Unsupervised vision methods based on image perceptual information

Eric Bazan

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à MINES ParisTech

Unsupervised Vision Methods Based on Image Perceptual Information

Méthodes de Vision non Supervisée
Basées sur l'Information Perceptive de l'Image

Soutenue par

Eric Bazán

Le 30 Juin 2021

École doctorale n°621

Ingénierie des Systèmes, Matériaux, Mécanique, Énergétique - ISMME

Spécialité

Morphologie Mathématique

Composition du jury :

M. Nicolas PASSAT Professeur, Université de Reims Champagne-Ardenne	*Président*
M. Thierry GÉRAUD Professeur, LRDE, École des Ingénieurs en Intelligence Informatique	*Rapporteur*
M. Petr MATULA Professeur Associé, Université de Masaryk, République Tchèque	*Rapporteur*
M. Gerado FLORES Professeur Associé, Centro de Investigaciones en Óptica, Mexique	*Examinateur*
Mme. Valery NARANJO Professeure, Universitat Politècnica de València, Espagne	*Examinatrice*
M. Petr DOKLÁDAL Chargé de Recherche, MINES Paris, Université PSL	*Directeur de thèse*
Mme. Eva DOKLÁDALOVÁ Professeure, ESIEE Paris, Université Gustave Eiffel	*Co-directrice de thèse*

MINES ParisTech | PSL

To my grandmother, La Juana.
You couldn't come to France but you charmed its people with your lifestyle.

Acknowledgements

These almost four years of thesis work have been a great experience both professionally and personally in my life. Along the way, I have met very valuable people whom I would like to thank because the Ph.D. would not have been what it has become without them.

First of all, I thank my thesis directors for accepting the challenge of guiding and introducing someone from the robotics area to the world of mathematical morphology. Thank you for trusting me and for the excellent guidance during this thesis and throughout the writing of this manuscript. I also thank M. Thierry Géraud and M. Petr Matula for agreeing to review my manuscript and be part of my jury; and Mme. Valery Naranjo, M. Gerardo Flores, and M. Nicolas Passat for agreeing to be part of my jury.

I also thank Anne-Marie and the great family of the CMM: Béa, Michel, Etienne, Samy, Bruno, Santi, Jesus, Jose-Marcio and François W. Thank you for the conversations about everything and nothing during coffee breaks and extracurricular activities, all that made me feel less far from home.

Without a doubt, I cannot but thank the group of Ph.D. students from DOPAMINES and CMM who introduced me (and continue to introduce me) to French culture: Laure, Jean, Marine, Aurelien, Pierre S., Albane, Kaïwen, Seb, Théo, Leo, David, Élodie, Francois, Nico, Robin, Bob Julien, Valentin, Mike, Ricardo, Alan, Bibiche, Tarek, Mateus, Angelique, etc. I really appreciated those beer times at the Glasgow, the tarot games, and the random and weird conversations that I occasionally did not understand.

To the people of the group of Hippies and Put1 Vegans (formerly Malakas) and other people I met during this time: Anais, Andrés, Fanny, Théo, Lydia, Benj, Élise, Mauro, Carol, Vincent, Cyril, Élsa, Sophie, Bassam, Pao, David, Laura. Thank you for becoming my new family and for your support during the bad times, but above all for the joys and all the good times we had.

Thanks to Chimpo for all her love and support throughout this thesis and to my family in Mexico, especially my parents Leonardo and Bertha and my sister Jocelyn for being so strong and supportive of my decision to do my Ph.D. abroad.

Finally, I thank CONACyT, the Center for Mathematical Morphology, and the ESSIE Paris for the financial support during the thesis work.

Abstract

This thesis work deals with extracting features and low-level primitives from perceptual image information to understand scenes. Motivated by the needs and problems in Unmanned Aerial Vehicles (UAVs) vision-based navigation, we propose novel methods focusing on image understanding problems. This work explores three main pieces of information in an image: intensity, color, and texture.

In the first chapter of the manuscript, we work with the intensity information through image contours. We combine this information with human perception concepts, such as the Helmholtz principle and the Gestalt laws, to propose an unsupervised framework for object detection and identification. We validate this methodology in the last stage of the drone navigation, just before the landing.

In the following chapters of the manuscript, we explore the color and texture information contained in the images. First, we present an analysis of color and texture as global distributions of an image. This approach leads us to study the Optimal Transport theory and its properties as a true metric for color and texture distributions comparison. We review and compare the most popular similarity measures between distributions to show the importance of a metric with the correct properties such as non-negativity and symmetry. We validate such concepts in two image retrieval systems based on the similarity of color distribution and texture energy distribution. Finally, we build an image representation that exploits the relationship between color and texture information. The image representation results from the image's spectral decomposition, which we obtain by the convolution with a family of Gabor filters. We present in detail the improvements to the Gabor filter and the properties of the complex color spaces. We validate our methodology with a series of segmentation and boundary detection algorithms based on the computed perceptual feature space.

Keywords: Image Processing, Low-level Primitives, Human Perception, Detection, Segmentation, Unsupervised Methods, Scene Understanding, Machine Learning, UAV.

Résumé

Ce travail de thèse porte sur l'extraction de caractéristiques et de primitives de bas niveau à partir des informations perceptuelles de l'image pour comprendre des scènes. Motivés par les besoins et les problèmes de la navigation basée sur la vision des véhicules aériens sans pilote (UAV), nous proposons de nouvelles méthodes en nous concentrant sur les problèmes de compréhension de l'image. Ce travail explore trois informations principales dans une image : l'intensité, la couleur et la texture.

Dans le premier chapitre du manuscrit, nous travaillons sur les informations d'intensité à travers les contours de l'image. Nous combinons ces informations avec des concepts issus de la perception humaine, tels que le principe de Helmholtz et les lois de la Gestalt, pour proposer un cadre non supervisé pour la détection et l'identification des objets. Nous validons cette méthodologie dans la dernière étape de la navigation par drone, juste avant l'atterrissage.

Dans les chapitres suivants du manuscrit, nous explorons les informations de couleur et de texture contenues dans les images. Tout d'abord, nous présentons une analyse de la couleur et de la texture en tant que distributions globales d'une image. Cette approche nous amène à étudier la théorie du transport optimal et ses propriétés comme véritable métrique de comparaison des distributions de couleur et de texture. Nous passons en revue et comparons les mesures de similarité les plus populaires entre les distributions pour montrer l'importance d'une métrique avec les propriétés correctes, telles que la non-négativité et la symétrie. Nous validons ces concepts dans deux systèmes de récupération d'images basés sur la similitude de la distribution des couleurs et de la distribution de l'énergie des textures.

Enfin, nous construisons une représentation d'image qui exploite la relation entre les informations de couleur et de texture. La représentation de l'image résulte de la décomposition spectrale de l'image, que l'on obtient par convolution avec une famille de filtres de Gabor. Nous présentons en détail les améliorations apportées au filtre Gabor et les propriétés des espaces colorimétriques complexes. Nous validons notre méthodologie avec une série d'algorithmes de détection des limites et de segmentation basés sur l'espace des caractéristiques perceptuelles calculé.

Mots clés: traitement d'image, primitives de bas niveau, perception humaine, détection, segmentation, méthodes non supervisées, compréhension de scène, apprentissage automatique, drone.

Contents

Introduction

In this thesis, we present the study of image information such as intensity, color, texture, and the relationship between them to extract low-level image primitives. Such primitives can characterize objects in images under various conditions, so we use them to build a representation of an image for high-level computer vision tasks such as scene understanding. We propose a novel methodology that combines this image information (intensity, color, and texture) with some human visual perception concepts. We focus on the image segmentation functionality of challenging applications performed under complex, uncontrolled conditions, with a lack of a priori knowledge. For this purpose, we favor traditional Computer Vision methods, which makes us independent of the disadvantages of today's methods: fine-tuning of parameters, a priori model, and explicability of the results. We obtain image features with a physical sense that can be used later in completely unsupervised algorithms. We validate such features and the proposed methodology in applications that are related and representing the problems of Unmanned Aerial Vehicles (UAVs) vision-based task.

Vision-based Techniques and Scene Understanding for UAVs Tasks

The methodology we propose in this thesis is an alternative to the difficulties in vision-based applications present in UAV tasks using the understanding of scenes. The advancement of computer vision techniques has favored their use in a wide range of applications. The development has been outstanding in already traditional application areas such as multimedia or medicine. However, new application areas such as augmented reality [Abu Alhaija et al., 2018], automated driving [Janai et al., 2020], robotics [Sankowski and Nowakowski, 2014], the Internet of Things (IoT) [Othman and

Aydin, 2017], Industry 4.0 [Zhong et al., 2017], human-computer interaction [Ke et al., 2018], and vision for the blind [Ahmed et al., 2020] continue to emerge.

Regardless of the application, computer vision systems must perform several tasks to achieve their goal. Generally, these tasks include techniques for acquiring, processing, analyzing, and understanding digital images; extracting real-world data to produce symbolic information, for example, in the form of decisions [Wiley and Lucas, 2018].

Scene understanding is the process that connects all these tasks to perceive, analyze and elaborate an interpretation of a dynamic 3D scene. Fig. 1 shows the pipeline of a conventional system for scene understanding. A system like this can use a wide variety of sensors (e.g., cameras, microphones, motion radars, among others) to characterize a scene [Bremond, 2007]. Therefore, this process consists mainly of relating information from monitoring sensors to models based on human observations and interpretations of the scene. We can then define scene understanding as to the crumbling of the symbolic image information using geometric, physical, statistical, or theory of learning models into descriptions of the world that can interact with other processes and provoke appropriate actions.

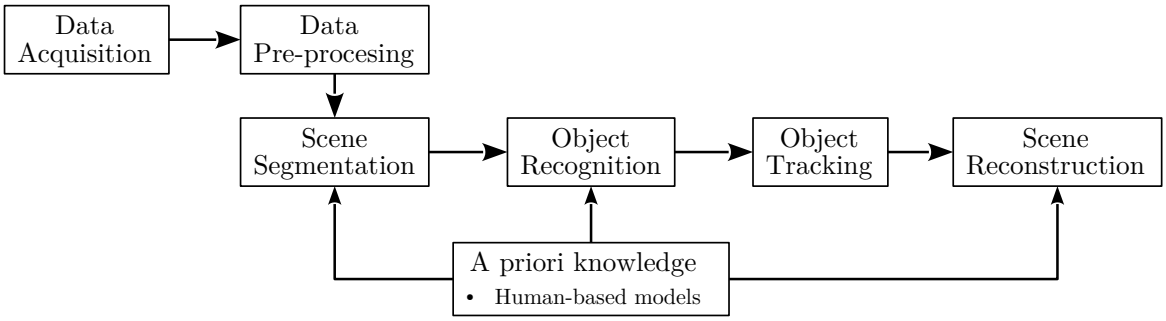


Figure 1: Typical pipeline of a scene understanding system.

Following the pipeline of Fig. 1, we can place the tasks of a scene understanding system into five general well-defined computer vision problems.

i **Data pre-processing**, whose objective is to remove the imperfections of an image generated by disturbances such as sensor noise or motion blur. Generally, we perform this task before passing it to a more complex algorithm. Image restoration and inpainting are some examples of this computer vision problem.

ii **Scene segmentation** is the process of partitioning an image into multiple (coherent) segments according to its features and properties. Depending on the application, we can formulate the image segmentation as the problem of classifying pixels with semantic labels (semantic segmentation), or partitioning of individual objects (instance segmentation), or both (panoptic segmentation).

iii **Object Recognition** is a classic computer vision problem responsible for determining whether an image contains an object, characteristic, or exercise. Some

variants of this problem are the classification, identification, and detection of objects from which many specialized tasks emerge. For example, content-based image search, pose estimation, optical character recognition, reading of 2-d codes, facial recognition, shape recognition, among others.

iv **Object Tracking** is the problem that searches to estimate the speed of one or more points of interest within an image or 3-d scene by processing a sequence of images. Some examples of similar task are egomotion, pose estimation, and optical flow.

v **Scene reconstruction** is the problem related to the computation of a 3-d model from one or more images of a scene. This model is intended to be a description of the scene as close to reality as possible.

These functionalities of computer vision and scene understanding systems are sought in the field of drones. UAVs (or drones) are flying engines that are increasingly present in our lives. We can find them in various sectors, such as the military, commercial or civil, where they can perform very specific tasks. However, in most cases, the development of such applications requires an expert pilot to control the aircraft.

Commonly, the UAV control is achieved using conventional sensors, such as inertial sensors (IMUs) for orientation and GPS for position. The combination of information coming from these sensors in a flight computer allows the drones to remain stable in the air. However, IMUs present some drawbacks; for example, they suffer from bias error propagation due to the integral drift. On the other hand, the GPS signal is not always guaranteed; for example, the satellite signal may be low or unexisting in urban or indoor environments. A recurrent technique to enhance the drone's position accuracy implies the data fusion of pressure, ultrasonic, radars, and laser range-finders sensors [Tomic et al., 2012]. The fusion of data can provide the advantages of each sensor. However, a significant limitation of these complex systems is flight time, a parameter mainly linked to the vehicle's total weight and the battery's capacity. Therefore, the use of multiple sensors onboard becomes expensive and impractical.

It is possible to extend the capabilities of a drone by integrating some visual sensor. Contrariwise to other sensors such as Lidars, visual sensors are passive, lightweight, and can acquire valuable information about the surrounding structures, including color and textures, and UAV's self-motion. The addition of visual sensors to perceive the environment has been a recurring strategy that has made these aerial robots more manipulable, safer, and even in some cases, autonomous [He et al., 2018], [Kyrkou et al., 2019], [Zhu et al., 2020]. That means that the drone can perform a task without the need for human intervention. For this, the drone must be able to move without getting lost; moreover, it must interpret and understand the present scene so that it can be able to detect and avoid potential obstacles on its way.

Today, one can use different visual sensors, such as monocular cameras [Padhy et al., 2018], stereo cameras [Seitz et al., 2006], RGB-D cameras [Huang et al., 2017], fish-eye cameras [Hrabar and Sukhatme, 2004], thermal cameras [Gaszczak et al., 2011], among others. This wide range of sensors offers more options and flexibility to deal with the problems mentioned above. The integration of such sensors in UAVs has allowed us to see the world from another perspective (literally), and the development of perceptual computer vision algorithms drives the technological improvement of these machines.

Today, there are applications in which vision algorithms have outstripped the capacity of human vision, so they have entirely replaced human personnel, for example, in industrial vision systems tasks, say, the inspection of production lines [Malamas et al., 2003]. However, in other imaging areas, computer vision systems are only responsible for supplementing specific routines that require a considerable amount of time and experience from human experts. This discrepancy in the vision systems' performance is mainly related to the complexity of the task and the environment's conditions where the task is performed. In industrial vision systems, we can control the working conditions in most cases, while in areas such as robotics and unmanned aircraft (or UAV), with uncontrolled conditions and without large databases, computer vision algorithms bring a real challenge, even though the acquisition system is the same.

Image Characteristics and Technical Locks in UAV Vision-based Applications

We can interpret the application and tasks made with drones as missions. Generally, such missions involve three central moments: take-off, navigation, and landing. The drone can perform such stages with conventional sensors; however, visual sensors provide valuable perceptual information about the environment.

Among the three moments that occur in drone missions, navigation and landing are the stages in which visual information (from onboard sensors) and computer vision algorithms most frequently intervene. In the landing stage, the needs and problems can be well-defined since it occurs at the end of the mission. Besides, we can control some conditions by adding pre-designed elements, such as landing targets or landing platforms, to facilitate the task. However, in the navigation stage, computer vision problems are mainly determined by the nature of drone applications.

Drone missions are generally carried out in complex scenes that change as the vehicle moves through space. For example, imagine all the scenarios that a delivery drone goes through during its mission: It can start its route in a commercial area, where the scenes mostly contain warehouses and big open spaces such as parking lots. Then, it could pass through rural areas, where the scenes can contain farmlands or wooded areas. Finally, when the drone reaches the delivery point within an urban zone, the environment may contain houses, trees, electricity, and telecommunications

poles, among others.

The mission through different environments generates considerable lighting changes and shadows, which results in overexposed and (or) dark images. Besides having no control over lighting conditions, we must also consider that the camera's position and orientation vary concerning the scene depending on the vehicle's height and orientation. Moreover, we must also consider the rolling shutter effect present in cameras using CMOS sensors. Therefore, the objects present in the images may have deformations because of the optic and the movement. Fig. 2 shows some images taken with a commercial drone in a natural setting. We can observe how the environment's lighting conditions and the nature of aerial applications introduce deformations to the images and objects present in the scene.

Finally, we must not forget that we acquire the input images from an onboard camera, which is generally not stabilized; therefore, the images may be noisy or blurry. Such problems limit a computer vision algorithm to be globally efficient in all or most situations.

(a) Presence of shadows

(b) Saturations

(c) Change of scale

Figure 2: Some examples of image degradations present in aerial imaging and UAV applications.

In addition to the problems related to the complex scene conditions, we must consider that a drone is subject to sudden changes in the environment, such as wind gusts, which can affect its stability and modify the visual information given by the onboard sensors. In such cases, the vision algorithms for drone navigation must process the input information fast enough to provide answers and transform them into real-time

decision actions.

Considering the conditions and problems of vision-based drone applications, we argue that a system for scene understanding is necessary for this kind of application. Moreover, scene understanding must use low-level information such as intensity, color, and texture in combination with perceptual tools to generate a robust image interpretation to the characteristic visual conditions of the aerial platforms. Among these classic computer vision problems, image segmentation is a crucial stage in the scene understanding pipeline. Considering the pipeline of an scene undertanding system Fig. 1, image segmentation is a mid-level task; however, it is critical for high-level applications such as object recognition, object tracking, scene reconstruction, and scene understanding. A robust segmentation to the changes and variants of complex environments allows the generalization of high-level tasks to new contexts and applications [Maninis et al., 2018].

Image Segmentation State of the Art

Image segmentation has a long history in computer vision and is present in many applications in medicine, biology, robotics, and physics. Here, we present a brief review of the state-of-the-art segmentation methods taking into account the thier relationship with vision perception (a more detailed version of the state-of-the-art of segmentation methods appears in chapter 6, section 6.2). For this purpose, we divide the image segmentation techniques into classical methods and Artificial Intelligence (AI) methods.

Classical Image Segmentation Techniques

Classical image segmentation methods can be organized into two groups: those that identify similarities or those that identify discontinuities [Zaitoun and Aqel, 2015]. The first kind of approach detects similar pixels in the image based on some specific threshold or criteria for split-merge and growing regions. The second category of methods tries to find the boundaries between dissimilar pixels in the image. A more specific classification according to the technique used divides segmentation methods into Threshold-based, edge-based, region-based, watershed-based, clustering-based, PDE-based, and Graph-based [Zaitoun and Aqel, 2015].

Threshold-based algorithms are one of the simplest image segmentation techniques. The threshold operation divides the image by comparing the intensity of the pixels to a specific threshold value [Sezgin and Sankur, 2010]. This kind of method can only segment images into background and foreground based mainly on the intensity pixel information. This property is convenient when there is a significant contrast difference between the objects and the background. The major challenge of such approaches is the choice of the threshold value. The simplest option is to use a global

threshold for the whole image; however, this option fails when the illumination in the image is uneven. Local thresholding methods solve this problem by proposing multiple thresholds [Niblack, 1986], [Sauvola and Pietikäinen, 2000]; however, the computation time can increase considerably. One of the most popular approaches in this category that automatically determine the threshold value is Otsu's method [Otsu, 1979].

