block this message [AP98].
- The active warden model: In this case, the warden modify the messages before sending to the other prisoner [Ett98, Cac98], Eve can intentionally modify the messages sent by Alice to Bob or conversely sent by Bob to Alice (as illustrated in

Figure 8.6 (b)). With this procedure, Eve hopes that the secret message will be destroyed. This type of action is interesting for the Internet monitoring where it is very difficult to know the steganography method used by Alice and Bob [FFPN03].

- The malicious warden model: In this case the warden will try to catch the prisoners' communicating and manages the communication as shown in Figure 8.6 (c). Eve will try to modify a large portions of the messages or even produce new messages by impersonating one of the prisoners, Alice or Bob [FIG06].

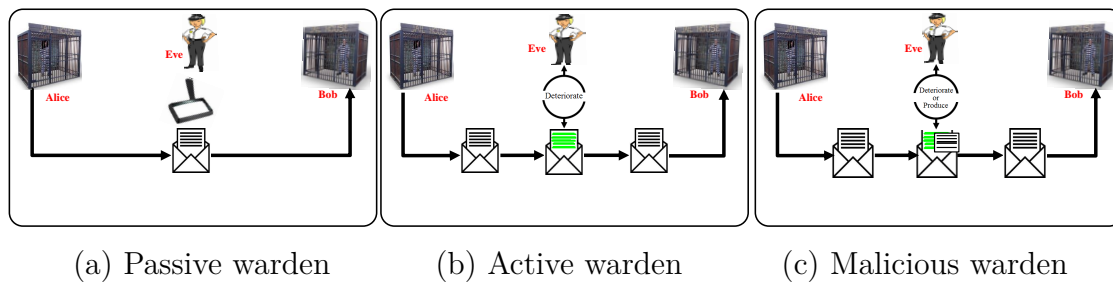


Figure 8.6: Types of models based on the behavior of the prison warden Eve.

8.5 Digital Image Steganography

Image steganography is the concept of embedding a message into digital image content and doing that a manner that it is difficult for an adversary to detect and at the same time difficult for an adversary to remove too.

The security of any digital image steganography method depends on the selection of an image pixels for embedding. More deeply image pixels in textured and noisy area are best choice for embedding secret data, because they are more difficult to detect by the steganalysis methods. Furthermore, Pixels in edges can be seen as noisy pixels because their intensities are either higher or lower than their neighboring pixels due to sudden change in the coefficient gradient. Due to these sharp changes in the visible and statistical characteristics or features, edges are more difficult to detect than the pixels in smoother area. Therefore, edges make an excellent option to embedding messages than any other region of an image where a small distortion is much more noticeable.

We can mathematically notation a steganography scheme. Let as notation K_s denote a key of stego that is generated from a set, K , of all secret stego keys, and M is the set of all embedded messages, and C is the set of all covers. More deeply, Fig. 8.7 illustrate the block diagram of steganography. The steganography scheme is formed by two mappings, the embedding mapping (E_{embed}) to hidden messages, and the extraction

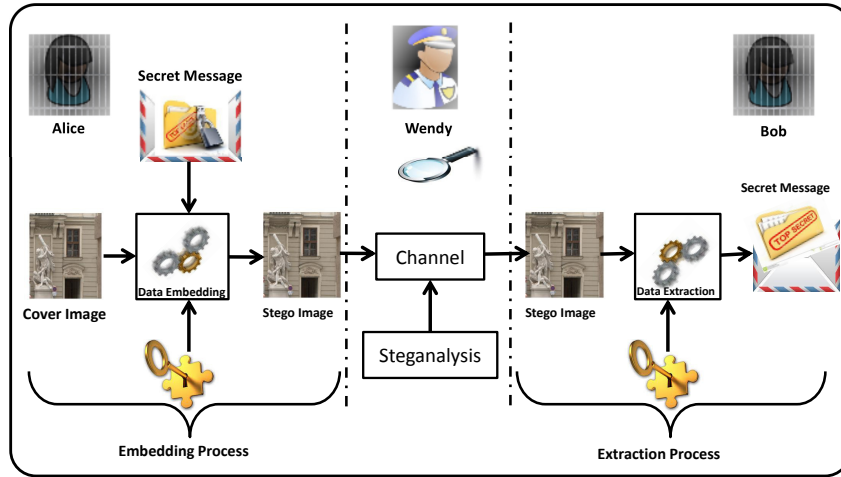


Figure 8.7: The block diagram of steganography.

mapping ($E_{extract}$) to extract the messages:

$$E_{embed} : C \times K \times M \rightarrow C$$

$$E_{extract} : C \times K \rightarrow M.$$

Steganography techniques can be used for almost all digital file formats, but the formats that have a high degree of redundancy are more suitable. Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display. The redundant bits of an object are those bits that can be altered without the alteration being detected easily. There are different types of steganography based on the medium that is used to hide the messages (Text, Audio, Video, Image and protocol). In the digital world, images are the more popular cover medium for steganography. For several reasons, easy handling, large amount of redundant bits present in it, and great proliferation of digital images in the Internet and social media [HDSG15].

There exist several image steganography techniques to hide data securely. We can classify them into two main categories:

- Based on using the key: There are several branches but the more popular are:
 - Pure steganography: Embedding the messages into the cover medium without using any private keys. This method has less security [HLVA02, VAH04].
 - Public key Steganography: uses two types of keys one for embedding messages and another for extracting the messages [Cra98].
 - Secret key steganography: The keys are deployed for embedding the data into the digital cover medium [TC05]. This method is similar to a symmetric cipher, where the sender chooses a cover and embeds the secret message into the cover using

a secret key. If the secret key used in the embedding process is known to the receiver, he can reverse the process and extract the secret message.

- Based on embedding domain: image steganography methods can be classified into four main types of steganography methods, steganography techniques that modify image files for hiding information include the following:

- Spatial-domain methods : In these methods the messages are embedded in intensity of pixels directly. More deeply, directly manipulation of the pixels in the cover image [KBB⁺13].

- Transform domain methods: These types are based on the manipulation of the orthogonal transform of the image rather than the image itself. The frequency image transformations include Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT). The orthogonal transform of the image has two components: one one hand, the magnitude consists of the frequency content of the image. On the other hand, The phase is used to restore the image back to the spatial domain [CCCMK10].

- Statistical algorithm: This methods depends on the existence of 1-bits, which embed one bit of information in a digital image. This is done by modifying the original image in a way that some statistical characteristics change significantly. Always the steganographic tries to change a little bit the characteristics of an image [PH03].

- Distortion algorithm: This algorithms require the knowledge of original cover in the decoding process. More deeply, store information by signal distortion and measure the deviation from the original cover in the decoding step. The total distortion is a sum of individual pixel distortions computed from the cover image [FJF11]. The sender applies a series of modifications to a cover in order to get the stego image. The sequence of modifications corresponds to a specific secret message the sender wants to transmit. The recipient measures the differences to the original image in order to reconstruct the sequence of modifications and this corresponds to the secret message [KDR06, SKZ09, KCP13] are examples of this philosophy.

In general, hidden message is considered as an encrypted data, where bits of encrypted message are embedded in pixels of the cover image. One of simplest steganography technique is based on hidden message it the least significant bit (LSB). One of simplest steganography technique is based on hidden message in the least significant bit (LSB), by modified the LSB of the color or gray image pixels to embed the secret message. The following example shows how the letter **G** has an ASCII code of 071, it will need three consecutive pixels for a 24-bit color image:

Image pixels before hidden:

10010101	00001101	11001001
10010110	00001111	11001011
10011111	00010000	11001011

Secret message is **G**: 01000111

Image pixels after hidden:

10010100	00001101	11001000
10010110	00001110	11001011
10011111	00010001	11001011

This technique has evolved

and produced two types of LSB steganography: First type, is LSB matching which is as follows steps. First, convert the secret message into a stream of bits. Then, take each pixel of the clean image (cover), it is possible to use a pseudo-random order generated by a shared secret key: if the LSB of the next cover pixel matches the next bit of secret data, do nothing; otherwise, choose to add or subtract one from the cover pixel value, at random [Ker05]. The second type, is LSB replacement it is very similar, except that the LSBs of the cover pixels are simply overwritten by the secret bit stream [CC04, Ker04]. In this thesis, we used three steganography methods The first method is the Wavelet Obtained Weights (WOWⁱ) steganography algorithm [HF12a]. The second method is the Spatial-UNIversal WAVElet Relative Distortion (S-UNIWARDⁱⁱ) steganography algorithm [HFD14b]. Finally, the third method is the Synchronizing the Selection Channel (Synch-HILLⁱⁱⁱ) steganography algorithm [DF15].

The WOW method proposed by [HF12b], used directional filters for calculating the detectability map. This method is deal with defining additive steganographic distortion in the spatial domain. More specifically, by employ a bank of directional high-pass filters to obtain the directional residuals, which are related to the predictability of the pixel in a certain direction. First, calculates the weighted difference between the residual wavelet coefficients of the cover image and the residual wavelet coefficients of the stego image and then aggregates the result obtained for Build a detectability map. WOW force the embedding cost to be high where the content is predictable in at least one direction (smooth areas and along edges) and low where the content is unpredictable in every direction (e.g., in textured or noisy areas).

The S-UNIWARD method proposed by [HFD14b], is in the form of a sum of relative

ⁱWOW steganography method is available at http://dde.binghamton.edu/download/stego_algorithms/.

ⁱⁱS-UNIWARD steganography method is available at http://dde.binghamton.edu/download/stego_algorithms/.

ⁱⁱⁱSynch-HILL steganography method is available at http://dde.binghamton.edu/download/stego_algorithms/.

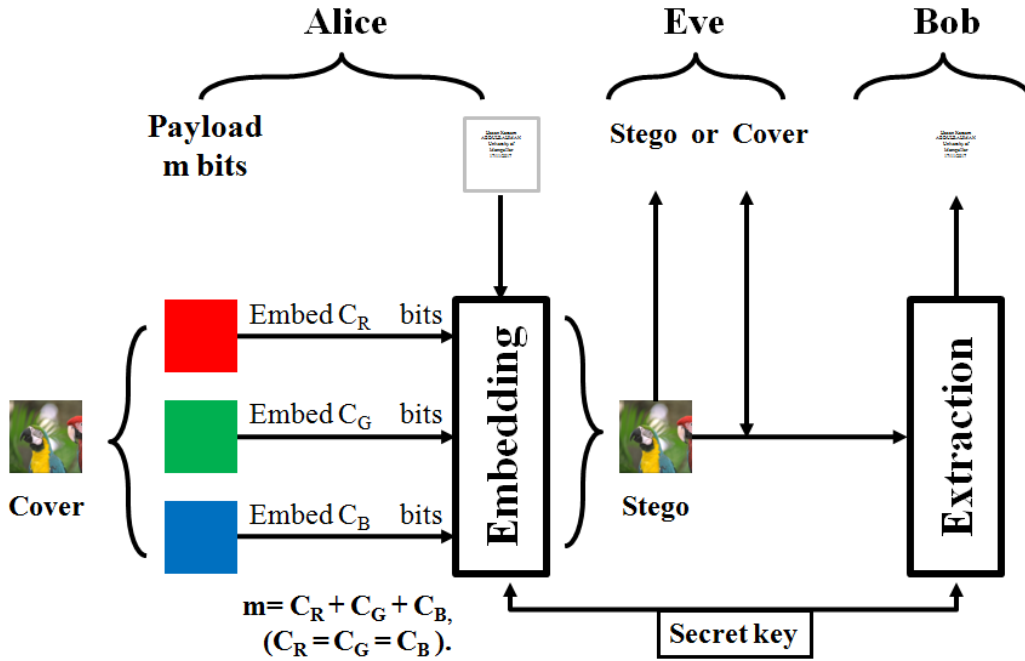


Figure 8.8: Architecture of color image steganography by embed messages into color images by decomposing the R , G and B channels with channel-dependent on the same payload partition.

changes between the stego and cover images represented in the wavelet domain like the WOW. Although WOW and S-UNIWORD have exploited more pixels in texture areas for hiding data.

Synchronizing the Selection Channel Synch-HILL method proposed by [DF15], The cost function of HILL securing all pixels within textural regions have relatively low costs, and is realized by using a high-pass filter and two low-pass filters, making more embedding changes concentrated in textural areas.

As explained in [MBL⁺09], the steganography methods have an important impact on the performance of the general methodology. These algorithms are used to embed messages into color images by decomposing the R , G and B channels like three gray-scale images and embedding the same proportion payload into each channel as shown in Fig. 8.8. Finally, 10,000 color images were used to test each of the seven different payload sizes: $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ Bit Per Channel (BPC).

8.6 Digital Image Steganalysis

Steganalysis is the art of detecting hidden information in a digital image, has received a great deal of attention in recent years. There are many researchers working on solutions ensuring the detection of hidden messages inside digital media. As a result, there are

many techniques and methods that are currently used in the field of steganography and steganalysis. In general, image steganalysis can be classified on the basis of the techniques used into two major categories, specific steganalysis and universal or Generic or Blind steganalysis techniques. The specific steganalysis is designed for a particular steganography algorithm means that the steganalyser knows the functioning and the properties of the steganography technique that used to hide the data. Usually, specific techniques look up for particular distortions. These steganalysis algorithms could be used with other steganography methods. but many times they cannot detect successfully the embedded message.

The universal steganalysis has two important components these are feature extraction and feature classification. In feature extraction, a set of distinguishing statistics are obtained from a data set of images. The second component, feature classification, hold in two classes. First class, is obtained by distinguishing statistics from both stego and cover images are used to train a classifier. The second class, is the trained classifier is used to classify an input image as either being a no hidden image or carrying a hidden message. This part of thesis is deals with universal color image steganalysis Fig. 8.9 illustrated the block diagram of steganalysis system. Many of the image steganalysis methods in the state-of-the-art use image feature based steganalysis and machine learning classification [FK12, PBF10, SLY⁺15].

Extracting features from an image gives a large vector composed of real values characterizing the image. In order to implement this methodology, the steganalyzer needs to extract a set of features from a training data set to train a classifier. Then the classifier which decides whether or not the image contains a message using a testing data set and, if the results are satisfactory, the classifier is considered successful. The models used in image steganalysis are usually obtained by representing images using a set of numerical features. Each image, $X \in C$, is mapped to a d -dimensional feature vector $f = (f_1(x), \dots, f_d(x)) \in R$, where each $f_i : C \rightarrow R$. Any steganalysis algorithm is a detector, which can be described by a map $F: R^d \rightarrow \{0, 1\}$, where $F(x) = 0$ means that x is detected as cover, while $F(x) = 1$ means that x is detected as stego [Fri09]. Imagine, if the similar cover image is used, the feature extraction process provides data sets with similar features and, therefore, the machine learning tools work properly and the classification results are clean image. While, if different cover image is used, the data sets obtained by feature extraction are also different [WFHP16]. This methodology is widely adopted in classification tasks.

The most widely used classifier of the state-of-the-art is the ensemble classifier [KFH12a]. Steganalysis can be broadly classified into two classes: signature steganalysis and statistical steganalysis.

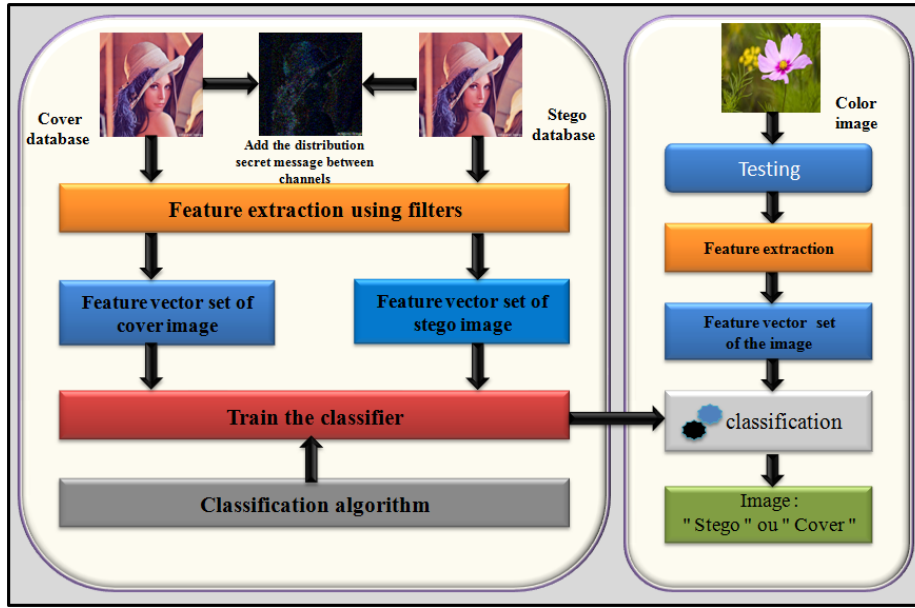


Figure 8.9: Block diagram of universal steganalysis approach.

8.6.1 Color image Steganalysis

In 2015, color image steganalysis has been studied by integrating the modern adaptive embedding method in an experimental evaluation. Previous steganalysis methods did not use recent grey-level embedding algorithms [FL00b, WP00, KB14, LSR⁺08] or did not use a machine learning approach with Rich-Model features [HP03, TAV12].

8.6.1.1 Color Spatial Rich Model Steganalysis

As it is well known, embedding a message in an image modifies some pixel values. Indeed, this modification provides slight changes to the pixel values where the message is embedded. It is a difficult task to detect and extract the sensitive features. Many methods apply high-pass filters to the target image, and then compute high order statistics on the filtered images. Goljan *et al.* [GFC14] have introduced efficient color image features which are an extension of the Spatial Rich Model [FK12], produced from two different sets of features. First of all, this method extracts the noise residual from each color channel separately. Let us note that X_{ij} is a pixel value of an 8-bit grayscale cover image. We can specify the red, green and blue channel of color images by the following formula:

$$\mathbf{R}_{ij} = \hat{X}_{ij}(\mathcal{N}_{ij}) - c \cdot X_{ij}, \quad (8.1)$$

where:

- $c \in \mathbb{N}$, is the residual order,
- \mathcal{N}_{ij} , is a local neighborhood of pixel X_{ij} at coordinates (i, j) ,
- $\hat{X}_{ij}(\cdot)$ is a predictor of $c \cdot X_{ij}$, $X_{ij} \notin \mathcal{N}_{ij}$, $X_{ij} \in \{0, \dots, 255\}$.

Many diverse submodels built from the differences between neighboring pixels are combined in the rich model, all of the submodels $(\mathbf{R}_{ij}) \in \mathbb{R}^{n_1 \times n_2}$ are formed from noise residual images of size $n_1 \times n_2$ computed using high pass filters of the following form:

$$\mathbf{R}_{ij} \leftarrow \text{tranc}_T \left(\text{round} \left(\frac{R_{ij}}{q} \right) \right), \quad (8.2)$$

where:

$$\bullet R_{ij} = \begin{cases} \text{tranc}_T(x) & = x & \text{for } x \in [-T, T], \\ \text{tranc}_T(x) & = T \cdot \text{sign}(x) & \text{otherwise.} \end{cases}$$

- q is the quantization step,
- round is a function for rounding to an integer value.

