Multiscale Line Feature and Keypoint Detection Based on Anisotropic Semi-Gaussian Filtering

Ghulam Sakhi Shokouh

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR
DE L'INSTITUT MINES-TÉLÉCOM (IMT) -
ÉCOLE NATIONALE SUPÉRIEURE DES MINES D'ALÈS (IMT MINES ALÈS)

En Informatique

École doctorale : I2S - Information, Structures, Systèmes
Portée par l'Université de Montpellier

Unité de recherche : EuroMov Digital Health in Motion

Multiscale Line Feature and Keypoint Detection Based on Anisotropic Semi-Gaussian Filtering

Présentée par Ghulam Sakhi SHOKOUH

Le 06 December 2022

Sous la direction de Philippe MONTESINOS
et l'encadrement de Baptiste MAGNIER

Devant le jury composé de :

William PUECH, Professeur, LIRMM, Université de Montpellier, CNRS	Président
Olivier MONGA, Directeur de recherche, Centre IRD - UPMC	Rapporteur
Désiré SIDIBÉ, Professeur, IUT, Université de Evry - Dpt QLIO]	Rapporteur
Ludovic MACAIRE, Professeur, CRIStAL, Université de Lille, CNU	Examinateur
Philippe FILLATREAU, MCF, LGP - Ecole Nationale d'Ingénieurs de Tarbes (ENIT)	Examinateur
Philippe MONTESINOS, HDR, MCF, EuroMov DHM, IMT Mines Alès	Directeur
Baptiste MAGNIER, MCF, EuroMov DHM, IMT Mines Alès	Encadrant

IMT Mines Alès
École Mines-Télécom

Abstract

Image processing and computer vision encompass a broad domain of applications, such as image segmentation for medical image analysis, aerial image segmentation for autonomous navigation, object recognition, biometric recognition, security surveillance, etc. The extraction of features as helpful information for the interpretation of digital image data is an essential task and interesting research subject in image processing and computer vision domain. In this report, image filtering for image low-level feature detection and extraction (early vision) is investigated, which mainly involves: an objective evaluation of ridge/valley detection techniques, proposing two multiscale line feature detection techniques, and proposing a new anisotropic corner detection approach. First, to have an intuitive context and introduction, image filtering basics, basic contour detection techniques using first order and second order operators, and evaluation metrics and tools are overviewed. Meanwhile, the scale space theory and axioms were also investigated in order to find the basic criteria and constraints in the development of multiscale feature detection methods. Secondly, objective evaluation of the state-of-the-art ridge/-valley detection and extraction techniques were performed. The objective analysis of a ridge characterized as a thin and complex image structure is essentially important, for choosing, which parameter values correspond to the suitable configuration to obtain accurate results and optimal performance. The optimal parameter configuration of each filtering technique aimed for the image salient feature analysis tool has been objectively investigated, where each chosen filter's parameters correspond to the width of the desired ridge or valley. The comparative evaluations and analysis results are reported on both synthetic images, distorted with various types of noises, and thereafter real images. Thirdly, to deal with the multiscale structure of line features, a new line feature detection and extraction method was proposed, which is based on the second order semi-Gaussian filter. The experiments were led on both synthetic and real images, and the obtained result demonstrated more optimality to the state-of-the-art line feature detection based on filtering approaches. Next, another multiscale line feature detection and extraction technique composed of bi-Gaussian and Semi-Gaussian Derivative Kernel was proposed. The proposed filter is able to precisely extract the complex, narrow, and adjoin linear structure and is adapted for multiscale capability. The proposed filter is validated with experiments on different images containing complex adjoin linear structures with different scales. The extracted linear structure on both synthetic and real images has shown to be more efficient than classic linear structure extraction techniques. Regarding the corner detection techniques, we performed first an objective repeatability evaluation of 12 state-of-the-art corner detection based on a filtering approach. There exist different techniques for keypoint detection; as filtering is concerned, we have focussed on direct computation on the gray-level analysis of interest point detection because of its straightforward implementation. Our evaluation as an application to feature matching has been executed in the context of underwater video sequences. Finally, in this work, a new anisotropic corner detection method based on the formulation of causal filtering is proposed. The proposed corner detector arguably performs better in the case of localization precision. The experiments were executed on synthetic, and real images for both pixel-level and subpixel-level precision.

Keywords: Multiscale, line feature, keypoint detection, anisotropic Semi-Gaussian, objective evaluation.

Résumé

Le traitement d'images et la vision par ordinateur englobent un large domaine d'application, tel que la segmentation d'images pour l'analyse d'images médicales, la reconnaissance d'objets, la vision robotique, la surveillance, etc. L'extraction de caractéristiques en tant qu'informations utiles pour l'interprétation des données d'images numériques est une tâche essentielle et un sujet de recherche attrayant dans le domaine du traitement d'images. Dans ce rapport, nous nous sommes intéressés au filtrage d'image pour la détection et l'extraction de primitives de bas niveau et plus particulièrement à une évaluation objective des techniques de détection de crêtes/vallées. Par suite, nous proposons deux nouvelles techniques de détection de primitives linéaires multi-échelles. Enfin une nouvelle approche est présentée pour la détection de coins par des méthodes de filtrage anisotrope. Tout d'abord, dans un contexte introductif, nous passons en revue les bases du filtrage d'image, les techniques originelles de détection de contour utilisant des opérateurs de premier et de second ordre, ainsi que les métriques d'évaluation. Parallèlement, la théorie des espaces échelles et les axiomes architecturaux sont également étudiés afin de mettre en évidence les critères ainsi que les contraintes de base dans le développement de méthodes de détection de primitives multi-échelles. L'analyse objective d'une crête caractérisée comme une structure mince et complexe est importante afin de permettre de choisir quelles valeurs de paramètres correspondent à la configuration appropriée pour obtenir des résultats précis et des performances optimales. Ainsi, la configuration optimale des paramètres de chaque technique de filtrage destinée à l'outil d'analyse des caractéristiques saillantes de l'image est objectivement étudiée où les paramètres de chaque filtre choisi correspondent à la largeur de la crête ou de la vallée souhaitée. Les évaluations comparatives et les résultats d'analyse sont étudiés à la fois sur des images de synthèse, entachées par différents types de bruits, et sur des images réelles. Afin de traiter l'aspect multi-échelle des primitives linéaires, nous proposons une nouvelle technique de détection et d'extraction basée sur un filtre semi-Gaussien du second ordre. Les résultats obtenus ont démontré une plus grande robustesse par rapport à des approches de filtrage plus classiques. Ensuite, nous proposons une nouvelle technique de détection et d'extraction de caractéristiques linéaires multi-échelles composée d'un noyau dérivé bi-Gaussien et semi-Gaussien. Ce filtre est une amélioration du filtre précédent, il est capable d'extraire avec précision des structures linéaires complexes adjacentes et est adapté au cas multi-échelle. Le filtre proposé est validé par des expériences sur différentes images contenant des structures linéaires adjacentes complexes à différentes échelles. En ce qui concerne les techniques de détection de coins, nous avons tout d'abord effectué une évaluation objective de la répétabilité de douze détecteurs de coins basés sur une approche de filtrage. Il existe différentes techniques de détection de points ; en ce qui concerne le filtrage, nous nous sommes concentrés sur les détecteurs de points d'intérêt dans des images en niveaux de gris. Notre évaluation a été réalisée dans le contexte de séquences vidéo sous-marines. Enfin, faisant suite à ce travail, une nouvelle méthode anisotrope de détection de coins basée à la fois sur des combinaisons de filtres anisotropes causaux et non causaux est proposée. Notre détecteur de coins donne des points dont la précision de localisation est meilleure qu'avec les approches existantes. Les expériences ont été réalisées sur des images synthétiques et réelles pour une précision pixel et sub-pixel. **Mots-clés :** Multi-échelle, caractéristique linéaire, détection de points clés, anisotrope Demi-Gaussien, évaluation objective

Acknowledgements

Undertaking my Ph.D. study and research has been a significant experience for me, and it would not have been possible to accomplish without the support and guidance that I received from many people.

First and foremost, I would like to express my sincere gratitude to my both thesis supervisors, Assoc. Prof. Philippe Montesinos and Assoc. Prof. Baptiste Magnier for their continuous support, invaluable advice, insightful comments and patience during my Ph.D. study.

I gratefully acknowledge and appreciate the government of France for funding this work, without which this work could not have begun.

Thanks also to Assoc.Prof. Binbin Xu for his useful remarks and help during writing papers and articles in the scope of this research.

I thank the reporters of this manuscript, Prof. Olivier Monga and Prof. Desire Sidibe for their reports and valuable remarks.

I would also like to thank all the committee members of my dissertation, particularly the president of the jury, Prof. William Puech.

I appreciate the supports of Prof. Jacky Montmain and Prof. Gerard Dray during this work.

Many thanks to Mrs Edith Teychene for her continuous assistance and kindness for the administrative processes.

I would also like to thank all my colleagues and friends with whom I have shared moments of conviviality during this period.

In particular and above all, I would like to thank my wife Mrs Fatema Shokouh for her immense love and constant support throughout my Ph.D. and all times.

Contents

List of Figures

List of Tables

Glossary

DOG Difference of Gaussian.

LOG Laplacian of Gaussian.

DOLP Differene of Low-Pass.

SOAGK Second Order Anisotropic Gaussian Kernel.

HK Half Kernel.

2DSBG 2D SEMI BI-GAUSSIAN.

SDSG Second Derivative Semi-Gaussian.

NMS Non Maximum Suppression.

$\mathscr{H}$ Hessian Matrix.

PM Performance Measure.

MCC Matthews correlation coefficient.

CM Confusion Matrix.

MCM Mean Curvature Motion.

PDE Partial Differential Equeation.

PDF Probability Distribution Function.

$\mathscr{N}$ New Figure of Merit.

HF Heaviside Function.

TP True Positive.

FP False Positive.

TN True Negative.

FN False Negative.

SLAM Simultaneous Localization and Mapping.

SNR Signal-to-Noise Ratio.

G_t Ground truth.

D_c Detected contour.

Chapter 1

Introduction

Contents

1.1 Context

Image processing and computer vision encompass a broad domain of applications, such as image segmentation for medical image analysis, aerial image segmentation for autonomous navigation, object recognition, biometric recognition, security surveillance, etc. In computer vision's state-of-the-art research subjects, such as object recognition, image retrieval, image matching and classification, and stereo vision, the feature is usually the keyword discussed and investigated. Features are the significant properties of an image, or object that are used as descriptors and part of the input to processes that lead to segmentation, classification, and finally recognition of objects.

Thus, features are fundamentally important to understand, characterize, and afterward detect and describe their properties. Subsequently, the detected and described features (namely descriptor) are used as a reduced image data input to the higher level applications enabling them for faster and more accurate decision-making (e.g., pattern recognition, object classification, etc.). Meanwhile, by knowing the image features, we can enhance or highlight the relevant details, and differentiable properties (distinctive data), and correspondingly remove the irrelevant information (noises, artifacts) for either objective or subjective purposes in the image processing and computer vision pipelines. The robustness (i.e., accuracy, localization, distinctiveness, and repeatability) in all the handcrafted, machine learning and deep learning techniques rely on the richness of image features information.

Classic machine learning techniques use handcrafted features for the purpose of computer vision tasks such as image classification, image retrieval, object recognition, 3D reconstruction, etc. These techniques are developed both on image/object local and global features, which involve image low-level features such as contour, edge, ridge/valley, corner, blobs, etc.

Deep learning methods learn the features automatically from the hierarchical representation of image data, and generally scale with more data and model optimization. These methods generally learn the image/object's global features, which consist of texture, color information, shape, global contour representation, etc. Image filtering approaches play a key role in enhancing, detecting, extracting, and describing image features. There exist many handcrafted approaches for image filtering, such as image active contour models, and contour-based/geometrical and statistical-based image filtering. There are many challenges yet to be resolved and addressed in image low-level vision, and our focus image filtering.

I categorize these challenges into three groups, which are addressed in our work:

1. In the literature, there are many images filtering methods proposed and developed for line feature, and each one claims/presents in some way the performance of the algorithm. However, the performances are not usually objectively evaluated and compared, for letting the users be able to choose the better technique for their intended applications. Therefore, there is always a need for both objective and subjective evaluation (mainly objective) of state-of-the-art image filtering techniques. In this research a substantial work is devoted to this objective evaluation of image filtering techniques for early vision or image low-level features (edge, ridge/valley, line, corner).

   First, an objective evaluation of the ridge characterized as a thin and complex image structure was investigated and analyzed. The result of this analysis lets us choose, which parameter's values correspond to the suitable configuration to obtain accurate results and optimal performance. An extensive analysis followed by a supervised and objective comparison of different filtering-based ridge detection techniques is detailed in our work. The optimal parameter configuration of each filtering technique aimed for the image salient feature analysis tool has been objectively investigated, where each chosen filter's parameters correspond to the width of the desired ridge or valley.

   Secondly, corner detectors based on filtering techniques are studied in this work. Corners, as a stable feature possessing the defined characteristics of a robust point of interest, remain an active research field for machine vision researchers due to their applications in motion capture, image matching, tracking, image registration, 3D reconstruction, and object recognition. There exist different techniques for keypoint detection; as filtering is concerned, we have focused on direct computation on the gray-level analysis of interest point detection because of its straightforward implementation. Thus, we performed, an objective comparison of 12 state-of-the-art keypoint detection techniques; an application to feature matching has been executed in the context of underwater video sequences. These videos contain various types of noises caused by the environment and all geometric and/or photometric transformations

2. The second challenge consists of the performance of image low-level feature detection algorithms in the condition of many unavoidable image geometric and photometric transformations (scale, view-point, rotation, translation, affine, illumination, etc.). The performance of the image low-level feature detectors degrades depending on the proportion of mentioned transformation. Change of scale is the major challenge widely open for research and development in image low-level feature analysis. We know that objects are composed of different structures at various

scales. Indeed, depending on the scale of observation, that real-world objects appear at a different scale, and creating the scales-space representation of the image is required for computer vision applications (to analyze unknown scenes) such as object recognition, automated cartography, 3D reconstruction, etc. In scale-space theory, the image structures are processed at different scales by representing the image as a one-parameter family of the smoothed version of images. The scale-space representation parametrized by the size of the smoothing kernel is used to suppress the image's fine-scale structures.

In the image processing and computer vision literature, many approaches have been introduced for image multiscale representation, including pyramid, wavelet, quad-tree, and multi-grid methods. Part of this thesis work is dedicated to developing two new multiscale line feature detectors. First, a multiscale filtering-based line detection approach using a second order semi-Gaussian anisotropic kernel is proposed. The applied algorithm calculates the strength of the observed line feature across the different scales.The experimental results and comparison of images containing different line feature widths with state-of-the-art techniques supported the effectiveness of our technique.

Second, we have proposed a new filter that can extract multiscale complex linear and adjoint structures. The proposed filter is composed of a bi-Gaussian and Semi-Gaussian Derivative filter which is able to precisely extract the complex, narrow, and adjoin linear structure and is adapted for multiscale capability. We have performed experiments on different images containing complex adjoin linear structures with different scales. The extracted linear structure on both synthetic and real images has shown to be more efficient than classic linear structure extraction techniques.

3. The third challenge involves the localization and precision of the image low-level feature detectors, such as keypoint/point of interest and/or corner detection methods. There are many state-of-the-art keypoint/corner detectors based on handcrafted/classic machine learning techniques, which generally lacks to present high precision and localization. We have developed a new anisotropic corner detector, arguably with better precision. Here, we are introducing an anisotropic curvature measure that is able to provide precise corner point detection by simply computing a curvature-like local maximum. First, we recall the well-known Kitchen-Rosenfeld corner operator, and its link to the Mean Curvature Motion (MCM), and Partial Differential Equation (PDE) scheme, then we present the new "anisotropic curvature" operator which is an extension of the Kitchen-Roselfeld operator in an anisotropic way.

1.1.1 Handcrafted versus deep learning approaches

In the domain of image processing and computer vision, it is well-known that since the last decade, deep learning techniques have highly outperformed classic hand-crafted techniques in most of the computer vision applications. Deep learning techniques have especially been the benchmark for high-level computer vision such as image classification, object detection, face recognition, pose estimation, etc. Considering the general low-level image processing such as image segmentation and contour detection, deep learning may arguably win if there are sufficient large datasets of good quality (precisely annotated) but still depends on the model of image feature - high-level features vs low-level features(edge, ridge, valley, blob, corner, etc.) and also the corresponding applications.

Early vision for certain tasks such as edge detection, line detection, ridge/valley detection, crest-line detection, and corner and junction detection when accurate precise detection and localization are demanded (e.g., medical image analysis where the enhanced blood vessel detection is a critical, accurately localized line and corner detection for image calibration and 3D reconstruction, etc.), the classic hand-crafted technique is still widely used (Junfeng et al. [2022]). The handcrafted technique is more suited/needed for some specific enhancement and optimization tasks, such as, multiscale analysis and enhancement for precise detection of complex and noisy contour structures (e.g., narrow and bent ridges, narrow and adjacent ridges/valleys). Furthermore, these techniques are faster and easy to implement. This certain enhancement for low-level complicated structures is not possible or at least recommended with deep learning techniques due to the following deep learning limitations.

First, the deep learning technique's performance is highly dependent on both the quantity and quality of the dataset. Preparing and annotating the dataset for low-level structures such as ridges/valleys, corners, and junctions with multiple scales and in the condition of noise is very complicated and very erroneous. That is why we can not find datasets of these kinds with ground truths available on the internet. Secondly, the whole task of human labor for dataset preparation and annotation, and then developing and optimizing a complicated deep learning architecture and training the machine with lots of manual tuning of parameters and hyperparameters to get a lower or comparative result with classic technique may not be useful or worthy enough. Even though there are some deep neural network techniques used for this low-level image processing, they are usually implemented with a combination of handcrafted techniques either in pre-processing stage or post-processing stage for enhancing the final results.

1.2 Main contributions

The main contributions of this work include objective evaluation of filtering-based edge, ridge/valley, and keypoint (corner) detection techniques, scales-space analysis of contour, proposing two new multiscale line feature detection techniques, and also a new anisotropic corner detection method. The key contributions are outlined in the following list:

- Extensive and objective evaluation of the state-of-the-art ridge/valley detection techniques
- Proposing multiscale line feature detection technique using second order semi-Gaussian filter
- Proposing multiscale line feature detection technique using the combination of bi-Gaussian and semi-Gaussian filter
- Extensive evaluation of repeatability rate of filtering-based corner detection techniques in tracking underwater image sequences.
- proposing new anisotropic corner detection technique using asymmetric causal filtering.

1.3 Publications

IMTA 2020/ICPR Workshop
Shokouh, G., Magnier, B., Xu, B., and Montesinos, P. (2021a). **An objective.comparison of ridge/valley detectors by image filtering.** In ICPR 2021 Workshop, volume 12665, pages 182-197. Springer, Cham.

PRIA 2021
Shokouh, G.-S., Magnier, B., Xu, B., and Montesinos, P. (2021b). **Ridge detection by image filtering techniques: a review and an objective analysis.** Pattern Recognition and Image Analysis, 31(2).

CAIP 2021
Magnier, B., Shokouh, G.-S., Xu, B., and Montesinos, P. (2021b). **A multi-scale line feature detection using second order semi-gaussian filters.** In CAIP, volume 13053 of LNCS, pages 98-108. Springer International Publishing.

CVAUI 2022/ICPR Workshop
Shokouh, G.-S., Magnier, B., Montesinos, P.,Xu, B. 5th Workshop on Computer Vision for Analysis of Underwater Imagery (CVAUI). **Repeatability evaluation of keypoint detection techniques in tracking underwater image sequences.** ICPR., August 2022, Springer, Montreal, Quebec, Canada.

EUVIP 2022
Pruvost, E., Tulet, H., Delort, E., Shokouh, G.-S., Montesinos, P., Magnier, B., Jourdan, C., Belaud, E., Hedde, M. **Invertebrates detection with YOLOv5: Towards study of soil organisms using deep learning.** 10th European Workshop on Visual Information Processing (EUVIP). IEEE. September 2022,Lisbon, Portugal.

1.4 Thesis structure

The content of this dissertation is organized as follows:

Chapter 1 briefs the key content of the thesis, including context, key contribution, and thesis organization. The main aim of this chapter is to give a quick context of the thesis.

Chapter 2 as the bibliography chapter, overviews generally the basic image low-level vision approaches and also partially about the evaluation techniques. The effort is made to briefly introduce the image low-level processing terms and methods including image processing needs, importance, application, filtering based on first order, filtering based on second order, and evaluation metrics. The experiments carried out in this chapter mainly focused on edge detection approaches and evaluation techniques. Edge detection techniques are usually chosen as an introductory section for most of the image filtering and segmentation subjects. The aim of this chapter is to establish a technical introduction and foundation for the subsequent chapters, with contributions involving advanced topics.

Chapter 3 as scale-space literature is devoted to reporting on the fundamental theory and state-of-the-art methods in scale-space analysis in image processing and computer vision. This chapter examines the basic theory and axioms of scale space, reviewing pioneer work in scale space vision, and demonstrates for example the anisotropic and causal filtering for contour and corner detection techniques. The aim of this chapter is to set up

the theoretical foundation and introduction for the multiscale analysis in low-level vision, particularly helping for our proposed multiscale line detection (introduced in chapter 5, and chapter 6) and the proposed new anisotropic corner detection technique (introduced in chapter 8).

Chapter 4 studies and objectively evaluate the ridges/valleys detection techniques (Shokouh et al. [2021a], Shokouh et al. [2021b]). The evaluation involves basic isotropic and anisotropic filtering techniques, steerable, oriented filters, and logical linear filter. This chapter contributes to highlight the optimization techniques; in investigating each studied technique's parameters characterization and configuration for ridge/valley detection and extraction techniques. The aim of this chapter is to provide a practical guide map for the state-of-the-art ridge/valley segmentation technique's important parameter setting, depending on the intended application.

Chapter 5 introduces the proposed multiscale line detection technique (Magnier et al. [2021]). This multiscale line detection consists of using the second order derivative of a semi-Gaussian filter for line detection. The aim and significance of this proposed technique are detailed in the chapter.

Chapter 6 introduces another multiscale line feature detection technique. The proposed method consists of benefiting the feature of anisotropic semi-Gaussian and bi-Gaussian filter. This method combines the semi-Gaussian and bi-Gaussian filters for optimal detection of multiscale lines in the condition of complex, close, and adjoint lines.

Chapter 7 contributes in objectively evaluating the repeatability quality of the 12 state-of-the-art filtering-based corner detection techniques for tracking complex underwater video frames. In this contribution, the repeatability rate of each filtering-based corner detection technique has been computed and both objectively and subjectively evaluated. The aim of this chapter is to serve both as the bibliography and introduction to filtering-based corner detection techniques for the proposed new anisotropic corner detection technique in the chapter 8.

Chapter 8 introduces a new anisotropic corner detection method. The proposed approach is based on causal filtering, which is evaluated comparatively with state-of-the-art corner detection operators. The new technique argues for better localization than compared techniques.

Chapter 9 concludes the thesis.

Chapter 10 contains the summary of thesis in French.

In a brief outline, the chapters of this report are arranged as following:

- **Chapter 1:** Introduction
- **Chapter 2:** Filtering Basics and Evaluation
- **Chapter 3:** Scale Space Analysis in Image Processing
- **Chapter 4:** Objective Evaluation of Ridge/Valley Detection Techniques
- **Chapter 5:** Multiscale Line Detection Using Second Order Semi-Gaussian Filter
- **Chapter 6:** Multiscale Line Detection Using Semi and bi-Gaussian Filter
- **Chapter 7:** Repeatability Evaluation of Filtering Based Corner Detection Techniques
- **Chapter 8:** New Anisotropic Corner Detection Technique

- **Chapter 9:** Conclusion and Perspective
- **Chapter 10:** Resume en Francais

Chapter 2

Image Filtering Basics and Evaluation

Contents

2.1 Introduction

In this chapter, the basics of image filtering are overviewed. Mainly, the filtering (linear, convolution, non-linear, segmentation, edge detection, first order and second order) is introduced. The aim of this introductory chapter is to cover all principle basics in low-level image processing in order to establish a basic context and foundation for the subsequent chapters with contributions, such as:

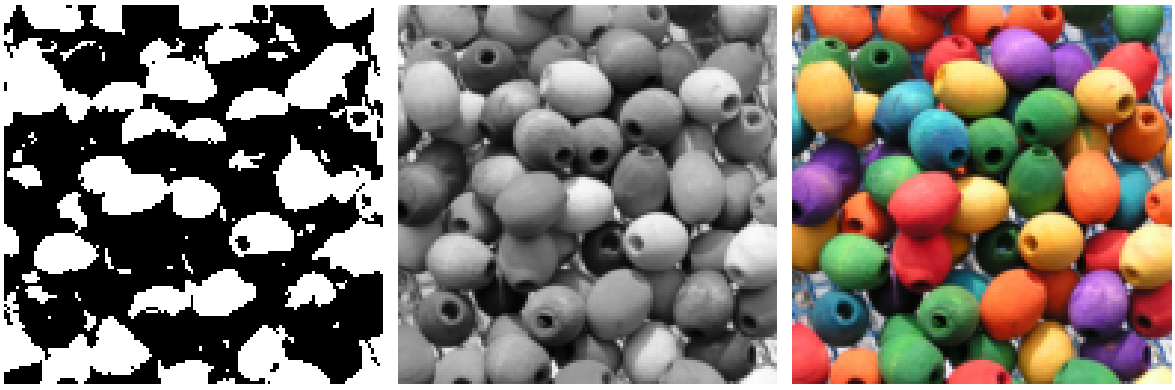
- Scale-space analysis in image processing: Basic theory and axioms
- Objective evaluation of ridges/valleys detection techniques
- Multiscale line detection using second order semi-Gaussian filtering

- Multiscale line detection using combination of bi-Gaussian and semi-Gaussian
- Repeatability evaluation of filtering based corner detection techniques
- New corner detection using anisotropic filter

Basically, an image is a spatial representation of a two-dimensional or three-dimensional scene, and correspondingly a digital image consists of a two-dimensional array or a matrix of pixels. Fundamentally, "pixel" means "picture element" and denotes the smallest elements or units of a digital image. Thereupon, a binary or boolean image is made of only two colors values (black and white), which by convention the black is represented by the numeric value of zero and the color white is represented by the numeric value of one, this is also known as Monochrome image. The binary representation of images is especially useful as it allows easy separation of an object from the background, which is done in the segmentation process by assigning the corresponding label to each pixel as either "background" or "object".

Accordingly, a digital grayscale image is a matrix with pixel elements containing numeric values in the range from 0 (black pixels) to 255 (white pixels), which is also called 8-bit color image format. The grayscale representations of images are useful for extracting descriptors directly from the grayscale image, benefiting both from the algorithmic simplification and computational complexity reduction.

Consequently, a color image is composed of three color channels, normally RGB (for Red Green, and Blue). If each channel is coded on 8 bits, one color pixel takes 24 bits, there exist $2^8 \times 2^8 \times 2^8 = 2^{24}$ color possibilities (256 possible colors for the red, 256 for the green and 256 for the blue, respectively).



(a) Binary Image (b) Gray-level Image (c) Color Image

Figure 2.1: Example of binary versus gray-level image and color image.

In order to represent an image mathematically, it can be formulated as a function, here called I, from $\mathbb{R}^2 \rightarrow \mathbb{R}$, where $f(x,y)$ gives the intensity at position (x,y). Practically, the image is defined over a rectangle, with finite range. Here to give an example of a binary image as following Eq. 2.1 image definition:

$$\mathrm{I} : [a,b] \times [c,d] \rightarrow [0,1], \tag{2.1}$$

which, a, b, c and d are the quantized discrete values. Likewise, a color image of three RGB channels is the composition of these three functions combined as following Eq. 2.2 color

image definition:

$$\mathrm{I}(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}, \tag{2.2}$$

where $r(x,y)$ red channel, $g(x,y)$ green channel, and $b(x,y)$ as blue channel denoted. In the computer memory, a discrete image is sampled in 2D-space (regular grid or matrix of integer values) and each sample is quantized (rounded to the nearest integer). If the samples are Δ apart, a discrete image can be mathematically demonstrated as following Eq. 2.3 discrete image definition:

$$\mathrm{I}[i,j] = \mathrm{Quantize}\{\mathrm{I}(i\Delta, j\Delta)\} \tag{2.3}$$

Image processing basically involves processing by modifying an image and its values. The aim is both to ameliorate its pictorial information for human interpretation and also rendering it more appropriate for autonomous machine perception. Practically, this processing of an image consists of enhancing an image, such as smoothing an image to remove unnecessary details or artifacts, sharpening the edge of an image to weigh the significant information in the image, improving the contrast for better visual appearance, highlighting some parts/objects in the image.

Image processing comprises a wide range of applications including noise or artifact removal -denoising, image enhancement, pre-processing related to the specific application, blurring and deblurring, segmentation, feature detection and description, object detection and classification, etc.

Image processing application mainly involves the following:

- **Removing noise, artifacts - denoising:** The first and ever demanding application of image processing is the removal of noise. Removing noises from the image is an essential step in almost all application of image processing and computer vision. In the definition of noise, noise is basically considered as unwanted, undesired and unnecessary information, which is any random error due to acquisition process or any other photometric artifact in an image. Noise is a very common problem in data transmission, all sorts of electronic components may modify or distort data passing through them, and the results may be undesirable. One major example of processing the image after the acquisition is removing the motion blur. In most of the image processing pipelines, the unnecessary details or artifacts must be removed as a pre-processing step prior to the main tasks. As some examples, denoising is used for efficient image restoration process (Buades. et al. [2004]). Likewise, for measurement or counting purposes using image processing and computer vision application, sometimes we may not be interested in all the details of an image, which we consider them as artifact to be removed. As instance, while considering machine inspected items on an assembly line, the only matters of interest may be shape, size or color. For such cases, we may want to simplify the image processing to only handle the things of interest and remove everything else, considering them as noise (Kishi et al. [1993]).

- **Image deblurring or image enhancement:** Removing the noise assumes the image is corrupted by additive noise, in which each pixel is corrupted by a noise value, independent of neighboring pixels. While the image blur, such as motion blur, happens when the object moves. It is like each pixel value is the sum of surrounding pixels, same as when we apply a smoothing or blurring filter. Image blur is general

artifacts, and it is hard to avoid. Image enhancement or deblurring is necessary to reduce blur amount from the image. Image deblurring is a process used to reduce the blur quantity in a blurred image and make the degraded image into a sharpened and clear image. When deblurring images, the cause of blurring is very important to increase the effect of the deblurring to get good result. Image deblurring can be done by inverse of blur filter. The first visual example in the Fig. 2.2 presents the usefulness of image deblurring, applying the non-linear transformation technique (Kramer and Bruckner [1975]). The second example for image enhancement using

(a) Original image (256×256)

(b) Deblurred image

Figure 2.2: Image deblurring example Kramer and Bruckner [1975].

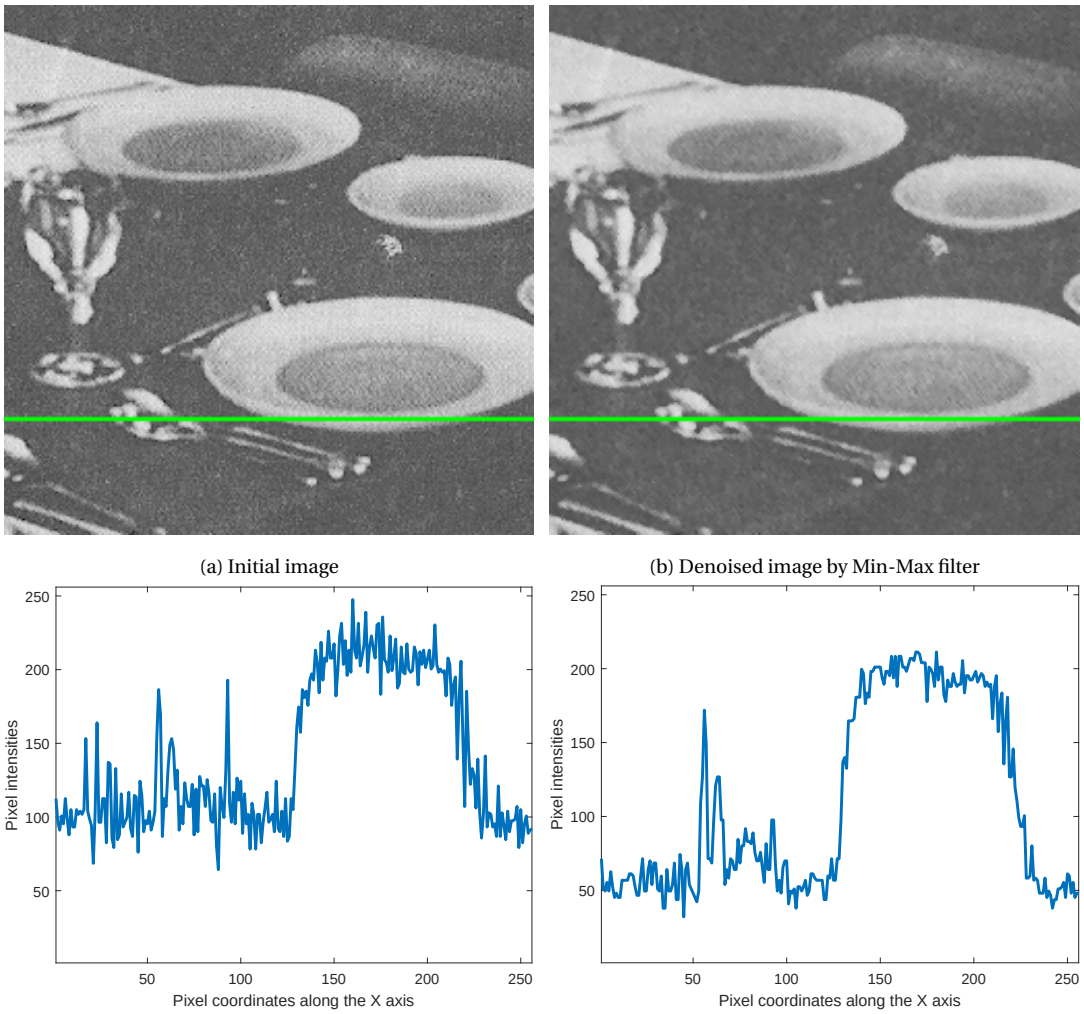
the Min-Max non-linear operator (Werman and Peleg [1985]) is illustrated in the Fig. 2.3. For quick visual understanding of how the image is enhanced, the 1-D signal by highlighting with green line is illustrated, which show both the signal in the original image and then in the enhanced image.

- **Feature detection and description:** The third example of needs of image processing for computer vision is obtaining and detecting the features of an image. This process is essentially necessary for the measurement of objects in an image, object detection, object recognition, tracking, calibration, stereo vision, homographic adaptation, image registration, etc. Edge and contour detection algorithms with their tied gradient information are the first step in edge enhancement and object detection (Komorkiewicz et al. [2012]; Leng et al. [2019]; Tuytelaars and Mikolajczyk [2008]).

  In the Fig. 2.4 the application of image processing in the feature detection and description is demonstrated. For the Fig.2.4 (a) this contour and corner feature can be used for the autonomous navigation, and likewise, in the Fig. 2.4 (b) detected contour and corner describe a house object.

Few common techniques frequently called and used in this manuscript, such as, segmentation, non-maximum suppression and thresholding is briefly introduced subsequently:

- **Image Segmentation:** Image segmentation as its name signifies is segmenting and subdividing an image into its separate parts or extracting and isolating certain parts/aspects of an image. For example, finding contours, edges, lines, or particular



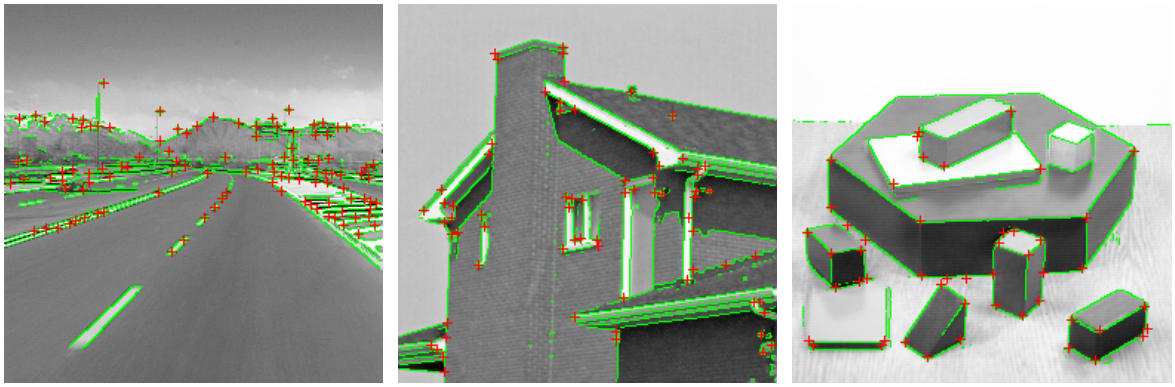
(a) Initial image

(b) Denoised image by Min-Max filter

(c) Original signal (green highlighted line in (a))

(d) Denoised signal (green highlighted line in (b))

Figure 2.3: Example of Min-Max non-linear filtering operation (Werman and Peleg [1985]).



(a) Contour and corner of the road image

(b) Contour and corner of the house image

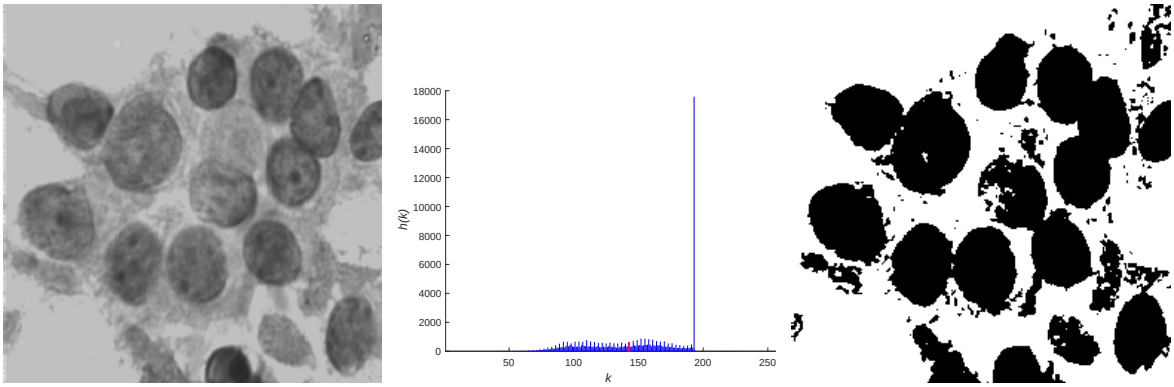
(c) Contour and corner of the blocks image

Figure 2.4: Example of feature detection (contour and corner) using Prewitt [1970] and Beaudet [1978] techniques for edge and corner detection, respectively.

shapes (circles, ellipses, etc.) or in an aerial/satellite images (identifying cars, trees, buildings, or roads, etc.). Low-level image segmentation applied in binary images is usually meant to detect edges, lines, ridges/valleys, crest lines, and corners or junctions. High-level image segmentation is useful for image analysis aiming to

recognize objects in a scene, giving the sense of sight to a robot, but also in many industrial applications in order to automate the control and inspection of production in detecting imperfect, defective and malformed products in the chain.

- **Thresholding:** Usually, a thresholding operation in image segmentation consists in obtaining a binary image from a gray-level image. Mainly, thresholding is the process of separating the background from the region of interest, by setting the minimum and/or maximum as a signal intensity threshold. There are some commonly used thresholding methods and in particular single thresholding, Otsu-thresholding, and hysteresis thresholding which is used by famous Canny edge detection method (see section 2.2.2). Simple thresholding produces an image in which all values below a given threshold have been put to zero and values greater than or equal to the threshold get set to one. Otsu-thresholding is determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance by reading through the image histogram. In this technique, the threshold value is found where the weighted variance between the foreground and background pixels is the minimum by iterating through all the possible values of threshold and measure the spread of background and foreground pixels (Otsu [1979]). The Fig. 2.5 illustrates the Otsu-thresholding applied on the gray-level image, and the red highlighted in the histogram shows the Otsu threshold. Thresholding by hysteresis, which repre-



(a) Original Image (b) Histogram of the thresholded image (c) Otsu-thresholding result

Figure 2.5: Otsu-thresholding operation and its histogram (the red highlighted in the histogram presents Otsu threshold) (Otsu [1979])

sents the most common thresholding method, often used for contour detection. In hysteresis thresholding method, two thresholds; low and high are used. The high threshold is used to select the reliable contour and the low threshold is used to select the continuous contour. For standard way of thresholding for segmentation purposes, it is critically important to be objective (applying the same algorithm to all the images in the experiments), consider reliability (check with many images when selecting the algorithm), and finally reproducibility (documenting and validating the algorithm as a standard work flow) (Glasbey [1993]).

- **Non-Maximum Suppression "NMS":** Non-local maximum suppression in image segmentation process is a contour (edge, line, junctions, etc.) or interest point thinning technique. The NMS algorithm check if it is a local maximum in its neighborhood. Regarding contour extraction, a local maximum is retained as an edge pixel in the direction of gradient. NMS outputs a more accurate representation of real edges in an image. Some edges are more bright than others. The brighter ones can

be considered as strong edges, but the lighter one can actually be edges, or they can be because of noise. For corner detection, extracted keypoints correspond to pixels that are a maximum in a neighborhood of a radius pixel or in a rectangular/square grid.

For the analysis of image's different contour types such as edge, ridge/valley, corner, it is essential to differentiate the various contour models of an image. Therefore, in the subsequent part, the image contour models are introduced briefly.

2.1.1 Image contour models

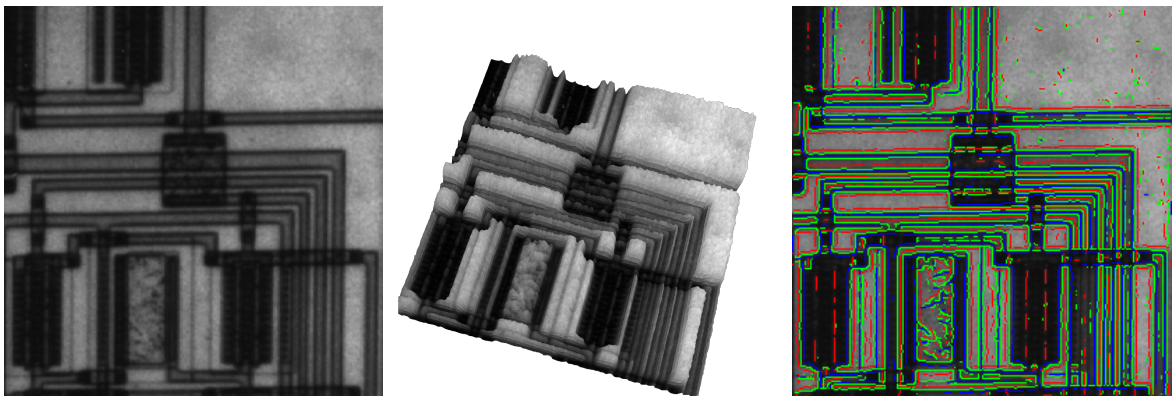
Generally, the contours of an image mean the transition of intensity signals (i.g. in binary image it means from black to white or vise versa). Bright intensity shows strong and rapid variations in the function of light intensity, for which the derivative operations can easily detect these transitions.

Nevertheless, the noises also correspond to small, rapid fluctuations of these signals, especially strong noise. This noise causes problems as soon as a derivative filtering is applied. Actually, the derivative operation amplifies the variations of the light intensity signal at the edges but also amplifies these small fluctuations.

To obtain usable result, it is necessary to take this noise into account. Contour detection usually rely on derivative operation but also on filtering and regularization methods as pre-processing stage, allowing to eliminate these small fluctuations while preserving as much as possible the important structures of the image.

Basically digital image consist of different contour types and models including: step-edge, line (ridges/valleys), crest line, ramp, etc. In the Fig. 2.6 the original image, its different contour models/profile in image surface in 3D and its segmented contour types are highlighted.

Fig. 2.7 further presents the different contour basic profile, such as mainly the most common types including: step edge, ramp edge, roof edge or crest line, and ridge/valley. It is mainly on these 1D and 2D contour models, on which much of the contour detection work is based.



(a) Original Image (b) Contour in Image Surface (3D) (c) Different contours highlighted

Figure 2.6: Contour models, which different contours segmented and highlighted: the green highlighted are step edges, the red highlighted are the line/ride/crest lines/roof edges, and finally the blue highlighted are the valleys.

To clarify a bit, the difference between contour and edge detection; edge detection just gives points where image intensity changes drastically. It may or may not form a closed

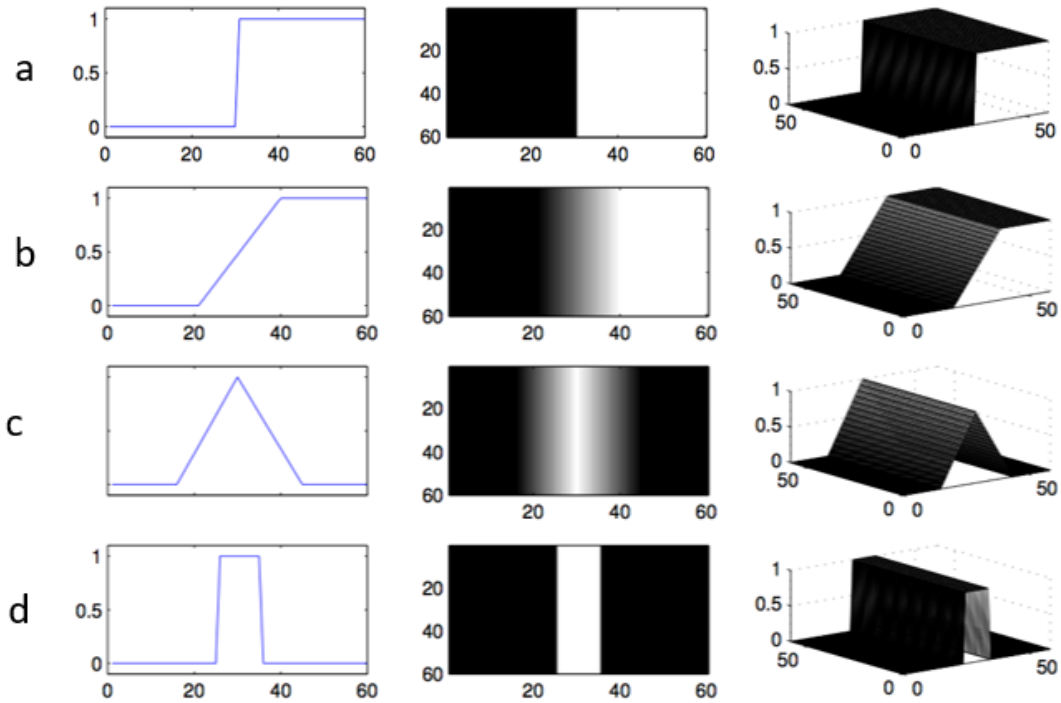


Figure 2.7: Different contour profile. a) Step edge. b) Ramp edge, c) Roof edge - crest lines. d) Peak edge - ridges/valleys (Magnier [2011])

shape. Contour is the edge closing an object. So you can think as higher level of edge detection (Papari and Petkov [2011]). So if an edge define an object, it becomes a contour.

2.1.2 Image filtering

In image processing, a filter is usually used to smooth or sharpen an image, to extract some structures underlying the image via suppressing certain pixel values. Filtering an image is a low-level process where it can notably be implemented in: linear, non-linear, isotropic, anisotropic or recursive methods. It is useful either to get better information (e.g., enhancing contrast, removing noise, etc.) or to transforming an image to make it suitable for further processing.

This is to note that the term filter, mask, kernel and convolution/correlation matrix are generally used synonymously, but sometimes maybe used in a context to emphasize a certain meaning.

Filtering is an operation that takes an image as input and produces a new image as output with changing certain pixel values such as coefficient values of kernel, either normally or giving weights to central coefficient (weighted kernels). There is no solid definition of best filter, which filter needed or is defined as best mostly depends on the context and application.

Mathematically, filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. For linear filtering, the output of the filter is the linear combination of the neighborhood pixels. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel.

There are many commonly used filter in image processing such as mean filter or box

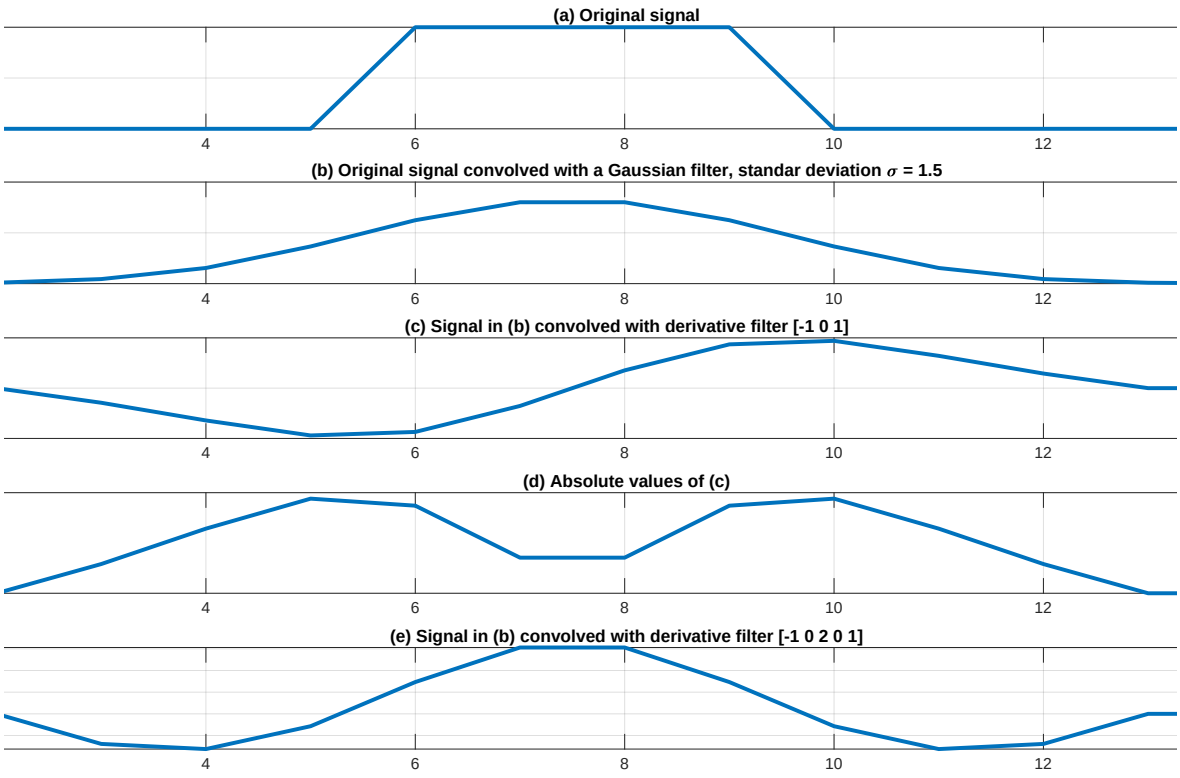


Figure 2.8: 1D Signal (Magnier [2011]): low-pass vs high-pass filtering: a) Original signal, b) Smoothing (low-pass), c) 1st derivative (high-pass), d) Absolute value of (c), e) 2nd derivative (high-pass) examples

filter, Gaussian filter called as low-pass filter, high-pass filter such as derivative filter, Laplacian, etc. As a simple example, the mean filter contains the same coefficients values for the filter; it replaces each pixel with an average of its neighborhood and consequently achieve smoothing effect by removing sharp details/features.

In the Fig. 2.8, the original 1D signal, low-pass, high-pass (first order) and high-pass (second order) signal filtering is illustrated orderly. In the illustration, the low-pass filter smoothed the original ramp signal. Regarding the first order operation (high-pass), the peak/maximum value determines the intensity change (i.e., representing the contour of type edge), while for the second order derivatives (high-pass) the contour of type's edge (step edge) is determined by zero crossing.

There are linear filtering and non-linear filtering methods. Linear filtering occurs when the operation performed on each pixel is a simple mathematical operation, in which the result is the output of the linear combination of the input's neighboring pixels. Mean filter is a typical example of linear filtering. For example, if we are multiplying the intensity of each pixel by 2, then the entire image gets intensified by a factor of two, which means we have effectively multiplied the image matrix by 2. On the other hand, in a non-linear operation such as median filter, the overall effect on the image cannot be predicted just by the operation performed on each pixel. For example, squaring each pixel is not the same as squaring the image matrix.

2.1.2.1 Linear filtering

Linear filtering is the process of filtering with properties of linearities such as following the laws of associativity, shift-invariance, commutativity and distributive property. Linear fil-

tering of images is implemented through two-dimensional convolution or correlation in case of symmetric filter/kernel, such as isotropic Gaussian kernel. In convolution, the value of an output pixel is computed by multiplying elements of two matrices and summing the results. Correlation is similar to convolution, with the only difference being that the kernel is flipped 180 degrees before the process. One of these matrices represents the image itself, while the other matrix is the filter.

In the continuous case, the convolution of two functions in 1D is defined as the following Eq. 2.4 :

$$h(x) = \mathrm{I}(x) * g(x) = \int_{-\infty}^{\infty} \mathrm{I}(t) \cdot g(x-t)\,dt, \tag{2.4}$$

where the $f(x)$ and $g(x)$ can denote image and kernel, the kernel size in 1D, and an integration variable t. One of the two functions is returned ($t \rightarrow -t$ for example $\mathrm{I}(t) \rightarrow \mathrm{I}(-t)$), then its origin is shifted by x, we obtain the function $\mathrm{I}(x-t)$. Finally, the two functions are multiplied together point by point, the convolution of the two functions is obtained by integration according to the variable t in the interval $[-\infty,\ \infty]$. Then, the function $h(x)$ of the variable x is obtained.

Likewise, in the discrete case, the convolution is now written with discrete sums, such as Eq. 2.4:

$$h(x) = \mathrm{I}(x) * g(x) = \sum_{l=-\infty}^{\infty} \mathrm{I}(l) \cdot g(n-l). \tag{2.5}$$

The convolution is a linear operation, having the commutative, associative and distributive properties. Convolution in 2D, in the case of images, the luminous intensity function is written $\mathrm{I}(x,y)$, it is a 2D function, and each pixel (x,y) corresponds to its luminous intensity. In this 2D case, the convolution of two functions will be written in the continuous case as Eq. 2.6, where l and m denoting the kernel dimension.

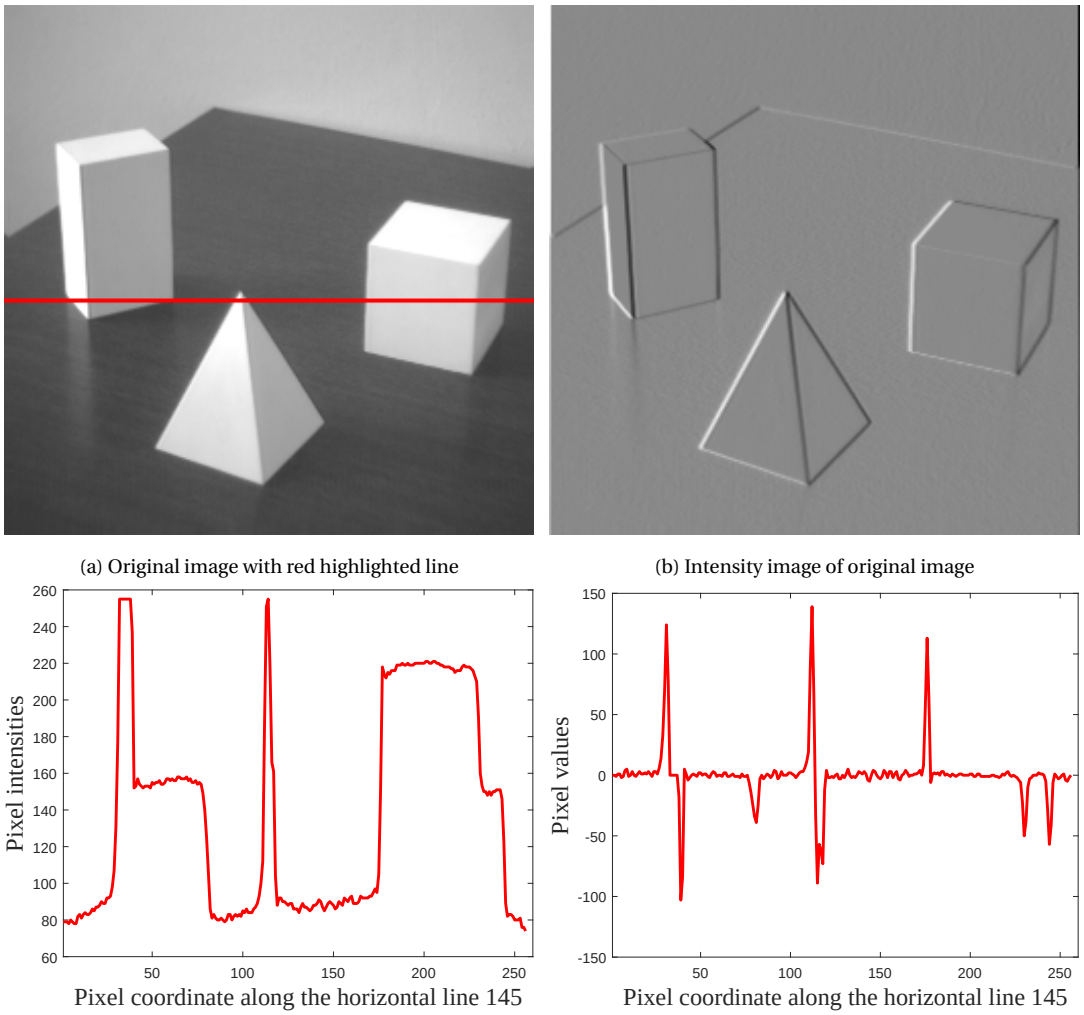
$$h(x,y) = \mathrm{I}(x,y) * g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{I}(l,m) \cdot g(x-l, y-m)\,dl\,dm. \tag{2.6}$$

The application difference between cross-correlation and convolution is that normally the correlation is used for similarity measurement and matching purposes, whilst the convolution operation is used for filtering. That is why the above notation of $\mathrm{I}(x,y)$ is called filter based on the objective of operation. In the discrete case of the convolution of an image with a filter, as for the one-dimensional case, the continuous integrals are replaced by discrete sums, such as Eq. 2.7:

$$h(i,j) = \mathrm{I}(i,j) * g(i,j) = \sum_{l=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} \mathrm{I}(l,m) \cdot g(i-l, j-m). \tag{2.7}$$

For the illustration of the convolution, the derivation filter [−1 0 1]on the light intensity profiles in Fig. 2.9 is applied, each significant peak marks an image contour. An example of a noisy image and intensity profile of the green highlighted line is given in Fig. 2.10 (a) and (b). Finally, the light intensity profile of the derived image at the highlighted line is shown in Fig. 2.10 (c). It appears clearly on this example that the noise this time is preponderant, and that the filter presented above will not allow extracting contours in this image.

The noise is visibly greater than on the initial profiles. Under certain conditions, this noise can become greater than the peaks of the contours we want to detect, for example, a blurred contour, whose slope is weak, can present a weaker signature than the noise. A variation in lighting may cause a reduction in the contrast between the "white" and "black" zones, from where the obtaining of weaker peaks, therefore possibly lower than the peaks due to the noise.