The **Edge-based** segmentation methods attempt to solve the image segmentation problem by detecting edges in an image according to the differences in texture, contrast, grey level, color, saturation, and other properties [Saini and Arora, 2014]. Some of the more well-known methods in this category employ operators that use the first and second derivatives of the image to identify abrupt changes in the intensity of the image, for example, the Sobel [Sobel and Feldman, 1990], Roberts [Roberts, 1963], Gradient [Maître, 2003], Prewitt [Prewitt, 1970], and Laplacian [Marr and Hildreth, 1980] operators. On the other hand, one of the state-of-the-art reference work is the Probability-boundary (Pb) [Malik et al., 2001], which uses the intensity and color, and texture information to obtain the edges of the image.

The **Region-based** segmentation methods partition the image into similar regions according to predefined criteria. Depending on the strategy used to arrive at the final segmentation, they can be organized into region growing and splitting and merging techniques [Sezgin and Sankur, 2010]. Region growing techniques define a group of seed pixels from which regions start to grow [Adams and Bischof, 1994; Zucker, 1976]. Regions grow by appending to each seed pixel those neighboring pixels that have predefined properties similar to those of the seed pixels (e.g., intensity or color). Regions stop growing when they reach a particular predefined stop criterion (e.g., size or shape of the region). Conversely, splitting and merging techniques do not require seed pixels. This technique successively divides the image into quadrants based on a homogeneity criterion, then similar regions are merged to form the final segmentation. This strategy includes the quad-tree data structure [Horowitz and Pavlidis, 1976], which means a parent-child node relationship. In practical applications, the region growing and splitting and merging algorithms are usually used in combination [Ikonomatakis et al., 1997]. This combination is more effective for the segmentation of complex scenes defined by some complex objects or the segmentation of certain natural scenes, such as image segmentation with insufficient prior knowledge.

The **Watershed-based** segmentation is a technique that utilizes image morphology and combines the characteristics of edge- and region-based methods described above. First, this method computes the gradient of an image. We can see this gradient as a map that reflects the topography of the image through the intensity values of the pixels. Then, segmenting an image is equivalent to flooding the topography from a group of seed pixels, where the edges of the image appear as the highest ridges where the flood water meets [Beucher and Meyer, 1993; Meyer and Beucher, 1990]. The watershed method is strictly linked to hierarchical segmentation methods [Najman and

Schmitt, 1996]. This feature of hierarchical dependence complexifies the efficient implementation in embedded processors. Many strategies introduce other definitions of the watershed transform to solve the complexity problem, which simplifies and accelerates its computation [Roerdink and Meijster, 2000] , [Dejnozkova and Dokladal, 2003a], [Chabardès et al., 2016].

Another alternative to obtain the segmentation of an image is by using clustering methods. The **clustering-based** segmentation methods are unsupervised techniques that classify the image pixels into clusters (disjoint groups) with similar features. The objective of pixel clustering is to maximize inter-class differences and minimize intra-class differences; that is, the pixels in each class should be as similar as possible, and those in the different groups should be as different as possible [Steinley, 2006]. The k-means technique is known as a hard-clustering technique since each pixel can belong only to one class. Fuzzy algorithms (soft-clustering) relax that condition, and each data point can belong to more than one cluster. This behavior is suitable in applications where there are no crisp boundaries between objects, such as tissue classification [Caldairou et al., 2011] and tumor detection [Preetha and Suresh, 2014]. Among the soft-clustering methods, fuzzy C-means clustering [Dunn, 1973] is one of the most used.

PDE-based segmentation methods use Partial Differential Equations to model the image contours and obtain an image segmentation. Active Contour Model (or Snakes) transform the segmentation problem into PDE. Some famous methods of PDE used for image segmentation are Snakes [Kass et al., 1988], Level-Set [Osher and Sethian, 1988], Fast Marching [Forcadel et al., 2008], and Mumford Shah method [Mumford and Shah, 1989]. One of the main problems of these methods is the high computational time for the resolution of the PDE, which limits its use on embedded platforms. This limitation has been addressed in the implementation level through architectures that allow multi-core parallel calculation [Dejnozkova and Dokladal, 2003b, 2004].

The last group of classical methods for image segmentation is **Graph-based**. These methods utilize graph theory and represent images or their parts as graphs. Typically, a pixel or a group of pixels are associated with nodes, and the edge weights define the affinity between neighboring pixels. Then, we can partition the graph according to a criterion designed to model good clusters. Each resulting partition of nodes is considered a segmented object in the image. Some popular algorithms in this category are normalized cuts [Jianbo Shi and Malik, 2000], random walker [Grady, 2006], minimum cut [Wu and Leahy, 1993], isoperimetric partitioning [Grady and Schwartz, 2006], and minimum spanning tree [Zahn, 1971]. Some of the segmentation methods can combine strategies. For example, spectral clustering [Ng et al., 2001] uses the graph theory and the similarity of the graph edges to cluster the image pixels into coherent regions. On the other hand, [Cousty et al., 2009] define the watershed cuts cut on edge-weighted graphs using the Minimum Spanning Forest.

This group of approaches and techniques (known as classical, conventional, or tra-

ditional) can use structural, stochastic, or hybrid techniques to segment an image. Structural techniques require structural data from the image, such as distributions, histograms, pixel density, or color distribution. Stochastic techniques require information about the discrete values of the pixels. Machine learning methods, such as the clustering techniques, fall into this category. Finally, hybrid techniques may use structural information of image regions and the discrete values of the pixels of the whole image for the segmentation. The choice of the method to segment an image depends on the type of image and the type of segmentation that we seek to obtain (for example, over-segmentation or segmentation to pixel precision). Regardless of this, we consider it essential to consider the perceptual elements of the data to achieve a meaningful interpretation of the scene.

AI Image Segmentation Techniques

The **Artificial Neural Networks- (ANN) based** techniques (a.k.a. Deep Learning (DL) techniques) are probably the most widely used methods today because of their efficiency and accuracy. Based on the learning rules and training process, learning in ANNs can be sorted into supervised, reinforcement, and unsupervised learning. Reinforcement and unsupervised learning are different from each other in many aspects. Reinforcement learning includes learning policy by maximizing a few rewards. The objective of unsupervised learning is to exploit the similarities and differences in the input data. Unsupervised learning plays out the tasks of pattern recognition and data dimensionality reduction. Some models of unsupervised ANN are Boltzmann Machines [Salakhutdinov and Hinton, 2009] (and its variations, such as Restricted Boltzmann Machines [Fischer and Igel, 2012]), Auto-Encoder architectures [Makhzani et al., 2016], and Variational Auto-Encoders (VAE) architectures [Doersch, 2021].

The techniques mentioned above are unsupervised learning methods, although technically, they are trained using supervised learning methods, referred then to as self-supervised. Supervised techniques require an annotated database for training, validation, and testing. In this case, the input neurons can correspond to the pixel value of an image or to complex data like graphs and multi-dimensional points (Geometric Neural Networks [Bronstein et al., 2017]), which means the image information is mapped to the neural network. Then, the image in the form of the neural network is trained using labeled data to find the connection between neurons. Lastly, the new images are segmented from the trained model.

In recent years, neural network techniques have led to new models for image segmentation. We can classify these methods roughly according to the architecture they use[1]. Convolutional Neural Networks (CNNs) are among the most widely used and suc-

[1]The architectures often differ based on the solved task, e.g., image classification, object detection, semantic segmentation, instance segmentation. For a detail classification please see [Sultana et al., 2020] and [Minaee et al., 2021]

cessful architectures in computer vision. This model, initially proposed by Fukushima [1980], is inspired by the model of the human visual cortex.S Some of the best-known CNNs in the literature include LeNet [Lecun et al., 1998], AlexNet [Krizhevsky et al., 2012], VGGNet [Simonyan and Zisserman, 2015] and ResNet [He et al., 2016].

Due to their characteristics, CNNs may require dense layers for pixel-level prediction, which means a huge number of parameters to learn, making it highly computationally expensive. Fully Convolutional Networks (FCN) [Long et al., 2015] solve this drawback by stacking several convolution Layers with similar padding to preserve the dimension and output a final segmentation map of the same size as the input image. Some of the best-known models are VGG16 and GoogleNet [Szegedy et al., 2014].

Other deep learning backbones are the Encoder-Decoder and Auto-Encoder architectures. This type of model is known as two-stage networks. The first stage, encoding, compresses the input information into a space-latent representation, while the second stage, decoding, predicts an output from the representation. Some examples of networks that follow this architecture are DeConvNet [Noh et al., 2015], SegNet [Badrinarayanan et al., 2017], U-Net [Ronneberger et al., 2015], W-net [Xia and Kulis, 2017], Linknet [Chaurasia and Culurciello, 2017], among others.

Object detection and image segmentation are complementary tasks in computer vision. Consequently, some architectures for object detection, such as Regional CNN (R-CNN), have been successfully adapted for image segmentation. Some examples are the Faster R-CNN [Ren et al., 2017], Mask R-CNN [He et al., 2020] and Masklab [Chen et al., 2018] architectures. The operation principle of these architectures is to extract the features of certain regions of interest to infer the class and the coordinates of the bounding box of the object.

A very recent family of architectures are those based on Generative Adversarial Networks (GANs) [Goodfellow et al., 2014]. This architecture consists of two networks, a generator, and a discriminator. The generator has the task of reproducing distributions similar to the real samples. On the other hand, the task of the discriminator is to distinguish the fakes samples from the real ones. GANs models include Convolutional-GANs [Radford et al., 2016], Conditional-GANs [Mirza and Osindero, 2014], and Wasserstein-GANs [Arjovsky et al., 2017].

Other popular DL architectures for image segmentation include Feature Pyramidal Networks (FPN) [Lin et al., 2017], which takes a multi-scale approach, or hybrid ones that combine classical methods such as the Active Contour Model [Kass et al., 1988] and CNNs, or the watershed transform in the deep watershed architecture [Bai and Urtasun, 2017].

The literature on methods based on DL architectures for image segmentation is vast. For a more detailed survey of the state of the art of ANN-based methods for image segmentation, please check [Sultana et al., 2020] and [Minaee et al., 2021].

AI-based image segmentation methods are experiencing popularity and growth that

has benefited from advancements in computing power and the recent creation of publicly accessible annotated databases. However, its use in particular applications where there are not (yet) large enough annotated databases is complicated. Furthermore, despite the performance in challenging benchmarks, arguably, the best-known disadvantage of neural networks is their *black box* nature, i.e., we are unaware of how or why a ANN obtained a certain output. This yield to a lack of result interpretability, especially in classification tasks.

Vision-based UAV Navigation Related Works

In the literature, we can find many works that deal with computer vision for drone navigation. The different approaches are strongly related and motivated by the application's aim and the conditions in which the task is developed. We can differentiate two main techniques for UAV navigation; 1) localization and mapping and 2) obstacle avoidance.

Simultaneous Localization and Mapping (SLAM) falls within the first group of techniques, where drone navigation is a necessary (but not sufficient) condition for drones to acquire the ability to navigate. This technique estimates the drone's local pose and builds a 3-d model of its surroundings employing visual sensors. Visual Odometry (VO) [Scaramuzza and Fraundorfer, 2011] is responsible for the robot motion estimation while the maps are built with occupancy grid algorithms [Thrun and Bü, 1996]. According to the image information used to perform a SLAM, we can classify these approaches into feature-based methods, which extract a set of image features (e.g., lines, points) in a sequence of images, and direct-based methods, which make use of the image intensity information to estimate the structure and the motion of the robot [Taketomi et al., 2017]. The importance of a correct segmentation of the image is that we can also create a depth chart of the scene from it and consequently achieve the visual odometry [Drouyer, 2017; Drouyer et al., 2017].

The use of SLAM techniques for UAV navigation presents remarkable advantages. Feature-based methods can use various feature detectors, which typically count with an optimization stage to produce fast algorithms. Direct-based methods have the advantage of being robust to image degradations; they can deal better with images with texture and blurred zones; besides, the map produced is of an acceptable resolution. Interestingly, the strengths of the first group of methods are the weak points of the second and vice versa. A method that tries to gather the benefits of both approaches is the Semi-direct Visual Odometry [Forster et al., 2014]; however, in general, the state-of-the-art SLAM methods is more mature in the autonomous vehicle environment [Singandhupe and La, 2019].

There are approaches for drone navigation that, in parallel to SLAM, favor the avoidance of obstacles. This capability is essential for achieving free collision missions

in both indoor and outdoor environments. A recurrent solution, as we early mentioned, is the multi-sensor data fusion. Gageik et al. [2015] present a platform using low-cost ultrasound and IR sensors; however, despite the obtained results, it utilizes several sensors to retrieve environment information, and yet, it does not get a perceptual representation of the scene due to the low resolution and perceptive capacity of the sensors. On the other hand, vision-based techniques for obstacle avoidance could detect obstacles and, in some cases, recognize and classify the object representing the obstacle [Li et al., 2016].

We can classify the visual methods for avoidance of obstacles into two groups. The first, SLAM-based techniques, make use of the principles stated above. The 3-d reconstruction provides accurate and sophisticated maps and allows the air vehicle to travel with more information about the environment. Moreno-Armendáriz and Calvo [2014] take this advantage to develop an obstacle avoidance approach for static and dynamic obstacles. The second group is the flow-based methods which historically, were inspired by the navigation of insects such as bees [Srinivasan and Gregory, 1992] or flies [Franceschini et al., 2009]. Many insects in the wild identify obstacles through the intensity of light. During the flight, their eyes produce an optical flow that provides accurate spatial information. Currently, there are also works inspired by the behavior of the human eye [Al-Kaff et al., 2016]. The technique measures the object size from the idea that objects in the robot's vision field are more significant as the obstacle is close.

In general cases, obstacle avoidance techniques are strongly linked to the camera parameters and acquisition conditions. The algorithms are often fine-tuned. Given the condition, a drone operates in an environment without prior knowledge and under uncontrolled conditions. Hence some more general, unsupervised methods are needed.

Today, the most efficient algorithms are those based on Neural Network (NN) architectures and supervised learning techniques. Nevertheless, these techniques have remarkable disadvantages, for example, their dependence on large annotated databases and the related uncertainty of whether an available database is sufficient or not. That question their usability and applicability in real-life drone missions [Treboux et al., 2018]. From a practical and even economic point of view, there is a limit to the number of applications in which we can use supervised methods given the fact that we need a lot of annotated data [Xu et al., 2020]. The collection and the correct labeling of data representing a problem are valid only for a small number of applications.

The need for abundant information comes with high computational times required for model learning, ranging from a couple of hours to weeks. Of course, we can minimize this variable by increasing our machines' computing power; however, today, only those with large computing infrastructures can afford to train models with hundreds of billions of parameters.

The above statement introduces the next disadvantage of deep neural network-

based learning models: hyperparameters. We can roughly divide hyperparameters into two categories: 1) optimizer hyperparameters, which include the learning rate, the batch size, and the number of epochs, and 2) model-specific hyperparameters, which include the number of hidden layers, the first hidden layer, and the number of layers. Choosing the appropriate hyperparameters plays a crucial role in the success of neural network architectures because they control the learning algorithm's behavior, define the network structure, and define how the network is trained. Although there are methods to optimize their choice, generally, this task is a heuristic process, and their fine-tuning is a function of the specific application. It is possible to follow some rules based on experience, copy the same values from some other problem or make the setting by trial and error, though we cannot know the best value for a hyperparameter.

We can thus conclude that it is crucial to have the means to understand the scene, depending less on the parameters fixed in advance or the data sets prepared for a particular mission. In the following paragraphs, we present the contribution of this thesis to this issue.

Scope of the Thesis

The interaction between computer vision and applications made with unmanned aerial vehicles is extensive. This collaboration has generated new methodologies and approaches, both theoretical and practical, but has also given way to new research questions.

Knowing the fundamental limitations of aerial robots and the complexity of drone applications, we explore computer vision theory to propose algorithms that improve and provide assistance in drone navigation tasks. In this sense, we are interested in studying the scene's perceptual information for their treatment and interpretation.

We focus primarily on the perceptual image decomposition to develop computer vision taks such as the perceptual segmentation and the recognition/classification of objects (see Fig. 3). We argue that these computer vision tasks are crucial for image understanding and have to be carefully treated to be robust with images under complex and uncontrolled environments (see Fig. 2. From this perspective, we focus on using low-level image features to extract perceptual information.

Based on Fig. 1 and considering the scope of the work in this thesis, we propose a new pipeline for scene understanding systems. In this new pipeline, the pre-processing of the image involves its perceptual decomposition, while the segmentation stage considers the perceptual elements of the image. The proposed pipeline for scene understanding is showed in Fig. 3. The peculiarity of this pipeline is that it eliminates the dependency on the a priori models, at least for the tasks of segmentation and image recognition (tasks that we study in this thesis).

Throughout this work, we develop algorithms that are capable of handling a variety

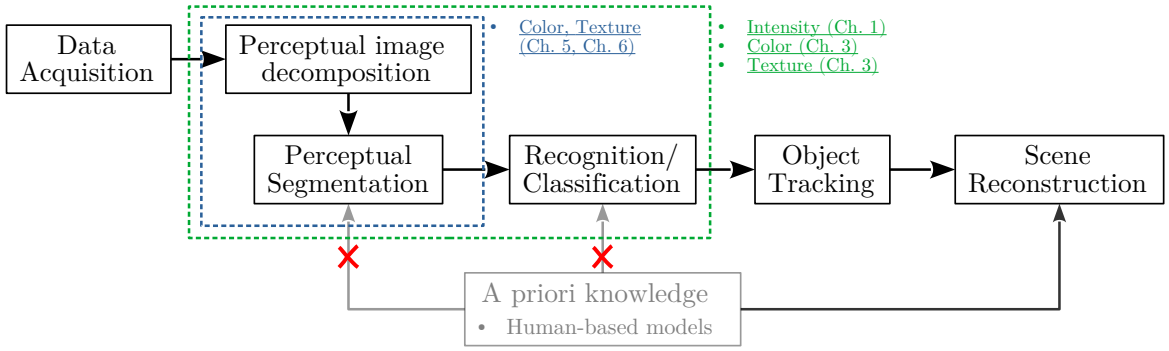


Figure 3: Proposed pipeline of a scene understanding system.

of real-world conditions. All these algorithms aim to segment the physical objects that we, as humans, define as perceptually interesting. For this purpose, we use intensity, color, and texture image information to extract low-level primitives, such as contours. In Fig. 3, we show the stages of scene understanding that we study in this thesis and the low-level primitives involved in the task: in green, the recognition and classification of objects use intensity, color, and texture, and in blue, the perceptual segmentation uses color, texture and the relationship between them. We use these primitives in conjunction with statistical and geometric tools from computer vision and signal theory, such as anomaly detector, optimal transport, and Gabor functions. Instead of using supervised methods, we focus on decomposing the image information from the point of view of signal theory and physics to use it later on non-supervised or mathematical morphology methods.