The Spatio-Color Rich Model consists of two different components. On one hand, the Spatial Rich Model (*SRMQ1*) [FK12] with a fixed quantization $q = 1$ and truncation $T = 2$ yields a dimensionality of 12753 features. These features are computed from each R , G and B color channel separately. Finally, the three dimensionality features are added together to keep the same dimensionality as for grayscale images. On the other hand, from the same noise residuals (i.e. *SRMQ1*), the *CRMQ1* builds a collection of 3D color co-occurrence matrices, taking three color values at the same position (across the three channels of each pixel). Thus, with fixed truncation $T = 3$ and quantization $q = 1$, *CRMQ1* produces 5404 features per image.

8.6.1.2 CFA-aware features steganalysis

Digital cameras capture color images using a single sensor in conjunction with a Color Filter Array (*CFA*) interpolation. The *CFA* allows us to capture only one part of the spectrum through the sensor so that only one color is measured at each pixel (red, blue or green) and so the resulting images are called mosaic images. To construct a color image, a demosaicking algorithm is used in order to interpolate each color plane (i.e. *CFA* interpolations). Several patterns exist for the color filter array, with the most common

being Bayer *CFA* [Bay76]. During this process, the green color channel is the most important factor which determines the luminance of the color image, 50% of the pixels in the Bayer *CFA* structure are assigned to the green channel, while 25% are assigned to the red channel and 25% to the blue color channel [WZH11].

Goljan *et al.* introduced in [GF15] the CFA-aware *CRM* for color image steganalysis. The features are made from two parts, the first one is the Color Rich Model *CRMQ1* explained in section 8.6.1.1 with $T \in \{2, 3\}$. The second part is the CFA-aware feature, which consists of three combinations: *RB/GG split*, *R/B/GG split* and *NII/INI split*.

Let us note, if X has a true-color image size of $n_1 \times n_2$, where n_1 and n_2 are even numbers, ($0 \leq i < n_1$, $0 \leq j < n_2$). Considering a typical Bayer mosaic, the G sub-image has twice as many pixels as the R and B sub-images. We must mention that, all the color images used in this method are cropped from one pixel position which is the upper left pixel corresponding to a non-interpolated blue in the Bayer *CFA*. The color noise residuals $Z = (z_{ij}^{(R)}, z_{ij}^{(G)}, z_{ij}^{(B)})$ is computed as Eq.8.1, corresponding to *CFA* used map. First of all, the following four index sets must be generated:

$$\begin{aligned} X_B &= \{(i, j) | i \text{ even}, j \text{ even}\}, \\ X_{G1} &= \{(i, j) | i \text{ odd}, j \text{ even}\}, \\ X_{G2} &= \{(i, j) | i \text{ even}, j \text{ odd}\}, \\ X_R &= \{(i, j) | i \text{ odd}, j \text{ odd}\}. \end{aligned}$$

Four 3D co-occurrence matrices are computed from residual samples due to the above index sets.

$$C_{d_1 d_2 d_3}^{(B)} = \sum_{(i,j) \in X_B} \left[(z_{ij}^{(R)}, z_{ij}^{(G)}, z_{ij}^{(B)}) = (d_1, d_2, d_3) \right], \quad (8.3)$$

$$C_{d_1 d_2 d_3}^{(G1)} = \sum_{(i,j) \in X_{G1}} \left[(z_{ij}^{(R)}, z_{ij}^{(G)}, z_{ij}^{(B)}) = (d_1, d_2, d_3) \right], \quad (8.4)$$

$$C_{d_1 d_2 d_3}^{(G2)} = \sum_{(i,j) \in X_{G2}} \left[(z_{ij}^{(R)}, z_{ij}^{(G)}, z_{ij}^{(B)}) = (d_1, d_2, d_3) \right], \quad (8.5)$$

$$C_{d_1 d_2 d_3}^{(R)} = \sum_{(i,j) \in X_R} \left[(z_{ij}^{(R)}, z_{ij}^{(G)}, z_{ij}^{(B)}) = (d_1, d_2, d_3) \right]. \quad (8.6)$$

From the above four co-occurrence matrices, three combinations of features are generated to form the total number of features with the *CRMQ1* set:

The first combination is called *RB/GG split* which generates 4146 features. $C_{d_1 d_2 d_3}^{(R)}$ and $C_{d_1 d_2 d_3}^{(B)}$ are treated and added together, the same thing is applied to $C_{d_1 d_2 d_3}^{(G1)}$ and $C_{d_1 d_2 d_3}^{(G2)}$

as in Eq.'s 8.5 and 8.6.

$$C_{d_1 d_2 d_3}^{(RB)} = C_{d_1 d_2 d_3}^{(B)} + C_{d_3 d_2 d_1}^{(B)} + C_{d_1 d_2 d_3}^{(R)} + C_{d_3 d_2 d_1}^{(R)}, \quad (8.7)$$

$$C_{d_1 d_2 d_3}^{(GG)} = C_{d_1 d_2 d_3}^{(G1)} + C_{d_3 d_2 d_1}^{(G1)} + C_{d_1 d_2 d_3}^{(G2)} + C_{d_3 d_2 d_1}^{(G2)}. \quad (8.8)$$

R/B/GG split represents the second set and produces 10323 features. This part can be considered as an important component in this method, because it gives a considerable number of features. It can be generated from the concatenation of $C_{d_1 d_2 d_3}^{(R)}$, $C_{d_1 d_2 d_3}^{(B)}$, and $C_{d_1 d_2 d_3}^{(G1)} + C_{d_1 d_2 d_3}^{(G2)}$.

The third set corresponds to the *NII/INI split*; 'N' meaning non-interpolated and 'I' interpolated respectively, in the *RGB* triple. The 'NII' pixels correspond to the same set as RB but the two co-occurrence matrices are directionally symmetrized differently. This set generates 5514 features from two co-occurrence matrices:

$$C_{d_1 d_2 d_3}^{(NII)} = C_{d_3 d_2 d_1}^{(B)} + C_{d_1 d_2 d_3}^{(R)}, \quad (8.9)$$

$$C_{d_1 d_2 d_3}^{(INI)} = C_{d_1 d_2 d_3}^{(GG)}. \quad (8.10)$$

All these features are gathered in a one dimensional vector, while all detectors are trained as binary classifiers implemented using Kodovsky ensemble classifiers [KFH12a], as explained in the following Section 8.6.2.

8.6.2 The ensemble classifiers

An ensemble of classifiers [Die00] is a set of classifiers whose individual decisions are combined and organized into weighted or unweighted votes to classify the data sets (in this work, features represent these data sets, as detailed in the previous sub-section). Modern steganalysis methods for digital images are based on feature extraction. These methods need machine learning techniques to detect if the media contains hidden messages or not. In our work, we choose ensemble classifiers [KFH12a] because of their efficient classification performance for large scale learning.

Kodovsky *et al.* [KFH12a] proposed ensemble classifiers^{iv} which is a machine learning tool for steganalysis, consisting of many classifier L independently trained (B_l) designed to keep complexity to a minimum and make the overall process simple.

Each base learner is trained on randomly selected subspaces d_{sub} -dimensionals of the original feature space, from the entire full d -dimension feature space. The authors use Ficher Linear Discriminants (*FLD*) as base learners and the final decision is made by

^{iv}Ensemble classifier is available at <http://dde.binghamton.edu/download/ensemble>.

aggregating the decision of individual base learners. Let d be a full dimensional feature space, N^{trn} and N^{tst} the number of training and testing samples from each class. First, the classifiers construct a number L of *FLD* base learners (B_l) with $l \in \{1, \dots, L\}$. Each one performs its learning on a subspace of d_{sub} dimension, where $d_{sub} \ll d$. From the i^{th} image, a feature vector, $f_i \in \mathbb{R}^d$, is extracted, and then mapped, such as $\mathbb{R}^d \rightarrow \{0, 1\}$, where '0' stands for cover and '1' for stego.

In the learning phase, each classifier learns to map a feature vector f_i , to the correct class number:

$$FLD_l : \mathbb{R}^d \rightarrow \{0, 1\}$$

$$f_i \rightarrow FLD_l(f_i).$$

Each classifier uses the training database to compute the orthogonal vector to the hyperplane separating the two classes. For a test feature, the l^{th} base learner reaches its decision by computing a projection and comparing it to a threshold. After collecting all L decisions, the final classifier selects the class which has received the most votes. Then, the decision threshold of each base learner is adjusted to minimize the total detection error under an equal prior on the training data [KFH12a]:

$$P_E = \min_{P_{FA}} \frac{1}{2} [P_{FA} + P_{MD}(P_{FA})], \quad (8.11)$$

where P_{FA} represents the false alarm probability and P_{MD} the missed detection probability.

8.7 Summary

In this chapter, we presented the different concepts of steganography, wherein we give an overview of the steganography and steganalysis methods. Also, the history of steganography and steganalysis through these ages are presented. The important difference points between the three highly linked disciplines: cryptography, watermarking and steganography techniques. Each technique is different from the other one and are used depending on the purpose. A review of each technique has been given. In reality, the steganography can be used for legal and illegal purpose, but in recently it has received a great attention from national security seasons. Throughout the history, there has always been an urgent need to use of some steganography. Image steganography is the concept of embedding a message into digital image content and doing that a manner that it is difficult for an adversary to detect and at the same time difficult for an adversary to remove too. The security of any digital image steganography method depends on the selection of an image pixels for embedding. More deeply, image pixels in textured and edge area are best choice

for embedding secret data, because they are more difficult to detect by the steganalysis methods. There exist several image steganography techniques to hidden data securely. We can classified into two main categories: based on using the key and based on embedding domain. We describes the latest recent methods in color image steganalysis by recalling the color rich model method (CRM) [GFC14], the features are computed from each R , G and B color channel separately. and CFA-aware features steganalysis (CFARM) [GF15]. The features are made from two parts, the first one is the Color Rich Model $CRMQ1$ explained in section 8.6.1.1 with $T \in \{2, 3\}$. The second part is the CFA-aware feature, which consists of three combinations: RB/GG split, $R/B/GG$ split and NII/INI split. Finally, modern steganalysis methods for digital images are based on feature extraction. These methods need machine learning techniques to detect if the media contains hidden messages or not. In our work, we choose ensemble classifiers [KFH12a] because of their efficient classification performance for large scale learning.

COLOR CHANNEL CORRELATION AND GEOMETRIC STEGANALYSIS

This chapter describes our steganalysis method for color images by extracting features using color channel correlation and geometric mirror transformations.

Contents

9.1	Introduction	152
9.2	Methodology description	152
9.2.1	RGB Channel Correlation	153
9.2.2	Mirror transformations	154
9.2.3	Complete feature set	155
9.3	Building Image Dataset	156
9.4	Experimental results	158
9.4.1	Embedding methods	158
9.4.2	Results and Discussion	159
9.5	Summary	162

9.1 Introduction

Digital color images have become a good medium for digital steganography due to their easy manipulation as carriers via Internet, e-mails, or used on websites. As a result, there are many techniques and methods that are currently used in the field of steganography and steganalysis [RO13]. Although, the real-world uses significantly more color images than grayscale images, there is a lot of research in steganalysis of grayscale images compared to color images [FL00a]. The main goal of steganalysis is to detect the presence of hidden messages in a digital media. For this reason, a new method for color image steganalysis using RGB color channels correlation is proposed and discussed in this chapter.

Our method uses two types of features computed between color image channels. The first type of feature reflects local Euclidean transformations [ACMM15] and the second one reflects mirror transformations [ACMM16b]. These geometric measures are obtained by the sine and cosine of gradient angles between all the color channels. Features are extracted from co-occurrence correlation matrices of measures. Fusing our features with those obtained from Color-Rich Models [GFC14] allows increasing the detectability of hidden messages in the color images.

9.2 Methodology description

Our proposition is to enrich the Spatial Color Rich Model *SCRMQ1* with an inter-channel correlation which is composed of three sets of features. The first set, produced by [GFC14], gives 18157 features. The second set, produced by our first method [ACMM15], gives 3000 features. Additionally, the third set, produced by a second method, gives 3000 features; they are obtained from the new correlation of different R , G and B channel gradients, as shown in Table 9.1.

Table 9.1: Features description with their dimmensionalities corresponding to q and T .

Feature set	<i>SCRMQ1</i>	$\mathcal{C}_{RG}/\mathcal{C}_{RB}$	$\mathcal{S}_{RG}/\mathcal{S}_{RB}$
Dim. Symmetry	yes	yes	yes
Dimension	18157	3000	3000

The following section recalls the RGB Channel Correlations which gives an explanation to our proposition.

9.2.1 *RGB Channel Correlation*

In this section, we introduce an inter-channel correlation measure, and demonstrate that it can be linked to first order Euclidean invariants [Hil93]. Such invariants have mainly been used for stereo-matching [GMP98]. In our method, we show that the information provided can enhance steganography detection. The underlying idea here, is that if one channel has been affected by steganography, the inter channel correlation will measure the local modifications. As a result, we can easily detect the modification in the color image.

Starting from the local correlation of red and green channels (similarly for the correlation of red and blue channels) is defined as:

$$Corr_{R,G}(i, j, k, l) = \sum_{(i', j') \in \mathcal{W}_{i,j}} X_{i',j'}^{(R)} \cdot X_{k+i', l+j'}^{(G)} \quad (9.1)$$

with:

- $X_{i',j'}^{(R)} \in [0, 255]$, being a pixel value at position (i', j') in the red channel,
- $X_{k,l}^{(G)} \in [0, 255]$, being a pixel value at position (k, l) in the green channel,
- $\mathcal{W}_{i,j}$, representing a small window centered in (i, j) .

Considering $(k, l) = (0, 0)$ and a limited development of $X^{(R)}$ and $X^{(G)}$ around (i, j) , then:

$$\begin{aligned} Corr_{R,G}(i, j, 0, 0) = & \sum_{\substack{\mathbf{h} = (i'-i, j'-j) \\ (i', j') \in \mathcal{W}_{i,j}}} \left(X_{i,j}^{(R)} + \nabla X_{i,j}^{(R)} \cdot \mathbf{h} \right) \left(X_{i,j}^{(G)} + \nabla X_{i,j}^{(G)} \cdot \mathbf{h} \right). \end{aligned} \quad (9.2)$$

Developing this equation leads to four terms, three of them are constant or not informative, then there is only one informative term :

$$\nabla X_{i,j}^{(R)} \cdot \nabla X_{i,j}^{(G)}. \quad (9.3)$$

If only one channel has been altered locally, the gradient in this channel is modified. Consequently, the scalar product of two channel gradients reflects the change in the cosine of the difference between the two gradient angles.

Similarly, we can apply the same computation for the red and blue channel and then obtain :

$$\nabla X_{i,j}^{(R)} \cdot \nabla X_{i,j}^{(B)}. \quad (9.4)$$

As stated by Gouet *et al.* [GMP98] (and following the Hilbert theory [Hil93]), it is unnecessary to investigate the $\nabla X_{i,j}^{(G)} \cdot \nabla X_{i,j}^{(B)}$ term, as it is already implicitly contained in the first two expressions (Eq. 9.3 and 9.4).

Normalizing these expressions, we obtain the cosine of rotation angles, between channel gradients:

$$\mathcal{C}_{RG} = \frac{\nabla X_{i,j}^{(R)} \cdot \nabla X_{i,j}^{(G)}}{|\nabla X_{i,j}^{(R)}| |\nabla X_{i,j}^{(G)}|}, \quad (9.5)$$

$$\mathcal{C}_{RB} = \frac{\nabla X_{i,j}^{(R)} \cdot \nabla X_{i,j}^{(B)}}{|\nabla X_{i,j}^{(R)}| |\nabla X_{i,j}^{(B)}|}. \quad (9.6)$$

Fig. 9.1 illustrates our preprocessing steps [ACMM15] to obtain the cosine of rotation angles, between channel gradient. We recall that the gradients derivatives of each channel are estimated by a convolution with a $[-1; 1]$ mask (horizontal and vertical).

9.2.2 Mirror transformations

In the preceding section, we have seen that the inter-channel correlation is linked with the scalar product of gradients (i.e. Euclidean invariants). This means that if we are able to measure the absolute value of a rotation angle between two channel gradients, we still

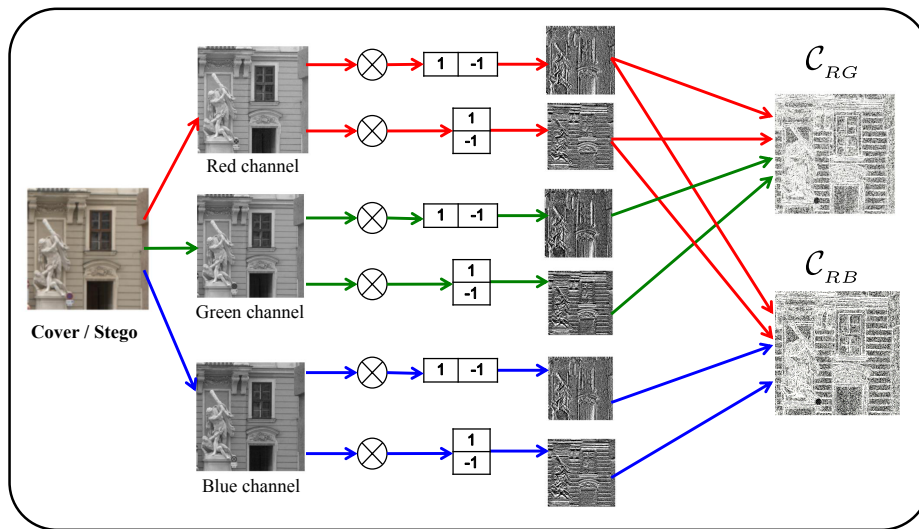


Figure 9.1: Features extraction: Cosine of the gradient angles [ACMM15].

need the direction of the rotation, which is linked this time to Mirror transformations (as illustrated in Fig. 9.2).

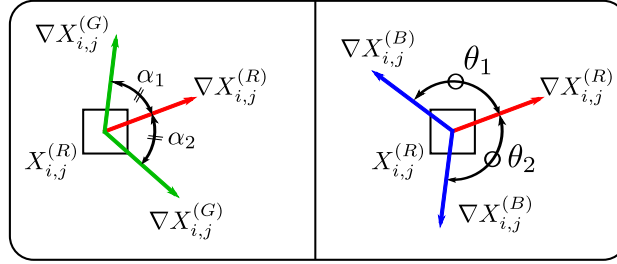


Figure 9.2: Rotation angle between two channel gradients

$\cos(\alpha_1) = \cos(\alpha_2)$ but $\sin(\alpha_1) = -\sin(\alpha_2)$

$\cos(\theta_1) = \cos(\theta_2)$ but $\sin(\theta_1) = -\sin(\theta_2)$.