(a) Original image with red highlighted line

(b) Intensity image of original image

(c) Pixel values of highlighted line 145 in (a)

(d) Intensity pixels of highlighted line 145 in (b)

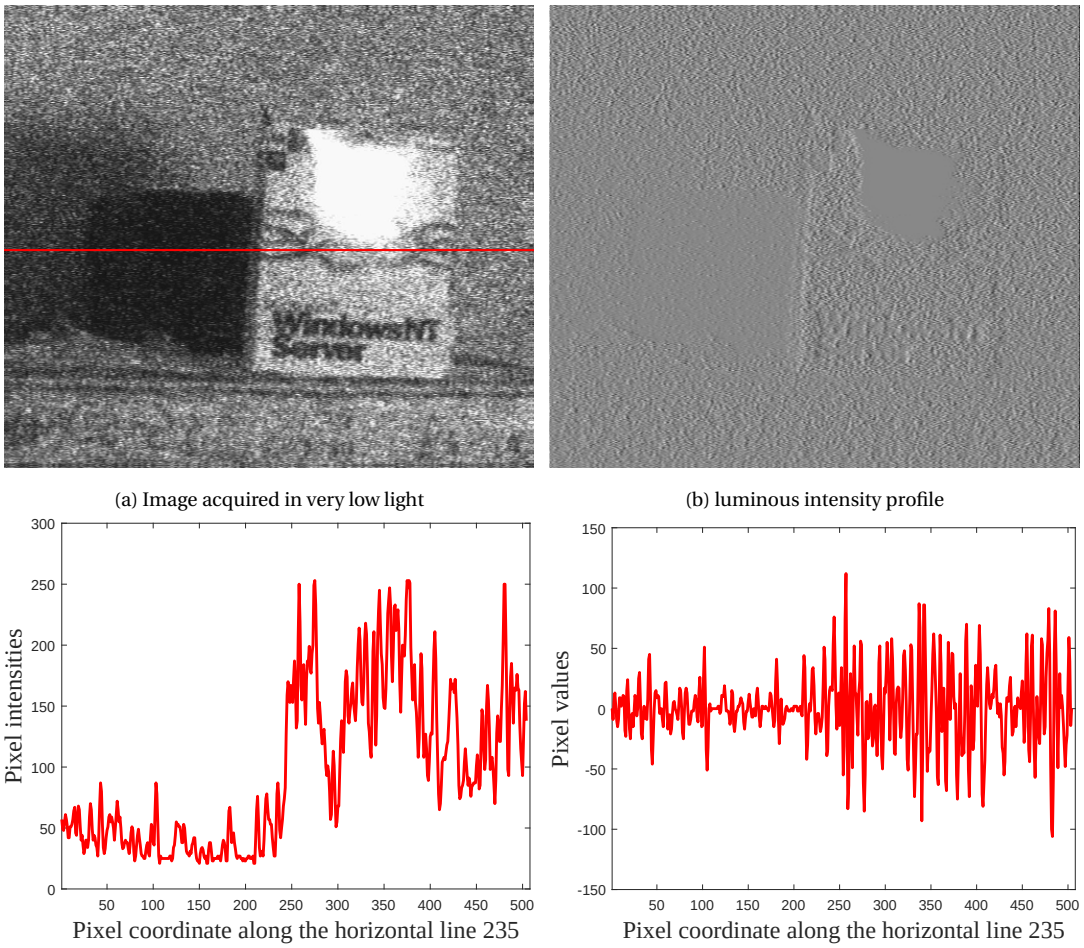
Figure 2.9: Discrete derivative of the luminous intensity function, and comparing the normal pixel values of a randomly chosen line and its intensity values derived from the derivative along the x-axis.

2.1.2.2 Non-linear filtering

Besides the convolution methods, there are many other filtering approaches to implement through sliding neighborhoods. Many of these techniques are non-linear in nature. For example, the sliding neighborhood operations can be applied in which the value of an output pixel is equal to the standard deviation of the values of the pixels in the input pixel's neighborhood.

The well-known median filter as a non-linear filter can also be used for smoothing and noise suppression purposes in the digital image. Median filter basically replaces each pixel by the median of its neighboring pixels. As a result, it removes some spikes introduced by noise, such as, impulse or salt and pepper noise. By applying the median filter, the stand-alone noise pixels with extreme intensities like black and white will be removed. The second advantage of median filter is that it does not introduce new pixel values since it only re-use existing pixel values from the window.

Further, unlike other averaging filters, such as mean and Gaussian, it removes noise while keeping the contour or edge information. Therefore, the median filter which is a non-linear filter can't be implemented via convolution process (which is a linear opera-



(a) Image acquired in very low light

(b) luminous intensity profile

(c) Highlighted line pixel intensity of (a)

(d) Highlighted line pixel values of (b)

Figure 2.10: Example of noisy image demonstrating the sensitivity of derivative with noise signals

tion).

We can observe in the Fig. 2.11 that when the noise level is too high, although the amount of noise pixel decreases with increasing Gaussian filter size, they still exist in the image. Median filter, on the other hand, already remove most noise pixels with 3 x 3 filter size. By applying larger filter size, median filter further exclude noise pixels, but it also diminishes a lot of image information, details and structure.

(a) Original Image

(b) Image with salt and pepper noise

(c) Image after median filter is applied

Figure 2.11: Example of median filter on the image with impulse or salt and pepper noise.

2.1.3 Smoothing and regularization

Each type of noise in the image has specific properties, so it is interesting to look locally at the signal (in a given support) to try to reduce it by taking local averages of the image signal. Image filtering complexity increases with the square of the size of the mask. Unfortunately, if the size of the mask increases, the number of operations to perform per pixel becomes quickly prohibitive.

For example, a mask of size 25×25 will cost 625 operations per pixel. In order to remedy this problem, we can explore the mathematical properties of filters to find ways to reduce this complexity. There are many mean regularization filters in particular: Gaussian filter, Deriche filter, Shen filter, etc.

Some filters can be written as the product of two one dimensional (1D) filters, one in X and the other in Y. The product of convolution can then be implemented by a sequence of two 1D convolutions. To return to the example of the average filter, this filter can be mathematically written as the product of two functions.

At first, we will not take into account the normalization coefficient of the filter. 2D convolution becomes separable and can be written as two successive 1D convolutions in X then Y (or vice versa). Now concerning the algorithmic complexity; in 2D the complexity is $N \times M$, while in separable 2D, the complexity decreases as $N + M$.

In the case of separability of the filter, it needs to be decomposed into two 1D filters of size N for the filter in X and of size M for the Y filter. The second way for faster implementation of convolution is to implement it via the recursive way.

In recursive implementation of a filter, the algorithmic complexity decreases with the total cost of four operations per pixel. First, the filter is applied in the horizontal direction/ x-axis of the input image with the cost of two operation per pixel, for which a temporary image is obtained. Then, the filter is implemented to the vertical direction/ y-axis to the temporary image previously obtained, with a cost of two operations per pixel, and finally the desired filtered image is obtained.

A low-pass filter (in the context of image in frequency domain) is the basis for most smoothing methods. An image is smoothed by decreasing the disparity between pixel values by averaging nearby pixels. Using a low-pass filter tends to retain the low frequency information within an image while reducing the high frequency information. An example of a low-pass filter is an array of ones divided by the number of elements within the kernel.

In the following parts, the usual low-pass/smoothing filters are introduced with demonstration, which first the mean filtering as an intuitive example is introduced.

2.1.3.1 Mean filtering

Mean filtering is a simple, intuitive and easy to implement method of smoothing images, i.e., reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images.

The idea of mean filtering is simply to replace each pixel value in an image with the mean (average) value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter. Like other convolutions, it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 square kernel is used, although larger kernels (e.g., 5×5 squares) can be used for more severe smoothing.

Note that a small kernel can be applied more than once in order to produce a similar but not identical effect as a single pass with a large kernel.

The average/mean/box filter smooth images by reducing the variation of intensities between the neighboring pixels. In Eq. 2.8, the $f(x,y)$ formulates the 3×3 mean kernel, and the Eq. 2.12 demonstrate the mean filtering result, showing the smoothing effects of small vs larger filter size.

$$f(x,y) = 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.8}$$

(a) Original Image (b) Mean filtering of filter size (3×3) (c)Mean filtering of filter size (7×7)

Figure 2.12: Example of mean filtering result with different filter size

The major problem of applying the box filter or mean/average filter is that it introduce the ringing artifacts, losing large portion of fine image detail. Mean that a single outlier or outstanding value can significantly affect the average of all the pixel values in the neighborhood. So, it can blur the contour such as edge, line and corner.

Therefore, to keep the property of smoothing in order to remove noises in images, another optimal alternative is to use weighted mean such as Gaussian smoothing filter.

2.1.3.2 Gaussian filtering

The Gaussian smoothing filter as a low-pass filter is a 2D convolution operator that is used to smooth images and remove detail and noise. Gaussian filter is a useful filter which has the following interesting properties:

- **Steerability:** or orientability; as a Gaussian filter is basically symmetric, it is efficient when the filter is oriented/steered in different direction when applied.
- **Separability:** Gaussian filter can be applied in a computationally efficient way: meaning that larger filters (e.g. 2D) can be decomposed into smaller 1D filter in order to obtain speed efficiency while implementation.
- **Recursive implementation:** Gaussian filter can also be applied recursively with some approximation methods, such as, Deriche approximation. As mathematically there is no Z-transform of Gaussian, it can't be directly implemented in recursive way.
- **Scale parameter:** the degree of smoothing is controlled and determined by the given standard deviation in the Gaussian formula, usually denoted σ.

- **Low-pass filter:** Gaussian as a low-pass filter is very effective for filtering the high frequency components of the signals/images, usually modeled as Gaussian noise.
- **Gaussian Pyramid:** Essential when down-sampling images, particularly for the scale space analysis and solutions of images.
- **Fourier's decomposability:** The Fourier transform of a Gaussian is also Gaussian, which is an interesting property for studying the Gaussian in frequency domain. In contrast, the Fourier of mean filter/box filter is not the same. In time domain it is square, while in frequency domain it is more sinusoidal shape. Therefore, we can do multiplication which is much faster than convolution, because the convolution in time domain correspond to product or multiplication in the frequency domain.

Mainly, most of the noises in image processing appear as Gaussian noise, so using the Gaussian smoothing filter is one of the optimal solution to remove or reduce noises and artifacts. By analyzing the types of observed noise (e.g., Gaussian noise in frequency domain seems as high frequency components), we can apply the Gaussian filter to suppress those high frequency component and also control the level of removal by the standard deviation σ parameter.

Though, Gaussian filter is roughly similar to the mean filter (weighted mean specially), but it uses a different kernel that represents the shape of a Gaussian (bell-shaped, see Fig. 2.13). Therefore, the Gaussian filter is an optimal filter comparing the mean filter, because with the Gaussian filter the level of smoothing and regularization can be more precisely controlled via the given σ parameter.

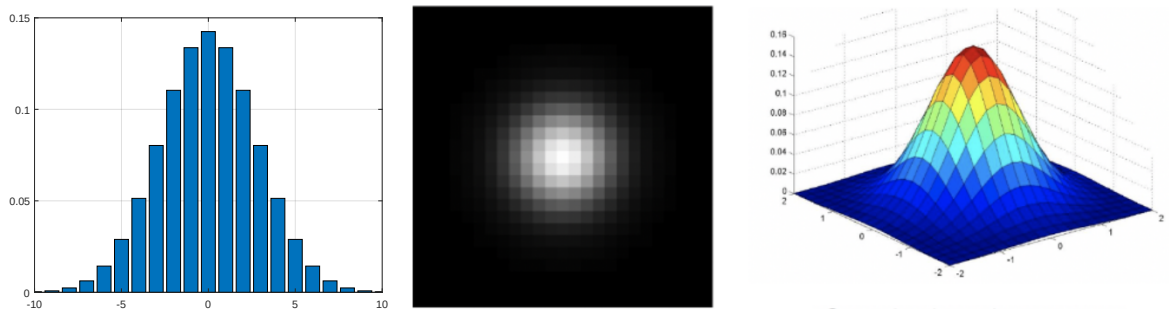
Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scale. The Gaussian distribution in 1D and 2D continuous domain is formulated as Eq. 2.9, and Eq. 2.10:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{2.9}$$

In 2D, an isotropic (i.e. circularly symmetric) Gaussian has the form:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.10}$$

where σ is the standard deviation of the distribution. We also assumed that the distribution has a mean of zero (i.e., it is centered on the line x=0). The distribution is illustrated in Fig. 2.13



(a) 1D Gaussian distribution, $\sigma = 2.8$ (b) 2D Gaussian (isotropic property) (c) 3D Gaussian/ surface

Figure 2.13: Gaussian Kernel: a) 1D Gaussian distribution, b) 2D Gaussian (isotropic property) with zero mean, c) 3D Gaussian /surface

2.2 Edge detection based on first order derivatives

In image processing, roughly, edge detection as a fundamental tool is to find and characterize the boundaries of objects within an image, which has many applications including: detecting junctions, corners, segmenting the image, detecting shapes, analyzing the scene, etc. As presented in Fig. 2 of James [2016], the problem of edge detection is still one of the state-of-the-art research subject with the scope to optimize the precision and efficiency (Ziou and Tabbone [1998]). Edge detection in image processing means to find discontinuities or sharp changes in image brightness; and in 1D signals is known as step detection, as previously shown in Fig. 2.7.

The aim of identifying these discontinuities or abrupt changes in pixel intensities is to extract and interpret important information, such as, discontinuities in surface orientation, discontinuities in depth, etc.

2.2.1 Edge detection basic filters

Edge is where the intensity is not continuous and is the maxima of the derivative. Gradient is equal to the square root of image derivative in both x and y direction. The magnitude of the gradient shows the strength of the edge (how quickly the intensity is changing), while the gradient direction shows the direction in which the image intensity is changing. Because the derivative operators are sensitive to noise, it is essential to eliminate first the high frequency or do edge labeling by increasing the signal-to-noise ratio (SNR) via the smoothing operators.

To obtain a smoothed image, as mentioned in the previous sections, there are mean smoothing (easy and uniform filtering), Gaussian smoothing (weighted averaging), exponential filtering operators, etc. There could be many methods to implement the derivative operators; simple methods based on the use of convolution masks approximating the operator gradient by finite differences, and more sophisticated methods based on optimal approaches with search for an ideal contour model.

In all cases, the filtered images are computed by convolution of the input image with directional masks approximating the derivative operators. Hence, in a two-dimensional framework, it is needed to estimate derivatives in two different direction, i.e. in general the directions of the axes of the image X and Y. One finds in the bibliography (Roberts [1965]; Sobel and Feldman [1968]; Prewitt [1970], etc.), many different approximations of the partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$.

2.2.1.1 Sobel and Prewitt filters

Sobel and Feldman [1968] calculate the approximate opposite of the gradient of the image intensity function. It uses convolution kernels of dimensions, usually 3×3 for the purpose of getting the gradient at each point. The gradients in the direction of each axis are used for determining whether the pixel is part of an edge or not. Prewitt [1970] and Sobel and Feldman [1968] proposed 3×3 masks along the X and Y axes to estimate the gradient ∇I in an image:

$$\nabla \mathrm{I} = \begin{bmatrix} \frac{\partial \mathrm{I}}{\partial x} \\ \frac{\partial \mathrm{I}}{\partial y} \end{bmatrix} = \begin{bmatrix} \mathrm{I}_x \\ \mathrm{I}_y \end{bmatrix}$$

The orientation of the gradient with respect to the X axis is given by: $\theta \mathrm{I} = arctan\left(\frac{\mathrm{I}_y}{\mathrm{I}_x}\right)$. To do so, they proposed two masks H_1 and H_2 computed the horizontal and vertical image derivatives respectively:

$$\mathrm{H}_1 = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix}, \quad \mathrm{H}_2 = \begin{bmatrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{bmatrix}$$

, with $c = 1$ for Prewitt and $c = 2$ for Sobel. Here, for example, $\mathrm{H}_1 1$ can be decomposed into a gradient operator in X and a smoothing operator in Y is:

$$\mathrm{D}_{1x} = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}, \quad \mathrm{S}_{1y} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

So, $\mathrm{H}_1 = \mathrm{D}_{1\mathrm{X}} \cdot \mathrm{S}_{1\mathrm{Y}}$. Denoting D as derivative operation and S as smoothing operation, we can decompose H_2 similarly into a smoothing mask in X and a differentiation mask in Y, as $\mathrm{I}_1 = \mathrm{H}_1 * \mathrm{S}$ and $\mathrm{I}_2 = \mathrm{H}_2 * \mathrm{D}$. The enhanced image will be obtained by calculating the norm of the gradient, by example: $|\nabla \mathrm{I}| = \sqrt{\mathrm{I}_x^2 + \mathrm{I}_y^2}$, and the image of the orientations of the gradient will be calculated by:

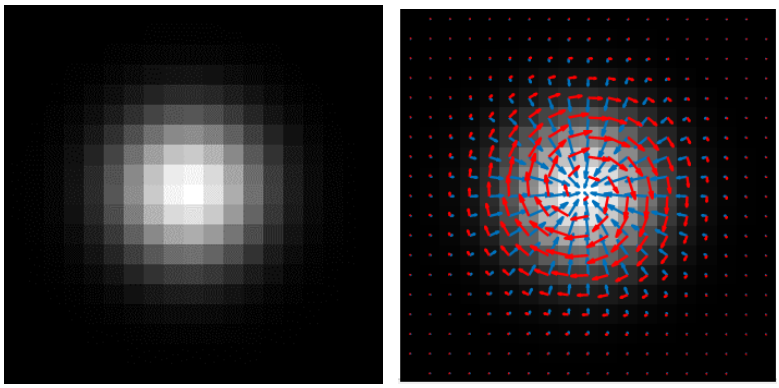
$$\Theta = arctan\left(\frac{\mathrm{I}_2}{\mathrm{I}_1}\right)$$

The Fig. 2.14 illustrates the image gradient and gradient direction. These methods used

(a) Initial image (b) Image gradient (c) Gradient direction/angle

Figure 2.14: Example of Image gradient and its direction

with the suppression of non-local maxima according to the direction of the gradient, and



(a) Initial image (b) Image isophotes and gradient direction

Figure 2.15: Image gradient and isophotes direction highlighted in red and blue respectively.

with hysteresis thresholding, give quite good results (when the noise is low) with low algorithmic complexity. Note that it is also possible to implement these filters in recursive form. Other methods proposed by Kirsch [1971] consist in filtering the image with 8 directional masks, and then obtaining the gradient and orientation of the contour as the following formulation (see the Kirsch [1971] for the details of the given equation).

$$\mathrm{G}(i,j) = \max_{k=1,8}(\mathrm{I}_k(i,j))$$

2.2.2 Edge detection based on optimal filtering

The problem with normal/basic derivative filter or filtering by only taking the difference of two subsequent pixels are that, they are too sensitive to noise, too many contours are detected and likewise they are sensitive to horizontal and vertical edges than the others, in spite of the advantage of speed performance and easiness. In the context of optimal filtering, the framework of separable filters or one-dimensional filtering is considered. Canny [1983] proposed an optimal filter for the detection of an ideal contour embedded in white Gaussian noise. The proposed filter is optimal in localization and maximizes the signal-to-noise ratio. For Canny, an ideal contour is represented by a Heaviside function, as Eq. 2.11.

$$\mathrm{H}(t) = \begin{cases} 0 & \text{if} \quad t < 0 \\ 1 & \text{if} \quad t \geq 0 \end{cases} \tag{2.11}$$

An image containing a contour at $x = 0$ can therefore be modeled by Eq. 2.12:

$$x(t) = \mathrm{AH}(t) + n(t) \tag{2.12}$$

where A represents the amplitude of the jump and $n(t)$ a Gaussian white noise of variance $\sigma^2 = \mathrm{E}(n^2(t))$. Canny assumes that the ideal filter sought is linear and antisymmetric like a gradient operator. The sharpened image will be calculated by Eq. 2.13

$$y(t) = \int_{-\infty}^{\infty} x(t')\mathrm{C}(t-t')\,dt' = \tag{2.13}$$

The accentuator $\mathrm{C}(t)$ is chosen to maximize the signal-to-noise ratio of the filtered image at the edge point. The optimal edge detection general structure from input image, derivative operation in both X and Y direction, gradient, gradient direction, NMS and final thresholding is presented in Fig. 2.16

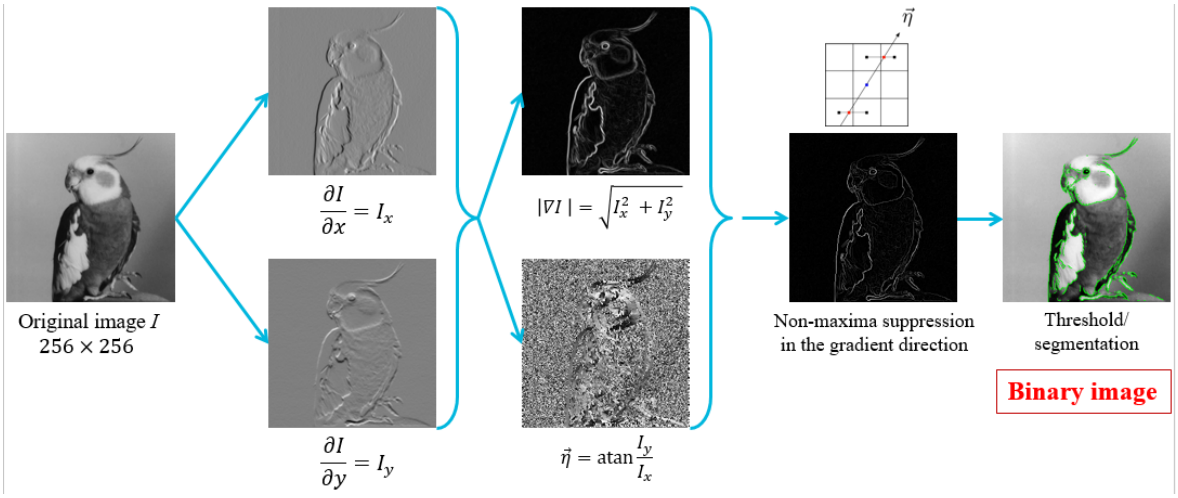


Figure 2.16: Edge Detection Overview

2.2.2.1 Canny optimal filtering

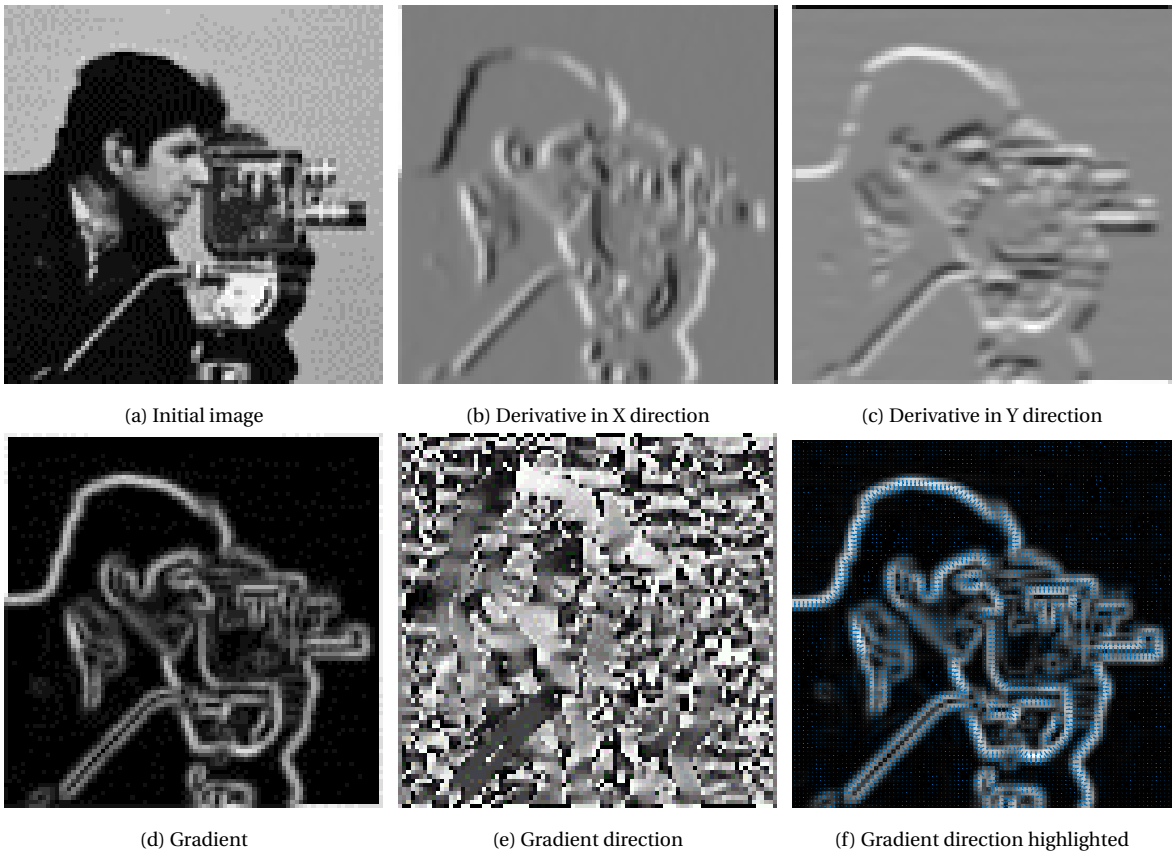
Canny's aim was to discover the optimal edge detection algorithm (Canny [1983]). In this situation, an 'optimal' edge detector means that it marks as many real edges as possible, that the marked ones are as close as possible to the real edges, and noise effects are avoided also as much as possible. To do so, Canny's algorithm uses the calculus of variations, which is a technique that finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.

To reduce noise, it uses a filter based on the first derivative of a Gaussian, giving as a result a slightly blurred version of the original image, not affected by a possible single noisy pixel. Since an edge may point out in a variety of directions, the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator returns a value for the first derivative in the horizontal and vertical directions. From this, the edge gradient and direction can be determined as presented in Fig.2.17.

Later, from a stage referred to as non-maximum suppression, a set of edge points, in the form of a binary image, is obtained. These are sometimes referred to as "thin edges". Then, after thresholding is complete (using also a hysteresis process), a binary image is obtained where each pixel is marked as either an edge pixel or a non-edge pixel, presented in Fig. 2.18.

This is one of the most efficient and successful edge detection methods. It utilizes a multi-stage algorithm, operating on the gray-scale version of the image under consideration.

The algorithm involves computations of the rate of change of pixel values in any particular direction. If the change is very high, then, similar to the basic algorithm we discussed before, the pixel is expected to be part of an edge. The algorithm also checks whether the gradient is a local maxima at that point, which helps it to be more precise. The thresholds of this edge detector are also very important. The upper threshold specifies the value above which a gradient would definitely be considered part of an edge. As for the lower threshold, gradients below it are to be discarded entirely, while values in between the two are to be investigated for possible edge linking, i.e., if nearby pixels happen to be part of an edge or not, and if they fulfill other criteria, which in the Fig. 2.19 the effi-



(a) Initial image (b) Derivative in X direction (c) Derivative in Y direction

(d) Gradient (e) Gradient direction (f) Gradient direction highlighted

Figure 2.17: Example of Canny edge detection approach involving gradient norm and gradient direction.

ciency of canny edge detection via Gaussian smoothing is self explained. To summarize, Canny characterized the quality of good edge detection as:

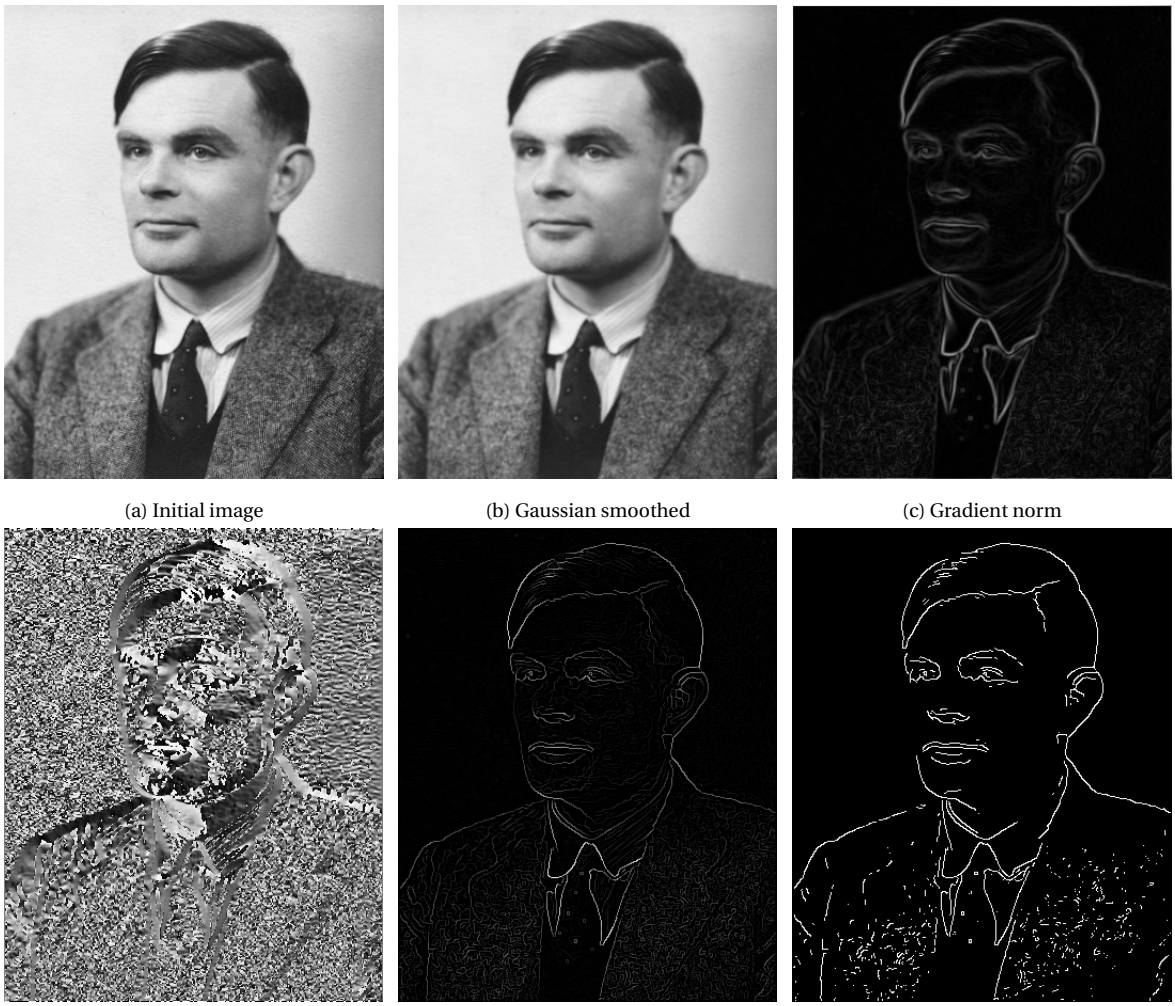
- Good detection: means the detector must minimize the false positive and false negative.
- Good localization: the detected edges must be close to the true edges (compared against ground truth)
- Single response: the detector must have one response for each edge point.

As explained previously in detail the main challenge of edge detection with derivative operator is the noise, where the derivative operator amplifies the noise and the noise should be suppressed to have optimal result as illustrated in Fig. 2.20

2.2.2.2 Deriche optimal filtering

The Deriche filter is basically a low-pass/smoothing filter which was designed to optimally detect, along with a derivative operator, the contours in an image (Canny criteria optimization) Deriche [1987]. Besides, as this filter is very similar to a Gaussian filter, but much simpler to implement (based on simple first order Infinite Impulse Response (IIR) filters), it is also used for general image filtering.

Deriche method is a multistep algorithm used to obtain an optimal result of edge detection in a discrete 2D image. Consequently, this algorithm is the combination of Canny's optimal edge detection criteria and his (Deriche) criteria for optimal edge detection, which follows:



(a) Initial image (b) Gaussian smoothed (c) Gradient norm

(d) Gradient direction (e) Non-Maximum Suppression(NMS) (f) Thresholded

Figure 2.18: Example of Canny edge detection implementation and result

- Detection quality - all existing edges should be marked, and no false detection should occur.
- Accuracy - the marked edges should be as close to the edges in the real image as possible.
- Unambiguity - a given edge in the image should only be marked once. No multiple responses to one edge in the real image should occur.

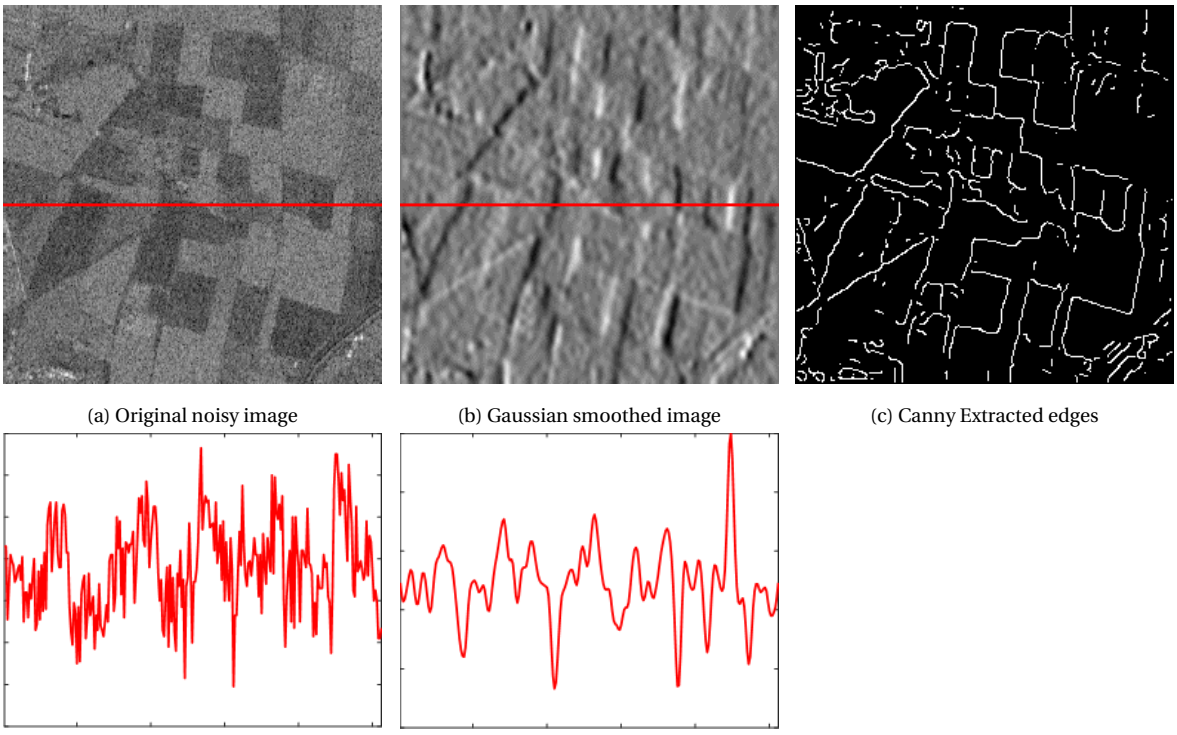
Deriche edge detector, like Canny edge detector, also includes the smoothing, magnitude with gradient direction, NMS and hysteresis thresholding. The difference between the Canny and Deriche approach in the edge detection is the first two steps, in which the Deriche edge detector uses the IIR filter. After conditional limits, the derivative operator in 1D is given by Eq. 2.14:

$$D(x) = x \cdot e^{-\alpha \cdot |x|}, \tag{2.14}$$

whereas the low-pass filter in 1D is computed by Eq. 2.15:

$$S(x) = (\alpha \cdot |x| + 1) \cdot e^{-\alpha \cdot |x|}. \tag{2.15}$$

The filter optimizes the Canny criteria. The advantage of this filter is that it can be adapted to the characteristics of the processed image using only one parameter. If the value of α



(a) Original noisy image (b) Gaussian smoothed image (c) Canny Extracted edges

(d) Intensity pixels on red highlighted in (a) (e) Intensity pixels on red highlighted in (b)

Figure 2.19: Example of Canny edge detection efficiency on noisy images. For direct understanding, the derivative of red highlighted lines with no smoothing and also with Gaussian smoothing is plotted

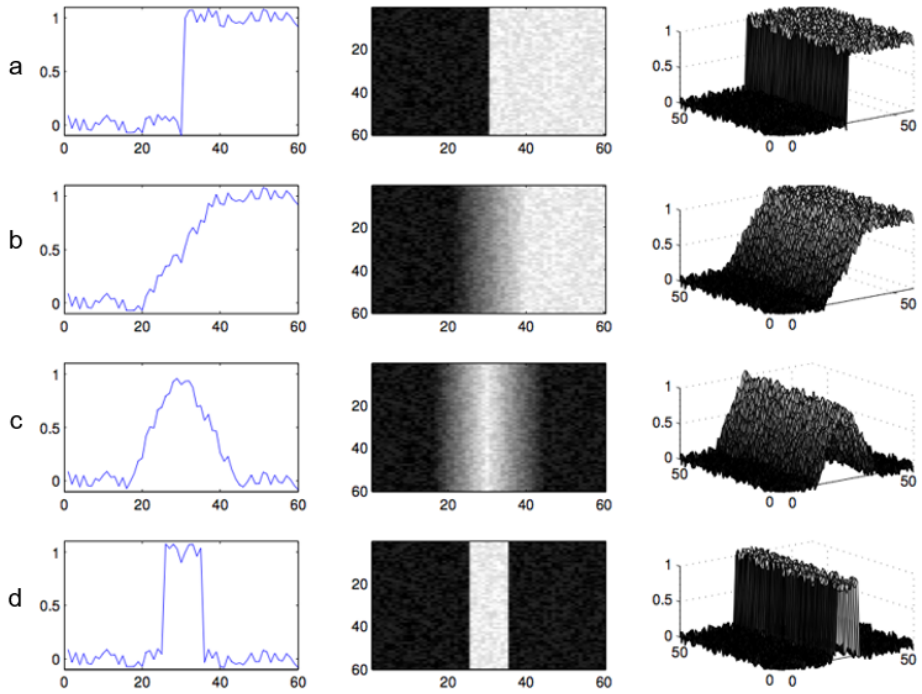
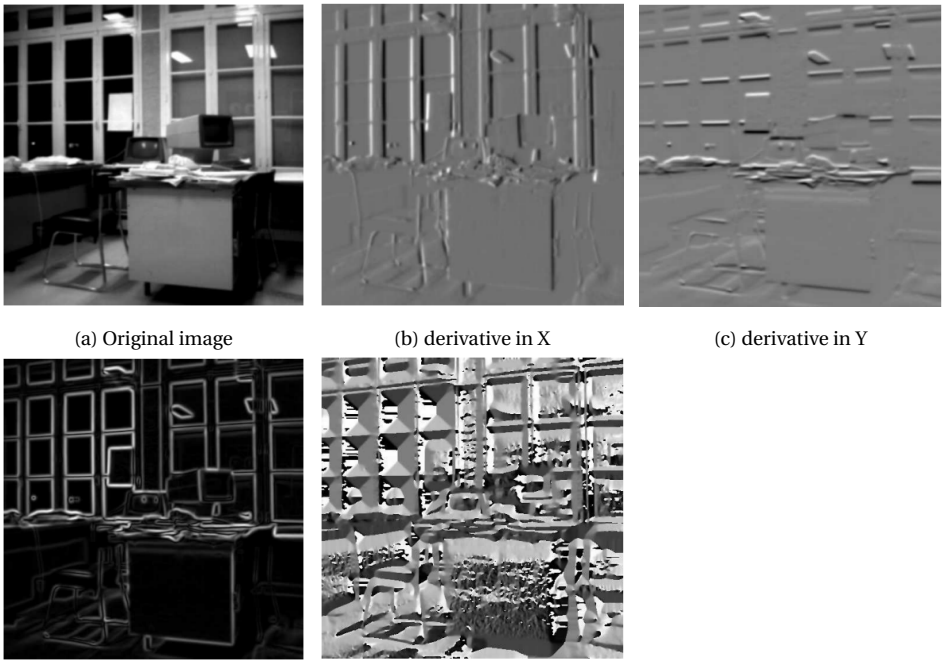


Figure 2.20: Contour with noises Magnier [2011]: a) step edge, b) ramp edge, c) ridge or crest line, d) line

is small (usually between 0.25 and 0.5), it results in better detection. On the other hand, better localization is achieved when the parameter has a higher value (around 2 or 3).

For most of the normal cases' a parameter value of around 1 is recommended. Using the IIR filter makes sense, especially in cases where the processed image is noisy or a large amount of smoothing is required (which leads to large convolution kernel for FIR filter). In these cases, the Deriche detector has considerable advantage over the Canny detector, because it is able to process images in a short constant time independent of the desired amount of smoothing.

The implementation result of the Deriche optimal filtering with gradient extraction is presented in the Fig. 2.21. The NMS and thresholding can follow the same as canny edge detection to obtain thin edges.



(a) Original image (b) derivative in X (c) derivative in Y

(d) Gradient (e) angle of gradient

Figure 2.21: Deriche gradient : (a) initial image (b) derivative in X, (c) derivative in Y, (d) Gradient, (e) angle of gradient ($\alpha = 1.0$).

2.2.3 Recursive implementation

Those filters which can be transformed to Z-transform can be implemented in recursive way. The example of filters which can be implemented recursively are: Shen-Castan, Deriche, Gaussian, etc. The recursive implementation of mean filter takes O(2) operation per pixel. Gaussian can't be implemented in recursive way, because we don't have its solution in Z-transform. However, there are may approximation via other techniques (e.g., Deriche [1992], Ziou [2000], Van Vliet et al. [1998], Getreuer [2013]) which the algorithm complexity is O(3).

2.3 Edge detection based on second order derivatives

Another large class of edge detection methods use contour operators based on 2nd-order differentiation operators. In the case of first derivatives (gradient method) it is to detect directional local maxima. In the case of edge detection methods based on the second

derivatives, it is necessary to detect zero crossings of the directional second order derivative.

Let I_x and I_y be the first derivatives of the image in the x and y directions. Let I_{xx} and I_{yy} denote the second derivatives of the image in the x and y directions and I_{xy} the cross directional second derivative. Let us also denote η the direction of the gradient and ζ the direction of the tangent to the contour. Then the directional second derivative in the direction of the gradient (η) is written as Eq. 2.16:

$$I_{\eta\eta} = \frac{I_x^2 I_{xx} - 2 I_x I_y I_{xy} + I_y^2 I_{yy}}{I_x^2 + I_y^2}. \tag{2.16}$$

and the directional second derivative in the direction of the tangent (ζ) is written as Eq. 2.17:

$$I_{\zeta\zeta} = \frac{I_y^2 I_{xx} - 2 I_x I_y I_{xy} + I_x^2 I_{yy}}{I_x^2 + I_y^2}. \tag{2.17}$$

To perform second-order edge detection, it is necessary to detect the zero crossings of the image $I_{\eta\eta}$. This type of approach poses a problem of complexity as it is necessary to calculate on the one hand the 5 derivatives: I_x, I_y, I_{xx}, I_{yy} and I_{xy}, but also the gradient image of the denominator, etc. In most works found in the bibliography, it is preferred to use the simpler Laplacian operator, which is the sum of the two operators $I_{\zeta\zeta}$ and $I_{\eta\eta}$ as Eq. 2.18.

$$\Delta I = I_{\zeta\zeta} + I_{\eta\eta} = I_{xx} + I_{yy}. \tag{2.18}$$

As each of the operators $I_{\zeta\zeta}$ and $I_{\eta\eta}$ are Euclidean invariants (ζ and η are the directions of a local frame related to the image signal) then: ΔI is also Euclidean invariant. ΔI is therefore an isotropic operator, allowing to detect contours in all orientation. Indeed, $I_{\zeta\zeta}$ being related to the curvature of the isophotes, when the contour is straight or slightly curved, $I_{\zeta\zeta}$ is zero and the Laplacian reduces to the directional second derivative.

On the other hand, the detection of contours can be disturbed at the level of the corners of the objects. $I_{\zeta\zeta}$ represents the curvature of the isophotes multiplied by the norm of the gradient. This term is therefore important. When both the curvature of the isophotes and the gradient are strong. This is the case at the corners of objects. This operator is known as the Kitchen and Rosenfeld [1982] operator, which is one of the first referenced operators used for corner detection.

2.3.1 Discretization of second order operator

The Laplacian is a 2D isotropic measure of the second spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge and line detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise, and hence the two variants will be described together here. The operator normally takes a single gray-level image as input and produces another gray-level image as output.

The Laplacian $L(x, y)$ of an image with pixel intensity values $I(x, y)$ is obtained by the Laplacian Eq. 2.19, and the implementation can be obtained by convolution operation:

$$L(x, y) = \frac{\sigma^2 I}{\sigma x^2} + \frac{\sigma^2 I}{\sigma y^2} \tag{2.19}$$

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. In the discrete case, the operator $\sigma^2 I, \sigma x^2$ is discretized as Eq. 2.20:

$$\sigma_x^2 f = \sigma_x \sigma_x f = f(i+1,j) - 2f(i,j) + f(i-1,j) \tag{2.20}$$

Which the σ_x represents the finite difference operator. This operation can therefore be performed by convolving the input signal with the mask: $[1 \quad -2 \quad 1]$ The Laplacian in two dimensions will therefore be represented by the masks, which are two commonly used discrete approximations to the Laplacian filter. Note, here the Laplacian is defined using a negative peak because this is more common; however, it is equally valid to use the opposite sign convention, as in Eq. 2.21.

$$\Delta = \begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{2.21}$$

There are many variants of Laplacian filters, in general obtained by the difference of two filters of smoothing of different sizes (see chapter 4 for more details). Here is an example of using average filters, Eq. 2.22:

$$\frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{2.22}$$

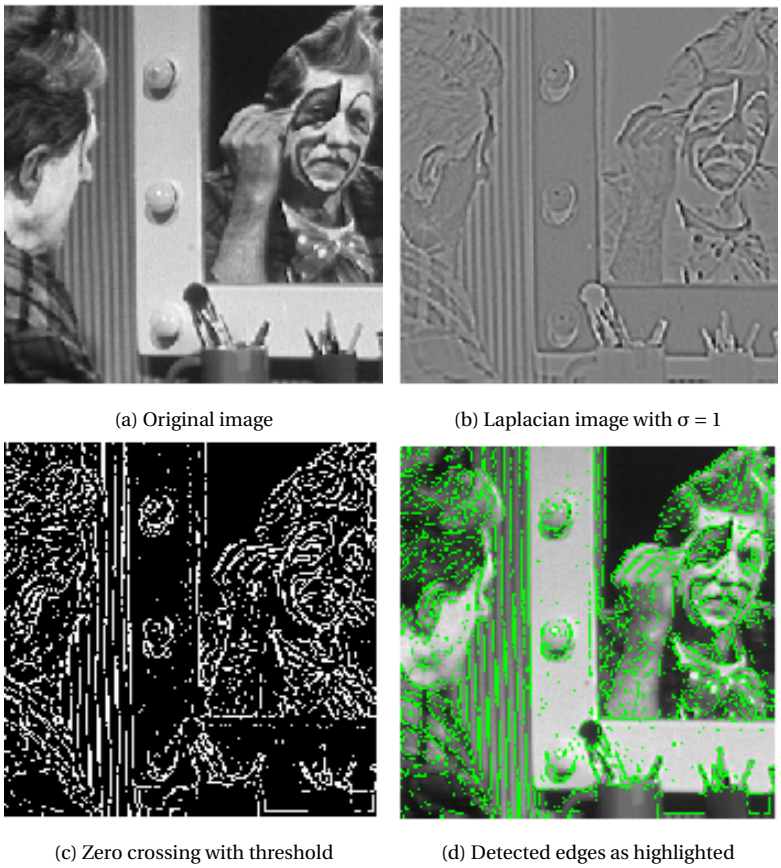
Smoothing is important, because the more the derivative order increases, the more the amplification of the noise increases. Therefore, it will be necessary to smooth the image before implementing the derivative operation. We also have the 3×3 mask Eq. 2.23, which indeed provides an approximation of the Laplacian, which is not normalized.

$$\Delta = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{2.23}$$

This property is known as double low-pass filtering. It is quite possible to use other average filters, such as, 5×5, 7×7, etc. An approximation of the second derivative can be obtained by the difference of two mean filters of different sizes. In practice, this property is not limited to filters only. Any type of smoothing filter can have this approximation, for example Deriche filters, Gaussian filters give fairly robust results while giving the obtained Laplacian a Euclidean invariance.

Marr and Hildreth [1980] which is one of the old technique for detecting edges in digital images. They considered the edge as continuous curves where there are strong and rapid variations in image brightness. This method is simple and operates by convolving the image with the Laplacian of the Gaussian function, or, as a fast approximation, by difference of Gaussian. Then, zero crossings are detected in the filtered result to obtain the edge. The Marr-Hildreth operator suffers from two main limitations (Sponton and Cardelino [2015]). It generates responses that do not correspond to edges, so-called "false edges", and the localization error may be severe at a curved edge as illustrated in the Fig. 2.22. The algorithm is as following:

- Smooth image by Gaussian: $S = G_\sigma * I$.
- Apply Laplacian to find 2nd order derivatives: $\nabla^2 S = \frac{\sigma^2}{\sigma X^2} S + \frac{\sigma^2}{\sigma y^2} S$.



(a) Original image (b) Laplacian image with σ = 1

(c) Zero crossing with threshold (d) Detected edges as highlighted

Figure 2.22: Marr and Hildreth edge detection: a) Original image, b) Laplacian image with σ = 1, c) Zero crossing with threshold, d) Detected edges as highlighted

- Obtain the zero crossing as edge. $\nabla^2 S = \nabla^2(G_\sigma * I) = \nabla^2 G_\sigma * I$
  Or to explicitly express the Laplacian of Gaussian, such as:
  $\nabla^2 G_\sigma = -\frac{1}{\sqrt{2\pi\sigma^3}} \cdot (2 - \frac{x^2+y^2}{\sigma^2}) \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$

The Laplacian method is quite close to that described by Marr-Hildreth, the second derivative and the Gaussian filtering are performed in a single step. It will therefore be necessary to "threshold" the zero crossings. For this, we introduce the gradient of the image by multiplying the zero crossings obtained by the norm of the gradient (for example, the Sobel gradient).

Regarding the Laplacian of Gaussian (LOG) method, as these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.

In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first, and then convolve this hybrid filter with the image to achieve the required result. Doing things this way has two advantages: First, since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations. Second, the LoG kernel can be precalculated in advance, so only one convolution needs to be performed at runtime on the image.

Note that as the Gaussian is made increasingly narrow, the LoG kernel becomes the same as the simple Laplacian kernels. This is because smoothing with a very narrow

Gaussian ($\sigma < 0.5$ pixels) on a discrete grid has no effect. Hence, on a discrete grid, the simple Laplacian can be seen as a limiting case of the LoG for narrow Gaussian.

Laplacian of Gaussian can also be used for blob detection. Observing the LoG filter matrix reveals that it is circularly symmetric. Thus, it can be used for blob and ridge detection, see chapter 4. It is to note that the Laplacian gradient responses are thresholded for blob or spot detection, whereas the Laplacian responses are tested and applied for zero-crossings (change of sign) as application to edge detection.

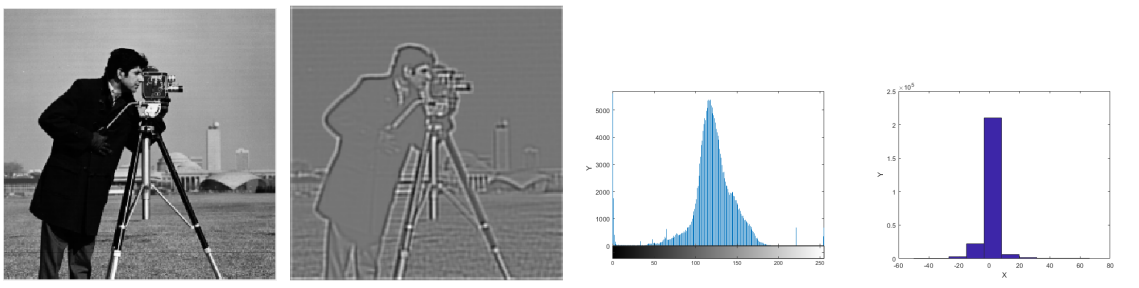
2.3.2 Difference of Gaussian (DoG): Experiments

In image processing, difference of Gaussian (DoG) is also used a feature enhancement algorithm that involves the subtraction of one Gaussian blurred version of an original image from another, less blurred version of the original. In the simple case of grayscale images, the blurred images are obtained by convolving the original grayscale images with Gaussian kernels having differing width (standard deviations).

Blurring an image using a Gaussian kernel suppresses only high-frequency spatial information. Subtracting one image from the other preserves spatial information that lies between the range of frequencies that are preserved in the two blurred images. Thus, the DoG is a spatial band-pass filter that attenuates frequencies in the original grayscale image that are far from the band center.

A simple example of DOG implementation as approximation of second order filtering or Laplacian is demonstrated in the Fig. 2.23. In the below example, the standard deviation 1 and 2 are chosen for both Gaussian filters. Meanwhile, the determination of standard deviation value for both Gaussian filter is application dependent. For wider difference of standard deviation values such as 0 vs 3 or 5 we can extract texture from the images. For lower differences such as values of 1 vs 2 we can extract the edges or lines, through or depending on also the post-processing such as NMS, thresholding, polygonal approximation or morphological processing. In the Fig. 2.23 e) and f) the standard deviation of 2 vs 3 is chosen.

As a feature enhancement algorithm, the difference of Gaussian can be utilized to increase the visibility of edges and other detail present in a digital image. A wide variety of alternative edge sharpening filters operate by enhancing high frequency detail, but because random noise also has a high spatial frequency, many of these sharpening filters tend to enhance noise, which can be an undesirable artifact. The difference of Gaussian algorithm removes high frequency detail that often includes random noise, rendering this approach one of the most suitable for processing images with a high degree of noise. A major drawback to application of the algorithm is an inherent reduction in overall image contrast produced by the operation (Canny [1983]).



(a) Initial image (b) DOG filtered image (c) Histogram of (a) (d) Histogram of (b)

(e) Image, σ = 2 (f) Image, σ = 3 (g) Original image (h) DOG filtered image

Figure 2.23: Example of difference of Gaussian (DOG) result with histogram and different σ values. a) Initial image, b) DOG filtered image, c) Histogram of initial image, d) Histogram of DOG filtered image, e) Image, σ = 2, f) Image, σ = 3, g) Original image, h) Result of DOG

2.4 Contour detection evaluation

All information processing systems, structures, methods and algorithms developed, must be evaluated before being deployed for any application. The evaluation is aimed either for choosing the optimal one among many or to check the performance, accuracy, robustness, reliability, etc., for the related applications. Image processing algorithms and techniques are not an exception and so needs to be evaluated prior to being deployed for any automatic and critical tasks.

As there are different techniques and algorithms in image processing for the detection of various types of contour such as edge, boundary, line, corner, blob, etc., there is need of evaluation mechanism to assess their performance.

In the Fig. 2.24, the classic edge detection techniques such as Robert, Sobel, Kirsch, and Canny evaluation is demonstrated. Depending on the different context and application we should be able to measure the algorithm's robustness against noise sensitivity, localization performance, accuracy level, etc. The usual performance criterion to be considered in assessing the performance of an algorithm includes Wirth et al. [2006] :

- **Robustness**: the capacity of a technique in tolerating various conditions;
- **Accuracy**: the performance of an algorithm with respect to ground truth (reference image), depending on either it is supervised or unsupervised;
- **Efficiency**: the applicability of a technique in time; and space;
- **Sensitivity**: the sensitivity a technique is to small changes in features;
- **Adaptability**: how the technique deals with variability in images;
- **Reliability**: the stability of a technique.

2.4.1 Evaluation metrics

To evaluate the performance of an algorithm for the contour detection, the confusion matrix remains a corner stone. Let Gt be the reference contour map corresponding to ground truth and Dc the detected contour map of an image I. Comparing pixel per pixel Gt and Dc, the first criterion to be assessed is the common presence of edge/non-edge points. A basic evaluation is compounded from statistics resulting from a confusion matrix (Abdulrahman et al. [2017]; Magnier et al. [2018]). As a base to evaluate contour detection methods and define evaluation metrics, 4 quantities need to be set:

- True Positive (TP): correctly detected contour.
- False Positive (FP): incorrectly detected contour.
- True Negative (TN): correctly undetected contour.
- False Negative (FN): incorrectly undetected contour.

Based on these four logical values, many evaluation metrics are defined, such as *Dice, Accuracy Precision, Recall,* F_α, MCC_n and New Figure of Merit notated as $\mathcal{N}$ measure. The output value of all this metric is in the interval [0,1], which value approaching the 1 qualify as good contour detection and value close to 0 determines the weak contour detection. *Precision* metric in Eq. 2.24 is the ratio of correctly predicted positive observations to the total predicted positive observations, and, the *Recall* metric in Eq. 2.25 measure, which measures the model's performance in detecting positive samples found comparing against ground truths. The *Recall* ignores the false positive. The *Recall* measures the ratio between the number of positive samples correctly classified as positive to the total number of positive samples. Precision is also known as positive predictive value, and recall is also known as sensitivity in diagnostic binary classification.

$$Precision = \frac{TP}{TP + FP}. \tag{2.24}$$

$$Recall = \frac{TP}{TP + FN}. \tag{2.25}$$

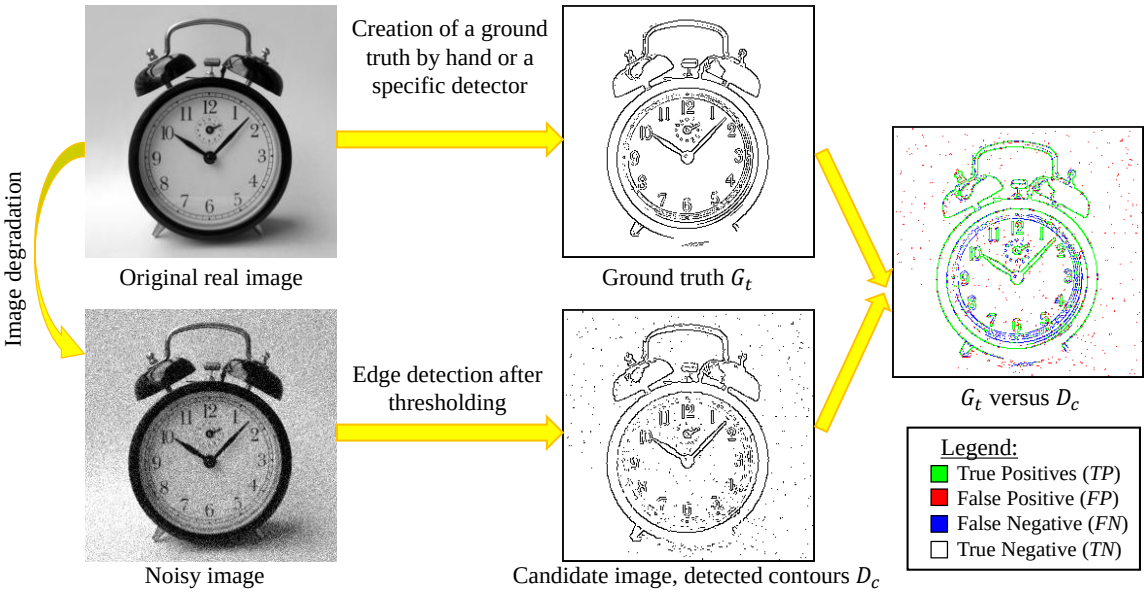


Figure 2.24: Usual process of supervised edge detection evaluation

Accuracy metric as Eq. 2.26 is the most intuitive performance measure, and it is simply a ratio of correctly predicted observation to the total observations.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.26}$$

Dice metric as Eq. 2.27 is also one of the intuitive and basic evaluation metric. Dice is a better evaluation measure comparing the Accuracy, particularly when there is a complete misplacement of all contour values. This is because the Accuracy measure takes into account the true negative value.

$$Dice = \frac{2TP}{2TP + FP + FN} \tag{2.27}$$

$Performance_{measure}$ or P_m metric as Eq. 2.28 is another alternative to Dice, which is less, weighing the true positive value.

$$P_m = \frac{TP}{TP + FP + FN} \tag{2.28}$$

F_α metric as Eq. 2.29 is the weighted average of *Precision* and *Recall*. It is the harmonic mean of the *Precision* and *Recall*. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F_α is usually more useful than accuracy, especially if we have an uneven class distribution. *Accuracy* works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it is better to look at both *Precision* and *Recall*.

$$F_\alpha = \frac{Recall \cdot Precision}{\alpha \cdot Precision + (1-\alpha) \cdot Recall}, \quad \text{with} \quad \alpha \in [0,1]. \tag{2.29}$$

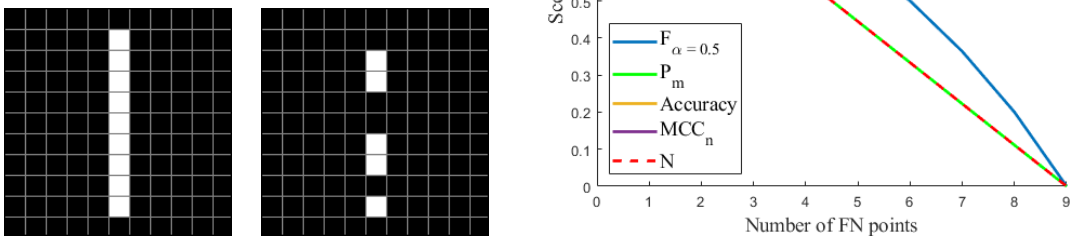
If $\alpha = 0$, then, $F_\alpha = Precision$ whereas if $\alpha = 1$, then $F_\alpha = Recall$. Also, if $\alpha = 0.5$, then $F_\alpha = Dice$; this value was chosen in the experiments presented in chapters 4. Note that if $\alpha > 0.5$, α penalizes more the undetected edges (i.e., FN points); it could be a good indicator because the desirable object becomes unrecognizable, as illustrated in Fig. 2.29.