Regarding the nature of the input data, we use only gray-level or color images as input information, favoring monocular cameras among the wide range of visual sensors reviewed previously. This choice allows replicating the algorithms with low-cost cameras that can be easily embedded in a drone.

Contributions of the Thesis

The primary objective of this Ph.D. thesis is to propose a new methodological framework for the perceptual decomposition of natural images. Such a framework should use only the low-level perceptual information of the image: intensity, color, and texture. Moreover, this framework must contemplate and intelligently handle the dimensions and order of each image cue's dimensions, e.g., the frequency and orientation of textures or the distance between two colors. This property will allow combining the cues without defining parameters of importance for each of them. With this strategy, we intend to remove the need for a learning base necessary for model training.

The perceptual decomposition of the image is inserted in the scene understanding pipeline, so we propose to validate this framework in tasks such as perceptual image segmentation and object detection and classification. Therefore, some secondary ob-

jectives include quantitative and qualitative measurement of segmentation and object detection/classification algorithms.

Finally, we consider vision-based UAV applications as the advanced application of a scene understanding system. Therefore, our idea is to apply the methodological framework to assist control and decision-making in drone navigation tasks. To this end, the framework must be robust to image degradations existing in environments with uncontrolled conditions, in addition to being independent of the choice of specific parameters for its operation.

We extend the study of primary image information such as intensity, color, texture, and texture color and low-level image primitives such as contours. Therefore, some secondary objectives involve building a representation of the image in feature space using concepts from signal theory, geometry, and statistics, in addition to concepts from human perception. We seek to validate the proposed image representation using unsupervised approaches in real applications following traditional machine learning and segmentation algorithms.

Fig. 4 shows a general flowchart of the contributions of this thesis. The first stage of the flowchart shows low-level information and the methods we use to extract it; here, we work with intensity, color, texture, and the relationship between color and texture. The second stage of the diagram shows the different representations of the image obtained from the perceptual information contained in the primitives. Finally, the diagram's third section collects the different applications used to validate the image's feature spaces constructed in this thesis. Although the idea of a single framework itself is an ambitious goal, in this thesis, we present several algorithms that apply the proposed methodology with one or more features to solve different computer vision problems such as object detection and recognition, image retrieval systems, perceptual object boundaries detection, and image segmentation.

The interest of obtaining a representative space of the image information from low-level hand-made features lies in the possibility of using it in a semi-supervised pipeline. By injecting annotated information into the frame, it might make generalizations and obtain medium- or high-level features such as the importance of color and texture information to a human when segmenting an image.

Specifically, the contributions of this thesis include:

- A novel non-parametric framework for fully unsupervised object detection robust to the image degradations present in complex, uncontrolled environments (chapter 1).
- A qualitative and quantitative study between the most popular measures in the comparison of distributions (chapter 3).
- An unsupervised image retrieval system based on global color/texture information (chapter 3).

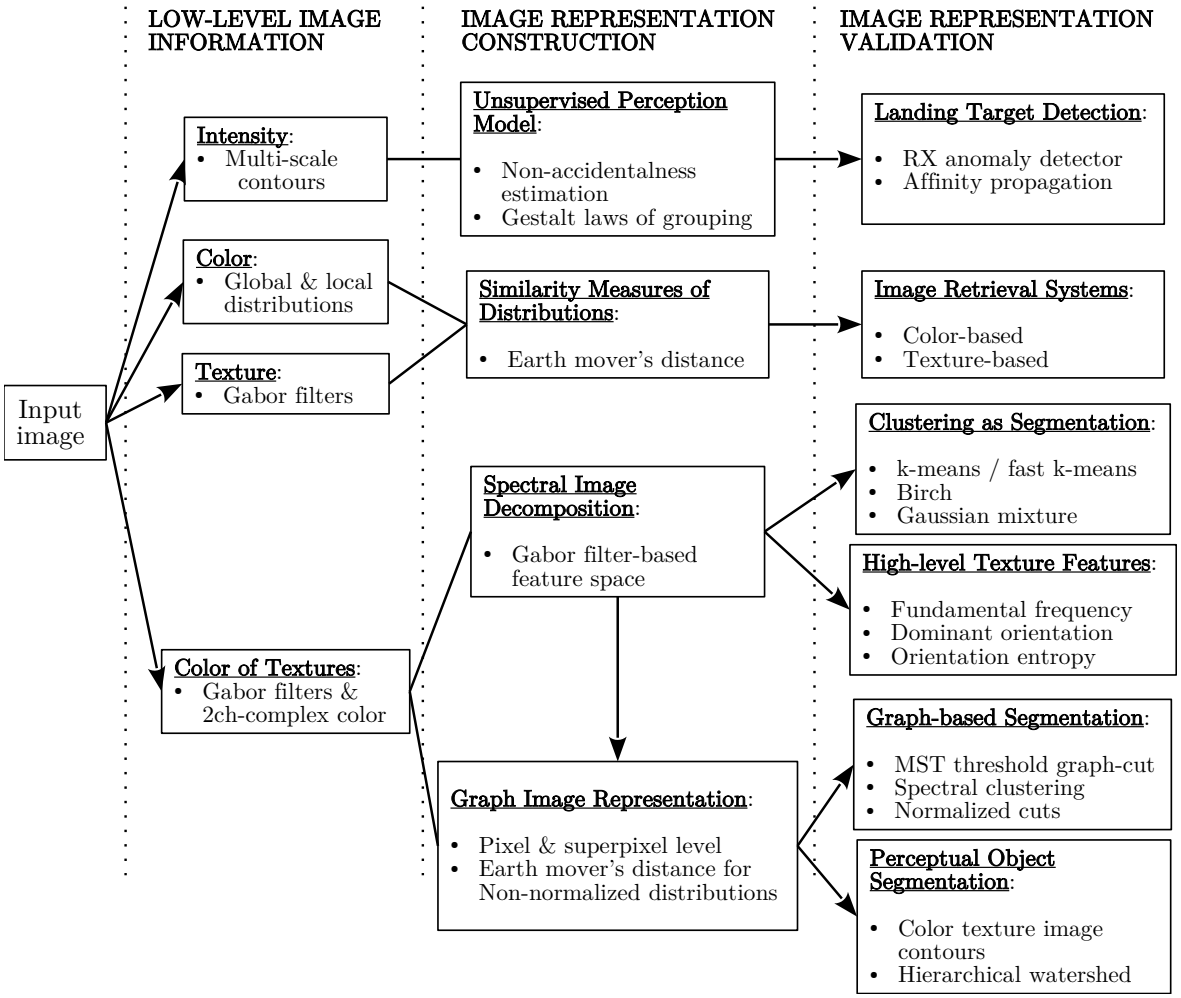


Figure 4: Flowchart of thesis contributions.

- Extensive analysis of Gabor filters and their properties in the space-frequency domains (chapter 4).
- Generation of a feature space that includes the color and texture information of an image (chapter 5).
- Unsupervised framework for natural image segmentation (chapter 6).

Finally, this thesis aims to show that traditional computer vision methods are (still) a reliable option to develop object detection and recognition for relatively complex tasks. We place this argument in the current context of computer vision, where there are hundreds of algorithms based on Neural Networks and Artificial Intelligence. Besides the NN and AI algorithms for image segmentation and object detection are highly performant, they lack a physical (and in many cases logical and argued) explanation and interpretation [2] of its results.

[2]Interpretability is also important for image segmentation (not only for classification task), because we have to understand how it works before we can say under what conditions it works and where the limitations are.

Organization of the Document

To communicate our proposal and the objectives mentioned above in a clear and structured way, we present a thematic and chapter organization of the document. The thematic organization follows, to some extent, the complexity of the three low-level image information we used during this thesis: intensity, color, and texture. Then, we can identify three main parts in this thesis.

1. The first part is dedicated to studying the intensity information of the image, in which we review in detail some of the classic methods for obtaining image contours. We use this information in conjunction with the *a contrario* method and the Gestalt organizing laws to detect and identify landing targets. This part includes chapter 1.

2. The second part main topic is studying the properties of color and texture of an image. We are interested in the global distribution of this information and the existing metrics to measure the similarity between the distributions; we apply and validate these concepts in two image retrieval systems. This part covers from chapter 2 to chapter 4.

3. The third part extends the study of color and texture in images, exploding the local distribution of these primitives and studying the influence of color information on the generation of textures in an image. We propose a multi-spectral image decomposition helpful on the object segmentation tasks using classic clustering algorithms and for the generation of high-level texture features. Moreover, we propose a completely unsupervised framework for the detection of perceptual boundaries. We also explore different strategies to obtain the segmentation of natural images using the obtained perceptual boundaries. This part includes chapter 5 to chapter 6.

The organization by chapters is structured as follows:

- **Chapter 1** addresses the bases of the Gestalt theory, including the grouping laws and the Helmholtz principle. We formalize these concepts of human perception mathematically and formulate a non-parametric algorithm that follows an unsupervised framework based on an image's contours. We use the developed framework in the autonomous drone landing problem, specifically detecting and identifying landing targets. The chapter also presents a review and a quantitative comparison of different traditional methods for extracting image contours.

- **Chapter 2** presents a detailed review of the different ways to represent the color and texture information in an image. The chapter contains a review of various color spaces and their main properties and an introduction to the different

techniques for characterizing textures in the literature. Such information is of relevant importance in constructing the framework and the approaches to measure similarity between distributions.

- **Chapter 3** presents the analysis between different similarity measures between distributions, showing the advantages and disadvantages of each of them. In particular, we focus on the theory of optimal transport through the Earth Mover's Distance. We show the advantages of this metric over traditional similarity measures using an image retrieval system based on an image's global color and texture information.

- **Chapter 4** explores the physical and human perception aspects of Gabor's filters. We show the steps involved in designing an optimized and efficient Gabor family of filters. The proposed filter family models and captures the texture information through an energy-efficient decomposition of the image. Such spectral decomposition of the image deals with Heisenberg's uncertainty principle. The chapter presents the description of parameters that allow complete customization of the filter family according to the application.

- **Chapter 5** brings an analysis of the texture information present in color images, showing the strong relationship between those two features. Using the spectral analysis of an image with the previously defined Gabor filters, we generate a feature space that simultaneously captures the color and texture information. We show the richness of such feature space by performing unsupervised image segmentation only using simple clustering techniques. Moreover, we show some novel high-level texture features resulting from the spectral image decomposition.

- **Chapter 6** introduces a framework for detecting perceptual boundaries of objects present in natural images. This framework brings together concepts addressed in this document, such as the spectral decomposition of images, the optimal transport as a true metric, and the relationship between color and texture information. Besides, using the hierarchical segmentation technique, we segment natural images in an unsupervised manner. We perform a quantitative and qualitative validation of our method using a known database.

Chapter 1

Intensity Image Contours and Information's Perceptual Organization

Résumé

Dans ce chapitre, nous nous intéressons à l'intensité d'une image en tant que primitive de bas niveau. Nous explorons cette primitive à travers les contours d'intensité de l'image à plusieurs échelles. L'idée est de proposer un modèle sans paramètre pour détecter des objets suffisamment robustes aux conditions présentes dans des environnements complexes et incontrôlés. Le défi est de savoir comment gérer les perturbations d'image telles que les ombres générées par les changements d'intensité, les changements d'échelle et de perspective, les vibrations, le bruit, le flou, entre autres. Nous introduisons un modèle robuste non supervisé qui nous permet de détecter un objet d'une manière inspirée par la perception, en utilisant les principes de Gestalt de non-accidentalité et de regroupement. Notre modèle extrait les contours les plus importants de l'image en tant que valeurs aberrantes à l'aide du détecteur d'anomalies RX, puis calcule une mesure de proximité et de similitude. Nous validons notre modèle dans une application présente dans la dernière étape de navigation d'un drone: la détection de cible d'atterrissage. De plus, nous montrons le code Hamming de correction d'erreur pour générer des cibles d'atterrissage numérotées et réduire les erreurs de reconnaissance.

Abstract

In this chapter, we are interested in the intensity of an image as a low-level primitive. We explore this primitive through the image intensity contours at multiple scales. The idea is to propose a parameter-free model for detecting objects sufficiently robust to conditions present in complex, uncontrolled environments. The challenge is how to deal with image disturbances such as shadows generated by intensity changes, scale and perspective changes, vibrations, noise, blur, among others. We introduce a robust unsupervised model that allows us to detect an object in a perception-inspired manner, using the Gestalt principles of non-accidentalness and grouping. Our model extracts the most important contours from the image as outliers using the RX anomaly detector and then computing a measure of proximity and similarity. We validate our model in an application present in the last stage of navigation of a drone: the landing target detection. Furthermore, we show the error correction Hamming code to generate numbered landing targets and reduce the recognition errors.

1.1 Introduction

We consider that scene understanding is an essential aspect in image analysis for the robot navigation problem. Then, the proposal is to build a framework that obtains perceptual information using low-level primitives of the image. The idea is to focus on the first problems of scene understanding and object recognition tasks: detecting and identifying objects.

This chapter of the thesis focuses on studying the intensity information in an image and its properties. We review some concepts of human perception, such as Helmholtz's principle, and we interpret and apply them to the contours extracted from the intensity information. Specifically, we use the non-accidentalness (*a contrario* approach) of the image contours to avoid modeling the possible objects to detect. Instead, we detect objects as a deviation of the normality represented by a random configuration model. This approach is inspired by postulates of works from the beginning of the 20th century that we strive to formalize mathematically. We use this algorithm for the autonomous drone landing task. Therefore, the objects to be identified are a series of numbered landing targets specifically designed for this problem.

The main contributions of this chapter are:

1. Study and comparison of different classical approaches to image contours detection.

2. A new methodology that uses the *a contrario* method and Gestalt theory for searching and interpreting image perceptual information using the geometrical properties of intensity contours.

3. A novel non-parametric framework for fully unsupervised object detection robust to the image degradations present in complex, uncontrolled environments.

1.1.1 Landing Target Detection Problems and State-of-the-art

Nowadays, the target detection for UAVs' autonomous landing is a recurring subject in the industrial sector. This task is crucial so that applications such as air parcel delivery can be developed. Some strategies to address this problem are creating landing stations; infrastructures that could harbor extra elements, such as GPS, infrared markers, or telecommunication sensors, which serve to locate and differentiate the landing zone from other areas. This option may be feasible for small-scale applications; however, in applications that require multiple landing points, this becomes impractical.

Instead, we propose a monocular vision-based system for the detection and identification of custom landing targets. For this, we imagine the situation when a drone is ready to land as follows: first, the drone reaches a certain horizontal/vertical distance from a possible landing target, then it activates the visual system and analyzes the scene where there may be a target; if the drone recognizes a pattern as a landing target and the ID is correct, the drone lands. The Fig. 1.1 represents the actions that a drone must perform to land at the correct point.

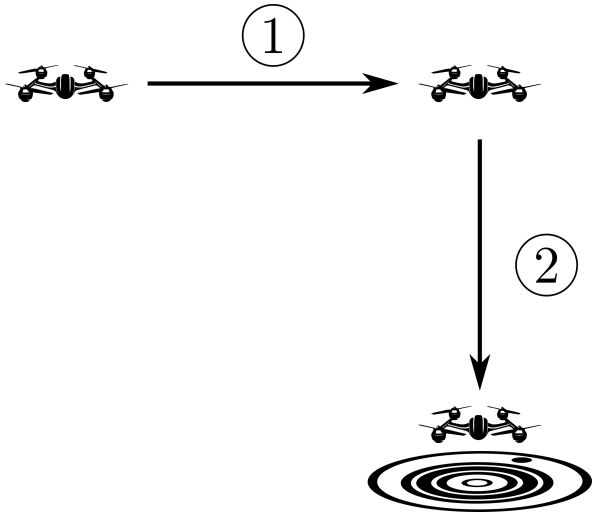


Figure 1.1: Graphic representation of the two stages involved in vision-based autonomous landing: 1. Approach to the landing zone; 2. Detection and recognition of the landing target.

Notwithstanding, using visual systems in outdoor environments presents challenges as many uncontrolled variables affect and impair the object detection task. The main problems to face in the aerial landing target detection task are the non-controlled light changes that generate shadowing, reflectance, and saturation on the surfaces; the perspective and distance of the camera that deforms the objects; the motion and

vibrations that blur the images and; the noise generated by a low-quality sensor.

Regarding the strategies and algorithms to detect a landing target, we can model this task as an object detection problem, where the two primary purposes are: Identify the object within the image and locate the object within the image. Object detection techniques draw bounding boxes around the detected objects. These bounding boxes give information about the location of the detected object and what object it is; however, we cannot accurately estimate some measurements such as the area or perimeter of an object in the image. On the other hand, image segmentation as a further extension of object detection helps us gain a more particular understanding of the shapes/curves of objects and also to know to which class each pixel of the image belongs. These two tasks of computer vision (object detection and object segmentation) are related, therefore, we can consider landing target detection as a segmentation problem, where there is a wide range of developed methods.

The variational framework [Mumford and Shah, 1989] offers a general method for image segmentation; however, its mathematical complexity and the endless selection of fidelity and regularization parameters make its use complex. Also, the number of iterations needed to find the solution makes it impossible to have real-time results. Conversely, threshold-based methods have been used to detect landing targets [Lacroix and Caballero, 2006], [Lange et al., 2008] for their ease of use. However, to achieve good detection, their use is limited to indoor spaces, where the light conditions are controlled [Araar et al., 2017].

Recently, convolutional neural networks (CNN) techniques offer the possibility to recognize (detect and segment) objects from a large set of classes with high reliability [Carrio et al., 2017]. Nevertheless, we need to train CNN methods with a database containing the object classes in a wide range of situations, and, in case of changes in the object or the scene, the database must be rebuilt [Yao et al., 2017], [Furukawa, 2018]. Besides, in some cases, the computation is carried out off-board the drone, implying the need for network infrastructure and limitation of autonomy [Lee et al., 2017].

From a practical point of view, we can interpret landing targets as fiducial markers. However, fiducial markers are designed to serve as measurement or reference points into a scene. We can find in the literature different proposals for the automatic generation of fiducial markers and their respective detection algorithms [Fiala, 2010], [Naimark and Foxlin, 2002]. However, detection algorithms can fail for several reasons, such as poor lighting conditions, rapid camera movements, and occlusions. A common approach to improve the robustness of a marker detection system is the use of marker boards, i.e., a pattern composed of multiple markers [Garrido-Jurado et al., 2014].

We chose to design and detect our own landing targets using the image contours as input information and build a feature space. We then mathematically interpret the human perception concepts (Gestalt theory and Helmholtz principle) to find the landing targets as deviations of a random model. With these elements, we detect the

targets in an unsupervised way. We describe below such concepts and the different proven approaches to contour extraction and target identification.

1.1.2 Visual Perception Concepts

Humans can carry out the process of perception in a natural way [Petitot, 2008]. We, as humans, identify meaningful features and exciting events in a scene (such as points, lines, edges, textures, colors, movement), and with the help of our memory and the learning capacity, we can recognize and classify objects. The identification of primitives is a consequence of their non-accidental apparition, i.e., they are not generated randomly [Attneave, 1954]. This behavior is roughly the Helmholtz principle, which states that we do not perceive any structure in a uniform random image. However, whenever some deviation from randomness occurs, it is possible to find a structure. In other words, events that could not happen by chance are immediately perceived. This principle is represented in the Fig. 1.2.