Sine is essential to determine the direction of the rotation.

Our proposition is to add two new features sets based on the determinants of channel gradients. Similar to that applied in previous subsection, the features are directly linked to the correlation in order to obtain new features of Sine of the gradients angle. Finally, as illustrated in Fig. 9.3, we normalize these determinants by gradient norms to obtain the sine of the rotations:

$$\mathcal{S}_{RG} = \frac{\nabla X_{i,j}^{(R)}[0] \cdot \nabla X_{i,j}^{(G)}[1] - \nabla X_{i,j}^{(R)}[1] \cdot \nabla X_{i,j}^{(G)}[0]}{|\nabla X_{i,j}^{(R)}| |\nabla X_{i,j}^{(G)}|}, \quad (9.7)$$

$$\mathcal{S}_{RB} = \frac{\nabla X_{i,j}^{(R)}[0] \cdot \nabla X_{i,j}^{(B)}[1] - \nabla X_{i,j}^{(R)}[1] \cdot \nabla X_{i,j}^{(B)}[0]}{|\nabla X_{i,j}^{(R)}| |\nabla X_{i,j}^{(B)}|}, \quad (9.8)$$

Where $\nabla X[0]$ (resp. $\nabla X[1]$) is the first (resp. second) component of the vector ∇X i.e. corresponding to the horizontal and the vertical derivatives (see Fig. 9.3).

9.2.3 Complete feature set

Our features, are computed from \mathcal{C}_{RG} , \mathcal{C}_{RB} , \mathcal{S}_{RG} and \mathcal{S}_{RB} correlations by computing the co-occurrence matrices as in the Rich Model [FK12]. We used different values of the quantization $q \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ with fixed truncation $T=1$. The reason for using these different values of quantization q is that \mathcal{G}_{RG} , \mathcal{G}_{RB} , \mathcal{S}_{RG} and \mathcal{S}_{RB} belong to $[-1, 1]$. Moreover, the use of these values gives more accurate features and avoids the generation of too many zero values caused by the truncation step in the co-occurrence vector. For each quantization, we obtain 12 submodels from methods 1 [ACMM15] and 12 submodels

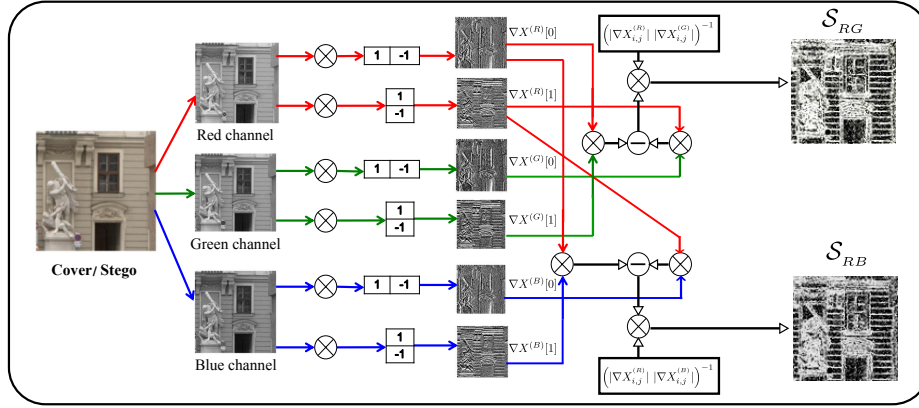


Figure 9.3: Features of extraction: Sine of the gradients angles extracting information from the direction of the local rotation.

from the new proposed method 2ⁱ The submodels from the Color Rich Models [GFC14] give 18157 features, those of the method 1 [ACMM15] give 3000 features, and those of our proposed method 2 give 3000 features. Accordingly, the final feature vector collects a final set of 24157 features as shown in Table 9.1.

9.3 Building Image Dataset

A raw image is a class of computer file containing untouched pixel information coming from the digital camera sensor (i.e. the pure information). These files hold a large amount of meta-information about the image generated by the camera [YS11].

In our work, the color image database is very carefully built depending on the CFA idea. We collected raw images from two subsets which are the most standard, and have the highest number of images captured (i.e. the Dresden Image Database's [GB10] 3500 full-resolution Nikon digital camera raw color images and the Break Our Steganographic System (BOSSbaseⁱⁱ), with 1000 Canon digital camera raw color images).

In order to obtain color images in Portable Pixel Map (PPM) format of size 512×512, all images take the same CFA map layout, as illustrated in Fig.9.5. For this process, two steps are required. The first step consists of using a demosaicking algorithm to convert raw images into demosaicked images. The second step consists of cropping five areas from one image. Fig. 9.4 shows sample images produced by the cropping step.

ⁱ For method 1 (resp. method 2) we use one symmetrized spam14h and one spam14v submodel, with 25 features each. We also use the minmax22h, minmax22v, minmax24, minmax34h, minmax34v, minmax41, minmax34, minmax48h, minmax48v, and minmax54 submodels with 45 features for each. All submodels are gathered in a one dimension vector to erect a dimensionality of $(2 \times 25 + 10 \times 45) \times 6 = 3000$ features. For more details on submodels construction, the reader is invited to look at article [FK12].

ⁱⁱBOSSbase can be accessed at <http://www.agents.cz/boss/BOSSFinal>.

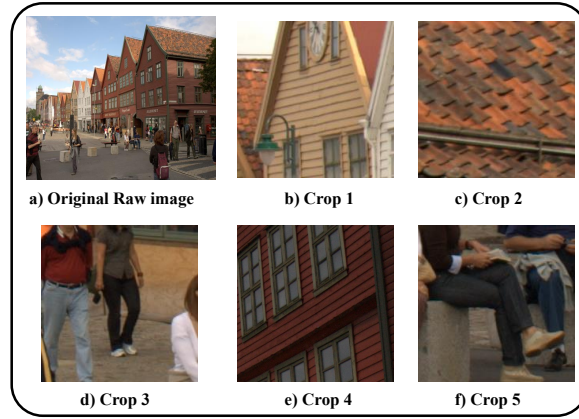


Figure 9.4: Sample images of our database built by random cropping from locations of red channel pixels (even position) in a Bayer pattern :

- a) Original raw image 3906×2602 , b)] crop 1 position $x=2116, y=1928$,
- c) crop 2 position $x=902, y=1182$, d)] crop 3 position $x=3080, y=436$,
- e) crop 4 position $x=1866, y=1778$, f)] crop 5 position $x=650, y=1032$.

First we used the demosaicking algorithm Patterned Pixel Grouping (PPG) from the dcraw softwareⁱⁱⁱ to convert raw images into RGB images. As illustrated in Fig.9.5, the obtained images are such that the Bayer Pattern is always of the type RGBG (red channel pixel is placed at an even position). We wrote a spatial code to start the crop from the red channel position. Indeed, from one image, this code randomly selected the red channel position and cropped five images using a size of 512×512 pixels, so that all blocks share the same CFA map layout. The final number of images is 10000 *RGB* color images with a size of 512×512 .

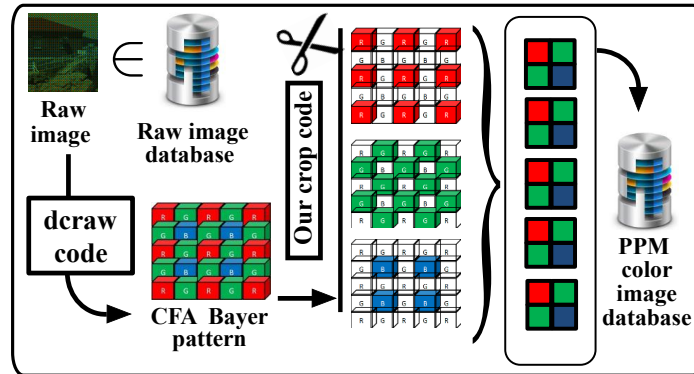


Figure 9.5: The preprocessing steps for building our database depending on the *CFA* idea.

ⁱⁱⁱdcraw code is available at <http://www.cybercom.net/defin/dcraw>.

9.4 Experimental results

All our features are calculated and formed in a one dimensional vector from 10000 color covers and 10000 color stego images for each payload of steganography methods. These features are ready to enter in the classifier. The classifiers were implemented using the ensemble classifier [KFH12a] with many FLD as a base learner as discussed in Chapter 8, Section 8.6.2. In this experiment, the detection accuracy is measured by the total probability of the average of testing errors under equal priors as in Eq. 8.11. 5000 images from a database are randomly chosen for the training sets and 5000 for the testing sets. The ensemble classifiers apply a vote to estimate the error of detection. This process is repeated 10 times to obtain \bar{P}_E , the average of testing errors. \bar{P}_E quantify the detectability and are collected for each method and payload to evaluate the steganalysis method. Given the decision values, *ROC* curves are obtained. As illustrated in Fig. 9.7, the area under the *ROC* curves is calculated as the accuracy of the ensemble classifiers.

9.4.1 Embedding methods

The stego images are obtained using three spatial-domain steganography algorithms. The first method is the Spatial-UNiversal Wavelet Relative Distortion (S-UNIWARD^{iv}) steganography algorithm [HFD14a]. The second method is the Wavelet Obtained Weights (WOW^v) steganography algorithm [HF12a]. Finally, the third method is the Synchronizing the Selection Channel (Synch-HILL^{vi}) steganography algorithm [DF15].

These algorithms are used to embed messages into color images by decomposing the *R*, *G* and *B* channels as three grayscale images and embedding the same proportion payload into each channel. Also, different tested payload sizes are used {0.01, 0.05, 0.1, 0.2, 0.3, 0.4 and 0.5} Bit Per Channel (*BPC*).

^{iv}S-UNIWARD steganography method is available at http://dde.binghamton.edu/download/stego_algorithms/.

^vWOW steganography method is available at http://dde.binghamton.edu/download/stego_algorithms/.

^{vi}Synch-HILL steganography method is available at http://dde.binghamton.edu/download/stego_algorithms/.

9.4.2 Results and Discussion

This section contains the experimental results of our proposed method. We illustrate these results in Table 9.2. S-UNIWARD, WOW and Synch-HILL methods were tested with different relative payloads $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ (bpc) against three approaches: method 1 [ACMM15], the Color Rich Model [GFC14] and the CFA-aware features steganalysis [GF15]. We used the same set of payload values with the same embedding methods. Our proposed second method, that uses both the sine and cosine of the gradients angle, achieved higher performance by registering 88.76%, 87.93% and 88.07% detection rates for S-UNIWARD, WOW and Synch-HILL respectively (with the payload 0.5 bpc). The Color Rich Model method [GFC14] is less efficient because it achieved respectively 86.14%, 85.27% and 85.25% detection. Also, the CFA-aware features method [GF15] is less efficient because it achieved respectively 87.61%, 87.04% and 87.42% detection rates. Close to the CFA-aware features method, our method 1 [ACMM15] is less efficient because it achieved respectively 87.54%, 86.63% and 86.77% detection rates. We noted the same trend with the rest of the payload values, as shown in Table 9.2.

Additionally, as shown in Table 9.2, our method 1 [ACMM15], that use the cosine of the gradients angle, achieved higher performance than Color Rich Model method [GFC14]; by registering 87.54%, 86.63% and 86.77% detection rates for S-UNIWARD, WOW and synch-HILL respectively with the payload 0.5 bpc. For the same payloads range, the Color Rich Model method [GFC14] is less efficient because it achieved respectively 86.14%, 85.27% and 85.25% detection rates on the same test samples. Also, as shown in Table 9.2, our proposed second method, that uses the sine and cosine of the gradients angle, achieved higher performance than CFA-aware features steganalysis method [GF15]; by registering 88.76%, 87.93% and 88.07% detection rates for S-UNIWARD, WOW and synch-HILL respectively with the payload 0.5 bpc. The CFA-aware features steganalysis method [GF15] is less efficient because it achieved respectively 87.61%, 87.04% and 87.42% detection rates on the same test samples.

Moreover, curves in Fig.9.6 (a) S-UNIWARD, (b) WOW and (c) synch-HILL steganography methods, illustrate the comparison between the proposed second method and the compared methods. As a result, the average testing error of the proposed second method is less than the first proposition, the Color Rich Model and CFA-aware features method. That proves the importance of the additional 3000 features proposed by the second method.

Another experiment involved embedding the entire payload in only one channel of the color image, i.e. with payload 0.2 bpc and 0.4 bpc in the green channel only. In this case, the detection rate becomes higher than the same payload distributed equally between the three color channels. Table 9.3 illustrates the comparison of detection rates between the S-UNIWARD, WOW and synch-HILL methods with payloads 0.2 bpc and 0.4 bpc embedded in one channel only and in the three channels separately. Fig. 9.7 (a), (b) and

Table 9.2: Numerical values of the average testing error \bar{P}_E and the detection rate $\mathcal{P}_D\%$ for three steganography methods. For easier navigation the dark gray background column presents our first method [ACMM15] and the light gray background column presents the our second proposed method [ACMM16b].

payload	Color Rich		CFA-Aware		Method 1		Method 2	
	\bar{P}_E	$\mathcal{P}_D\%$	\bar{P}_E	$\mathcal{P}_D\%$	\bar{P}_E	$\mathcal{P}_D\%$	\bar{P}_E	$\mathcal{P}_D\%$
S-UNIWARD								
0.01	0.4841	51.59	0.4863	51.37	0.4830	51.70	0.4680	53.20
0.05	0.4045	59.55	0.4072	59.28	0.4010	59.90	0.3859	61.41
0.1	0.3298	67.02	0.3194	68.06	0.3203	67.97	0.3037	69.63
0.2	0.2498	75.02	0.2317	76.83	0.2370	76.30	0.2191	78.09
0.3	0.1947	80.53	0.1806	81.94	0.1808	81.92	0.1623	83.77
0.4	0.1599	84.01	0.1429	85.71	0.1470	85.30	0.1289	87.11
0.5	0.1386	86.14	0.1239	87.61	0.1246	87.54	0.1124	88.76
WOW								
0.01	0.4850	51.50	0.4875	51.25	0.4836	51.64	0.4753	52.47
0.05	0.4092	59.08	0.4174	58.26	0.4042	59.58	0.3906	60.94
0.1	0.3397	66.03	0.3275	67.25	0.3317	66.83	0.3161	68.39
0.2	0.2654	73.46	0.2440	75.60	0.2502	74.98	0.2381	76.19
0.3	0.2081	79.19	0.1895	81.05	0.1918	80.82	0.1793	82.07
0.4	0.1783	82.17	0.1487	85.13	0.1574	84.26	0.1384	86.16
0.5	0.1473	85.27	0.1296	87.04	0.1307	86.63	0.1207	87.93
Synch-HILL								
0.01	0.4893	51.07	0.4843	51.57	0.4814	51.83	0.4687	53.13
0.05	0.3991	60.09	0.4030	59.70	0.3879	61.21	0.3720	62.80
0.1	0.3311	66.89	0.3189	68.11	0.3258	67.42	0.3086	69.14
0.2	0.2595	74.05	0.2394	76.06	0.2438	75.62	0.2269	77.31
0.3	0.1997	80.03	0.1753	82.47	0.1829	81.71	0.1607	83.93
0.4	0.1684	83.16	0.1478	85.22	0.1540	84.60	0.1311	86.89
0.5	0.1475	85.25	0.1258	87.42	0.1323	86.77	0.1193	88.07

Table 9.3: Our proposed method 2 detection rate of S-UNIWARD, WOW and Synch-HILL steganography methods at 0.2 bpc and 0.4 bpc payload embedding in the green channel compares with equal embedding in three channels.

Payload	S-UNIWARD		WOW		Synch-HILL	
	G%	RGB%	G%	RGB%	G%	RGB%
0.2	90.02	78.09	88.51	76.19	89.23	77.31
0.4	96.77	87.11	94.83	86.16	94.87	86.89

(c) show the *ROC* curves, illustrating the performance of our method 2. Finally, this experiment revealed that it is easier to detect a hidden message in only one channel than a message that is spread across all channels.

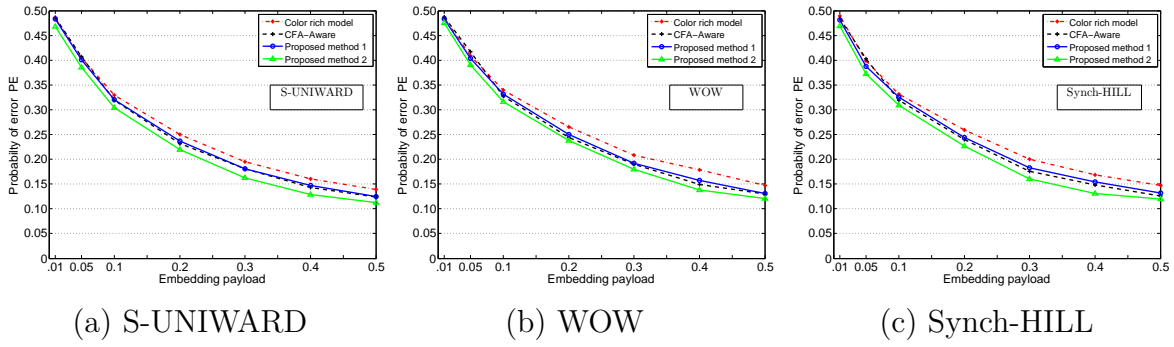


Figure 9.6: Average testing error \bar{P}_E as a function of the payload for (a) S-UNIWARD, (b) WOW and (c) Synch-HILL steganography methods, comparison between the steganalysis methods (Color Rich Model, CFA-aware features steganalysis, our method 1 [ACMM15] and our method 2 [ACMM16b]).