Matthews [1975] had proposed a new accurate performance measurement method named MCC (Matthews correlation coefficient) for the binary classifications and their confusion matrices. This technique pointed out the issue with the Accuracy and F_α score computed on confusion matrices, that these statistical measures can dangerously show overoptimistic inflated results, especially on imbalanced datasets, see Chicco and Jurman [2020]. MCC is a more reliable statistical rate, which produces a high score only if the prediction obtained good results in all the four confusion matrix categories, proportionally both to the size of positive elements and the size of negative elements in the dataset, as Eq. 2.30:

$$MCC = \frac{TP \cdot TP - FP \cdot FN}{\sqrt{(TP + FN) \cdot (TP + FN) \cdot (TN + FN) \cdot (TN + FP)}} \tag{2.30}$$

MCC has a range of values between -1 and 1; as a consequence, we compute a normalized version $MCC_n = \frac{MCC+1}{2}$ such that the obtained score close to 1 corresponds to a good segmentation, whereas a score close to 0 is tied to a poor classification.

Magnier [2019] introduced a new contour evaluation technique, as Eq. 5.4, which corresponds to a supervised evaluation by quantifying differences between a reference edge map and a candidate, computed by a performance measure/criterion. This measure provides an overall evaluation of the quality of a contour map, by taking into account the



(a) Gt image (11×11) (b) False negative value added (c) Evaluation result

Figure 2.25: Assessment of contour detection with different evaluation methods including: F_α (= $Dice$, when $\alpha = 0.5$),P_m, $Accuracy$, MCC_n, and $\mathcal{N}$ for false negative

amount of false positives, false negatives and also importantly the degrees of shifting or displacement distance.

$$\mathcal{N} = \frac{1}{FP+FN} \cdot \left[\frac{FP}{|D_c|} \cdot \sum_{p \in D_c} \frac{1}{1+K_{FP} \cdot d^2_{Gt}(p)} + \frac{FN}{|Gt|} \cdot \sum_{p \in Gt} \frac{1}{1+K_{FN} \cdot d^2_{Dc}(p)} \right], \tag{2.31}$$

where $(K_{FP}, K_{FN}) \in [0,1]^2$ represent two scale parameters and the coefficient $\frac{1}{FP+FN}$ normalize the $\mathcal{N}$ function. When ($K_{FP} < K_{FN}$), the $\mathcal{N}$ measure penalizes more FN points. Meanwhile, $|\cdot|$ denotes the cardinality of a set, and $d_A(p)$ is the minimal Euclidean distance between a pixel p and a set A.

So, if there are no error, i.e., FP=FN=0, then it corresponds to a perfect score: $\mathcal{N}$=1. Therefore, the measure $\mathcal{N}$ calculates a standardized dissimilarity score; the closer the evaluation score is to 1, the more the edge detection is qualified as suitable.

2.4.2 Evaluation as a function of false negative, false positive and contour displacement

In the Fig. 2.25 the evaluation of contour against ground truth in case of adding false negative values via the F_α (= $Dice$, when $\alpha = 0.5$), P_m, $Accuracy$, MCC_n and$\mathcal{N}$ measure have been computed for false negative points. As we notice here, for the false negative points, the accuracy measure does not show strong sensitivity. It is because of counting the true negative points, which is statistically much more points than the added false negative points. For the F_α, its curve is decreasing due to added false negative points in a bit gradual way. Regarding the MCC_n its curve is decreasing in smooth way, showing smoother sensitivity with the added false negative points, it is because this measure is also taking into account the true negative points in the denominator of the formulation. About the P_m and $\mathcal{N}$ figure of merit, they behave both as straight line in case, with no false positive and displacement. Both this measure are weighing more the true positive points.

In the Fig. 2.26 the evaluation of contour against ground truth in case of adding false positive values of the contour via the F_α (= $Dice$, when $\alpha = 0.5$), P_m $Accuracy$, MCC_n and $\mathcal{N}$ measure have been computed for false positive points. In this figure, we see that the accuracy measure drops and shows linear behavior in case of true positive points.

MCC_n shows different behavior of being more sensitive to false positive points in the beginning, but then balances with accumulation of true negative points.F_α is also dropping down with added false positive points due to this metric is weighing more the true positive points. The P_m and $\mathcal{N}$ likewise presents sensitivity with added false positive points. For both cases of added false negative and added false positive points, these two metrics have the same behavior when there are no contour displacements.

In the Fig. 2.27 evaluation of contour against ground truth in case of contour displacement via the F_α (= $Dice$, when $\alpha = 0.5$), P_m $Accuracy$, MCC_n and $\mathcal{N}$ measure have been computed. As a reference-based edge map, a quality measure requires that a displaced edge should be penalized not just according to false positives and/or false negatives, but also according to the distance from the position where it should be located. Therefore, this figure presents the stability of $\mathcal{N}$ metric in case of contour displacement.

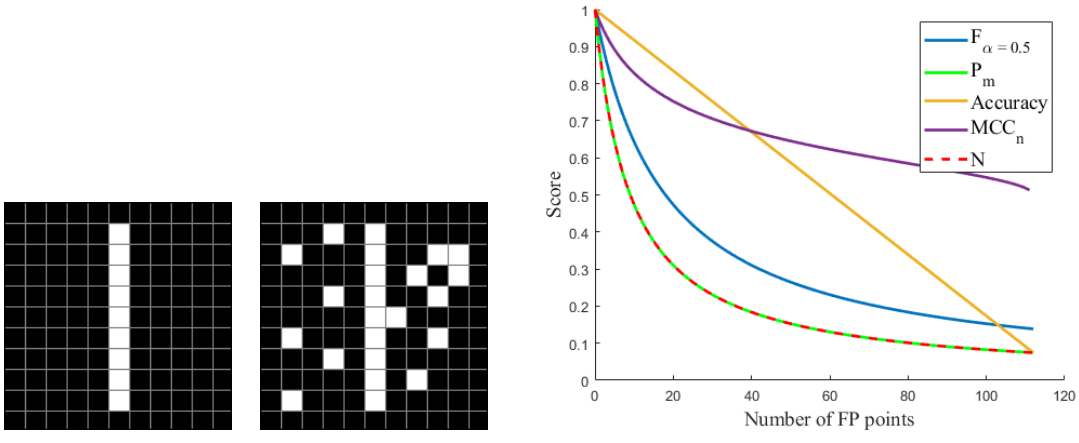
The final note about the evaluation metrics again states that, in general, the choice of which metrics are optimal is made according to the desired final application of the contour or filtering techniques. As an example, it depends on that what values from the confusion matrix (TP, FP, TN, FN) are important for the targeted application.

2.4.3 Evaluation as a function of SNR

Normally, to evaluate the robustness of any segmentation techniques, it will be very intuitive to compute and compare the result as a function of Signal-to-Noise Ratio (SNR). In the SNR Eq. 2.32, the S and R denotes the Signal and Noise respectively: $\mathrm{SNR}(dB) = 10\log_{10}(S/N)$. Considering two images I and J of the same size M×N, the SNR is computed by:

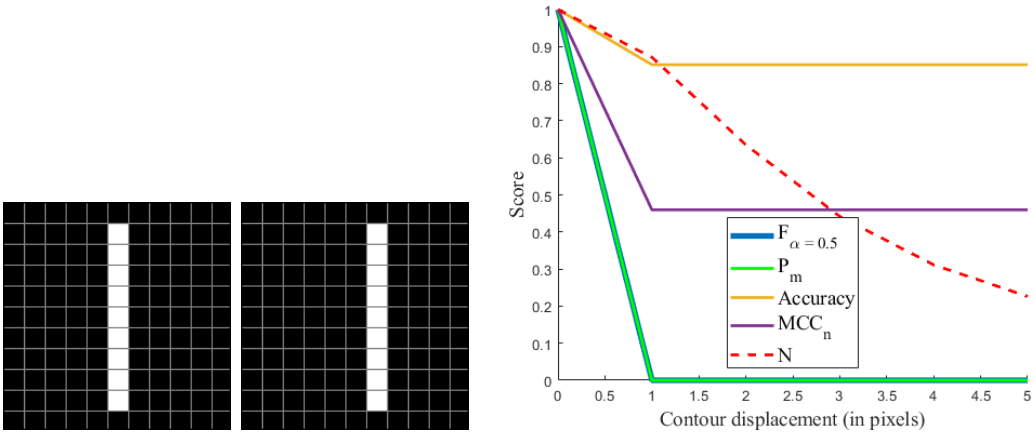
$$\mathrm{SNR}(dB) = 10\log_{10}\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} I^2(i,i)}{\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,i)-J(i,j))^2}. \tag{2.32}$$

By this way, we can evaluate both: the segmentation techniques and the evaluation metrics. Therefore, here we present the two objective and visual evaluation of usual edge detection techniques such as, Roberts, Sobel, Kirsch and Canny as a function of SNR on the noisy images (see Fig. 2.28), with considering two threshold methods (single constant thresholding and Otsu-thresholding).



(a) Gt image (11×11) (b) false positive values added (c) Evaluation result for added true positive

Figure 2.26: Assessment of different evaluation metrics including: F_α (= $Dice$, when $\alpha = 0.5$), P_m, $Accuracy$, MCC_n and $\mathcal{N}$ metric in case of added false positive values



(a) Gt image (11×11) (b) Contour displacement (c) Evaluation result for contour displacement

Figure 2.27: Assessment of different evaluation metrics including: F_α (= $Dice$, when $\alpha = 0.5$), P_m, $Accuracy$, MCC_n and $\mathcal{N}$ metric in case of contour displacement (1 pixel on the right to 5 pixels).

In the Fig. 2.29 the objective valuation of four mentioned edge detection techniques as function of SNR by different state-of-the-art evaluation metrics (Dice, F_α, MCC, $\mathcal{N}$) and two threshold methods (single and Otsu) is demonstrated. In this evaluation for both thresholding methods, the scores are computed on NMS of the gradient norm. The demonstrated result is clear for which edge detection technique (i.e., Canny) and with evaluation metric performs better comparatively.

These obtained scores illustrates that the Canny edge detection is more robust against the noise but not optimum on noise-free images because the scores of Sobel and Kirsch are better between 30 and 20dB. This is especially visible when the detected edges are extracted using a fixed threshold for each noise level. With the Otsu threhold method, the curves tied to the different detectors are more closed. However, this evaluation is not objective since the score of all the edge detection methods are not decreasing; worse still, the obtained scores are better around 15-10dB. It is caused by the Otsu threshold by selecting the two classes; it selects a threshold values avoiding more FN contour points than less noisy images.

Eventually, the $\mathcal{N}$ evaluation metric has shown different graph because it computes the pixel's displacement distance.

Theoretically, to be objectively compared, the ideal edge map for a measure must correspond to a detected contour at which the evaluation obtains the maximum score; this process was chosen in our experiments in the chapters 4, 5 and 6.

Furthermore, in the Fig. 2.28 the visual evaluation of the mentioned edge detection techniques on the images with different level of noises (SNR: 30 dB, 25 dB, 20 dB, 15 dB, 10 dB, 5 dB) with both single thresholding and Otsu thresholding is illustrated. The demonstrated visual result is very intuitive to see which edge detection with certain thresholding method is more robust.

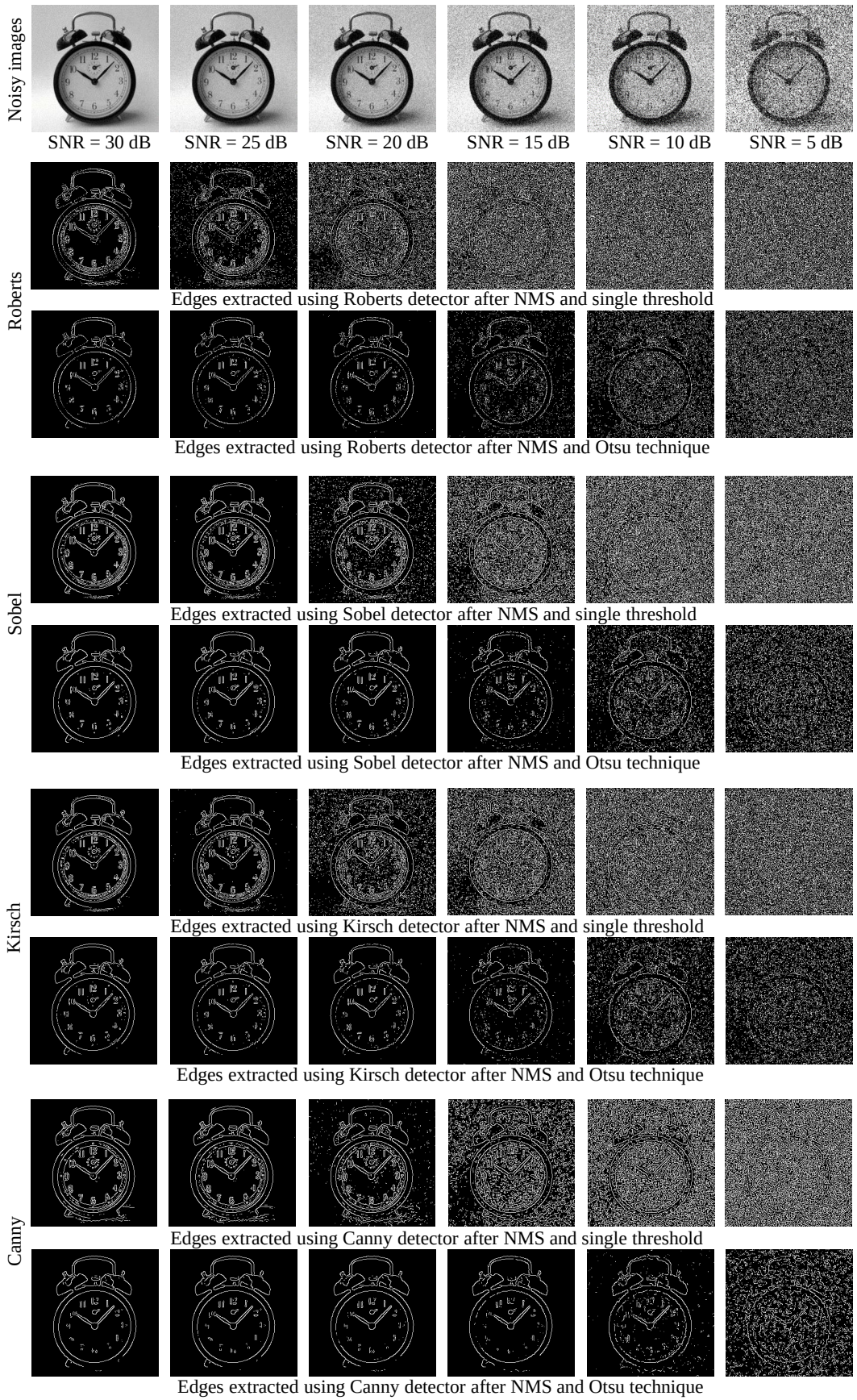
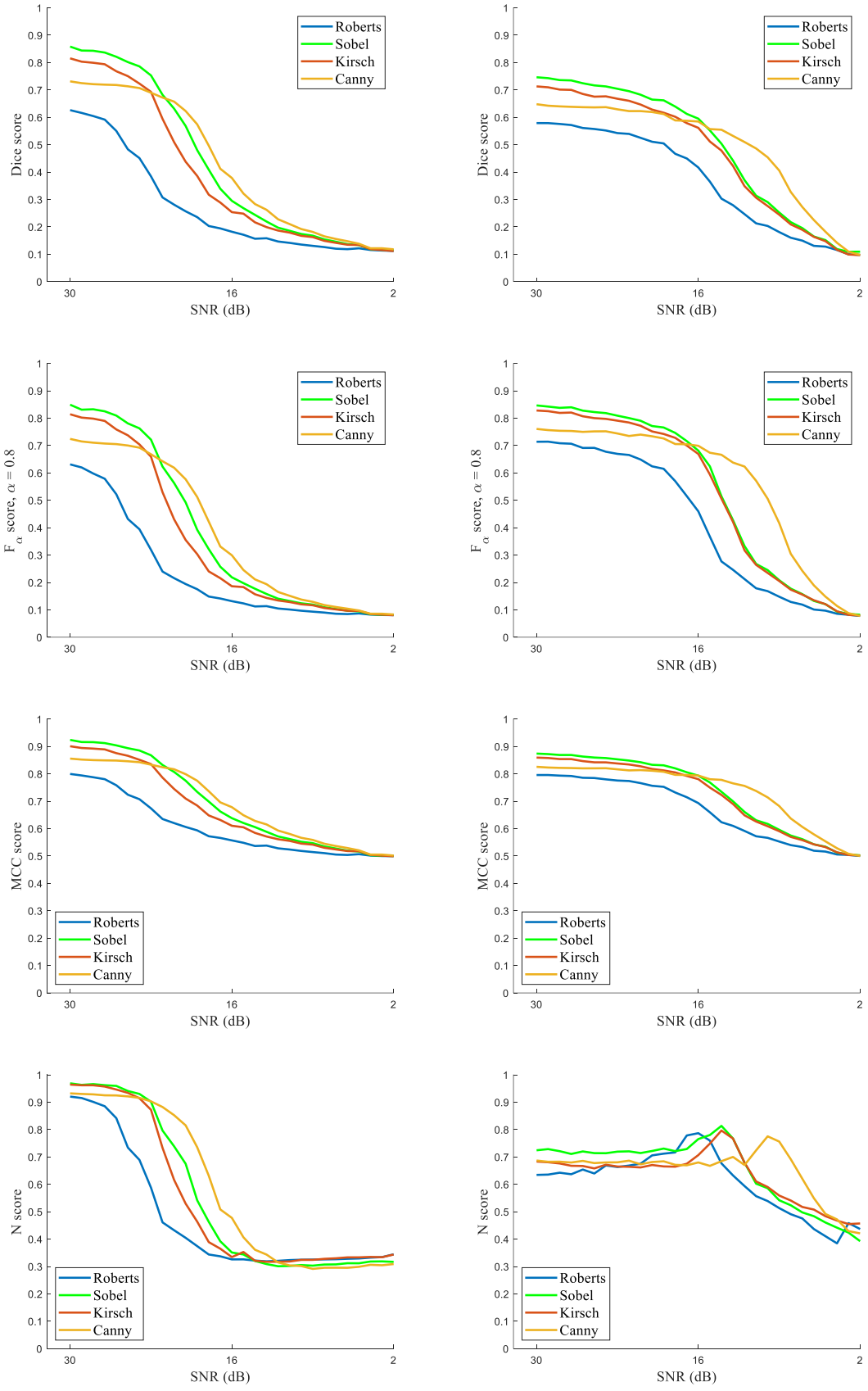


Figure 2.28: Visual results of usual edge detection techniques as function of SNR using two different threshold techniques.



(a) Scores computed applying a single threshold on NMS of the gradient norm

(b) Scores computed applying the threshold calculated by Otsu technique on NMS of the gradient norm

Figure 2.29: Evaluation of usual edge detection techniques presented in Fig. 2.28 as function of SNR by different state-of-the-art evaluation metrics and two threshold techniques.

2.5 Summary

In this chapter containing four sections, basics of image filtering and evaluation methods, especially for edge detection, have been explained. In subsection 2.1, image filtering needs and application, the basic terms and definition such as image contour models, segmentation, thresholding, non-maximum suppression, linear filters (mean and Gaussian), non-linear filter, smoothing and regularization was overviewed..

In subsection 2.2, basic filtering techniques based on first order derivatives (Roberts, Sobel, Prewitt) and also optimal filtering techniques (Canny, Deriche) were overviewed and experimented.

In subsection 2.3, filtering based on second order derivative, such as, Laplacian and its different derived methods and approximation such as Laplacian Of Gaussian (LOG), Difference Of Gaussian (DOG) were studied. Experiments were led with different smoothing and thresholding techniques. Importantly, the regularization property of the Gaussian filter was demonstrated.

In subsection 2.4, the evaluation metrics to assess the performance of different contour detection techniques such as $Dice$, P_m, $Accuracy$, MCC_n, F_α, and $\mathcal{N}$ metrics were introduced. Intuitive evaluation of usual edge detection techniques with these different state-of-the-art evaluation metrics were presented by comparing two thresholding techniques as a function of the SNR level. The $\mathcal{N}$ metrics enables a better supervised edge detection assessment, it is used for the objective evaluation of ridges/valleys explained in the chapters 4, 5, 6.

Chapter 3

Scale Space Analysis in Image Processing

Contents

3.1 Introduction

We know that the objects are composed of different structures at various scales. Indeed, depending on the scale of observation in which objects appear at different scales, computer vision system require scales-space representation of image for different applications, such as, object recognition, automated cartography, 3D reconstruction, etc. likewise, in image processing's scale-space theory, the image structures are processed at different scale via representing image as a one parameter family of the smoothed version of images.

The scale-space representation parametrized by the size of the smoothing kernel is used to suppress the image's fine-scale structures. In the computer vision literature, many

approaches have been introduced for image multiscale representation, including pyramid, wavelet, quad-tree and multi-grid methods. Fig. 3.1 illustrates the pyramid representation, which is the most common representation of scale-space.

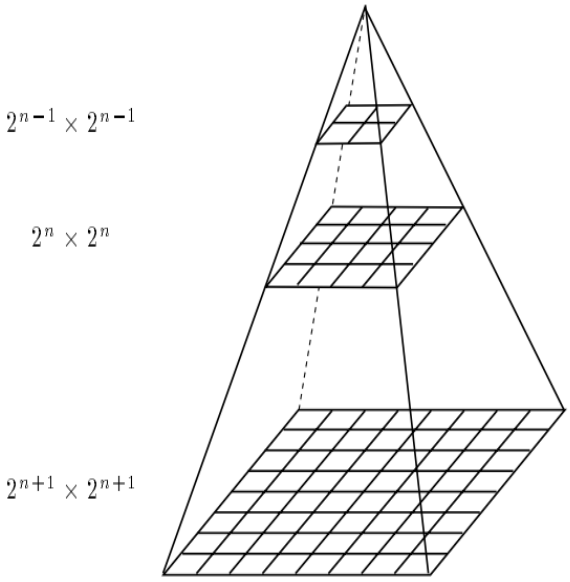


Figure 3.1: A pyramid representation obtained by successively reducing the image size by combined smoothing and sub-sampling. (Adapted from Lindeberg [1994]).

3.2 Multiscale feature detection

In the computation of image structure descriptor based on one dimensional image features, such as edges and ridges, the extracted information strongly relies on the scales at which the image operators are employed. The notion of scale-space edge and ridge is defined as a connected set of points in scale-space at which; the gradient magnitude assumes a local maximum in the gradient direction, and the edge response over scales are maxima in normalized measure of the edge or ridge (Lindeberg [1998]).

The core technique in multiscale image structure detection and extraction is image feature regularization and smoothing in both pre-processing and post-processing stages. Gaussian filter has been extensively used in image processing for various purposes such as, smoothing, regularization and many more. The Zeroth Order Gaussian (ZOG) operator is commonly used for regularization as linear smoothing. The issue with linear smoothing is, that we can not apply selective smoothing for some useful structures, which we do not want to smooth. In order to avoid smoothing these useful structures (edges, ridges, blobs, etc.) which manifests in the object boundaries, so non-linear smoothing as an alternative solution can be implemented (Perona [1992]).

The problem of which level of smoothing to apply is critical for multiscale feature detection and extraction. The trade-off is normally acquired empirically to get the desirable result for both feature detection and localization. The larger amount of smoothing enhance the detection with higher suppression of noise, but in the finer scale. Whilst a smaller amount of smoothing improves the localization with the cost of poor noise suppression and coarser scale (Lindeberg [1994]).

The elemental problem regarding the image feature descriptor is related to what image operator to use in order to be able to flexibly smooth/regularize with parameter and non-linearity. The fundamental concept for handling this issue is addressed in the scale space theory (Witkin [1983]). The theory state that image structure, such as object, exist in certain range of scales, and we estimate the operator's scale in advance for describing them. From this concept it can be interpreted as an assumption that the first phase of visual processing undergoes without bias and thoroughly undecided, then the convolution with Gaussian kernels and their derivatives of different widths is treated as regular class of low-level operators. Therefore, for any image $f : \mathscr{R}^2 x \rightarrow \mathscr{R}$, the representation of scale-space for $\mathrm{L} : \mathscr{R}^2 \times \mathscr{R}_+ \rightarrow \mathscr{R}$ is formulated as Eq. 3.1

$$\mathrm{L}(.;t) = \mathrm{G}(\mathrm{G}.;t) * f \tag{3.1}$$

which G denotes the Gaussian kernel in scale-space as Eq. 3.2

$$\mathrm{G}(x;t) = \frac{1}{2\pi t} \cdot \mathrm{e}^{-\frac{x^2+y^2}{2t}} \tag{3.2}$$

here f the diffusion equation, t is the scale parameter, and the scale-space derivatives are defined as Eq. 3.3

$$\mathrm{L}_{x^\alpha y^\beta}(.;t) = \delta_{x^\alpha y^\beta}(.;t) * f \tag{3.3}$$

which (α, β) are the order of differentiation. These operators can be employed for many visual operations, such as image feature detection, feature and/or object matching and shape cue computation (Lindeberg [1994]). The main problem in multiscale feature detection techniques is the decision of what scale the features should be extracted, or how to determine the desirable scale when the features are extracted at multiple scales. Many techniques have been developed for dealing with this problem. An early technique for handling the blob-like image feature was studied by Lindeberg [1998]. The idea was to select scale levels at a definite normalized measure, presumed as local maxima over scales.

3.3 Gaussian scale-space

Based on the above axioms, the Gaussian convolution is the only low-pass operator that satisfies the scale-space axioms as Eq. 3.4.

$$f(x,y,s) = f(x,y) * \mathrm{G}^s \tag{3.4}$$

Denoting the G as 2D Gaussian kernel, such as Eq. 3.5:

$$\mathrm{G}^s(x,y) = \frac{1}{2\pi s^2} e^{-\frac{x^2+y^2}{2s^2}} \tag{3.5}$$

The function f is a one parameter family of image, which for every scale $s > 0$ there should be an image $f(.,s)$. The Gaussian scale-space function f is the solution of the diffusion equation as Eq. 3.6:

$$f_s = s\nabla^2 f = s(f_{xx} + f_{yy}) \tag{3.6}$$

Koenderink [1984] derived in the linear scale-space with the initial condition of $f(.,0) = f^0$, describing that the small scale change in the function f is proportional to the Laplacian differential structure of f at scale s. Alternatively, the scale-space theory can also be defined as the solution of heat diffusion equation as Eq. 3.7:

$$\sigma_t f = \frac{1}{2}\nabla^2 f \tag{3.7}$$

with the initial condition of $f(x, y; 0) = f(x, y)$. Furthermore, to choose locally an appropriate scale automatically based on local maxima (minima) over scale-normalized derivative is via the scale parameter such as λ over a range of scale, such as $\gamma \in [0, 1]$. The Fig. 3.2 visually explains the main idea of scale-space representation. This figure shows a one-dimensional signal that has been smoothed by convolution with Gaussian kernels of increasing width.

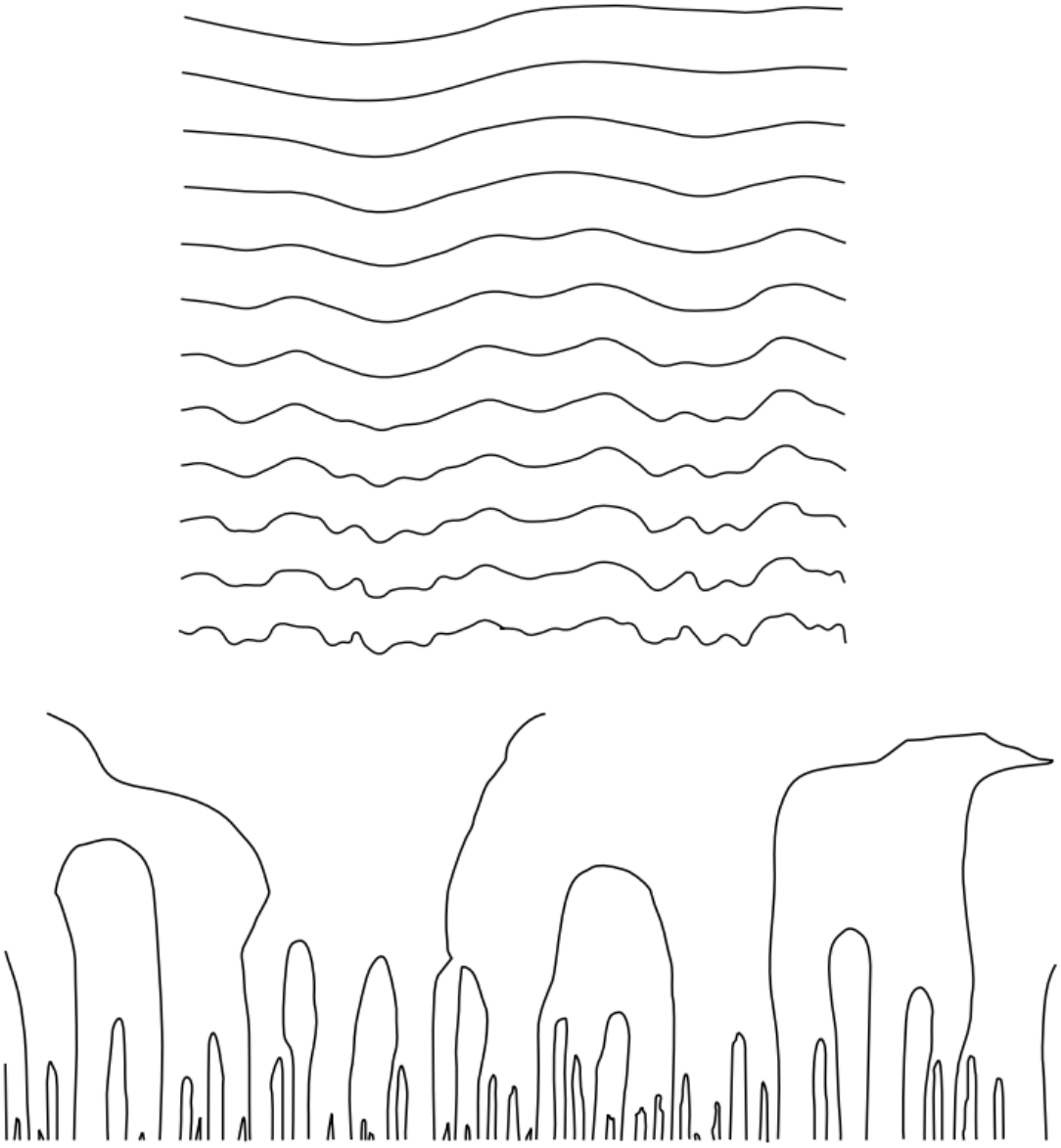


Figure 3.2: The scale-space representation of a signal via generating a family of derived signals in which the fine-scale information is successively suppressed. (Adapted from Witkin [1983]).

3.3.1 Perona and Malik

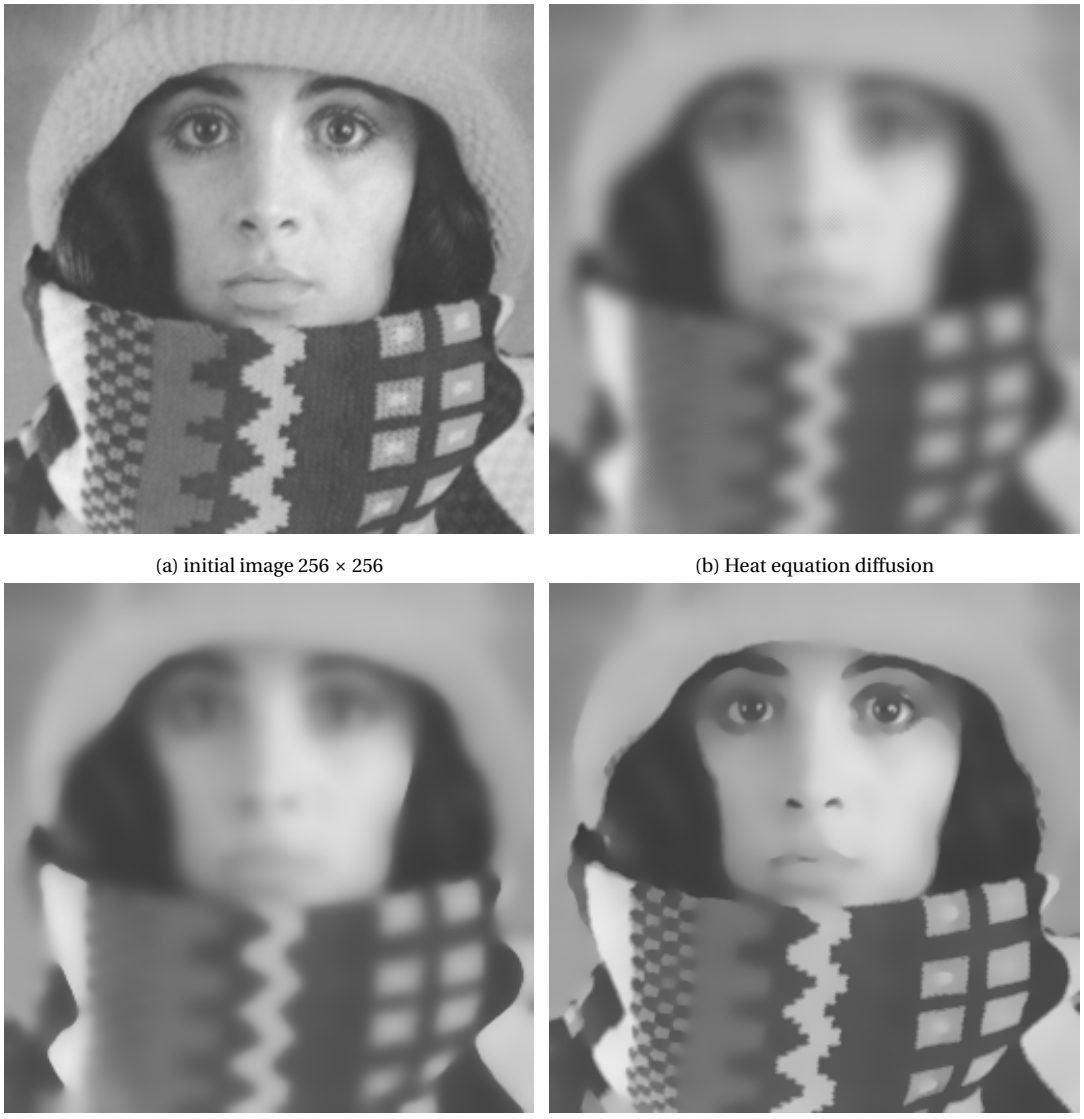
Due to the low accuracy of contour localization in isotropic approach of scale-space, Perona and Malik [1990] proposed the anisotropic diffusion approach for better localization of contour, here edges. In fact, this algorithm can be applied in a non-linear way, such that, we can selectively apply smoothing for intra-region as preferred to inter-region,

while keeping the conventional axioms of scale-space theory in not creating new local maxima. Perona and Malik further formalized the scale-space axioms with the notion of "Causality"," Immediate Localization", and " Piece wise Smoothing", which the formulation consist of Eq. 3.8:

$$\mathrm{I}_t = div(c(x,y,t)\nabla \mathrm{I}) = c(x,y,t)\Delta \mathrm{I} + \nabla c.\nabla \mathrm{I} \tag{3.8}$$

$$c(\|\nabla \mathrm{I}\|) = e^{(-\frac{\|\nabla \mathrm{I}\|}{k})^2}, \quad k \in \mathbb{R} \tag{3.9}$$

Which the "div" indicates the divergence operator and the ∇ and Δ respectively the gradient and Laplacian operators. It is to note that the Eq. 3.9 is the function of gradient to control the diffusion for intra and inter region involving the parameter "k" for controlling the diffusion, aimed for not blurring the edge, as when the $c(x,y,t)$ stay constant, the formula will be the heat diffusion equation as $\mathrm{I}_t = c\Delta \mathrm{I}$. The Perona and Malik [1990] implementation is demonstrated in Fig. 3.3 of 20 iteration diffusion, different values of "k" parameter, and kernel size of (7×7) compared with heat equation (isotropic diffusion). In the next subsection, we are going to present the modern theory underlying this idea of scale-space, generalizing Gaussian scale-space to multiscale analysis.



(a) initial image 256 × 256

(b) Heat equation diffusion

(c) Of 20 iteration, k=0.07

(d) Of 20 iteration, k=0.7

Figure 3.3: Comparison of heat diffusion on a real image with Perona and Malik [1990] method.

3.4 Axioms and scale-space generalization

A multiscale analysis consists in generating a sequence of images from an initial image, in which each image of the sequence is an image increasingly blurred from the original one as the sequence goes on. As described in the previous section, pyramid is one way to obtain a such sequence.

It is defined here an operator $(T_t)_{t\geq 0}$ applied to an original image $I(0, x, y) = I_0(x, y)$ leading to the image $I_t(x, y)$ of the sequence such that $I_t(x, y) = T_t(I_0)(x, y)$ (and t a parameter, $t \in \Re^+$). Then :

$$\begin{cases} I(0, x, y) &= I_0(x, y) \\ I(t, x, y) &= T_t(I_0)(x, y) \end{cases} \tag{3.10}$$

In the Eq. 3.10, to ensure the stability of this analysis the operator $(T_t)_{t\geq 0}$ must be causal, and to a mathematical point of view, this operator must verify some architectural axioms such as recursivity, comparison (local comparison) and regularity.

3.4.1 Recursivity and pyramidal structure axiom

Suppose s and t two different "instants" on the sequence ($s, t \geq 0$) then the Eq. 3.11:

$$T_s \circ T_t = T_{s+t} \tag{3.11}$$

Which states that the image at “time” $s + t$ can be obtained considering the image at time s and a transition operator from s to t T_t depending only on t. For example, considering the Gaussian scale-space, it is obvious that the convolution of two Gaussian remains a Gaussian, so then the Gaussian scale-space follows this axiom.

3.4.2 Local comparison axiom

Local comparison axiom states the local conservation of intensities. Roughly speaking, if one image is brighter than another one, then the order must be conserved along the analysis. Let us consider two images: I_t and J_t, if exists a circular neighborhood $N(\mathbf{m}_0, r)$ centered at pixel $\mathbf{m}_0$ having a radius r, such that $I_t(\mathbf{m})$ is smaller than $J_t(\mathbf{m})$ over $N(\mathbf{m}_0, r)$ then for a small h ($h \in \Re^+$) :
$(T_{t+h,t}(I_t))(\mathbf{m})$ is smaller than $(T_{t+h,t}(J_t))(\mathbf{m})$

3.4.3 Regularity axiom

This axiom requires a regularity along the evolution. If an image I is quadratic in the neighborhood of a pixel $\mathbf{m}_0$ as Eq. 3.12:

$$I(\mathbf{m} = a + p^T . (\mathbf{m} - \mathbf{m}_0) + \frac{1}{2} (\mathbf{m} - \mathbf{m}_0)^T . \mathbf{A} . (\mathbf{m} - \mathbf{m}_0) \tag{3.12}$$

with :

a	:	a scalar value
$\mathbf{p}$	:	a two dimensional vector
$\mathbf{A}$	:	a 2×2 matrix

The regularity is then expressed by the existence of a function F as Eq. 3.13:

$$F(t, \mathbf{m}, a, \mathbf{p}, \mathbf{A}) \quad : \quad [0, +\infty[\times \Re^2 \times \Re \times \Re^2 \times S(2) \longrightarrow \Re \tag{3.13}$$

with S(2) : the space of 2×2 matrices having real coefficients , and

$$\lim_{h \to 0} \frac{T_{t+h,t}(I_t)(\mathbf{m}) - I_t(\mathbf{m})}{h} = F\left(t, \mathbf{m}, a, \mathbf{p}, \mathbf{A}\right) \tag{3.14}$$

3.4.4 Fundamental theorem

If a multiscale analysis follows the three axioms described hereinabove then

$$\begin{array}{rcl} I(0, x, y) & = & I_0(x, y) \\ I(t, x, y) & = & (T_t(I_0))(x, y) \end{array} \tag{3.15}$$

is a viscosity solution of the Eq. 3.16 PDE ;

$$\frac{\partial I}{\partial t}(x, y) = F(H(I), \nabla I, I, x, y, t) \tag{3.16}$$

with :

$I(0, x, y)$	:	is the original image
F	:	a non decreasing function
∇I	:	represents the gradient of image I
H(I)	:	represents the hessian matrix of image I

3.5 Multiscale analyses classification

Having this important general result, we can now examine some properties of analyses.

3.5.1 Linearity and Euclidean invariance

$$T_{t,t+h}(aI_t + bJ_t) = aT_{t,t+h}(I_t) + bT_{t,t+h}(J_t) \tag{3.17}$$

where :

a, b	:	are two scalars
I, J	:	are two images

in other words, the analysis of the ponderated sum of two images is the ponderated sum of two analyses. In this case, the PDE simplifies to the heat equation as Eq. 3.18

$$\begin{cases} I(0, x, y) = I_0(x, y) \\ \frac{\partial I}{\partial t}(t, x, y) = \Delta I \end{cases} \tag{3.18}$$

The general solution of this PDE is the convolution of the original image with a Gaussian of parameter t Eq. 3.19:

$$g(t, x, y) = \frac{1}{4\Pi t} e^{-\frac{x^2+y^2}{4t}} \tag{3.19}$$

and there is a direct relation between the time of diffusion and the standard deviation σ of the Gaussian Eq. 3.20:

$$\sigma = \sqrt{2t} \tag{3.20}$$

3.5.2 Morphological and Euclidean invariance

Morphological invariance is insured when any strictly monotonous contrast function is applied to an image. Contrast is change, but shapes of isophotes remain unchanged. Then a morphological-scale space must verify the property of Eq. 3.21:

$$T_{t,t+h}(g \circ I_t) = g \circ T_{t,t+h}(I_t) \tag{3.21}$$

In other words, the multiscale analysis must permute with the contrast function. In this case, the solution obtained can be written as Eq. 3.22:

$$\begin{cases} I(0,x,y) & = & I_0(x,y) \\ \frac{\partial I}{\partial t}(t,x,y) & = & t|\nabla I|\mathrm{div}\left(\frac{\nabla I}{|\nabla I|}\right) \end{cases} \tag{3.22}$$

After timescale change ($t' = \frac{1}{2}t^2$) we obtain Eq. 3.23:

$$\begin{cases} I(0,x,y) & = & I_0(x,y) \\ \frac{\partial I}{\partial t}(t,x,y) & = & |\nabla I|\mathrm{div}\left(\frac{\nabla I}{|\nabla I|}\right) \end{cases} \tag{3.23}$$

which simplifies to Eq. 3.24:

$$\begin{cases} I(0,x,y) & = & I_0(x,y) \\ \frac{\partial I}{\partial t}(t,x,y) & = & I_{\xi\xi}(t,x,y) \end{cases} \tag{3.24}$$

Where ξ represent the tangent to isophotes. This multiscale analysis is known as Mean Curvature Motion (MCM).

3.6 Causal filtering and segmentation

Several works that are presented in this document are based on anisotropic causal filtering. Generally, in image segmentation based on differential operators, derivative filtering are applied to an image, and the derivative obtained are combined into a differential operator. For example, for edge detection with first order operator, the image gradient is obtained using the X and Y derivatives.

As an image is a 2D signal, while the image is derived along X the image is also smoothed along the Y direction and inversely to obtain the X and Y estimates of the derivatives to be combined into an operator. In the causal filtering case, we still have 2D filters, but the smoothing filter is replaced by a causal one.

We will now be considering the 2D filters as a combination of two 1D filters oriented at 90 degrees each, having also possibly different standard-deviations (if Gaussian filter), such that one is a causal smoothing filter and the other one a derivative filter. We present here the X derivative filter as Eq. 3.25:

$$F(x,y) = C_1 x H(y) e^{-\frac{x^2}{2\sigma_\eta^2} - \frac{y^2}{2\sigma_\xi^2}} \tag{3.25}$$

where :

σ_ξ	:	is the standardt-deviation of the causal Gaussian smoothing filter
σ_η	:	is the standardt-deviation of the Gaussian derivation filter
$H(y)$	:	a Heaviside function along the Y axis
C_1	:	is a normalization coefficient

Generally σ_ξ is large (between 5 and 10) while σ_η is much smaller (between .7 and 2). As

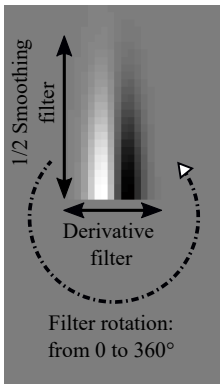


Figure 3.4: Causal anisotropic Gaussian Filters with derivation of first order.

these type of filter is strongly anisotropic, it is necessary to use many rotated filters in order to scan the image information around a pixel, for example by steps of $\Delta\theta = 10$ degrees (Fig. 3.4). But for an implementation point of view, rotating such filter leads to important computational complexity since rotated filters does not remain separable, and then we have to consider large 2D convolutions. Then the best way to proceed it to rotate the image, apply a separable and recursive filter having low computational complexity and rotate back the filtered image. The response to a given filter is then obtained by Eq. 3.26:

$$Q(x,y,\theta) \;=\; I_\theta * C_1 x H(y) e^{-\frac{x^2}{2\sigma_\eta^2} - \frac{y^2}{2\sigma_\xi^2}} \tag{3.26}$$

where :

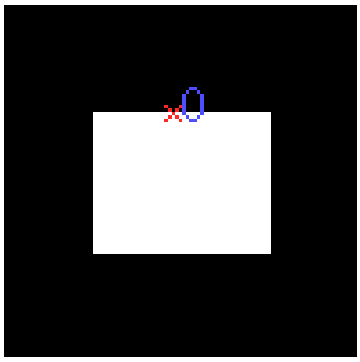
I_θ : is the rotated image at angle θ

3.6.1 Edge detection using causal filtering

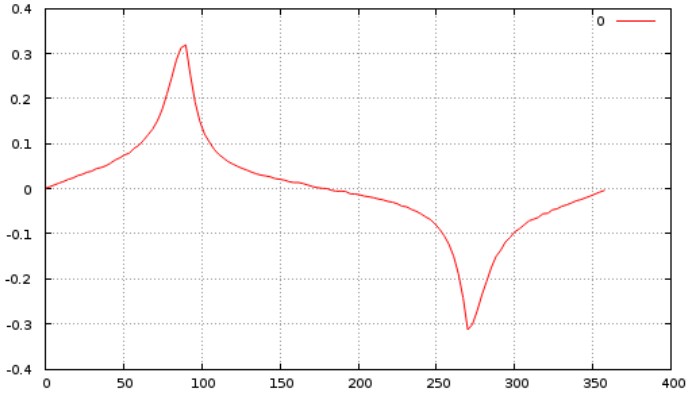
The Fig. 3.5 presents an example of the responses to rotated filters at an edge point (horizontal edge). We obtain two main directions (90° and 270°) with a strong response separated by 180 degrés. The Fig. 3.6 presents an example of the responses to rotated filters at corner points. For each corner point, we obtain two main directions separated by 90 degrees for which, the image derivative is maximal or minimal. these two directions θ_1

Type of operator	Fixed operator	Oriented Filters	Causal filter
Example of filter			
Gradient magnitude	$\lvert\nabla I\rvert = \sqrt{I_x^2 + I_y^2}$	$\lvert\nabla I\rvert = \max_{\theta\in[0,180°[} \lvert I_\theta\rvert$	$\lvert\nabla I\rvert = \max_{\theta\in[0,360°[} I_\theta - \min_{\theta\in[0,360°[} I_\theta$
Gradient direction	$\eta = \arctan\left(\frac{I_y}{I_x}\right)$	$\eta = \arg\max_{\theta\in[0,180°[} \lvert I_\theta\rvert + 90°$	$\eta = \left(\arg\max_{\theta\in[0,360°[} I_\theta + \arg\min_{\theta\in[0,360°[} I_\theta\right)/2$

Table 3.1: Gradient magnitude and orientation computation for a scalar image I where I_θ represents the image derivative at the θ orientation (in degrees).



(a) Initial image (simple square) (point 0)



(b) Obtained responses (two peaks separated by 180 degrees)

Figure 3.5: Filter responses at an edge point. a) Initial image (simple square) (point 0). b) Obtained responses (two peaks separated by 180 degrees).

and θ_2 are given by Eq. 3.27:

$$\theta_1 = \underset{\theta\in[0,360[}{\arg\max}(Q(x,y,\theta))$$
$$\theta_2 = \underset{\theta\in[0,360[}{\arg\min}(Q(x,y,\theta)) \tag{3.27}$$

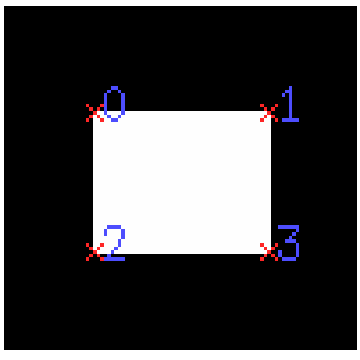
Then a causal anisotropic gradient can be defined by Eq. 3.28:

$$\|\nabla I\| = \mathcal{Q}(x,y,\theta_1) - \mathcal{Q}(x,y,\theta_2) \tag{3.28}$$

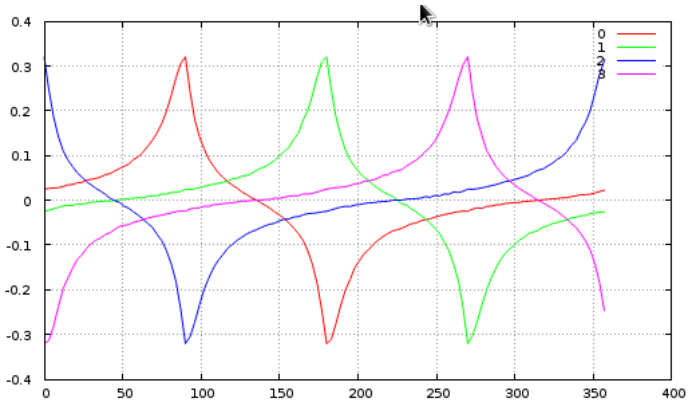
Finally, we present an edge detection result obtained using causal anisotropic filters at sub-pixel precision. The figure (Fig. 3.7) present an edge detection result obtained on the well known image "Lena". filter parameters are : $\sigma_\xi = 10$ $\sigma_\eta = 1$ and $\Delta\theta = 10°$.

3.6.2 Causal filtering and Scale-space

We check here causal filtering against scale-space axioms in order to understand mathematical properties of the causal smoothing filter used in combination with derivative fil-



(a) Initial image (corner have been selected)



(b) Obtained responses at each corner point (two peaks separated by 90 degrees).)

Figure 3.6: Filter responses at corner points. a) Initial image (corner have been selected). b) Obtained responses at each corner point (two peaks separated by 90 degrees).

(a) Initial image (b) Edge detection at precision $\frac{1}{2}$ pixel

Figure 3.7: Causal anisotropic edge detection. a) Initial image. b) edge detection at precision $\frac{1}{2}$ pixel.

ters, for example in edge and line detection. For this study, we will only have to consider the 1D causal filter without normalization coefficient as in Eq. 3.29:

$$\mathrm{F}(y) \;=\; \mathrm{H}(y)\cdot e^{-\frac{y^2}{2\sigma^2}} \tag{3.29}$$

As described earlier, we have to test the 3 axioms : pyramidal structure, comparison, and regularity.

3.6.2.1 Pyramidal structure

Let's consider the Fourier Transform of this filter as Eq. 3.30:

$$\tilde{\mathscr{F}}(v) \;=\; \int_{-\infty}^{\infty} \mathrm{H}(y)\, e^{-\frac{y^2}{2\sigma^2}}\, e^{-2\Pi v y} dy \;=\; \frac{1}{2}\left[\delta(v) - \frac{i}{\Pi}\mathrm{V}p\left(\frac{1}{v}\right)\right] * \sqrt{2\Pi\sigma^2}\, e^{-2\Pi^2 v^2} \tag{3.30}$$

then :

$$\tilde{\mathscr{F}}(v) \;=\; \frac{1}{2}\sqrt{2\Pi\sigma^2}\, e^{-2\Pi^2 v^2} \;-\; \frac{1}{2}\frac{i}{\Pi}\mathrm{V}p\left(\frac{1}{v}\right) * \sqrt{2\Pi\sigma^2}\, e^{-2\Pi^2 v^2} \tag{3.31}$$

This expression contains two terms :

- The first term is a Gaussian function in the frequency domain.
  This term is the Fourier transform of the Gaussian function as Eq. 3.32:

$$e^{-\frac{y^2}{2\sigma^2}} \tag{3.32}$$

  This term shows clearly that the convolution of two Gaussian functions remains a Gaussian function : in the frequency domain the convolution operation is converted into a multiplication and the multiplication of two Gaussian functions is a Gaussian function.

- The second term is more complex, it contains a convolution with the distribution :

$$\mathrm{V}p\left(\frac{1}{v}\right)$$

  Then obtaining a transition convolution operator to obtain a causal Gaussian function for example at σ_2 from causal Gaussian function at σ_1 such that :

$$\mathrm{I}_{sigma_2} = \mathrm{T}_{\sigma_2-\sigma_1}(\mathrm{F}_{\sigma_1}(y) * \mathrm{I}_0(y)$$

  is not possible in a linear way, since it's necessary at least to find an inverse of convolution of a Gaussian.

It's then obvious that there is no pyramidal structure using these type of filters. But In any way, if we need to compute an increasing causal scale at some orientation for an input image, it is possible to proceed to multiple filtering starting each time from the original image, since the implementation of the Gaussian filter is recursive and that the computational cost is small.

3.6.2.2 Comparison axiom

The demonstration of this axiom is obvious : the support of a causal Gaussian function filter is half the support of a Gaussian function, then the local comparison axiom holds.

3.6.2.3 Regularity axiom

Here also this axiom holds : if the input image is a quadratic function along the Y axis then its limited development around a pixel contains at most terms of order 2 and convolving such function with any operator (along Y axis) leads to a result containing a most a term of order 2. Then the regularization is unsure due to the convolution properties of convolving a signal with a smoothing operator (half Gaussian).

3.7 Summary

In this chapter, the basic theory of scale space and its application in the domain of image processing has been presented. The pioneer work for multiscale feature detection using scale space theory such as the work of Perona and Malik, Lindeberg, and Witkin was reviewed. The spacial focus was made on the Gaussian scale-space solution using Pyramidal solution of Gaussian smoothing. Meanwhile, the scale space axioms and generalization were detailed. Finally, the causal anisotropic filtering for edge and corner detection were introduced in pixel level and multiscale formulation.

Chapter 4

Objective Evaluation of Ridge/Valley Detection Techniques

Contents

4.1 Introduction

Roughly ridges (resp. valley) as one of the many images' salient features, are a set of curves whose points are local maxima (resp. minima) in an image, as shown in Fig. 4.1. The precise detection, localization and extraction of this salient features along with their accurate characterization of its geometric structure are important image processing tasks, related to its wide range of application. Exhaustive researches have been accomplished on this significant image features. Ridges have shown to be the most eminent and useful structure amongst for image analysis and various related applications.

Digital images comprise varying types of salient features, such as, edges, blobs, corners, textures, whereas ridges (also called crest lines or roof edges) represent a special type of contours, as shown in Fig. 4.2. Classical edge detection techniques are optimized to extract step or ramp edges Canny [1986]; nonetheless, they fail to detect ridges or crest

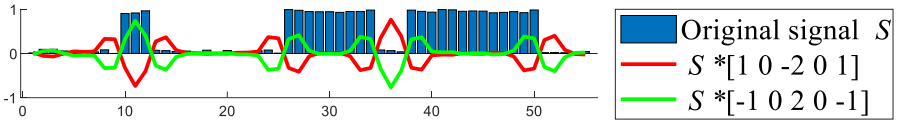


Figure 4.1: Ridge and Valley highlighted on a discrete 1D signal using Laplacian

lines in images. A step/ramp edge extraction functions return two edges at both sides of the crest line because narrow ridges or valleys on the image surface are composed of two locally parallel step or ramp edges. Roof edges are defined as thin nets inside the image; describing roads or rivers in satellite images, blood vessels in medical images or plant roots. Therefore, finding these dense and thin structures is an important task in image processing.

Concurrently, there are many inescapable challenges in image processing task, such as, noise, artifacts, etc., that necessitate to be researched concerning the specific application. Generally, the acceptable ridge/valley detection involves cumbersome and manual tuning in order to overcome the issues for specific application. In consequence, to contribute the research process in the domain of ridges detection and extraction techniques, an extensive evaluation of the different state-of-the-art filtering techniques and approaches in the scope of its most useful application, is crucially necessary. This work is aimed for objective and extensive analysis of state-of-the-art filtering techniques for ridge(resp. valley) detection and extraction.

Regarding a curve in a gray-level image I, ridges correspond to directional maxima, valleys to direction minima resp. (Haralick [1983]), as illustrated in Fig. 4.2. In the Cartesian space, considering the image surface I_s, thus $\vec{I_s}$ defines all pixel coordinates: $\vec{I_s}(x,y) = (x \quad y \quad I(x,y))^T$. Let $I_T(x,y) = \{\vec{I_{s_x}}(x,y), \vec{I_{s_y}}(x,y)\}$ be the tangent plan of the surface $\vec{I_s}(x,y)$ in all points where:

$$\vec{I_{s_x}}(x,y) = \frac{\partial \vec{I_s}(x,y)}{\partial x} = (1 \quad 0 \quad I_x(x,y))^T = (1 \quad 0 \quad \frac{\partial I(x,y)}{\partial x})^T$$
$$\vec{I_{s_y}}(x,y) = \frac{\partial \vec{I_s}(x,y)}{\partial y} = (0 \quad 1 \quad I_y(x,y))^T = (0 \quad 1 \quad \frac{\partial I(x,y)}{\partial y})^T,$$

with I_x and I_y the partial derivatives of I respectively, along the x and the y axis. Ridges and valleys are given by the points where the values of $\vec{I_s}(x,y)$ are maxima (resp. minima) in the orthogonal direction of the curve at (x,y). Consequently, to detect and extract ridges and valleys in a signal, the Laplacian is used as a high pass filter to enhance its high frequency component. As a result, to extract peaks in a one-dimensional (1D) signal $S(t)$, the (continuous) Laplacian operator Δ is simply the second derivative of S: $\Delta S(t) = \frac{\partial^2 S}{\partial t^2}(t)$. The discrete Laplacian is an approximation to the continuous Laplacian. The common approximation to the second derivative of a discrete signal $S(t)$ is:

$$\Delta S(t) = \frac{2S(t) - S(t-h) - S(t+h)}{h^2}, \text{ with } h \in \{1,2,3...\}, \tag{4.1}$$

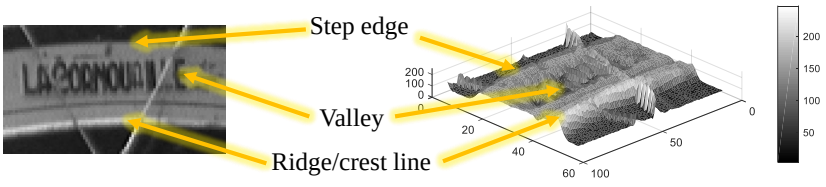
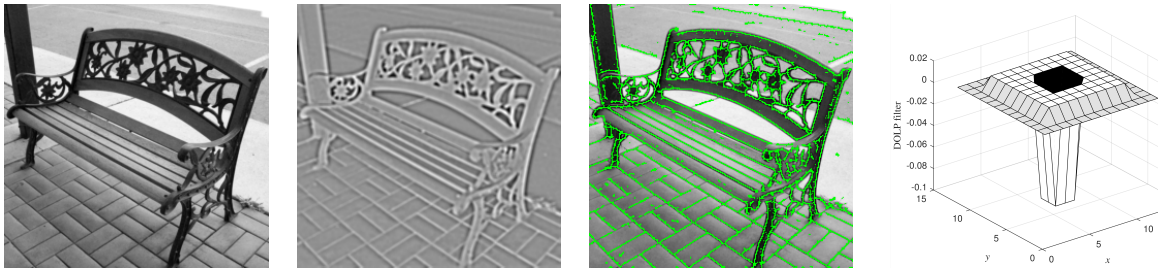


Figure 4.2: Illustration of features in images by elevation of the image intensity.