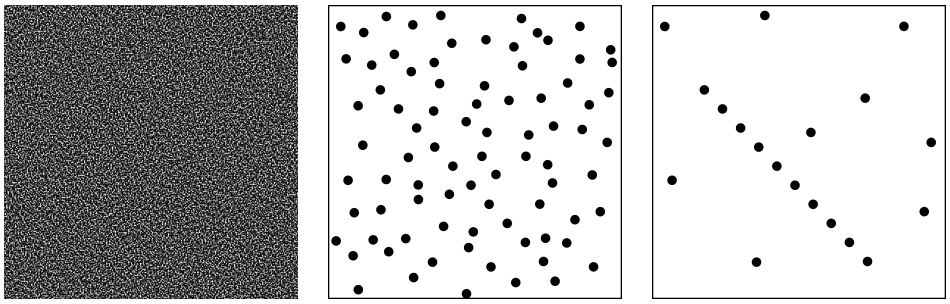
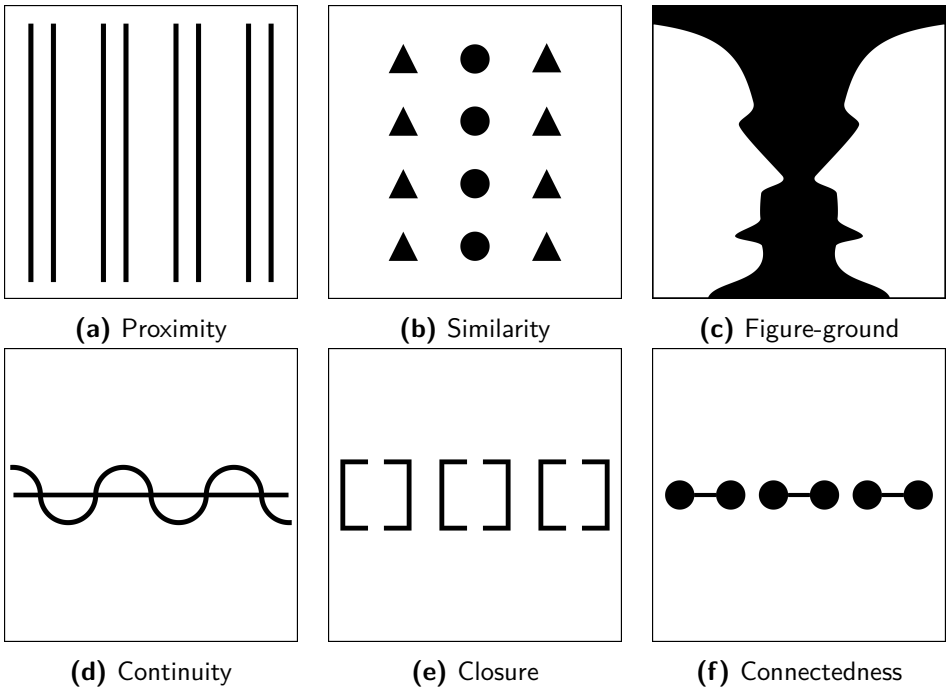


Figure 1.2: Representation of Helmholtz's principle: **(a)** Uniform random image where no structure can be found. A group of ten aligned dots exists in both images **(b)**and **(c)**, but this structure can hardly be seen in the central image. Otherwise, in the right-most image, the alignment stands out as a significant deviation from the randomness that cannot happen by chance and is therefore perceived.

The Gestalt theory [Wertheimer, 1923] states that we can build a whole (a gestalt) through the grouping of non-accidental detected primitives. That means that the human mind recognizes objects as a whole before examining their individual parts, and the observer perceives the information that is not related in size, shape, or orientation as chaotic and disorganized. The grouping of individual elements in a whole follows a set of laws defined by the Gestalt theory; some of them are (see Fig. 1.3):

- Similarity law
- Proximity law
- Continuity law
- Closure law

- Connectedness law
- Figure-ground law



(a) Proximity **(b)** Similarity **(c)** Figure-ground

(d) Continuity **(e)** Closure **(f)** Connectedness

Figure 1.3: Graphic representation of the grouping Gestalt laws.

In this work, we explore the above ideas and propose a novel approach to detect a landing target in the same way humans do, imitating the human perception process. To achieve the detection, we use the intensity image contours retrieved at different scales. We obtain the most perceptual contours from this set of contours: those not generated by chance using an a contrario approach. After this procedure, we take advantage of the predefined form of the targets to propose some measures representing the grouping laws of similarity and proximity of the Gestalt. Finally, we do the decoding and correction of target identification errors using Hamming's code. The diagram shown in Fig. 1.4 groups the stages of our method for the perceptual detection of landing targets.

The following sections are devoted to detailing the framework for the detection of landing targets. In section 1.2, we evaluate different threshold-based methods for obtaining contours in an algorithm that uses the hierarchy of contours to detect landing targets. After that, we develop our perception model in section 1.3. Specifically, subsection 1.3.1 describes how to retrieve image contours as meaningful primitives, and subsection 1.3.2 describes how to group the contours to detect a landing target perceptually. The section 1.4 contains the description of the landing target and the strategy used for the generation and coding of information into the landing targets. Later, in section 1.5, we present the implementation of our methodology and some tests with both synthetic and real-life images. We also present some conclusions and perspectives in section 1.6. Finally,

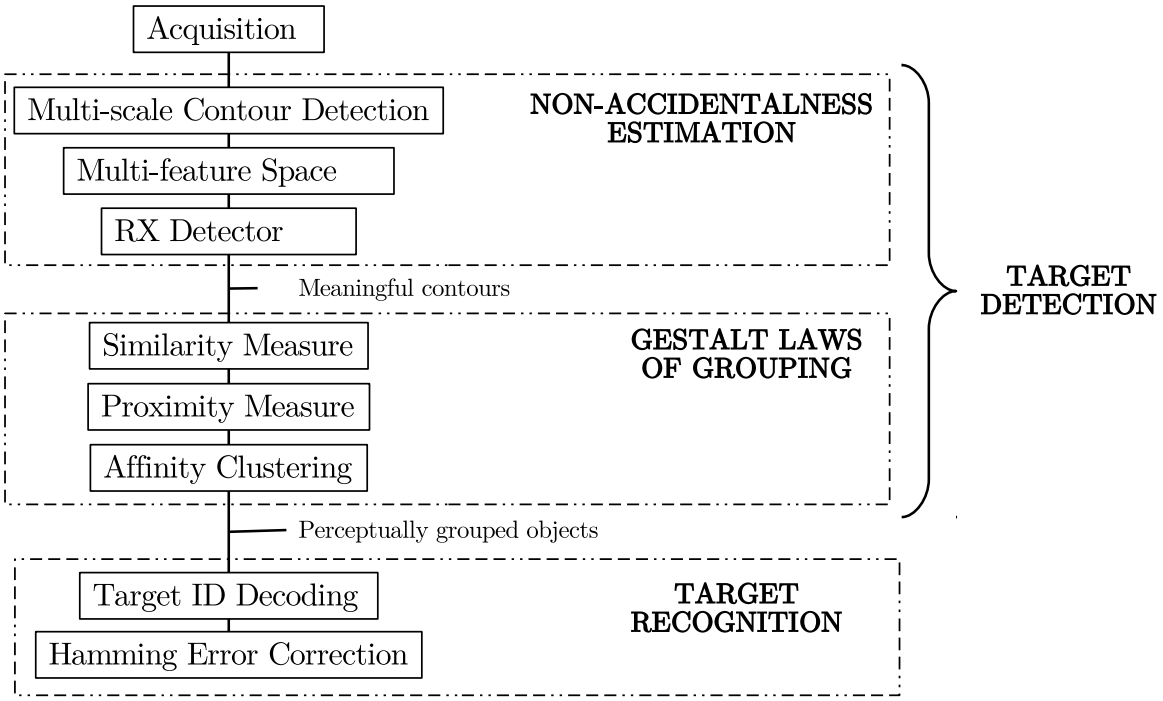


Figure 1.4: Diagram of the phases involved in the landing target detection and recognition task.

1.2 Hierarchical Countours for Target Detection

Initially, the idea of landing targets detection is inspired by the needs of the Internest company. The objective is to design a landing marker and an algorithm for its detection; all of this in the context of the UAV's autonomous precision landing task. A first approach, developed during the traineeship period of a master student [Baquedano, A., 2017], seeks to solve the task straightforwardly using highly studied techniques. The algorithm is based on finding the contours of a binary image generated by the threshold method proposed by Otsu [1979]. Since the landing target they suggest is composed of nested concentric circles, they heuristically use the hierarchy of the found contours to detect a landing target. Their methodology consists of discriminating the contours that are not nested through conditional evaluations at each hierarchy level. The conditions are hierarchically dependent, which means that the landing target detection is ineffective if the conditions are not strictly fulfilled.

They tested this approach; however, similarly to some other works mentioned in section 1.1, the algorithm works well only under certain circumstances. The tests show that the algorithm fails in most cases because it is not able to find all the target contours, which compromises the hierarchical condition for detection. This effect generally occurs when the landing target is exposed to conditions that degrade the quality of the image. Given the nature of the aerial object detection task, factors such as the change in height and orientation of the UAV modify an object's perception, introducing

disturbances such as noise, changes in lighting and contrast, deformation of objects, etc. Such image degradations complicate the operation of threshold-based methods and consequently the detection of contours. We classify the disturbances suffered by landing targets into four types:

1. Change in size w.r.t. the scene.
2. Presence of noise.
3. Presence of shadows.
4. Deformation due to perspective.

Fig. 1.5 shows a landing target affected by the disturbances mentioned above.

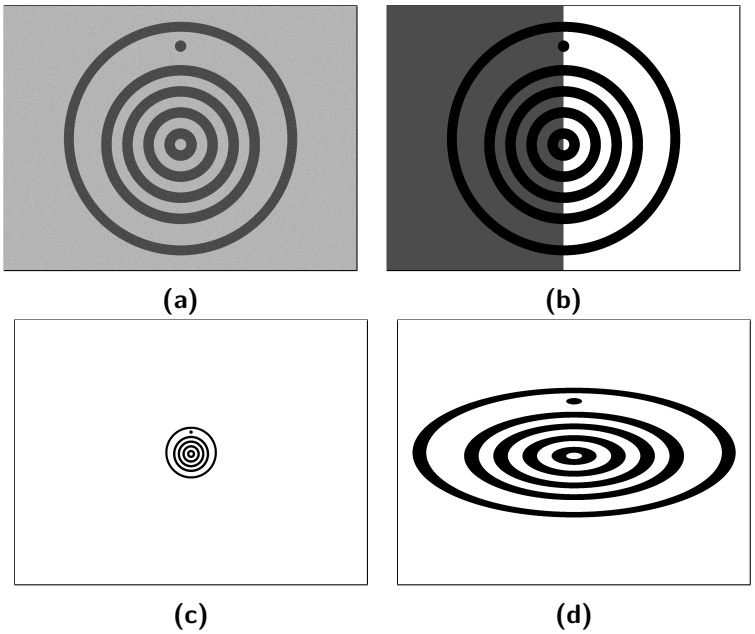


Figure 1.5: Landing target degradations: **(a)** Noise, **(b)** Shadow, **(c)** Change of size and **(d)** Perspectice deformation.

One of the main disadvantages of the hierarchy method for the landing targets detection is that its effectiveness lies with the contours detected on the binary image generated by Otsu threshold method, which does not work well in images with high contrast or severe lighting changes. However, other threshold-based methods for contour detection could better face the image degradations shown in the Fig. 1.5. Following the taxonomy for threshold-based methods proposed in [Sezgin and Sankur, 2010], there are clustering-based methods such as Otsu [1979] and Ridler and Calvard [1978] edge detectors; entropy-based methods such as Yen et al. [1995] and Li and Lee [1993] detectors; local methods such as Niblack [1986] and Sauvola and Pietikäinen [2000] operators; the adaptive method proposed by Bradley and Roth [2007] and finally, the mean and Gaussian pixel distribution as spacial methods. We use these nine

representative threshold-based methods to obtain a binary image, localize the image contours, and evaluate the best approach facing the image degradations shown in Fig. 1.5.

The contours obtained applying each threshold-based method are depicted in Fig. 1.6. In this figure, we can see that the result is better or worse depending on the situation. For example, we can see that the Otsu, Riddler, Yen and Li methods react appropriately to the target scale change, but none of these four methods except the Yen operator work correctly in the presence of shadows. This results from comparing four methods under two disturbances; however, out of the nine methods, none works correctly for all perturbations.

1.2.1 Threshold-based Method's Evaluation

We run the hierarchical algorithm proposed in [Baquedano, A., 2017] on a database of synthetic images. The database contains sixteen different landing targets perturbed by the image degradations of Fig. 1.5. The noise degradation is simulated by adding Gaussian noise with a mean of zero and a standard deviation variable from 0.02 to 0.2, where 0.02 is the minimum noise addition. The shadow perturbation is simulated shading the left-half of the image; the variation of the shadow is done between 0 and 1, where 0 indicates a darker left-half image. The last two degradations are related to the perspective and distance of the viewer (the camera). First, we change the size of the landing target by scaling forming circles on a 640×480p image. The range of the scale is from 0 to 1, where 1 indicates the real scale. Lastly, we consider that the circular target behaves like an ellipse when it is not seen from the center's perpendicular axis to achieve the perspective degradation. Therefore, we deform the target synthetically by augmenting the proportion of one axis (major and minor) of the ellipse in an interval between 1 and 2, where 2 indicates the maximum deformation.

For the test of the different contour detectors, we apply the maximum value of each degradation. We use the F1-score as a metric to evaluate each threshold-based method's accuracy under the various degradations.

$$\text{F1-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{1.1}$$

This metric can be interpreted as a weighted harmonic mean of precision and recall. The precision is the ratio $tp/(tp+fp)$ and the recall is the ratio $tp/(tp+fn)$, where tp is the number of true positives, fp the number of false positives, and fn is the number of false negatives. The F1-score reaches its best value at 1 and its worst score at 0. Fig. 1.7 shows the performance of all nine detectors on each disturbance separately in the form of bar plots. The graphs show the F1-score of the hierarchical detection algorithm without the Hamming error correction (gray bars) and with the use of the

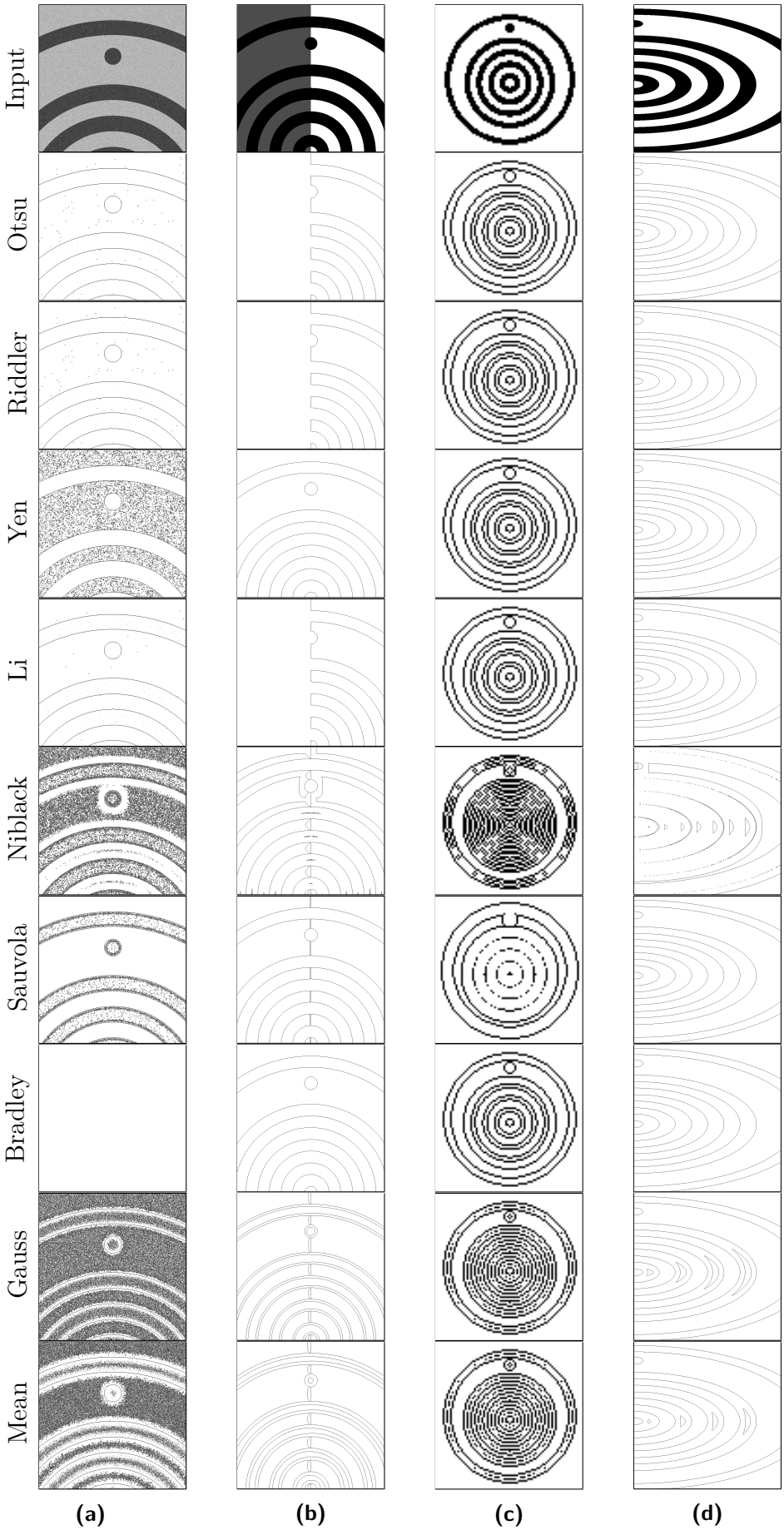


Figure 1.6: Contours obtained with the threshold-based methods applied on synthetic undergoing various degradations likely to occur in real-life conditions: **(a)** Presence of noise, **(b)** Presence of shadows, **(c)** Change of scale, and **(d)** Deformation degradations.

Hamming error correction (black bars) described in section 1.4.