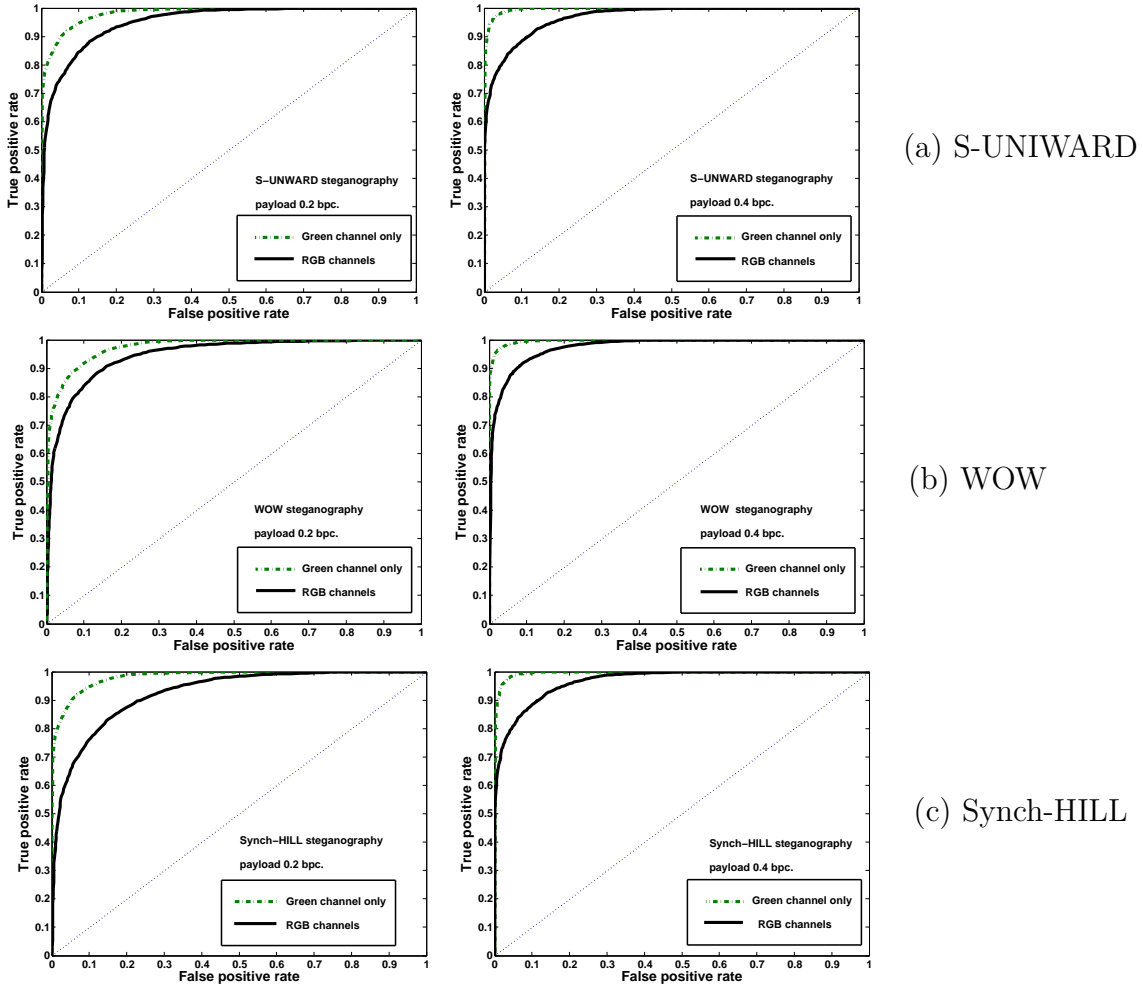


Figure 9.7: ROC curves using our proposed method 2 feature set, for (a) S-UNIWARD, (b) WOW and (c) Synch-HILL steganography methods for payloads 0.2 bpc (up) and 0.4 bpc (down), to compare the detectability when embedding messages in only one channel with embedding messages spread in all channels.

9.5 Summary

In this chapter, we have explained two new color image steganalysis methods the first method is based on color channel correlation and the second method is based on RGB channel geometric transformation measures. Starting from the Color Rich Model proposed by Goljan *et al.* [GFC14], we have shown that this method could be greatly enhanced by considering local deformation between channels. We have proposed to add to the Color Rich Model a new set of features based on local Euclidean and mirror transformation. The euclidean transformation, proposed by Abdulrahman *et al.* [ACMM15], is estimated by a first set of features derived from correlations between the gradients of red, green and blue channels. Since these features give the cosine of angles between gradients, we still do not know the direction of the rotation between two channel gradients. Then, we have shown that by taking into account mirror transformations, we can obtain the missing information of the direction of the local rotation. According to this analysis, we add a new set of features based on the sine of local rotation angles. These two sets of features are then incorporated in the Rich Model using co-occurrence matrices in order to obtain 6000 features. The first and second set gives 3000 features each [ACMM15]. The total feature set is formed from the Color Rich Model, plus the two new sets demonstrated in this work, in order to build a vector of a total of 24157 features. We used a quantization step with a set of values that differs from the Color Rich Models. All feature vectors are fed to the ensemble classifier. This methodology forms one of our contribution and resulted in the Journal publication [ACMM16b]. The ensemble classifier is used to detect the presence of hidden messages. Eventually, multiple steganalysis comparisons have been achieved between the proposed method and the initial Color Rich Model [GFC14] and CFA-aware features steganalysis method [GF15]. We have used three steganography methods (S-UNIWARD, WOW and Synch-HILL) with seven different payloads. All the experiments show that our new method outperforms the Color Rich Model and the CFA-aware feature steganalysis. Furthermore, in 2016 Tang *et al.* [TLLH16] has shown that our proposed methods [ACMM15] was the most reliable method in every case, even when the algorithm was embedding adaptively with a synchronization of the embedding between the color channels. Fig. 9.8 shown that our approach [ACMM15] can better

	BOSS-BIL				BOSS-BIC				BOSS-LAN				BOSS-NN			
	SRM	CRM	CCRM	SCCRM	SRM	CRM	CCRM	SCCRM	SRM	CRM	CCRM	SCCRM	SRM	CRM	CCRM	SCCRM
WOW	11.61	1.40	17.51	1.47	18.27	3.77	21.98	3.86	18.49	3.75	22.01	3.98	36.00	12.08	33.34	12.58
HILL	15.16	1.21	17.23	1.40	24.36	3.45	22.03	3.67	24.88	3.43	22.16	3.72	41.27	12.17	32.57	12.97
WOW-CMD	15.67	11.16	26.68	9.13	23.01	17.39	30.82	14.95	22.81	17.65	30.69	14.92	36.68	28.62	43.13	28.09
HILL-CMD	22.66	11.95	26.39	11.58	30.50	18.22	30.13	17.70	30.47	18.08	30.23	17.90	41.14	29.70	42.54	30.23
WOW-CMD-C	15.62	21.27	37.23	14.41	23.09	28.50	44.19	21.35	22.99	28.70	41.20	21.40	35.81	37.33	47.18	34.51
HILL-CMD-C	22.79	23.86	38.19	20.16	30.25	31.00	42.02	27.66	30.45	31.30	42.03	27.33	40.43	40.17	48.42	38.22

Figure 9.8: This table obtained from [TLLH16] shown that our approach SCCRM [ACMM15] can better detect the synchronize inter-channel steganography than Color Rich Model CRM steganalysis method [GFC14].

detect the synchronize inter-channel than Color Rich Model method [GFC14]. We also see that the BOSS-NN base is the hardest to detect than the (BOSS, demosaicking with Photoshop CS6, down-sampling with Nearest Neighbours).

10

STEGANALYSIS BASED ON STEERABLE GAUSSIAN FILTER

This chapter description the proposed method digital image steganalysis based on steerable Gaussian filter.

Contents

10.1 Introduction	166
10.2 Methodology description	166
10.2.1 Steerable Gaussian filters	167
10.2.2 Complete feature set	170
10.3 Experimental results	170
10.3.1 Performance of our proposed method	171
10.4 Summary	174

10.1 Introduction

Three color image steganalysis methods, based on a machine learning approach fed with rich model features are now well established. For these three methods, the machine learning algorithm is the Ensemble Classifier [KFH12b], and the rich features are: the Spatial Color Rich Model abbreviated to **CRM** [GFC14], the CFA-aware Rich Model abbreviated to **CFARM** [GF15], and the RGB Geometric Color Rich Model abbreviated to **GCRM** (RGB for the Red, Green and Blue channels) [ACMM16b]. Note that among these three features, the GCRM which is an extension of the CCRM (Correlation Color Rich Model [ACMM15]) seems to be the equivalent or better than the CRM and CFARM when recent adaptive grey-level embedding algorithms are used for embedding independently in each color channel. Additionally, note that a recent independent study has shown that CCRM was the most reliable method in every case, even when the algorithm was embedding adaptively with a synchronization of the embedding between the color channels [TLLH16]. At the beginning of 2016, **GCRM** is the natural choice for computing features that will be used for a color steganalysis (machine learning approach) for modern color embedding algorithms. Our method [ACMM16a] deals with color images steganalysis based on machine learning. The proposed approach enriches the features from the Color Rich Model by adding new features obtained by applying steerable Gaussian filters and then computing the co-occurrence of pixel pairs. Adding these new features to those obtained from Color Rich Models allows us to increase the detectability of hidden messages in color images. The Gaussian filters are angled in different directions to precisely compute the tangent of the gradient vector. Then, the gradient magnitude and the derivative of this tangent direction are estimated. This refined method of estimation enables us to unearth the minor changes that have occurred in the image when a message is embedded. The efficiency of the proposed framework is demonstrated on three steganographic algorithms designed to hide messages in images: S-UNIWARD, WOW, and Synch-HILL. Each algorithm is tested using different payload sizes. The proposed approach is compared to three color image steganalysis methods based on computation features and Ensemble Classifier classification: the Spatial Color Rich Model, the CFA-aware Rich Model and the RGB Geometric Color Rich Model.

10.2 Methodology description

In order to be less visible, most of the steganographic methods modify the pixel values in the texture/edge areas [PFB10, HFD14b, LWHL14]... Our method is to enrich the CRM method by introducing new sets of features obtained by applying steerable Gaussian

filters and then computing the co-occurrence of pixel pairs in eight different directions. The proposed features are composed of two distinctive sets. The first set, produced by Goljan *et al.* [GFC14], is made of 18,157 features. The second derivatives is made of 4406 features. In the first step, we computed a *tangent vector* of contour for each pixel and for each channel. This *tangent vector* corresponds to the edge direction and is orthogonal to the gradient magnitude image of each channel (R, G, B) ; it gives 2808 features. Then, in the second step, the co-occurrence matrices are computed, firstly, on the three gradient magnitude images and afterwards, on the three derivative images related to the *tangent vectors* for each channel (R, G, B); it gives 1598 features. The co-occurrences matrices are computed with triplets which means that co-occurrences have $(2 \times T + 1)^3$ bins. Four matrices are computed (horizontal right, and left, and vertical right, and left) and then summed. and four others matrices are computed (horizontal right, and left, and vertical right and left) and then summed. The two matrices are then concatenated in a feature vector. For each T values (except for the derivative image with T=3), the feature vectors of each channel are concatenated.

- For the gradient magnitude, $T \in \{2, 3\}$. This leads to a dimension $= 3 \times (2 \times 5^3) + 3 \times (2 \times 7^3) = 3 \times 250 + 3 \times 686 = 2808$ features,
- For the derivative image, T is equal to 1, 2 or 3. This leads to a dimension $= 3 \times (2 \times 3^3) + 3 \times (2 \times 5^3) + (2 \times 7^3) = 3 \times 54 + 3 \times 250 + 686 = 1598$ features.
- Note that for T=3, we sum the three co-occurrences matrices (R, G, B) channels instead of concatenate them (indeed, otherwise, bins values are too small for T=3).

This method is published in ACM Workshop on Information Hiding and Multimedia Security (IHMMSEC) 2016 [ACMM16b].

10.2.1 Steerable Gaussian filters

Due to multiple orientations, a filter bank allows us to better detect image features such as edges. One of the most popular filter banks is the steerable filters. As a solution to the above stated problem, Freeman and Adelson [FA91] introduced an elegant way for steerable filters that can be directed at specific angles using a linear combination of isotropic filters like Gaussian derivatives as discussed in the Section 3.5.1.

Let us note the basic derivatives of Gaussian filters $\partial \mathcal{G}_\sigma / \partial x$ and $\partial \mathcal{G}_\sigma / \partial y$ along the x -axis

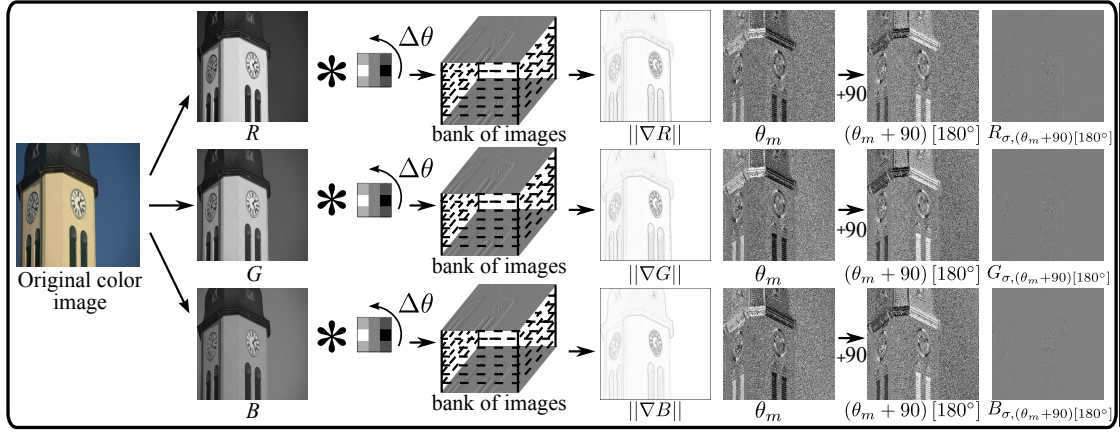


Figure 10.1: Features extraction: the image derivatives are extracted at orientation $(\theta_m + 90^\circ) [180^\circ]$ in each channel separately to compute a gradient and to estimate precise edges directions.

and y -axis respectively, for example:

$$\begin{cases} \frac{\partial \mathcal{G}_\sigma(x, y)}{\partial x} = \frac{-x}{2\pi\sigma^4} \cdot e^{-\frac{x^2 + y^2}{2\sigma^2}} \\ \frac{\partial \mathcal{G}_\sigma(x, y)}{\partial y} = \frac{-y}{2\pi\sigma^4} \cdot e^{-\frac{x^2 + y^2}{2\sigma^2}} \end{cases} \quad (10.1)$$

with σ the standard-deviation of the Gaussian filter.

Freeman and Adelson have shown that the first order directional Gaussian derivative $\mathcal{G}_{\sigma, \theta}$ at an angle θ can be generated by a linear combination of a rotation of the basic derivatives of isotropic Gaussian filters illustrated in Fig. 10.1:

$$\mathcal{G}_{\sigma, \theta}(x, y, \sigma) = \cos(\theta) \cdot \frac{\partial \mathcal{G}_\sigma}{\partial x}(x, y) + \sin(\theta) \cdot \frac{\partial \mathcal{G}_\sigma}{\partial y}(x, y). \quad (10.2)$$

The image derivative $I_{\sigma, \theta}$ is obtained by convolving the original grayscale image I with the oriented Gaussian kernels $\mathcal{G}_{\sigma, \theta}$:

$$I_{\sigma, \theta}(x, y) = (I * \mathcal{G}_{\sigma, \theta})(x, y). \quad (10.3)$$

Finally, the gradient magnitude $\|\nabla I(x, y)\|$ is calculated as the maximum absolute value response to the oriented operator $\mathcal{G}_{\sigma, \theta}$:

$$\|\nabla I(x, y)\| = \max_{\theta \in [0, 180[} (|I_{\sigma, \theta}(x, y)|), \quad (10.4)$$

$$\theta_m = \arg \max_{\theta \in [0, 180[} (|I_{\sigma, \theta}(x, y)|). \quad (10.5)$$

Note that θ_m , represents the *kernel angle* and it differs from the *gradient angle* which is equal to $(\theta_m + 90^\circ) [180^\circ]$.

In this work, the Gaussian filters are angled in different directions to compute the more precise gradient magnitude $\|\nabla I\|$ and its associated kernel angle θ_m . Thus, $\|\nabla I\|$ corresponds to the absolute value of the image derivative for the kernel angled at θ_m , as illustrated in Fig. 10.2. Note that $\|\nabla I\|$ and θ_m are different for each pixel of I . These techniques are applied to the three color channels R , G and B to obtain three gradient magnitude images $\|\nabla R\|$, $\|\nabla G\|$ and $\|\nabla B\|$ (see Fig. 10.1). As pointed out previously, the steganographic methods essentially modifies the pixel values in the textures and edge areas. For the edge areas, the embedding modifications have to be detected along the "isophote" lines i.e. along the curves of constant intensity when considering an image as a surface. This led us to consider the orthogonal vector to the gradient named the *tangent vector*, instead of the gradient; this means that the derivative along the edge must be computed. This derivative corresponds to the result of the convolution of the image with the steerable kernel angled at $(\theta_m + 90^\circ) [180^\circ]$ and is orthogonal to the kernel used for the gradient estimation (as illustrated in Fig. 10.3). The derivative image is named $I_{\sigma,(\theta_m+90)[180^\circ]}$.

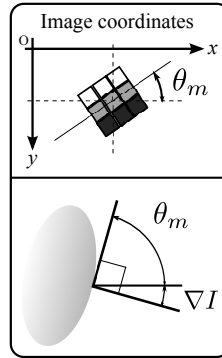


Figure 10.2: θ_m and η directions.

For a color image, each channel is considered separately. The tangent derivativesⁱ, as illustrated in Fig. 10.1, are respectively computed for each pixel at position (x, y) of each channel and named: $R_{\sigma,(\theta_m+90)[180^\circ]}(x, y)$ for the red, $G_{\sigma,(\theta_m+90)[180^\circ]}(x, y)$ for the green, and $B_{\sigma,(\theta_m+90)[180^\circ]}(x, y)$ for the blue channel. Fig. 10.1 shows an example of steerable Gaussian filters used to compute these new features. In this method, the Gaussian filters are angled in different directions to compute a precise gradient and a precise derivative along isophote lines. In order to detect the slight changes in the images, our experiments (see Section 10.3.1) leads to a $\sigma = 0.7$, with a filter support for the size 3×3 pixels, a rotation step for the filters bank $\Delta\theta = 10^\circ$, and a rotation range such as $\theta \in \{0^\circ, \dots, 180^\circ - \Delta\theta\}$ (it leads to 18 filter orientations as represented in Fig. 3.7).

ⁱAs these three images are derivatives, pixel values can be positives or negatives.

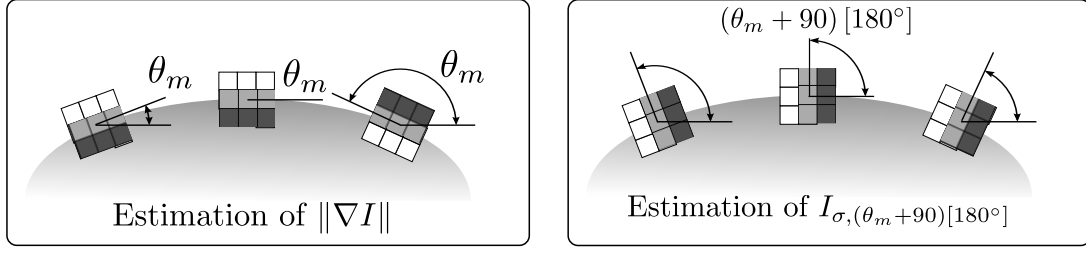


Figure 10.3: Positions of the steerable filters at the level of the edges to build the gradient image $\|\nabla I\|$ and image derivative $I_{\sigma,(\theta_m+90)[180^\circ]}(x, y)$.