(a) Image 256×256 containing valleys

(b) DOLP transform: mean filters of size 3×3 and 9×9

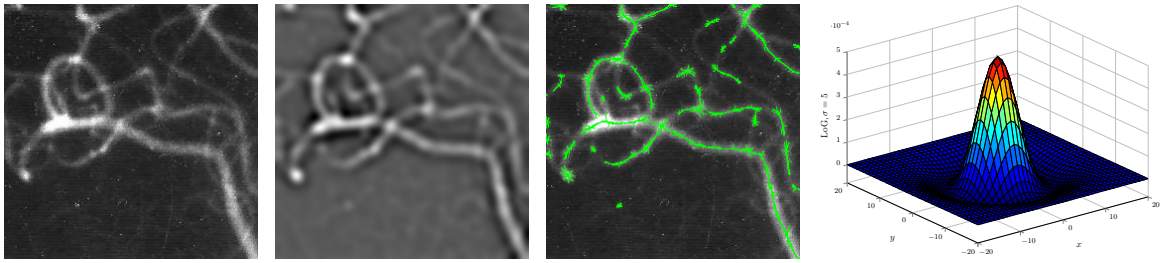
(c) Valley extracted on (a), after local maximum extraction

(d) DOLP filter: with filter 1 of size 3×3 and filter 2 of size 9×9

Figure 4.3: Valley detection using a DOLP (Difference Of Low Pass) filter.

hence, the use of a discrete Laplacian for 1D signals is the convolution of a signal with the vector ±[1 0 -2 0 1]. Regarding the 2-dimensional (2D) signals such as digital images, the Laplacian operator becomes $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, tied to the horizontal and the vertical second derivatives (called x and y directions resp.). This 2D operator allows approximating the second derivative of the image, in order to detect edges by zero-crossing (Marr and Hildreth [1980]) and highlight ridges or valleys. Unfortunately, it will also increase noise. So, a good practice is, before applying the Laplacian, to smooth the signal by convolving the signal with Gaussian as a pre-processing step, which consequently will not only reduce the noise but also avoid producing ringing artifacts that yields misclassified ridges and valleys.

In the next section, first, the mathematical definition of ridge has been detailed. Then, an extensive theoretical review of the most commonly used ridge detection and extraction filtering techniques are performed. Additionally, extensive analysis of ridge characterizations, driving parameters and its usefulness with different filtering techniques have been examined. Section 4.3 is devoted to objective experimental evaluations of the ridge detection and extraction filtering techniques on both synthetic and real images. Furthermore, the evaluation of the ridge filtering techniques is explored in the context of common noises in the images. At last, the main conclusions have been inferred in Section 4.4.



(a) Image 256×256 containing crest lines

(b) Crest lines in (a) enhanced by the LoG in (d)

(c) Crest lines extracted on (a), after local maximum extraction

(d) Discrete Laplacian of Gaussian (LoG) with parameter $\sigma = 5$, see Eq.2

Figure 4.4: Ridge detection using LoG (Laplacian of Gaussian) filter.

4.2 Ridge extraction in images

NW	**N**	**NE**
W	*P*	**E**
SW	**S**	**SE**

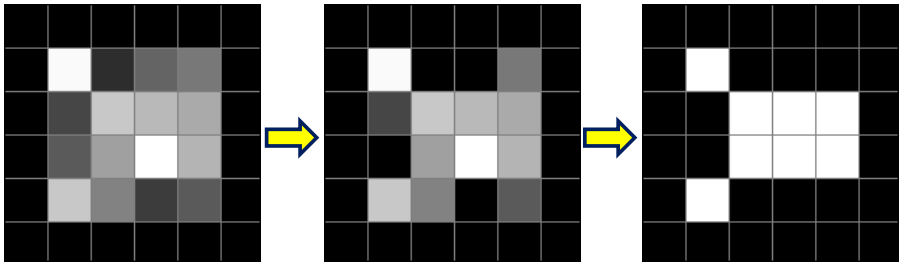
The current pixel value P is preserved only if it corresponds to the highest value along the 4 following axis:

- N-S
- NE-SW
- E-W
- SE-NW

(a) Non-maxima suppression technique in 4 directions inside a 3×3 mask

0	0	0	0	0	0
0	250	45	100	120	0
0	70	200	185	170	0
0	90	160	255	180	0
0	200	130	60	90	0
0	0	0	0	0	0

Pixel values of a processed image



Pixel intensities of a processed image

Non-maxima suppression using mask in (a)

Thresholded image: values > 150

(b) Different steps to obtain a thresholded image with non-maxima suppression in 4 directions

Figure 4.5: Non-maxima suppression technique and different steps to obtain a thresholded image in 4 directions (usually used for ridge/valley detection with DOLD or Laplacian).

Initially, a discrete definition for ridge appears in Crowley and Parker [1984], where the underlying function is the image convolved with a Difference Of Low-Pass (DOLP) transform. Considering two different low-pass filters L_1 and L_2 (i.e., two supports of different widths) both positioned over the center coefficient at the point (0,0), ridges, valleys and blobs may be extracted efficiently with the DOLP transform: these features are highlighted by applying two different low-pass filters to the same image and then subsequently subtracting these two filtered images. Note that the difference of the filters may be applied before convolving the image with the obtained DOLP filter. Afterwards, crest lines are extracted when the support of the low-pass filter L_1 is smaller than the support of the low-pass filter L_2 and inversely regarding valleys.

The final step of the ridge extraction consists in the suppression of the local non-maxima of the magnitude of the ridge/valley intensity, finally the image is thresholded in four directions to obtain thin nets of pixels. Regarding DOLP filter, the selected pixels correspond to points being local maxima in one of the 4 orientations (modulo 180° in degrees) associated with the 8-neighborhood of the pixels. Fig. 4.5 illustrates this process.

Even though the results obtained with square shapes are acceptable (as illustrated in Fig. 4.3(b-c)), the DOLP filter formed by subtracting circularly low-pass filters is preferable. Nevertheless, for their isotropy and circular symmetry properties, the sampled Gaussian filter represents a good achievement. Indeed, the Difference of Gaussian (D*o*G) remains effective in ridge detection and is an approximation of the Laplacian of Gaussian (L*o*G) when the ratio of the size filters is roughly equal to 1.6 (Marr and Hildreth [1980]). Usually called Mexican hat or Sombrero filter, the 2D equation of the L*o*G is given by:

$$\mathrm{LoG}(x,y) = \frac{1}{\pi\sigma^4}\cdot\left(1-\frac{x^2+y^2}{2\sigma^2}\right)\cdot e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{4.2}$$

where (x,y) represents the pixel coordinates and σ is the standard deviation of the Gaussian. A discrete L*o*G is presented in Fig. 4.4(d) and a ridge extraction example in Fig. 4.4(c)

after a non-maxima suppression in the 4 orientations associated with the 8-neighborhood of the pixels (same process than with DOLP filter, see Fig. 4.5). Also, in Haralick's approach (Haralick [1983]), the image function is approximated by a cubic polynomial which, sometimes, may distort the detection.

DOLP transform and L*o*G allow extracting roughly ridges and valleys, but suffer when the desired objects are too thin, thus the detection is disturbed by noise or undesirable artifacts. Besides, the angle selectivity may be improved by applying other operators, as presented in the following section.

4.2.1 Hessian Matrix

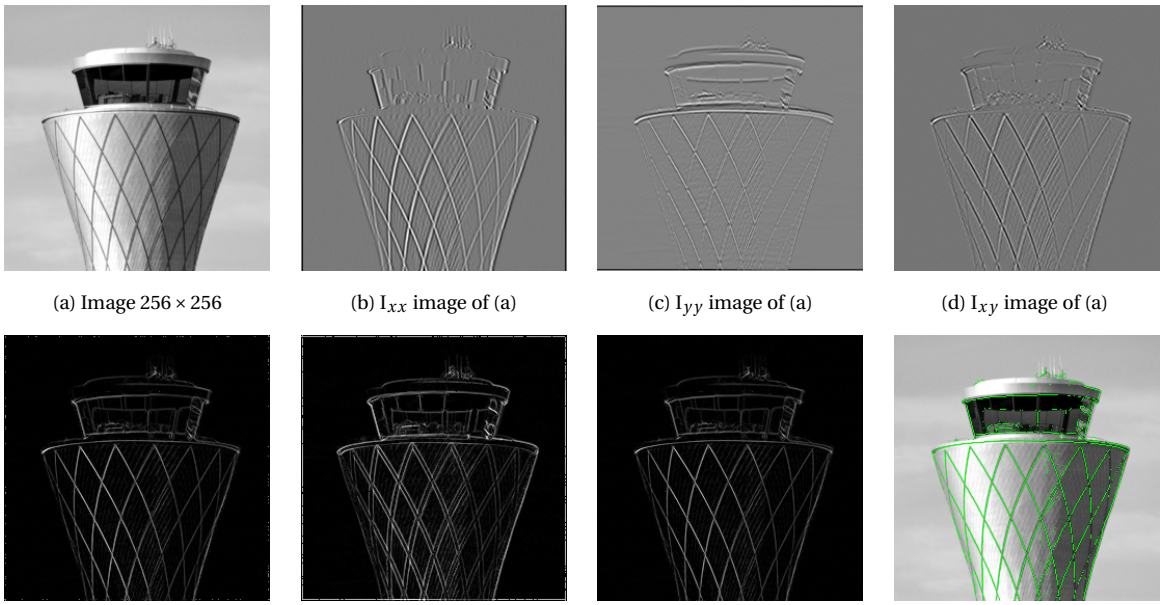
In image filtering, the second order derivative may be used to determine the location of the ridges. Indeed, bright or dark ridges correspond to, respectively, a maximum or minimum of the image intensity in the direction orthogonal to them and a constant image intensity in the direction parallel to them. Considering a gray-level image I and its partial derivatives:

- $\mathrm{I}_{xx} = \partial^2 \mathrm{I}/\partial x^2$, the 2nd image derivative along the x axis, see Fig. 4.6(b),
- $\mathrm{I}_{yy} = \partial^2 \mathrm{I}/\partial y^2$, the 2nd image derivative along the y axis, see Fig. 4.6(c),
- $\mathrm{I}_{xy} = \partial^2 \mathrm{I}/\partial x \partial y$, the crossing derivative of I, see Fig. 4.6(d),

the Hessian matrix $\mathscr{H}$ is often computed in image analysis:

$$\mathscr{H}(x,y) = \begin{pmatrix} \mathrm{I}_{xx}(x,y) & \mathrm{I}_{xy}(x,y) \\ \mathrm{I}_{xy}(x,y) & \mathrm{I}_{yy}(x,y) \end{pmatrix} = \begin{pmatrix} \mathscr{H}_{11} & \mathscr{H}_{12} \\ \mathscr{H}_{21} & \mathscr{H}_{22} \end{pmatrix}. \tag{4.3}$$

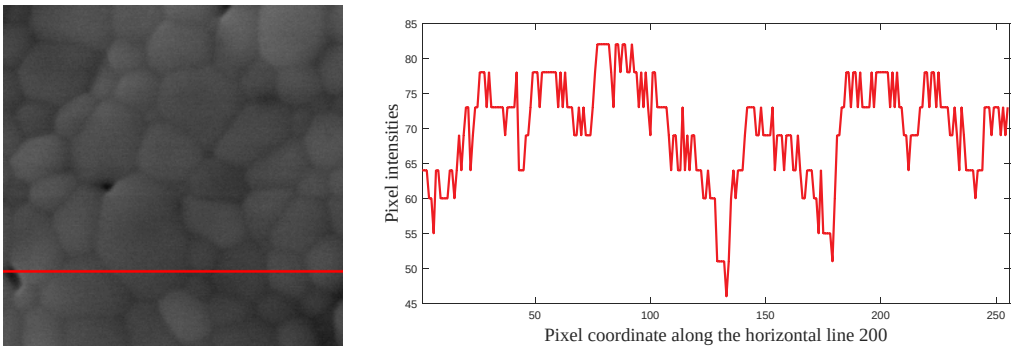
Image derivatives can be calculated by convolving the image with the ±[-1 0 2 0 -1] or the ±[-1 0 1] masks in the x and/or y directions. Note that ±[-1 0 2 0 -1] = ±[-1 0 1]∗[-1 0 1] in the discrete domain.



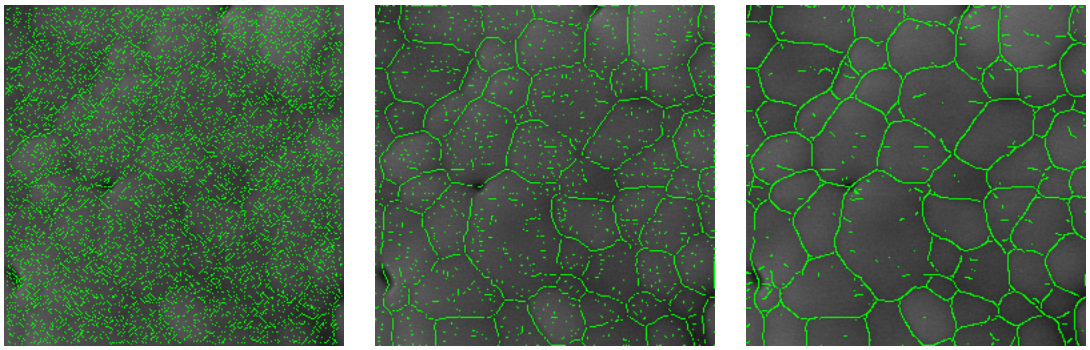
(a) Image 256 × 256 (b) I_{xx} image of (a) (c) I_{yy} image of (a) (d) I_{xy} image of (a)

(e) $|k_1|$ image of $\mathscr{H}$ (f) $|k_2|$ image of $\mathscr{H}$ (g) $|k_1|$ image of $\mathbf{W}$ (h) Valley detection, from (e)

Figure 4.6: Image derivatives and eigenvalue images using Hessian matrix (Eq. 4.3) or Weingarten $\mathbf{W}$ (Eq. 4.6).



(a) Image 256×256 (b) Plot of pixel intensities in (a)



(c) Discrete operator with $\mathscr{H}$ (d) Gaussian with $\mathscr{H}$, σ=1.88 (e) SF_4, σ=1.88

Figure 4.7: Comparison of valley detection on real images. The image in (a) is obtained using scanning electron microscopy of melt ceramic. Here, the valleys are detected with 3 different techniques: Hessian matrix $\mathscr{H}$ without and with Gaussian $\mathbf{G}_\sigma$ in (c)-(d) respectively and steerable filter of order 4 (SF_4) in (e).

The matrix $\mathscr{H}$ is symmetric, diagonalizing $\mathscr{H}$ provides the local normal to the ridge or the valley (that is given by the eigenvector related with the highest eigenvalue) and its sharpness (that is related to the values of these eigenvalues), as detailed in Eberly et al. [1994] nd Steger [1998]. Theoretically, eigenvalues (k_1, k_2) are computed by:

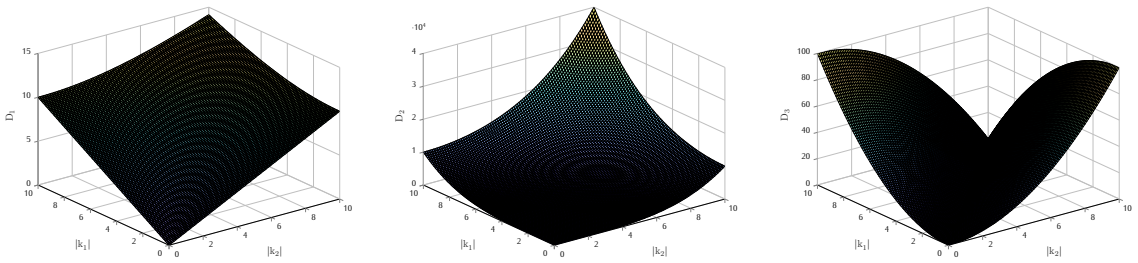
$$\begin{cases} k_1(x,y) & = \frac{1}{2}\cdot(\mathscr{H}_{11}+\mathscr{H}_{22}) - \frac{1}{4}\sqrt{(\mathscr{H}_{11}+\mathscr{H}_{22})^2 + 4\cdot\mathscr{H}_{12}^2} \\ k_2(x,y) & = \frac{1}{2}\cdot(\mathscr{H}_{11}+\mathscr{H}_{22}) + \frac{1}{4}\sqrt{(\mathscr{H}_{11}+\mathscr{H}_{22})^2 + 4\cdot\mathscr{H}_{12}^2}, \end{cases} \tag{4.4}$$

they are visible in Fig. 4.6(e)-(f). Then, eigenvectors, tied to the direction perpendicular to the ridge/valley, are given by:

$$\vec{\theta} = \begin{pmatrix} \mathscr{H}_{12} \\ k_1 - \mathscr{H}_{11} \end{pmatrix}. \tag{4.5}$$

The two eigenvalues k_1 and k_2 correspond to the two main curvatures of the local surface. Besides, there exists several functions $D_{i,i\in\{1,2,3,4\}}$ indicating the local image contrast (Tremblais et al. [2007]):

- $D_1 = k_1$, corresponding to the main eigenvalue, see Steger [1998]
- $D_2 = \sqrt{k_1^2 + k_2^2}$, see Lindeberg [1998]
- $D_3 = (k_1^2 + k_2^2)^2$, see Lindeberg [1998]
- $D_4 = |k_1 - k_2| \cdot |k_1 + k_2|$, see Tremblais et al. [2007].



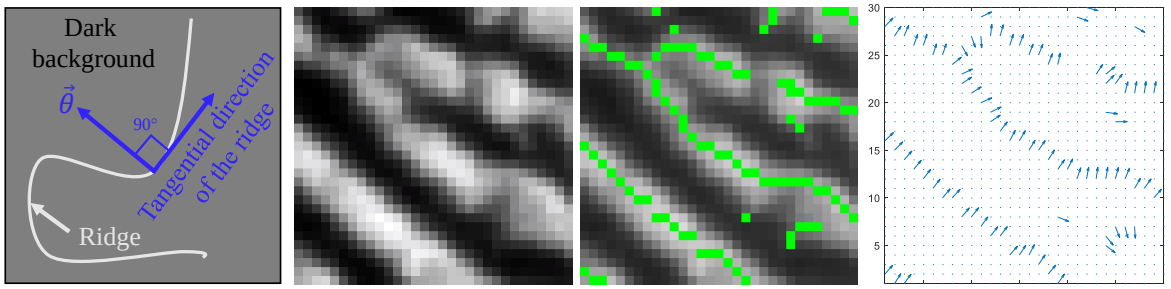
(a) D_2 function (b) D_3 function (c) D_4 function

Figure 4.8: Display of D_2, D_3 and D_4 computing local contrast as a function of (k_1, k_2).

These functions are plotted in Fig. 4.8 as a function of k_1 and k_2.

Eventually, a pixel is labeled as a ridge/valley pixel if $D_{i,i\in\{1,2,3,4\}}$ is maximum in the $\vec{\theta}$ direction. It is selected after non-maximum suppression (Canny [1986]) where the values of the magnitude are linearly interpolated between closest points in the 8-neighborhood. Fig. 4.9 illustrates a ridge detection after non-maxima suppression in the $\vec{\theta}$ direction.

In practice, regarding real images, due to the luminance variation, acquisition and/or compression noise, the detection of pure ridges/valleys is almost impossible. So, in order to more reliably extract the ridges, the convolution of the image with a low-pass filter is considered, as detailed in Sec. 4.2.3.



(a) Image with a ridge (b) Image 30×30 (c) Ridge detection in green (d) θ direction

Figure 4.9: Example of extracted ridges with their tied perpendicular directions.

4.2.2 Weingarten

Weingarten map represents the differential of the Gauss map (Do Carmo [2016]). This expression can be computed directly from the first (i.e., $I_x = \frac{\partial I}{\partial x}$ and $I_y = \frac{\partial I}{\partial y}$) and second derivatives in the x and y directions of the images. The linear invariants of the Weingarten map are the intrinsic curvatures of the surface: the eigenvalues are the principal curvatures, the trace is the mean curvature, and the determinant is the Gaussian curvature:

$$\mathbf{W}(x,y) = \frac{1}{(1+I_x^2+I_y^2)^{\frac{3}{2}}} \cdot \begin{pmatrix} 1+I_y^2 & -I_xI_y \\ -I_xI_y & 1+I_x^2 \end{pmatrix} \cdot \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}. \tag{4.6}$$

The eigenvalues and eigenvectors of $\mathbf{W}$ are extracted with the same procedure as in Eq. 4.4 and 4.5, regarding coefficients of the matrix $\mathbf{W}$. The same procedure applies for the non-maxima suppression in the $\vec{\theta}$ direction. In Armande et al. [1999], ridges or valleys are extracted by, first, smoothing the image with a Gaussian and then considering D_1.

4.2.3 Low pass filters for ridge detection

A low pass filter is the basis for most smoothing methods, as it is needed to apply a smoothing process over an image prior to the feature extraction. The optimization criteria, based on the Canny theory, are: (i) detection efficiency, (ii) location accuracy of the detected contour and (iii) uniqueness condition of filter response to its output for an input signal (Canny [1986]). Based on this theory, several low pass filters have been proposed in the literature. In the following, three low-pass filters and their second derivatives are discussed for ridge and valley detection.

4.2.3.1 Ziou Filter Z

In Ziou [2000], the author described an optimal line detector allowing an economic temporal complexity because it represents a second order recursive filter. Considering $t \in \mathbb{R}$, the equation of the 1D low pass filter Z is given by:

$$Z(t) = \frac{1}{s_z^2} \cdot (1 + s_z \cdot |t|) \cdot e^{-s_z \cdot |t|}, \tag{4.7}$$

where the filter parameter s_z represents a positive constant. The second derivative of Z is obtained by derivation as a function of t, two times:

$$z(t) = (s_z \cdot |t| - 1) \cdot e^{-s_z \cdot |t|}. \tag{4.8}$$

Note that the same procedure is available to obtain the 1st derivative of the filter Z, as for the following presented filters.

4.2.3.2 Gouton Filter R

Gouton et al. [2000] described a third order recursive filter. This ridge/valley-line detector function is able to modify its shape as a function of its parameter s_r:

$$R(t) = (K \cdot \sin(s_r \cdot |t|) + D \cdot \cos(s_r \cdot |t|) + E) \cdot e^{-s_r \cdot |t|}, \tag{4.9}$$

with: $A = \frac{-s_r \cdot (2s_r^2 - t^2)}{s_r \cdot (2s_r^2 + t^2)}$, $K = \frac{1}{4 \cdot s_r^4}$, $D = \frac{2s_r^2 \cdot A}{4 \cdot s_r^4}$ and $E = \frac{A \cdot s_r + s_r}{s_r^3}$.

Thus, the second derivative of R has the following form:

$$r(t) = (\cos(s_r \cdot |t|) - s_r \cdot \sin(s_r \cdot |t|) - (s_r + 1)) \cdot e^{-s_r \cdot |t|}, \tag{4.10}$$

the more the s_r parameter decreases, the more r enhances fine ridges/valleys. Furthermore, when s_r decreases, the shape of R is nearly a Gaussian, as shown in Fig. 4.10.

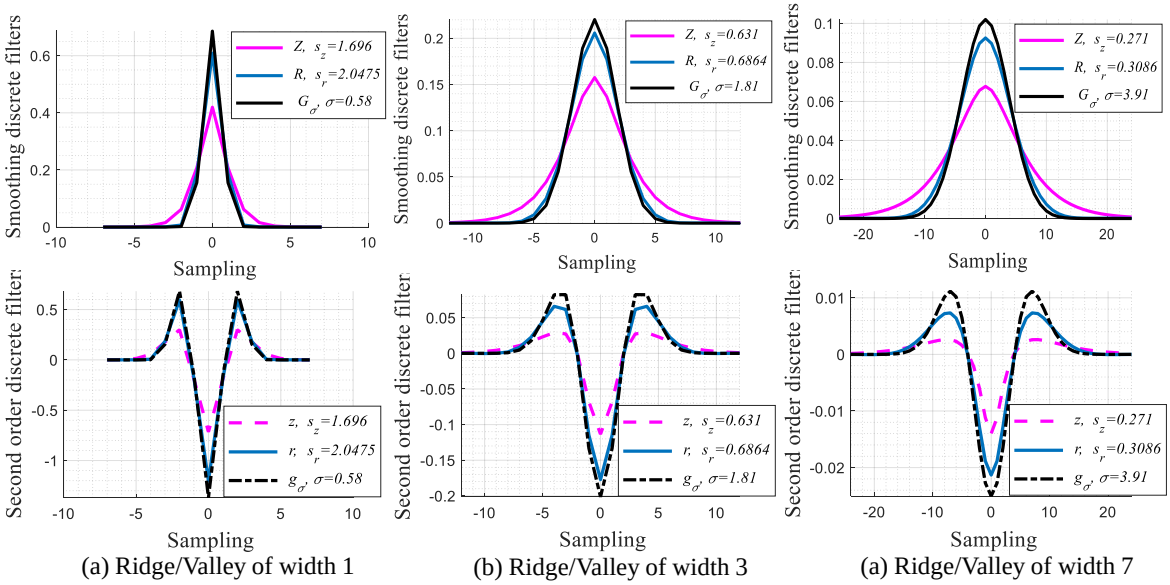
4.2.3.3 Gaussian Filter

Gaussian kernels are regularly used for their effectiveness in edge detection (Canny [1986]), the 1D equation is:

$$G_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{t^2}{2\sigma^2}} \tag{4.11}$$

with σ representing the standard deviation of the Gaussian. Thus, the second derivative of the Gaussian G_σ is given by:

$$g_\sigma(t) = \frac{\partial^2 G_\sigma}{\partial t^2}(t) = \frac{t^2 - \sigma^2}{\sqrt{2\pi}\sigma^5} \cdot e^{-\frac{t^2}{2\sigma^2}}. \tag{4.12}$$



(a) Ridge/Valley of width 1 (b) Ridge/Valley of width 3 (a) Ridge/Valley of width 7

Figure 4.10: Visual comparison of tested normalized 1D low-pass filters and 2nd order filters with the ideal parameters tied to the width of the ridge/valley.

The two dimensional Gaussian $\mathbf{G}_\sigma$ is built by combining with "$*$", a product of convolution, G_σ horizontally and vertically:

$$\mathbf{G}_\sigma(x,y) = \mathrm{G}_\sigma * \mathrm{G}_\sigma^\top(x,y) = \frac{1}{2\pi\sigma^2} \cdot \mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{4.13}$$

where $\top$ represents transpose. Using the Gaussian G_σ, the strategy is the same as to compute the second derivative on an image, with G_σ and $g_\sigma^\top$, as an example for an image derivative in y, see Fig. 4.11(d). Moreover, the section 4.2.4.4 is dedicated to the strategies of the two dimensional filters implementation.

Furthermore, these filters in Eq. 4.7, 4.10 and 4.12 are useful to smooth the image before extracting edges by computing $\mathscr{H}$ matrix presented in Eq. 4.3. Additionally, it is also possible to use the Weingarten (cf. Eq. 4.6) with the Gaussian, as in Armande et al. [1999].

4.2.3.4 Parameters

The three above-mentioned filters are suitable for ridge and valley detection. Considering one filter, it is adjustable by tuning only one parameter which is the same for the low pass and the derivative filter. Accordingly, parameters s_z, s_r and σ are chosen as a function of the width of the ridge or of the valley. Consequently, these parameters are selected by increasing the width of the filter as robust as possible in order to extract suitably the feature.

Here, the main idea is to compare equivalently the 3 filters z, r and g_σ as a function of the feature width. Indeed, the objective is to tune each filter for a specific width by selecting an appropriate parameter (s_z, s_r or σ). Thus, in the discrete domain, parameters s_z and s_r are decreasing, and parameter σ is increasing until the filter coefficients cross 0 and the shape filter contains the width of the feature at the same time. Fig. 4.10 illustrates the selected filers computed with different parameters as a function of the width of the feature. In addition, Tab. 4.1 references the optimum parameters for each filter as a

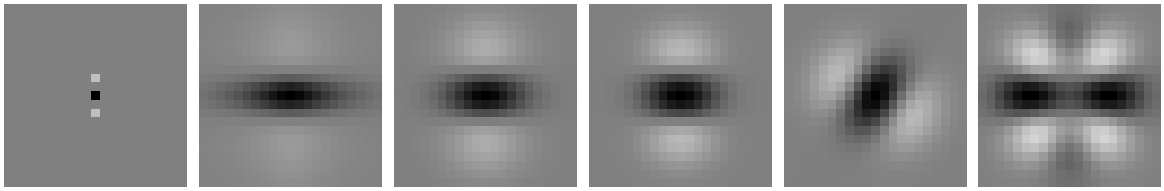
Table 4.1: Optimum parameters of the high-pass filters z, r and g_σ as a function of the width of the ridge or valley.

Width	1	3	5	7	9	11	13	15	17
s_z	1.696	0.63	0.37	0.27	0.21	0.17	0.15	0.13	0.12
s_r	2.05	0.69	0.42	0.31	0.24	0.20	0.17	0.15	0.13
σ	0.58	1.81	2.88	3.91	4.93	5.94	6.95	7.95	8.96

function of the features size from 1 to 15 pixels. Finally, the parameter σ of the Gaussian has the same properties regarding oriented filters widths.

4.2.4 Oriented filters

One common task in image processing and computer vision is applying the same filter on different angles in order to detect directional responses as Steerable Filter (Freeman and Adelson [1991], Jacob and Unser [2004]), Anisotropic Gaussian Kernel (Lopez-Molina et al. [2015]) and Logical Linear Filter (Iverson and Zucker [1995]).



(a) [1 0 -2 0 1] (b) $Z * z^\top$ (c) $R * r^\top$ (d) $G_\sigma * g_\sigma^\top$ (e) $SF_2^{\theta=30^\circ}$ (f) $SF_4^{\theta=0^\circ}$

Figure 4.11: Two dimensional discrete filters for valley detection in y direction of width 7, parameters are available in Tab.4.1. The negative of filters detect ridges.

4.2.4.1 Steerable filter

Two dimensional Gaussian kernels $\mathbf{G}_\sigma$ are very useful for their properties of isotropy, steerability or decomposability properties as separability (see Sec. 4.2.4.4). Freeman and Adelson proposed an efficient architecture to design oriented filters of arbitrary orientations from linear combinations of basis filters (Freeman and Adelson [1991]). Thus, applying filter steered in different directions, and then computing the filter responses gives significant description of the orientation for the considered pixel; also it allows determining analytically the filter output as a function of orientation. The initial step to extract ridges or valleys in images is to estimate their orientation by using even steered filters. Consequently, the steerable filter of second order (SF_2) considering $\mathbf{G}_\sigma$ in Eq. 4.13 oriented at the angle θ is:

$$SF_2^\theta = \cos^2(\theta) \cdot \frac{\partial^2 \mathbf{G}_\sigma}{\partial x^2} + \cos(\theta)\sin(\theta) \cdot \frac{\partial^2 \mathbf{G}_\sigma}{\partial x \partial y} + \sin^2(\theta) \cdot \frac{\partial^2 \mathbf{G}_\sigma}{\partial y^2}. \tag{4.14}$$

This allows computing an even filter at a specific orientation, as illustrated in Fig. 4.11(e). At the end, the calculation of the ridges or the valleys corresponds to the filter energy in the direction of the maximum response of the template.

Jacob and Unser [2004] extend the idea of the steerable filter of order 2 (SF_2) with operators having a better orientation selectivity. Indeed, they proposed higher order functions, issued of higher order derivatives of the Gaussian $\mathbf{G}_\sigma$ (2nd and 4th: $\mathbf{G}_{yy}$, $\mathbf{G}_{xx}$, $\mathbf{G}_{yyyy}$, $\mathbf{G}_{xxyy}$, $\mathbf{G}_{xxxx}$), resulting in more elongated templates, as visible in Fig. 4.11(f). Regarding ridge detection, this filter is specified so as to provide the best compromise in terms of signal-to-noise ratio, false detection, and localization (as illustrated in Fig. 4.7(d)). Thus, the even steerable filter of 4th order (SF_4) is formulated as:

$$SF_4(x,y) = \alpha_1 \cdot \mathbf{G}_{yy} + \alpha_2 \cdot \mathbf{G}_{xx} + \alpha_3 \cdot \mathbf{G}_{yyyy} + \alpha_4 \cdot \mathbf{G}_{xxyy} + \alpha_5 \cdot \mathbf{G}_{xxxx}, \tag{4.15}$$

with:

- $\alpha_1 = -0.392 \cdot \sigma$,
- $\alpha_2 = 0.113 \cdot \sigma$,
- $\alpha_3 = 0.034 \cdot \sigma^3$,
- $\alpha_4 = -0.184 \cdot \sigma^3$,
- $\alpha_5 = 0.025 \cdot \sigma^3$.

Such parameters control the template SF_4 not to produce undesirable oscillations and side-lobes along y which is contrary to the 3rd Canny criterion: unicity (see Jacob and Unser [2004]). This 2D template, presented in Fig. 4.11(f), can be steered in different orientations θ, as detailed in Jacob and Unser [2004], to extract ridges and valleys.

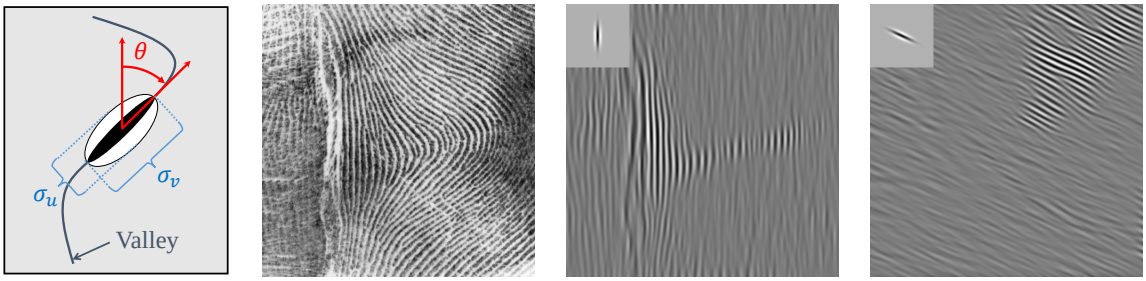
4.2.4.2 Anisotropic Gaussian Filter

Kernels based on the derivative of anisotropic Gaussian functions have been successfully applied in edge detection, showing certain advantages compared with the isotropic Gaussian derivatives (Perona [1992]). As an example, as it corresponds to a thin filter, the orientation selectivity becomes more reliable with the anisotropic Gaussian derivatives than the isotropic Gaussian filter. Moreover, the isotropic Gaussian often makes the ridge extraction difficult as crossing lines in images; also, parallel lines could be blurred into one line due to isotropic smoothing, especially if the smoothing parameter is too large (i.e., σ parameter in Eq. 4.11 and 4.12). The origin is that anisotropic property is more efficient at level of straight lines. It corresponds to a narrow filter which is oriented in different directions to extract the edges when it is steered in the edge direction.

The anisotropic Gaussian filter can thus take advantage of this property and overtake drawbacks of the isotropic filter. Thereafter, it is necessary to filter the image with a set of $360/\Delta\theta$ kernels oriented in a variety of directions, as such, leading to the characterization of the partial derivatives in $360/\Delta\theta$ different orientations. The most evident option to produce a single output from that information is to retain the result produced by the oriented kernel with the maximum absolute value. An anisotropic Gaussian filter in two dimensions is not built with the combination of isotropic kernels, as Eq. 4.15, its direct equation is given by Geusebroek et al. [2003]:

$$\mathscr{G}_{\sigma_u,\sigma_v,\theta}(x,y) = \frac{1}{2\pi\sigma_u\sigma_v} \cdot e^{-\frac{1}{2}\left(\frac{(x\cos\theta+y\sin\theta)^2}{\sigma_u^2} + \frac{(-x\sin\theta+y\cos\theta)^2}{\sigma_v^2}\right)}. \tag{4.16}$$

Here, (σ_v, σ_u) represent the two parameters of the anisotropic Gaussian, i.e., the standard deviations. When $\sigma_v = \sigma_u$, the kernel $\mathscr{G}$ reduces to an isotropic Gaussian kernel $\mathbf{G}_{\sigma_v}$ or



(a) SOAGK (b) Image 512×512 (c) $\mathscr{G}''$ filter, $\theta = 0°$ (d) $\mathscr{G}''$ filter, $\theta = 115°$

Figure 4.12: Representation and visualization of the second derivative, an image computed by convolution with the SOAGK with $\sigma_u = 2.88$ and $\sigma_v = 5\sigma_u$.

$\mathbf{G}_{\sigma_u}$. To extract ridges, the Second-Order Anisotropic Gaussian Kernel (SOAGK) can be applied (Lopez-Molina et al. [2015]). Considering the vertical anisotropic Gaussian directed at $\theta = 0$:

$$\mathscr{G}_{\sigma_u,\sigma_v,\theta=0}(x,y) = \frac{1}{2\pi\sigma_u\sigma_v} \cdot e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_u^2}+\frac{y^2}{\sigma_v^2}\right)},$$

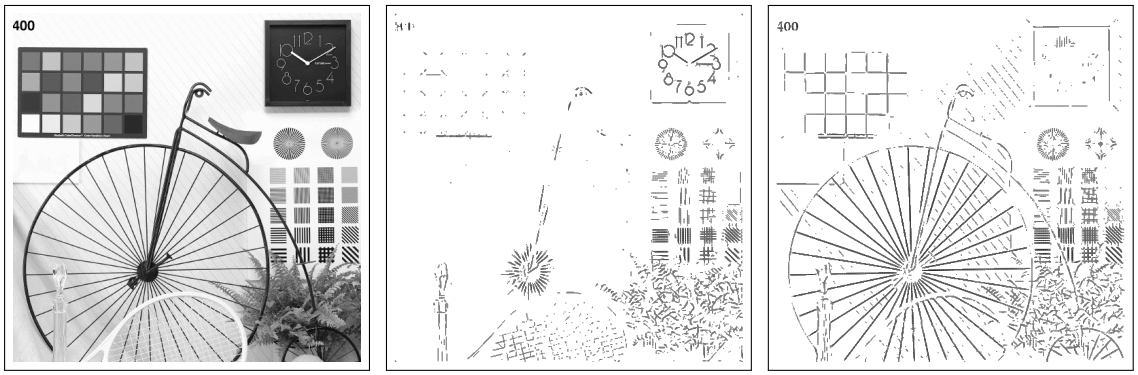
thus, the second derivative of $\mathscr{G}_{\sigma_u,\sigma_v,\theta=0}$ in the x direction is calculated by:

$$\mathscr{G}''_{\sigma_u,\sigma_v,\theta=0}(x,y) = \frac{\partial^2 \mathscr{G}_{\sigma_u,\sigma_v,\theta=0}}{\partial x^2}(x,y) = \frac{x^2-\sigma_u^3}{2\pi\sigma_u^5\sigma_v} \cdot e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_u^2}+\frac{y^2}{\sigma_v^2}\right)}. \tag{4.17}$$

The choice of $\sigma_v > \sigma_u$ enables to build a narrow filter smoothing mostly in the y direction while highlighting valleys in the x direction. Now, this 2D kernel can be oriented in different directions to capture valleys (or ridges with the opposite filter) in the image, as illustrated in Fig. 4.12. To this end, the anisotropic parameter produces a smoothing along the ridge/valley, which helps to extract easily elongated features, even disturbed by noise. On the contrary, kernels having parameters $\sigma_v/\sigma_v \approx 1$ highlight undesirable features as noise which are interpreted as small, non-elongated ridges (Lopez-Molina et al. [2015]).

4.2.4.3 Logical Linear Filter

Similar to the SOAGK, Iverson and Zucker proposed a hybrid filter by combining directional linear filters and a Linear-Logical (L/L) operator which helps to reduce the false



(a) Image 512 × 512 (b) Positive contrast lines (c) Negative contrast lines

Figure 4.13: Directions of contrast lines obtained by $\mathscr{S}_P$ and $\mathscr{S}_N$ (vectorial images)

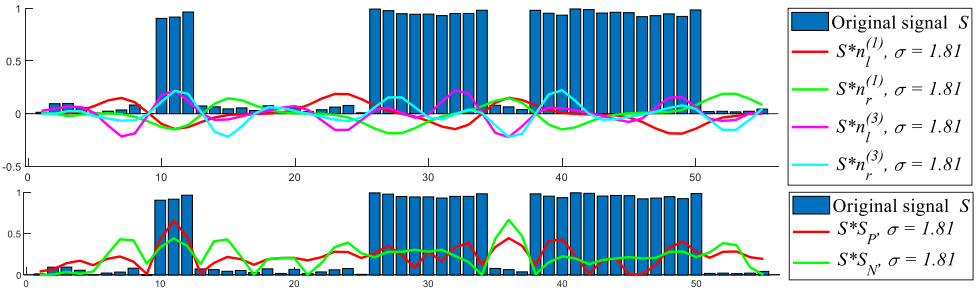


Figure 4.14: Convolution of a 1D signal with the n'_l, n'_r, $n^{(3)}_l$ and $n^{(3)}_r$ with ϵ=2 to compute Positive and Negative contrast lines with $\mathscr{S}_P$ and $\mathscr{S}_N$ respectively.

positive pixels of ridges/valleys (Iverson and Zucker [1995]). This technique allows selecting any inflection points within the 1D signal region $[t-\epsilon, t+\epsilon]$, with $\epsilon > 0$, see Fig. 4.14. First, it depends on the Gaussian G_σ (see Eq. 4.11) and its derivatives of the first and third order G'_σ and $G^{(3)}_\sigma$ by computing the four parameters:

$$\begin{cases} n'_l = G'_\sigma(t+\epsilon)/2\epsilon, & n'_r = G'_\sigma(t-\epsilon)/2\epsilon, \\ n^{(3)}_l = G^{(3)}_\sigma(t+\epsilon)/2\epsilon, & n^{(3)}_r = G^{(3)}_\sigma(t-\epsilon)/2\epsilon, \end{cases} \tag{4.18}$$

thereby, they can be applied to a signal, as shown in Fig.4.14.

Concretely, the L/L operator can be utilized on different edge types as ridge P (Positive contrast lines), valleys N (Negative) and Edges E (ramp or step). In this study, only P and N are focused and evaluated. These denoted functions $\mathscr{S}_P$ and $\mathscr{S}_N$ respectively combine linear operators in Eq. 4.18 by using the logical operator $\overline{\wedge}$ such that:

$$\begin{array}{rcl} \mathscr{S}_P & = & n'_l \,\overline{\wedge}\, n'_r \,\overline{\wedge}\, n^{(3)}{}_l \,\overline{\wedge}\, n^{(3)}{}_r \\ \mathscr{S}_N & = - & n'_l \,\overline{\wedge}\, -n'_r \,\overline{\wedge}\, -n^{(3)}{}_l \,\overline{\wedge}\, -n^{(3)}{}_r \end{array} \tag{4.19}$$

where the logical operator $\overline{\wedge}$ is represented by, for two hypotheses (a, b):

$$a \,\overline{\wedge}\, b \triangleq \begin{cases} a+b, & \text{if } a>0 \wedge b>0 \\ b, & \text{if } a>0 \wedge b\leq 0 \\ a, & \text{if } a\leq 0 \wedge b>0 \\ a+b, & \text{if } a\leq 0 \wedge b\leq 0 \end{cases} \tag{4.20}$$

In this way, $\mathscr{S}_P$ and $\mathscr{S}_N$ contribute to extract convex and concave points, as shown in Fig. 4.14. Next, to extract ridges or valleys and their tied directions, the 2D operator is expressed as the Cartesian product of orthogonal, 1D L/L operators $\mathscr{S}_P$ or $\mathscr{S}_N$ operators and a tangential operator $T(t)$. Moreover, this 2D operator is oriented and uses strategies of the logical operator $\overline{\wedge}$ with the tangential operator $T(t)$ to (a) discriminate between locally continuous and discontinuous curves along their tangent direction in the image; and, (b) align the line termination with the line ending (illustrated in Fig. 4.13, for more details refer to Iverson and Zucker [1995]). To sum up, the L/L operator is similar to the SOAGK, with the parameter σ_u tied to the normal operator ($\mathscr{S}_P$ and $\mathscr{S}_N$) and σ_v for the tangential operator T (see Fig. 4.12(a)).

4.2.4.4 Implementation and complexity

Presented filters Z, R and G_σ may be implemented with different strategies. Filters Z and R have been designed to be implemented recursively. Also known as Infinite Impulse Response (IIR) filters, they represent filters where the output sample is a linear combination

Table 4.2: Recursive orders of the filters and image computations as a function of the chosen technique.

Filter	Z and z	R and r	G_σ and g_σ
R order	2, Ziou [2000]	3, Gouton et al. [2000]	4, Deriche [1992] or 5, Van Vliet et al. [1998]

of some number of previous inputs and outputs. Even though there are other ways to apply them (i.e., via Fast Fourier Transform), the recursive implementation strategy is compared here. First, they correspond to separable filters, so they can be written as a product of two 1D filters which is equivalent to a typical 2D convolution operation providing a reduction of the operator computational cost.

Secondly, to reduce even more the number of operations per pixel, each 1D Z, R or G_σ filter may be also implemented recursively, Table 4.2 reports the order of these 3 detailed filters. To reduce the number of operations per pixel, a M-order recursive filter is obtained by calculating its Z-transform. Thus, the two-sided sequence of a filter F is the superposition of a causal filter F_- and anti-causal filter F_+: $F(n) = F_-(n) + F_+(n)$, for $n = \{1,...,M\}$. To minimize the computational complexity authors of Van Vliet et al. [1998] proposed to decompose series interconnection into a product of the causal and anti-causal parts, leading to a 3rd-order Gaussian filter, a 4th-order first derivative filter and a 5th-order second derivative filter (many fast approximations of the Gaussian have been proposed, some of them are detailed in Getreuer [2013]). .

Now, the first and second derivatives of the original image can be computed easily by applying the [-1 0 1] mask one or two times respectively to the smoothed image (i.e., smoothed image obtained by applying the Z, R or G_σ filters both in x and y directions). Besides, the derivatives of an image are computed by combining in the two directions x and y the different 1-dimensional filters presented in the beginning of the section 4.2.3 (which are implemented with the different strategies above). As an example, the second derivative in the x direction of an image with the filter Z can be obtained by applying the low pass filter Z in the y direction then the second derivative of z, called Z in the x direction of the filtered image.

Subsequently, Table 4.3 specifies the required number of image computations as a function of the segmentation technique (L*o*G, $\mathscr{H}$, **W**, SF_2, SF_4, or L/L) and Fig. 4.15 roughly schematize the complexity. The Hessian Matrix $\mathscr{H}$ needs the second derivatives of the image I_{xx}, I_{yy} and I_{xy}, using Z, R or G filters. Obviously, **W** is more computation-

Table 4.3: Image computations as a function of the chosen technique.

	Computed basis images	Rotation	Other
L*o*G	I_{xx} and I_{yy}	-	-
$\mathscr{H}$	I_{xx} , I_{yy} and I_{xy}	-	D_1, D_2, D_3 or D_4
W	I_x, I_y, I_{xx} , I_{yy} and I_{xy}	-	Usually D_1
SF_2	I_{xx} or I_{yy}	yes	max and argmax
SF_4	I_{xx}, I_{yy}, I_{xxxx}, I_{yyyy}, I_{xxyy}	yes	max and argmax
SOAGK	I_{xx} or I_{yy}	yes	max and argmax
L/L	I_x, I_{xxx}	yes	$\overline{\wedge}$, endline, stabilizer

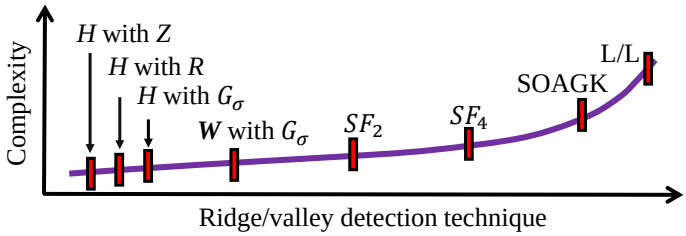


Figure 4.15: Complexity schema, depending on the recursive filters order, the number of calculated images (Tab. 4.3) and the filter rotation.

ally complex than the L*o*G or $\mathscr{H}$ because it needs more image derivatives. Regarding the steerable filters, an operation of filter rotation with an angle θ is necessary (with 360°/Δθ total rotations, where Δθ is the angular step); and 5 derivative images are calculated for the steerable filter of order 4 (see Eq. 4.15). On the other hand, the number of basis filters is large to extract features with the SOAGK, and the basis filters are non-separable, requiring high computational loads. In Geusebroek et al. [2003], the anisotropic Gaussian is decomposed into two Gaussian 1D filters by considering 360°/Δθ steps of rotation, allowing reducing the operation number per pixel (to approximate the SOAGK, the difference of anisotropic Gaussian by differentiating the whole image array with two different standard deviations σ_u in Eq. 4.16 is calculated, see Geusebroek et al. [2003]). Also, the L/L filter contains several steps of interpolation for the normal operator ($\mathscr{S}_\mathrm{P}$ and $\mathscr{S}_\mathrm{N}$) and for the tangential operator T which are directed in different directions in the image. Moreover, the L/L uses other strategies such as the endline or the stabilizer to qualify the segmentation; these steps add more filter complexity.

4.3 Experimental results and evaluation

Experiments are performed on synthetic and real images, showing qualitative and quantitative results. A first result presented in Fig. 4.16 illustrates the advantage to use sharp and narrow filter to extract thin and close objects, as filters z and r.

The aim here is to extract branches inside the dragonfly wings; as this image does not contain any noticeable noise, the Hessian matrix H with finite filters like [1 0 -2 0 1] gave interesting results for these thin objects, but created many undesirable edge points around certain valleys (similar segmentation also by SF_4). Elsewhere, H with the Gaussian g_σ and D_1 brings similar but less complete result. Segmentation obtained with $\mathscr{H}$ and D_2, D_3 and D_4 are worse with a lot of missing edge points, as with SF_2. However, the valley extraction using $\mathbf{W}$ is perfectible. On the other hand, the result using $\mathscr{H}$ with z filter is quite perfect (Fig. 4.16(b)), this justifies the need to use low pass filter. Among all the ridge/valley detectors, exponential (z or r) filters do not delocalize contour points (Laligant et al. [2007]), whereas they are sensitive to noise. Techniques using Gaussian filters are less sensitive to noise, but suffer from rounding bends and junctions like the oriented filters SF_2, SF_4 and the SOAGK. The more the 2D filter is elongated, the more the segmentation remains robust against noise. In the following sections, quantitative results are reported with different types and levels of noise in synthetic images. Then evaluations will involve real images.

4.3.1 Error quantification and evaluation procedure

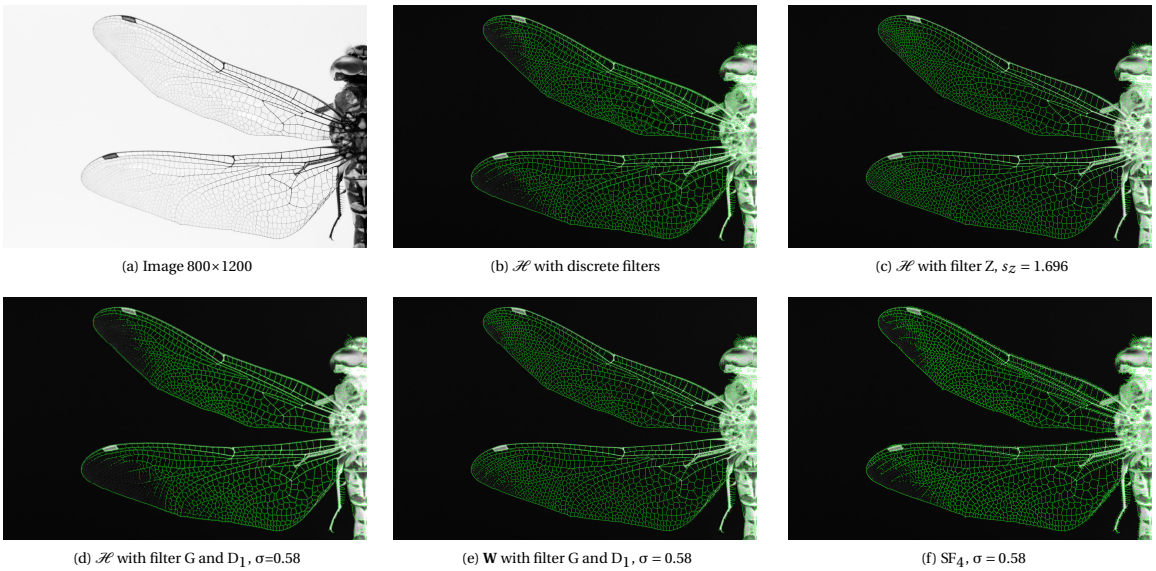
Evaluations are reported using synthetic images where the true positions of the edges are known. Let G_t be the reference contour map corresponding to the ground truth and D_c the detected contour map of an image I. Comparing pixel by pixel G_t and D_c, a basic evaluation is composed of statistics:

- True Positive (TP), common points of both G_t and D_c;
- False Positive (FP), spurious detected edges of D_c;
- False Negative (FN), missing boundary points of D_c;
- True Negative (TN), common non-edge points.

where $|\cdot|$ denotes the cardinality of a set. Several edge detection evaluation techniques involving only statistics have been developed, see Magnier et al. [2018]. It is clearly proved that poorly located or missing pixels should be penalized according to the distance from the position where they should be localized. Also, as demonstrated in Magnier et al. [2018], the evaluation of FP and FN should not be symmetrical, because such a penalty could alter the visibility of the outlines of the desired objects in an objective evaluation (see Magnier et al. [2018]): some measures calculate a large error for a single FP at a sufficiently large distance, while many desired contours are missing, but unfortunately, they are not penalized enough. Thus, described in Magnier [2019], the normalized N edge detection evaluation measure is, for $FN > 0$ or $FP > 0$:

$$\mathcal{N} = \frac{1}{FP + FN} \cdot \left[\frac{FP}{|D_c|} \cdot \sum_{P \in D_c} \frac{1}{1 + K_{FP} \cdot d^2_{Gt}(p)} + \frac{FN}{|Gt|} \cdot \sum_{P \in Gt} \frac{1}{1 + K_{FN} \cdot d^2_{Dc}(p)} \right], \tag{4.21}$$

where $(\delta, \kappa) \in]0,1]^2$ represent two scale parameters Magnier [2019], $|\cdot|$ denotes the cardinality of a set, and $d_A(p)$ is the minimal Euclidean distance between a pixel p and a set A Magnier et al. [2018]. Therefore, the measure $\mathcal{N}$ calculates a standardized dissimilarity



(a) Image 800×1200 (b) $\mathscr{H}$ with discrete filters (c) $\mathscr{H}$ with filter Z, $s_Z = 1.696$

(d) $\mathscr{H}$ with filter G and D_1, σ=0.58 (e) **W** with filter G and D_1, $\sigma = 0.58$ (f) SF_4, $\sigma = 0.58$

Figure 4.16: Valley detection in green on real image of a dragonfly, with thin, blurred and very close junctions. The original image is inverted for a better visualization.

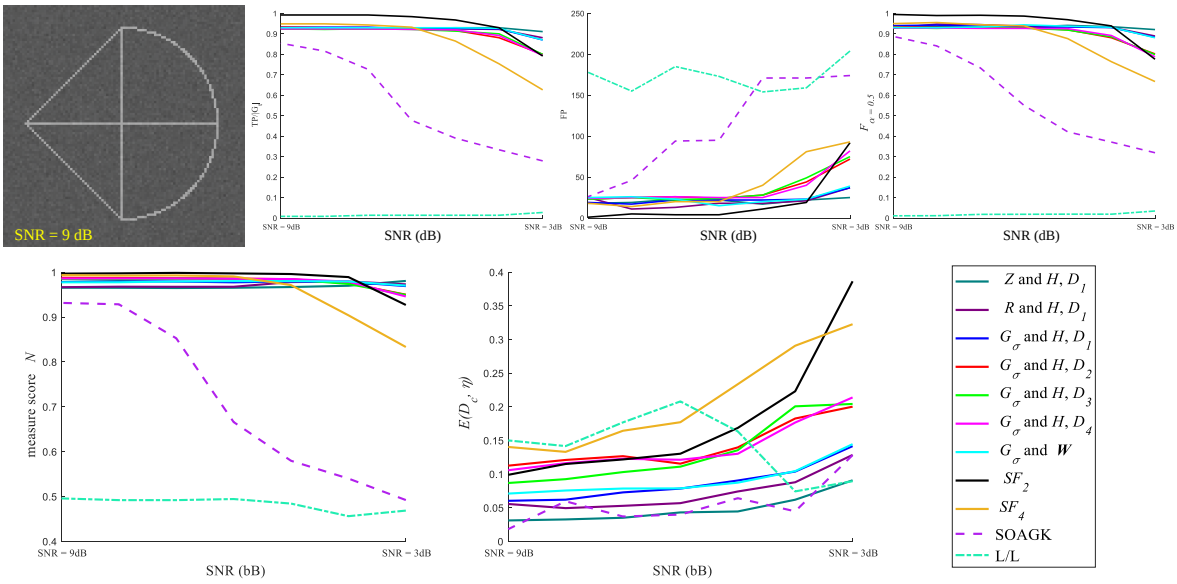


Figure 4.17: Evaluation of the different ridge/valley extraction techniques on synthetic images corrupted by Poisson noise.

score; the closer the evaluation score is to 1, the more the segmentation is qualified as suitable. On the contrary, a score closes to 0 corresponds to a poor detection of contours.

The aim here is to get the best contour map in a supervised way. For that, the contours are extracted after a suppression of the local non-maxima, then a threshold by hysteresis is applied to obtain a binary segmentation (Canny [1986]). Theoretically, to be objectively compared, the ideal contour map of a measure must be a D_c at which the supervised evaluation gets the highest score, see details in Magnier et al. [2018] and Magnier [2019]. For each better segmentation tied to $\mathcal{N}$, the FP and percentage of TP relative to the total number of edge pixels of G_t are also displayed (TP/G_t). In addition, the last evaluation measure concerns the angle tied to the ridge/valley, θ. Considering C_{D_c}, the set of contour chains in D_c (i.e., at least 2 pixels per chain), the angle evaluation is computed as follows:

$$E(C_{D_c},\theta) = \frac{1}{|C_{D_c}|} \cdot \sum_{p \in C_{D_c}} \sum_{d_k \in \omega} \left[1 - \frac{\left| 90° - \left| \overrightarrow{\theta_p} - \overrightarrow{\theta_{d_k}} \right| \right|}{90°} \right] / c_k,$$

where d_k represents a contour pixel belonging to ω, a 3×3 window centered on p, $\overrightarrow{\theta_{d_k}}$ the direction tied to d_k and c_k the number of contour pixels in ω, minus the central pixel. This evaluation linearly ranges from 0 for identical angles of $\overrightarrow{\theta_p}$ and $\overrightarrow{\theta_{d_k}}$ to 1 for angles that differ. Note that $\overrightarrow{\theta_{d_k}}$ and $\overrightarrow{\theta_p}$ angles belong to $[0;180°[$ and when one direction approximates 0 and the other direction 180°, the evaluation remains close to 0.

Also, from the proper binary confusion matrix, the precision (P_{rec}) and recall (R_{ec}) evaluations are computed, given the overall quality expressed in terms of the F_α-measure:

$$F_\alpha = \frac{P_{rec} \cdot R_{ec}}{\alpha P_{rec} + (1-\alpha) R_{ec}} \text{ with } P_{rec} = \frac{TP}{TP+FP} \text{ and } R_{ec} = \frac{TP}{TP+FN} \tag{4.22}$$

with α=0.5 allows an equal penalization between FN and FP.

These scores are presented throughout the remainder of this study, according to different images and noise types.

4.3.2 Synthetic images corrupted by Poisson noise

The first image in Fig. 4.17 contains ridges of width 1 pixel and is corrupt by Poisson noise. This noise distribution typically models shot noise in a sensor in which the time between photon arrivals is governed by Poisson statistics, and appears for example in medical imaging system. The Poisson noise density models the time statistics between photon arrivals, where the defined expected number of occurrence of integer Λ, the noise probability corresponds to the observed number of n instead of λ in pixel (Zhang et al. [2009]) k: Given λ an integer, the maximum probability is obtained for $t = \lambda$ and the variance of the distribution is also λ ; at a pixel x, the equation is given by: $P_{\lambda_x}(t) = \frac{\lambda^t e^{-\lambda}}{t!}$. Poisson noise appears from quantum effects of photons. The Poisson noise estimate is the square root of the number of detected photons, then the Cameras can be calibrated to know how many photons a certain pixel value is (Akiyama et al. [2015]).

As shown in Fig. 4.17, except SOAGK and L/L, all the other filters are robust to Poisson noise at this scale. SF_2 performed exceptionally well, with True Positive value $TP/|G_t|$ (and F_α) almost close to 1. It only started to drop from SNR 5dB. Filters with Z, R, G_σ & SF_4 ranked in the second place, their performances are still sufficiently well (TP / F_α over 0.9). But SF_4 seemed to be less robust to Poisson noise, whose TP dropped sharply from SNR 6dB. In comparison, SOAGK showed its relatively poor performance to resist the noise – starting with TP 0.85, ending with TP 0.3 at SNR 3dB. L/L failed completely in this task. It detected barely any true positive ridges. This poor performance of oriented filters is caused by the small size of these filters, where small-scale orientation deforms the kernels. The angular score E is the best for Z and R filters (using H) because they correspond to sharp filters, especially suitable for thin ridges.