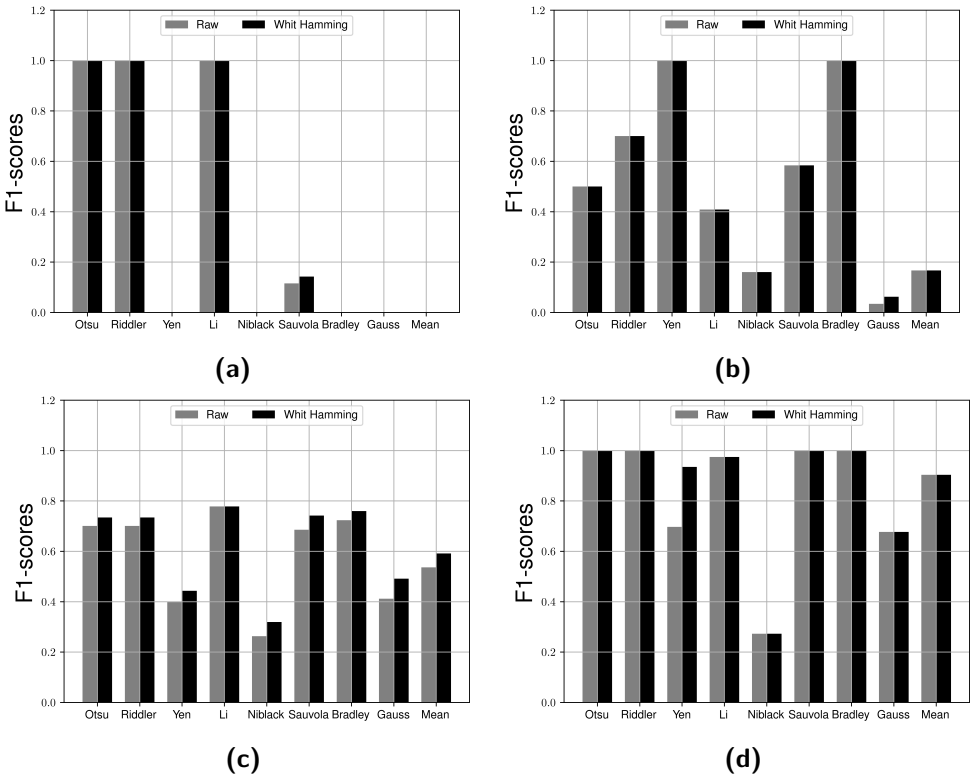
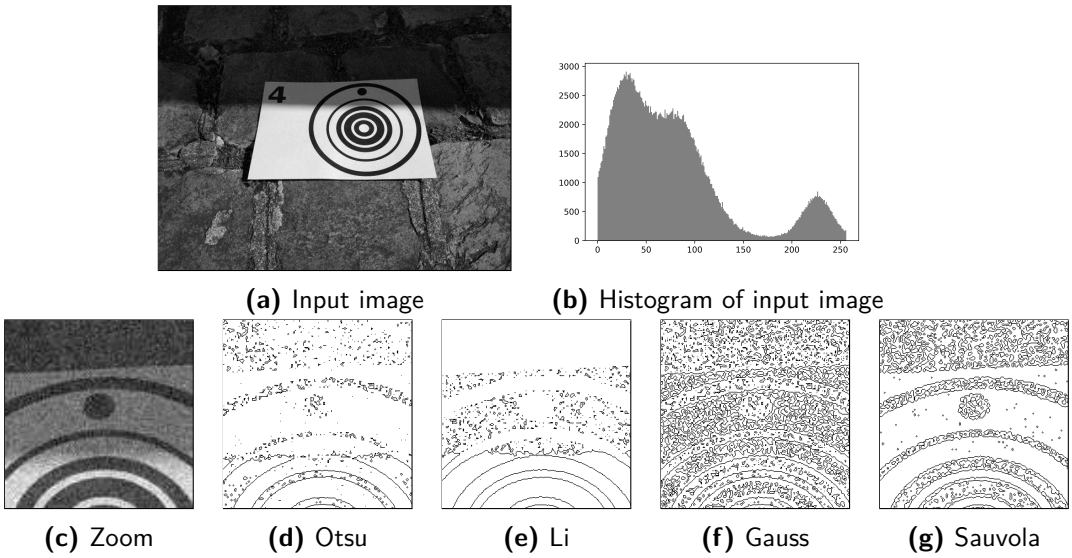


Figure 1.7: F1-score bar graphs: **(a)** Noise, **(b)** Shadow, **(c)** Change of size and **(d)** Perspectice deformation

Although the experiments use the highest target deformation values, they do not consider combining two or more degradations, which is closer to reality. Fig. 1.8a shows one of our targets (target ID 4) under real lighting conditions, i.e., in an outdoor environment where the four degradations of the experiment are present. We also show its intensity histogram to highlight the saturation levels of the scene and the contours obtained with a representative method of each class of the taxonomy in [Sezgin and Sankur, 2010]: clustering-based (Fig. 1.8d), entropy-based (Fig. 1.8e), spacial (Fig. 1.8f) and local (Fig. 1.8g) threshold-based methods.

The F1-score bar plots (Fig. 1.7) and Fig. 1.8 show that given the conditions in which we can find a landing target, no threshold-based method was robust to the set of perturbations. It is necessary to adjust parameters according to the condition to have acceptable results. Besides, we aim to recognize the landing targets in natural images where none, one, or more landing targets can be present, and the degradations are not isolated.



(a) Input image **(b)** Histogram of input image

(c) Zoom **(d)** Otsu **(e)** Li **(f)** Gauss **(g)** Sauvola

Figure 1.8: Landing target under non-controlled illumination conditions and the contours obtained with some threshold-based methods.

1.3 Unsupervised Perception Model for UAV Autonomous Landing Task

1.3.1 Non-accidentalness Estimation

Contour Detection

After developing the first algorithm by [Baquedano, A., 2017], we take some elements from their work to develop a more general approach that explores human perception principles. Precisely, we keep the concept of concentric circle patterns for the generation of landing targets (see section 1.4 or the description of the landing targets generation) and the use of image contours as input data.

Instead of using a threshold-based method, we obtain the image contours without fixing any parameter using the Marr and Hildreth [1980] operator. The Marr-Hildreth operator guarantees to obtain continuous and closed contours eliminating the possible noise in the image, while the contours of objects remain unchanged in the presence of shadows. This technique convolves the intensity image f with the 2-d Laplacian of Gaussian (LoG) operator $\nabla^2 G(x, y, \sigma)$ and generates an image l_σ,

$$l_\sigma = \nabla^2 (G(\sigma) * f) \tag{1.2}$$

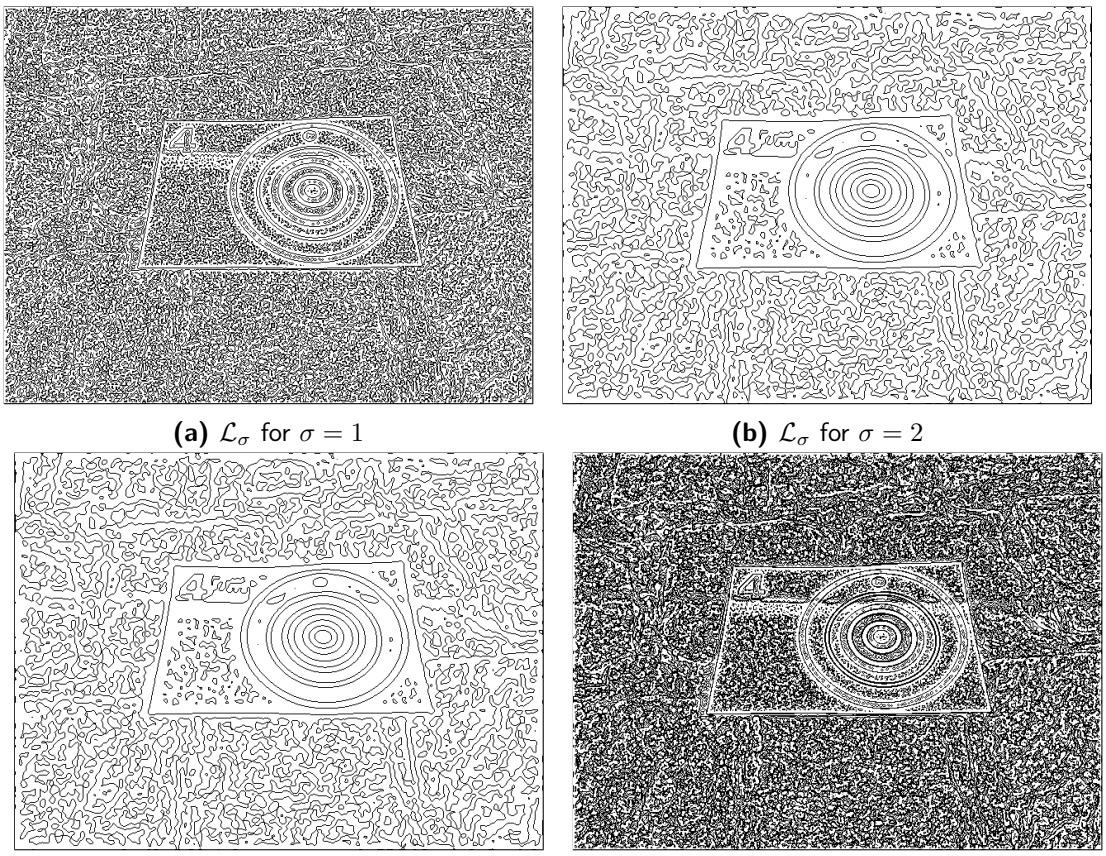
in which we localize the zero-crossings. Such zero-crossings define the contours of the image.

The parameter σ in Eq. (1.2) permits to control the amount of image smoothing and acts as a scale parameter, which generates different scale-space images when varied.

Since it does not exist optimal single filter simultaneously at all scales [Marr and Hildreth, 1980], we use a multi-scale analysis [Witkin, 1984] to detect the zero-crossings in l_σ at different scales to minimize the risk that some contour of interest is not detected. The image l_σ from Eq. (1.2) contains a set of contours $\mathcal{L}_\sigma = \{L_i^\sigma,\ \ i = 0, 1, \ldots, N\}$ for a given scale σ. Then,

$$\mathcal{L} = \bigcup_{\sigma} L_\sigma \tag{1.3}$$

represents all the contours of an image obtained at different scales. Fig. 1.9d shows the set of contours $\mathcal{L}$ found with $\sigma = [1, 2, 3]$. We can also notice that the objects' characteristics are more visible at a fine scale (see Fig. 1.9a), i.e., there are more contours. Conversely, there is a spatial distortion at coarse scales due to the smoothing, and therefore fewer contours appear (see Fig. 1.9c). However, those contours that had already appeared at a coarse-scale will not disappear. Then, there is the probability that those contours that spatially coincide on two or more scales belong to a change of intensity generated by the border of an object.



(a) $\mathcal{L}_\sigma$ for $\sigma = 1$ **(b)** $\mathcal{L}_\sigma$ for $\sigma = 2$

(c) $\mathcal{L}_\sigma$ for $\sigma = 3$ **(d)** Set $\mathcal{L}$ for $\sigma = [1, 2, 3]$

Figure 1.9: Image contours found at three different scales: **(a)** $\sigma = 1$, **(b)** $\sigma = 2$, **(c)** $\sigma = 3$ and, **(d)** joined in the set $\mathcal{L}$.

Multi-feature Space

The Helmholtz principle states that meaningful characteristics appear as large deviations from randomness, and that is how the human perception automatically works to identify an object [Attneave, 1954]. The a contrario model proposed in [Desolneux et al., 2008], formulates this principle statistically by setting the number of false alarms (NFA) below some acceptable level; however, this method cannot be easily extended to more complex shapes. Instead of setting the NFA, we use the RX detector [Reed and Yu, 1990] to detect outliers. The RX anomaly detector, initially called the Constant False Alarms Rate (CFAR) detection algorithm, can detect the presence of a known signal pattern in several signal-plus-noise channels. For that, it uses a $N \times Q$ multi-variable space $Z = [Z_1, \ldots, Z_Q]$ with Q observation vectors of dimension N. In our approach, the primitive is a closed contour. We build the multi-variable space with observations based on internal (geometrical features, e.g., circularity, roundness, area, perimeter) and external (e.g., mean gradient intensity, intensity inner area) properties of the contours.

Let $L_i \in \mathcal{L}$ be a closed contour, A_i the area of the region enclosed by the closed contour, and P_i its perimeter; we compute the circularity Eq. (1.4) and the mean gradient intensity Eq. (1.5) to build the multi-variable space $Z = [Z_1, Z_2]$, where

$$Z_1 = \left[\frac{4\pi A_i}{P_i^2}, \;\; i = 0, \ldots, N\right]^T, \;\; N = \text{card}(\mathcal{L}), \tag{1.4}$$

$$Z_2 = \left[\frac{1}{P_i}\sum_{x \in L_i} \mid \nabla f(x) \mid, \;\; L_i \in \mathcal{L}\right]^T. \tag{1.5}$$

RX Detector

The RX anomaly detector [Reed and Yu, 1990] is commonly used to detect outliers on such data. The space Z models the set of contours $\mathcal{L}$ with $Q = 2$ feature vectors describing the circularity Eq. (1.4) and the mean gradient intensity Eq. (1.5). The RX detector gives an anomaly score to each contour taking into account the mean of the distribution and covariance between the Q-features through the Mahalanobis distance:

$$y_i = (z_i - \mu_Z)^T \Sigma_Z^{-1} (z_i - \mu_Z), \tag{1.6}$$

where $\mu_Z = [\text{E}[z_1], \ldots, \text{E}[z_N]]^T$ is a vector of observation means and Σ_Z^{-1} the $N \times Q$ covariance matrix of the data. If the data have normal random distribution, then the score vector $Y = [y_i, \ldots, y_N]$ follows a chi-square distribution $\chi^2_Q(\varphi)$ with Q degrees of freedom, where φ is a confidence level [Lu et al., 2004]. The value of $\chi^2_Q(\varphi)$ with a confidence value $\varphi = 99.9\%$ operates as a threshold to identify all contours that behave as outliers in the multi-variable distribution. In our case, the contours belonging to a

landing target appear as outliers in the vast majority of random contours belonging to the background.

With the previous strategy, we preserve the anomalous contours with a mean gradient and circularity value deviating from the distribution's principal mode in the contours set $\widetilde{\mathcal{L}} = \{L_i \mid y_i > \chi^2_Q(\varphi)\}$. $\chi^2_Q(\varphi)$ is the value of the cumulative distribution at the confidence level φ and $\widetilde{\mathcal{L}} \subset \mathcal{L}$. At this point, it is essential to mention the importance of multi-scale contour detection of section 1.3.1; because it increases the number of samples in Z, allowing to build a richer multi-variable space.

Not all the contours of the set $\widetilde{\mathcal{L}}$ are part of the contours of the landing target. For example, in the Fig. 1.10, we can see that the paper sheet contours remain because they have a high circularity value. The same occurs with contours of objects with an important value of mean gradient (brightness step), as the number 4 (which indicates the ID of our target) at the top-left of the sheet, or the pebbles, sand, gravel textures of the background.

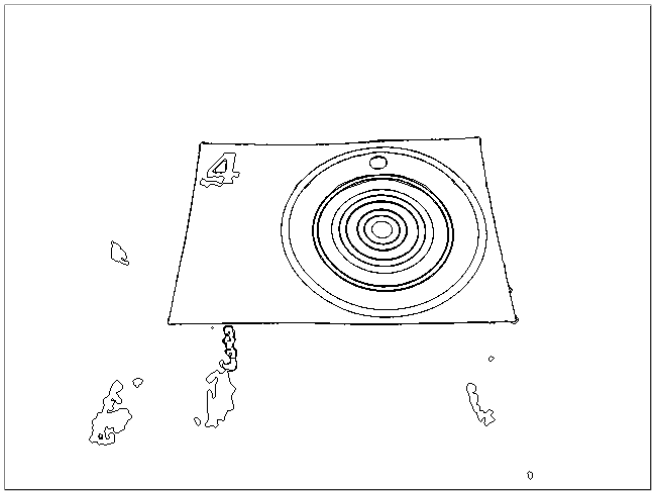


Figure 1.10: The contours from Fig. 1.9d that behave as outliers in the multi-feature space Z with a confidence value of $\varphi = 99.9\%$.

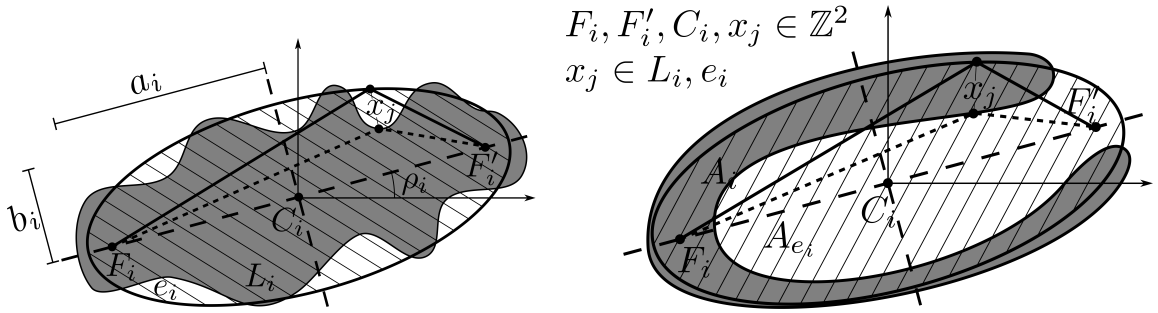
1.3.2 Gestalt Laws of Grouping

We use the Gestalt theory [Wertheimer, 1923] to group the meaningful contours $L_i \in \widetilde{\mathcal{L}}$ and detect landing targets. We primarily use two grouping laws: similarity (represented by the goodness of shape) and proximity (represented by the affinity clustering), which we will detail below.

Goodness of Shape

Since the landing targets have only circular contours, we evaluate the resemblance with an ellipse of all contours. This strategy allows us to deal with the perspective deformation of landing targets. Considering an ellipse e_i that fits one gray contour L_i

in Fig. 1.11a, we recover the centroid C_i, the rotational angle ρ_i, the semi-major axis a_i, the semi-minor axis b_i and the coordinates F_i and F_i' of the ellipse's foci. Then, the sum of the distances from any point of ellipse $x_j \in e_i$ to the foci is $\overline{x_jF_i} + \overline{x_jF_i'} = 2a_i$. If the contour L_i is an ellipse, the value $d_i = \left|(\overline{x_jF_i} + \overline{x_jF_i'}) - 2\alpha_i\right|$ must be zero or negligible $\forall x_j \in L_i$.



(a) Affinity of a fit ω_i **(b)** Difference of area Δ_{A_i}

Figure 1.11: Visual description of affinity of ellipse and difference of area.

Considering the geometrical form of the landing target, we estimate the similarity to an ellipse using two measures:

$$\omega_i = \mathrm{e}^{-\frac{d_i^2}{2\sigma^2}}, \tag{1.7}$$

$$\Delta_{A_i} = 1 - \frac{\left|A_{e_i} - A_i\right|}{\max(A_{e_i}, A_i)}. \tag{1.8}$$

The variable $\omega_i \to 1$ defines how well an eclipse fits into the contour. The affinity of the fit $\omega_i \to 1$ for contours with an ellipsoidal shape; however, if the contour L_i has a croissant shape (as in Fig. 1.11b) then, the Eq. (1.7) also has a high value (near to 1), but the contour is far from being an ellipse. The difference of areas Eq. (1.8) complements the affinity ω_i taking into account the area of the ellipse A_{e_i} and the area of the region enclosed by the closed contour A_i. To calculate the similarity to an ellipse, we use the harmonic mean of both variables:

$$\kappa_i = \mathcal{H}(\omega_i, \Delta_{A_i}), \;\; \kappa_i \in (0, 1), \tag{1.9}$$

where $\kappa_i \to 1$ for contours resembling to an ellipse and $\kappa_i \to 0$ otherwise. $\mathcal{H}$ denotes the harmonic mean $\mathcal{H} = N\left(\sum_{i=1}^{N} \xi_i^{-1}\right)^{-1}$, where $\xi_i \forall i = 1, \cdots, N$ are the variables on which we compute the harmonic mean.