10.2.2 Complete feature set

As explained in Fig.10.1, Eq.10.4 and detailed above, the co-occurrence matrices are completed from the three gradient magnitude images and also the three derivative images using their edge directions:

$$\begin{cases} \|\nabla R\|, & \|\nabla G\|, & \|\nabla B\|, \\ R_{\sigma,(\theta_m+90)[180^\circ]}, & G_{\sigma,(\theta_m+90)[180^\circ]}, & B_{\sigma,(\theta_m+90)[180^\circ]}. \end{cases}$$

Before we make the co-occurrence computation, different truncations are applied. For the gradient magnitude images, the truncation $T \in \{2, 3\}$, and for the derivative images, $T \in \{1, 2, 3\}$. Thus this leads to 2 triplets of quantized-truncated gradient magnitude images, and 3 triplets of quantized-truncated derivative images. Then, the pairs co-occurrence matrices are computed, like the Subtractive Pixel Adjacency Model (SPAM) [PBF10], with eight directions for the scan images. These direction feature subsets are as follows: $F_h^{\rightarrow}, F_h^{\leftarrow}, F_v^{\uparrow}, F_v^{\downarrow}, F_d^{\nearrow}, F_d^{\nwarrow}, F_{md}^{\nearrow}, F_{md}^{\nwarrow}$. 2808 features are created by the gradient magnitude and 1598 by the derivative images. All features are gathered in to a one dimension vector to erect a dimensionality of 4406 features. As a final set for the proposed method, 22,563 features are obtained, by concatenating these features with those obtained from CRM [GFC14].

10.3 Experimental results

All the experiments were carried out on 10,000 color images of size 512×512. All detectors were trained as binary classifiers implemented using the ensemble classifierⁱⁱ [KFH12b] with default settings.

ⁱⁱThe Ensemble classifier is available at <http://dde.binghamton.edu/download/ensemble>.

Three spatial domain steganography algorithms are used to produce stego images:

- Spatial UNiversal WAvelet Relative Distortion (S-UNIWARDⁱⁱⁱ) [HFD14b],
- Wavelet Obtained Weights (WOW^{iv}) [HF12b],
- Synchronizing Selection Channel (Synch-HILL^v) [DF15].

As explained in [MBL⁺09], the steganography methods have an important impact on the performance of the general methodology. These algorithms are used to embed messages into color images by decomposing the R , G and B channels like three gray-scale images and embedding the same proportion payload into each channel. Finally, 10,000 color images were used to test each of the seven different payload sizes: $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ Bit Per Channel (**BPC**).

10.3.1 Performance of our proposed method

To evaluate the performance of the proposed method, all steganalyzers were implemented as binary classifiers using the ensemble classifier [KFH12b]. In this method, the detection accuracy is measured by using the average of testing errors under equal priors: $\bar{P}_E = \min_{P_{FA}} \frac{1}{2} [P_{FA} + P_{MD}(P_{FA})]$, where, P_{FA} represents the false alarm probability and P_{MD} the missed detection probability. 5000 cover images are randomly chosen from the database

ⁱⁱⁱThe Matlab version of S-UNIWARD is available at http://dde.binghamton.edu/download/stego_algorithms.

^{iv}The Matlab version of WOW is available at http://dde.binghamton.edu/download/stego_algorithms.

^vThe Matlab version of Synch-HILL is available at http://dde.binghamton.edu/download/stego_algorithms.

Table 10.1: The probability of error \bar{P}_E to determine the efficient standard deviation (σ) and angle step ($\Delta\theta$) employed for steerable Gaussian filters using S-UNIWARD steganography method payload 0.3 bpc.

$\Delta\theta$	σ	Mask size	\bar{P}_E	Detection rate
10°	0.7	3×3	0.1559 ± 0.0022	84.41 %
10°	1	5×5	0.1896 ± 0.0031	81.04 %
10°	2	10×10	0.2028 ± 0.0037	79.72 %
10°	3	15×15	0.2539 ± 0.0036	76.41 %
5°	0.7	3×3	0.1768 ± 0.0026	82.32 %
10°	0.7	3×3	0.1559 ± 0.0022	84.41 %
15°	0.7	3×3	0.1602 ± 0.0026	83.98 %
20°	0.7	3×3	0.1653 ± 0.0019	83.47 %
30°	0.7	3×3	0.1854 ± 0.0027	81.46 %
45°	0.7	3×3	0.1893 ± 0.0012	81.07 %
90°	0.7	3×3	0.1996 ± 0.0031	80.04 %

Table 10.2: Error probability \bar{P}_E and the detection rate $\mathcal{P}_D\%$ for four steganalysis methods.

Steganography Method	Payload (bpc)	Proposed Method \bar{P}_E	22,563 Dim $\mathcal{P}_D\%$	CRM \bar{P}_E	18,157 Dim $\mathcal{P}_D\%$	CFARM \bar{P}_E	27,460 Dim $\mathcal{P}_D\%$	GCRM \bar{P}_E	24,157 Dim $\mathcal{P}_D\%$
S-UNIWARD	0.01	0.4664	53.36	0.4841	51.59	0.4863	51.37	0.4680	53.20
	0.05	0.3835	61.65	0.4045	59.55	0.4072	59.28	0.3859	61.41
	0.1	0.2984	70.16	0.3298	67.02	0.3194	68.06	0.3037	69.63
	0.2	0.2164	78.36	0.2498	75.02	0.2317	67.83	0.2191	78.09
	0.3	0.1559	84.41	0.1947	80.53	0.1806	81.94	0.1623	83.77
	0.4	0.1202	87.98	0.1599	84.01	0.1429	85.71	0.1289	87.11
	0.5	0.1117	88.83	0.1386	86.14	0.1239	87.61	0.1124	88.76
WOW	0.01	0.4687	53.13	0.4850	51.50	0.4875	51.25	0.4753	52.47
	0.05	0.3854	61.46	0.4092	59.08	0.4174	58.26	0.3906	60.94
	0.1	0.3091	69.09	0.3397	66.03	0.3275	67.25	0.3161	68.39
	0.2	0.2269	77.31	0.2654	73.46	0.2440	75.60	0.2381	76.19
	0.3	0.1685	83.15	0.2081	79.19	0.1895	81.05	0.1793	82.07
	0.4	0.1377	86.23	0.1783	82.17	0.1487	85.13	0.1384	86.16
	0.5	0.1206	87.94	0.1473	85.27	0.1296	87.04	0.1207	87.93
Synch-HILL	0.01	0.4651	53.49	0.4893	51.07	0.4843	51.57	0.4687	53.13
	0.05	0.3647	63.53	0.3991	60.09	0.4030	59.70	0.3720	62.80
	0.1	0.2946	70.54	0.3311	66.89	0.3189	68.11	0.3086	69.14
	0.2	0.2113	78.87	0.2595	74.05	0.2394	76.06	0.2269	77.31
	0.3	0.1536	84.64	0.1997	80.03	0.1753	82.47	0.1607	83.93
	0.4	0.1294	87.06	0.1684	83.16	0.1478	85.22	0.1311	86.89
	0.5	0.1125	88.75	0.1475	85.25	0.1258	87.42	0.1193	88.07

for the training sets and 5000 stego images for the testing sets. The ensemble classifiers apply a vote to estimate the error of detection. This process is repeated 10 times to obtain \bar{P}_E which quantify the detectability and are collected for each method and each payload in order to evaluate the steganalysis method.

As explained in Section 10.2.1, the experiments were run in such a way as to find the best filter bank parameters: the scale of the steerable filters σ (Eq. 10.2) and the rotation step ($\Delta\theta$) with angles evenly drawn from 0° to 180° . These experiments have been led using S-UNIWARD with a payload size of 0.3 bpc with the use of the 22,563 features obtained by concatenating the CRM with the gradient features. Table 10.1 shows that $\sigma = 0.7$ with $\Delta\theta = 10^\circ$ corresponds to the optimal parameters for the steerable filters operation in this steganalysis work because, compared to the other parameters, they bring the best detection rate. The experimental results are given in Table 10.2. Three algorithms have been tested: S-UNIWARD, WOW and Synch-HILL with different relative payloads sizes: $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Furthermore, the proposed method is tested against three other approaches: CRM [GFC14], CFARM [GF15] and GCRM [ACMM16b].

Table 10.2 demonstrates that the proposed method registered the highest performance. As an example, detection rates for, at a time, S-UNIWARD, WOW and Synch-HILL for a payload of 0.5 bpc are 88.83%, 87.94% and 88.75% respectively using the proposed approach, to the contrary of the other three compared methods. The CRM method [GFC14] achieved 86.14%, 85.27% and 85.25% respectively. The CFARM method [GF15] achieved 87.61%, 87.04% and 87.42% respectively.

Additionally, the GCARM method [ACMM16b] achieved 88.76%, 87.93% and 88.07% respectively. Moreover, curves in Fig. 10.4 for S-UNIWARD, WOW, and Synch-HILL steganography methods illustrate the comparison between the proposed method and the comparison methods. As a result, the error probability of the proposed method is less than the three steganalysis methods. This performance is due to the Gaussian filters bank, created by the steerable filters, which allow a more precise estimation of the gradient and its associated tangent vector. In order to increase the detectability rate of the GCRM method [ACMM16b], another experiment has been performed by concatenating the GCRM [ACMM16b] features with the new proposed features in one dimensional vector to produce 28,563 features. These new dimensional vectors achieved 85.03% for S-UNIWARD, and 85.07% for Synch-HILL steganography methods payload 0.3 bpc as a detection rate. It obtains a difference of 1.26%, 1.14% respectively more than GCRM [ACMM16b] alone and this result is close (slightly better) to our proposed approach results (less than 1%).

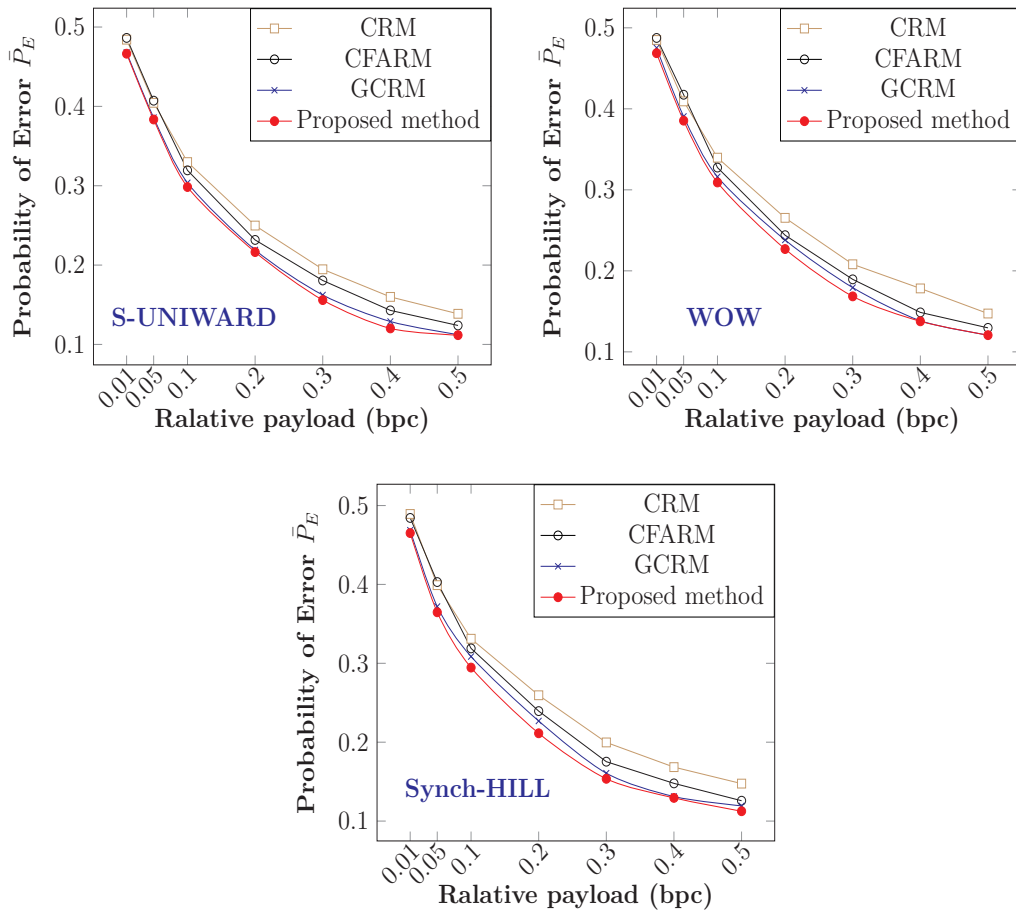


Figure 10.4: Error probability \bar{P}_E as a function of the payload for three steganography methods.

10.4 Summary

To summarize this chapter, here new method for color image steganalysis has been presented. Applying a Gaussian filters bank, to an order of 1, in different directions, enabled us to detect the slight changes in the images which occurred as a result of embedding the message. This proposed method treats the three color channels separately. Firstly, as steerable filters estimate precisely the edge directions in images, features correspond to the three image derivatives along the edges and the three gradient magnitude images. Secondly, features are extracted from these six images using the co-occurrence matrices of pixel pairs. Finally, our proposed features are integrated with the CRM features [GFC14] to get the new approach. This methodology forms one of our contribution and resulted in the publication [ACMM16a]. To evaluate the performance of the new approach, the embedding algorithms used are S-UNIWARD, WOW, and Synch-HILL at different payloads. Experimental results show that fusing proposed features with those obtained by CRM allows in the majority of cases, the detectability of hidden messages in the color images. Additionally, the new approach achieved higher detection rates than the three recent steganalysis approaches: CRM, CFARM, and GCRM. This observed detection improvement is due to a fine estimation of the tangent vector which is used for the estimation of the image derivatives in the edges directions. The proposed features allow the Ensemble Classifier to reveal the hidden message between the stego and cover images.

Part IV

CONCLUSION AND PERSPECTIVES

CONCLUSION AND PERSPECTIVES

The rest of this chapter summarises the research objectives achieved by this thesis and the main conclusions and provides discussions for future research.

Contents

11.1 Conclusion	178
11.2 Perspectives	179

11.1 Conclusion

The overall goal of this thesis was to investigate on feature extraction using different amounts of existing knowledge from image filtering to more advanced domain knowledge such as oriented filter bank. In addition to studying different error measure methods, based on the theory of supervised edge detection dissimilarity evaluations. This goal was achieved by developing a number of new methods to contours evaluation, corners detection and color image steganalysis. In this thesis, a new techniques are proposed to evaluate filtering edge detection. Moreover, we have proposed a new approach for corner detection based on the combination of oriented half kernels and homogeneity kernels. The half Gaussian kernels (HGK) allow to detect a relevant strength of a corner while the IRON indicates the more homogeneous directions. A such combination enables to remove undesirable corners near contour areas which could be considered as a ideal feature by the HGK. Informations captured by the directional kernels enable to describe precisely all the grayscale variations and the directions of the crossing edges around the considered pixel. Furthermore, In this thesis, we have proposed and developed three color image steganalysis methods, based on a machine learning approach. The GCRM [ACMM15] which is an extension of the CCRM (Correlation Color Rich Model [ACMM15]) seems to be the equivalent or better than the color rich model method and CFARM when recent adaptive grey-level embedding algorithms are used for embedding independently in each color channel. Additionally, note that a recent independent study has shown that CCRM was the most reliable method in every case, even when the algorithm was embedding adaptively with a synchronization of the embedding between the color channels [TLLH16]. Moreover, proposed a new approach [ACMM16b] enriches the features from the Color Rich Model by adding new features obtained by applying steerable Gaussian filters and then computing the co-occurrence of pixel pairs. Adding these new features to those obtained from Color Rich Models allows us to increase the detectability of hidden messages in color images. The Gaussian filters are angled in different directions to precisely compute the tangent of the gradient vector. Then, the gradient magnitude and the derivative of this tangent direction are estimated. This refined method of estimation enables us to unearth the minor changes that have occurred in the image when a message is embedded.

- In chapter 4 a new normalized supervised edge detection evaluation method was proposed, comparing a ground truth contour image, the candidate contour image and their associated spacial nearness [AMM17d]. Several referenced-based boundary detection evaluations are detailed, pointing their advantages and disadvantages through concrete examples of edge images [AMM17d, AMM17a].
- In chapter 5 Proposed a new technique to evaluate filtering of edge detection methods involving the minimum score of the considerate measures [AMM17a].

- In chapter 7 proposed a novel method for corner detection based on the combination of directional derivative and homogeneity kernels [AMM17c], [MAM17].
- In chapter 9 extract new features from color image for development a color image steganalysis based on color feature correlation and based on geometric measures obtained by the sine and cosine of gradient angles between all the color channels [ACMM15, ACMM16b]. and machine learning. Features are extracted from the channel correlation, and co-occurrences.
- In chapter 10 important contribution was present for color steganalysis approach by development of a new color steganalysis based on Steerable Gaussian Filters Bank. This approach enriches the features from the Color Rich Model by adding new features obtained by applying steerable Gaussian filters and then computing the co-occurrence of pixel pairs [ACMM16a]. Intensive experiments reveal that our steganalysis approaches is able to better detect the color adaptive steganographic approaches.

11.2 Perspectives

In the summary of each chapter, we discussed various theoretical and practical possibilities for the improvement of our approaches. In this section we aim to present these perspectives, that could improve the obtained results and also to start out new research work. In the future we would like to include the following improvements:

- In this thesis we presents a supervised edge detection assessment methods. In the future, our work program agenda is to compare more different edge detectors with their more different parameters. To deeply compare the more robustness edge detection algorithms and proposed a new measure in object recognition.
- In recent years, deep learning used in different areas, the use of deep learning networks challenges traditional two step approaches (feature extraction, and use of a classifier). We plan in the future, to study and use deep learning machine to automatically detect the edges, corners and junctions, also use deep learning with image steganalysis.
- In the future, we plan to study junctions types to enhance the our proposed method corner detection in a way to classify in a 2π periodic form features of type L, Y or X junctions.
- Steganography by modification is the art of modifying a digital media (image, sound, video, ...) in order to hide a secret message most often unrelated to the media, and so that changes are statistically undetectable. In the future works we consist of better

understanding WOW-CMD-C or HILL-CMD-C [TLLH16] embedding algorithms, which synchronize the color selection channel during the embedding process.