4.3.3 Synthetic images corrupted by Speckle noise

The second image in Fig. 4.18 contains ridges of width 3 pixels and is disturbed by a Speckle noise. This multiplicative noise appears with the image acquisition due to the level of noise in the sensor of a CCD or CMOS camera, increasing in proportion to lumi-

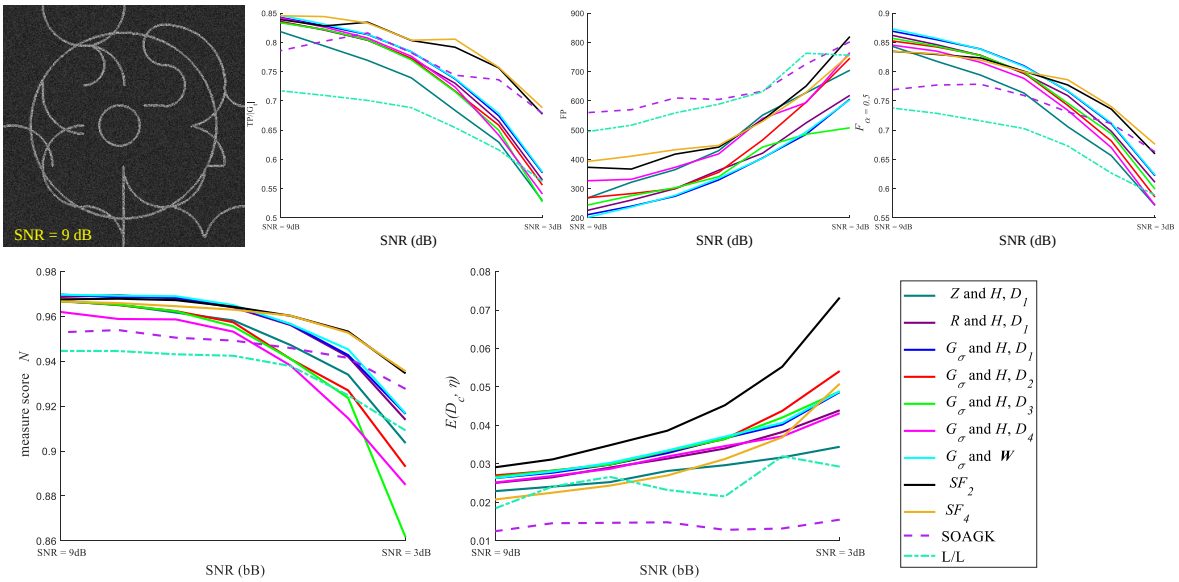


Figure 4.18: Evaluation of the different ridge/valley extraction techniques on synthetic images corrupted by Speckle noise.

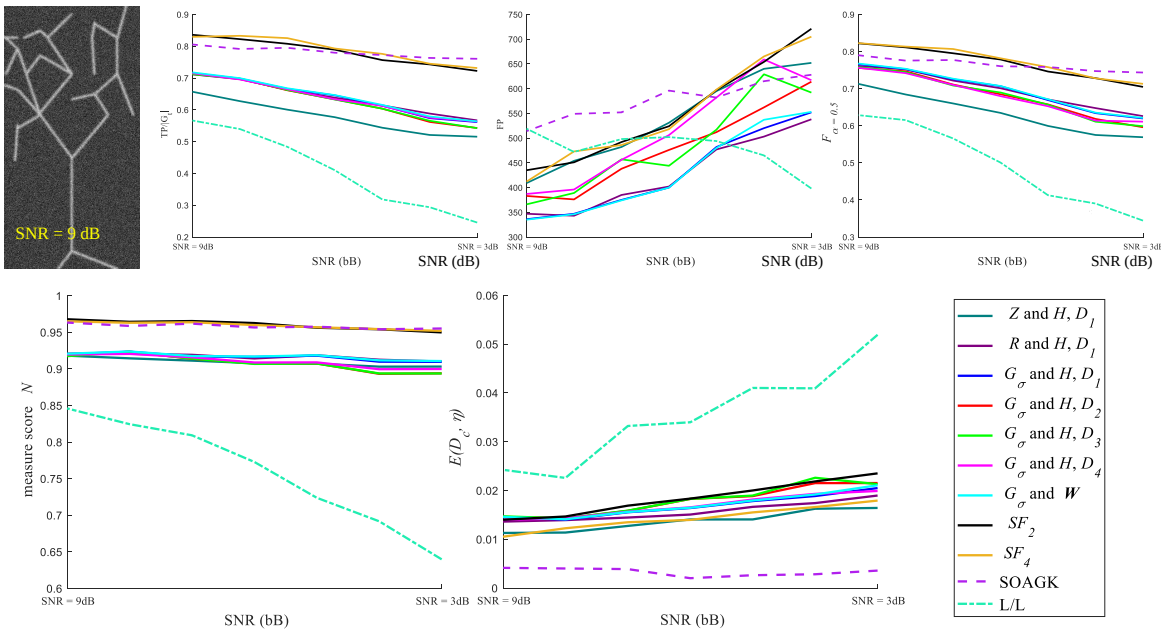


Figure 4.19: Evaluation of the different ridge/valley extraction techniques on synthetic images corrupted by Gaussian noise and Gaussian blur.

nosity (Laligant et al. [2013]). This noise model can be formulated as: $J = I + \sigma \cdot \eta \cdot I$, where J represents the observed image, I the noise-free image, η is a normalized Gaussian noise distribution centered at 0 of standard deviation σ.

Compared to Poisson noise, which is correlated to the original image, Speckle noise adds some independent noise to the images that could corrupt more the image's geometric structure. For filters providing quite good results in previous situation (Fig. 4.17), they are less efficient in case of Speckle noise. Instead of starting with TP in the range of [0.93, 1], they are now under the threshold of 0.85 corresponding a performance drop of 10% at SNR 9dB. And the robustness to noise level decreased much more. This decreasing behavior in the interval [0.85, 0.55] is similar to Speckle noise's granular effect property.

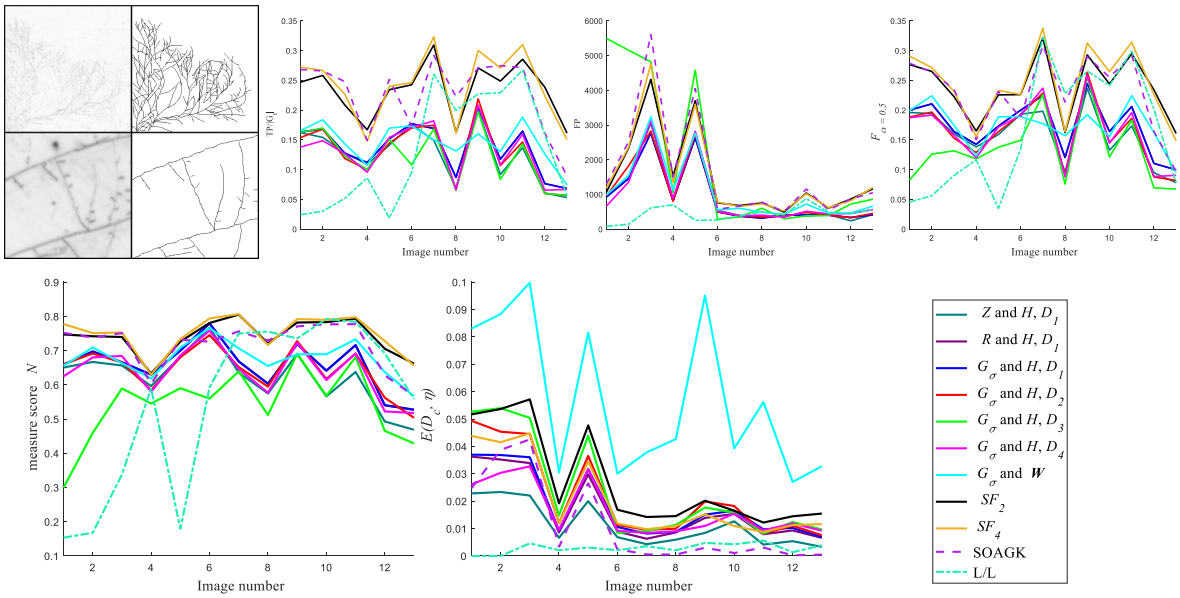


Figure 4.20: Evaluation of the different ridge/valley extraction techniques on real images.

At SNR 3dB, the TP are around 0.55, meaning only about 55% ridge pixels are perfectly detected. Unlike the total failure with Poisson noise, L/L filter worked correctly. However, its performance is still the worst compared to other filters. The main reason could be due to the fact that L/L is by definition a 1D filter with additional processing as the endline or the stabilizer, which are noise sensitive. This makes it to be much less robust to structure-correlated noise. As the structure-correlated noise could destroy the two-dimensional visual structures transformed in 1D filter space and cause thus the failure of detection. On the other hand, among the techniques using non-oriented filters, it is noticeable that $\mathscr{H}$ with D_1 and **W** with D_1 obtain best scores. Additionally, the extracted ridges are more continuous and less disturbed by undesirable FP pixels. **W** with D_1 allows a better quality of detected ridges than with other non-oriented filters.

Finally, the angular score E obtained by the SOAGK is less penalized because it corresponds to an elongated kernel applied on close-right structures, so the oriented filter is generally the same along these structures (same remark for the Gaussian noise).

4.3.4 Synthetic images corrupted by Gaussian noise

The last experiment with synthetic images in Fig. 4.19 concerns valleys corrupted by a Gaussian blur and Gaussian noise. This type of noise represents an additive noise, disturbing gray values in images. Its model is essentially designed and characterized by its Probability Distribution Function (PDF) or histogram normalization with reference to the gray value: $P(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{(x-\mu)^2}{2\sigma^2}}$ where x is the gray value, μ the mean, and σ the standard deviation. Gaussian noise blurred more the geometrical structure in images. So, as shown in Fig. 4.19, the general TP for all filters are decreased compared to those with Poisson and Speckle noises. SOAGK, SF_2, SF_4 filters gave better results (TP $\sim$ 0.8). L/L filter always showed the worst result, even at SNR 9dB, the TP is only 0.57 when noise becomes stronger performances decreases. In comparison, **W** with D_1 still detects a better quality of ridges than with other non-oriented filters, statistically and visually.

4.3.5 Evaluation with real images

After evaluating the filters on synthetic images with different types of noise, the ridge detection on real-world images is presented. These images are from the Ghent University Fungal Images together with their manually annotated ground-truth ridges (Lopez-Molina et al. [2015]). This database is extremely challenging. Here, 13 images with their tied ground truth images are selected randomly for this experiment. The images have very poor contrast and strong noises, see Fig. 4.20 Regarding the evaluation pixel per pixel, due to the hand-labeled ridge points which create inaccurate ground truth (G_t), the overall ridge detection with these filters is around TP 0.2, and they are image-dependent. In the best situation, TP can reach 0.3; otherwise, in worst cases, the TP will drop below 0.5 and are close to 0.06. Oriented filter SF_2, SF_4 and SOAGK performs well, regarding F_α and N, contrary to the L/L and H with D_4. Regarding **W** with D_1, its evaluation is better than other non-oriented filters, even though the angle evaluation E penalizes the directions perpendicular to the detected ridges (however the score remains under 0.1 where it was under 0.2 for Speckle noise).

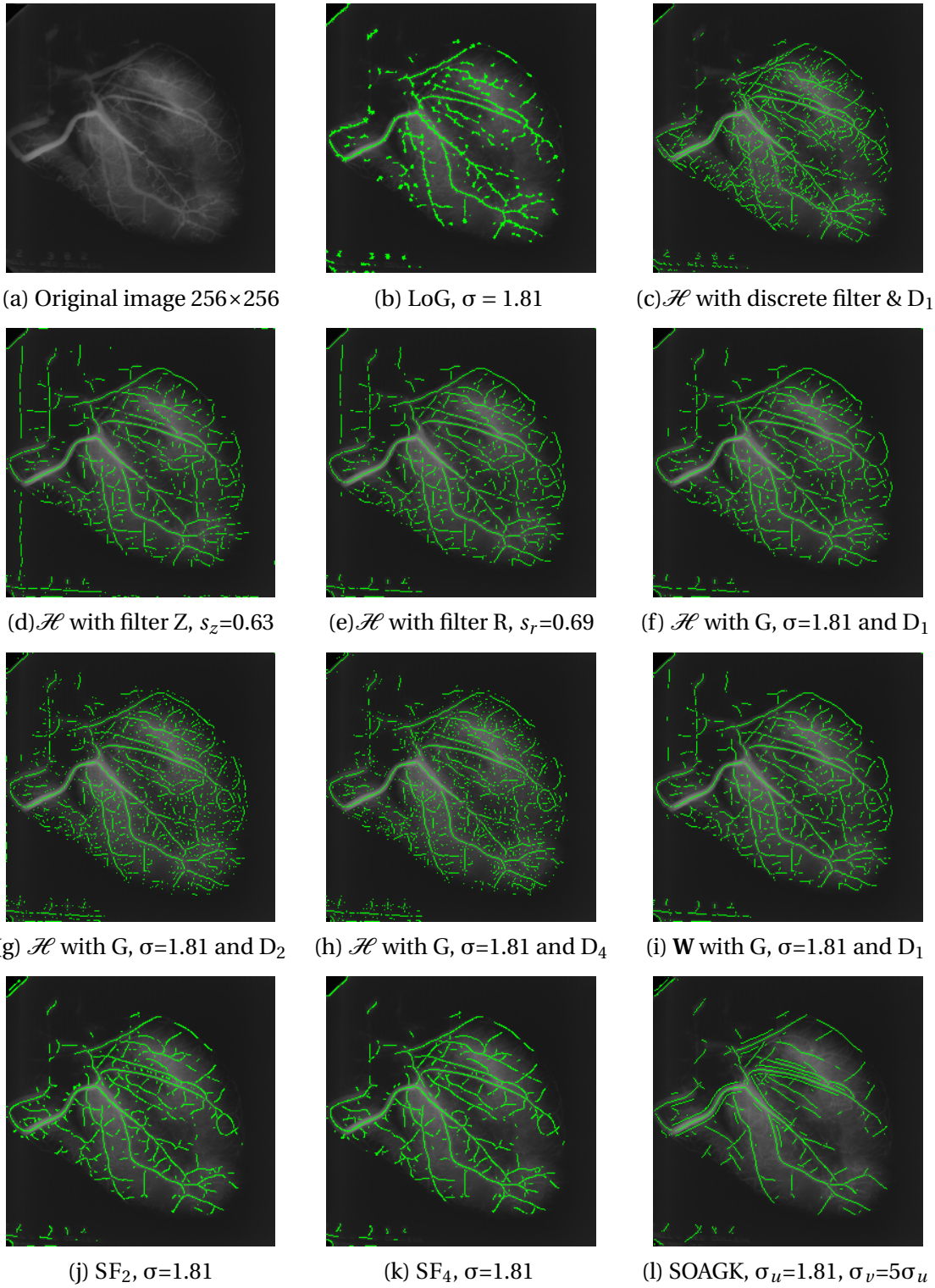
4.3.6 Visual comparison on real images

The experiment on real images have been performed on heart images for cartographic analysis and aerial noisy image which are available in Fig. 4.21 and 4.22 respectively.

For the first image presented in Fig. 4.21(a), the output filtered images demonstrate interesting results with all filtering techniques reviewed in this literature non-maxima suppression and thresholded images. Here, the thresholded images correspond to the 3500 highest computed points for each filtering techniques. The original image is not specifically noisy, but it is blurred. The selected parameters correspond to the detection of ridges of width 3 pixels (see Table 4.1). Usually, the blood vessels are well extracted, even with the Laplacian, even though ridges are roughly extracted, see Fig. 4.21(b). The discrete filter obtains the worst result with some false positive points and extracted ridges which are doubled. Regarding the Hessain matrix, Z, R and G_σ perform well using D_1 with the extraction of main blood vessels without many false positives. Same remark, the Weingarten is reliable contrary to the Hessian matrix with G_σ and D_2/D_4 which are corrupted by many false positive pixels and noncontinuous extracted ridges. Now, considering oriented filters in Figs. 4.21(j)-(l), SF_2 performs as well as Z, R and G_σ filters using D_1 while SF_4 obtains the stronger result with continuous extracted ridges without many undesirable extracted pixels. Elsewhere, the SOAGK in Fig. 4.21(l) performs for elongated ridges but too thin blood vessel are not well extracted, while some extracted ridges are tripled, penalizing this ridge detector.

Likewise, the same implementation procedure has been applied on Stanwick aerial image in Fig. 4.22(a) for aerial image analysis, where the original image is corrupted by a strong noise. Because the images contain too many ridges to be detected, the thresholded images after filtering correspond to 50% of the highest extracted points after non-maxima suppression (excepted for the discrete filter).

The selected parameters correspond to the detection of ridges of width 5 pixels (see Table 4.1). The extracted information in Fig. 4.22(b)-(l), clearly shows that each filtering technique detects different level of information after non-maximum suppression, as subject to different application analysis. For instance, the LoG (Fig. 4.22(b)) detects roughly disconnected contours as of blob like structure, and Hessian with discrete filter (Fig. 4.22(c)) has detected most of the discernible details, but the majority of detected ridges are misconnected. The Z and R filters in Figs. 4.22 (d), and (e) have extracted similar roughly disconnected ridges, which cannot be enhanced even with post morphological processing. The Hessian matrix ($\mathscr{H}$) using Gaussian and D_1 in Fig. 4.22 (f) shows more interesting result with extracting sufficiently ridges mostly connected without need of any further processing even though it contains many false positive points. The results obtained by $\mathscr{H}$ and D_2/D_4 in Figs. 4.22(g)-(h) show more ridges with sharp noises. The Weingarten in Fig. 4.22(i) displays less information than $\mathscr{H}$ with D_1, it is similar to Z and R. Finally, detected ridges extracted by oriented filters in Figs. 4.22(j)-(l) demonstrate quite interesting and directly usable information for its application analysis, in particular the SOAGK for the straight ridges.



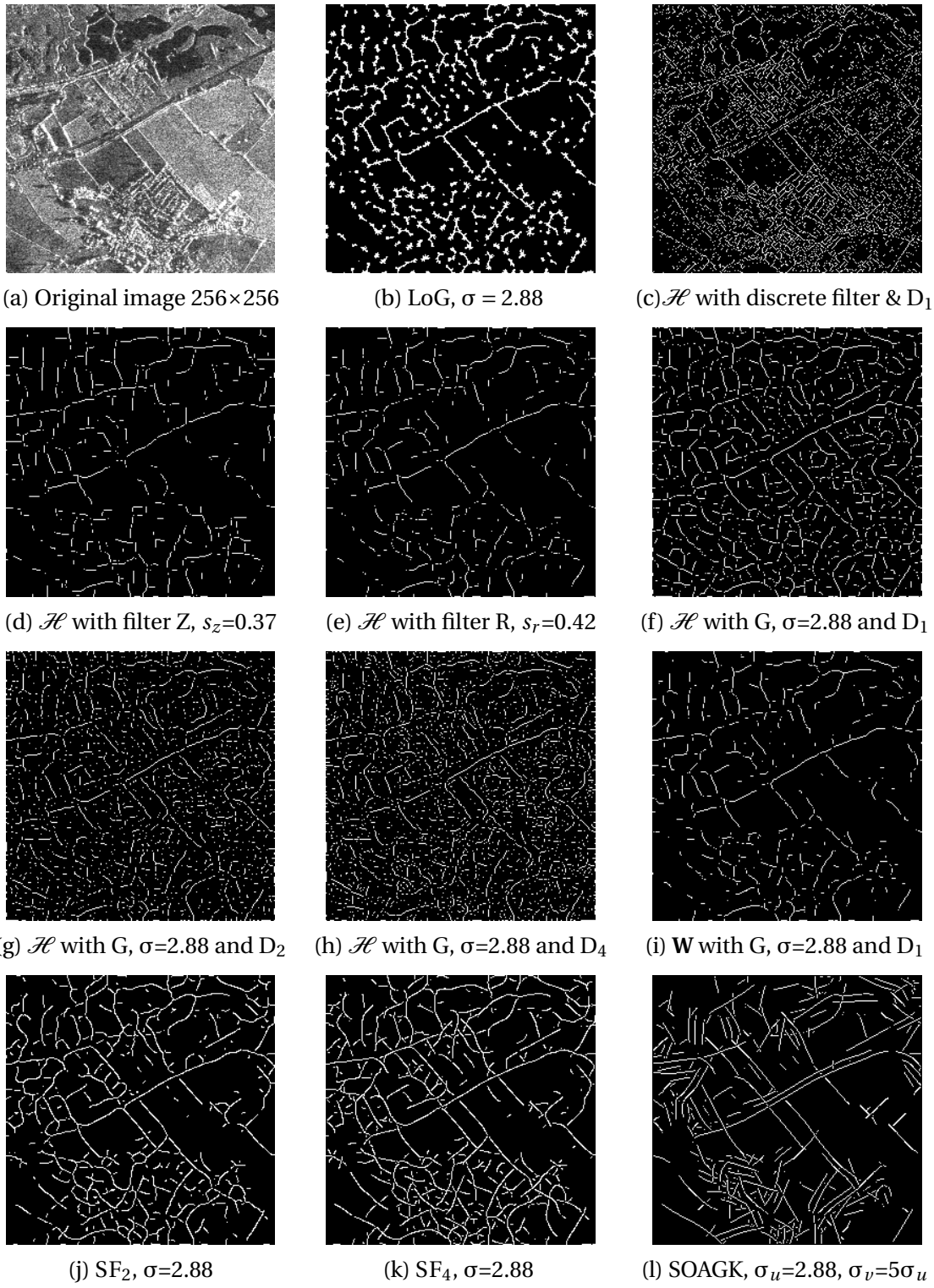
(a) Original image 256×256 (b) LoG, σ = 1.81 (c) $\mathscr{H}$ with discrete filter & D_1

(d) $\mathscr{H}$ with filter Z, s_z=0.63 (e) $\mathscr{H}$ with filter R, s_r=0.69 (f) $\mathscr{H}$ with G, σ=1.81 and D_1

(g) $\mathscr{H}$ with G, σ=1.81 and D_2 (h) $\mathscr{H}$ with G, σ=1.81 and D_4 (i) **W** with G, σ=1.81 and D_1

(j) SF_2, σ=1.81 (k) SF_4, σ=1.81 (l) SOAGK, σ_u=1.81, σ_v=5σ_u

Figure 4.21: Ridge detection on heart image. Detected ridges are displayed on the original image. Thresholded images correpond to the 3.500 highest computed points for each method.



(a) Original image 256×256 (b) LoG, σ = 2.88 (c) $\mathscr{H}$ with discrete filter & D_1

(d) $\mathscr{H}$ with filter Z, s_z=0.37 (e) $\mathscr{H}$ with filter R, s_r=0.42 (f) $\mathscr{H}$ with G, σ=2.88 and D_1

(g) $\mathscr{H}$ with G, σ=2.88 and D_2 (h) $\mathscr{H}$ with G, σ=2.88 and D_4 (i) **W** with G, σ=2.88 and D_1

(j) SF_2, σ=2.88 (k) SF_4, σ=2.88 (l) SOAGK, σ_u=2.88, σ_v=5σ_u

Figure 4.22: Ridge detection on Stanwick aerial image. Thresholded images correpond to 50% of the positive points after non-maxima suppression step, excepted for (c) where the 7.500 highest points are reported because it contains too many positive pixels.

4.4 Summary

This chapter presents an extensive evaluation and comparison of ridge/valley detection with image-based filtering techniques including, the ridge/valley mathematical properties, driving filter parameters and characterizations. Classical technique as Difference Of Low Pass (DOLP) and Laplacian filter are detailed, as the elegant way using the Hessian matrix $\mathscr{H}$. Different techniques exist to compute ridge or valley extraction with the eigenvalues of $\mathscr{H}$, they are detailed through this communication, as for the Weingarten. Three low pass filter are compared, namely Z, R which are exponential filters and the Gaussian G_σ. Elsewhere, method using oriented Gaussian filters are also reported. The evaluation and comparison of filtering techniques have been performed both theoretically and experimentally on synthetic and real images.

Each filtering techniques have been examined on complex images, where different types of noises have been applied. The acquired comparison and evaluation graphs exhibited which method is reliable as a function of the width feature and the noise type.

Regarding non-oriented filters, the Z filter performs well when the ridge or the valley are very thin (width of one pixel) and requires the less computational complexity computed the Hessian matrix $\mathscr{H}$. On the other hand, $\mathscr{H}$ associated with the Gaussian G_σ and the highest eigenvalue (D_1) is a good compromise when the feature widths are growing. Yet, the Weingarten $\mathbf{W}$ and its eigenvalue gives suitable and better continuous detected ridges. Steerable filters of order 2 (SF_2) and of order 4 (SF_4) obtain similar results, they are particularly reliable for images corrupted by noise, especially for bent features (SF_4 is a little more reliable), contrary to the SOAGK which is well adapted for straight features.

Eventually, this study would serve as ridge/valley optimal parameter configuration and adjustment guide for its interested applied researchers and application tools and domain such as satellite or aerial image analysis (road, river, etc.), medical image analysis (blood vessels, filaments, nerve system, etc.), lines detection, image segmentation, and object detection.

Chapter 5

Multiscale Line Detection Using Second Order Semi-Gaussian Filtering Technique

Contents

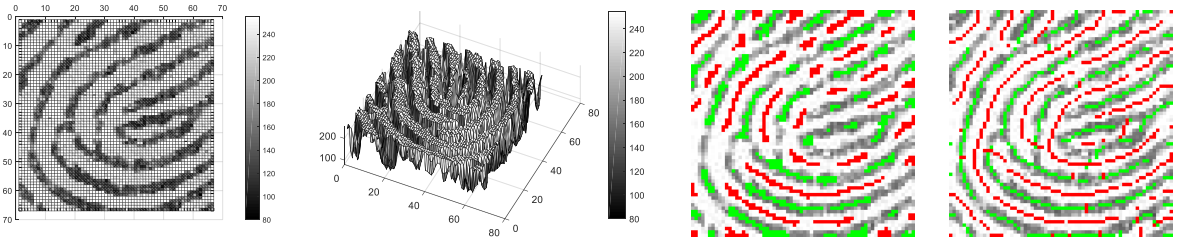
5.1 Introduction

Line features represent ridges and valleys in a digital image, they correspond to thin, elongated structures and ridges refer to the valleys of the inverted relief on the image surface, as illustrated in Fig. 5.1(b). Line feature characterization is the initial step in all aforementioned applications.

Amongst the low level image structures, such as, texture, edge, corner or junction, line features is the widely applied structure in the image processing literature. Line structures on multiscale like ridges or valleys contain determinative information required in image analysis problems, such as, scene understanding, photogrammetry, biomedical (Sanchez et al. [2019]) and remote sensing.

It is important to have a reliable line detector, especially adapted to different scales. Today, there are many post-processing methods to align segments, group or recognize shapes. These methods are all more effective when the line detector is reliable.

Two parallel step edges construct a line structure, they can be roughly extracted by the Laplacian operator, as in Fig. 5.1(c). Technically, the step or ramp edges correspond to local maxima of the first order derivative (Canny [1986]), while ridges are tied to local maxima of the second order derivative in the local analysis of Hessian matrices of an



(a) Gray level image of size 67×67 coded on 8 bits.

(b) Image surface of (a), ridges are at the top whereas valleys are at the bottom of the surface.

(c) Ridges/valleys extracted by Laplacian of Gaussian.

(d) Ridges/valleys extracted with the Hessian matrix.

Figure 5.1: 3D representation of ridges/valleys and their extractions using the Laplacian of Gaussian and Hessian matrix(Steger [1998]), σ=2.88, see Shokouh et al. [2021b]. In (c)-(d), extracted ridges in red and valleys in green are superimposed on the original image.

image (Lindeberg [1998] Steger [1998]). There exist other techniques to extract line structures, including oriented filters, as presented in the next section.

In this chapter, a multiscale Second Derivative of Semi-Gaussian (SDSG) filtering technique is proposed. The line feature profiles can vary across scale space; they are detected with different filter parameters, whose outputs would be merged later to create a single edge map. Hence, a function is proposed for the scale fusion considering the desired scale.

5.2 Multiscale ridge extraction: Related works

Gaussian kernels as well as their derivatives are the widely employed filtering techniques for the processing of low level image structures due to their isotropy, steerability and decomposability properties. The zeroth order Gaussian kernels are used for smoothing and regularization. This section covers the main theoretical principles of multiscale line-feature extraction in digital images.

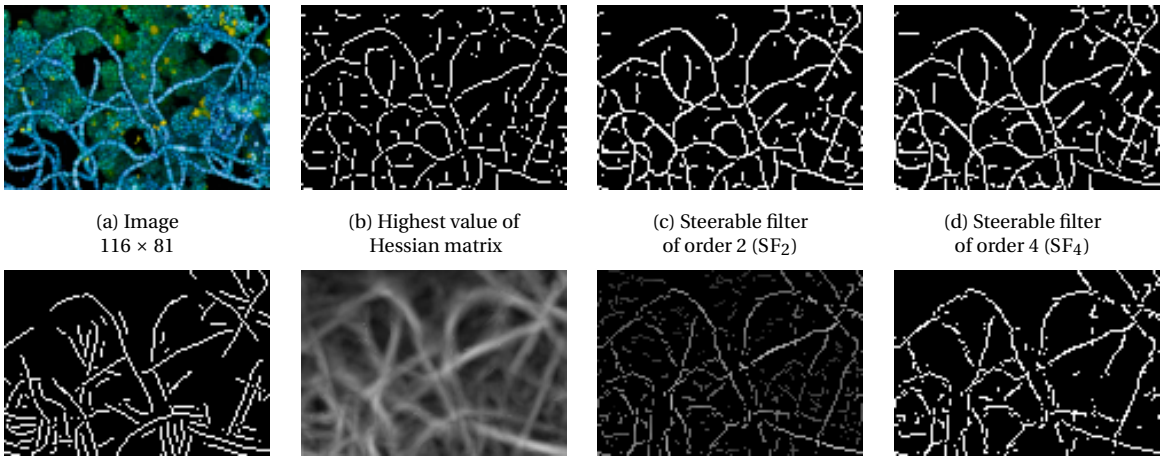
5.2.1 Isotropic filters

For the line-structure detection, several works are based on the eigen-decomposition of the Hessian computed at each image pixel (Lindeberg [1998]Shokouh et al. [2021b]Steger [1998]). The combination of the eigenvalues measures the overall strength of the ridge or the valley, as illustrated in Figs. 5.1(d) and 5.2(b). In scale space, theoretically, a pioneer work proposed by Lindeberg assumes that a ridge point is defined as a location for which the intensity assumes a local maximum (or minimum for the valleys) in the main principal curvature direction (Lindeberg [1998]). Considering an image I_σ smoothed by a Gaussian of standard deviation σ, the line-structure measure of the original image I is given by:

$$\mathscr{N}_\gamma(I) = \sigma^{2\gamma} \cdot \left(\left(I_{\sigma,xx} - I_{\sigma,yy} \right)^2 + 4 \cdot I_{\sigma,xy} \right), \tag{5.1}$$

where $I_{\sigma,xx}$ and $I_{\sigma,yy}$ represent the x and y derivatives of the image I_σ respectively, and $\gamma > 0$ is termed as the scale normalization factor.

Bae et al. [2015] extended the γ-normalized multiscale Hessian matrix of Eq. 5.1 to derive a width-invariant and contrast-proportional second derivative magnitude map. Then a high-level processing is performed for segment formation.



(a) Image 116 × 81 (b) Highest value of Hessian matrix (c) Steerable filter of order 2 (SF_2) (d) Steerable filter of order 4 (SF_4)

(e) SOAGK, $\Delta\theta = 5°$ $\sigma_u = 1.81$, $\sigma_v = 5\sigma_u$ (f) Linemap of SDSG, $\sigma_u = 1.81$, $\sigma_v = 5\sigma_u$ (g) Non-maxima suppression of (f) (h) Thresholding of image in (g)

Figure 5.2: Extraction of ridges corresponding of long chains of streptococcus pyogenes infecting grape-like clusters of MRSA biofilm: comparison of mono-scale ridges extractors. Original image source: https://www.nikonsmallworld.com

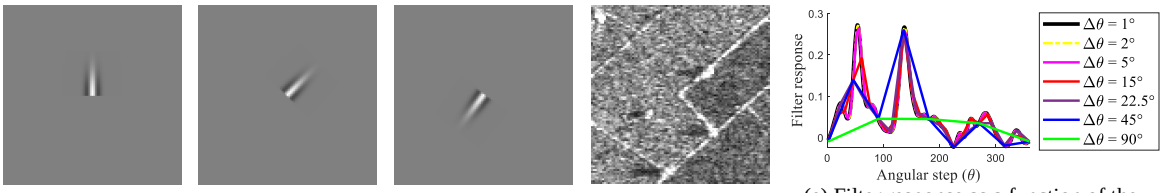
5.2.2 Oriented filters for line feature detection

The well-known and popular steerable filters (Freeman and Adelson [1991]Jacob and Unser [2004]) are built by linear combination of the direct rotation of the derivatives of the basic isotropic Gaussian. Thereafter, it captures the line structure energy in the direction of the maximum response of the filter.

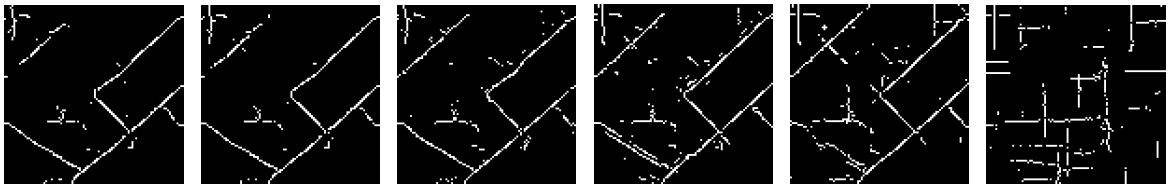
Edge detection techniques using elongated kernels are efficient to correctly detect large linear structures (Perona [1992]Jacob and Unser [2004]Lopez-Molina et al. [2015]). The robustness against noise depends strongly on the smoothing parameters of the filter, i.e., the parameter of the filter elongation. Moreover, the elongated filters enable us to capture discontinuous line features, as illustrated in Figs. 5.2(d)-(e). To extract ridges, the Second-Order Anisotropic Gaussian Kernel (SOAGK) can be applied (Lopez-Molina et al. [2015]). Considering the vertical anisotropic Gaussian directed at $\theta = 0°$, its second derivative in the x direction is:

$$\mathscr{G}''_{\sigma_u,\sigma_v,\theta=0}(x,y) = \frac{x^2-\sigma_u^3}{2\pi\sigma_u^5\sigma_v}\cdot e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_u^2}+\frac{y^2}{\sigma_v^2}\right)}. \tag{5.2}$$

The choice of $\sigma_v > \sigma_u$ enables to build a narrow filter smoothing mostly in the y direction while enhancing valleys in the x direction. Now, this 2D kernel can be oriented in different directions to capture line structures in the image, see Fig. 5.2(e). To this end, this anisotropic choice produces a smoothing alongside the ridge/valley, which helps to extract easily elongated features, even disturbed by noise. On the contrary, kernels having parameters $\sigma_v \approx \sigma_u$ are equivalent to a Steerable Filter of order 2 (SF_2, presented in Freeman and Adelson [1991]) and may highlight undesirable features as noise which are interpreted as small, non-elongated ridges (Lopez-Molina et al. [2015]). Finally, SOAGK at different scales are applied in Lopez-Molina et al. [2015] to detect the line structures and the combination is done by means of the maximum among the different obtained line-maps.



(a) $\theta = 0°$ (b) $\theta = 45°$ (c) $\theta = 217°$ (d) Image 96×96 (e) Filter response as a function of the angular sampling for the central pixel in (d)



(f) $\Delta\theta = 2°$ (g) $\Delta\theta = 5°$ (h) $\Delta\theta = 10°$ (i) $\Delta\theta = 22.5°$ (j) $\Delta\theta = 45°$ (k) $\Delta\theta = 90°$

Figure 5.3: Ridge detection as a function of the angular step Δθ, with $\sigma_d = 1.8$, $\sigma_s = 5\sigma_d$, and SDSG responses (clockwise) at a pixel tied to a bent ridge.

5.3 Second-Derivative of a Semi-Gaussian Filter (SDSG)

The basic idea of the developed filter is to consider paths (i.e., ridges or valleys) crossing each pixel. Inspired by Montesinos and Magnier [2010], the proposed technique Second-Derivative of a Semi-Gaussian Filter (SDSG) represents a truncated 2nd derivative of an anisotropic Gaussian which can be steered. It's able to detect bent ridges due to two elongated and oriented filters in two different directions.

5.3.1 Concept of the SDSG

The main idea of SDSG is to "*cut*" the second order anisotropic Gaussian kernel (Eq. 5.2) using a Heaviside function and, then, steer this filter in all directions around the considered pixel: from 0 to 360°. Hence, the SDSG can be built by combining a vertical semi-Gaussian on the one hand and its horizontal second derivative on the other hand. Mathematically, it is defined by:

1. a semi-Gaussian for the smoothing in the y direction (vertically):
   $\mathscr{G}(\sigma_s, t) = \mathrm{H}(t) \cdot \mathbf{e}^{\frac{-t^2}{2\cdot\sigma_s^2}}$, with $\sigma_s \in \mathbb{R}_+^*$, $t \in \mathbb{R}$ and H the Heaviside function,

2. a second derivative of a Gaussian in the x direction (horizontally):
   $\mathscr{G}''(\sigma_d, t) = \frac{t^2-\sigma_d^2}{\sigma_d^4} \cdot \mathbf{e}^{\frac{-t^2}{2\cdot\sigma_d^2}}$, with $\sigma_d \in \mathbb{R}_+^*$ and $t \in \mathbb{R}$.

For signal and image processing, t represents an integer. The Fig. 5.3(a) shows an example of SDSG, constructed with these two functions, respectively, $\mathscr{G}$ at the vertical and $\mathscr{G}''$ at the horizontal. In order to create an anisotropic (elongated) filter, the support of the smoothing half-filter must be greater than the support of the filter containing the derivative, that is to say $\sigma_s > \sigma_d$. Then, to obtain a rotated version of the SDSG, this filter is applied in several directions θ from 0 to 360°. The original rotation is centered in the middle of the basis filter; for a better understanding, the rotation center corresponds to the middle of the image in Fig. 5.3(a), and the SDSG is rotated from this point in Figs. 5.3(b) and (c). Thereafter, the image convolution with the steered filters allows computing derivative information at each desired direction (as shown in Fig. 5.3(e)). Then, the line structure strength $\mathscr{L}$ is calculated using a local directional maximization/minimization:

- addition of the two local maxima regarding ridge detection,
- addition of the two local minima regarding valley detection.

The direction perpendicular to the line structure, called, $\vec{\eta}$ is calculated by the bisector between these two local directions (maxima or minima). Then, the line structures can be extracted with non-maxima suppression (NMS) process by deleting local non-maxima in the $\vec{\eta}$ direction, same strategy as in Canny [1986]. The Fig. 5.3(e) illustrates two local maxima tied to the directions of a bent line (here the $\vec{\eta}$ direction is around 90°direction). When the angular step $\Delta\theta$ is well discretized, such a filtering technique allows computing two precise directions of the line structure. The two directions cannot be correctly calculated when the angular step is too spaced ($\Delta\theta > 5°$, Figs. 5.3(f)-(k)).

In Magnier et al. [2014], an anisotropic directional filter is implemented considering the difference of two halves rotating Gaussian filters (DoG mechanism), which can approximate the SDSG filter. Meanwhile, a multiscale approach has been developed by selecting the maximum response among the scales (Magnier et al. [2014]). Such a normalization function may miss some thin objects, as illustrated in Fig. 5.5(c)-(d) and next subsection.

5.3.2 Scale fusion of the SDSG

In one dimension, the σ of the Gaussian derivatives depends on the line width of the structures to be detected, as shown in Fig. 5.4(a)-(b). The SDSG corresponds to a semi-filter. It can be seen as a scan of the projected pixels in all the directions around the considered pixel, illustrated by the signals in Fig. 5.3(e). When the SDSG is steered in the line feature direction, the σ_s parameter allows an elongated smoothing in the line direction, whereas the σ_d (tied to the 2nd derivative which is perpendicular to the line) captures the line structure strength. For multiscale line structure detection, the maximum value among the different filter responses can be selected (Magnier et al. [2014]), as in Fig. 5.4(b). However, it may not be sufficient, especially in real conditions. Consequently, we propose the following improved scale function:

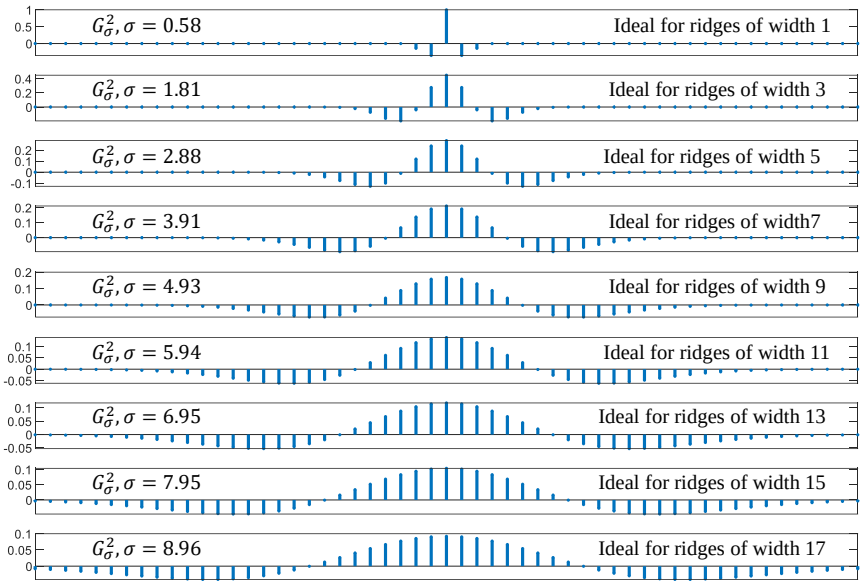
$$\mathscr{F}_\sigma(\mathscr{L}) = \left(\sigma^{\frac{1}{\sigma}} + \frac{1}{\sqrt{\sigma}}\right) \cdot \mathscr{L}, \tag{5.3}$$

where $\mathscr{L}$ represents the filtered image line structure at scale σ.

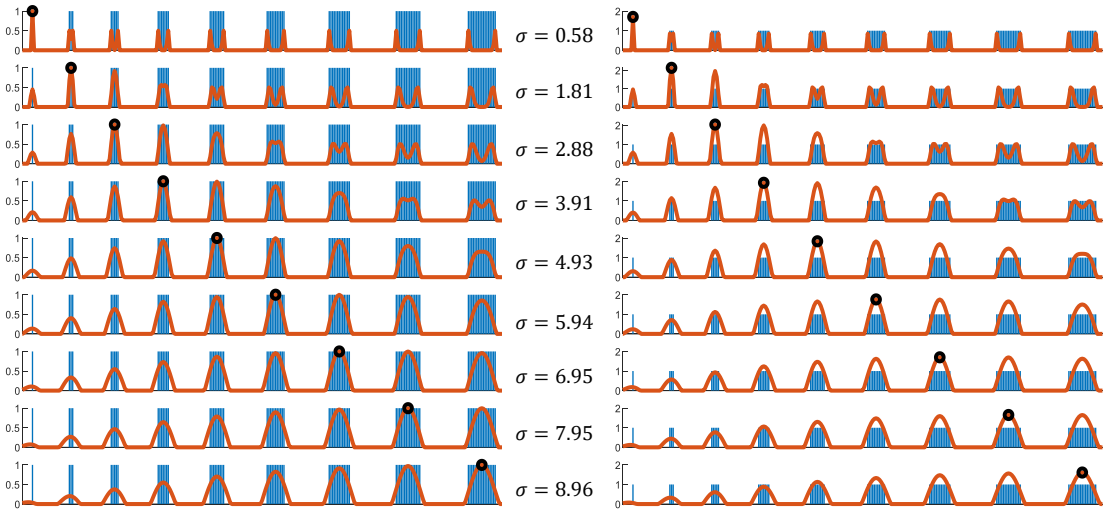
This function allows improving line structure enhancement at the corresponding scale. Its values are always superior to 1, which is efficient for large scales. Thin line features are also well highlighted, as illustrated in Fig. 5.4(c). Finally, the σ_d parameter is considered in the Eq. 5.3, regarding the SDSG filter, see Fig. 5.5(h) where thin and large elongated structures are better connected.

The fusion procedure of the multiscale SDSG can be summarized as follows:
(i) Filtering the image with each possible SDSG at different directions θ and scales (but with same ratio $\frac{\sigma_s}{\sigma_d}$), then compute the line strength $\mathscr{L}$.
(ii) Retaining the strongest response $\mathscr{L}$ after applying the Eq. 5.3 and its tied direction $\vec{\eta}$ for each pixel.
(iii) Suppress non-maxima pixels in the $\vec{\eta}$ direction of the fused image.

The next section presents evaluations and results on real images.



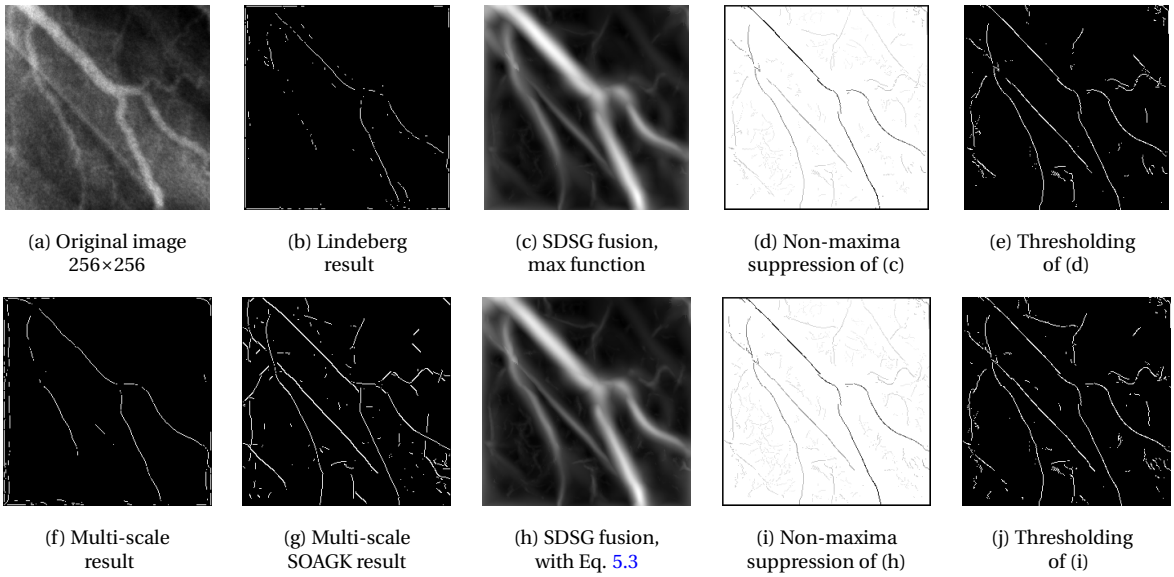
(a) Discrete second derivatives of the Gaussian with different parameters tied to the ridge width



(b) Convolution of the signal with the second derivatives of the Gaussian on the left (without normalization)

(c) Convolution of the signal with the second derivatives of the Gaussian on the left (with normalization)

Figure 5.4: Ridge highlighting in one dimension (1D) by convolution with different second derivative of Gaussian in (a), detailed in Shokouh et al. [2021b]. In (b), ridges are highlighted with the different Gaussian convolutions and (c) takes into account Eq. 5.3. The original signal containing separated ridges of growing widths: 1, 3, 5....17 is displayed by the blue bars in (b)-(c) while the convolved signals are plotted in orange and the maximum of the signal is displayed by the black circles for each scale σ, exhibited between (b) and (c).



(a) Original image 256×256 (b) Lindeberg result (c) SDSG fusion, max function (d) Non-maxima suppression of (c) (e) Thresholding of (d)

(f) Multi-scale result (g) Multi-scale SOAGK result (h) SDSG fusion, with Eq. 5.3 (i) Non-maxima suppression of (h) (j) Thresholding of (i)

Figure 5.5: Blood vessel extraction of different widths by non-maxima suppression (NMS) and thresholding in a Magnetic Resonance Angiography image. In (b), (f) and (g), 20% of the highest pixels are preserved while 30% are preserved for (e) and (j). Note that (d) and (i) are inverted images. Here, scales varied between 1.81 and 6.95, see Shokouh et al. [2021b].

5.4 Experimental results and evaluation

Experiments are carried out on real images. First, a dataset containing fungi images with manually annotated ground truth G_t is used (Lopez-Molina et al. [2015]). To evaluate the line feature detection, the *Normalized Figure of Merit* method (Magnier [2019]) is employed. Let D_c be the detected contour map of an image. Comparing pixel by pixel G_t and D_c, a simple evaluation based on pixel-wise comparison leads to the definition of the following indicators:

- True Positive (TP), common points of G_t and D_c,
- False Positive (FP), spurious detected edges of D_c,
- False Negative (FN), missing boundary points of D_c,
- True Negative (TN), common non-edge points.

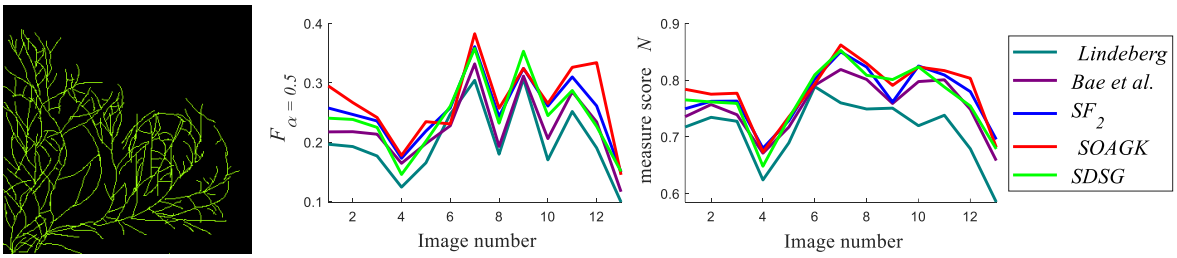
The normalized $\mathcal{N}$ edge detection evaluation measure is, for FN>0 or FP>0:

$$\mathcal{N} = \frac{1}{\mathrm{FP}+\mathrm{FN}} \cdot \left[\frac{\mathrm{FP}}{|\mathrm{D}_c|} \cdot \sum_{\mathrm{P}\in \mathrm{D}_c} \frac{1}{1+\mathrm{K}_{\mathrm{FP}} \cdot d^2_{\mathrm{G}t}(p)} + \frac{\mathrm{FN}}{|\mathrm{G}t|} \cdot \sum_{\mathrm{P}\in \mathrm{G}t} \frac{1}{1+\mathrm{K}_{\mathrm{FN}} \cdot d^2_{\mathrm{D}c}(p)} \right], \tag{5.4}$$

where $(\delta,\kappa)\in]0,1]^2$ represent two scale parameters (Magnier [2019]), $|\cdot|$ denotes the cardinality of a set, and $d_{\mathrm{A}}(p)$ is the minimal Euclidian distance between a pixel p and a set A.

So, if there are no error, i.e., FP=FN=0, then $\mathcal{N}$=1. Therefore, the measure $\mathcal{N}$ calculates a standardized dissimilarity score; the closer the evaluation score is to 1, the more the edge detection is qualified as suitable. On the contrary, a score close to 0 corresponds to a poor detection of contours.

The aim here is to get the best contour map in a supervised way. For that, the line features are extracted after a suppression of the local non-maxima, then a threshold by



(a) Image annotated by hand (b) F_α measure, tied to *TP*, *FP*, and *FN* (c) Edge detection evaluation $\mathcal{N}$ (d) Legend

Figure 5.6: Evaluation of the ridge extraction technique on real images (Fungal images Lopez-Molina et al. [2015]). Detected line structures of image 3 are available in Fig. 5.7(a).

hysteresis is applied to obtain a binary segmentation (Canny [1986]). Theoretically, to be objectively compared, the ideal contour map of a measure must be a D_c at which the supervised evaluation gets the highest score, as detailed in Magnier [2019] and Magnier et al. [2018]. In addition, from proper binary confusion matrix, the precision (P_{rec}) and recall (R_{ec}) evaluations are computed, given the overall quality expressed in terms of the F_α-measure with α=0.5 allowing a equal penalization between FN and FP:

$$F_\alpha = \frac{P_{rec} \cdot R_{ec}}{\alpha P_{rec} + (1-\alpha) R_{ec}} \text{ with } P_{rec} = \frac{TP}{TP+FP} \text{ and } R_{ec} = \frac{TP}{TP+FN}, \tag{5.5}$$

The SDSG filter is compared with 4 other multi-scale feature line detection techniques, namely: Lindeberg [1998], Bae et al. [2015], SF_2 (Freeman and Adelson [1991]) and SOAGK (Lopez-Molina et al. [2015]). Evaluation scores are presented in Figs. 6.6(b)-(c) for F_α and $\mathcal{N}$ measures respectively. Usually, scores achieved by SDSG are similar to those of the SOAGK, showing the reliability of the proposed filter. Both are better than Lindeberg, SF_2 and Bae *et al.* which uses a post processing segment formation.

Visually, detections obtained by the SDSG are close to those derived from the SOAGK, excepted that SOAGK creates many straight (small) segments for isolated points, see Fig. 5.7(a).

In order to interpret the output of the SDSG versus the state of the art techniques for line feature detection and extraction, the comparative tests have been carried out on different real images shown in Fig. 5.7, including (a) fungal, (b) cart wheel picture, (c) satellite image, (d) angiography to detect blood vessels, (e) aerial image, (f) a noisy biomedical image to detect and extract filaments.

Taking into account that the original images are noisy and blurred, the Lindeberg filtering (Lindeberg [1998]) extracted the impure desired lines in spite of non-maxima suppression. The Bae et al. [2015] output is better visually in extracting finer lines with higher precision because of its segment formation created by the high-level processing. Considering the steerable filter (SF_2), it has detected more line features of varied scale. The SOAGK has rather extracted more connected line features, which is considered a strength point in filtering.

The SOAGK in general demonstrates good results, particularly for elongated ridges; but too thin blood vessel as filaments and roads are not well detected, while some extracted lines are tripled and blobs are extracted as lines, penalizing this line detector. The proposed SDSG obviously has demonstrated significant result in case of noise suppression. Indeed, visually desired line features as with less or no post processing need for final output, while looking the original images.

The SDSG filter has extracted desirable line features in general, and in particular in Fig. 5.7 (c) roads (d) blood vessels, (e) large roads and (f) filament without too many

undesirable false positive points. Usually, the extracted lines are more pure using the same thresholding ratio, comparing other techniques.

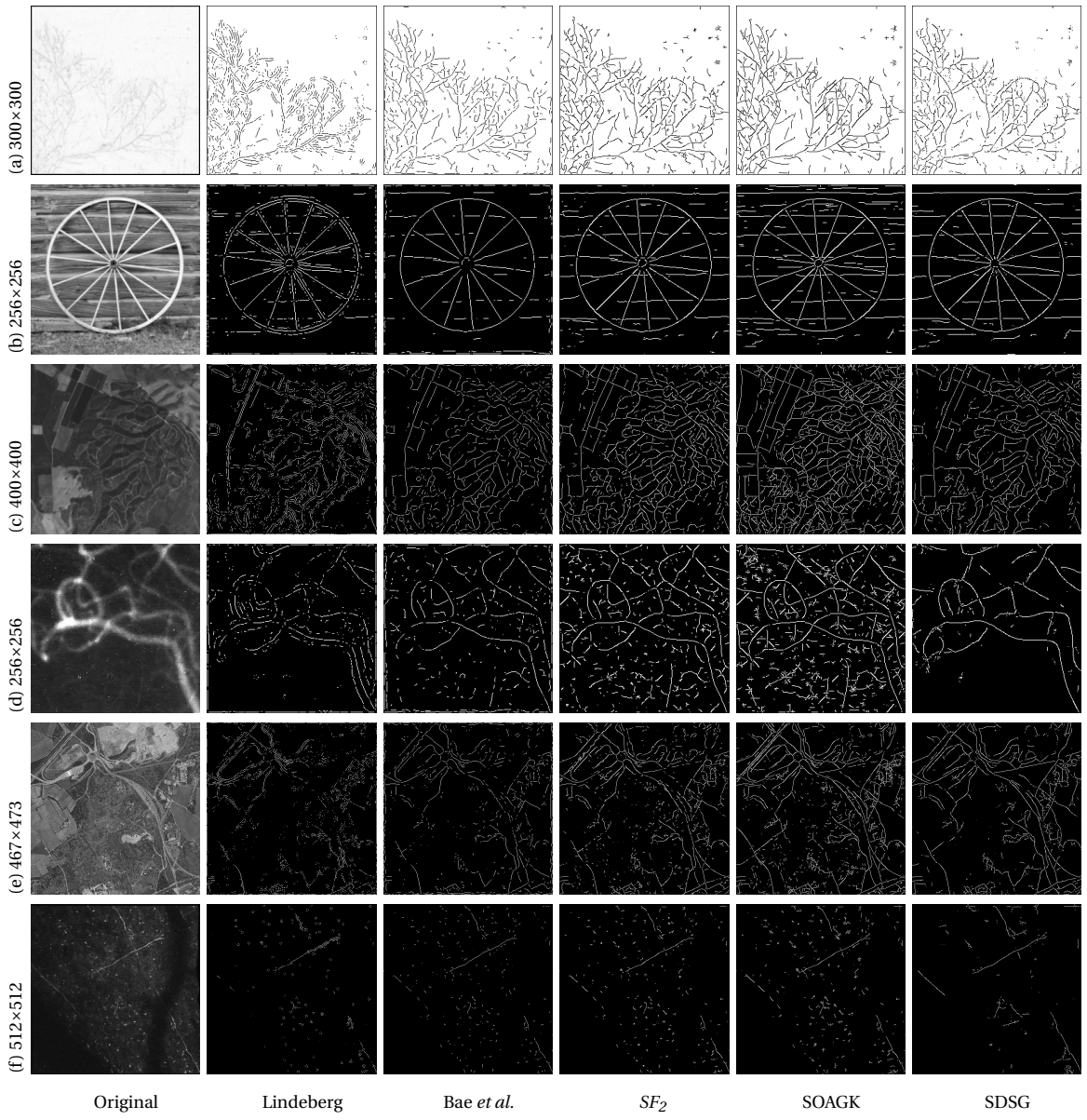


Figure 5.7: Line-feature extraction on real images with multi-scale filtering methods. The images in (a) correspond to the 3rd image in Fig. 6.6 in the evaluation. For the images (b)-(f), detected lines correspond to the same percentage of highest pixels after NMS per method, respectively (b) 60%, (c) 50%, (d) 40%, (e) 35%, (f) 5%.

5.5 Summary

In this chapter a multi-scale filtering approach for line feature detection has been proposed. The proposed approach can be adapted to noisy environments, and is also reliable to detect line feature with heterogeneous types, widths, and prominence. An optimal scale selection function for multi-scale processing is the main contribution of this approach.

This approach has been compared to different types of multi-scale filtering methods, including isotropic (using the Hessian matrix) and oriented filters (isotropic or anisotropic).

Quantitative and qualitative experiments regarding real images of different types and scales have shown the optimal efficiency and very promising results of the SDSG technique compared with the three major techniques of the state of the art.

Future work of this contribution will examine the fusion of line feature detection with different ratios of $\frac{\sigma_s}{\sigma_d}$ (described in Sec. 5.3.1) which will add another dimension to our model. Further evaluations could involve the scales of the detected features, not only the positioning of the detection, as assessed here with F_α measure. The multiscale responses can also serve as input layer of neural networks in biomedical applications to improve the contrast between line features and background, as explored in Sanchez et al. [2019]. SDSG could bring more improvement in this type of applications.

Chapter 6

2DSBG: A 2D Semi Bi-Gaussian Filter Adapted For Adjacent and Multiscale Line Detection

Contents

6.1 Introduction

The detection of features in images is a computationally intensive process and remains a primary step in many low-level computer vision tasks. Linear structures (ridges, edges, etc.) are widely used features in various computer vision applications. To detect these linear structures, numerous filtering approaches are implemented. The optimal approach is chosen based on how to retain the maximum amount of desired information, whilst removing the noise to obtain an optimal segmentation result depending on the application.