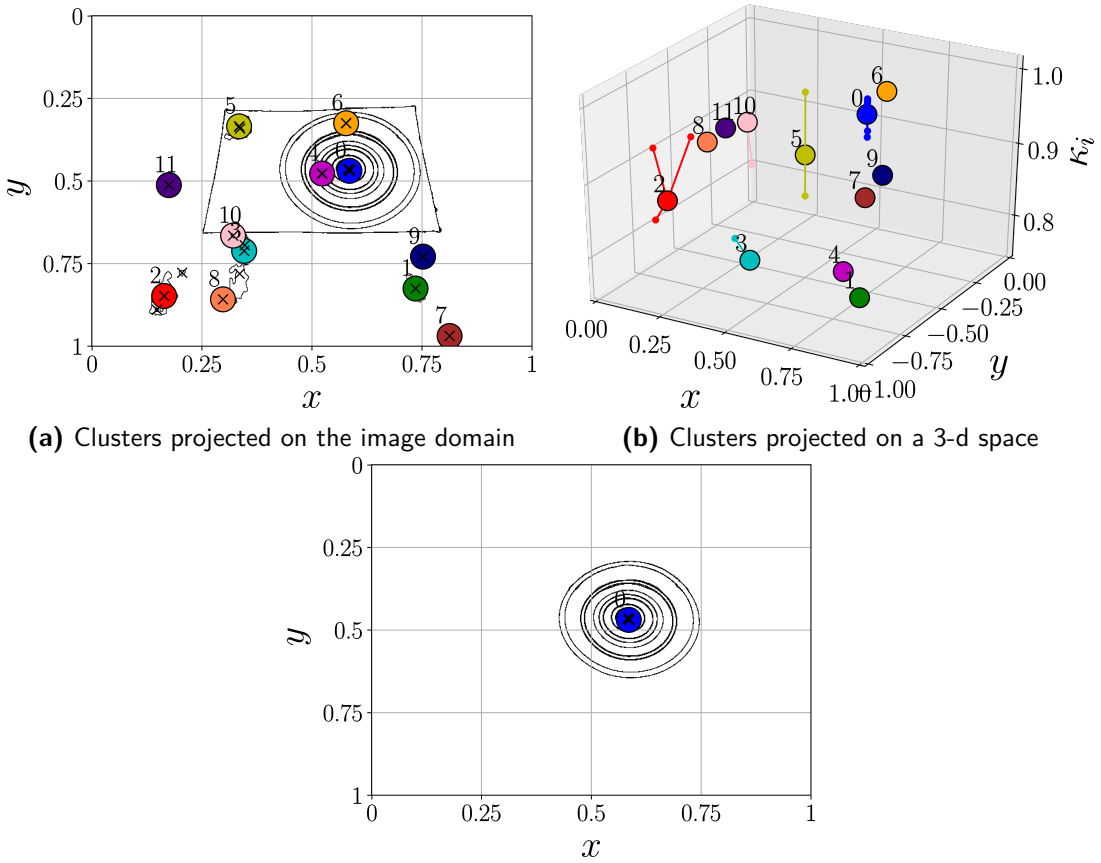
Proximity Measure

The Gestalt law of proximity states that we group those meaningful elements if they are spatially close to each other. In the case of contours, we take the coordinates of

their centers C_i to measure their spatial proximity.

Affinity Clustering

The normalized coordinates of the centroid C_i and the ellipse similarity κ_i map the contour $L_i \in \widetilde{\mathcal{L}}$ into the 3-d space $(x, y, \kappa) \in \mathbb{R}^3$. We use the affinity propagation clustering method [Frey and Dueck, 2017] to group the contours using the feature matrix $X = [C_i, \kappa_i]$ of size $N \times P$, where $P = 2$ describes the two features we use for grouping: contour centroids (C_i) and the affinity with an ellipse (κ_i). This technique yields a set of $K \in \mathcal{C}(X)$ clusters, where the operator $\mathcal{C}(X)$ defines the affinity propagation technique over the feature matrix X. Because the landing target has ten different contours (see section 1.4), the clusters with $|K| \geq 10$ elements and an important similarity value $\mathcal{H}(\kappa_i) \geq 0.8$, represent the candidate contours of a landing target.



(a) Clusters projected on the image domain

(b) Clusters projected on a 3-d space

(c) Target candidate cluster

Figure 1.12: Clusters obtained by affinity propagation of contour from Fig. 1.10.

To illustrate the use of affinity clustering, let us take as an example the contours resulting from the RX anomaly detector of Fig. 1.10. The affinity propagation technique groups the contours into $K = 12$ clusters. Projecting the clusters in a 2-d plane (Fig. 1.12a), we note that there are clusters relatively close to each other in the image domain, for example, clusters 3 and 10; however, the respective contours of these clusters are

differentiated and separated by the algorithm. This separation is due mainly to the influence of κ_i in the clustering process. We can better notice the influence of ellipse similarity on a 3-d plot, where the z-axis represents this measure (see Fig. 1.12b). We notice that even if the contours are nearby, it can form a new cluster if there is a considerable distance κ. A clear example is the clusters 0 and 4 (blue and purple, respectively) that correspond to the contour centers of the landing target and the center of the paper sheet; they are spatially close to each other, but their similarity is not. Applying the threshold values $\text{card}(\mathcal{C}_K) \geq 10$ and $\mathcal{H}(\kappa_i) \geq 0.8$, we obtain the candidate clusters to form a landing target (see Fig. 1.12c).

Heretofore, we have built a model based on perceptual characteristics for the landing target detection. However, there could be false detections if there are round objects with concentric borders in the image. We implement a relevant functionality that, on the one hand, is a stage that suppresses false detections and, on the other, identifies unique targets based on a unique ID. We code an ID number in the target design to differentiate a landing target from an object with concentric circular edges. In the following section, we review in detail the coding of landing targets and the generated landing target database

1.4 Landing Target Description

This section describes the landing target design process used in chapter 1 for detection and identification in the autonomous landing task. The landing target is formed by a set of black and white circles (see figure 1.13) that generate contours when stacked. Two of the circles ($\varnothing_9$ and $\varnothing_{10}$) have a constant diameter and form the ring that defines the target. The black circle ($\varnothing_{11}$) is an orientation reference and has the same diameter as the smallest circle, $\varnothing_{11} = \varnothing_1$. The other circles $\varnothing_1, \ldots, \varnothing_8$ are coding circles.

1.4.1 Landing Target ID Encoding

Let $\varnothing = (\varnothing_1, \varnothing_2, \ldots, \varnothing_n)$ denote the nominal diameters of the coding circles. We can set the nominal diameters, e.g., $\varnothing_i = \frac{i}{n}\varnothing_n$ for a target without the encoding capability. To encode a number in the target form, we modify the nominal diameters $\varnothing$ to obtain $\varnothing' = (\varnothing'_1, \varnothing'_2, \ldots, \varnothing'_n)$ by adding/subtracting a positive constant Δh

$$\varnothing'_i = \begin{cases} \varnothing_i + \Delta h, & \text{if } w_i = 1 \\ \varnothing_i - \Delta h, & \text{otherwise} \end{cases} \tag{1.10}$$

and obtain a binary message $W = [w_1, \ldots, w_n]$. The message W is protected from errors by error-correction Hamming code [Hamming, 1950]. It provides a set of different codewords $W = D \times M$ of size $n = k + m$, where D is useful data, $M = [I_k \mid 1 - I_k]$

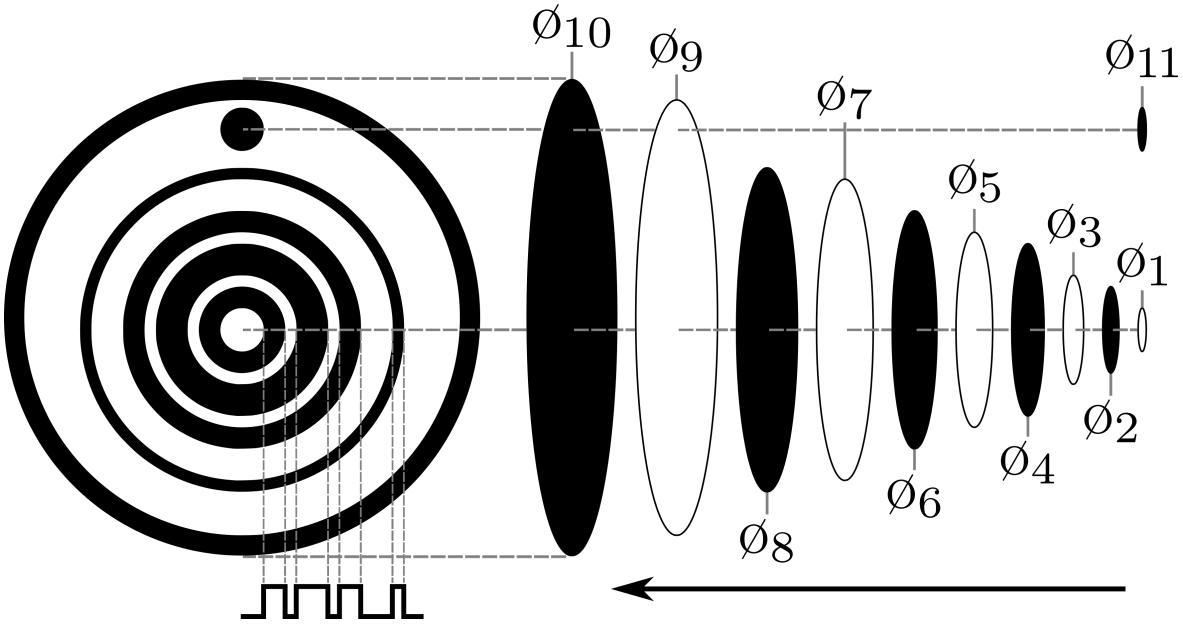


Figure 1.13: Landing target design and description

the generator matrix, and I_k is the $k \times k$ the identity matrix. The data vector D comes from the decimal to binary conversion of the landing target ID number. For the experiments showed in chapter 1, we create landing targets with $n = 8$ coding circles allowing to have four rings and 8 contours $\varnothing_1, \ldots, \varnothing_8$. This design allows us to use the extended $[n, k]$ Hamming code with $k = 4$ data bits and $m = 4$ parity bits to generate $2^4 = 16$ landing targets. We can see the set of landing targets generated using this configuration in figure 1.14.

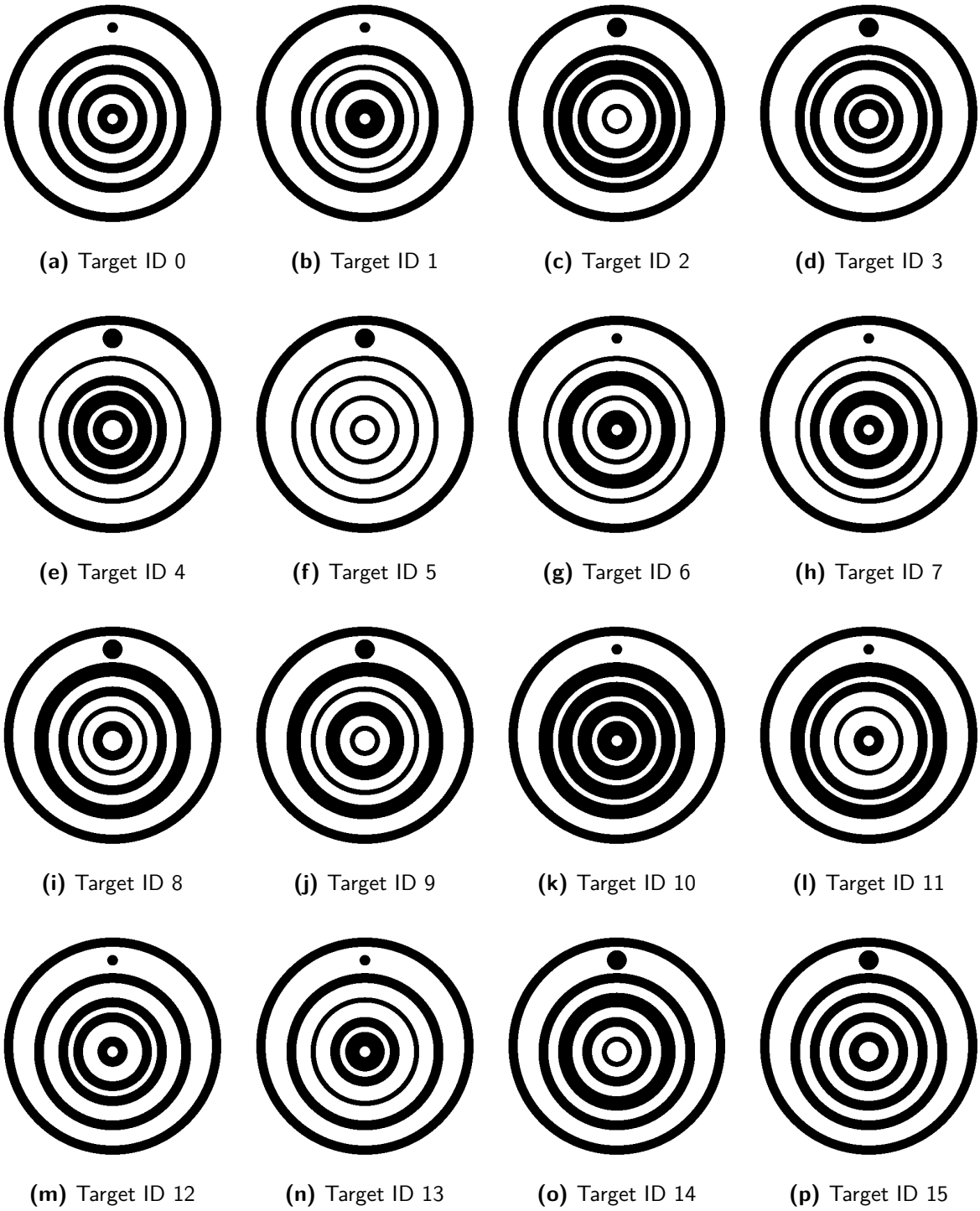
1.4.2 Landing Target ID Decoding

After the clustering stage of section 1.3.2, we rank by size the ellipses' major axes α_i by size and normalize them w.r.t. the largest value α_{10} to obtain $\vec{\alpha} = \frac{\varnothing_{10}}{\alpha_{10}}(\alpha_1, \ldots, \alpha_{10})$

We compare the received and normalized axis $\vec{\alpha}$ with the nominal diameters of the coding circles $\varnothing$ and transform them into a binary vector $\widehat{W}$

$$\widehat{W} = \begin{cases} 1, & \text{if } \alpha_i - \varnothing_i > 0 \\ 0, & \text{otherwise} \end{cases} \quad \forall i = 1, \ldots, n \tag{1.11}$$

The Hamming syndrome vector $S = \widehat{W} \times H^T$ (with $H = [1 - I_k \mid I_k]$ as the parity-check matrix) indicates whether an error has occurred. The syndrome is a null vector $S = 0$ when no error has occurred, otherwise, $S \neq 0$ and $\widehat{W} = W + E$. The element $e_i = 0$ of the error vector $E = H^T - S$ indicates an error at the position i. The $[8, 4]$ Hamming code can find up to two erroneous bits and correct one. Once the algorithm corrects the error (if there is), the vector $\widehat{W}$ is decoded by using the modulo 2 of the product $\widehat{D} = \widehat{W} \times M^T$.



(a) Target ID 0 **(b)** Target ID 1 **(c)** Target ID 2 **(d)** Target ID 3

(e) Target ID 4 **(f)** Target ID 5 **(g)** Target ID 6 **(h)** Target ID 7

(i) Target ID 8 **(j)** Target ID 9 **(k)** Target ID 10 **(l)** Target ID 11

(m) Target ID 12 **(n)** Target ID 13 **(o)** Target ID 14 **(p)** Target ID 15

Figure 1.14: Landing target database generated with error-correction Hamming code.

The coding of information allows discriminating between several landing targets and circular objects. The following section shows some tests we carry out to validate our model and target landing detection and identification results.

1.5 Model Validation and Test

We validate the methodology presented in this chapter on landing target images under simulated and real situations. First, we tested the algorithm in a synthetic image database which simulates the four image degradations reviewed in this chapter: noise, shadows, target deformation, and change of size. Second, for real situations, we perform a series of tests in indoor and outdoor scenarios. Fig. 1.15 collects some of the target detection and identification results, together with the output images of each of our target detection algorithm stages.

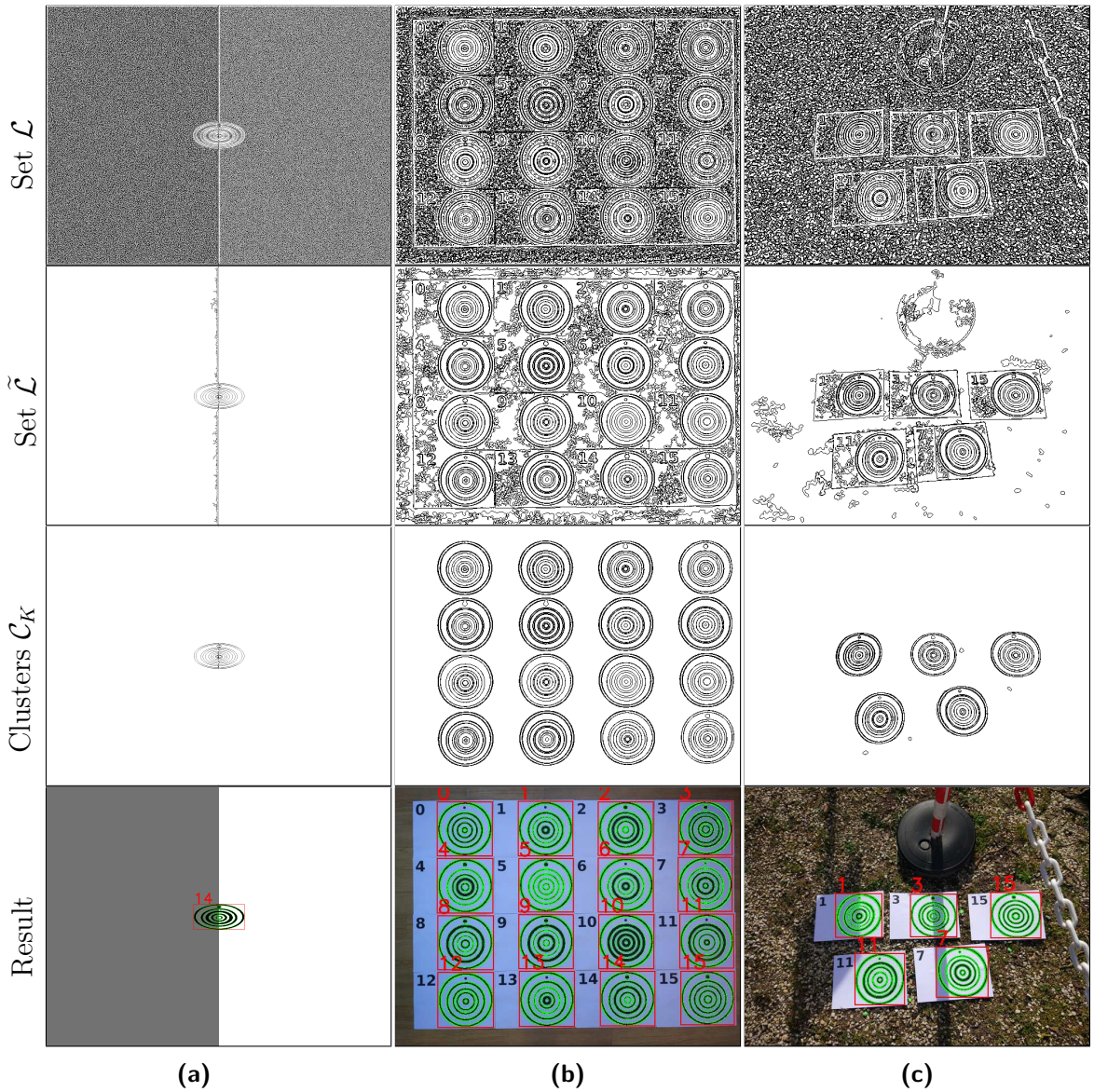


Figure 1.15: Algorithm validation: **(a)** Synthetic image target under simulated degradations, **(b)** The 16 targets in an indoor environment, **(c)** Five of our targets in an outdoor scenario under non-controlled image degradations

The first experiment (Fig. 1.15a) consists of combining the four synthetic degradations simultaneously on landing target ID 14. For the experiment, we subjected the

target to the maximum degradation value that the algorithm can handle. The second experiment (Fig. 1.15b) was done in an indoor space to show the sixteen possible landing targets. There are no other objects in the scene; however, the lighting level is low and constant concerning the outside environment. Finally, the last experiment (Fig. 1.15c) shows five landing targets in a more complex outdoor environment. Notice the presence of other objects, different background textures, irregular shadows, and the landing targets' perspective deformation and scale change.