- Because the Facebook is available now for more than two milliards people around the world, we plan in the future to study exactly what Facebook is doing to image during compression by detailed comparison of the uploaded and downloaded images. On other hand, propose a new stegnography method that can hide texts inside color image. Also, propose a new steganalysis method to detect hidden data from the modified image in the Facebook.

12

RÉSUMÉ EN FRANCAIS

Ce chapitre souligne brièvement la motivation et les objectifs de cette thèse.

Contents

12.1 Introduction	182
12.2 Motivations	184
12.3 Contributions	185
12.4 Conclusion	186
12.5 Perspectives	186

12.1 Introduction

En informatique, tous les systèmes d'image numérique, en particulier les structures automatiques de traitement de l'information, doivent être évalués avant d'être développés, principalement pour des applications industrielles ou des données médicales. En effet, il y a une littérature torrentielle couvrant différents aspects du traitement des images. Le traitement de l'image numérique est lié au développement d'un système numérique qui effectue des opérations sur une image numérique en utilisant un ordinateur numérique [GW02]. Les méthodes de traitement d'image numérique s'intéressent à deux tâches d'application majeures. La première est l'amélioration de l'information picturale pour l'interprétation humaine, tandis que la seconde est le traitement de données pour la perception automatique des machines [Dou09].

Une image numérique contient des informations différentes, telles que les objets, la couleur et l'orientation. La discrimination des objets à partir de leur arrière-plan est le premier problème qui se pose avant tout traitement ultérieur. Afin d'extraire le contour d'un objet, les arêtes qui forment cet objet doivent être détectées et ce fait révèle l'importance constitutionnelle de la détection de bord en vision par ordinateur et traitement d'image. Les fonctions de détection de bord supportent une large gamme d'applications telles que la reconnaissance, la compression, l'amélioration de l'image, la restauration, l'enregistrement, la récupération, le tatouage, la stéganographie, la stéganalyse et etc. [OH10].

Dans la littérature, plusieurs travaux de recherche développent des techniques de détection de bord. En général, une méthode de détection de bord peut être divisée en trois étapes. Tout d'abord, un processus de réduction du bruit est effectué afin d'obtenir de meilleures performances de détection de bord. Cette réduction de bruit est habituellement obtenue en effectuant un filtre passe-bas car le bruit additif est normalement un signal de haute fréquence. Vu que les bords sont des signaux haute fréquence, ils peuvent être supprimés pendant le processus de réduction du bruit. Par conséquent, la sélection du filtre approprié avec le meilleur paramètre est importante pour préserver les informations de bord. Pour cette raison, cette thèse fournit plus de détails sur la source de bruit dans le pipeline de construction d'images numériques. Dans la deuxième étape, un filtre passe-haut tel qu'un filtre orienté est généralement effectué pour trouver les bords. Enfin, un processus de localisation de bord est effectué pour identifier les bords réels, qui se distinguent des réponses similaires provoquées par le bruit.

Une partie de cette thèse cible l'évaluation des méthodes de détection de bord. En effet, plusieurs références de méthodes d'évaluation sont étudiées, tout en indiquant leurs avantages et leurs inconvénients à travers des exemples concrets d'images de bord et les appliquant à une analyse objective. D'autre part, une nouvelle mesure de qualité de carte de bord supervisée est proposée. En outre, cette thèse présente et démontre comment construire une nouvelle carte de bord de vérité terrestre qui est marquée d'une manière

semi-automatique en images réelles. Afin d'évaluer les performances des détecteurs de filtrage/détection de bord de rampe, une technique est proposée pour évaluer les méthodes de détection de bord de filtrage impliquant le score minimal des mesures considérables.

Dans cette thèse, nous nous concentrons sur des filtres orientés tels que le filtre gaussien orientable et le noyau demi-gaussien (HGK). En outre, nous avons proposé une nouvelle méthode en combinant deux filtres : le noyau demi-gaussien avec un filtre asymétrique calculant l'homogénéité le long des bords. D'autre part, les réponses maximales du HGK indiquent les directions (2II Périodiques) des bords. En plus, la variance orientée détermine si les directions de la maxima du HGK correspond aux bords ou à d'autres types de pixels (texture, région homogène, etc.).

La stéganalyse de l'image peut donc être considérée comme un processus de reconnaissance de formes, et considère deux classifications des images de tests : image originale et image stego. Le problème clé pour la stéganalyse, tout comme pour la reconnaissance de motifs, est l'extraction de caractéristiques. En effet, les caractéristiques doivent être sensibles au processus de dissimulation des données. En d'autres termes, les caractéristiques devraient être plutôt différentes pour l'image sans message caché et pour l'image stego.

Cette thèse propose aussi trois nouvelles méthodes de stéganalyse en couleurs basées sur l'extraction de caractéristiques en utilisant des techniques d'apprentissage. La première méthode étudie la corrélation entre les canaux d'image en couleurs et l'extraction de caractéristiques. La seconde méthode étudie deux types de caractéristiques basées sur la corrélation entre les canaux d'image en couleurs. Le premier type de caractéristiques reflète les transformations euclidiennes locales et le second reflète les transformations miroir. Ces transformations géométriques se calculent à la base du sinus et cosinus des angles gradients entre tous les canaux d'image en couleurs. La troisième méthode utilise la banque de filtrage d'orientation pour détecter les messages cachés dans l'image, et se base sur l'extraction de caractéristiques à partir d'images en couleurs avec des techniques et d'apprentissage. Cette nouvelle méthode de stéganalyse en couleurs utilise des filtres gaussiens orientés. Dans cette méthode, les filtres gaussiens sont inclinés dans différentes directions pour calculer précisément la tangente du vecteur gradient. Ensuite, la grandeur du vecteur gradient et la dérivée de cette direction tangente sont estimées. Cette méthode d'estimation raffinée nous permet de découvrir les changements mineurs qui se sont produits dans l'image lorsqu'un message est intégré. Nous avons calculé un vecteur tangent de contour pour chaque pixel et pour chaque canal. Ce vecteur tangent correspond à la direction du bord, et est orthogonal à l'image de la grandeur du vecteur gradient pour chaque canal de couleur.

12.2 Motivations

Dans le domaine de traitement d'image, la détection de bord représente une problématique clé qui a de nombreuses applications. En effet, une méthode efficace de détection de contour doit localiser les bords correctement avec un minimum de pixels mal classés. Les bords représentent une caractéristique de bas niveau, tandis que les caractéristique de haut niveau sont définies en termes d'objet dans l'image. Les contours sont des points dans l'image projetée d'un objet où il y a une discontinuité de profondeur ou une discontinuité d'orientation de surface sur l'objet.

De nombreuses approches d'extraction de caractéristiques sont basées sur des points d'intérêt, tels que les coins, pour localiser les caractéristiques en deux dimensions. Dans la littérature, on trouve plusieurs algorithmes développés, mais uniquement quelques algorithmes fournissent une comparaison de performance objective. En effet, l'évaluation devrait produire un résultat qui est en corrélation avec la qualité perçue de l'image de bord, et qui repose sur le jugement humain. En d'autres termes, une carte de bord fiable devrait caractériser toutes les structures pertinentes d'une image. D'autre part, un minimum de pixels parasites ou de trous (oversights) doit être créé par le détecteur de bord en même temps. Par conséquent, une évaluation efficace peut être utilisée pour évaluer et améliorer un algorithme, ou pour optimiser les paramètres du détecteur de bord [HSSB97].

Les filtres orientés sont utilisés dans de nombreuses approches de traitement de la vision et de l'image, telles que la détection de bord, l'analyse de texture, la compression d'image, la détection d'objets, la stéganographie et l'amélioration de l'image [NA12].

Pour être moins visible, la plupart des méthodes stéganographiques modifient les valeurs de pixels dans les zones de texture/bord [PFB10, HFD14b, LWHL14]...

Durant de nombreuses années, la dissimulation de l'information a stimulé l'imagination des chercheurs. La stéganographie est l'une des techniques qui peuvent être utilisées pour cacher un secret [HH03]. À titre d'exemple, en 2001, un journal international intitulé USA Today, a annoncé que la stéganographie était utilisée par les terroristes [Kel01], afin de planifier l'attaque du 11 septembre 2001. Cela a également été confirmé par d'autres rapports [KOL01, Tib02].

Dans le passé, beaucoup de gens ne croient pas que les techniques de stéganographie ont été utilisées par les criminels et les terroristes. Maintenant, cette idée est acceptée et comprise. Pour cette raison, les techniques de stéganalyse doivent être développées pour détecter les communications suspectes.

La stéganographie et la stéganalyse ont prospéré à l'ère numérique. De nombreuses techniques intéressantes de stéganalyse ont été créées en fonction des images en niveaux de gris, et la stéganalyse poursuit son évolution. En particulier, il est nécessaire de détecter un

message caché dans les images en couleurs [KBB⁺13]. Pour ces raisons, une des principales motivations de cette thèse est de développer une nouvelle méthode de stéganalyse pour détecter un message caché dans des images numériques en couleurs.

12.3 Contributions

Les contributions clés de cette thèse se basent sur l'extraction de caractéristiques en utilisant des filtre orientés. D'une part, on propose de nouvelles méthodes pour la détection des contours et des coins, et en plus leur évaluation. D'autre part, on développe de nouvelles méthodes de stéganalyse pour détecter les messages cachés dans l'image, en extractant des caractéristiques des images en couleurs avec des techniques d'apprentissage. On résume les contributions de cette thèse dans la liste ci-dessous :

- Une nouvelle méthode pour la détection de coins basée sur la combinaison de dérivées directionnelles et de grains d'homogénéité [AMM17c].
- Une nouvelle technique pour évaluer le filtrage des méthodes de détection des arêtes impliquant le score minimal des mesures préventives [AMM17a].
- Une nouvelle méthode d'évaluation de détection de bordure supervisée normalisée, en comparant une image de contour de vérité terrestre, l'image de contour de candidat et leur proximité spatiale associée [AMM17d].
- Plusieurs méthodes d'évaluation de la détection des limites identifiées dans l'état de l'art sont discutées, en détaillant leurs avantages et leurs inconvénients à travers des exemples concrets d'images de bord [AMM17d], [AMM17a].
- Une nouvelle stéganalyse d'image en couleurs basée sur la corrélation des caractéristiques de couleurs avec des techniques d'apprentissage. Les caractéristiques sont identifiées à travers la corrélation du canal et les co-occurrences [ACMM15].
- Une nouvelle stéganalyse d'image en couleurs basée sur une banque de filtres gaussiens orientés. Cette approche enrichit les caractéristiques du modèle riche en couleurs en ajoutant de nouvelles caractéristiques obtenues en appliquant des filtres gaussiens orientés, puis en calculant la co-occurrence de paires de pixels [ACMM15].
- Des expériences intensives révèlent que nos approches de stéganalyse permettent de mieux détecter les approches stéganographiques appliquées en couleurs.

12.4 Conclusion

L'objectif global de cette thèse était d'enquêter sur les caractéristiques l'extraction en utilisant différentes quantités de connaissances existantes depuis le filtrage d'image jusqu'à des connaissances de domaine plus avancées telles qu'une banque de filtres orientés. En plus d'étudier différentes méthodes de mesure d'erreur, basées sur la théorie des évaluations de dissimilarité de détection de bord supervisé. Cet objectif a été atteint en développant un certain nombre de nouvelles méthodes pour l'évaluation des contours, la détection des coins et la stéganalyse des images couleur. Dans cette thèse, on propose de nouvelles techniques pour évaluer la détection de bord de filtrage en se basant sur la théorie des évaluations de dissimilation de détection de bord supervisées. En outre, on propose une nouvelle approche pour la détection de coins en fonction de la combinaison des demi-noeuds orientés et des grains d'homogénéité. Les grains demi-gaussiens (HGK) permettent de détecter une résistance importante d'un coin tandis que le filtre IRON indique les directions les plus homogènes. Une telle combinaison permet d'éliminer les coins indésirables près des zones de contours qui pourraient être considérées comme une caractéristique idéale par le HGK. Les informations capturées par les noyaux directionnels permettent de décrire précisément toutes les variations de niveaux de gris et les directions des bords de passage autour du pixel considéré.

En outre, une nouvelle approche proposée [ACMM16b] enrichit les fonctionnalités du modèle riche en couleurs en ajoutant de nouvelles caractéristiques obtenues en appliquant des filtres gaussiens orientés, puis en calculant la co-occurrence de paires de pixels. L'ajout de ces nouvelles fonctionnalités à celles obtenues à partir de du modèle riche en couleurs nous permet d'augmenter la détection des messages cachés dans les images en couleurs. Les filtres gaussiens sont inclinés dans différentes directions pour calculer précisément la tangente du vecteur gradient. Ensuite, la grandeur de gradient et la dérivée de cette direction tangente sont estimées. Cette méthode d'estimation raffinée nous permet de découvrir les changements mineurs qui se sont produits dans l'image lorsqu'un message est intégré.

12.5 Perspectives

Dans le résumé de chaque chapitre, nous avons discuté de diverses possibilités théoriques et pratiques pour l'amélioration de nos approches. Dans cette section, nous visons à présenter ces perspectives, qui pourraient améliorer les résultats obtenus et aussi commencer de nouveaux travaux de recherche. Dans les perspectives, on vise les améliorations suivantes :

- Dans cette thèse, on présente des méthodes d'évaluation de détection de bord supervisées. Comme perspective, on va ajouter de nouveaux détecteurs de bord pour les comparer avec leurs différents paramètres. Cette comparaison permet d'évaluer la robustesse de plusieurs algorithmes de détection de bord et utiliser la nouvelle mesure dans la reconnaissance d'objet.
- Récemment, la technique d'apprentissage profond (Deep Learning) est appliquée dans différents domaines, en défiant les approches traditionnelles qui se basent sur deux étapes (extraction des caractéristiques et utilisation d'un classificateur). On prévoit l'application de la technique d'apprentissage profond (Deep Learning) pour détecter automatiquement les bords, les coins et les jonctions ou la stéganalyse.
- On vise également l'étude de plusieurs types de jonctions pour améliorer notre détection de coins afin de classifier avec des caractéristiques de forme périodique 2π du type de jonctions L, Y ou X.
- La stéganographie est l'art de modifier un média (image, son, vidéo, ...) afin de cacher un message secret le plus souvent non lié aux médias, de sorte que les changements sont statistiquement indétectables. La stéganalyse est l'art de détecter la présence d'un message secret. Dans de futurs travaux, on compte de mieux comprendre les algorithmes d'intégration *WOW – CMD – C* ou *HILL – CMD – C* [TLLH16], qui synchronise le canal de sélection de couleurs pendant le processus d'intégration.
- Plus que deux milliards de personnes utilisent Facebook dans le monde. Pour cette raison, on cible l'étude de méthodes de compression appliquées dans Facebook durant l'émission (upload) et la réception (download) de l'image. Ceci nous permet de développer de nouvelles méthodes de stéganographie pour cacher des messages à l'intérieur de l'image, et de nouvelles méthodes de stéganalyse pour détecter ces messages cachés dans les images de Facebook.

BIBLIOGRAPHIE

- [ACG10] Wadood Abdul, Philippe Carré, and Philippe Gaborit. Human visual system-based color image steganography using the contourlet transform. *Media Forensics and Security*, 12, 2010.
- [ACMM15] Hasan Abdulrahman, Marc Chaumont, Philippe Montesinos, and Baptiste Magnier. Color image steganalysis using correlations between rgb channels. In *Proceedings of the 10th International Conference on Availability, Reliability and Security (ARES), 4th Int. Workshop on Cyber Crime (IWCC), Toulouse, France*, pages 448–454. IEEE, Aug. 24– 28, 2015.
- [ACMM16a] Hasan Abdulrahman, Marc Chaumont, Philippe Montesinos, and Baptiste Magnier. Color image steganalysis based on steerable gaussian filters bank. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security (IHMMSEC)*, pages 109–114. ACM, 2016.
- [ACMM16b] Hasan Abdulrahman, Marc Chaumont, Philippe Montesinos, and Baptiste Magnier. Color images steganalysis using rgb channel geometric transformation measures. *Wiley Journal on Security and Communication Networks*, (DOI : 10.1002/sec.1427) :12 pages, Feb. 2016.
- [All04] David Alleysson. 30 ans de démosaïçage - 30 years of demosaicing. *Traitement du signal- Signal processing*, 21(6) :561–581, 2004.
- [AMFM09] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. From contours to regions : An empirical evaluation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2294–2301. IEEE, 2009.
- [AMFM11] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5) :898–916, 2011.
- [AMM17a] H. Abdulrahman, B. Magnier, and P. Montesinos. From contours to ground truth : How to evaluate edge detectors by filtering. In *the 25th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), Pilsen, Czech Republic, 29 May - 2 June, 2017. published in the Journal of WSCG, Vol.25, No.1-2, pages 133-142, ISSN 1213-6972, ISBN 978-80-86943-43-5*. Journal of WSCG, 2017.

- [AMM17b] H. Abdulrahman, B. Magnier, and P. Montesinos. A new objective supervised edge detection assessment using hysteresis thresholds. In *First International Workshop on Brain-Inspired Computer Vision (WBICV2017), held as part of the conference ICIAP2017, 11-15 September, Catania, Sicily (Italy). volume 10590, 11 pages, chapter 1, of the Lecture Notes in Computer Science Series. Springer*, 2017.
- [AMM17c] H. Abdulrahman, B. Magnier, and P. Montesinos. Oriented half kernels for corner detection. In *Proceedings of the 25th European Signal Processing Conference (EUSIPCO2017), Kos Island, Greek, 28 August - 2 September, ISBN 978-0-9928626-7-1, pages 808-812,, 2017*.
- [AMM17d] Hasan Abdulrahman, Baptiste Magnier, and Philippe Montesinos. A new normalized supervised edge detection evaluation. In *IbPRIA 2017, 8th Iberian International Conference on Pattern Recognition and Image Analysis, Faro, Portugal, June 20-23, 2017*, pages 203–213. Springer, 2017.
- [AMS⁺06] T. Aach, C. Mota, I. Stuke, M. Muhlich, and E. Barth. Analysis of superimposed oriented patterns. *IEEE TIP*, 15(12) :3690–3700, 2006.
- [AMY10] George Abboud, Jeffrey Marean, and Roman V Yampolskiy. Steganography and visual cryptography in computer forensics. In *Systematic Approaches to Digital Forensic Engineering (SADFE), 2010 Fifth IEEE International Workshop on*, pages 25–32. IEEE, 2010.
- [AOAKOA12] C Abikoye Oluwakemi, S Adewole Kayode, and J Oladipupo Ayotunde. Efficient data hiding system using cryptography and steganography. *International Journal of Applied Information Systems (IJ AIS), Foundation of Computer Science FCS, New York, USA, ISSN*, pages 2249–0868, 2012.
- [AP79] I. E. Abdou and W. K. Pratt. Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proc. of the IEEE*, 67 :753–763, 1979.
- [AP98] Ross J Anderson and Fabien AP Petitcolas. On the limits of steganography. *IEEE Journal of Selected Areas in Communications, Special Issue on Copyright and Privacy Protection*, 16(4) :474–481, 1998.
- [ASH05] David Alleysson, Sabine Süsstrunk, and Jeanny Hérault. Linear demosaicing inspired by the human visual system. *IEEE Transactions on Image Processing*, 14(4) :439–449, Apr. 2005.
- [AT13] Alexander Andreopoulos and John K Tsotsos. 50 years of object recognition : Directions forward. *Computer Vision and Image Understanding*, 117(8) :827–891, 2013.