To that end, linear structure detection techniques require the analysis of the first or second order derivative of the images, which is obtained by filtering the image using a kernel convolution (Ziou and Tabbone [1998]; Papari and Petkov [2011]). The derivative's order, the type of kernel, and their related parameters control which feature type is aimed to be detected. Gaussian kernels are the most popular and widely used filtering techniques due to their useful isotropy, steerability and decomposability properties related to the implementation of integration and differentiation in images . The zeroth order Gaussian kernels G are used for regularization (Canny [1986]):

$$G(\sigma, x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-x^2/2\sigma^2}, \text{ with } \sigma \in \mathbb{R}_+^*, \; x \in \mathbb{R}, \tag{6.1}$$

where σ represents the standard deviation. The first and second order Gaussian kernels are commonly used for linear structure detection. However, these kernels are prone

(a) SF_2 (b) SOAGK (c) SDSG (d) 2DSBG , $\rho = 0.5$

Figure 6.1: 2D discrete filters of second order steered at 0° and 20°. For the derivative part G″ or BG″ in (d), $\sigma = 3.91$ whereas for (b)-(d), the anisotropic parameter $\sigma_s = 5 \cdot \sigma$.

to noise, relative to both the derivative order and the σ parameter. Gaussian multiscale is the primary method in multiscale feature detection. These techniques are mostly based on the eigen-decomposition of the Hessian matrix (Steger [1998]; Lindeberg [1998]; Tremblais et al. [2007]; Shokouh et al. [2021b]).

Meanwhile, steerable filters of second order (SF_2) are an elegant technique to capture ridge information; they are generated by the linear combination of basis filters such as the second derivative of the Gaussian G″ in one dimension (1D):

$$G''(\sigma, x) = \frac{x^2 - \sigma^2}{\sigma^4} \cdot \mathbf{e}^{\frac{-x^2}{2\cdot\sigma^2}}, \quad \text{with } \sigma \in \mathbb{R}_+^*,\ x \in \mathbb{R}. \tag{6.2}$$

To improve the precision of the detection, elongated oriented filters were designed in terms of a better compromise between noise rejection and localization accuracy (Perona [1992]; Arbelaez et al. [2010]; Lopez-Molina et al. [2015]). Then, to extract line feature, the Second-Order Anisotropic Gaussian Kernel (SOAGK) can be applied in two dimensions (2D):

$$\text{SOAGK}(\sigma, \sigma_s, x, y) = \frac{x^2 - \sigma^3}{2\pi\sigma^5\sigma_s} \mathrm{e}^{-\frac{1}{2}\left(\frac{x^2}{\sigma^2} + \frac{y^2}{\sigma_s^2}\right)}, \quad \text{with } (\sigma, \sigma_s) \in \mathbb{R}_+^* \times \mathbb{R}_+^*. \tag{6.3}$$

The choice of $\sigma_s > \sigma$ enables to build a narrow filter, then the filter is oriented to extract ridges, as illustrated in Fig. 6.1(b).

The second order Gaussian kernels are even and the filter's coefficients distant from the filter center have opposite signs; they correspond to the filter support, which unfavorably gets enlarged when the σ parameter is growing. Indeed, this enlargement yields robustness against noise but on the other hand exposes the defect of the second order Gaussian filtering. Therefore, an empirical trade-off when adjusting the parameter's configuration is unavoidable in conventional manner. In particular, adjacent linear structures of different widths cannot be accurately extracted with this type of support.

The proposed solution consists in seizing the properties of both a precisely orientable filter (Semi-Gaussian) and an adjustable filter under certain conditions of adjacent linear structures. As a result, a new filter is generated, named 2D-Semi-Bi-Gaussian filter (2DSBG). The 2DSBG rigorously minimizes the interference of adjacent line features while retaining adequate line feature information thanks to the fine-tuning of σ and the scale ratio parameter, as detailed below.

6.2 Proposed approach: 2DSBG

The objective is to detect several line features in an image at different unknown positions and orientations. The detection procedure is formulated as a rotational matched filtering.

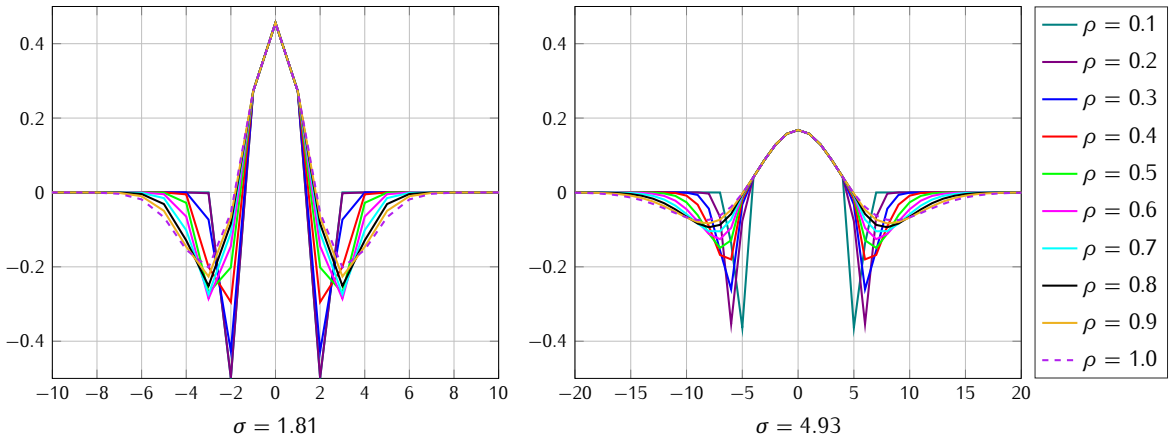


Figure 6.2: Discrete second derivatives of bi-Gaussians in 1D computed using different parameters ρ, see Eqs. (6.4) and (6.5).

6.2.1 Bi-Gaussian filter

The second order Gaussian G″ (Eq. (6.2)) is useful to determine the location of linear structures (Steger [1998]; Lindeberg [1998]; Lopez-Molina et al. [2015]; Magnier et al. [2021]). However, this simple Gaussian kernel relies on only one parameter to determine its shape: σ. This denotes one of the main, well-known drawbacks of the Gaussian filter. Due to the length of its support, this is therefore not sufficient to differentiate between adjacent or closely related structures, especially when the σ value is large (Xia et al. [2014]).

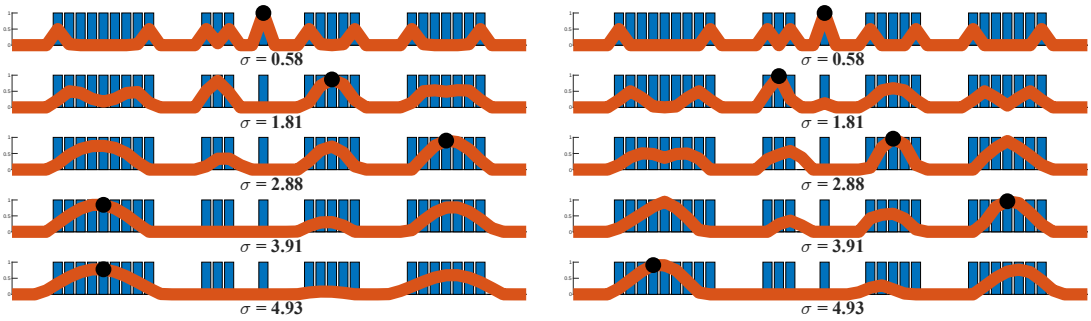
Consequently, linear structures cannot be suitably separately detected without any delocalization or fusion due to the regularization filter (Laligant et al. [2007]), as illustrated in Fig. 6.3(a). To address this drawback, the main idea is to transform the initial Gaussian filter into a bi-Gaussian, which combines the merits of the Gaussian and the Rectangular kernels. The benefits of this kernel are that it has a scale ratio able to clearly separate adjacent structures and, at the same time, the Gaussian part gives it robustness against noise. To tune the BG″ filter, a σ_b parameter allows us to play on the width and the sharpness of the curves on both sides of the central part (Xia et al. [2014]). To simplify, a parameter ρ is defined as the scale ratio:

$$\rho = \frac{\sigma_b}{\sigma}. \tag{6.4}$$

A ρ value ranging in]0, 1] improves the detection of peaks, especially for adjacent contours, by making the bi-Gaussian kernel narrower. The influence of ρ value is studied in the next section. The second order bi-Gaussian filter is expressed as follows:

$$\mathrm{BG}''(\sigma,\sigma_b,x) = \begin{cases} \rho^2 \cdot \mathrm{G}''(\sigma_b, x-\sigma_b+\sigma) & \text{if } x \le -\sigma \\ \mathrm{G}''(\sigma,x) & \text{if } |x| < \sigma \\ \rho^2 \cdot \mathrm{G}''(\sigma_b, x+\sigma_b-\sigma) & \text{if } x \ge \sigma. \end{cases} \tag{6.5}$$

When ρ = 1, the BG″ filter is equivalent to the 2nd derivative of the Gaussian G″. The Fig. 6.2 shows the 1D normalized BG″ filters for different values of σ and ρ. This filter's shape behaves in an opposite manner to Ziou's filter, which is very sharp in the middle, but contains the large length of its support(Ziou [2000]Shokouh et al. [2021b]). For the multiscale step, the highest filtered value is retained along the different scales (see Shokouh et al. [2021b]). The Fig. 6.3(b) illustrates the application of the BG″ filter at different scales on a 1D signal containing different ridges of variable widths. The convolved signals are plotted, and the maximum of the signal is displayed where each ridge is tied to a specific scale



(a) Convolution with G″

(b) Convolution with BG″, ρ = 0.5

Figure 6.3: Ridge highlighting in 1D for different scales σ. G″ and BG″ convolutions are used in (a) and (b) respectively. The blue bars represent the original signal containing separated ridges while the convolved signals are plotted in orange and the maximum of each signal is displayed by black circles.

(width of value 1, 3 5, 7 and 9). Contrary to the Gaussian kernel in Fig. 6.3(a), the BG″ best fits the signal along the different scales, revealing the great interest of this filter shape.

6.2.2 2DSBG: 2D Semi Bi-Gaussian filter

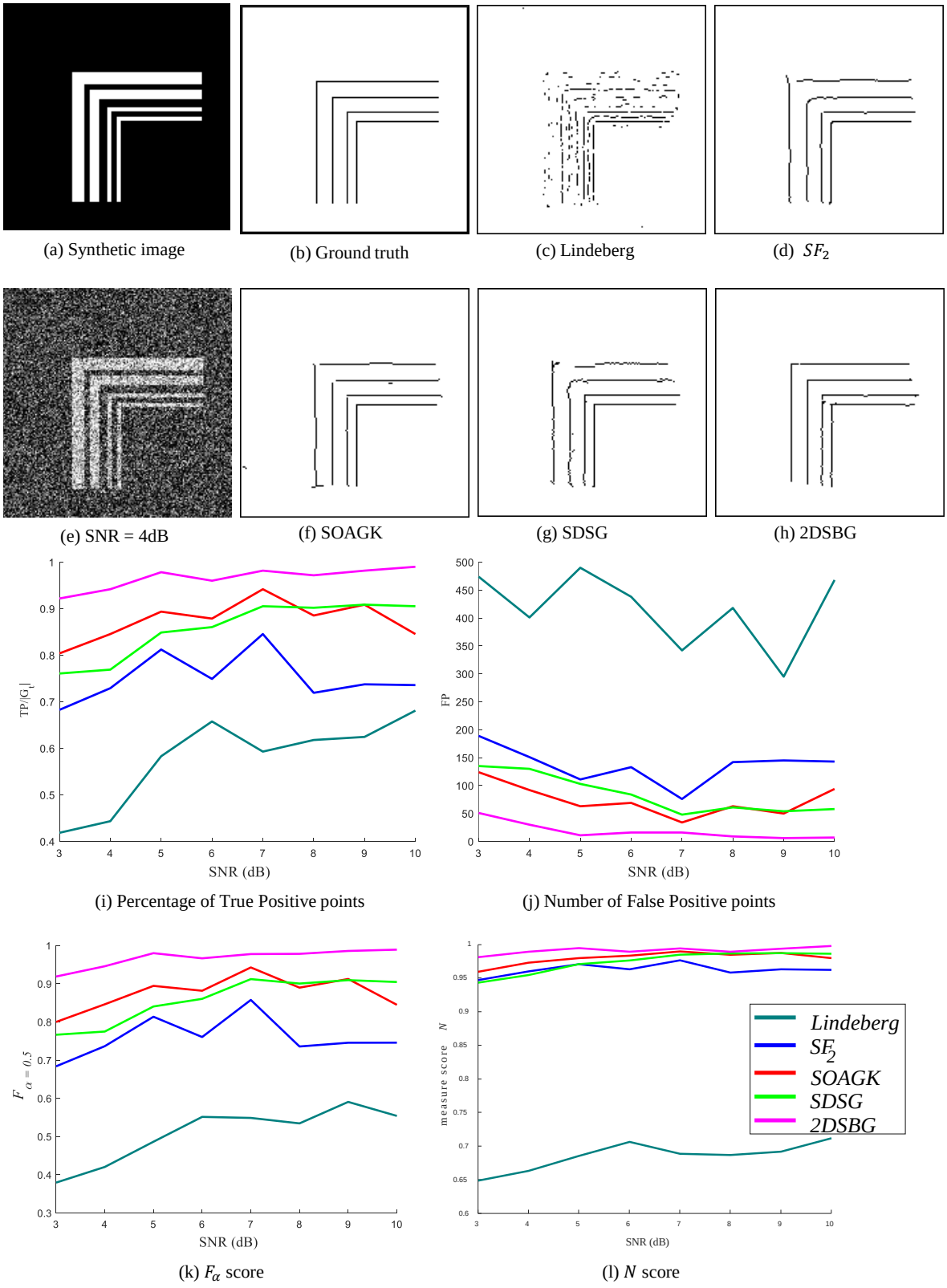
The proposed technique consists in combining a bi-Gaussian and a Semi-Anisotropic Gaussian filter which can be steered (Magnier et al. [2011, 2014, 2021]). The main idea of the developed filter is to consider: (i) close and parallel neighboring ridges and linear feature and (ii) paths (i.e. ridges or valleys) crossing each pixel. To innovate, the proposed filter can detect close and parallel narrow bent ridges of different widths in two different directions thanks to the semi bi-Gaussian capacities.

It is inspired by the SDSG (Second-Derivative of a Semi-Gaussian) where the main idea is to "*cut*" the second order anisotropic Gaussian kernel (Eq. (6.3)) using a Heaviside function and, then, steer this filter in all directions around the considered pixel: from 0 to 360°. Mathematically, the SDSG filter is defined by the combination of:

- a semi-Gaussian $\mathscr{G}$ for the smoothing, vertically truncated by a Heaviside function H, for $\sigma_s \in \mathbb{R}_+^*$ and $x \in \mathbb{R}$: $\mathscr{G}(\sigma_s, x) = \mathrm{H}(x) \cdot \mathbf{e}^{-x^2/2\cdot\sigma_s^2}$,
- a second derivative of a Gaussian G″ horizontally.

The proposed 2D filter substitutes the second order derivative of a Gaussian G″ for a second order derivative of a bi-Gaussian BG″ presented in Eq. (6.5): it is composed of Semi-Gaussian and bi-Gaussian operators. Note that for ρ = 1, the 2DSBG filter becomes the SDSG filter, see Fig. 6.1(c)-(d).

To adapt the multiscale strategy, the response of the filter for different scaling parameters is configured - and the maximum value among the different filter responses can be selected (Magnier et al. [2014]). When this new filter is steered towards the linear structure direction, the σ_s parameter allows an elongated smoothing in the line direction, whereas the σ captures the line structure strength (Eq. (6.5)). Then, the line structure strength is calculated using a local directional maximization/minimization (see).



(a) Synthetic image (b) Ground truth (c) Lindeberg (d) SF_2

(e) SNR = 4dB (f) SOAGK (g) SDSG (h) 2DSBG

(i) Percentage of True Positive points (j) Number of False Positive points

(k) F_α score (l) N score

Figure 6.4: Evaluation of the ridge extraction techniques on synthetic images corrupted by a Gaussian noise between SNR=10dB and SNR=3dB ($\sigma_s = 5 \cdot \sigma$ for SOAGK, SDSG and 2DSBG). On the bottom: visualization of the best segmented images at SNR=4dB.

6.3 Experimental evaluation and results

The proposed technique is evaluated on a set of both synthetic and real images-containing complex linear features such as close adjacent lines and ridges with varied scales. Figures/chapter4/datasets containing fungi images with manually annotated ground truth G_t are used. To evaluate the linear structure extraction, the *Normalized Figure of Merit* method (Magnier [2019]) called $\mathcal{N}$ is employed. Thus, $\mathcal{N}$ calculates a standardized dissimilarity score; the closer the evaluation score is to 1, the more the linear structure is qualified as suitable. On the contrary, a score close to 0 corresponds with poor contour detection.

The aim here is to get the best contour map in a supervised way. Theoretically, to be objectively compared, the ideal contour map must be the one for which the supervised evaluation gets the highest score (Magnier [2019]; Magnier et al. [2018]); and the overall quality is expressed in terms of the F_α-measure with α=0.5 (Magnier et al. [2018]).

In this context, the 2DSBG filter is compared with 4 other multiscale linear feature extraction techniques via filtering, namely: Lindeberg , SF_2 , SOAGK , and SDSG. Evaluation scores for synthetic cases are presented in Figs. 6.4(i)-(l) for percentage of true positive, false negative points, F_α and N measures respectively. In most cases, scores achieved by 2DSBG are superior to those of other techniques, showing the reliability of the proposed filter.

Fig. 6.5 shows that the optimum parameter ρ for the 2DSBG belongs to $[0.5, 0.7]$ and its reliability increases when ρ<1, compared to the SDSG filter (corresponding to ρ=1).

The results obtained with real images of fungus in Fig. 6.6 indicate that Linderberg's method and especially the SF_2 produce erroneous contours and thus have a low segmentation quality. Although SDSG has been able to detect most of the contour details as well as the bent and narrow structures, it also generated erroneous points due to its second-order noise sensitivity. Accordingly, with the previous performance results, the 2DSBG clearly extracts the most contours, especially in the case of narrow and adjacent lines, with barely any erroneous points. Note that the evaluation score here could be influenced by the hand annotated mispositioned ground truth, as detailed in Abdulrahman et al. [2017] for supervised evaluations.

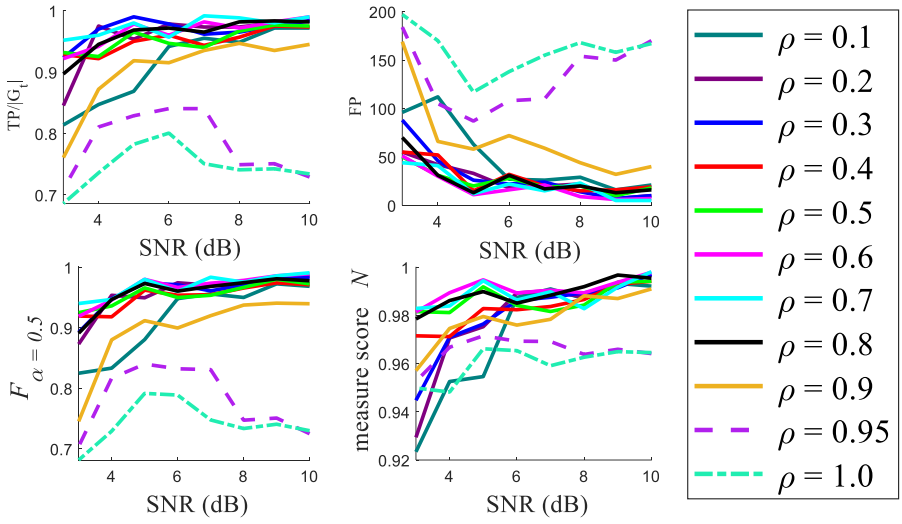


Figure 6.5: Estimation of the best ρ parameter of the 2DSBG on synthetic images corrupted by a Gaussian noise ($\sigma_s = 5 \cdot \sigma$).

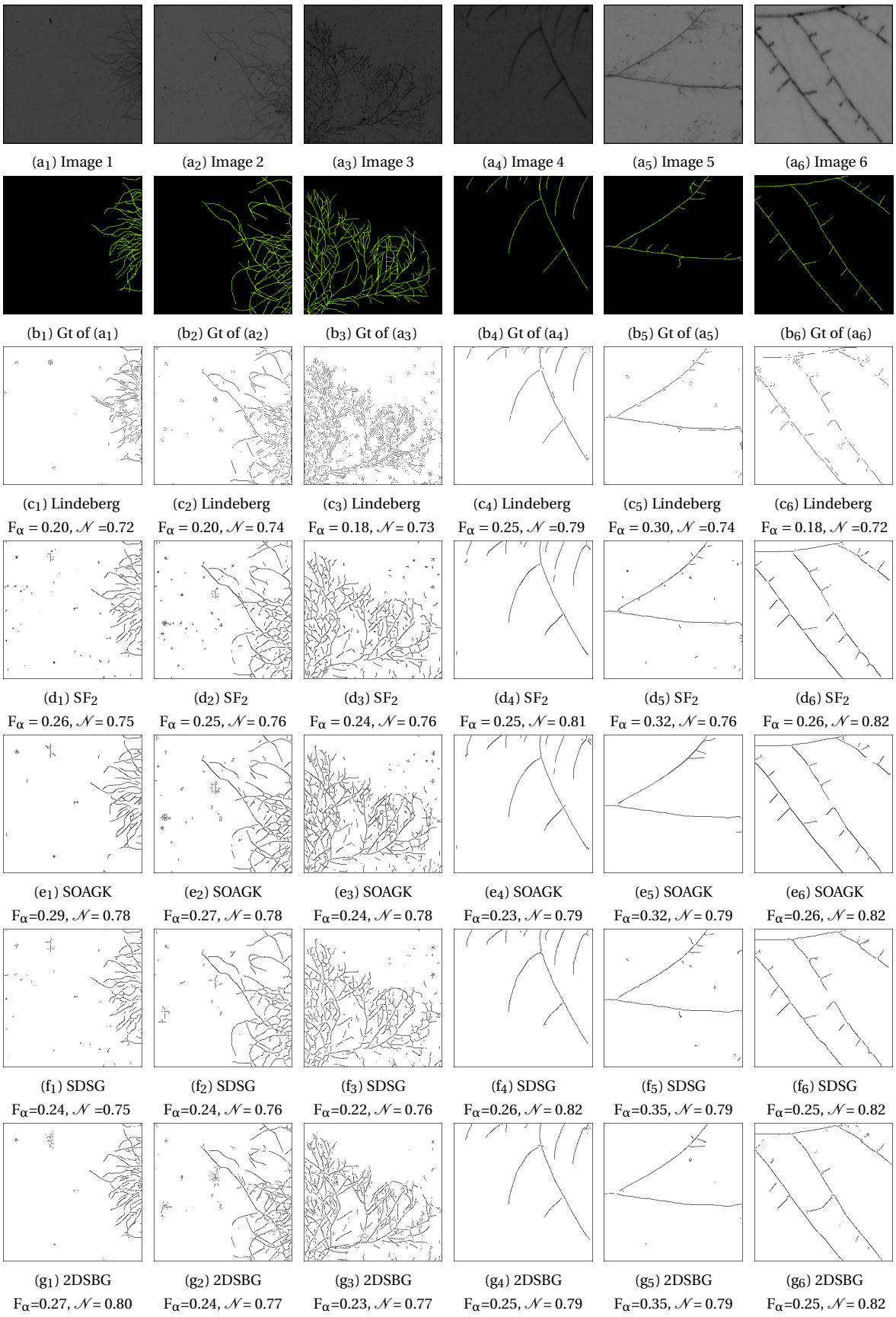


Figure 6.6: Images of size 300×300. Comparison of the proposed filter-2DSBG with the four state-of-the-art filters in the detection of linear structures in real fungus images. Parameters for SOAGK, SDSG and 2DSBG filters are the same: $\sigma_s = 5 \cdot \sigma$.

6.4 Summary

2DSBG, a new filter for multiscale linear features extraction in images, constructed from bi-Gaussian and second-order semi-Gaussian filters is presented. This filter exploits the advantages of the bi-Gaussian for the detection of adjacent linear features, as well as the qualities presented by semi-Gaussian kernels for the analysis of bent and complex linear structures. The particular merit of bi-Gaussian is the detection of adjacent linear features, and the precision of the Semi-Gaussian kernel for bent and complex linear structures have been exploited. Experiments on synthetic and real images were performed, allowing us to find the optimal parameter configuration ($\rho \in [0.5, 0.7]$), and thus confirming the novelty and merit of the 2DSBG over the existing filtering methods.

Chapter 7

Repeatability Evaluation of Keypoint Detection Techniques in Tracking Underwater Image Sequences

Contents

7.1 Introduction

The importance and interest in keypoint detection (i.e, corner or junction as a stable interest point) in a digital image lies notably in its application in image matching, tracking, motion estimation, panoramic stitching, object recognition, and 3D reconstruction. Image matching through feature tracking is extensively used in many real-time applications including autonomous driving, security surveillance, and manufacturing automation (Schmid et al. [2010]). Corner detection techniques can be effectively applied in these applications depending on their repeatability ratio.

The reason for the corner detection's wide range of applications is that the corner is easier to localize than other low-level features such as edges or lines, particularly taking into consideration the correspondence problems (e.g., aperture problem in matching). Hence, an objective evaluation of the frequently applied corner detection techniques by direct computation on the gray-level analysis relating to their real-time application is primarily invaluable, an example is available in Haggui et al. [2018].

The image matching and feature tracking in complex real-time scenes such as underwater videos are extremely challenging (Ferrera et al. [2019]). In this type of image sequence, concerning all types of image transformation (rotation, scale, affine transformations, translation, etc.), photometric transformation (illumination, occlusion, clutter, etc.), and various types of noises plus moving particles in different directions, the robustness of interest point, can be truly evaluated both objectively and visually. Repeatability is the main evaluation metric widely used for interest point matching, where the obtained points must be independent of varying image conditions (Schmid et al. [2010], Rodehorst and Koschan [2006], Mokhtarian and Mohanna [2006] and Rey-Otero et al. [2015]). In this chapter, the repeatability rate of the twelve commonly applied corner detection operators are objectively evaluated on the challenging underwater video dataset (Ferrera et al. [2019]).

In the literature, several approaches for detecting corners and junctions in digital images have been developed: (i) involving contour chains (Awrangjeb et al. [2012]), (ii) using templates (Rosten et al. [2008], Xia et al. [2014]) or, (iii) by image filtering techniques. Mainly, the corner detection operators via the direct computation on the gray-level analysis corresponding to the label (iii) can be categorized in three general approaches: Hessian based (Beaudet [1978]), curvature analysis (Kitchen and Rosenfeld [1982], Zuniga and Haralick [1983], Blom et al. [1992], Wang and Brady [1995], Zheng et al. [1999], Achard et al. [2000]), and structure tensor based (Forstner and Gulch [1987], Noble [1988], Harris and Stephens [1988], Shi and Tomasi [1994], Rohr [1994], Kenney et al. [2005]). These methods are easily developed by image filtering because they involve only image convolutions horizontally and vertically.

Therefore, they can be implemented with less computational time on different devices, see details in Haggui et al. [2018]. This paper is devoted to an extensive evaluation of filtering-based corner detection methods via repeatability performance measurement in video sequences consisting of frames with different types of transformations. It is to mention that the terms corner, junction, salient point, keypoint, and interest point are used synonymously in this work.

7.2 Studied keypoint detectors by gray-level direct computation

In this section, a set of corner detection techniques including the general scheme and the related parameters have been investigated. There are different approaches to determining the cornerness measure by direct computation using filtering techniques. Generally, in image filtering, the first or second derivatives may be utilized to determine corners in an image. Considering a gray-level image I, its partial derivatives are:

- $I_x = \frac{\partial I}{\partial x}$, the 1st image derivative along the x axis,
- $I_y = \frac{\partial I}{\partial y}$, the 1st image derivative along the y axis,
- $I_{xx} = \frac{\partial^2 I}{\partial x^2}$, the 2nd image derivative along the x axis,

- $I_{yy} = \frac{\partial^2 I}{\partial y^2}$, the 2nd image derivative along the y axis,

- $I_{xy} = \frac{\partial^2 I}{\partial x \partial y}$, the crossing derivative of I.

These image derivatives can be calculated by convolving the image with the [1 0 -1] or the [1 0 -2 0 1] masks in the x and/or y directions for the first and second derivatives, respectively (Shokouh et al. [2021b]). The first derivatives are useful for the detection of step and ramp edges, whereas the second derivatives are convenient for the contour extraction of types: line, roof edges as ridge/valley features. Regarding the image surface, corners are defined as the curvature extremum along the edge line (Rohr [1994]). Usually, approaches to detect directly corners on the gray scale level use filtering techniques by combining image derivatives of the 1st and 2nd order, then by computing the Hessian matrix, the curvature or the structure tensor. All the necessary technical details of the studied corner detection methods including mainly the formula, denomination, parameter(s), and name of the authors are listed by year of publication in Tab. 4.1.

7.2.1 Determinant of the Hessian Matrix

Mathematically, the Hessian matrix ($\mathscr{H}$) indicates significant values near edges, through which corners can be estimated by the large variations of intensity values in both x and y directions. Indeed, $\mathscr{H}$ represents a square matrix of 2nd order partial derivative of image intensity; it is often computed and is useful in feature detection and characterization:

$$\mathscr{H} = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}. \tag{7.1}$$

In that respect, Beaudet [1978] uses the image 2nd derivative for calculating the determinant of $\mathscr{H}$, which is related to Gaussian curvature of the image surface (Lipschutz [1969]). The computation is straightforward because it combines only three 2nd image derivatives. Even though this technique is rotation invariant, it is sensitive to noise and unstable against scale changes.

7.2.2 Curvature analysis

Technically, these techniques are based on the change of gradient direction along an edge contour and/or image surface curvature. They can be easily computed by the combination of the image derivatives of 1st and 2nd order. The pioneer work in this category was originally led by Kitchen and Rosenfeld (KR) who defined the cornerness measure for each pixel intensity based on the change of 2nd order gradient direction along the edge weighted by the local gradient magnitude (Kitchen and Rosenfeld [1982]).

Theoretically, the gradient feature vector ∇I is normal to the edge and hence projecting the change of gradient direction along the edge and multiplying the result by the local gradient magnitude $|\nabla I|$ results the final cornerness measure. Inspired by this initial contribution of Kitchen and Rosenfeld [1982], other related techniques were developed by Zuniga and Haralick [1983], Blom et al. [1992] and Wang and Brady [1995].

Zheng et al. [1999] and Achard et al. [2000] use a smoothed image with a Gaussian of parameter ρ and then the combination of its derivatives with the derivatives of the non-smoothed image. The cornerness measure for each technique is listed in the Tab. 4.1.

Table 7.1: Cornerness measure formulas computed from image derivatives. Here, $\overline{I_x}$, $\overline{I_y}$, $\overline{I_{xx}}$, $\overline{I_{yy}}$ and $\overline{I_{xy}}$ denote the convolutions with a Gaussian with a standard deviation $\sigma > 0$ of images derivatives I_x, I_y, $\hat{I}_{xx}$, I_{yy} and I_{xy} respectively. As a reminder, (λ_1, λ_2) represents the eigenvalues of the structure tensor $\mathscr{M}$.

Name	Cornerness Measure/Formula	Parameter(s)	Reference(s)	Year
DET	$\mathrm{DET}(\mathscr{H}) = I_{xx}I_{yy} - I_{xy}^2$	-	Beaudet [1978]	1978
KR	$\frac{I_x^2 I_{yy} - 2 \cdot I_x I_y I_{xy} + I_y^2 I_{xx}}{I_x^2 + I_y^2}$	-	Kitchen and Rosenfeld [1982]	1982
ZH	$\frac{\mathrm{KR(I)}}{\lvert\nabla \mathrm{I}\rvert} = \frac{I_x^2 I_{yy} - 2 \cdot I_x I_y I_{xy} + I_y^2 I_{xx}}{\left(I_x^2 + I_y^2\right)^{3/2}}$	-	Zuniga and Haralick [1983]	1983
F	$\frac{Det(\mathscr{M})}{Trace(\mathscr{M})} = \frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2} = \frac{\lambda_1\lambda_2}{\lvert\nabla\bar{\mathrm{I}}\rvert^2} = \frac{\overline{I_x}^2\overline{I_y}^2 - \overline{I_{xy}}^2}{\overline{I_x}^2 + \overline{I_y}^2}$	ρ	Forstner and Gulch [1987], Noble [1988]	1987
HS	$Det(\mathscr{M}) - k \cdot (Trace(\mathscr{M}))^2 = \overline{I_x}^2\overline{I_y}^2 - \overline{I_{xy}}^2 - k \cdot \left(\overline{I_x}^2 + \overline{I_y}^2\right)^2$	ρ, k	Harris and Stephens [1988]	1987
BB	$\lvert\nabla \mathrm{I}\rvert \cdot \mathrm{KR(I)} = I_x^2 I_{yy} - 2 \cdot I_x I_y I_{xy} + I_y^2 I_{xx}$	-	Blom et al. [1992]	1992
Ro	$Det(\mathscr{M}) = \lambda_1\lambda_2 = \overline{I_x}^2\overline{I_y}^2 - \overline{I_{xy}}^2$	ρ	Rohr [1994]	1994
KLT	$Min(\lambda_1, \lambda_2)$	ρ	Shi and Tomasi [1994]	1994
RTC	$\frac{(1+I_x^2)I_{yy} - 2 \cdot I_x I_y I_{xy} + (1+I_y^2)I_{xx}}{\left(1 + I_x^2 + I_y^2\right)^{3/2}}$	-	Wang and Brady [1995]	1995
GD	$I_x^2 I_{yy}^2 + I_y^2 I_{xx}^2 - \frac{\overline{I_x}^2\overline{I_{yy}^2} + \overline{I_y}^2\overline{I_{xx}^2}}{\left(\overline{I_x}^2 + \overline{I_y}^2\right)^2} \cdot \left(I_x{}^2 + I_y{}^2\right)^2$	ρ	Zheng et al. [1999]	1999
ABD	$\frac{I_x^2\overline{I_y}^2 + I_y^2\overline{I_x}^2 - 2 \cdot I_x I_y \overline{I_x I_y}}{\overline{I_x}^2 + \overline{I_y}^2}$	ρ	Achard et al. [2000]	2000
KZ	$\frac{1}{\left(\lambda_1^{-p} + \lambda_2^{-p}\right)^{1/p}}$	ρ, $p > 0$	Kenney et al. [2005]	2005

7.2.3 Structure tensor

The third group uses the symmetric structure tensor $\mathscr{M}$:

$$\mathscr{M} = \begin{pmatrix} \overline{I_x^2} & \overline{I_x I_y} \\ \overline{I_x I_y} & \overline{I_y^2} \end{pmatrix}, \tag{7.2}$$

where $\bar{\bullet}$ indicates convolution with a low-pass filter; here a Gaussian filter of standard deviation ρ is considered (see Eq. (7.3) with a parameter ρ instead of σ).

The structure tensor is derived from the gradient structure tensor, which is achieved through, computing the Cartesian product of the gradient vector with itself at each point of the image. Spatial averaging of this tensor, usually with a Gaussian filter, leads to the structure tensor. As to note that averaging is needed as the plain gradient tensor has only one non-zero eigenvalue and therefore represents only innately edge features. Spatial averaging, here tied to ρ parameter, distributes this information over a neighborhood, and points that receive contributions/input from edges with different orientations will have two positive eigenvalues, which allows them to be recognized as intrinsically 2D.

Eigenvectors of the gradient structure tensor indicate local orientation, whereas eigenvalues λ_1 and λ_2 give the strength or magnitude as a measure of eigenvalues in flat regions are very small (negligible), in the edges λ_1 or λ_2 is small depending on the horizontal or vertical edge, and noticeably both values λ_1 and λ_2 are large in corner points.

Based on this assumption, various corner measurement formulations have been proposed; they are listed and denominated in the Tab. 7.1 and summarized here:

- Forstner and Gulch [1987] and Noble [1988] use an auto-correlation matrix $\mathscr{M}$ with the function F to identify salient points, which converges toward the point closest to

all the tangent lines of the corner in a neighborhood and is a least-square solution. The function combines the eigenvalues, aiming to classify the keypoints from other types of local features.

- Harris and Stephens [1988] also named as Plessey operator is based on principal curvature of local auto-correlation using first order derivative. This operator's response is theoretically isotropic, but often computed in anisotropic way. This cornerness measure HS yields two positive values at corners and two negative values in the case of straight edges. HS and F methods differ only in their criterion determination because the Fostner algorithm disregards the parameter k ($k > 0$) introduced in Harris and Stephens by computing a fraction.

- Rohr used a parametric model fitting as a point of maximal curvature in the direction of the steepest slope (Rohr [1994]). He convolved an analytical corner model with a Gaussian and adjusted the parameter of the model by a minimization technique to have the model near the observed signal. It is remarkable that this algorithm corresponds to HS technique for a value $k = 0$.

- Shi and Tomasi estimated that the corners are primitives which remain more stable for tracking, contrary to other features (Shi and Tomasi [1994]). Consequently, the minimum eigenvalue between λ_1 and λ_2 of the matrix $\mathcal{M}$ is conserved for a salient point along a video; then this detector led to the well-known KLT (Kanade-Lucas-Tomasi) feature tracker.

  Kenney et al. [2005] combines λ_1 and λ_2 with a cornerness measure which is constrained by the numbers of required axioms to satisfy. The axioms mainly formulate the isotropy condition (rotation invariant corner), orthonormality of the matrix M, constant eigenvalues relating to the norm, and finally definition of the maximum value of the isotropic point over the set of constant eigenvalues. As detailed in Kenney et al. [2005], KZ detector technique is equivalent modulo for the choice of a suitable matrix norm and a normalization constant to:

  - ☐ F when $p = 1$,
  - ☐ KLT when $p \to +\infty$,
  - ☐ $\sqrt[p]{2}$R for $p \to 0$.

  In our experiments, p is fixed to 2.

These six corner detection techniques have in common the tensor $\mathcal{M}$, which is tied to the same low-pass filter parameter, here denoted ρ: the standard deviation of the Gaussian. This ρ value is identical for each compared technique in the experiment presented in the next section.

7.2.4 Detection of a corner: Final step

For corner extraction, the final step concerns the binarization of the detected salient points or corners. The obtained features from the techniques presented in subsections 7.2.1–7.2.3 and listed in the Tab. 4.1 compute cornerness measure by applying non-maximum suppression where the local maxima are tied to the corner positions (here a window of size 15×15 is chosen to avoid too close keypoints). Eventually, corner points are highlighted by thresholding the extracted points or by setting the number of corner points to be detected objectively (this last solution was adopted in our experiments).

7.3 Keypoint repeatability assessments

Repeatability measure is the defacto standard and is commonly applied for the performance characterization of salient point detectors (Tuytelaars and Mikolajczyk [2008] Rodehorst and Koschan [2006]). The repeatability rate measures the detector's robustness in being able to detect the same features in the condition of image perturbations (e.g., a corner detector that is robust in the condition of image perturbation is rated as a highly repeatable detector). To pursue a vigorous evaluation of techniques detailed and nominated in Tab. 4.1, our experiments are carried out on a specific database containing strong perturbations, as reported in Tab. 7.2 and detailed in the next subsection.

Table 7.2: Experimental protocols of selected videos in the AQUALOC dataset.

Sequence №	№ of Frames	Description of image transformation
Video sequence 1	1000	rotation, affine, illumination, scale
Video sequence 2	2000	rotation, affine, illumination, perspective
Video sequence 3	2000	rotation, illumination, homogeneous, scale
Video sequence 4	2000	rotation, illumination, occlusion, translation
Video sequence 5	3000	rotation, illumination, affine, scale, clutter

7.3.1 Experimental protocol

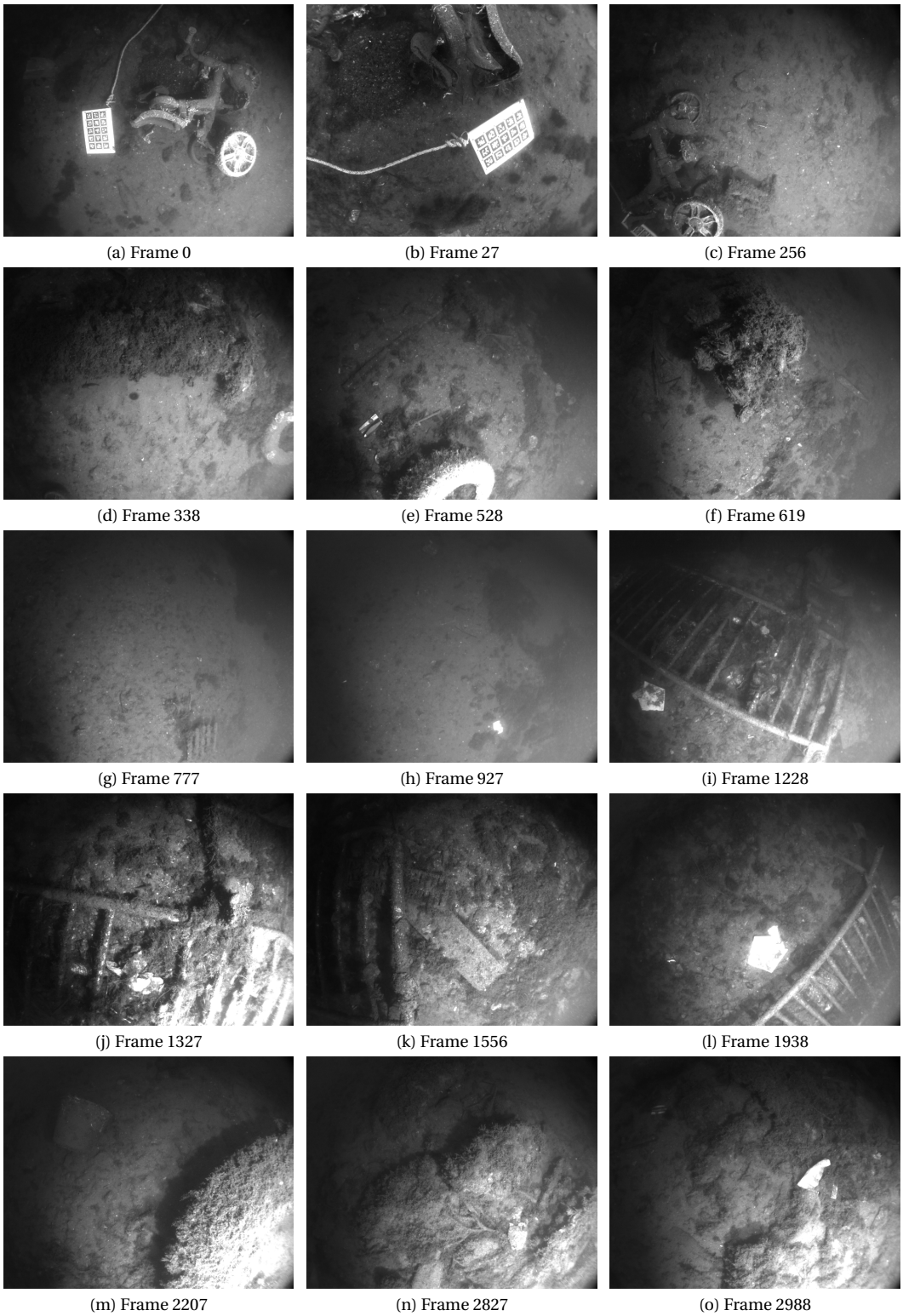
In this assessment, the repeatability rate is considered as the percentage of the total number of points that are repeated between two subsequent video frames in the AQUALOQ dataset[1]. This dataset is an underwater video sequence dedicated to the localization of underwater vehicles navigating close to the seabed. These videos have been recorded from remotely operated vehicles equipped with a monocular monochromatic camera. The image data is complex in consisting of all types of geometric and photometric transformation plus different types of noises and concurrent moving particles moving in different directions as illustrated in the Fig. 7.1. Nevertheless, the frame movements are very smooth, denoting a small inter-frame distance. For our statistical experiments, we randomly selected 5 videos with different numbers of frame sequences. With a total of 10 000 frames, video 1 includes 1000 frames, videos 2, 3 and video 4 contain 2000 frames, finally, video 5: 3000 frames, as listed and detailed in the Tab. 7.2.

In order to remove the noise in the image and obtain more relevant keypoints, the zeroth order two-dimensional Gaussian kernel G is used for regularization by convolution with the image (Canny [1986]). Its equation is given by:

$$G(\sigma, x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{with } \sigma \in \mathbb{R}_+^*, \quad \text{and } (x, y) \in \mathbb{R}^2, \tag{7.3}$$

where σ represents the standard deviation of the Gaussian G and (x, y) the pixel coordinates. Hence, the images are smoothed with G of parameter $\sigma = 1$ before applying keypoint detector techniques. Regarding these twelve keypoint detectors, detailed in Section 7.2 and listed in Tab. 4.1, all of them detect the 100 best points per frame. The choice of ρ parameter value for the detectors consisting of ρ parameter, is usually made empirically

[1]The AQUALOC Dataset Ferrera et al. [2019] is available at: https://www.lirmm.fr/aqualoc/



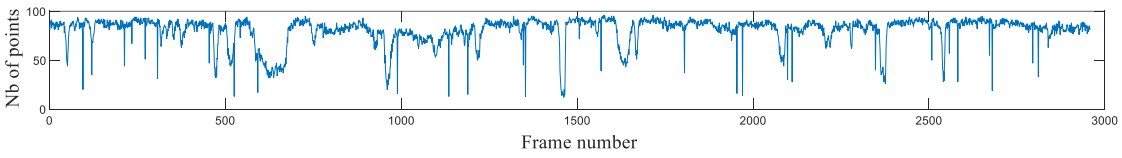
(a) Frame 0 (b) Frame 27 (c) Frame 256

(d) Frame 338 (e) Frame 528 (f) Frame 619

(g) Frame 777 (h) Frame 927 (i) Frame 1228

(j) Frame 1327 (k) Frame 1556 (l) Frame 1938

(m) Frame 2207 (n) Frame 2827 (o) Frame 2988



(p) Number of detected points by Ro as a function of the frame number along the video 5.

Figure 7.1: Selected images of video 5 of AQUALOC dataset (Ferrera et al. [2019]) for visual explanation of all types of transformation and noises, images of size 512×640.

because a too large value can delocalizes the keypoint position and will "disrupt" the repeatability. Indeed, when the ρ value increases from certain threshold, the keypoints can get misplaced increasingly (see example in Rodehorst and Koschan [2006]) and some of them could be merged. Meanwhile, too small ρ value is a restricting threshold, limiting the detection of structure and will result the low repeatability ratio for matching. Therefore, for each detector, the ρ value is estimated in the Sec. 7.3.2.1.

7.3.2 Evaluation via ZNCC process

Tracking by matching the features is defined as obtaining the features (i.e., keypoints) in the first frame I_1 of the video sequence and then finding the corresponding pairs of points in the subsequent frame I_2. After detecting a feature point in I_1, a feature point in I_2 is generally estimated and located by computing the intensity variation between a patch in I_1 and patches in I_2. There are several straightforward metrics to estimate the similarity between the two intensity patches, such as Sum of Squared Differences (SSD), Sum of Absolute Difference (SAD), Normalized Cross-Correlation (NCC), and Zero-Mean Normalized Cross-Correlation (ZNCC). For our assessments, the ZNCC is chosen as the optimal evaluation metric in matching and tracking, because it is more precise as being less sensitive to proportional changes of intensity:

$$\mathrm{ZNCC}(\Omega_1,\Omega_2) = \frac{\sum_i^{\mathrm{N}}\sum_j^{\mathrm{N}}[(\Omega(i,j)-\mu_1).(\Omega_2(i,j)-\mu_2)]}{\mathrm{N}^2\cdot\sigma_1\cdot\sigma_2}, \tag{7.4}$$

where, Ω_1 and Ω_2 correspond to the frame patches of size N×N pixels, (μ_1,σ_1) and (μ_2,σ_2) are the mean and standard deviations of the intensities of the patches Ω_1 and Ω_2 respectively. In case where a keypoint is calculated in a homogeneous region: $\sigma_1 = 0$ or $\sigma_2 = 0$, consequently Eq. (7.4) does not compute a value in the desired range, so $\mathrm{ZNCC}(\Omega_1,\Omega_2) = 1-|\mu_1-\mu_2|$. The ZNCC computes the similarity measures between the two equally sized image patches (I_1,I_2), and gives a scalar in the range $[-1;1]$. The value/score between $[0;1]$ indicates the ratio of positive correlativity of the features. The closer to 1 the score is, the similarity between the patches is. As there are small displacements between each frame in the studied dataset, consequently, the ZNCC descriptor is applicable for matching because patches spatially close to another patch in the subsequent frame will obtain a positive ZNCC score.

In the evaluation process, for scoring the repeatability ratio of each studied feature detector, three statistical metrics are first computed, namely: (i) mean of matched points (percentage of matched points exactly), (ii) Standard deviation (Std) of matched points and (iii) ZNCC mean for each frame.

Thereafter, in order to obtain an evaluation as objective as possible, the feature point detectors having a ρ parameter are compared by varying this parameter. Indeed, ρ is increasing from 0.5 to 4.5 by a step of 0.5 and the repeatability ratio is estimated for each scale, see Sec. 7.3.2.1. Consequently, the Table 7.3 reports the mean and Std of matched points as a function of the video number for each detector (and best ρ). To complete the evaluation, a final score is computed in Eq. (7.5) to estimate the reliability of the detectors as a function of 3 entities:

- $\mathscr{M}_{Matched}$: the usual mean of matched points among 100 detected points along the 5 videos.
- $Std_{Matched}$: the usual standard deviation of matched points along the 5 videos.

- $\mathscr{M}_{\text{ZNCC}}$: the mean of ZNCC scores for all the matched points along the 5 videos.

These 3 values are displayed in Fig. 7.2(a)-(d) as a function of the ρ values regarding the 7 feature detectors: F, H, R, KLT, GD, A and K. Usually, it is worth noting that the $Std_{Matched}$ and $\mathscr{M}_{\text{ZNCC}}$ score can be inversely proportional to the number of detected points; an example, a detector can match few number of points by having a low $Std_{Matched}$ value, as ZH in Tab. 7.3. Furthermore, $\mathscr{M}_{Matched}$, $Std_{Matched}$ and $\mathscr{M}_{\text{ZNCC}}$ are normalized and the final score is computed by:

$$\mathscr{T}_{\text{S}} = \frac{a \cdot \text{N}(\mathscr{M}_{Matched}) + b \cdot \text{N}(Std_{Matched}) + c \cdot \text{N}(\mathscr{M}_{\text{ZNCC}})}{a+b+c}, \tag{7.5}$$

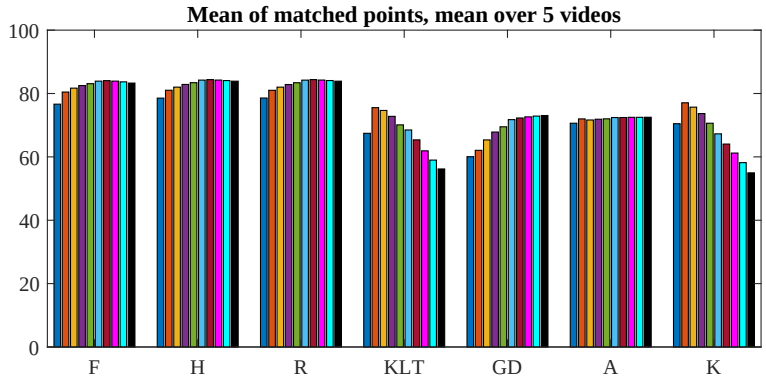
where N represents the normalization function, and (a, b, c): 3 positive coefficients such that $(a+b+c) = 1$. In our experiments, $a = 0.4$, $b = 0.3$ and $c = 0.3$ to correspond to weights such that the mean of detected points remains the main entity. The Fig. 7.2(d) reports different scores as a function of the ρ variation. This total score is also computed for the detector without ρ parameters, namely: DET, KR, ZH, BB and RTC. Finally, all the statistics are reported in the Tab. 7.4. The more the $\mathscr{T}_{\text{S}}$ score is close to 1, the more the feature detector is qualified as suitable for repeatability. On the contrary, a score close to 0 indicates a low reliability of a detector.

7.3.2.1 Scale ρ parameter fitting

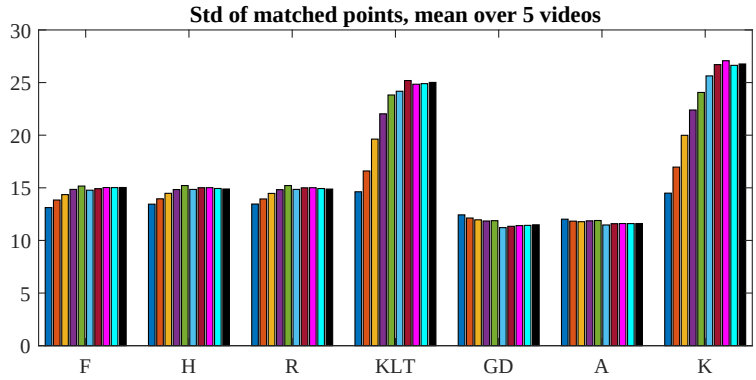
Depending on the ρ parameter, the detectors F, H, and R shows the highest stability for different scale ratio as presented in the Fig. 7.2(d) total score $\mathscr{T}_{\text{S}}$. Here, the detectors, KLT and K total score decreases for ρ values greater than 2.5, whereas the detectors GD and A produce average total score values. Correspondingly, the statistics tied to the best scores regarding ρ parameter of these detectors are reported in the Table 7.4, as for statistics in the Tab. 7.3. It is important to note that values are rounded in Tab. 7.3; whereas original/exact values serve for Tab. 7.4.

Table 7.3: Percentage and Std of detected points per video

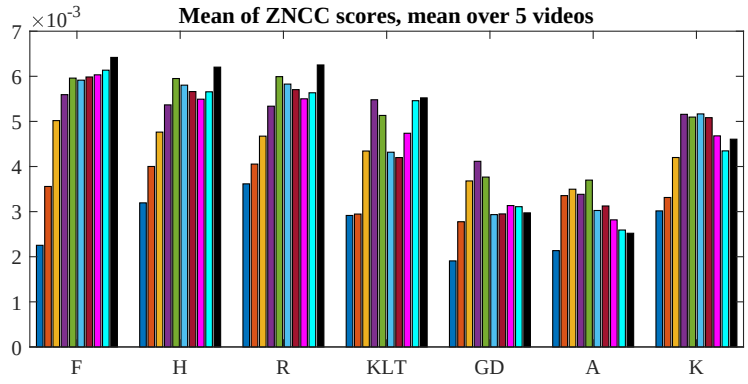
Detector	Video 1		Video 2		Video 3		Video 4		Video 5	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
DET	81.9	7.6	76.4	14.6	80.1	13.1	68.7	15.7	77.9	11.9
KR	76.9	7.1	70.6	13.9	74.6	12.0	64.6	14.4	71.5	11.0
ZH	8.3	5.1	10.3	9.0	11.7	6.8	10.1	8.5	12.2	9.1
F	**90.2**	8.7	81.8	17.0	77.0	17.9	**88.8**	11.6	**84.5**	15.4
HS	90.1	8.6	**82.0**	17.0	**77.2**	17.7	88.5	12.4	82.9	16.5
BB	84.6	7.9	77.9	15.3	81.5	13.5	10.1	8.6	79.3	12.7
Ro	90.1	8.6	**82.0**	17.0	**77.2**	17.7	88.5	12.4	82.9	16.5
KLT	81.0	9.5	70.1	23.1	57.3	29.1	86.5	12.0	71.6	27.5
RTC	82.2	7.7	76.6	14.8	80.6	13.2	71.3	16.2	78.2	12.0
GD	64.4	11.8	68.8	12.9	64.6	13.2	71.2	9.2	69.6	12.0
A	78.1	7.0	71.1	13.4	67.1	13.8	73.8	10.0	72.0	12.5
KZ	86.3	9.9	71.7	23.7	57.8	29.6	87.4	12.1	72.2	27.8



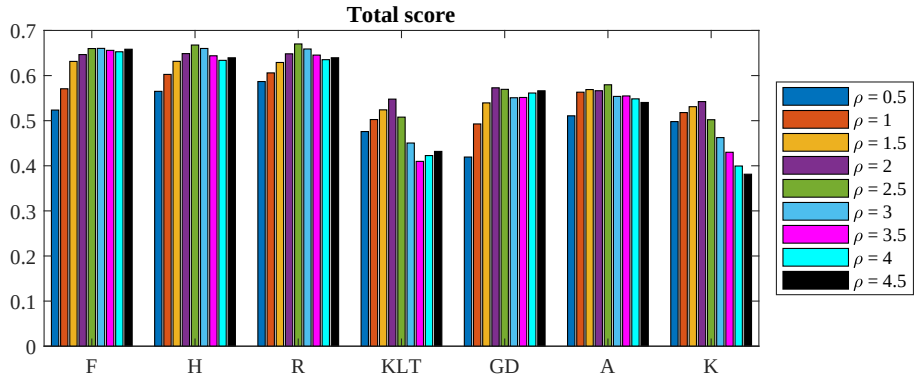
(a) Mean of matched points $\mathscr{M}_{Matched}$ among 100 detected points



(b) Standard deviation of matched points $Std_{Matched}$



(c) Mean of ZNCC scores $\mathscr{M}_{\text{ZNCC}}$ for all the matched points $\mathscr{M}_{Matched}$ in (a)



(d) Total score $\mathscr{T}_S$ computed as in Eq. (7.5)

Figure 7.2: Matched and repeatability statistics as function of ρ values for 7 filtering techniques averaged on 5 videos: ρ values in the range of 0 to 4.5

Table 7.4: Main repeatability statistical scores of matched points, mean ZNCC (rounded here) and final conclusive scores $\mathcal{T}_S$ in Eq. (7.5).

Detector	DET	KR	ZH	F	HS	BB	Ro	KLT	RTC	GD	A	KZ
ρ parameter	-	-	-	3	2.5	-	2.5	2	-	2	2.5	2
$\mathcal{M}_{Matched}$	76.62	71.11	10.92	83.90	83.41	66.15	83.40	72.79	77.36	67.83	72.01	73.66
$Std_{Matched}$	13.01	12.04	8.09	14.76	15.22	12.06	15.21	22.03	13.19	11.85	11.89	22.39
$\mathcal{M}_{ZNCC}$	0.008	0.007	0.15	0.006	0.006	0.010	0.006	0.005	0.008	0.004	0.003	0.005
$\mathcal{T}_S$, see Eq. (7.5)	0.34	0.32	0.07	0.66	0.66	0.30	0.67	0.55	0.35	0.57	0.58	0.54

7.3.2.2 Usual statistical evaluation

The mean ZNCC values for each operator demonstrate the cumulative average similarity or correlation of only the corresponding patches in the given image sequences; they are also reported in the Tab. 7.4. The obtained values are all positive values, implying certain similarities and measures of repeatability of the corresponding matched points. Nevertheless, the mean ZNCC values are a function of matched points per detected points. As an example, the mean ZNCC values for the two detectors Ro and ZH (best versus worse detectors) correspond to the number of matched points per total number of detected points. It is to note that, a solid interpretation of the obtained values is complicated as each different operator's performances with different types of image transformation, occlusion and noises vary widely, as illustrated in Fig. 7.1. To be recalled again, for this experiment the displacement of frames are smooth and small, so keypoints in consecutive frames should not be too far spatially from the successive frames in the experimental videos. As an example, the detector DET shows a high percentage of matched corner points in the first video sequences; however, in the video 4, drastic changes in scores appear relating to the different transformations (81.9 against 68.7 in Tab. 7.3). Hence, the repeatability ratio is always dependent on the type of image transformation.

Another example concerns a group of structure tensor based techniques having ρ scale parameter. Even though the detector Ro, HS, and F obtained the optimal matching score, signifying higher stability of repeatability in key-point detection, HS (well-known Harris) as often called the benchmarking corner detector has not obtained the highest matching score among all, concerning the effects of image transformation detailed in Tab. 7.2.

Furthermore, the detectors A, GD, KLT, and KZ have shown significant repeatability scores in the descending order. Besides the detectors, RTC, DET, KR and BB have shown low final scores $\mathcal{T}_S$ which their Std of matched points is also low. This objective repeatability assessment enables a valuable choice of the ρ scale parameter. Indeed, a bad ρ value regarding F, HS or Ro detectors (ρ=0.5 or 1, see Fig. 7.2) produces poor statistics than BB detector which is one of the least reliable detector (see Tab. 7.4). To conclude this part, since the movement of the camera is smooth, a score correlated to ZNCC constitutes a technique to assess the keypoint detectors –enabling to estimate the optimum ρ parameter–. Finally, the detector ZH repeatability score is the lowest, conveying that in the condition of numerous image transformations and noises (such as underwater videos), this detector is unreliable to use; and this drawback is mainly due to sensitivity of this detector to strong illumination changes.