Fig. 1.15 shows the results of each stage or our algorithm for the three experiments described above:

1. Multi-scale analysis generates a rich family of contours (first row of Fig. 1.15).

2. The non-accidentalness estimation stage eliminates the contours generated by noise with low circularity and mean gradient values (second row of Fig. 1.15).

3. The grouping stage filters random contours generated by intensity changes like shadows to keep contours with an important similarity and proximity value (third row of Fig. 1.15).

We invite the reader to see the compilation of the experiments performed under real conditions in https://youtu.be/igsQc7VEF2c.

1.6 Conclusion

In this chapter, we have shown the usefulness of a low-level primitive such as intensity. First, we have shown a methodology that straightforwardly uses the contour's hierarchy to represent and detect objects. Then, in section 1.3, we have applied concepts of human perception, such as Gestalt laws and Helmholtz's principle, to develop a perceptual framework for object detection. This model uses the geometric properties of the contours obtained at multiple scales to generate a representation of the image. This approach allows us to obtain a scene representation, avoiding the loss of information due to the objects' size change or the presence of shadows and noise; common degradations in robot navigation. We validate our model in a specific drone application; the detection and recognition of a landing target. We have used the geometry properties of the target to build a perceptual object and the Hamming error codes to perform the landing target recognition. The experiments show that the proposed methodology is robust to uncontrolled light conditions and other image degradations existing in complex environments.

With this approach, we have provided a solution to the problem of object detection. This methodology is entirely unsupervised, free of fine-tuning of parameters or a priori knowledge of the environment.

It is important to note that, so far, we have only considered the image intensity as a low-level primitive for model generation. Considering the results obtained in this chapter, the natural question that arises is what other information we can use to enhance the image representation. In the following chapters, we explore the low-level color and texture primitives to obtain a better (and complete) representation of the image objects. We study this information in a more theoretical way, looking for possible relationships with elements of human perception.

On the other hand, we would like to mention that, in the following chapters, the validation of our approaches in specific drone applications is less present. One of the reasons is the lack of accessible databases rich enough for this kind of application. Consequently, we decided to validate our methods on well-known, open-access databases in the image processing field, which allows us to compare our results with state-of-the-art works.

Chapter 2

Global Representations of Color and Texture

Résumé

Ce chapitre présente une compilation des différentes manières de représenter les informations de couleur et de texture présentes dans une image. En ce qui concerne les informations de couleur, nous présentons certains des espaces colorimétriques les plus utilisés, leurs origines et leur relation avec la perception humaine. Par ailleurs, nous présentons quelques techniques pour synthétiser ces informations. Dans le cas de la texture, nous présentons les différentes méthodologies pour son étude, en mettant en évidence les avantages et les inconvénients de chaque méthode.

Abstract

This chapter presents a compilation of the different ways of representing the color and texture information present in an image. When it comes to color information, we present some of the most popular color spaces used and their origins and relationship to human perception. Besides, we present some techniques to synthesize this information. In the case of texture, we present different methodologies for its study, highlighting each method's advantages and disadvantages.

2.1 Introduction

In part 1, we show that low-level features, such as intensity image contours, provide useful perceptual information that can be used to solve complex problems. We pre-

sented a framework that uses human perception concepts and contours information for the unsupervised detection of landing targets. This framework can identify the marker under degraded operating conditions using only exogenous features from the contours identified on gray-level images. We can improve the framework by adding other features that provide perceptual information of a scene.

In this part of the thesis, we review two more low-level image features: color and texture. Both features are widely involved in the perceptual process of humans, and their study can be pervasive. This part explores the image color and texture features for their future integration into a general framework for object detection. For this purpose, the chapter that opens this part seeks to recall the definition of color and texture in the field of computer vision. Moreover, it reviews different approaches to color representation as well as different strategies for characterizing texture features. Then, we study the color features in two different frameworks.

First, we are interested in the global distribution of color and texture information of an image. Therefore, in chapter 3, we take an interest in comparing distributions, particularly in the Optimal Transport (OT) study as a metric for measuring the similarity between distributions and their application in the field of computer vision.

In the second stage of the study of color and texture, we use the color and texture information united in a single feature space. We deepen in the study of Gabor filters, and we explore the spectral decomposition of an image in a complex color space. We recover the objects' local texture information in an image with this strategy, taking into account the scene's luminance and chrominance information. With the use of classic vision methods and the feature space developed, we recover the perceptual contours of the objects in an image and, consequently, their segmentation. We show the versatility of this space using different techniques for object segmentation. While this framework is fully unsupervised, we show that it is also helpful in identifying the importance of color and texture in human-made segmentations. Finally, we show that it is possible to obtain high-level features from this spectral decomposition.

Throughout the following chapters, we address an extensive study of color and texture properties of an image using different kinds of images containing the information of interest to test our methods' robustness. In the first stage, we carry out the analysis of the color and the texture separately using images, in the case of color, containing low color variation and, for texture, using grayscale images with homogeneous textures. For the second stage of the analysis, we mainly use natural color images.

The main contributions of this part are:

1. Review of the state of the art of global color representations and texture characterizations.

2. Review of the state of the art of similarity measures, particularly the interpretation of the OT in computer vision: the Earth Mover's Distance (EMD).

3. A qualitative and quantitative study between the most popular measures in the comparison of distributions and the EMD.

4. An unsupervised image retrieval system based on global color/texture information.

5. Extensive analysis of Gabor filters and their properties in the space-frequency domains.

6. Generation of a feature space that includes the color and texture information of an image.

7. Unsupervised framework for natural image segmentation.

The overall distribution of color in an image is a helpful clue that contributes to describing a natural image's content. If we look around us, we can see that many of the environment's materials and objects only exist with specific colors. For example, the clouds are primarily white; the grass is green; the ocean is blue, and so on. Performing the same experience, but this time with textures, we realize that we are surrounded by them everywhere. We find textures, for example, at textiles, buildings, tilings, and on skins or objects surfaces. The color and texture of an image is valuable information; it helps to characterize images that contain landscapes with mountains, jungles, urban environments, deserts, or other scenes with different elements by their color and texture distributions. Therefore, the perception of such information is a powerful tool for classifying and recognizing particular objects and materials.

For decades several vision algorithms have sought to exploit this information. Color and texture are of relevant importance for their use as a feature to characterize objects. Therefore, this chapter addresses the definition and the various representations of the color and the texture information. As far as color information is concerned, we give a brief introduction to what color is and how we can represent it. In the case of texture, we present a brief introduction to textures, including their types and an overview of various methodologies for its analysis, highlighting each method's advantages and disadvantages.

2.2 Color

The study of color is one of the most perplexing and exciting subjects in vision. Although understanding the basic concept of the color spectrum is easy to explain, the theory of color is an infinitely more complex subject with scientific and artistic origins. For example, Newton was interested in the physical properties of color and discovered, with his famous experiment of the light beam projected on a prism, that white light

combines all colors across the color spectrum [Newton, 1704]. On the other hand, authors like Goethe dedicated their work about color to a more human-centered analysis. He analyses the perception of color information through a series of experiments that measure the eyes' response to specific colors [von Goethe, 2015].

Today we know that the electromagnetic spectrum visible to humans is between 380 and 750 nm (see figure 2.2a). We call visible light (or simply light) the electromagnetic radiation between this range of wavelengths. Therefore, we can define color as the main property of visible light by which a human observer can distinguish different kinds of light [Kerr, 2003].

From a biological point of view, we can perceive different colors thanks to the human visual system (HVS) composed, roughly, by the eye's elements such as the retina and its photoreceptors (cones and rods), the nervous system, and the part of the brain that interprets information [Fairchild, 2005]. The cones are the primary photoreceptors for color vision because they act when more light is available, whereas rods are active mainly in low illumination conditions. The three types of cones we have are appropriately referred to as L, M, and S since they are sensitive to different wavelengths of light; long, medium, and short wavelengths, respectively

The figure 2.1 shows different examples of color images. Specifically, the leftmost image is a synthetic (human-made) image where the colors encode the image's perceptual information. The remaining three images are natural images that show the importance of color information and how many elements of nature have a specific color representation distribution.

(a) **(b)** **(c)** **(d)**

Figure 2.1: Some examples of color images: **(a)** Synthetic image, and **[(b) (c) (d)]** three natural images.

2.2.1 Color Theory

Historically, there have been several attempts to explain the HVS and interpret its function in color vision. Two of the most popular theories of the mechanism of color vision are the trichromatic theory and the opponent-colors theory [Fairchild, 2005]. The first of these, proposed by Maxwell, Young, and Helmholtz [Von Helmholtz, 1867; Young, 1802], is based on the fact that we have three types of receptors (L-M-S cones). They assume that the receptors are roughly sensitive to the red, green, and blue wavelengths. Consequently, this theory suggests that each receptor generates an image weighted by

the brain to sort out the color appearances.

The second theory is based on Hering's subjective observations [Hering, 1878]. In his experiments, he noted that certain hues were never perceived to occur together. For example, the color perception was never described as reddish-green or yellowish-blue. This response suggested that the red–green and yellow–blue color pairs had something fundamental that caused them to behave like opponent colors. This theory gained strength in the mid-20th century, where, supported by quantitative data, the stage theory emerged. This modern theory suggests that color perception is done in two stages. The first stage coincides with the trichromatic theory, so the LMS cones generate three color-separation images; however, in the second stage, the retina neurons encoded the colors into opposing signals [Fairchild, 2005].

2.2.2 Color Representations

Human color perception depends on the amount and wavelength of light captured by the eyes. Therefore, perceived colors can vary due to several factors, such as the type of surfaces (or objects) where the light is reflected, the environment, and even the observer's eyes. However, it is definite that the perception of color is an entirely arbitrary creation of our nervous system, and it is not contained in wavelengths or light-reflecting objects and materials [Goldstein, 2009]. In other words, the interpretation of this information is entirely subjective. A clear example of this is the naming of colors. When an incident spectrum contains all frequencies in the range of visible wavelengths, humans perceive objects that reflect all frequencies as clear, luminous, or *white*. In the opposite case, when the material absorbs and does not reflect the visible frequencies, it is perceived as dark, opaque, or *black*. Some works in this regard state that the naming of colors varies according to culture and language [Berlin and Kay, 1991]. However, it is possible to find a correlation between languages and identify eleven basic color terms in the English language that seem to be anchored across the different languages as points in a particular color representation [Kay and Regier, 2003].

At first glance, the tasks and experiments mentioned above appear to be simple and straightforward for a human being; however, replicating this in a machine is quite challenging. Therefore, the definition of a coherent method of describing color is essential to represent it and for its use in digital image processing.

A *color model* is a mathematical way of describing colors. Such abstract mathematical models are the result of the theories of color described above. We see this in the fact that most models represent a color using three values. This consensus has allowed the development of color models representing colors as a 3-d property using vectors or tuples of numbers [Douglas and Kerr, 2005]. Like any property in 3-d, real colors can be represented as a point in space using a specific coordinate system. A *color space* is thus the method of mapping the visible colors a the color model, and consequently,

they define the range of colors that can be displayed or reproduced on a medium.

In principle, there are differences between color models and color spaces. For example, models are independent of physical devices (e.g., screens, printers) while color spaces are not. However, in abuse of language, in this document, we will use the term color space to mean a particular fully specified color model. In the following subsection, we describe some color spaces that are important both in the theoretical field and in the technical field for the representation of color and its use in the computer vision applications developed in this thesis.

Color models and color spaces

Color spaces are the quantitative links between the wavelength distributions of visible light and the colors psychologically perceived by the HVS. Since this perception is entirely subjective, it is necessary to take the observers into account when modeling a color space. The *Commission Internationale de l'Éclairage* (CIE) is one of the main contributors to creating color models. In 1931, they defined a color-mapping function based on a standard observer, representing an average human's chromatic response within a 2° arc, to primaries at $R_0 = 435.8$ nm, $G_0 = 546.1$ nm, and $B_0 = 700$ nm [Bull, 2014]. The positive color-matching functions specify the three standard primaries of color X, Y and Z [CIE, 1932], which allow defining any visible color of the spectrum (see figure 2.2a) as a weighted sum of three primary colors [Wright, 2007].

With the standard primaries, it is possible to quantify an object's colors using a standardized method that considers the human eye's (observer) response to these colors through the **XYZ color model** (also known as the CIEXYZ color model). In the model X, Y and Z are the amounts of each primary needed to produce the desired color.

$$c(\lambda) = (X, Y, Z) \tag{2.1}$$

The primary Y is chosen such that its color-matching function exactly matches the luminous-efficiency function for the human eye, i.e., Y measuring the luminance of a color [Wright, 2007]. Therefore, to define a color in the XYZ color space, we need to provide the weights for the X, Y, and Z primaries, for example, $c = xX + yY + zZ$, where

$$\begin{aligned} x &= \frac{X}{X+Y+Z} \\ y &= \frac{Y}{X+Y+Z} \\ z &= \frac{Z}{X+Y+Z} \end{aligned} \tag{2.2}$$

Under this representation, we can ignore the luminance's dimension by normalizing

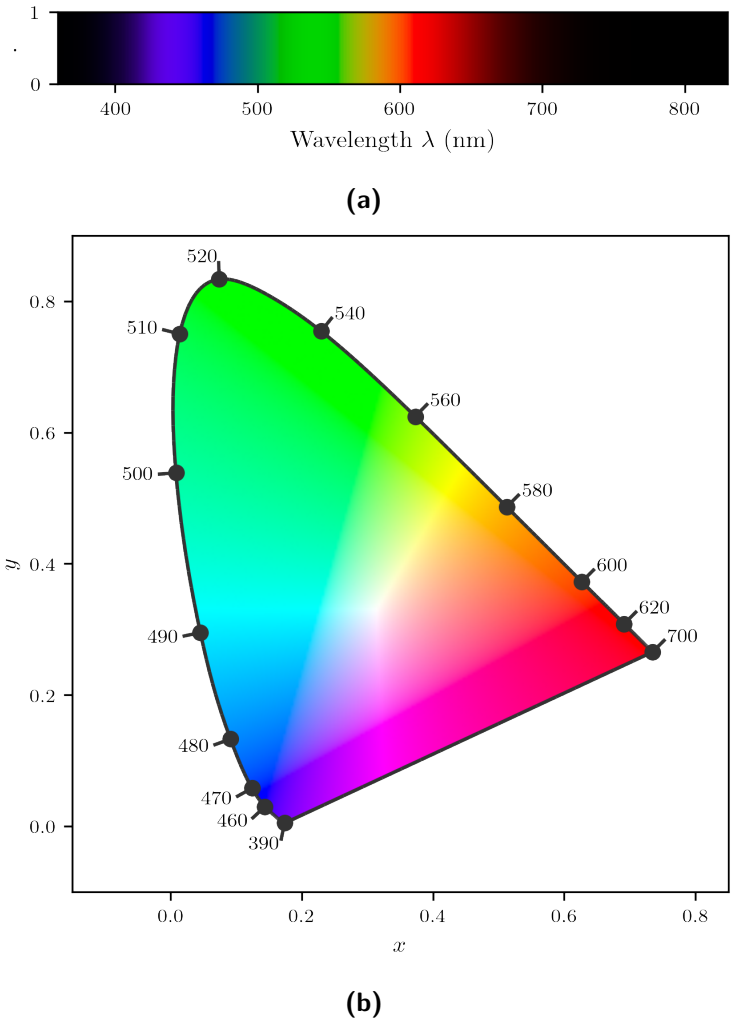


Figure 2.2: CIE 1931 2 2° Standard Observer: **(a)** Visible light spectrum, and **(b)** Chromaticity diagram.

the primaries with the total light intensity; $x+y+z=1$. This strategy allows showing all visible colors of the spectrum in a diagram. Figure 2.2b shows such a diagram known as the CIE 1931 2° standard observer chromaticity chart. The x and y axis of the diagram give the normalized amounts of the X and Y primaries for a particular color, and hence $z = 1 - x - y$ gives the amount of the Z primary required. The diagram reveals that large x values correspond to red or orange hues, large values of y correspond to green, and large z values correspond to blue, violet, or purple hues. The chromaticity depends on the dominant wavelength and saturation and is independent of luminous energy. Colors with the same chromaticity but different luminance all map to the same point within this region. Moreover, the chart boundary represents maximum saturation for the visible colors, and the diagram forms the boundary of all perceivable hues [Bull, 2014].

The CIEXYZ model is a reference that has been used as a basis for defining other color spaces, and therefore as a standard basis in image processing for moving from one color space to another. The color gamut that can be created through combinations of

any three primary colors (e.g., RGB) can be represented on the chromaticity diagram by a triangle joining the three colors' coordinates.

The **RGB color model** is one of the most popular models in computer vision and image analysis. This is an additive model coming directly from the three-component theory; this means that three color light beams (their wavelength light spectra) are added together to make a final color [Gonzalez and Woods, 2008]. The model consists of three independent planes, represented as a three-dimensional vector, one in each of the primary colors: red, green, and blue. Therefore, to define a color in this model, we need to specify the proportion of red, green, and blue colors.