- [Bad92] A. J. Baddeley. An error metric for binary images. *Robust Computer Vision : Quality of Vision Algorithms*, pages 59–78, 1992.
- [Bay76] Bryce E Bayer. Color imaging array. In *U.S. patent 3,971,065, TO Eastman Kodak company, Patent and Trademark Office, Washington D.C., (Rochester, NY)*, July 1976.
- [BdKM⁺04] Elisabeth Brusseau, Chris L de Korte, Frits Mastik, Johannes Schaar, and Anton FW van der Steen. Fully automatic luminal contour segmentation in intracoronary ultrasound imaging—a statistical approach. *IEEE transactions on medical imaging*, 23(5) :554–566, 2004.
- [Ber97] Hal Berghel. Watermarking cyberspace. *Communications of the ACM*, 40(11) :19–24, 1997.
- [BETVG08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3) :346–359, 2008.
- [BG09] A. G. Boaventura and A. Gonzaga. Method to evaluate the performance of edge detector. 2009.
- [BKD01] K. Bowyer, C. Kranenburg, and S. Dougherty. Edge detector evaluation using empirical roc curves. In *CVIU*, pages 77–103, 2001.
- [BLM99] Jack T Brassil, Steven Low, and Nicholas F Maxemchuk. Copyright protection for the electronic distribution of text documents. *Proceedings of the IEEE*, 87(7) :1181–1196, 1999.
- [BLMO95] Jack Brassil, Steven Low, NF Maxemchuk, and Larry ÓGorman. Hiding information in document images. In *Proceedings of the Conference on Information Sciences and Systems (CISS)*, pages 482–489. Johns Hopkins University, Baltimore, MD, March 22-24, 1995.
- [BM98] Serge Belongie and Jitendra Malik. Finding boundaries in natural images : A new method using point descriptors and area completion. *Computer Vision—ECCV’98*, pages 751–766, 1998.
- [BS12] Eli Biham and Adi Shamir. *Differential cryptanalysis of the data encryption standard*. Springer-Verlag New York Inc. Edition : Softcover reprint of the original 1st ed. 1993, Springer Science & Business Media, ISBN-13 : 978-1461393160, 2012.
- [BWBM06] T. Brox, J. Weickert, B. Burgeth, and P. Mrázek. Nonlinear structure tensors. *IVC*, 24(1) :41–55, Elsevier, 2006.

- [Cac98] Christian Cachin. An information-theoretic model for steganography. In *Information Hiding, 2nd International Workshop, D. Aucsmith, editor, volume 1525 of Lecture Notes in Computer Science*, pages 306–318. Springer-Verlag, New York, 1998.
- [Can86a] J. Canny. A computational approach to edge detection. *IEEE TPAMI*, (6) :679–698, 1986.
- [Can86b] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6) :679–698, 1986.
- [CC04] Chi-Kwong Chan and Lee-Ming Cheng. Hiding data in images by simple lsb substitution. *Pattern recognition*, 37(3) :469–474, 2004.
- [CCCMK10] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography : Survey and analysis of current methods. *Signal processing*, 90(3) :727–752, 2010.
- [CFB10] Roberto Caldelli, Francesco Filippini, and Rudy Becarelli. Reversible watermarking techniques : an overview and a classification. *EURASIP Journal on Information Security*, 2010 :2, June 2010.
- [CLRE08] S. Chabrier, H. Laurent, C. Rosenberger, and B. Emile. Comparative study of contour detection evaluation criteria based on dissimilarity measures. *EURASIP J. on Image and Video Processing*, 2008 :2, 2008.
- [CMB⁺07] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. 2nd Edition, The Morgan Kaufmann Series in Multimedia Information and Systems, Printed in the United States of America, November 27, 2007.
- [Cra98] Scott Craver. On public-key steganography in the presence of an active warden. In *International Workshop on Information Hiding*, pages 355–368. Springer, 1998.
- [DF15] Tomáš Denemark and Jessica Fridrich. Improving steganographic security by synchronizing the selection channel. In *Proc. of the 3rd ACM Workshop on Inf. Hiding and Multimedia Security (IH&MMSec), Portland, Oregon*, pages 5–14, June 2015.
- [DG93] R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *IJCV*, 10(2) :101–124, 1993.
- [Die00] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

- [DJ94] M-P Dubuisson and Anil K Jain. A modified hausdorff distance for object matching. In *IEEE ICPR*, volume 1, pages 566–568, 1994.
- [Dou09] Geoff Dougherty. *Digital image processing for medical applications*. Cambridge University Press, 2009.
- [DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
- [DZ86] S. Di Zenzo. A note on the gradient of a multi-image. *CVGIP*, 33(1) :116–125, 1986.
- [EGFML98] Abbas El Gamal, Boyd A Fowler, Hao Min, and Xinqiao Liu. Modeling and estimation of fpn components in cmos image sensors. *Proceedings of SPIE, Solid State Sensor Arrays : Development and Applications II*, vol. 3301–20, San Jose, CA, pages 168–177, January 1998.
- [Ett98] J Mark Ettinger. Steganalysis and game equilibria. In *Information Hiding, 2nd International Workshop, D. Aucsmith, editor, volume 1525 of Lecture Notes in Computer Science*, pages 319–328, Portland, OR, April 14–17. Springer-Verlag, New York, 1998.
- [FA91] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE TPAMI*, 13(9) :891–906, 1991.
- [FDvdM02] Steven M. de Jong Freek D. van der Meer. *Imaging Spectrometry Remote Sensing and Digital Image Processing*. Springer, 2002.
- [FFPN03] Gina Fisk, Mike Fisk, Christos Papadopoulos, and Joshua Neil. Eliminating steganography in internet traffic with active wardens. In *Information Hiding, 5th International Workshop, F. A. P. Petitcolas, editor, volume 2578 of Lecture Notes in Computer Science*, pages 18–35, Noordwijkerhout, October 7–9,. Springer-Verlag, New York, 2003.
- [FGCPMCMC08] N.L. Fernández-García, A. Carmona-Poyato, R. Medina-Carnicer, and F.J. Madrid-Cuevas. Automatic generation of consensus ground truth for the comparison of edge detection techniques. *IVC*, 26(4) :496–511, 2008.
- [FGMCCP⁺04] N.L. Fernández-García, R. Medina-Carnicer, A. Carmona-Poyato, F.J. Madrid-Cuevas, and M. Prieto-Villegas. Characterization of empirical discrepancy evaluation measures. *Patt. Rec. Lett.*, 25(1) :35–47, 2004.
- [FIG06] Guillermo A Francia III and Tyler S Gomez. Steganography obliterators : an attack on the least significant bits. In *Proceedings 3rd annual conference on Information security curriculum development (InfoSecCD*

- 06), pages 85–91, Kennesaw State University, Kennesaw, GA, September 22–23,. ACM, 2006.
- [FJF11] Tomáš Filler, Jan Judas, and Jessica Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security*, 6(3) :920–935, 2011.
- [FK12] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3) :868–882, 2012.
- [FK13] Jessica J Fridrich and Jan Kodovský. Multivariate gaussian model for designing additive distortion for steganography. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vancouver, BC*, pages 2949–2953, May 26-31, 2013.
- [FL00a] J. Fridrich and M. Long. Steganalysis of lsb encoding in color images. In *IEEE International Conference on Multimedia and Expo (ICME)*, volume 3, pages 1279–1282 vol.3, 2000.
- [FL00b] Jessica Fridrich and M Long. Steganalysis of lsb encoding in color images. In *IEEE Int. Conf. on Multimedia and Expo (ICME) 2000, New York, NY, USA*, volume Vol.3, pages 1279–1282, July 2000.
- [För86] W. Förstner. A feature based correspondence algorithm for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26(3) :150–166, 1986.
- [Fri09] Jessica Fridrich. *Steganography in digital media : principles, algorithms, and applications*. Printed in the United Kingdom, Cambridge University Press, 2009.
- [GB10] T. Gloe and R. Böhme. The dresden image database for benchmarking digital image forensics. In *ACM*, pages 1584–1590, 2010.
- [GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [GF15] Miroslav Goljan and Jessica Fridrich. CFA-aware features for steganalysis of color images. In *Proc. IS&T/SPIE Electronic Imaging, Int. Society for Optics and Photonics (SPIE), San Francisco, CA, USA*, volume 94090V, page (13), Feb. 2015.
- [GFC14] M. Goljan, J. Fridrich, and R. Cogranne. Rich model for steganalysis of color images. In *IEEE Workshop on Information Forensic and Security, GA*, 2014.

- [GGVdWG12] Theo Gevers, Arjan Gijsenij, Joost Van de Weijer, and Jan-Mark Geusebroek. *Color in computer vision : fundamentals and applications*, volume 23. Published by John Wiley&Sons, Inc., Hoboken, New Jersey, USA, ISBN : 9780470890844, 2012.
- [GMP98] V. Gouet, P. Montesinos, and D. Pelé. A fast matching method for color uncalibrated images using differential invariants. In *BMVC*, pages 1–10, 1998.
- [GPW03] C. Grigorescu, N. Petkov, and M. Westenberg. Contour detection based on nonclassical receptive field inhibition. *IEEE TIP*, 12(7) :729–739, 2003.
- [GSvdW02] J.M. Geusebroek, A. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *ECCV 2002*, pages 99–112, 2002.
- [GW02] Rafael C Gonzalez and Paul Wintz. *Digital image processing*. Second Edition, Printed by Prentice-Hall, Inc. Upper Saddle River, New Jersey, United States of America, ISBN : 0-201-18075-8, 2002.
- [Har84] Robert M Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1) :58–68, 1984.
- [HDSG15] Jason Hiney, Tejas Dakve, Krzysztof Szczypiorski, and Kris Gaj. Using facebook for image steganography. In *Availability, Reliability and Security (ARES), 2015 10th International Conference on*, pages 442–447. IEEE, 2015.
- [Her96] Herodotus. *The Histories*. Pantheon Books, London, Translated by Aubrey de Sélincourt, 1996.
- [HF12a] V. Holub and J. Fridrich. Designing steganographic distortion using directional filters. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 234–239, 2012.
- [HF12b] Vojtěch Holub and Jessica Fridrich. Designing steganographic distortion using directional filters. In *Proc. IEEE Int. Workshop on Inf. Forensics and Security (WIFS), Tenerife, Spain*, pages 234–239, Dec. 2012.
- [HFD14a] V. Holub, J. Fridrich, and T. Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP J. Information Security*, pages 1–13, 2014.
- [HFD14b] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security, Special Issue on Revised Selected*

- Papers of the 1st ACM IH&MMS Workshop*, Vol.2014(no.1) :1–13, Jan. 2014.
- [HH03] C. Hosmer and C. Hyde. Discovering covert digital evidence. In *Paper presented at the 3rd Annual Digital Forensic Research Workshop (DFRWS)*, 2003.
- [Hil93] D. Hilbert. *Theory of algebraic invariants*. Cambridge University Press, 1993.
- [HLVA02] Nicholas J Hopper, John Langford, and Luis Von Ahn. Provably secure steganography. In *Annual International Cryptology Conference*, pages 77–92. Springer, 2002.
- [Hol98] Gerald C Holst. Ccd arrays, cameras, and displays. *2nd edition*. JCD Publishing and SPIE Pres, USA, 1998.
- [HP03] J. J. Harmsen and W. A. Pearlman. Steganalysis of additive-noise modelable information hiding. In *Electronic Imaging 2003*, pages 131–142. International Society for Optics and Photonics, 2003.
- [HR93] D.P. Huttenlocher and W.J. Rucklidge. A multi-resolution technique for comparing images using the hausdorff distance. In *IEEE CVPR*, pages 705–706, 1993.
- [HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50, Citeseer, 1988.
- [HSSB97] M. D. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer. A robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE TPAMI*, 19(12) :1338–1359, 1997.
- [HW04] Chun-Hsiang Huang and Ja-Ling Wu. Attacking visible watermarking schemes. In *IEEE Transactions on Multimedia*, 6(1) :16–30, 2004.
- [HYK13] X. Hou, A. Yuille, and C. Koch. Boundary detection benchmarking : Beyond f-measures. In *IEEE CVPR*, pages 2123–2130, 2013.
- [Hyt06] Heli T Hytti. Characterization of digital image noise properties based on raw data. *Proceedings of SPIE-IS&T Electronic Imaging, Image Quality and System Performance III, Vol. 6059, San Jose, CA, January 17. Bellingham, WA*, pages 1–12, 2006.
- [Jan01] James R Janesick. Scientific charge-coupled devices. *SPIE PRESS Monograph, vol. PM83, SPIE. The International Society for Optical Engineering*, 2001.
- [JU04a] M. Jacob and M. Unser. Design of steerable filters for feature detection using canny-like criteria. *IEEE TPAMI*, 26(8) :1007–1019, 2004.

- [JU04b] M. Jacob and M. Unser. Design of steerable filters for feature detection using canny-like criteria. *IEEE TPAMI*, 26(8) :1007–1019, 2004.
- [Kah96] David Kahn. The history of steganography. In *Proceedings of First International Workshop on Information Hiding, Cambridge, UK, May 30 - June 1, 1996, Lecture notes in Computer Science, Vol.1174*, pages 1–5. Springer Berlin Heidelberg, 1996.
- [KB14] M. Kirchner and R. Bohme. Steganalysis in technicolor boosting ws detection of stego images from cfa-interpolated covers. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3982–3986, 2014.
- [KBB⁺13] Andrew D Ker, Patrick Bas, Rainer Böhme, Rémi Cogramne, Scott Craver, Tomáš Filler, Jessica Fridrich, and Tomáš Pevný. Moving steganography and steganalysis from the laboratory into the real world. In *Proccessding 1st ACM workshop on Information hiding and multimedia security (IH&MMSec), Montpellier, France*, pages 45–58, 2013.
- [KCP13] Sarra Kouider, Marc Chaumont, and William Puech. Adaptive steganography by oracle (aso). In *IEEE International Conference on Multimedia and Expo (ICME), San Jose, California, USA*, pages 1–6. IEEE, July 2013.
- [KDHM99] Ton Kalker, Geert Depovere, Jaap Haitsma, and Maurice J Maes. Video watermarking system for broadcast monitoring. In *Electronic Imaging'99*, pages 103–112. International Society for Optics and Photonics, 1999.
- [KDR06] Younhee Kim, Zoran Duric, and Dana Richards. Modified matrix encoding technique for minimal distortion steganography. In *International Workshop on Information Hiding*, pages 314–327. Springer, 2006.
- [Kel01] Jack Kelley. Terror groups hide behind web encryption. *USA Today News. An article : Available at <http://www.usatoday.com/tech/news/2001-02-05-binladen.htm>*, May 02,2001.
- [Ker04] Andrew D Ker. Improved detection of lsb steganography in grayscale images. In *International workshop on information hiding*, pages 97–115. Springer, 2004.
- [Ker05] Andrew D Ker. Steganalysis of lsb matching in grayscale images. *IEEE signal processing letters*, 12(6) :441–444, 2005.
- [KFH12a] J. Kodovsky, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*,, 7(2) :432–444, April 2012.

- [KFH12b] Jan Kodovský, Jessica Fridrich, and Vojtěch Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Trans. on Inf. Forensics and Security*, Vol.7(no.2) :432–444, Apr. 2012.
- [KJ05] Zahra Zahedi Kermani and Mansour Jamzad. A robust steganography algorithm based on texture similarity using gabor filter. In *Proceedings IEEE, The Fifth International Symposium on Signal Processing and Information Technology*, pages 578–582. IEEE, 2005.
- [KJZC14] Asma Kerbiche, Saoussen Ben Jabra, Ezzedine Zagrouba, and Vincent Charvillat. Video watermarking based on interactive detection of feature regions. In *International Conferences on Advances in Multimedia-MEDIA 2014*, pages pp–77, 2014.
- [KL14] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.
- [KOL01] Gina KOLATA. Veiled messages of terror may lurk in cyberspace. *New York Times News. An article : Available at <http://www.nytimes.com/2001/10/30/science/physical/30STEG.html>*, October 30, 2001.
- [Köt03] U. Köthe. Edge and junction detection with an improved structure tensor. In *Joint Patt. Rec. Symp.*, pages 25–32, Springer, 2003.
- [KR80] L. Kitchen and A. Rosenfeld. Gray-level corner detection. Technical report, DTIC Document, 1980.
- [Kri14] Scott Krig. *Computer vision metrics : Survey, taxonomy, and analysis*. Apress, 2014.
- [KSM06] Mehdi Kharrazi, Husrev T Sencar, and Nasir Memon. Cover selection for steganographic embedding. In *Proceedings of International Conference on Image Processing (ICIP), Atlanta, GA*, pages 117–120. IEEE, October. 2006.
- [Lap05] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3) :107–123, 2005.
- [LBMR10] Fulu Li, James Barabas, Ankit Mohan, and Ramesh Raskar. Analysis on errors due to photon noise and quantization process with multiple images. In *In 44th Annual Conference on Information Sciences and Systems (CISS), Princeton, USA*, pages 1–6. IEEE, March 17-19, 2010.
- [Leo05] Cornelius T. Leondes. Medical imaging systems technology. a 5-volume set / :Methods in cardiovascular and brain systems [electronic resource] :. 2005.