To conclude this part, since the movement of the camera is smooth, a score correlated to ZNCC constitutes a technique to assess the repeatability of the keypoint detectors and enabling to estimate the optimum ρ parameter for several keypoint detectors.

7.3.2.3 Visual results

In Fig. 7.3(a), 50 random frame sequences have been chosen to illustrate the curves of the percentage of matched points. The 100 detected points along the sequence are displayed in blue in the first frame, while the linked points by the ZNCC are plotted in green in the Fig. 7.3(b)-(m). Visually the Ro, HS, and F are the best keypoint detection methods with smooth lines whereas the ZH, GD, KR and A detectors seem not stable because the displayed lines are sharped (zig-zag), illustrating the misplacement in the images of the Figs. 7.3 (c)-(d)-(k)-(l), reflecting the weak reliability of these techniques.

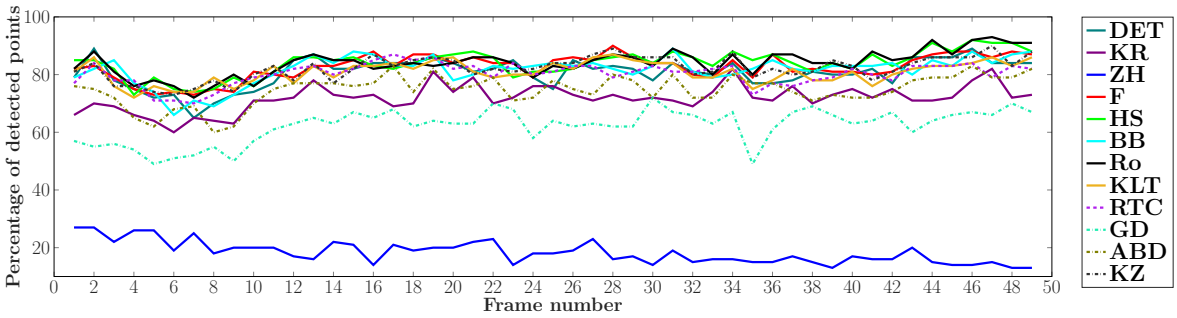
7.4 Summary

In this chapter, an extensive investigation of the twelve state-of-the-art keypoint detection techniques with the application to feature matching in the context of a complex video scene (AQUALOQ: containing all types of image transformation and various natural noises caused by the water and sand) has been conducted. The repeatability rate of each detection operator has been both statistically and visually demonstrated in presenting a guideline of which detectors are robust depending on video frame complexity.

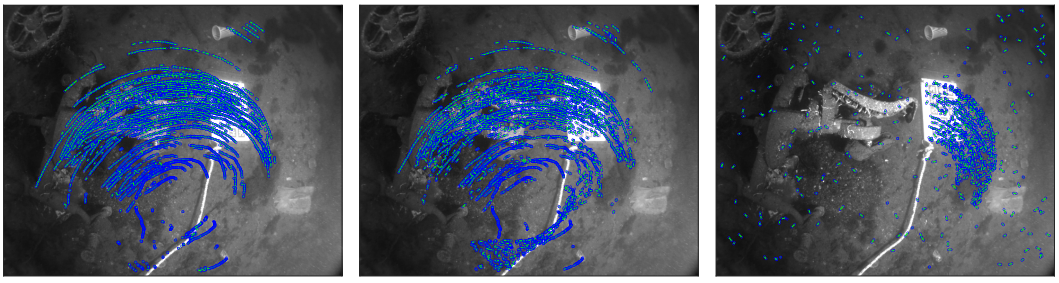
This work can contribute as a directive to the practitioners of this domain for choosing the appropriate keypoint detectors relating to the specific application (i.e., Ro, HS, and F exemplify the robust salient point detectors with the highest stability). The scale parameter ρ of the studied keypoint detectors have been studied for an objective and complete evaluation. The results show that KLT and Harris-Stephens (HS), two particularly popular detectors, perform well but not the best among the twelve tested, especially when the ρ parameter is not well selected.

This evaluation emphasizes on the filtering technique which is fast and straightforward than other approaches along with the keypoint matching methods (ZNCC) which is few time-consuming and can be easily implemented. Accordingly, the filtering techniques are useful for certain cases of image processing and optimization which are either used independently or can be used alongside with deep learning models either in preprocessing or post-processing stages.

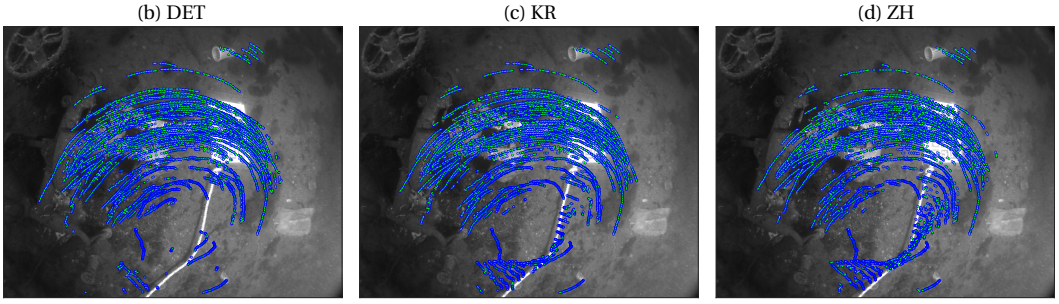
In closing, this study could be performed on SLAM (Simultaneous Localization and Mapping) of these video sequences. SLAM is the most important problems in the pursuit of building truly autonomous mobile robots. With SLAM the spatial map of environment while simultaneously localizing the robot relative to this model can be acquired. The SLAM of this keypoint detection techniques will realize many new general purpose applications of mobile robotics.



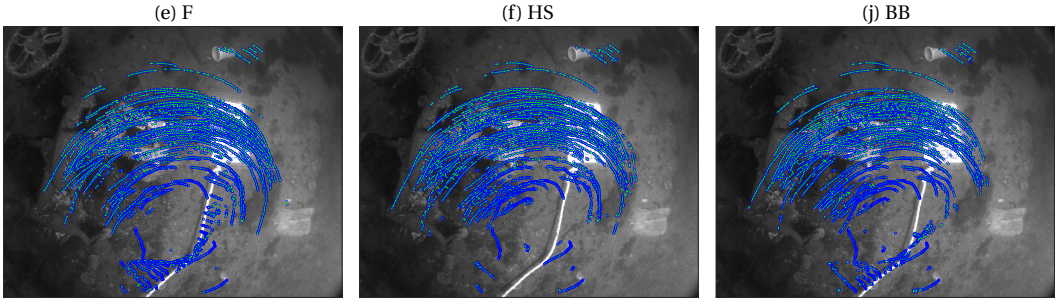
(a) Percentage of matched points per detector along 50 successive images



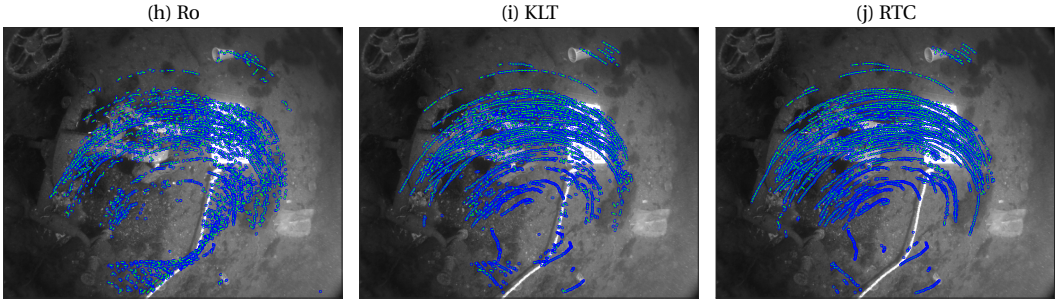
(b) DET (c) KR (d) ZH



(e) F (f) HS (j) BB



(h) Ro (i) KLT (j) RTC



(k) GD (l) ABD (m) KZ

Figure 7.3: 100 matched keypoints on a sequence of 50 images. In (b)-(m), detected points appear in blue along the sequence, while links are in green.

Chapter 8

New Anisotropic Corner Detection using Causal Filtering

Contents

8.1 Introduction

Causal filtering has proven its efficiency in many segmentation domains, such as edge or line detection. In this section, we are presenting a new segmentation method for corner detection based on causal filtering, anisotropic filtering and asymmetric filtering. The basic idea is inspired from a curvature-like operator similar to the Kitchen-Rosenfeld operator, but implemented using asymmetric anisotropic filtering. In addition, several anisotropic techniques are also presented for the elimination of false corner points.

8.2 Definition of a new "anisotropic curvature"

Here we are introducing an anisotropic curvature measure which is able to provide precise corner point detection by simply computing a curvature-like local maxima. First we recall

the well known Kitchen-Rosenfeld corner operator, and it's link to the MCM PDE scheme detailed in chapter 3, then we present the new "anisotropic curvature" operator which is an extension of the Kitchen-Roselfeld operator in an anisotropic way.

8.2.1 Kitchen-Rosenfeld corner operator (recall)

Considering a curve traced on the image plane, the curvature is defined as $\frac{d\theta}{ds}$ where θ is the tangent to the curve and s the curvilinear coordinate along the curve. As an image $I(x,\ y)$ is a Cartesian parametrized surface as Eq. 8.1:

$$z = I(x\,y) \tag{8.1}$$

An image can then be considered also as a set of isophotes lines, at each pixel, there is in isophote line going through this pixel. We can then define the dense (at each pixel) the curvature of isophote lines in an image is Eq. 8.2:

$$\frac{d\theta}{ds} = \frac{I_y^2 I_{xx} - 2I_x I_y I_{xy} + I_x^2 I_{yy}}{(I_x^2 + I_y^2)^{\frac{3}{2}}} \tag{8.2}$$

where :

I_x	:	Is the X derivative of the image $I(x,\ y)$
I_y	:	Is the Y derivative of the image $I(x,\ y)$
I_{xx}	:	Is the X 2^{nd} derivative of the image $I(x,\ y)$
I_{xy}	:	Is the XY cross 2^{nd} derivative of the image $I(x,\ y)$
I_{yy}	:	Is the Y 2^{nd} derivative of the image $I(x,\ y)$

This operator is the basis of the well-known corner operator "Kitchen-Rosenfeld" which is formed by the multiplication of the curvature of isophotes lines and the norm of the gradient. This formula means that a corner point must be an edge point with a strong curvature. Then the Eq. 8.3:

$$KR = \frac{I_y^2 I_{xx} - 2I_x I_y I_{xy} + I_x^2 I_{yy}}{(I_x^2 + I_y^2)^2} \tag{8.3}$$

It is also well-known that the Laplacian operator applied to an image can be written as Eq. 8.4:

$$\Delta I = I_{xx} + I_{yy} = \frac{I_y^2 I_{xx} - 2I_x I_y I_{xy} + I_x^2 I_{yy}}{(I_x^2 + I_y^2)^2} + \frac{I_x^2 I_{xx} + 2I_x I_y I_{xy} + I_y^2 I_{yy}}{(I_x^2 + I_y^2)^2} \tag{8.4}$$

Then as Eq. 8.5 :

$$\Delta I = I_{xx} + I_{yy} = I_{\xi\xi} + I_{\eta\eta} \tag{8.5}$$

It comes that $I_{\xi\xi}$ is the second derivative of the image along the direction of the isophote ξ and $I_{\eta\eta}$ is the second derivative of the image along the direction of the gradient η. As evidence KR is equivalent to $I_{\xi\xi}$.

8.2.2 Diffusion scheme

We have presented in the scale space section of this document two important diffusion schemes.

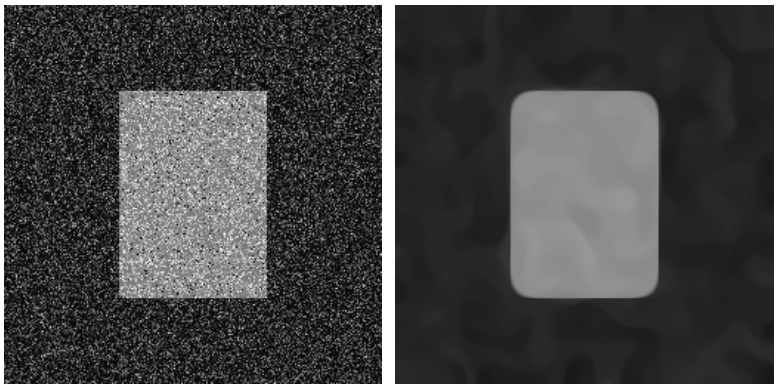
- The Euclidean linear scale space :
  this scale space is described by the heat equation, whose solution is a convolution of the original image with a Gaussian. Then the Kitchen-Rosenfeld operator can be seen as the curvature multiplied by the gradient at a certain level of diffusion.

- The Euclidean morphological scale space :
  This non-linear scale-space is obtained by applying the MCM scheme Eq. 8.6.

$$\begin{cases} \mathrm{I}(0,x,y) &= \mathrm{I}_0(x,y) \\ \frac{\partial \mathrm{I}}{\partial t}(t,x,y) &= \mathrm{I}_{\xi\xi}(t,x,y) \end{cases} \tag{8.6}$$

  where ξ still represent the tangent to isophotes.

  When iterating this scheme, isophote lines are moving in function of their Euclidean curvature. The Fig. 8.1 illustrates this property. If iterations goes on, the rectangle will be changed to a circle of decreasing radius.



(a) Initial image (b) Result of MCM diffusion (100 iterations)

Figure 8.1: MCM behavior. a) initial image, b) result of MCM diffusion (100 iterations).

It is clear that such diffusion schemes are moving in the corners as well as Gaussian scale space. In both schemes, the $\mathrm{I}_{\xi\xi}$ term is present.

8.2.2.1 An asymmetric diffusion scheme

In this study we will use a completely different scheme based on asymmetric diffusion (Montesinos and Magnier [2017]) which corresponds exactly to our needs of detecting corners precisely. Then we are going to show that one curvature-like measure that is used to drive the numerical scheme provides a better alternative to the $\mathrm{I}_{\xi\xi}$ operator for corner detection. The idea behind this work is summarized here. At each pixel P, are defined five distinct directions :

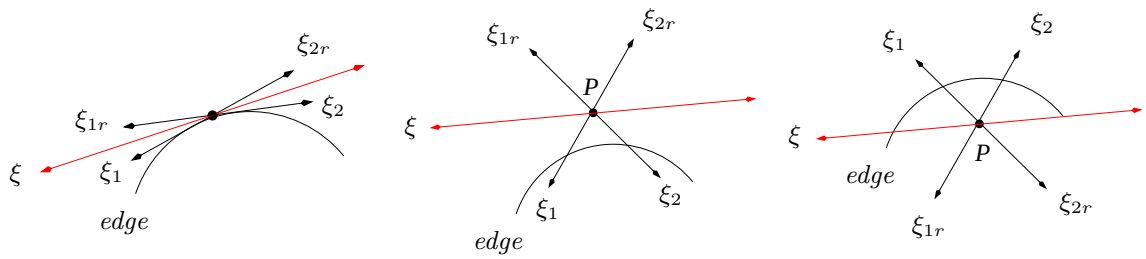
- ξ_1 and ξ_2 are the direction defined by the application of a bank of first derivative causal filters. These directions are the direction given by the smoothing component of the filter giving the external response (ξ_1 corresponds to the maximal positive response, ξ_2 to the minimal negative one).
- ξ is the orientation of the tangent to the isophote. This orientation is computed using ξ_1 and ξ_2.
- ξ_{1r} and ξ_{2r} are the mirrored orientations of ξ_1 and ξ_2 by the axis ξ (See Fig. 8.2a)).

This scheme can then be written as Eq. 8.7:

$$\begin{cases} \mathrm{I}(0,x,y) &= \mathrm{I}_0(x,y) \quad \longleftarrow \text{ initial image} \\ \frac{\partial \mathrm{I}}{\partial t}(t,x,y) &= \mathrm{I}_{\hat{\xi_1}\hat{\xi_2}}(t,x,y) = \underset{\mathrm{I}_{\xi_1\xi_2},\mathrm{I}_{\xi_{1r}\xi_{2r}},\mathrm{I}_{\xi\xi}}{\arg\min} |x| \end{cases} \tag{8.7}$$

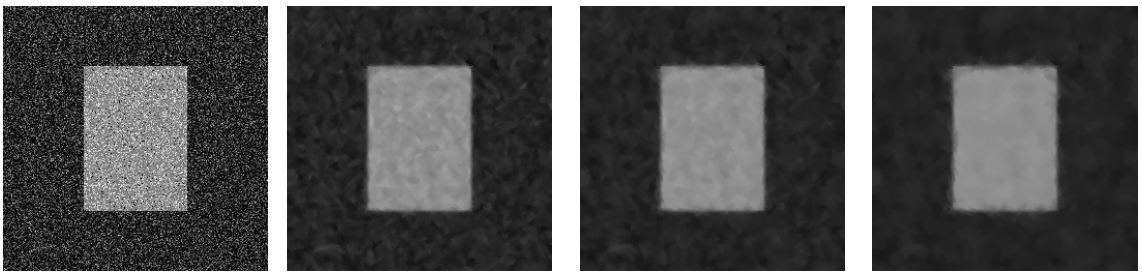
This scheme has a geometrical interpretation, illustrated at the Fig. 8.2. We have already seen that orientations ξ_1 and ξ_2 are influenced by the presence of edges, then at each pixel, we are searching for the directions that are the less influenced by edges in order to preserve these edges at most as possible by asymmetric diffusion. The diffusion will be achieved along ξ, along ξ_1 and ξ_2, or along ξ_{1r} and ξ_{2r}, in respect to the minimum absolute value of the asymmetric second derivative of the image. In the configuration of Fig 8.2b), or Fig. 8.2c) the diffusion is applied in the direction ξ_{1r} and ξ_{2r}, preserving the edges. On the configuration of Fig. 8.2a), the pixel under consideration is located on an edge, the diffusion may be either along ξ_1 and ξ_2 or simply along, ξ depending on the local curvature. For this scheme, the only parameter is the number of iterations.

For regularizing the input image, we just proceed to several iterations of this asymmetric scheme (in general, 100 iterations gives good results). The Fig. 8.3 presents results of regularization obtained on the "rectangle image" a), 8.3b) presents results obtained with 100 iterations, 200 iterations at c) and 500 iterations at d). As we can see, corners are not affected even with a high number of iterations.



(a) Pixel located on an edge (b) Pixel located outside edge (c) Pixel located inside edge

Figure 8.2: Causal Orientation. a) The pixel is located on an edge, the diffusion may be either along ξ_1 and ξ_2, b) Pixel is outside edge and the diffusion is applied in the direction ξ_{1r} and ξ_{2r}, preserving the edges, c) Pixel is inside edge and the diffusion is applied in the direction ξ_{1r} and ξ_{2r}, preserving the edges



(a) Initial image (b) Regularization 100 iterations (c) Regularization 200 iterations (d) Regularization 500 iterations

Figure 8.3: Asymmetric regularization. a) initial image, b) asymmetric regularization 100 iterations, c) asymmetric regularization 200 iterations, d) asymmetric regularization 500 iterations.

8.2.3 Asymmetric curvature

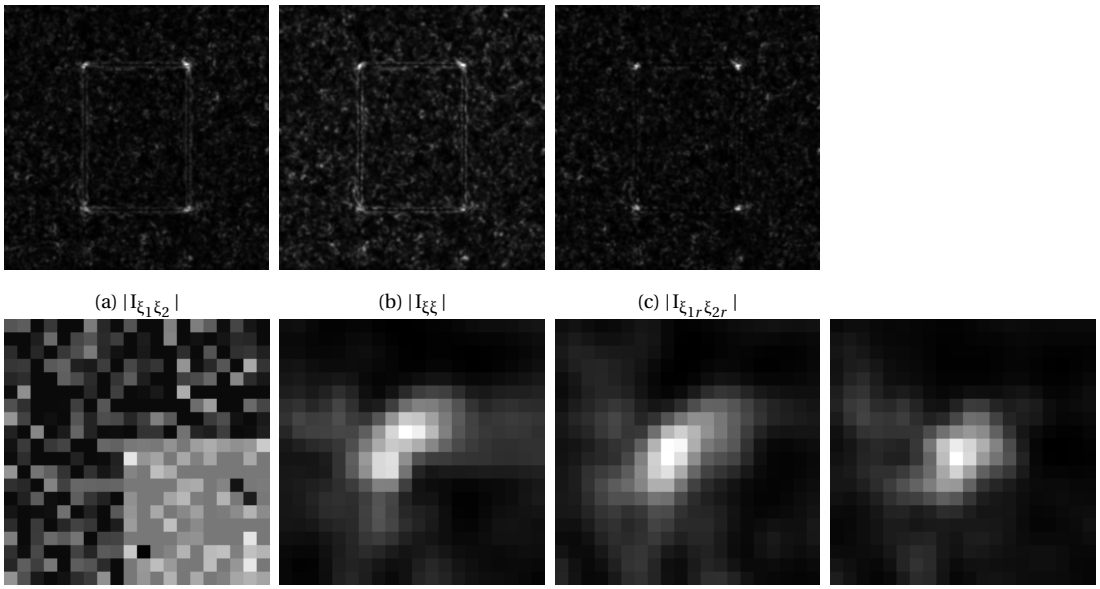
This scheme uses three curvature-like expressions to perform the diffusion :

- $|I_{\xi_1\xi_2}|$:
  a corner point is first an edge point with a smallest value of $|I_{\xi_1\xi_2}|$ under a certain neighborhood because ξ_1 and ξ_2 are both directions of isophotes then, at the corner point location, the three gray-levels involved in $|I_{\xi_1\xi_2}|$ are similar.

- $|I_{\xi\xi}|$:
  this measure is similar to the Kitchen-Rosenfeld measure (direction ξ may be somewhat different because filters involved are different) estimated locally in a 3×3 window. This measure is suitable but not optimal, the next measure will be preferred.

- $|I_{\xi_{1r}\xi_{2r}}|$:
  this measure is maximized since the directions, ξ_{1r} and ξ_{2r} indicates the directions where gray-levels are the most different from the considered pixel. For this reason, the response obtained is less noisy than the one obtained with the preceding measure.

Then, for characterizing the curvature at corner points, we have chosen to use the expression $|I_{\xi_{1r}\xi_{2r}}|$ that best characterizes the corners. The Fig. 8.4 presents the three asymmetric curvature results obtained after 100 iterations. As explained beforehand, the $|I_{\xi_1\xi_2}|$ (Fig. 8.4a)) gives 2 responses at each side of corners. This measure is minimum at corner points, for this reason it will be complicated to use such measure to characterize correctly corners. The $|I_{\xi\xi}|$ (Fig. 8.4b)) is similar to a Kitchen Rosenfeld measure, the response is overall noisy and edges are also responding. Finally, the $|I_{\xi_{1r}\xi_{2r}}|$ (Fig. 8.4c)) measure gives the best information able to characterize reliably the corner points.



(a) $|I_{\xi_1\xi_2}|$ (b) $|I_{\xi\xi}|$ (c) $|I_{\xi_{1r}\xi_{2r}}|$

(d) Original image (e) $|I_{\xi_1\xi_2}|$ measures (f) $|I_{\xi\xi}|$ measures (g) $|I_{\xi_{1r}\xi_{2r}}|$ measures

Figure 8.4: Anisotropic curvature measures obtained after 100 iterations. a) $|I_{\xi_1\xi_2}|$, b) $|I_{\xi\xi}|$, c) $|I_{\xi_{1r}\xi_{2r}}|$. d), e), f), g) Details in the upper left corner (d) original image) e, f, g) respectively $|I_{\xi_1\xi_2}|$, $|I_{\xi\xi}|$, and $|I_{\xi_{1r}\xi_{2r}}|$ measures.

8.2.4 Corner detection using $|I_{\xi_{1r}\xi_{2r}}|$

The complete algorithm for asymmetric corner detection is summarized as follows :

1. Depending on the precision needed, magnify or not the initial image (Montesinos and Datteny [1997]) using a very small Gaussian standard-deviation ($\sigma \simeq 0.6$), apply several iterations (in general 10) of an heat inverse equation scheme then apply several iterations (in general 4) of a shock filter (Osher and Rudin [1990]).

2. Apply several iterations of the asymmetric scheme (in general 100).

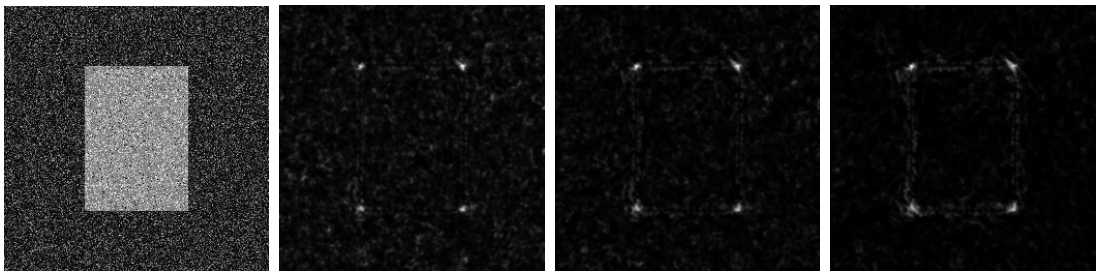
3. Compute the $|I_{\xi_{1r}\xi_{2r}}|$ image after the regularization.

4. Compute the local maxima of the $|I_{\xi_{1r}\xi_{2r}}|$ for example in a circular window (generally with a radius of 2 pixels multiplied by the precision).

8.2.4.1 Results obtained on the "rectangle" image

The Fig. 8.5 presents the $|I_{\xi_{1r}\xi_{2r}}|$ operator results obtained on the “rectangle” image at simple pixel precision, after varying number of iterations (100, 200, 500). As iterations go on, the noise is filtered. For comparison, the Fig. 8.6 presents the Kitchen-Rosenfeld operator results obtained with a Gaussian standard-deviation $\sigma = 1$ and $\sigma = 3$. The Kitchen-Rosenfeld operator gives noisy results. When computed with a Gaussian filter having a parameter σ equal to 1 the results obtained are generally too noisy to obtain interesting results (Fig. 8.6b)). If the parameter σ increases, it is possible to obtain a more reliable information, but local maxima are moving (Fig. 8.6c)).

Finally, the Fig. 8.7 present results obtained by local maxima extraction for all operators : Harris ($\sigma = 1$), Kitchen-Rosenfeld ($\sigma = 3$) and $|I_{\xi_{1r}\xi_{2r}}|$ with 100, 200 and 500 iterations. As discussed beforehand, Harris and Kitchen-Rosenfeld give noisy results. Moreover, for Kitchen-Rosenfeld, corners are often detected at a distance greater than 2 pixels from the true location. Increasing the Gaussian parameter σ improves the curvature SNR, but precision of corner localization decreases. Concerning the $|I_{\xi_{1r}\xi_{2r}}|$ operator, the response is less noisy, and the precision seems better than Harris (around 1 pixel from the true corner locations).

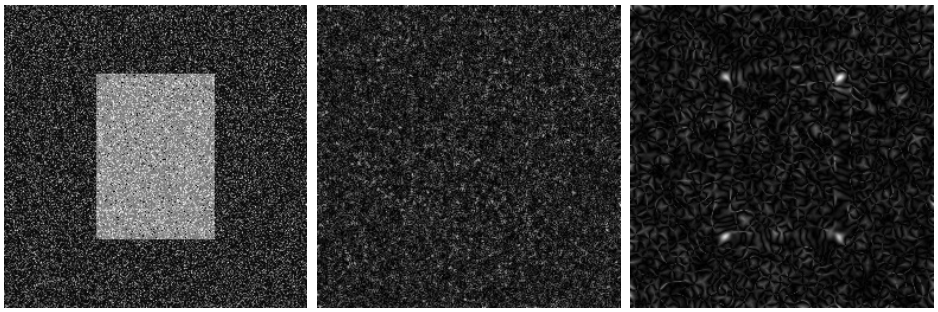


(a) Original image (b) $|I_{\xi_{1r}\xi_{2r}}|$ at 100 iterations (c) $|I_{\xi_{1r}\xi_{2r}}|$ at 200 iterations (d) $|I_{\xi_{1r}\xi_{2r}}|$ at 500 iterations

Figure 8.5: Asymmetric curvature $|I_{\xi_{1r}\xi_{2r}}|$ on image "rectangle". a) original image, b) $|I_{\xi_{1r}\xi_{2r}}|$ at 100 iterations, c) $|I_{\xi_{1r}\xi_{2r}}|$ at 200 iterations, d) $|I_{\xi_{1r}\xi_{2r}}|$ at 500 iterations.

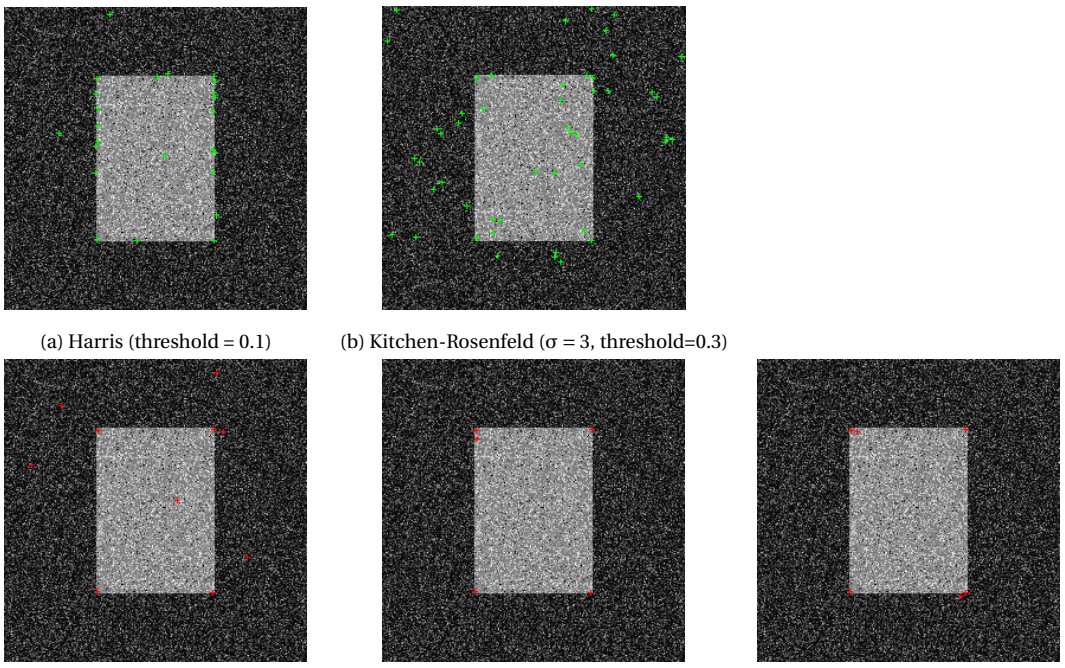
8.2.4.2 Results obtained on the "inria" image

The Fig. 8.8 compares the Harris operator and the anisotropic curvature operator on the "inria" image : a) shows the initial image, b) presents regularization results (100 iterations) and c) presents the $|I_{\xi_{1r}\xi_{2r}}|$ operator result. The Fig. 8.8 d) and e) show respectively the results of the Harris corner detector and the $|I_{\xi_{1r}\xi_{2r}}|$ corner detection. Then the Fig. 8.8 f) present 4 manually corner selection and the results obtained with both operators are going to be detailed in g) to n). In (g) h)), (i) j)), and (m) n)), corner angle is less or equal to 90°, the new operator performs better than Harris. For (k) l)) corner angle is 108° (angle is wider) the results are similar. But if angle value increases Harris completely loses the corner point, The new operator still performs correctly (see the wide angles of the black carpet corners on the floor).



(a) Original image (b) Kitchen-Rosenfeld ($\sigma = 1$) (c)Kitchen-Rosenfeld ($\sigma = 3$)

Figure 8.6: Kitchen-Rosenfeld operator on image "rectangle". a) original image, b) Kitchen-Rosenfeld ($\sigma = 1$) regularization is not enough to obtain reliable curvature, c) Kitchen-Rosenfeld ($\sigma = 3$) curvature appears, but noise is still present and strong.



(a) Harris (threshold = 0.1) (b) Kitchen-Rosenfeld ($\sigma = 3$, threshold=0.3)

(c) Anisotropic (100 ite, threshold = 0.3) (d) Anisotropic (200 ite, threshold = 0.3) (e) Anisotropic (500 ite, threshold = 0.3

Figure 8.7: Corner detection. a) Harris corner detector (threshold = 0.1), b) Kitchen-Rosenfeld corner detector ($\sigma = 3$, threshold=0.3), c) Anisotropic corner detector (100 iterations, threshold = 0.3), d) Anisotropic corner detector (200 iterations, threshold = 0.3), e) Anisotropic corner detector (500 iterations, threshold = 0.3.

8.2.4.3 Sub-pixel precision, results image "toys"

The Fig. 8.9 compares Harris corner detector (threshold=0.001) and $|I_{\xi_{1r}\xi_{2r}}|$ corner detection (threshold=0.05) at pixel precision. We are interested here on the results obtained on the small windows of the central house (windows size is around 5×5 pixels). Harris operator (Fig. 8.9 b)) gives many responses on the windows, but the points detected are often on the windows frame rather than the corner. For the $|I_{\xi_{1r}\xi_{2r}}|$ (Fig. 8.9 c)) corner detection, the detected point is more often at the corner. On the large, dark windows of the left house, Harris performs better.

The Fig. 8.10 present results obtained at precision=2 ($\frac{1}{2}$ pixel). In Fig. 8.10 a) presents the sub-pixel ($\frac{1}{2}$ pixel) Harris corner detector, and in Fig. 8.10 b) presents the sub-pixel ($\frac{1}{2}$ pixel) $|I_{\xi_{1r}\xi_{2r}}|$ corner detector. The Fig. 8.10 c) and d) present the details only a region of

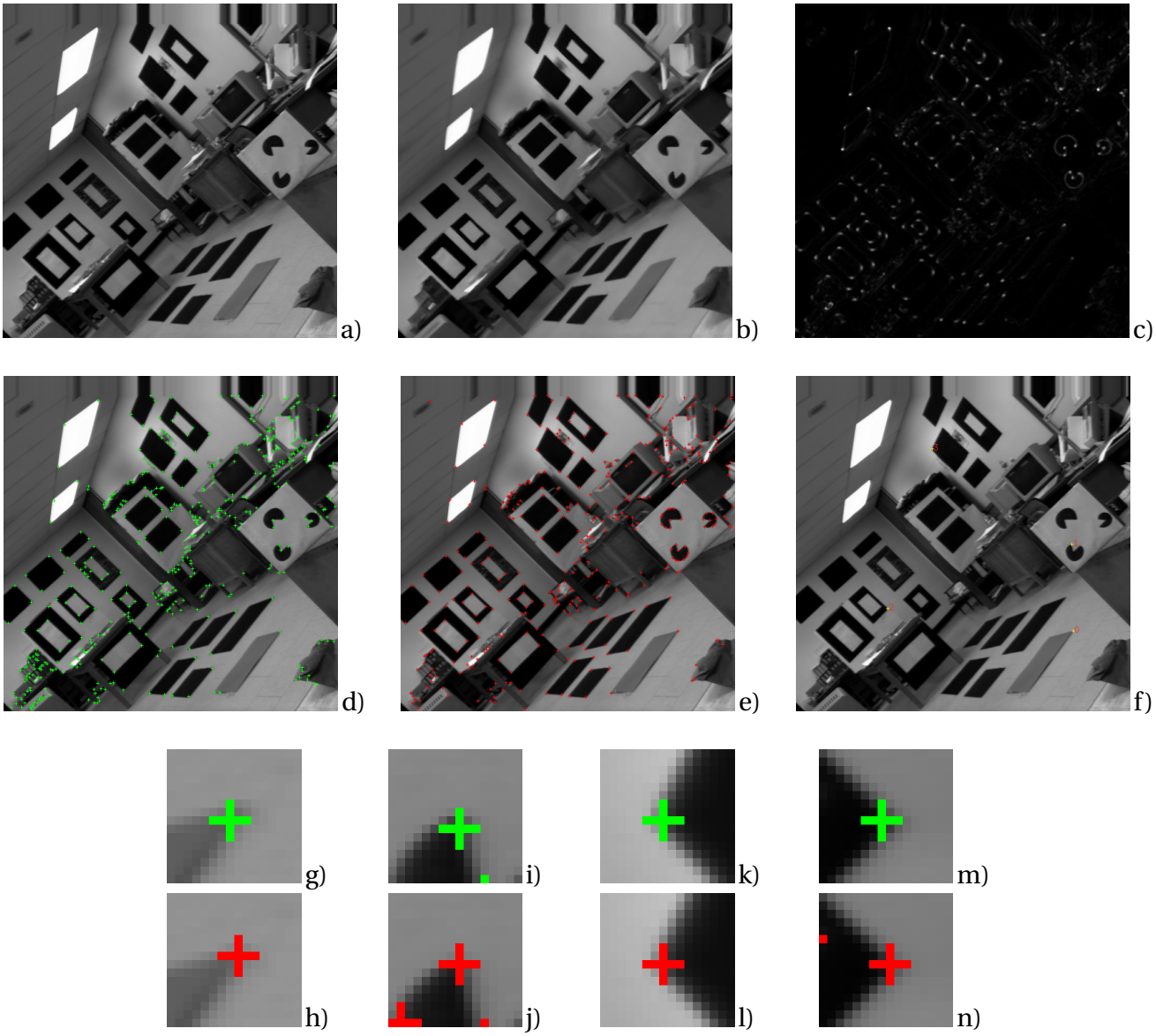
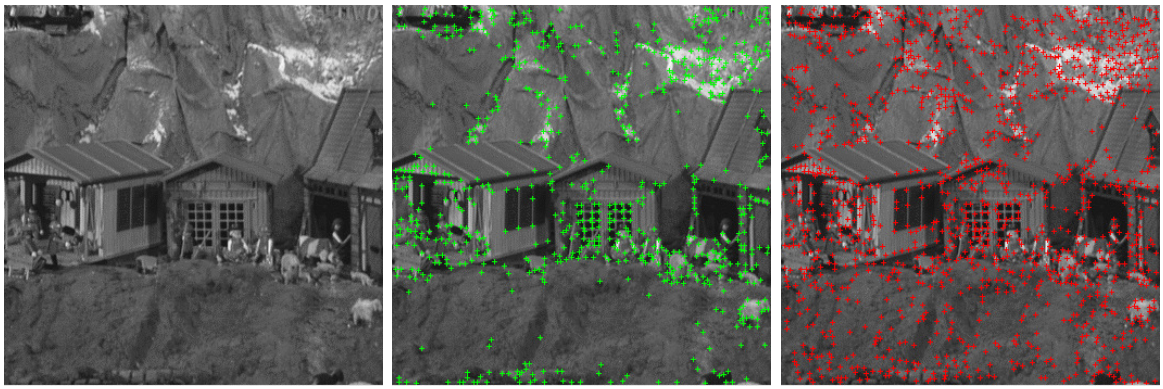


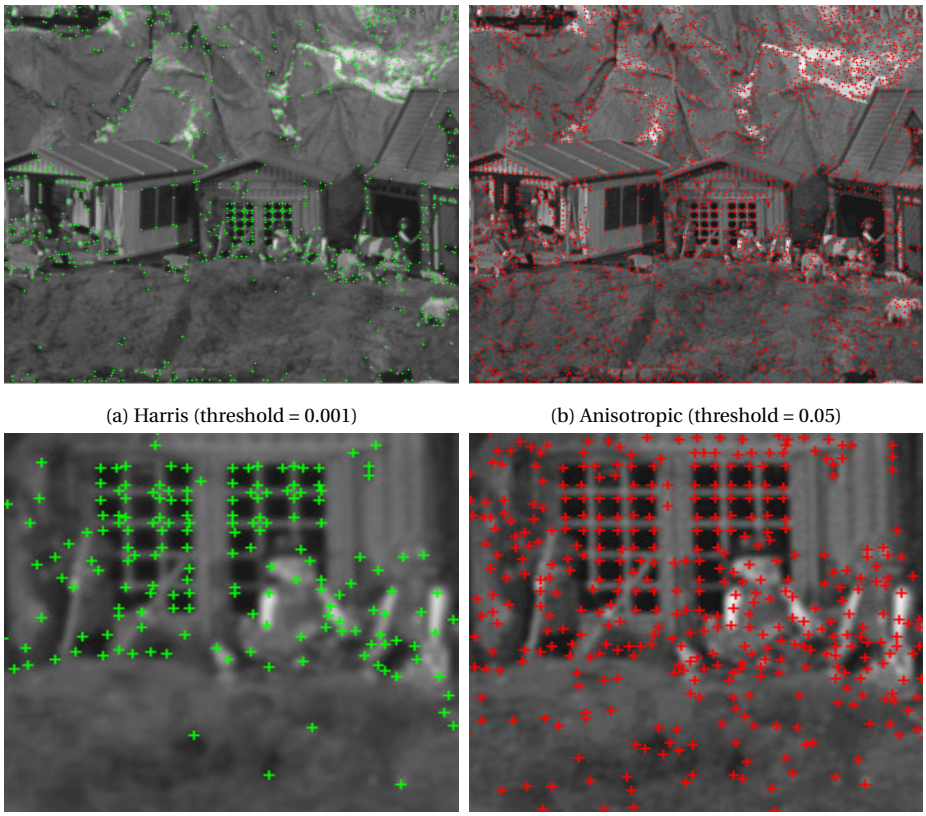
Figure 8.8: *Corner detection image "inria". a) Harris corner detection (threshold = 0.001), b) Anisotropic corner detection (threshold = 0.1). c) Detail of corner detection at location (424, 390). d) Detail of corner detection at location (421, 257). e) Detail of corner detection at location (209, 112). f) Detail of corner detection at location (140, 355).*

interest 200 × 200 on the windows of the central house (c) Harris, d) $|I_{\xi_{1r}\xi_{2r}}|$.



(a) Initial image (b) Harris (threshold = 0.001 (c) Anisotropic (threshold = 0.05)

Figure 8.9: Corner detection image "toys", pixel precision. a) initial image. b) Harris corner detector (threshold = 0.001), c) Anisotropic corner detector (threshold = 0.05).



(a) Harris (threshold = 0.001) (b) Anisotropic (threshold = 0.05)

(c) Harris (window details) (d) Anisotropic (window details)

Figure 8.10: Corner detection image "toys", precision = $\frac{1}{2}$ pixel. a) Harris corner detector (threshold = 0.001), b) Anisotropic corner detector (threshold = 0.05), c) Harris corner detector (window details), d) Anisotropic corner detector (window details).

8.3 Anisotropic corner detector

We propose here a new corner operator based on anisotropic filtering, we are going to see that it could be used in conjunction with the preceding corner detector based on anisotropic curvature leading to a robust corner detector. This operator is inspired by the Harris operator. The Harris operator in based on the structure tensor as Eq. 8.8;

$$T = \begin{pmatrix} (I_x^2) * G_{\sigma_s} & (I_x I_y) * G_{\sigma_s} \\ (I_x I_y) * G_{\sigma_s} & (I_y^2) * G_{\sigma_s} \end{pmatrix} \tag{8.8}$$

which represents the local auto-correlation of the image signal. I_x and Y_y are image derivatives along respectively X and Y axis. They are obtained using a Gaussian derivative filter with a σ_d standard-deviation.
G_{σ_s} is a Gaussian smoothing filter with a σ_s standard-deviation.
Then the Harris operator is obtained from this tensor as Eq. 8.9:

$$H = Det(T) - k \cdot Trace(T)^2 \tag{8.9}$$

- The determinant is the product of eigen values, it is strong when both eigen values are strong.
- The trace is the sum of eigen values and is strong at edges.
- The parameter k has been determined empirically to do a balance between corners and edges. k is generally set to 0.04.

After the computation of this operator, the negative values are suppressed and finally the points of interest are extracted as local maxima of the response obtained (a result has already been presented in Fig. 8.8 d)).

8.3.1 Anisotropic corners

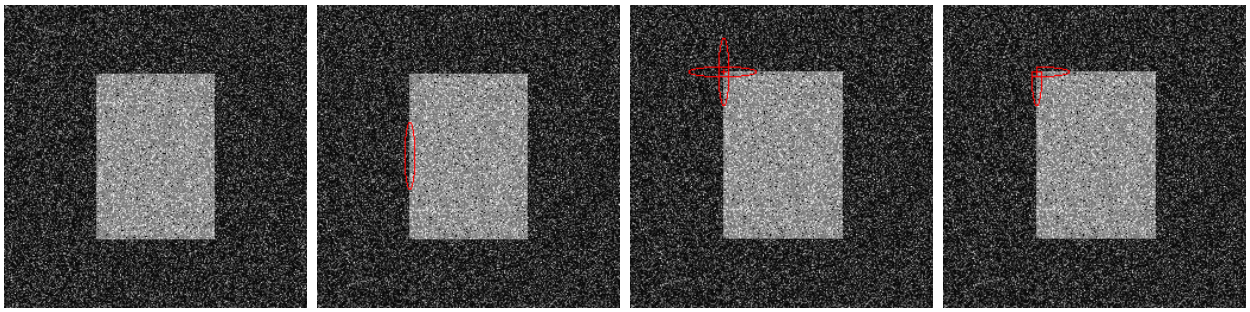
This new method is partially inspired by Harris operator, in subtracting a corner-edge detector with a strictly edge detector. The idea sustaining this new method is that anisotropic causal filters does not delocalize corners while classical anisotropic filters does. Then the corner-edge detector can be simply a causal gradient and the edge detector a classical anisotropic gradient. Then corners can be obtained by the difference of these two gradients : the first one is obtained by causal anisotropic filtering, as the second one is obtained by classical anisotropic filtering (not causal). The solution proposed here is based on causal filtering, such as Eq. 8.10:

$$Q1(x,y,\theta) = I_\theta * C_1 x H(y) e^{-\frac{x^2}{2\sigma_\eta^2} - \frac{y^2}{2\sigma_\xi^2}} \tag{8.10}$$

and anisotropic filtering :

$$Q2(x,y,\theta) = I_\theta * C_2 x e^{-\frac{x^2}{2\sigma_\eta^2} - \frac{y^2}{2\sigma_\xi^2}} \tag{8.11}$$

This property is illustrated on an image of a rectangle at the Fig. 8.11. We can see that for the classic anisotropic filters, at an edge location $\frac{1}{2}$ of the filter is outside the rectangle (Fig. 8.11b), at a corner location $\frac{3}{4}$ of the filter is outside the rectangle (Fig. 8.11c) then the gradient at the corner location is affected by local curvature. On the contrary, for the causal filter this ratio remains of $\frac{1}{2}$ for both extremal response kernels (Fig. 8.11d).



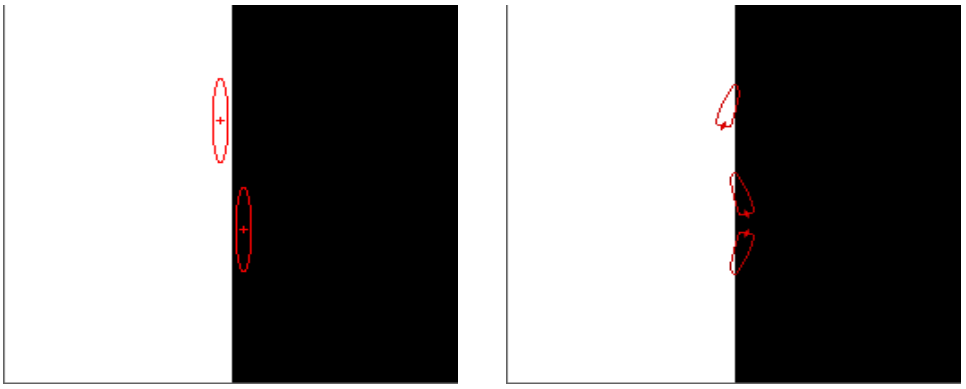
(a) Initial image (b) Classic anisotropic at edge (c) Classic anisotropic at corner (d) Causal anisotropic at corner

Figure 8.11: Input synthetic image of a rectangle. a) Initial image. b) classic anisotropic kernel located at an edge point, c) classic anisotropic kernels located at a corner point, and d) causal anisotropic kernels located at a corner point.

8.3.2 Filter normalization

We have seen in the previous section that we are going to apply difference of "gradients" obtained using kernels of different types : causal anisotropic Gaussian kernels and classical anisotropic Gaussian kernels. By the term "gradient", we mean a non-linear operation consisting of computing at each pixel the response to the filter having the maximum value minus the response to the filter having the minimum value over all angles.

If we want to be able to compare the response of filter banks of different types, it is fundamental for the filters to be normalized correctly. Due to the nature of causal filters, the gradient obtained is wider than the gradient obtained with classical anisotropic kernels. This property is illustrated in Fig. 8.12. Fig. (8.12a) present the behavior of a classical filter near edge points, the orientation corresponding to the maximal response corresponds to tangent to the contour. But for causal filters, as the filter support is half the support of a classical filter, the orientation of the maximal response is strongly influenced by the presence of the edge and this orientation does not, in general, correspond to the tangent of the contour.



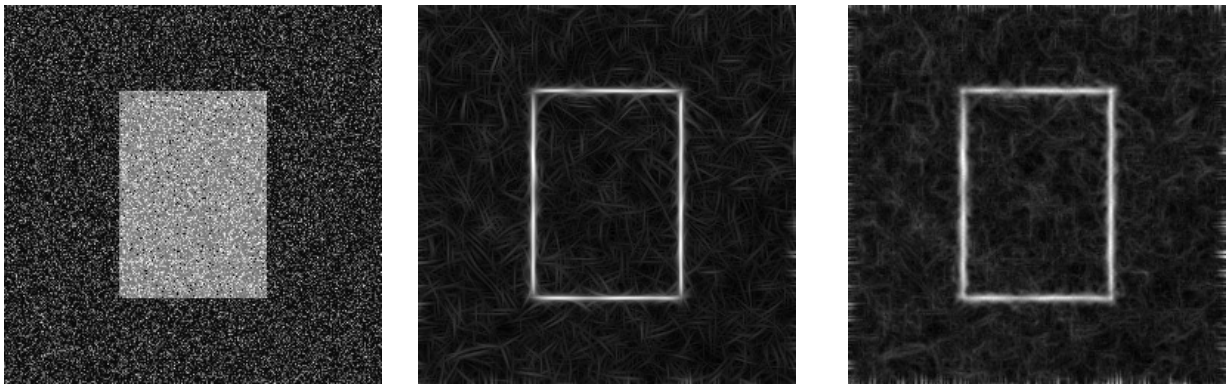
(a) Classical anisotropic kernels located near edge points (b) Causal anisotropic kernels located near edge points

Figure 8.12: Classic vs Causal anisotropic kernel at edge points: a) classical anisotropic kernels located near edge points b) Causal anisotropic kernels located near edge points b).

For this reason, the gradient obtained with causal filters is wider than the gradient obtained with classical anisotropic filters. Also, the noise remaining after application of causal filters is more important than the noise remaining after application of classical anisotropic filters. But in practice the noise does not really lead to problems, even for

causal filters, since the kernels used are generally large and then the remaining noise is often negligible.

The Fig. 8.13 presents the behavior of both classical and causal filters. This figure also illustrates the behavior of both filters at corner points. The corners are well-defined at Fig. 8.13c) at the contrary, the gradient is vanishing at corner points for classical filters 'see Fig. 8.11c)).



(a) Original noisy image of a rectangle (b) Gradient (max - min) via classical anisotropic (c) Gradient (max - min) via causal anisotropic

Figure 8.13: Gradient obtained with classical filters and causal filters. a) original noisy image of a rectangle. b) Gradient (max - min) obtained using classical anisotropic filters. c) Gradient (max - min) obtained using causal anisotropic filters.

8.3.2.1 Normalization process

As the gradients obtained with the different types of filters does not have the same thickness, for filter normalization it is necessary to consider both amplitude of response and standard-deviation of Gaussian. We consider here the parameters of causal filters as the main parameters of our new corner detector :

- σ_ξ the largest standard deviation of causal Gaussian (in general chosen between 6 and 10),
- σ_η the smallest standard deviation of causal Gaussian (in general chosen between .7 to 2,

Firstly, the σ_ξ must be identical for causal and classical filters : we have to compare things at the same scale (even if causal filters does not strictly belong to a scale space). In a second time, we set the σ_η standard deviation of causal Gaussian then we have to adapt the $\sigma_{\eta 2}$ standard deviation for classical filters, which must be larger than σ_η (this parameter depends both on σ_ξ and σ_η). The amplitude of the filters is computed at the same time. For this normalization, we have pre-computed two matrices (or tables) :

- The first one $M_{\sigma_{\eta 2}}$ gives $\sigma_{\eta 2}$ function of σ_η and σ_ξ

$$\sigma_\eta \left\{ \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \sigma_{\eta 2} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \right.$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\sigma_\xi}$$

- The second one M_{norm} gives the normalization factor to be applied to the classical anisotropic filter function of σ_η and σ_ξ

$$\sigma_\eta \left\{ \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & norm_2 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \right.$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\sigma_\xi}$$

The values stored in these matrices are obtained by a mean squares optimization of the gradient computed with classical filters towards the gradient computed with causal filter. We consider an image containing an edge (Heaviside function) for example a vertical edge,

- for each values of σ_η and σ_ξ (discretized by steps of 0.1)
  - The causal anisotropic gradient is computed using the filter bank having σ_η and σ_ξ as parameters,
  - Iteratively, we are seeking for the classical gradient having $\sigma_{\eta 2}$ and the same σ_ξ such that the mean square error is minimal, the corresponding normalization factor is computed at the same time.

As an example, we give here the first elements of a $M_{\sigma_{\eta 2}}$ matrix for $\sigma_\eta \in [0.7\ 7.0]$ and $\sigma_\xi \in [2.0\ 14.0]$ with for a discretization angle of 5° :

σ_ξ

$\downarrow$

$$
\begin{array}{c}
\\
2.0 \rightarrow \\
2.1 \rightarrow \\
2.2 \rightarrow \\
2.3 \rightarrow \\
2.4 \rightarrow \\
2.5 \rightarrow \\
2.6 \rightarrow \\
\vdots \\
6.0 \rightarrow \\
\vdots \\
7.0 \rightarrow \\
\vdots \\
8.0 \rightarrow \\
\vdots \\
9.0 \rightarrow \\
\vdots \\
10.0 \rightarrow \\
\vdots \\
\\
\sigma_\eta \rightarrow
\end{array}
\begin{pmatrix}
1.130000 & 1.220000 & 1.310000 & 1.400000 & 0.0 & 0.0 & 0.0 & 0.0 & \cdots \\
1.160000 & 1.250000 & 1.340000 & 1.440000 & 0.0 & 0.0 & 0.0 & 0.0 & \cdots \\
1.180000 & 1.280000 & 1.370000 & 1.470000 & 1.550000 & 0.0 & 0.0 & 0.0 & \cdots \\
1.200000 & 1.300000 & 1.400000 & 1.500000 & 1.590000 & 0.0 & 0.0 & 0.0 & \cdots \\
1.230000 & 1.330000 & 1.430000 & 1.520000 & 1.620000 & 1.710000 & 0.0 & 0.0 & \cdots \\
1.250000 & 1.350000 & 1.450000 & 1.550000 & 1.640000 & 1.740000 & 0.0 & 0.0 & \cdots \\
1.270000 & 1.370000 & 1.470000 & 1.570000 & 1.670000 & 1.770000 & 1.860000 & 0.0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
1.570000 & 1.700000 & 1.830000 & 1.960000 & 2.090000 & 2.230000 & 2.370000 & 2.510000 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
1.610000 & 1.740000 & 1.870000 & 2.000000 & 2.140000 & 2.290000 & 2.430000 & 2.580000 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
1.630000 & 1.760000 & 1.890000 & 2.030000 & 2.180000 & 2.330000 & 2.480000 & 2.630000 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
1.660000 & 1.790000 & 1.920000 & 2.060000 & 2.210000 & 2.360000 & 2.520000 & 2.670000 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
1.690000 & 1.820000 & 1.950000 & 2.090000 & 2.250000 & 2.410000 & 2.560000 & 2.710000 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \\
0.7 & 0.8 & 0.9 & 1.0 & 1.1 & 1.2 & 1.3 & 1.4 & \cdots
\end{pmatrix}
$$

We can see for example that for a combination of $\sigma_\xi = 2.2$ and $\sigma_\eta = 0.8$ we must use a classical filter bank with $\sigma_\xi = 2.2$ and $\sigma_{\eta 2} = 1.28$ as parameters. We also print most commonly used parameters : from $\sigma_\xi = 6.0$ to $\sigma_\xi = 10.0$ and from $\sigma_\eta = 0.7$ to $\sigma_\eta = 1.4$

The corresponding normalization matrix only for common parameters (for a discretization angle of 5°) is also given :

σ_ξ

$\downarrow$

$$
\begin{array}{c}
\vdots \\
6.0 \rightarrow \\
\vdots \\
7.0 \rightarrow \\
\vdots \\
8.0 \rightarrow \\
\vdots \\
9.0 \rightarrow \\
\vdots \\
10.0 \rightarrow \\
\vdots \\
\\
\sigma_\eta \rightarrow
\end{array}
\begin{pmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
1.680354 & 1.683072 & 1.681973 & 1.676716 & 1.668353 & 1.664730 & 1.659319 & 1.652792 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
1.715444 & 1.716456 & 1.713684 & 1.706756 & 1.703952 & 1.705199 & 1.697689 & 1.695307 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
1.733078 & 1.733181 & 1.729565 & 1.729313 & 1.732448 & 1.732235 & 1.729750 & 1.725763 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
1.759554 & 1.758366 & 1.753457 & 1.752002 & 1.753823 & 1.752518 & 1.755424 & 1.750118 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
1.786047 & 1.783575 & 1.777394 & 1.774617 & 1.782518 & 1.786345 & 1.781085 & 1.774535 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \\
0.7 & 0.8 & 0.9 & 1.0 & 1.1 & 1.2 & 1.3 & 1.4 & \cdots
\end{pmatrix}
$$

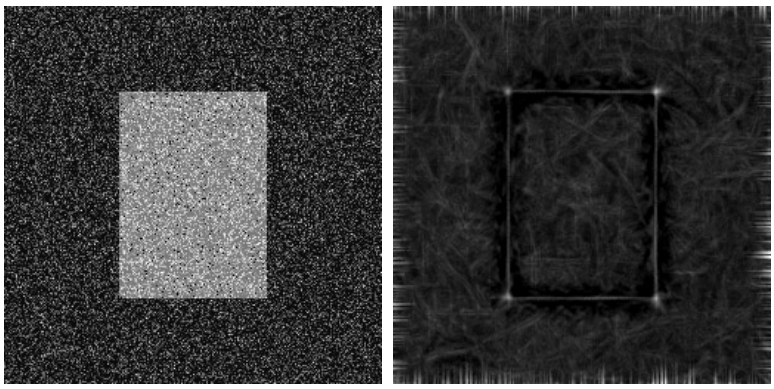
8.3.3 A new anisotropic corner measure

The algorithm is quite straightforward : Considering the parameters, standard-deviations σ_ξ and σ_η plus the discretization angle $\Delta\theta$:

1. Load the normalization matrices.

2. Apply causal filter bank (σ_ξ, σ_η, $\Delta\theta$).
3. Compute causal gradient (max - min) over all causal filter responses : $\rightarrow I_{cg}$.
4. Apply classical filter bank (σ_ξ, $\sigma_{\eta 2}$ (sigma matrix used), $\Delta\theta$).
5. Compute classical gradient (max - min) over all causal filter responses.
6. Normalize classical gradient using norm matrix) : $\rightarrow I_g$.
7. Subtract gradients : $I_{cg} - I_g \rightarrow I_c$.
8. Suppress negative values of $I_c \rightarrow I_{c0}$.

The figure (Fig. 8.14b)) presents the enhanced corner image I_{c0} obtained on the "rectangle" image (Fig. 8.14a)) using $\sigma_\xi = 10$, $\sigma_\eta = 1$ and $\Delta\theta = 5°$ Then, the corner points can



(a) Original noisy image of a rectangle (b) Enhanced corner image

Figure 8.14: Anisotropic corner operator. a) Original noisy image of a rectangle. b) Enhanced corner image.

be simply extracted by searching for local maxima of the I_{c0} image, for example in a 3×3 window. Unfortunately, as seen in (Fig. 8.14b) the edge points are not completely suppressed by the subtraction of gradient images (of course, the normalization process is an approximation). From this remark, we can easily imagine that finding a suitable threshold for obtaining only corners in a complex image could become difficult. To get rid of this problem, we modify slightly the point 7 of the preceding method by introducing a parameter in the subtraction which provide the capability to remove non-stable points.

8.3.4 Removing non-stable points

The first enhancement to this method goes through the elimination of non-stable points using two linear combination parameters. This is achieved by modifying the preceding algorithm and replacing the two last steps (7 and 8). We add two new parameters t_1 and t_2 for a better control of the linear combination at the gradient subtraction steps. These coefficients are modifying the edge suppression stage, then non-stable points located on edges may slightly move, and can be easily eliminated. The new algorithm become :

7. Subtract gradients : $I_{cg} - t_1 I_g \rightarrow I_c$.
8. Suppress negative values of $I_c \rightarrow I_{c0}$.

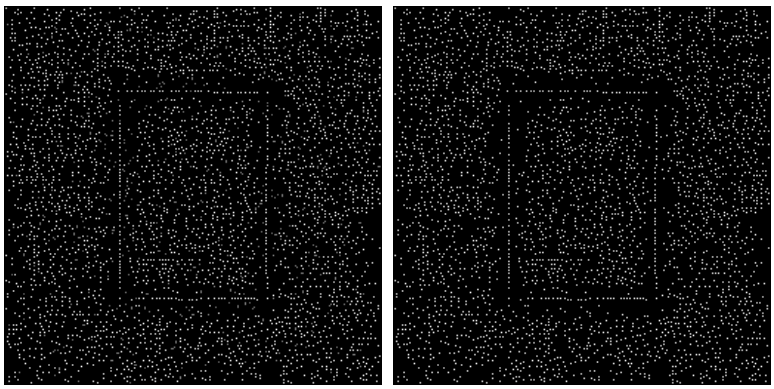
9. Extract local maxima of I_{c0} $\rightarrow I_{m1}$.
10. Subtract gradients : $I_{cg} - t_2 I_g \rightarrow I_c$.
11. Suppress negative values of I_c $\rightarrow I_{c0}$.
12. Extract local maxima of I_{c0} $\rightarrow I_{m2}$.
13. Keep stable points over the 2 thresholds $\rightarrow I_m$.

8.3.5 Combining operators

Combining corners operators can simply be achieved by directly multiplying this anisotropic operator described in section 8.3.3 with the $|I_{\xi_{1r}\xi_{2r}}|$ curvature operator. Then the corners can be extracted by removing negative values and computing local maxima of the combined operator.

8.3.5.1 Anisotropic corner detection results

We present here the results obtained using this new method on several images. The Fig. 8.15 presents the results obtained with this variant, the threshold used for local maximization is very small : 0.00001. As can be seen at the Fig. 8.15 this variant is able to remove most of false corner points, especially around edges which are more affected than others by the coefficients in the subtractions. The gray points on figure, Fig. 8.15a) are those that are not stable and are eliminated (Fig. 8.15b)). As demonstrated by the results obtained



(a) All points (in gray, non-stable points (b) Remaining stable points

Figure 8.15: Effect of varying the subtraction coefficients $t_1 = 0.9$ and $t_2 = 1.1$. a) All points (in gray, non-stable points. b) Remaining stable points.

here, we still have to find a robust solution to remove edge points and noise. This can be achieved by simply considering the anisotropic curvature that have been presented at section 8.2.3. The Fig. 8.16 presents the results obtained by the anisotropic corner detection on the image “rectangle”. The result obtained in Fig. 8.15b) is simply multiplied by the $|I_{\xi_{1r}\xi_{2r}}|$ curvature and thresholded (here threshold is 0.1). Here the noise is very strong and corners obtained within this method are more precise than the one obtained with anisotropic curvature. Now we are going to present results on real images obtained at pixel and $\frac{1}{2}$ pixel precision. The Fig. 8.17 show the results obtained on image "inria" at pixel precision. This experiment on real data shows that the quality of the results depends strongly on the size and proximity of structures in the corner neighborhood. The

first method based on anisotropic curvature performs better on small structures such as the windows on the "toys image", especially at sub-pixel precision but the precision decreases as the noise increases. At the contrary, this second method performs better than the first one in case of important noise. Then we are going to compare here three corner detectors :

- Harris corner detector : in Fig. 8.17a) local maxima threshold is set to 0.001.
- The method eliminating non-stable points presented here is applied at Fig. 8.17b) with $t_1 = 0.9$ and $t_2 = 1.1$ local maxima thresholds.
- A hybrid method fusing data by a simple multiplication at operator level of the anisotropic corner detection (with no elimination of non-stable points : only one threshold $t = 1.0$) and the anisotropic curvature operator $|I_{\xi_{1r}\xi_{2r}}|$. Then local maxima is computed with a threshold of 0.02. The Fig. 8.17c) shows the results of this combination.

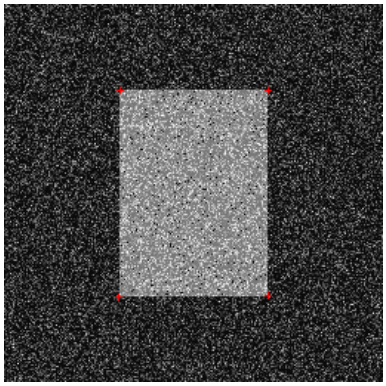
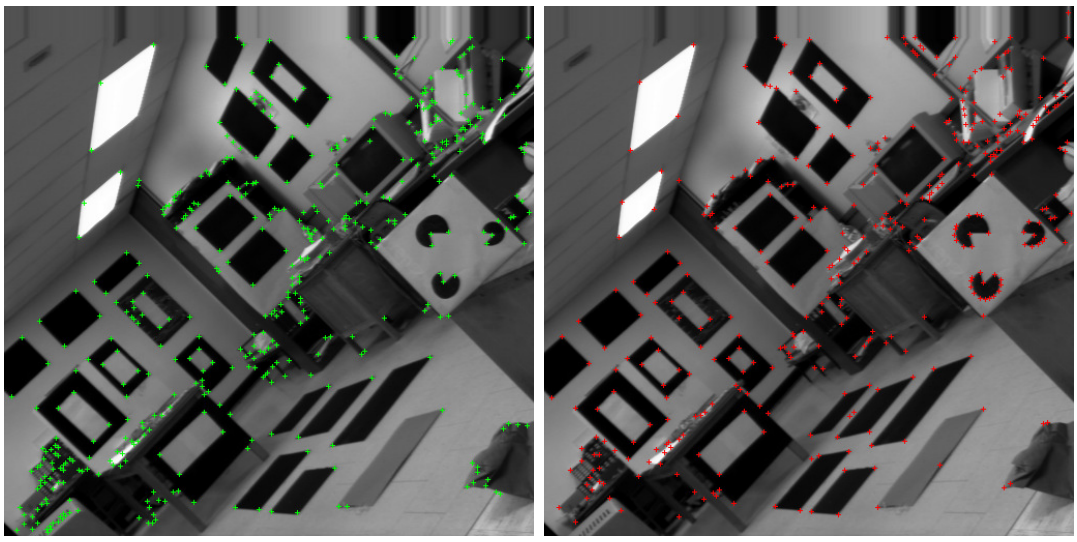
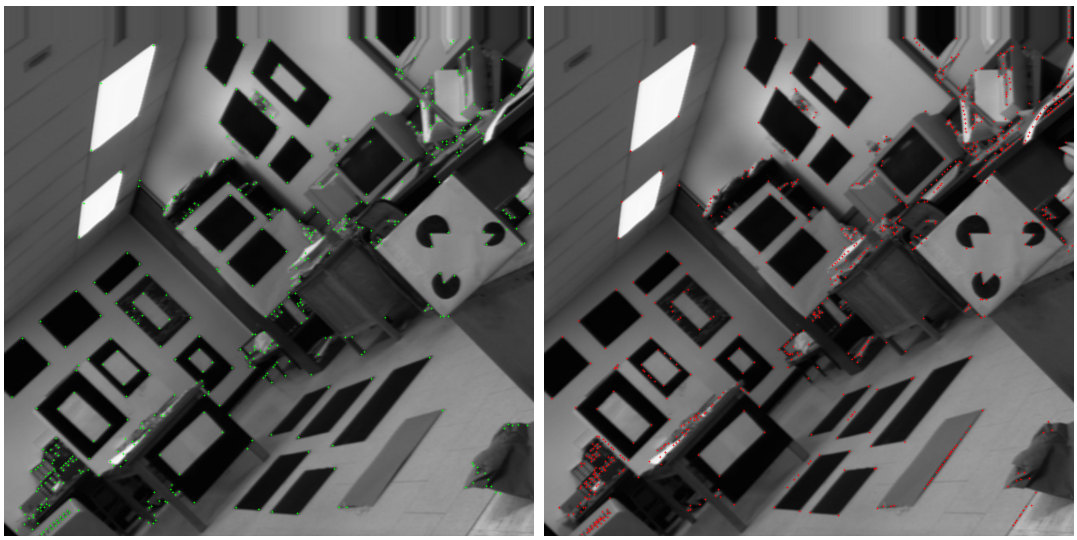


Figure 8.16: Corner detection : Filtering result of figure 8.15b) by anisotropic curvature (threshold = 0.1).



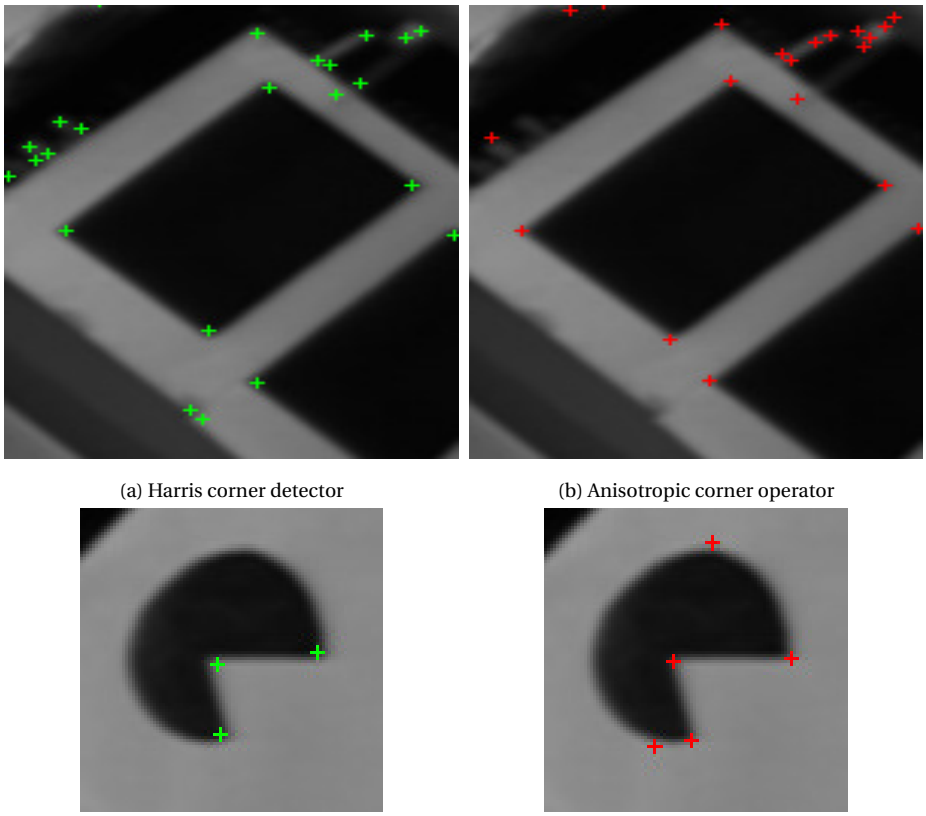
(a) Harris corner detector (b) Anisotropic corner operator

Figure 8.17: Corner detection. a) Harris corner detector (threshold = 0.001), b) Anisotropic corner operator $\times |I_{\xi_{1r}\xi_{2r}}|$ curvature local maxima threshold = 0.02.



(a) Harris corner detector (b) Anisotropic corner operator

Figure 8.18: Corner detection precision $\frac{1}{2}$ pixel. a) Harris corner detector (threshold = 0.001), b) Anisotropic corner operator $\times$ $|I_{\xi_{1r}\xi_{2r}}|$ curvature (threshold = 0.02).



(a) Harris corner detector (b) Anisotropic corner operator

(c) Harris corner detector (d) Anisotropic corner operator

Figure 8.19: Corner detection at $\frac{1}{2}$ pixel precision details. a) Harris corner detector (threshold = 0.001), b) Anisotropic corner operator $\times$ $|I_{\xi_{1r}\xi_{2r}}|$ curvature (threshold = 0.02).

8.4 Summary

Two new anisotropic corner operators were proposed in this chapter. The first one is defined using an anisotropic curvature operator $|I_{\xi_{1r}\xi_{2r}}|$. The second one is simply obtained by the difference of the application of two banks of filters : a classical anisotropic filter bank and a causal anisotropic filter bank. Both methods provide very interesting results, they both are compared with the reference Harris corner detector, and show better results without ambiguity. In any way, these two operators have specificities and provide results that may vary with the image local structure. The first one performs better on small structures or at sub-pixel precision, as the second one is more robust against noise. In order to obtain less sensitive results we use a simple data fusion scheme consisting in the multiplication of both operators followed by non-local maxima suppression allowing to benefit from both operators advantages.

Chapter 9

Conclusion and Perspective

Contents

9.1 Conclusion

The performance of computer vision and image processing techniques can have an important impact on the system equipped with the computer vision based automatic tasks, and also on the human image data analyst/observer that analyze and decide on the image information. These performances are particularly critical for real-time applications, medical image analysis, satellite/aerial image processing, and some robotic tasks.

In order to validate the performance of these mentioned techniques, they are usually assessed via both ways of objective and subjective (visual) evaluation metrics. The aim of the evaluations is to understand and highlight the intrinsic characteristics and behavior of methods, their limitation, compare/improve parameters, and finally compare these performances with different techniques. The result of the evaluation and validation of the image processing algorithms primarily helps in giving insight for further improving its performance.

In the image processing and computer vision state of the arts, both approaches of handcrafted and classic machine learning and deep learning methods are widely used. Both approaches have got their usefulness, effectiveness, and also limitation in certain cases and applications. Image processing using filtering techniques and classic supervised machine learning is still widely used and is vital for many applications involving from medical image processing to aerial image processing and autonomous action in computer vision systems and robotics.

Image filtering methods represent a corner-stone in image analysis, they are the backbone of all basic early vision tasks. Also, image filtering techniques are mainly used for segmentation tasks, low-level analysis of image information, pre-processing requirements, and certain optimization needs.

In this thesis, mainly five contributions in the domain of filtering-based low-level computer vision were proposed. These five contributions involve two extensive and objective evaluation (edge, ridge/valley detectors, and keypoint detectors), two multiscale line detection techniques, and one new anisotropic corner detection technique. First, an

extensive review and objective evaluation of state-of-the-art early vision filtering for segmentation (i.e., contour, edge, ridge/valley "line"), and scale space, subpixel, isotropic and anisotropic analysis in image processing were accomplished in chapter 2, 3, 4 For the first part of the objective evaluation and comparison of ridge/valley detection based on filtering techniques including, the driving filter parameters and characterizations were analyzed.

The extensive evaluation and comparison of ridges/valleys detectors were carried out both theoretically and experimentally on synthetic and real images. Each filtering technique was examined on complex images, where different types of noises were applied (namely Gaussian, Speckle and Poisson noises). The obtained comparative evaluation graphs revealed which method is reliable as a function of the width feature and the noise type.

Regarding non-oriented filters, the Z filter performs well when the ridge or the valley is very thin (width of one pixel) and requires less computational complexity computed by the Hessian matrix $\mathscr{H}$. On the other hand, $\mathscr{H}$ associated with the Gaussian G_σ and the highest eigenvalue (D_1) is a good compromise when the feature widths are growing. Yet, the Weingarten **W** and its eigenvalue give suitable and better continuous detected ridges. Steerable filters of order 2 (SF_2) and of order 4 (SF_4) obtain similar results, they are particularly reliable for images corrupted by noise(s), especially for bent features (SF_4 is a little more reliable), contrary to the SOAGK which is well adapted for straight features. This evaluation work can suggest which ridge/valley optimal parameter configuration and adjustment are optimal for its interested applied researchers and application tools and domain such as satellite or aerial image analysis (road, river, etc.), medical image analysis (blood vessels, filaments, nerve system, etc.), lines detection, image segmentation, and object detection.

Secondly, beginning with the overview of scale-space analysis in image processing theory and axioms, two multiscale line detection techniques based on Semi-Gaussian filters (namely SDSG and 2DSBG) were proposed and evaluated, as presented in chapters 5 and 6. Both multiscale techniques' objective and subjective evaluation presented an optimal performance for extracting complex scenes with bent lines and especially for the second method, close adjacent lines.

In multiscale line detection (chapter 5), the proposed SDSG (Second Derivative Semi-Gaussian) can be adapted to noisy environments and is also reliable to detect line features with heterogeneous types, widths, and prominence. An optimal scale selection function for multiscale processing is the main contribution of this approach. This technique has been compared to different types of multiscale filtering methods, including isotropic (as using the Hessian matrix) and oriented filters (isotropic or anisotropic). Quantitative and qualitative experiments regarding real images of different types and scales have shown the optimal efficiency and very promising results of the SDSG technique compared with the three major techniques of the state of the art.

Additionally, a second multiscale filter for line feature extraction, named 2DSBG (2nd Derivative Semi and Bi-Gaussian) were proposed, which is constructed from bi-Gaussian and second-order semi-Gaussian filters. This combined filter exploits the advantages of the bi-Gaussian for the detection of adjacent linear features, as well as the qualities presented by semi-Gaussian kernels for the analysis of bent and complex linear structures. The special usefulness of bi-Gaussian is presented for the detection of adjacent linear features, and the precision of the Semi-Gaussian kernel for curved linear structures. Finally, the presented results on synthetic and real images allowed to find the optimal parameter configuration ($\rho \in [0.5, 0.7]$), and thus confirming the novelty and merit of the 2DSBG over

the existing filtering methods.

Thirdly, an objective evaluation of filtering-based corner detection techniques was first studied (chapter 7), and following that a new anisotropic corner detection algorithm was proposed in chapter 8. For the objective evaluation of filtering-based corner detection techniques, we conducted an extensive assessment of the 12 state-of-the-art corner detection techniques (Shokouh et al. [2022]) with the application to feature matching in the context of a complex underwater video scene named AQUALOQ dataset. The AQUALOQ database (Ferrera et al. [2019]) contains all types of image transformation and various natural noises. The repeatability rate of each detection operator has been both statistically and visually demonstrated in presenting a guideline of which detectors are robust depending on video frame complexity.

This work can contribute as a directive to the practitioners of this domain for choosing the appropriate keypoint detectors relating to the specific application (i.e., Ro, HS, and F exemplify the robust salient point detectors with the highest stability). The scale parameter ρ of the studied keypoint detectors have been studied for an objective and complete evaluation in this specific context. The test result indicated that KLT and Harris-Stephens (HS), two particularly popular detectors, perform well but not the best among the 12 tested, especially when the ρ parameter is not well selected. This evaluation emphasizes on the filtering technique which is fast and straightforward than other approaches along with the keypoint matching methods (ZNCC) which is few time-consuming and can be easily implemented. Accordingly, the filtering techniques are useful for certain cases of image processing and optimization which are either used independently or can be used alongside with deep learning models either in pre-processing or post-processing stages.

Finally, our newly proposed anisotropic corner operator which is based on banks of causal anisotropic and classic anisotropic filters presented interesting result specially for better localization and accuracy. The proposed anisotropic method is compared with Kitchen-Rosenfeld and Harris corner detection methods. The visual results show the robustness of the technique.

9.2 Perspective and Discussion

We know that there are still many challenges to be addressed in the image processing and computer vision research and development, particularly in the image low-level feature detection and description (concerning our scope of research). Meanwhile, it is also known that the process of development and optimization in the image processing and computer vision system pipeline is usually contextual and application dependent. Usually, it is due to both complexity of the image scene and the ever-increasing demand for better accuracy, speed, precision, and automation. Therefore, each development, optimization, and contribution can be considered significant for certain needed tasks and applications. At the same time, one can always argue the contribution flaws and limitations.

Concerning the handcrafted and deep learning approaches, it is obvious that the recent advances and breakthrough in the deep learning-based computer vision approaches, the handcrafted and classic machine learning techniques in this domain seems to be (unfortunately) overshadowed by deep learning benchmarks and models. However, we argue that, in spite of the remarkable result of deep learning models, particularly for their automatic learning of images' global features, high-level scene understanding, and generalizability of models; these models have limitations for image low-level processing such as edge, ridge/valley and corner extraction in case of localization and accuracy.

One of the major challenges of deep learning techniques is to develop a large and

accurately/precisely labeled dataset, which is not easily achievable for image low-level features. Secondly, the developed deep learning techniques are quite over-complicated models with arguably comparative results for some basic low-level vision tasks. Therefore, we argue that the room for handcrafted and classic machine learning techniques for computer vision tasks will persist to possess its demands and importance. The handcrafted and classic techniques are particularly needed for image pre-processing or/and post-processing optimization also.

Besides, one can argue that image pre-processing can distort or changes the true nature of the raw data, but we also insist that the knowledgeable use of image pre-processing can ultimately lead to comparatively optimal local and global feature detection, description, and final output of the pipeline. Image pre-processing, as ours concerning filtering techniques, for image low-level feature enhancements can favorably or unfavorably affect feature analysis downstream in the computer vision pipeline, depending on how the pre-processing technique is employed and as also on which application.

Furthermore, concerning the perspectives for the low-level image processing of certain tasks such as edge detection, line feature detection, and corner/junction detection when accurate precise detection and localization are the main demand (e.g., medical image analysis where the enhanced blood vessel detection is a critical, accurately localized line and corner detection for image calibration and 3D reconstruction, etc.), the classic hand-crafted technique is still widely used as referenced in the general introduction chapter. The handcrafted technique is more suited for certain and specific enhancement and optimization tasks, such as multiscale analysis and enhancement for precise detection of complex and noisy contour structures (e.g., narrow and bent ridges, narrow and adjacent ridges/valleys). Additionally, the developed handcrafted techniques can work on all types of images without any pre-learning.

Apart from what is presented in this report, we would like to mention that we also carried out a preliminary experimental benchmarking of deep learning techniques for both low-level image contour detection and high-level object detection application (Pruvost et al. [2022]). Regarding edge detection using deep learning, we executed the dense extreme inception network for edge detection developed by Soria et al. [2020]. The visual final segmentation result is interesting, as of blending of the different types of low-level features (edge, contour, ridge/valley, corner).

Nevertheless, what we perceive from the experiments is that machine learning and deep learning models are better for training the machine to have rather a more semantic understanding of the generic low-level features (contour, edge, boundary, ridge/valley, corner) useful for segmentation, object detection, etc. Whereas, the handcrafted techniques are simple, straightforward, and focuses directly on local intensity changes -specific types of low-level features such as step/ramp edge, contour, ridge/valley, boundary, corner, etc., without trying to globally understand the scene.

Along with, to get rather a more practical context of how deep learning models perform, we developed an invertebrate dataset and applied one of the latest object detection model, such as YOLOv5 (Jocher [2020]), which is mentioned in the list of publications in the introduction chapter 1. Although it was a novel application of the underground soil fauna detection via the deep learning techniques, however, we observed the limitations and complexity of deep learning approaches so far. The demonstrated Fig. 9.1 shows the complexity of problems and deep learning based solution, in spite of using well developped ground truth dataset and latest deep learning model with the transfer learning. Our aim was to have a benchmarking of the handcrafted vs deep learning techniques for low-level feature analysis.

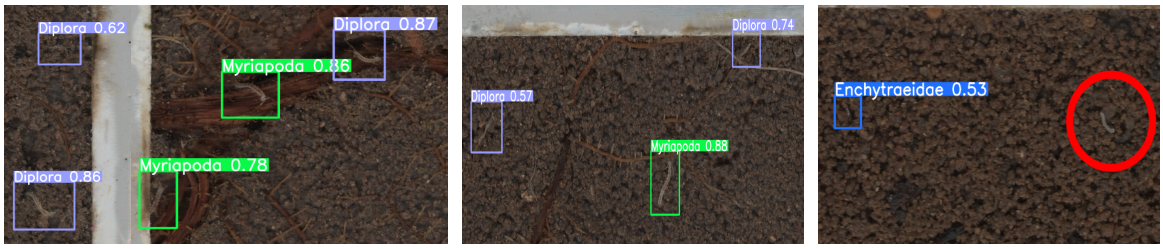


Figure 9.1: Examples of invertebrates' detection in underground scanned images: from left to right, true positives (myriapods and diplura –diplora–), false positives (2 roots detected as diplura), and false negative (undetected myriapod, circled in red), see details in Pruvost et al. [2022]. Images of size 449×760 , 434×908 and 356×573 respectively.

But, we found it overcomplicated to develop a benchmarking metrics rather not worthy, interesting and useful - deep learning highly depends first on the accurately developed ground truth dataset -, and secondly on the architecture and model, whilst handcrafted techniques mainly depend directly on the handcrafted algorithm. Therefore, based on our scope of observation, research and experiments, we concluded that both deep learning and handcrafted techniques have their pros and cons on certain task, application and context. The argument of which approaches and techniques performs better and optimal will depend on the application and context "what is meant better and/or optimal", and there we have no absolute and generic definition of optimality. Depending on the application and the task, we need to choose between handcrafted and deep learning techniques or the composition of both.

Chapitre 10

Résumé en Francais

Contents

10.1 Introduction

Le traitement d'images et la vision par ordinateur englobent un large domaine d'applications, telles que l'analyse d'images médicales, d'images aériennes pour la navigation autonome, la reconnaissance d'objets, la vision robotique, etc. Dans des domaines d'applications de la vision par ordinateur, tels que la reconnaissance d'objets, la mise en correspondance ou la perception 3D, l'extraction de primitives constitue généralement une étape clé. Par exemple dans le cadre d'une mise en correspondance les primitives extraites peuvent être utilisées pour calculer des descripteurs locaux qui permettront l'appariement de l'image d'un objet dans la scène, ou encore de deux images droite et gauche en stéréovision, etc.

D'autres types d'applications comme par exemple l'indexation d'images peuvent se concentrer sur des descripteurs plus globaux sans toutefois nécessiter d'étapes de segmentation ou d'extraction de primitives. Cependant d'autres applications peuvent utiliser à la fois sur les caractéristiques locales et globales de l'image/de l'objet, ce qui implique à la fois des descripteurs glogaux calculés sur l'image entière et l'utilisation de primitives bas niveau telles que des contours, des lignes crête/vallée, des coins, etc.

Récemment de nouvelles approches faisant intervenir des méthodes d'apprentissage automatique on fait leur apparition notamment pour la reconnaissance d'objets, la reconnaissance faciale, la détection de personnes, l'analyse de la gestuelle humaine, ou encore simplement pour la restauration d'images bruitées ou dégradées. On pourra citer les méthodes basées sur de l'apprentissage en profondeur ou deep-learning qui ont démontré des capacités extrèmement impressionnantes dans des applications complexes ou le paradigme segmentation/mise en correspondance devient difficile définir et à mettre en œuvre. Ces méthodes d'apprentissage en profondeur apprennent automatiquement des descriptions en construisant généralement une représentation hiérarchique des données

image pour aboutir à classification souhaitée (méthodes supervisées). Ces descriptions sont géneralement contitués de résultats de filtrage (filtres appris dans des réseaux convolutionnels), d'informations colorimétriques, de texture ou de formes.

Cependant ces méthodes restent pour l'instant cantonnées à des applications globales et leur utilisation pour la segmentation d'images ou l'extraction de primitives reste marginale. Il n'existe pas réellement à l'heure actuelle de base de données d'apprentissage par exemple pour la détection de contours, ou de coins. En détection de contours par exemple l'amalgame entre contour et ligne de crête/vallée est souvent fait, de même pour les coins-jonctions. Les travaux que l'on trouve dans les bibliographies sont souvent basées sur des architectures de réseau ad-ok difficilement reproductibles. Enfin la constitution d'une base de données de segmentation pertinente nécessiterait un coût humain extrèmement important de labellisation afin de mettre en œuvre un apprentissage supervisé.

En revanche les approches plus classiques par filtrage dérivatif sont basées sur une modélisation mathématique très précise de la surface image et des primitives recherchées, un point de contour est de nature différente d'un point de crête (ce n'est pas le même ordre de dérivation). Ces méthodes restent donc les plus pertinentes pour les applications qui nécessitent des primitives images ainsi que leur descriptions.

Compte tenu de ces considérations nous nous sommes intéressés à la détection de lignes et de coins en utilisant des approches par filtrage dérivatif. Nous avons abordé trois points importants :

1. Evaluation de méthodes de segmentation :

   Il existe de nombreuses méthodes de filtrage d'images proposées et développées pour la détection de crêtes/vallées, et chacune revendique et présente d'une certaine manière les performances de l'algorithme proposé. Les performances de ces méthodes ne sont généralement pas objectivement évaluées et comparées pour permettre aux utilisateurs de choisir la meilleure technique pour leurs applications prévues. Par conséquent, il y a toujours un besoin d'évaluation à la fois objective et subjective (mais le plus souvent l'évaluation est simplement subjective) des techniques de filtrage d'images.

   Nous avons consacré une partie de ce travail de thèse à cette évaluation objective des techniques de filtrage d'image pour les caractéristiques de bas niveau (bord, crête/vallée, ligne, coin). Tout d'abord, nous avons travaillé sur l'analyse objective des détecteurs de lignes caractérisées comme des structures d'image minces et complexes. Notre analyse nous permet de choisir quelles valeurs de paramètres correspondent à la configuration appropriée pour obtenir des résultats précis et des performances optimales. Une analyse approfondie suivie d'une comparaison supervisée et objective des différentes techniques de détection de crêtes basées sur le filtrage est détaillée dans notre travail. La configuration optimale des paramètres de chaque technique de filtrage destinée à l'outil d'analyse des caractéristiques saillantes de l'image a été objectivement étudiée, où les paramètres de chaque filtre choisi correspondent à la largeur de la crête ou de la vallée souhaitée.

   Dans un deuxième temps, nous avons étudié les performance des détecteurs de coins basés sur des techniques de filtrage. Les coins, en tant que caractéristique stable possédant les caractéristiques définies d'un point d'intérêt robuste, restent un domaine de recherche actif pour les chercheurs en vision artificielle en raison de leurs applications dans la capture de mouvement, l'appariement d'images, le suivi, l'enregistrement d'images, la reconstruction 3D, la reconnaissance d'objets,

etc. Il existe différentes techniques de détection de points d'intérêt ; nous nous sommes concentrés seulement la détection de coins dans des images en niveaux de gris. Nous avons donc effectué une comparaison objective de douze méthodes différentes de détection de points d'intérêt courament utilisées ; une application de feature matching a été exécutée dans le cadre de séquences vidéo sous-marines. Ces vidéos contiennent du bruit et de nombreuses transformations géométriques et/ou photométriques

2. Une nouvelle méthode de détection de lignes multi-échelle :

   Dans la plupart des applications réelles, les images apparaissent sous de nombreuses transformations géométriques et photométriques (transformations euclidiennes, affine ou projectives, changement d'illumination, etc.). Les performances des détecteurs de primitives bas niveau d'image basés sur un modèle idéal vont donc se dégrader en fonction de la proportion de transformation considérée. Parmi toutes les transformations géométriques possibles, le changement d'échelle reste cependant la transformation la plus importante à prendre en compte. Nous savons que les objets sont généralement composés d'une part de différentes structures de taille différentes et d'autre part, la taille de ces structures va varier en fonction de l'échelle d'observation.

   Ces structures vont donc apparaître à des échelles différentes dont il faudra tenir compte au niveau des filtres utilisés et des seuils de détection des algorithmes. Pour cela, la création de représentations espace-échelles de l'image est nécessaire pour analyser des scènes a priori inconnues. Dans la théorie des espaces échelle, une famille d'images est générée contenant les versions lissées de l'image d'origine avec un paramètre de lissage. Dans l'espace échelle gaussien (euclidien, linéaire) la représentation de l'espace échelle est paramétrée par la taille du noyau de lissage gaussien (écart-type). Plus l'échelle d'observation augmente, plus les structures à échelle fine de l'image sont supprimées Historiquement, de nombreuses approches ont été introduites pour la représentation multi-échelle des images, y compris les méthodes pyramidales, ondelettes, quad-tree et multi-grille.

   Nous avons consacré une partie de ce travail de thèse au développement de deux nouveaux détecteurs de lignes multi-échelles. Tout d'abord, nous avons proposé un nouveau filtre capable d'extraire des structures multi-échelles complexes et linéaires adjacentes. Le filtre proposé est composé d'un filtre dérivé bi-gaussien et semi-gaussien et est adapté au cas multi-échelle.

   Nous avons réalisé des expériences sur différentes images contenant des structures linéaires adjacentes complexes à différentes échelles. La structure linéaire extraite sur les images synthétiques et réelles s'est révélée plus efficace que les techniques classiques d'extraction de structure linéaire. Deuxièmement, nous avons proposé un autre nouveau filtre multi-échelle composé d'un noyau bi-gaussien et d'un noyau semi-gaussien, capable de mettre en évidence des structures linéaires complexes telles que des crêtes et des vallées de différentes largeurs, avec une robustesse au bruit. Des expériences ont été réalisées sur un ensemble d'images synthétiques et réelles contenant des éléments linéaires adjacents.

3. Détection de points clés/points d'intérêt et/ou de coins :

   Nous nous sommes intéressés à la détection de coins avec un souci de précision de la localisation des points détectés. Nous avons développé un nouveau détecteur de

coin anisotrope, présentant une meilleure précision que les méthodes modernes les plus employées et qui ont été évaluées dans la partie évaluation des détecteurs de ce document. Nous introduisons tout d'abord une mesure de courbure anisotrope capable de fournir une détection précise du cois en calculant simplement un maximum local de type "courbure anisotrope". Ce nouvel nouvel opérateur est une extension/variante de l'opérateur de Kitchen-Roselfeld calculé de manière anisotrope et assymétrique. Nous proposons dans un second temps, un detecteur de coins basé là encore sur des techniques de filtrage anisotrope. Ces deux opérateurs complémentaires définissent un détecteur de coins permettant une détection robuste et extrèmement précise indépendante dans une grande mesure des paramètres des filtres utilisés. Les coins détectés sont peu dépendant de l'échelle des filtres et les coins peuvent aussi être détectés au niveau sub-pixelique.

10.2 Approches classiques ou par apprentissage en profondeur

Dans le domaine du traitement d'images et de la vision par ordinateur, depuis la dernière décennie, nous avons assisté à un nombre de travaux conséquent utilisant des techniques d'apprentissage en profondeur. Pour certaines applications ces techniques ont surpassé les techniques classiques en général. Les techniques d'apprentissage en profondeur font actuellement partie des méthodes de référence pour des tâches de vision de haut niveau comme la classification d'images, la détection d'objets, la reconnaissance faciale, l'estimation de pose, etc. Ces technique nécéssitent de disposer d'un jeu de données d'apprentissage suffisant et de bonne qualité pouvant aller jusqu'à plusieurs millions d'images annotées ce qui représente suvent un coût de travail humain considérable.

En revanche, pour des tâches de segmentation d'image de bas niveau tel que la détection de contours, la détection de lignes de crêtes/vallées, ou encore la détection de coins et jonctions, les méthodes classiques surpassent les méthodes par apprentissage. On pourra citer des applications comme l'analyse d'images médicales, la détection de vaisseaux sanguins ou encore les applications qui nécessitent une localisation précise et robuste de points d'intérêt par exemple pour la calibration de caméras, la reconstruction 3D, etc.

Souvent pour la segmentation bas niveau un modèle mathématique précis de la primitive recherchée peut être établi. Ainsi, la plupart du temps des résultats mathématiques permettent d'aboutir à des solutions optimales avec une mise en œuvre de faible complexité algorithmique. Ce qui évidemment est préférable à une solution par apprentissage dans laquelle il aurait fallu constituer une base d'apprentissage forcément très importante Junfeng et al. [2022]. De même lorsque l'aspect multi-échelle doit être pris en compte, les techniques classiques sont plus appropriées.

Cependant, même s'il existe des techniques de réseau de neurones profonds utilisées pour ce traitement d'image de bas niveau, elles sont généralement mises en œuvre avec une combinaison de techniques classiques soit en phase de pré-traitement, soit en phase de post-traitement afin d'améliorer les résultats finaux.

10.3 Principaux apports

Les principales contributions de ce travail comprennent l'évaluation objective des techniques de détection de lignes de crête/vallée et de points d'intérêt (coin) basées sur le filtrage, et l'analyse multi-échelle. Une nouvelle technique de détection de lignes multi-échelles et la proposition d'une nouvelle méthode de détection de coin anisotrope et assymétrique.

- Évaluation objective des techniques de détection de crêtes/vallées.
- Proposition d'une technique de détection d'entités linéaires multi-échelles à l'aide d'un filtre semi-gaussien de second ordre.
- Proposition d'une technique de détection de caractéristiques linéaires multi-échelles utilisant la combinaison d'un filtre bi-gaussien et semi-gaussien.
- Évaluation de la répétabilité des techniques de détection de coin basées sur le filtrage dans le suivi des séquences d'images sous-marines.
- Proposition d'une nouvelle technique de détection de coin anisotrope à l'aide d'une combinaison de filtres anisotropes causaux et non causaux, et de filtrage anisotrope assymétrique.

10.4 Conclusion

Les performances des techniques de traitement d'image peuvent avoir un impact important sur un système de vision par ordinateur, ainsi que sur la fiabilité des informations fournies. Les performances des tâches automatiques basées sur la vision par ordinateur sont particulièrement critiques pour les applications en temps réel, l'analyse d'images médicales, le traitement d'images satellitaires/aériennes ou la vision robotique.

Afin de valider les performances des techniques mentionnées, celles-ci sont généralement évaluées via des mesures d'évaluation objectives et subjectives (visuelles). Ces évaluations ont pour but de comprendre et de mettre en évidence les caractéristiques et comportements intrinsèques des méthodes, leurs limites, et enfin de pouvoir comparer les performances des différentes techniques. Les résultats d'évaluation et de validation des algorithmes aident principalement à donner un aperçu de leurs capacité et par la la suite éventuellement améliorer encore leurs performances.

Les méthodes de filtrage d'image forment l'épine dorsale de la plupart des tâches de base en traitement bas niveau. Les techniques de filtrage d'images sont principalement utilisées pour les tâches de segmentation, l'analyse de bas niveau des informations d'image, les exigences de prétraitement et certains besoins temps réel par exemple en vision robotique, réalité augmentée, etc. Ce travail de thèse s'est donc principalement orienté filtrage et segmentation en explorant les propriétés du filtrage anisotrope, du filtrage anisotrope causal, et du filtrage assymétrique.

Cette thèse, présente cinq contributions dans le domaine de la vision par ordinateur bas niveau basées sur le filtrage. Ces cinq contributions impliquent deux évaluations approfondies et objectives (détecteurs de bord, de lignes de crêtes/vallées et détecteurs de coins), deux techniques de détection de lignes multi-échelles et une nouvelle technique de détection de coin anisotrope. Tout d'abord, un état de l'art du filtrage en vision par ordinateur pour la segmentation d'images par des méthodes différentielles (contour, bord,

crête/vallée) est réalisé, les espaces échelles et les techniques de filtrage isotrope et anisotrope sont abordées. Un premier thème concerne la détection de lignes : évaluation de détecteur existants puis proposition de deux nouveaux detecteurs plus preformants.

Ce thème présente tout d'abord une évaluation et une comparaison objective de plusieurs détecteurs de crêtes/vallées basée sur des techniques de filtrage. Ces méthodes sont aussi évaluées en fonction des paramètres de filtres utilisés. Chaque technique de filtrage a été examinée sur des images complexes, où différents types de bruits ont été appliqués. Suite à cette évaluation des détecteurs de lignes, deux nouvelles techniques de détection de lignes multi-échelles basées sur des filtres gaussiens anisotropes sont proposées et évaluées. L'aspect multi-échelle est abordé afin d'obtenir des performances optimales pour extraire des des lignes courbes et des lignes adjacentes proches dans des scènes complexes. La première technique proposée est adaptée au cas ou les données images sont bruitées, elle est aussi définie dans cadre multi-échelle permettant de détecter des lignes d'épaisseurs hétérogènes. la principale contribution de cette approche concerne la fonction de sélection d'échelle automatique optimale.

La deuxième technique proposée concerne l'extraction de structures linéaires adjacentes. Cette approche utilise des filtres bi-Gaussiens et semi-gaussiens de second ordre. Nous utilisons le filtre bi-Gaussien pour la détection d'éléments linéaires adjacents et la précision du noyau semi-gaussien pour les lignes courbes et complexes.

Le deuxième thème corcerne cette fois la détection de coins. Comme pour le thème précédent une évaluation objective de détecteurs existants est réalisée et un nouveau détecteur de coins est proposé.

Tout d'abord, nous effectuons une évaluation approfondie des douze détecteurs de coins en vue d'une application de mise en correspondance dans un contexte de scènes sous-marines (vidéos complexes de la base de données AQUALOQ). Cette base de données contient de nombreux les types de transformations géométriques et divers bruits naturels. Le taux de répétabilité des opérateurs de détection est montré statistiquement en fonction de la complexité de la vidéo. Ce travail contribue en tant que directive au choix du détecteurs de coins appropriés à l'application envisagée.

Cette évaluation met l'accent sur la technique de filtrage qui doit être rapide et simple par rapport aux autres approches, ainsi que sur les méthodes d'appariement des coins qui sont peu coûteuses et qui peuvent être facilement mises en œuvre.

Suite à cette évaluation, nous nous sommes intéressés à la précision en détection de coins. Le but de ce travail est de fournir une alternative aux filtres isotropes comme par exemple le filtre gaussien qui délocalise les coins obtenus en fonction du paramètre le lissage (écart-type). Nous recherchons une méthodes ne délocalisant pas les coins obtenus tout en conservant un lissage important pour la robustesse au bruit. Les filtres gaussiens anisotropes causaux sont de bons candidats dans la mesure ou cette fois l'écart-type varie selon les axes du filtre.

Nous proposons deux nouveaux opérateurs : le premier basé sur la définition d'une courbure anisotrope assymétrique, et deuxième opérateur basé sur la différence de deux filtrages anisotropes causal et non-causal. Les résultats sont comparés avec l'opérateur de Harris et ont montré une bien meilleure localisation des coins obtenus et une plus grande immunité au bruit sur des images réelles et synthétique. Les résultats on aussi été validés en précision sub-pixelique $\frac{1}{2}$ et $\frac{1}{4}$ de pixel.

10.5 Perspectives

Le traitement d'images est un vaste domaine qui ne peut pas avoir de résultat et de conclusion solides en travaillant sur des domaines spécifiques. La conclusion de la recherche doit être valable pour la portée définie et limitée des études. Ce que j'ai appris de mes efforts de recherche au cours de ma thèse, c'est qu'il est impossible de réaliser ou de développer des méthodes et des applications de traitement d'image sans limitations et besoins d'optimisation continus. Ainsi, il y aura toujours de la place pour la recherche et l'optimisation en vision par ordinateur.

Cela est dû à la fois à la complexité des scènes et à la demande toujours croissante pour une meilleure précision, vitesse et automatisation. Par conséquent, chaque développement, optimisation et contribution peut être considéré comme significatif pour certaines tâches et applications.

Pendant ce temps, après les avancées récentes et les percées dans les approches de vision par ordinateur basées sur l'apprentissage profond, on pourrait penser que les technique classiques dans ce domaine vont être éclipsées par les modèles d'apprentissage profond.

Cependant, nous continuons à soutenir que, malgré le résultat remarquable des modèles d'apprentissage en profondeur, en particulier pour leur apprentissage automatique des caractéristiques globales des images, la compréhension de la scène de haut niveau et la généralisabilité des modèles, ces modèles présentent des limites importantes et des inconvénients dans tous les cas où un traitement de bas niveau tel que l'extraction des bords, des crêtes/vallées et des coins avec des contraintes localisation et de précision est primordial.

L'un des principaux défis des techniques d'apprentissage en profondeur est d'avoir un ensemble de données étiqueté avec précision, ce qui n'est pas facilement réalisable pour les primitives de bas niveau de l'image (ou pas fait à l'heure actuelle). Enfin pour ces techniques : ce qui n'a pas été appris ne sera sera pas reconnu et il est évidemment difficile voire impossible de faire apprendre à une machine toutes les configuration ou contextes possibles dans un cadre de vision généraliste.

En outre, on peut affirmer que le prétraitement des images peut déformer ou modifier la véritable nature des données brutes, mais nous soutenons également que l'utilisation éclairée du prétraitement des images peut finalement conduire à une meilleure détection, description et sortie finale de primitives ou caractéristiques locales et globales. Le prétraitement d'image, comme le nôtre concernant les techniques de filtrage de bas niveau de l'image peut affecter favorablement la qualité des résultats obtenus par une chaîne de traitement selon la façon dont ces techniques sont utilisées, agencées et aussi sur quelle application.

Enfin, concernant notre point de vue pour le traitement d'image de bas niveau de certaines tâches telles que la détection des contours, la détection des lignes, la détection des crêtes/vallées, la détection des lignes de crête et la détection des coins et des jonctions lorsqu'une détection et une localisation précises sont exigées (par exemple, analyse d'image où la détection améliorée des vaisseaux sanguins est une détection de ligne et de coin critique et localisée avec précision pour l'étalonnage de l'image et la reconstruction 3D, etc.), les techniques classiques sont encore largement utilisées, comme indiqué dans le chapitre d'introduction générale. Les techniques classiques sont plus adaptées à certaines tâches spécifiques, comme par exemple, la détection de lignes étroites, courbées, adjacentes, ou de coins.

Bibliographie

Abdulrahman, H., Magnier, B., and Montesinos, P. (2017). From contours to ground truth : How to evaluate edge detectors by filtering. *J. of WSCG*, 25(2) :133–142. 36, 95

Achard, C., Bigorgne, E., and Devars, J. (2000). A sub-pixel and multispectral corner detector. In *ICPR*, volume 3, pages 959–962. 99, 100, 101

Akiyama, H., Tanaka, M., and Okutomi, M. (2015). Pseudo four-channel image denoising for noisy CFA raw data. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4778–4782. 73

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2010). Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5) :898–916. 91

Armande, N., Montesinos, P., Monga, O., and Vaysseix, G. (1999). Thin nets extraction using a multi-scale approach. *CVIU*, 73(2) :248–257. 62, 64

Awrangjeb, M., Guojun, L., and Clive, F. (2012). Performance comparisons of contour-based corner detectors. *IEEE TIP*, 21(9) :4167–4179. 99

Bae, Y., Lee, W.-H., Choi, Y., Jeon, Y., and Ra, J. (2015). Automatic road extraction from remote sensing images based on a normalized second derivative map. *IEEE GRSL*, 12(9) :1858–1862. 81, 87

Beaudet, P. (1978). Rotationally invariant image operators. In *ICPR*, pages 579–583. v, 12, 99, 100, 101

Blom, J., Romeny, B. H., and Koenderink, J. (1992). Affine invariant corner detection. *Doctoral Dissertation*. 99, 100, 101

Buades., A., Coll., B., and Morel, J. M. (2004). On image denoising methods. In *TECHNICAL NOTE, CMLA (CENTRE DE MATHEMATIQUES ET DE LEURS APPLICATIONS*. 10

Canny, J. (1983). Finding edges and lines in images. *M.I.T. Artificial Intelligence Laboratory*, pages TR–720. 25, 26, 34

Canny, J. (1986). A computational approach to edge detection. *IEEE TPAMI*, pages 679–698. 56, 62, 63, 72, 80, 84, 87, 90, 103

Chicco, D. and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics 21*, 6. 37

Crowley, J. L. and Parker, A. C. (1984). A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE TPAMI, no. 2*, pages 156–170. 59

Deriche, R. (1987). Optimal edge detection using recursive filtering. In *First International Conference on Conputer Vision, London.* 27

Deriche, R. (1992). Recursively implementing the gaussian and its derivatives. In *ICIP*, pages 263–267. 30, 69

Do Carmo, M. P. (2016). *Differential geometry of curves and surfaces : revised and updated second edition.* Courier Dover Publications. 62

Eberly, D., Gardner, R., Morse, B., Pizer, S., and Scharlach, C. (1994). Ridges for image analysis. *JMIV*, 4(4) :353–373. 61

Ferrera, M., Creuze, V., Moras, J., and Trouv, P. P. (2019). Aqualoc : An underwater dataset for visual-inertial-pressure localization. In *The International Journal of Robotics Research.* viii, 99, 103, 104, 132

Forstner, W. and Gulch, E. (1987). A fast operator for detection and precise location of distinct points, corners and circular features. In *Fast Processing of Photogrammetric Data*, pages 281–305. 99, 101

Freeman, W. and Adelson, E. H. (1991). The design and use of steerable filters. *IEEE TPAMI*, 13(9) :891–906. 65, 82, 87

Getreuer, P. (2013). A survey of gaussian convolution algorithms. *Image Processing On Line*, (3) :286–310. 30, 69

Geusebroek, J.-M., Smeulders, A., and Van De Weijer, J. (2003). Fast anisotropic gauss filtering. *IEEE TIP*, 12(8) :938–943. 66, 70

Glasbey, C. A. (1993). An analysis of histogram-based thresholding algorithms. *CVGIP Graph. Model. Image Process.*, 55 :532–537. 13

Gouton, P., Laggoune, H., Kouassi, R., and Paindavoine, M. (2000). Ridge-line optimal detector. *Optical Eng.*, 39(6) :1602–1612. 63, 69

Haggui, O., Tadonki, C., Lacassagne, L., Sayadi, F., and Bouraoui, O. (2018). Harris corner detection on a numa manycore. *Future Generation Computer Systems*, 88 :442–452. 98, 99

Haralick, R. M. (1983). Ridges and valleys on digital images. *Comput. Vis. Graph. Image Process.*, 22(1) :28–38. 57, 60

Harris, C. G. and Stephens, M. J. (1988). A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151. 99, 101, 102

Iverson, L. and Zucker, S. (1995). Logical/linear operators for image curves. *IEEE TPAMI*, 17(10) :982–996. 65, 68

Jacob, M. and Unser, M. (2004). Design of steerable filters for feature detection using canny-like criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8) :1007–1019. 65, 66, 82

James, A. P. (2016). Edge detection for pattern recognition : a survey. *ArXiv*, abs/1602.04593. 23

Jocher, G. (2020). ultralytics/yolov5 : v3.1 - Bug Fixes and Performance Improvements. *Zenodo*. 133

Junfeng, J., Shenjuan, L., Gang, W., Weichuan, Z., and Changming, S. (2022). Recent advances on image edge detection : A comprehensive review. *Neurocomputing*. 4, 138

Kenney, C., Zuliani, M., and Manjunath, B. (2005). An axiomatic approach to corner detection. In *CVPR*, volume 1, pages 191–197. 99, 101, 102

Kirsch, R. (1971). Computer determination of the constituent structures of biomedical images. *Computer and Biomedical Research. USA*, pages 315–328. 25

Kishi, T., Hibara, T., and Nakano, N. (1993). Cpt inspection systems with image processing. In *Proceedings of IECON '93 - 19th Annual Conference of IEEE Industrial Electronics*, pages 1893–1897 vol.3. 10

Kitchen, L. and Rosenfeld, A. (1982). Gray-level corner detection. *Pattern Recognit. Letters*, 1 :95–102. 31, 99, 100, 101

Koenderink, J. (1984). The structure of images. *Biol. Cybern. Springer-Verlag. Netherlands*, 50 :363 370. 46

Komorkiewicz, M., Kluczewski, M., and Gorgon, M. (2012). Floating point hog implementation for real-time multiple object detection. In *22nd International Conference on Field Programmable Logic and Applications (FPL)*, pages 711–714. 11

Kramer, H. P. and Bruckner, J. B. (1975). Iterations of a non-linear transformation for enhancement of digital images. *Pattern Recognit.*, 7 :53–58. v, 11

Laligant, O., Truchetet, F., and Fauvet, E. (2013). Noise estimation from digital step-model signal. *IEEE TIP*, 22(12) :5158–5167. 74

Laligant, O., Truchetet, F., and Meriaudeau, F. (2007). Regularization preserving localization of close edges. *IEEE Sign. Proc. Lett.*, 14(3) :185–188. 70, 92

Leng, C., Zhang, H., Li, B., Cai, G., Pei, Z., and He, L. (2019). Local feature descriptor for image matching : A survey. *IEEE Access*, 7 :6424–6434. 11

Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*. Springer Link. vi, 45, 46

Lindeberg, T. (1998). Edge detection and ridge detection with automatic scale selection. *Int. J. of Comput. Vis.*, 30(2) :117–156. 45, 46, 61, 81, 87, 91, 92

Lipschutz, M. (1969). Differential geometry. *McGfaw-Hill*. 100

Lopez-Molina, C., De Ulzurrun, G., Baetens, J., Van den Bulcke, J., and De Baets, B. (2015). Unsupervised ridge detection using second order anisotropic gaussian kernels. *Sign. Proc.*, 116 :55–67. viii, 65, 67, 75, 82, 86, 87, 91, 92

Magnier, B. (2011). *Détection de Contours et Diffusion Anisotropique dans les Images*. PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc. v, 15, 16, 29

Magnier, B. (2019). Edge detection evaluation : A new normalized figure of merit. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2407–2411. 37, 71, 72, 86, 87, 95

Magnier, B., Abdulrahman, H., and Montesinos, P. (2018). A review of supervised edge detection evaluation methods and an objective comparison of filtering gradient computations using hysteresis thresholds. *Journal of Imaging*, 4(6) :74. 36, 71, 72, 87, 95

Magnier, B., Aberkane, A., Borianne, P., Montesinos, P., and Jourdan, C. (2014). Multi-scale crest line extraction based on half gaussian kernels. In *IEEE ICASSP*, pages 5105–5109. 84, 93

Magnier, B., Diep, D., and Montesinos, P. (2011). Perceptual curve extraction. In *IEEE IVMSP Workshop : Perception and Visual Signal Analysis*, pages 93–98. IEEE. 93

Magnier, B., Shokouh, G.-S., Xu, B., and Montesinos, P. (2021). A multi-scale line feature detection using second order semi-gaussian filters. In *CAIP*, volume 13053 of *LNCS*, pages 98–108. Springer International Publishing. 6, 92, 93

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proc. of the R. Soc. of Lond.*, 207(1167) :187–217. 32, 58, 59

Matthews, B. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2) :442–451. 37

Mokhtarian, F. and Mohanna, F. (2006). Performance evaluation of corner detectors using consistency and accuracy measures. *CVIU*, 102(1) :81–94. 99

Montesinos, P. and Datteny, S. (1997). Sub-pixel accuracy using recursive filtering. In *Scandinavian Conference on Image Analysis*. 115

Montesinos, P. and Magnier, B. (2010). A new perceptual edge detector in color images. In *ACIVS*, pages 209–220. Springer Berlin Heidelberg. 83

Montesinos, P. and Magnier, B. (2017). Des filtres anisotropes causaux pour une diffusion non controlled. In *GRETSI*. 113

Noble, J. A. (1988). Finding corners. *Image Vis. Comput.*, 6(2) :121–128. 99, 101

Osher, S. and Rudin, L. (1990). Feature-oriented image enhancement using shock filter. In *SIAM Journal of Numerical Analysis*. 115

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1) :62–66. v, 13

Papari, G. and Petkov, N. (2011). Edge and line oriented contour detection : State of the art. *IVC*, 29(2) :79–103. 15, 90

Perona, P. (1992). Steerable-scalable kernels for edge detection and junction analysis. *IVC.*, 10(10) :663–672. 45, 66, 82, 91

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7) :629–639. vi, 47, 48

Prewitt, J. (1970). Object enhancement and extraction. *Picture processing and psychopictories. B.S.Lipkin and A.Rosenfeld, ed. Academic Press, New York, USA*. v, 12, 23, 24

Pruvost, E., Tulet, H., Delort, E., Shokouh, G.-S., Montesinos, P., Magnier, B., Jourdan, C., Belaud, E., and Hedde, M. (2022). Invertebrates detection with yolov5 : Towards study of soil organisms using deep learning. In *10th European Workshop on Visual Information Processing (EUVIP). IEEE. September 2022,Lisbon*, volume A, page to appear. x, 133, 134

Rey-Otero, I., Delbracio, M., and Morel, J. M. (2015). Comparing feature detectors : A bias in the repeatability criteria. In *ICIP*, pages 3024–3028. 99

Roberts, L. (1965). Machine perception of three dimentional solids. . *Optical and Electro-optical Information Processing. J.T. Tipett and al. ed. M.I.T. Press Cambridge Massachussetts USA*. 23

Rodehorst, V. and Koschan, A. (2006). Comparison and evaluation of feature point detectors. In *International Symposium Turkish-German Joint Geodetic Days*. 99, 103, 105

Rohr, K. (1994). Localization properties of direct corner detectors. *Mathematical Imaging and Vision*, 4 :139–150. 99, 100, 101, 102

Rosten, E., Porter, R., and Drummond, T. (2008). Faster and better : A machine learning approach to corner detection. *TPAMI*, 32(1) :105–119. 99

Sanchez, C.-F., Ivan, C.-A., Arturo, H.-A., Martha Alicia, H.-G., and Sergio Eduardo, S.-M. (2019). Automatic segmentation of coronary arteries in x-ray angiograms using multiscale analysis and artificial neural networks. *Applied Sciences*, 9(24). 80, 89

Schmid, C., Mohr, R., and Bauckhage, C. (2010). Evaluation of interest point detectors. *IJCV*, 37 :151–172. 98, 99

Shi, J. and Tomasi, C. (1994). Good features to track. In *CVPR*, pages 593–600. 99, 101, 102

Shokouh, G., Magnier, B., Xu, B., and Montesinos, P. (2021a). An objective.comparison of ridge/valley detectors by image filtering. In *ICPR 2021 Workshop*, volume 12665, pages 182–197. Springer, Cham. 6

Shokouh, G.-S., Magnier, B., Montesinos, P., and Xu, B. (2022). Repeatability evaluation of keypoint detection techniques in tracking underwater image sequences. In *5th Workshop on Computer Vision for Analysis of Underwater Imagery (CVAUI)*, volume A, page to appear. 132

Shokouh, G.-S., Magnier, B., Xu, B., and Montesinos, P. (2021b). Ridge detection by image filtering techniques : a review and an objective analysis. *Pattern Recognition and Image Analysis*, 31(2) :to appear. vii, viii, 6, 81, 85, 86, 91, 92, 100

Sobel, I. and Feldman, G. (1968). A 3x3 isotropic gradient operator for image processing. *talk presented at the Stanford Artificial Intelligence Project (SAIL)*. 23, 24

Soria, X., Riba, E., and Sappa, A. (2020). Dense extreme inception network : Towards a robust cnn model for edge detection. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1912–1921. 133

Sponton, H. and Cardelino, J. (2015). A review of classic edge detectors. *Image Process. Line*, 5 :90–123. 32

Steger, C. (1998). An unbiased detector of curvilinear structures. *IEEE TPAMI*, 20(2) :113–125. vii, 61, 81, 91, 92

Tremblais, B., Capelle-Laize, A., and Augereau, B. (2007). Algorithms for the extraction of various diameter vessels. *Cellular and Molecular Biology*, 53(2) :62–74. 61, 91

Tuytelaars, T. and Mikolajczyk, K. (2008). *Local Invariant Feature Detectors : A Survey*. Now Foundations and Trends. 11, 103

Van Vliet, L., Young, I., and Verbeek, P. (1998). Recursive gaussian derivative filters. In *ICPR*, volume 1, pages 509–514. IEEE. 30, 69

Wang, H. and Brady, M. (1995). Real-time corner detection algorithm for motion estimation. *IVC*, 13(9) :695–703. 99, 100, 101

Werman, M. and Peleg, S. (1985). Min-max operators in texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(6) :730–733. v, 11, 12

Wirth, M. A., Fraschini, M., Maek, M., and Bruynooghe, M. (2006). Performance evaluation in image processing. *EURASIP Journal on Advances in Signal Processing*, 2006 :1–3. 35

Witkin, A. P. (1983). Scale-space filtering. In *IJCAI*. vi, 46, 47

Xia, G.-S., Delon, J., and Gousseau, Y. (2014). Accurate junction detection and characterization in natural images. *IJCV*, 106(1) :31–56. 92, 99

Zhang, L., Lukac, R., Wu, X., and Zhang, D. (2009). PCA-Based Spatially Adaptive Denoising of CFA Images for Single-Sensor Digital Cameras. *IEEE Transactions on Image Processing*, 18(4) :797–812. 73

Zheng, Z., Wang, H., and Teoh, E. (1999). Analysis of gray level corner detection. *Pattern Recognit. Letters*, 20(2) :149–162. 99, 100, 101

Ziou, D. (2000). Optimal line detector. In *ICPR*, volume 3, pages 530–533. 30, 63, 69, 92

Ziou, D. and Tabbone, S. (1998). Edge detection techniques : an overview. *Int. J. on Patt. Rec. and Image Anal.*, 8(4) :537–559. 23, 90

Zuniga, O. A. and Haralick, R. M. (1983). Corner detection using the facet model. *Pattern Recognit. and Image Processing*, pages 30–37. 99, 100, 101