- [LFG05] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. Determining digital image origin using sensor imperfections. In *A. Said and J. G. Apostolopoulos, editors, Proceedings of SPIE : Image and Video Communications and Processing , volume 5685*, pages 249–260. International Society for Optics and Photonics, 2005.
- [LFG06] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2) :205–214, June 2006.
- [LI03] Zheng Liu and Akira Inoue. Audio watermarking techniques using sinusoidal patterns based on pseudorandom sequences. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(8) :801–812, 2003.
- [Lin98] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2) :79–116, 1998.
- [LLG05] Lian-Shan Liu, Ren-Hou Li, and Qi Gao. A robust video watermarking scheme based on dct. In *Proceedings of International Conference on Machine Learning and Cybernetics*, volume 8, pages 5176–5180. IEEE, 2005.
- [LMB⁺05] Dimitri A Lisin, Marwan A Mattar, Matthew B Blaschko, Erik G Learned-Miller, and Mark C Benfield. Combining local and global image features for object class recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, pages 47–47, 2005.
- [LMDBB13] C. Lopez-Molina, B. De Baets, and H. Bustince. Quantitative error measures for edge detection. *Patt. Rec.*, 46(4) :1125–1139, 2013.
- [LMDBB16] C. Lopez-Molina, B. De Baets, and H. Bustince. Twofold consensus for boundary detection ground truth. *Knowledge-Based Syst.*, 98 :162–171, 2016.
- [LMY10] Olivier Losson, Ludovic Macaire, and Yanqin Yang. Comparison of color demosaicing methods. *Preprint submitted to Advances in Imaging and Electron Physics*, 162 :173–265, 2010.
- [Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2) :91–110, 2004.
- [LSR⁺08] Qingzhong Liu, Andrew H Sung, Bernardete Ribeiro, Mingzhen Wei, Zhongxue Chen, and Jianyun Xu. Image complexity and feature mining for steganalysis of least significant bit matching steganography. *J. of Information Sciences*, Vol.178(1) :21–36, Jan. 2008.

- [LWHL14] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li. A new cost function for spatial image steganography. In *Proc. IEEE, Int. Conf. Image Processing (ICIP), Paris, France*, pages 4206–4210, Oct. 2014.
- [MAB⁺14] B. Magnier, A. Aberkane, P. Borianne, P. Montesinos, and C. Jourdan. Multi-scale crest line extraction based on half gaussian kernels. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5105–5109, 2014.
- [MAM17] B. Magnier, H. Abdulrahman, and P. Montesinos. Détection de coins par combinaison de filtres asyétriques orientés. In *Proceedings of the 25th edition GRETSI 2017, Juan-les-Pins, France, 5–8 September, - to appear-*, 2017.
- [MBL⁺09] Yoan Miche, Patrick Bas, Amaury Lendasse, Christian Jutten, and Olli Simula. Reliable steganalysis using a minimum set of samples and features. *EURASIP J. on Inf. Security*, vol.2009, article ID 901381 :(13), 2009.
- [MDA07] M. Mühlich, T. Dahmen, and T. Aach. Doubly-rotated matched filtering. In *EUSIPCO*, pages 2169–2173. IEEE, 2007.
- [MDCL⁺07] F. Michelet, J.-P. Da Costa, O. Laviaille, Y. Berthoumieu, P. Baylou, and C. Germain. Estimating local multiple orientations. *Signal Proc.*, 87(7) :1655–1669, 2007.
- [MFM04] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE TPAMI*, 26(5) :530–549, 2004.
- [MFTM01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE ICCV*, volume 2, pages 416–423, 2001.
- [MJYR⁺10] Robert A Maschal Jr, S Susan Young, Joe Reynolds, Keith Krapels, Jonathan Fanning, and Ted Corbin. Review of bayer pattern color filter array (cfa) demosaicing with new quality assessment algorithms. Technical report, U.S. Army Research Laboratory, Adelphi, MD 20783-1197, DTIC Document, January 2010.
- [MLZ16] B. Magnier, A. Le, and A. Zogo. A quantitative error measure for the evaluation of roof edge detectors. In *IEEE IST*, pages 429–434, 2016.
- [MM96] Bangalore S Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, 18(8) :837–842, 1996.

- [MM10] Philippe Montesinos and Baptiste Magnier. A new perceptual edge detector in color images. In *Advanced Concepts for Intelligent Vision Systems - 12th International Conference, ACIVS 2010, Sydney, Australia, December 13-16, 2010, Proceedings, Part I*, volume 6474 of *Lecture Notes in Computer Science*, pages 209–220. Springer, 2010.
- [MM14a] B. Magnier and P. Montesinos. Oriented half gaussian kernels and anisotropic diffusion. In *VISAPP*, volume 1, pages 73–81, 2014.
- [MM14b] Baptiste Magnier and Philippe Montesinos. Oriented half gaussian kernels and anisotropic diffusion. In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications VISAPP 2014, Volume 1, Lisbon, Portugal, 5-8 January, 2014*, pages 73–81. SciTePress, 2014.
- [MMD11a] B. Magnier, P. Montesinos, and D. Diep. Fast Anisotropic Edge Detection Using Gamma Correction in Color Images. In *IEEE ISPA*, pages 212–217, 2011.
- [MMD11b] Baptiste Magnier, Philippe Montesinos, and Daniel Diep. Texture removal by pixel classification using a rotating filter. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, pages 1097–1100. IEEE, 2011.
- [MMD11c] Baptiste Magnier, Philippe Montesinos, and Daniel Diep. Texture removal in color images by anisotropic diffusion. In *Proceedings of the Sixth International Conference on Computer Vision Theory and Applications VISAPP 2011, Vilamoura, Algarve, Portugal, 5-7 March, 2011*, pages 40–50. SciTePress, 2011.
- [MMD13] Baptiste Magnier, Philippe Montesinos, and Daniel Diep. A tool for brain magnetic resonance image segmentation. In *Proceedings of the International Conference on Computer Vision Theory and Applications VISAPP 2013, Volume 2, Barcelona, Spain, 21-24 February, 2013.*, pages 75–79. SciTePress, 2013.
- [Moo91] Jonathan Martin Mooney. Effect of spatial noise on the minimum resolvable temperature of a staring sensor. *Applied optics*, 30(23) :3324–3332, 1991.
- [MP10] Mihai Mitrea and Françoise Prêteux. From watermarking to in-band enrichment : Theoretical and applicative trends. *Advanced Techniques in Multimedia Watermarking : Image, Video and Audio Applications : Image, Video and Audio Applications*, page 111, 2010.

- [MPB⁺12] R. Moreno, L. Pizarro, B. Burgeth, J. Weickert, M. A. Garcia, and D. Puig. Adaptation of tensor voting to image structure estimation. In *New Developments in the Visualization and Processing of Tensor Fields*, pages 29–50. Springer, 2012.
- [MSC⁺13] Anton Mosunov, Vineet Sinha, Heather Crawford, John Aycock, Daniel Medeiros Nunes de Castro, and Rashmi Kumari. Assured supraliminal steganography in computer games. In *In Proceeding Information Security Applications, 14th International Workshop, (WISA), Jeju Island, Korea, Vol. 8267*, pages 245–259. Berlin, Heidelberg : Springer, August 19-21. 2013.
- [MXM13] Baptiste Magnier, Huanyu Xu, and Philippe Montesinos. Half gaussian kernels based shock filter for image deblurring and regularization. In *VISAPP 2013 - Proceedings of the International Conference on Computer Vision Theory and Applications, Volume 1, Barcelona, Spain, 21-24 February, 2013.*, pages 51–60. SciTePress, 2013.
- [NA12] Mark S Nixon and Alberto S Aguado. *Feature extraction & image processing for computer vision*. Academic Press, 2012.
- [NMCHGU⁺15] M Nakano-Miyatake, M Cedillo-Hernandez, Francisco Garcia-Ugalde, A Cedillo-Hernandez, and Hector Perez-Meana. Copyright protection of color imaging using robust-encoded watermarking. *Radioengineering*, 2015.
- [OBBC02] C. Odet, B. Belaroussi, and H. Benoit-Cattin. Scalable discrepancy measures for segmentation evaluation. In *IEEE ICIP*, volume 1, pages 785–788, 2002.
- [OH10] Mohammadreza Asghari Oskoei and Huosheng Hu. A survey on edge detection methods. *University of Essex, UK*, 2010.
- [Ots75] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296) :23–27, 1975.
- [PA95] A. J. Pinho and L. B. Almeida. Edge detection filters based on artificial neural networks. In *ICIAP*, pages 159–164. Springer, 1995.
- [PBF10] Tomáš Pevný, Patrick Bas, and Jessica Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. on Inf. Forensics and Security (TIFS)*, Vol.5(no.2) :215–224, June 2010.
- [PBR13] W Puech, AG Bors, and JM Rodrigues. Protection of colour images by selective encryption. In *Advanced Color Image Processing and Analysis*, pages 397–421. Springer, 2013.

- [Per92] P. Perona. Steerable-scalable kernels for edge detection and junction analysis. *IVC*, 10(10) :663–672, 1992.
- [PFB10] Tomáš Pevný, Tomáš Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In *In R. Bohme and R. Safavi-Naini, editors, Information Hiding, 12th International Conference, volume 6387 of Lecture Notes in Computer Science, Calgary, Canada*, pages 161–177. Springer-Verlag, New York, June 28-30, 2010.
- [PGAN16] K. Panetta, C. Gao, S. Agaian, and S. Nercessian. A new reference-based edge map quality measure. *IEEE Trans. on Systems Man and Cybernetics : Systems*, 46(11) :1505–1517, 2016.
- [PH03] Niels Provos and Peter Honeyman. Hide and seek : An introduction to steganography. *IEEE security & privacy*, 99(3) :32–44, 2003.
- [PKSM05] Kyle Petrowski, Mehdi Kharrazi, Hüsrev T Sencar, and Nasir D Memon. Psteg : steganographic embedding through patching. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing ICASSP(2), Philadelphia, PA*, pages 537–540. Citeseer, March 18-23, 2005.
- [PM82] T. Peli and D. Malah. A study of edge detection algorithms. *CGIP*, 20(1) :1–21, 1982.
- [PM90] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *ICCV*, pages 52–57. IEEE, 1990.
- [Pra13] William K Pratt. *Introduction to digital image processing*. CRC Press, 2013.
- [Pre70] Judith MS Prewitt. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1) :15–19, 1970.
- [PTC⁺12] William Puech, Alain Trémeau, Philippe Carré, Christine Fernandez-Maloigne, Frédérique Robert-Inacio, and Ludovic Macaire. Protection of color images. *Digital Color*, pages 227–264, 2012.
- [PTP16] Pauline Puteaux, Dave Trinel, and William Puech. High-capacity data hiding in encrypted images using msb prediction. In *Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on*, pages 1–6. IEEE, 2016.
- [RFM06] Xiaofeng Ren, Charless C Fowlkes, and Jitendra Malik. Figure/ground assignment in natural images. In *European Conference on Computer Vision*, pages 614–627. Springer, 2006.

- [RO13] Mark Rhodes-Ousley. *Information security the complete reference*. McGraw Hill Professional, 2013.
- [Ros01] P. L. Rosin. Unimodal thresholding. *Patt. Rec.*, 34(11) :2083–2096, 2001.
- [RPD10] E. Rosten, R. Porter, and T. Drummond. Faster and better : A machine learning approach to corner detection. *IEEE TPAMI*, 32(1) :105–119, 2010.
- [RSA78] Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2) :120–126, February 1978.
- [RT71] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Trans. on Computers*, 100(5) :562–569, 1971.
- [SB97] S. M. Smith and J. M. Brady. Susan : a new approach to low level image processing. *IJCV*, 23(1) :45–78, Springer, 1997.
- [SB11] Ivan Sipiran and Benjamin Bustos. Harris 3d : a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11) :963–976, 2011.
- [SC92] J. Shen and S. Castan. An optimal linear operator for step edge detection. *CVGIP*, 54(2) :112–133, 1992.
- [SCF16] Vahid Sedighi, Rémi Cogranne, and Jessica Fridrich. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 11(2) :221–234, 2016.
- [SCP13] Zafar Shahid, Marc Chaumont, and William Puech. H. 264/avc video watermarking for active fingerprinting based on tardos code. *Signal, Image and Video Processing*, 7(4) :679–694, 2013.
- [set11] Berkeley segmentation data set and benchmarks 500. URL <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html>, 2011.
- [SF96] E. P. Simoncelli and H. Farid. Steerable wedge filters for local orientation analysis. *IEEE TIP*, 5(9) :1377–1382, Citeseer, 1996.
- [SG91] C. Strasters and J. Gerbrands. Three-dimensional image segmentation using a split, merge and group approach. *Patt. Rec. Let.*, 12(5) :307–325, 1991.

- [Sim83] Gustavus J Simmons. The prisoners : problem and the subliminal channel. In *In D. Chaum , editor, Advances in Cryptology, CRYPTO*, pages 51–67. New York : Plenum Press, Santa Barbara, CA, August 22-24, 1983.
- [SJ09] Hedieh Sajedi and Mansour Jamzad. Secure steganography based on embedding capacity. *International Journal of Information Security*, 8(6) :433–445, 2009.
- [SKZ09] Vasiliy Sachnev, Hyoung Joong Kim, and Rongyue Zhang. Less detectable jpeg steganography method based on heuristic optimization and bch syndrome coding. In *Proceedings of the 11th ACM workshop on Multimedia and security*, pages 131–140. ACM, 2009.
- [SL10] Yifeng Sun and Fenlin Liu. Selecting cover for image steganography by correlation coefficient. In *Proceedings of Second International Workshop on Education Technology and Computer Science (ETCS)*, volume 2, pages 159–162. IEEE, 2010.
- [SLY⁺15] Xiaofeng Song, Fenlin Liu, Chunfang Yang, Xiangyang Luo, and Yi Zhang. Steganalysis of adaptive jpeg steganography using 2d gabor filters. In *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pages 15–23. ACM, 2015.
- [SM05] Ronald W Schafer and Russel M Mersereau. Demosaicking : color filter array interpolation. *IEEE Signal Processing Magazine*, 22(1), 2005.
- [Sma03] Nigel Paul Smart. *Cryptography : an introduction*, volume 5. Printed in the McGraw-Hill New York, ISBN 0-077-09987-7, 2003.
- [SMB00] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2) :151–172, 2000.
- [SNMM15] Seyed Mohammad Seyedzadeh, Benyamin Norouzi, Mohammad Reza Mosavi, and Sattar Mirzakuchaki. A novel color image encryption algorithm based on spatial permutation and quantum chaotic map. *Nonlinear Dynamics*, 81(1-2) :511–529, 2015.
- [Sob70] I.E. Sobel. Camera Models and Machine Perception. *PhD Thesis, Stanford University*, 1970.
- [ST94] J. Shi and C. Tomasi. Good features to track. In *IEEE CVPR*, pages 593–600, 1994.
- [SZ13] P.-L. Shui and W.-C. Zhang. Corner detection and classification using anisotropic directional derivative representations. *IEEE TIP*, 22(8) :3204–3218, 2013.

- [Sze10] Richard Szeliski. *Computer vision : algorithms and applications*. Springer Science & Business Media, 2010.
- [Tac90] A Tacticus. *How to Survive Under Siege/Aineias the Tactician*. Oxford, UK : Clarendon Ancient History Series, 1990.
- [TAV12] P. Thiyagarajan, G. Aghila, and V. P. Venkatesan. Steganalysis using color model conversion. *arXiv preprint arXiv :1206.2914*, 2012.
- [TC05] Shalin Trivedi and Rajarathnam Chandramouli. Secret key estimation in sequential steganography. *IEEE Transactions on Signal Processing*, 53(2) :746–757, 2005.
- [TG03] Peter Thorsteinson and Arun Ganesh. *.NET Security and Cryptography*. 1st ed. Prentice Hall Professional Technical Reference, ISBN-13 : 978-0-13-100851-9, 2003.
- [Tia02] Jun Tian. Reversible watermarking by difference expansion. In *Proceedings of workshop on multimedia and security*, pages 19–22, December 2002.
- [Tib02] Patrick S Tibbetts. Terrorist use of the internet and related information technologies. Technical report, DTIC Document, 2002.
- [TLLH16] Weixuan Tang, Bin Li, Weiqi Luo, and Jiwu Huang. Clustering steganographic modification directions for color components. *Signal Processing Letters, IEEE*, Vol.23(No.2) :197–201, Feb. 2016.
- [VAH04] Luis Von Ahn and Nicholas J Hopper. Public-key steganography. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 323–341. Springer, 2004.
- [VMDM15a] Darshan Venkatrayappa, Philippe Montesinos, Daniel Diep, and Baptiste Magnier. A novel image descriptor based on anisotropic filtering. In *Computer Analysis of Images and Patterns - 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015 Proceedings, Part I*, volume 9256 of *Lecture Notes in Computer Science*, pages 161–173. Springer, 2015.
- [VMDM15b] Darshan Venkatrayappa, Philippe Montesinos, Daniel Diep, and Baptiste Magnier. Rsd-hog : a new image descriptor. In *Scandinavian Conference on Image Analysis. Lecture Notes in Computer Science*, pages 400–409. Springer, 2015.
- [VR95] S. Venkatesh and P. L. Rosin. Dynamic threshold determination by local and global edge evaluation. *CVGIP*, 57(2) :146–160, 1995.

- [WBSS04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment : from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4) :600–612, 2004.
- [WFHP16] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining : Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
- [WL98] Min Wu and Bede Liu. Watermarking for image authentication. In *IEEE International Conference on Image Processing, volume (2)*, pages 437–441, 1998.
- [WM08] Ying Wang and Pierre Moulin. Perfectly secure steganography : Capacity, error exponents, and code constructions. *IEEE Transactions on Information Theory, Special Issue on Security*, 54(6) :2706–2722, 2008.
- [WP00] Andreas Westfeld and Andreas Pfitzmann. Attacks on steganographic systems. In *Proc. on Information Hiding, In : Pfitzmann A. (eds.) : 3rd Int. Workshop Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg*, volume 1768, pages 61–76, 2000.
- [WPD99] Raymond B Wolfgang, Christine I Podilchuk, and Edward J Delp. Perceptual watermarks for digital images and video. *Proceedings of the IEEE*, 87(7) :1108–1126, 1999.
- [WZH11] J. Wang, C. Zhang, and P. Hao. New color filter arrays of high light sensitivity and high demosaicking performance. In *IEEE International Conference on Image Processing (ICIP)*, pages 3153–3156, 2011.
- [XDG14] G.S. Xia, J. Delon, and Y. Gousseau. Accurate junction detection and characterization in natural images. *IJCV*, 106(1) :31–56, Springer, 2014.
- [YGB78] W.A. Yasnoff, W. Galbraith, and J.W. Bacus. Error measures for objective assessment of scene segmentation algorithms. *Analytical and Quantitative Cytology*, 1(2) :107–121, 1978.
- [YP03] Y. Yitzhaky and E. Peli. A method for objective edge detection evaluation and detector parameter selection. *IEEE TPAMI*, 25(8) :1027–1033, 2003.
- [YS11] L. Yuan and J. Sun. High quality image reconstruction from raw and jpeg image pair. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2158–2165, 2011.
- [ZT98a] D. Ziou and S. Tabbone. Edge detection techniques : an overview. *International Journal on Pattern Recognition and Image Analysis*, 8(4) :537–559, 1998.

- [ZT⁺98b] Djemel Ziou, Salvatore Tabbone, et al. Edge detection techniques-an overview. *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii*, 8 :537–559, 1998.