Contrary to the XYZ model, the RGB color space is a device-dependent space; that is, different devices may reproduce or detect the same RGB value differently since the color elements and their response to the individual R, G, and B levels vary according to the manufacturer.

The RGB color space is the most popular color space based on the RGB color model. We can geometrically represent the set of colors of this color space as a cube (see Subfig. 2.3a) that maps the red, blue, and green dimensions onto the x, y, z axes of the 3-d Cartesian coordinate system in a Euclidean space. The non-negative values of the RGB triplet (r, g, b) are in the range $[0, 1]$, where the origin at the vertex $(0, 0, 0)$ encodes the color black, and the vertex $(1, 1, 1)$ encodes the color white.

The **HSV color model** and the **HSL color model** are cylindrical color models that remap the RGB primary colors into dimensions that are easier for humans to understand. The HSV and HSL color models share two of their three dimensions, the hue and the saturation. The third dimension of the HSV model is the value, while the HSL model has a lightness dimension. Below is a more detailed description of these dimensions.

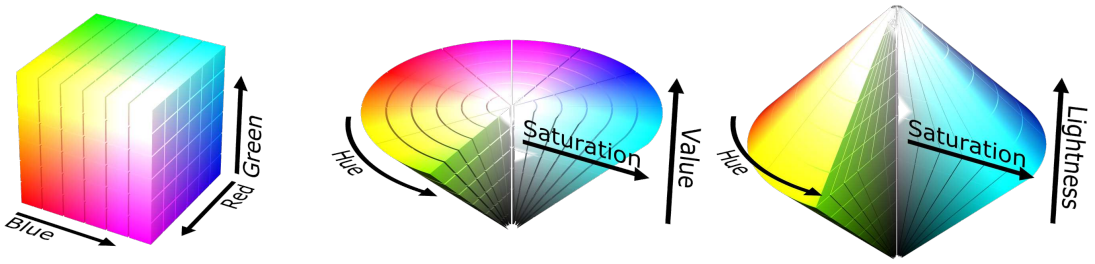
- Hue specifies the angle of the color on the RGB color circle. A 0° hue results in red, 120° results in green, and 240° results in blue.

- Saturation or colorfulness controls the amount of color used. One particular thing about the saturation between the two cylindrical color spaces is that even though the saturation dimension theoretically is similar between them (controlling how much pure color is used), the resulting saturation scales differ between the models caused by the brightness to lightness remapping (see differences between saturation image channels in figure 2.10). Therefore, for the HSV model, a color with 100% saturation will be the purest color possible, while 0% saturation yields grayscale. On the other hand, to obtain the purest color in the HSL model, we need 50% lightness.

- Value controls the brightness of the color. A color with 0% value is pure black, while a color with 100% value has no black mixed into the color. Because this

dimension is often referred to as brightness, the HSV color model is sometimes called HSB.

- Lightness controls the luminosity of the color. This dimension is different from the HSV value dimension in that the purest color is positioned midway between the black and white ends of the scale. A color with 0% lightness is black, 50% is the purest color possible, and 100% is white.



(a) RGB cube **(b)** HSV cone **(c)** HSL bicone

Figure 2.3: Geometrical representation of colors in the RGB, HSV and HSL color models.

It is important to note that the three dimensions of the HSV/HSL color models are interdependent. If the value/lightness dimension of color is set to 0%, the amount of hue and saturation does not matter as the color will be black. Likewise, if the saturation of a color is set to 0%, the hue does not matter as there is no color used.

Unlike the RGB color space, we can not represent the HSV/HSL triplet values $(h, s, v)/(h, s, l)$ in a 3-d Euclidean space. The hue's circular nature forces the first dimension to be in an angular space between 0° and 360°, while the remaining two dimensions inhabit a linear space with values between 0 and 1. Consequently, the HSV color space is best visualized as a 3-d cone and the HSL color space as a 3-d bicone (see Subfigs. 2.3b and 2.3c respectively).

The **LAB color model** (also referred to as CIEL*a*b* or CIELAB) results from the opponent-process theory of human perception. In it, we also express the color with three values. Channel L represents the perceptual luminance, whereas channel A represents the scale from red to green and channel B from yellow to blue. The arrangement above is consistent with the two opponent color pairs that humans cannot perceive simultaneously.

The color space from the LAB model was born to be a perceptually uniform space, i.e., a space where a given numerical change corresponds to the same perception of color change. As a non-linear transformation of the XYZ color space, the LAB color space is a device-independent space. This property implies that its gamut is related to the CIE standard observer model, and therefore, it is impossible to generate a visual representation that displays all the colors of its gamut. The triplet (l, a, b) gives the

LAB space coordinates. The first value represents the luminance L, and it may take values from 0 to 100, where 0 points to black and 100 indicates (diffuse) white. The remaining two coordinates are technically unbounded, though it is commonly mapped to the range $[-128, 127]$. Negative values indicate green and positive values red for channel A, while negative values indicate yellow and positive values blue for channel B.

This color space offers some advantages over the spaces described above, especially in the image processing field. This space was constructed to approximate human color vision, making it helpful in calculating differences between two neighboring colors with high precision. However, this color model contains colors that are not physically representable by the devices, and a poor color quantization (bits per channel) can generate significant errors.

Figure 2.4 shows the three channels of the different color spaces reviewed so far. The input image is a natural color image where the three primary colors, red, green, and blue, naturally stand out. We transform channel values in the range between 0 and 255 and displayed them on a grayscale for the visualization of each color channel.

Luminance-chrominance color spaces

Color spaces mostly map the perception of color as a three-dimensional property [Douglas and Kerr, 2005]. However, these dimensions can be encompassed in only two aspects; therefore, we can classify them into luminance-chromaticity and luminance-chrominance color spaces [Kerr, 2003]. In both cases, luminance is the property that describes the brightness of the light; however, both categories have different ways of defining color. Chromaticity-based spaces define color independently of luminance (or the luminance equivalent in a particular color space). In a chrominance-based space, the chrominance values of the image change as the light intensity varies.

We can obtain a color space based on chromaticity from the classic trichromatic models described in the previous section. The chromaticity of these models consists mainly of two independent parameters. For example, the xyY color space, from which we obtain the CIE chromaticity chart (see figure 2.2b), uses the X and Y dimensions to calculate chromaticity. In RGB space, it is possible to obtain a space based on chromaticity following the same principle, using only the R and G channels for its calculation. For cylindrical color spaces (HSV, HSL), the independent parameters that describe the chromaticity are hue and colorfulness (saturation) dimensions.

The chromaticity-based models are reputed for their use in image editing and color correction. However, the chrominance-based spaces are helpful if we are looking for a uniform distribution of the color information in an image. Some examples of color spaces in this category are CIELAB and CIELUV. In them, the colorfulness information of an image is found in channels A and B (resp. U and V), and the final color is defined

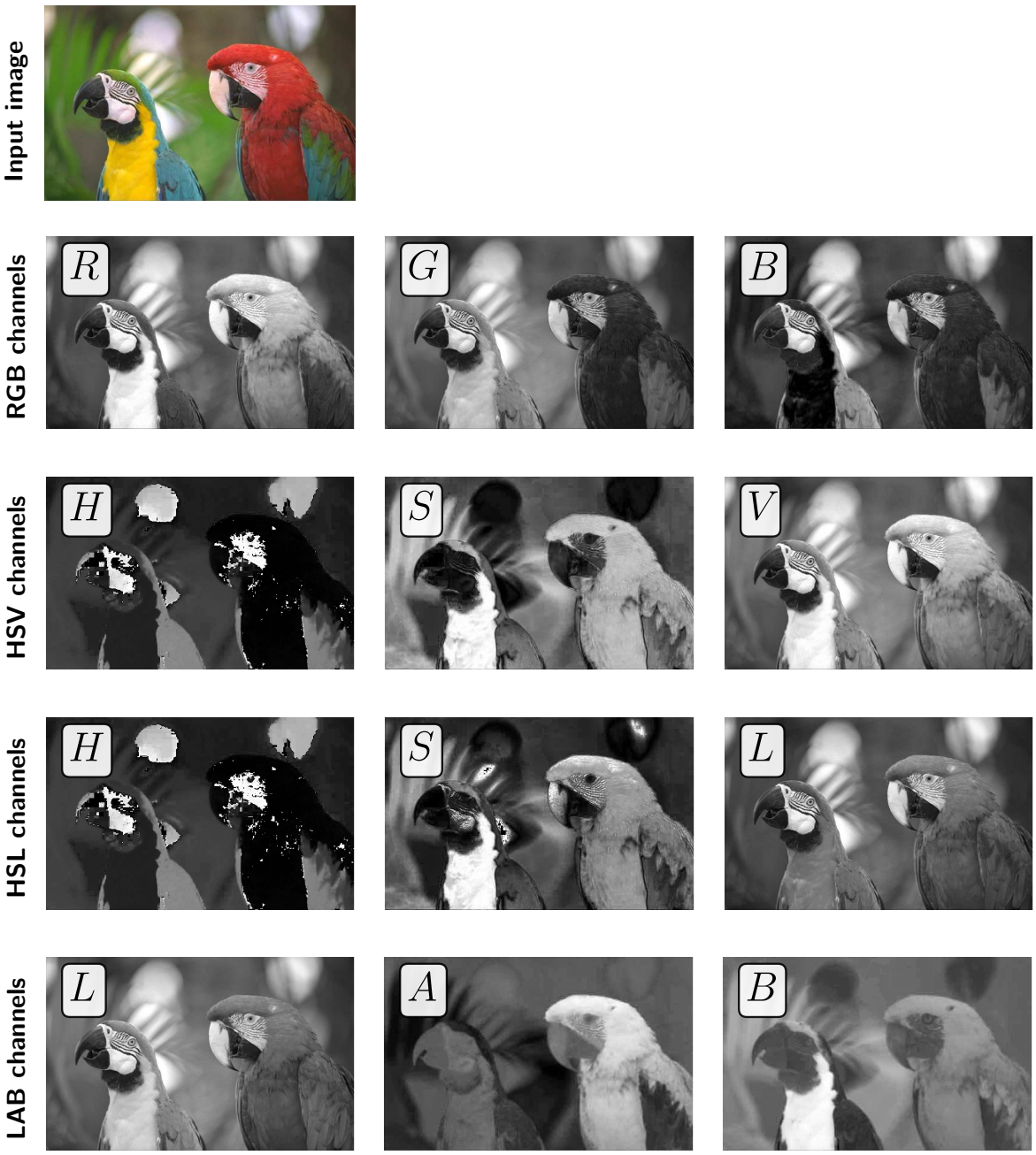


Figure 2.4: Color channels of an image in different color spaces in a grayscale.

by the brightness of the light defined by the luminance L.

The use of spaces based on luminance-chrominance reduces the dimensionality of the color to two channels L and C. To make this two-channel color representation possible, we combine the values of channels A and B with the chrominance function.

$$C = A + \mathrm{i}B \tag{2.3}$$

In the same way, it is possible to obtain a luminance-chrominance color space using the dimensions of the cylindrical HSV/HSL color spaces. In that case, the hue H and

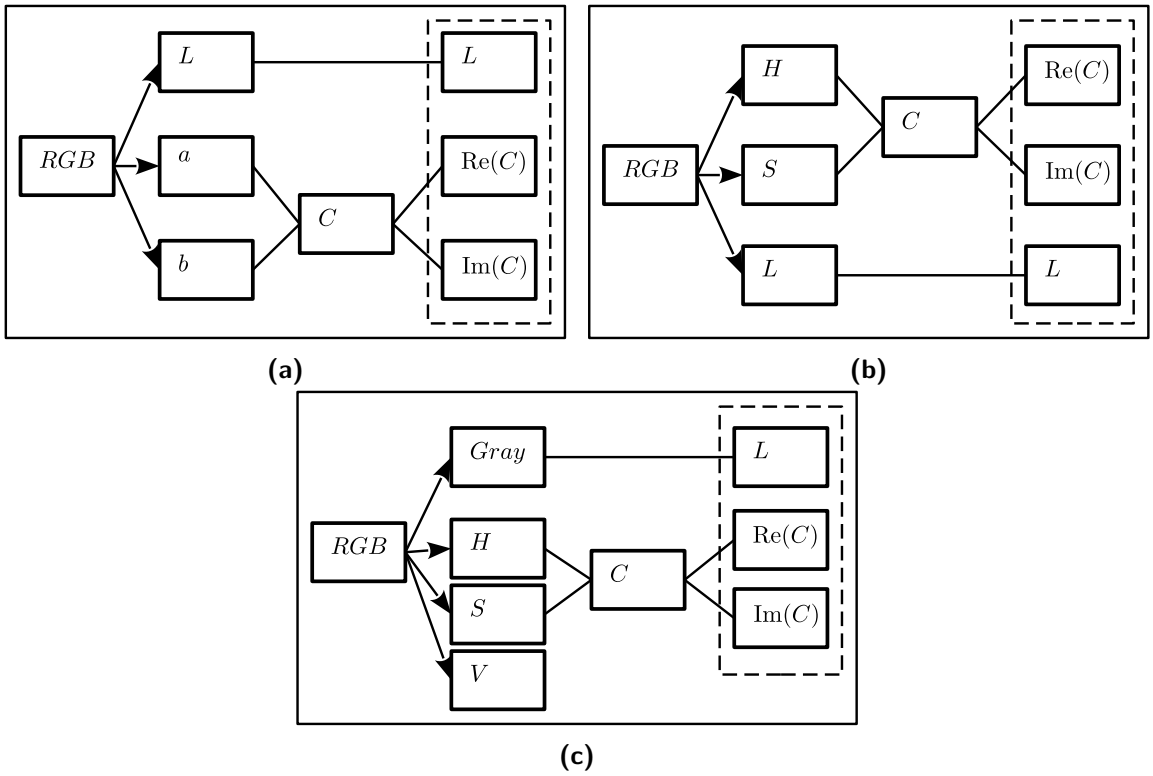


Figure 2.5: Graphic display of tree alternatives to obtain the complex color representation of an image: **(a)** from LAB color space, **(b)** from form HSL color space, and **(c)** from HSV color space.

saturation S channels describe the chrominance such that the function

$$C = S\mathrm{e}^{\mathrm{i}H} \tag{2.4}$$

defines the complex chrominance channel.

These representations have the advantage of reducing the dimensionality of the color information. In image processing, we can obtain these color spaces from the transformation of the image from the RGB space to the LAB, HSV, and HSL spaces, from which we get the chrominance variables and, consequently, the complex chrominance channel. Regarding the luminance, it may differ depending on the input color space. For example, for the LAB and HSL spaces, this variable is naturally defined; however, in the HSV space, the luminance channel is obtained from transforming the RGB input image to a grayscale image. Figure 2.5 shows the diagrams for obtaining an image represented in two channels.

Figure 2.6 shows the transformation of a natural image (first row of the figure array) from RGB space to complex two-channel color space. Each row in the figure shows the luminance channel and the two parts (real and imaginary) of the chrominance channel. Note that as it happened in the representation of the classic color spaces (figure 2.4) since the method to compute saturation differs in HSV and HSL spaces, the chrominance channel is different. The same effect occurs with the L luminance

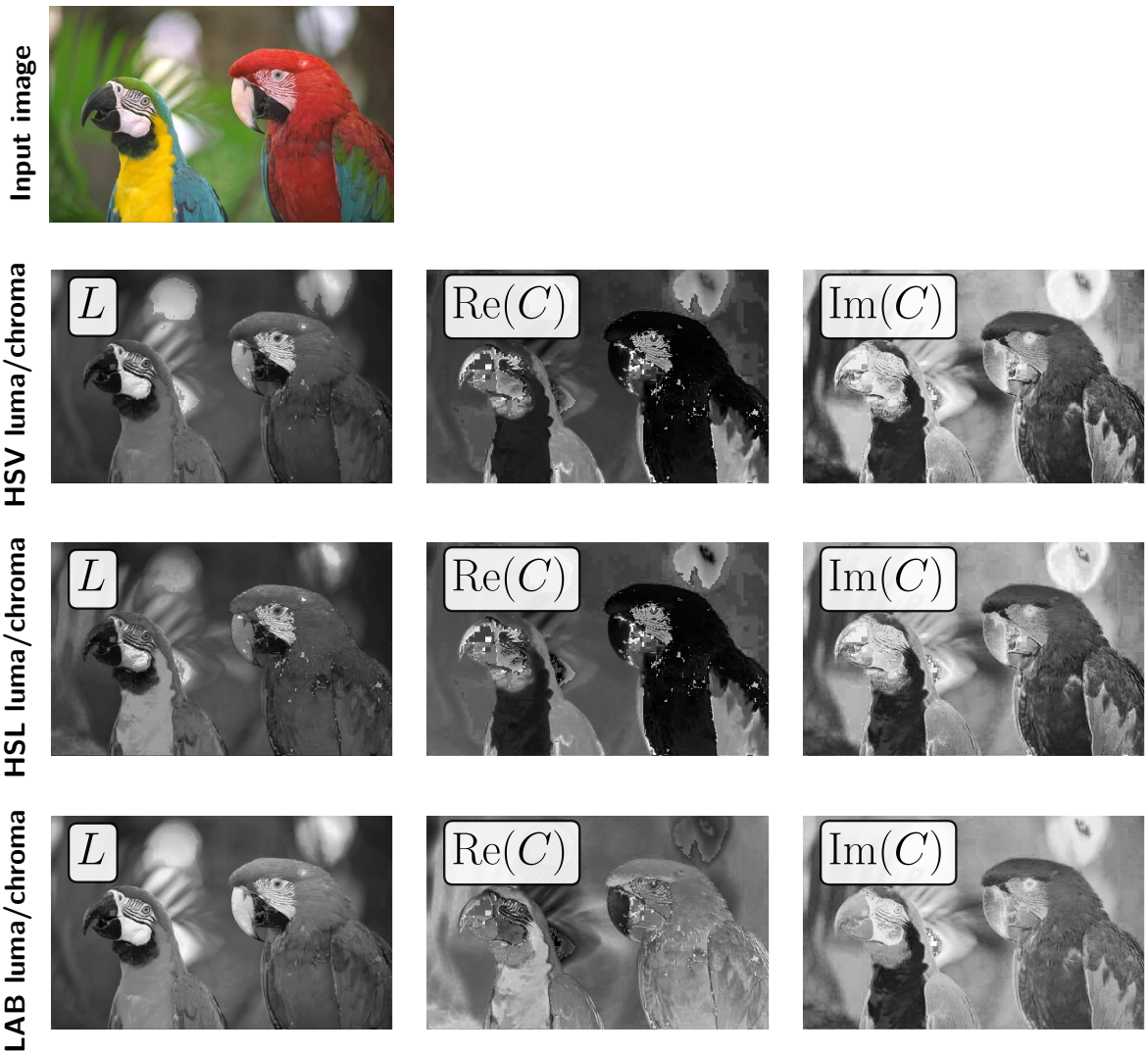


Figure 2.6: Color channels of an image in different color spaces in grayscale.

channel from LAB space and the luminance from the RGB to gray transformation; in theory, both are the same, but we can see differences in practice.

2.2.3 Compact Color Representations for Image Processing

Color spaces serve as links between color theories and representation for better visualization and mathematical interpretation of colors. However, it is often necessary for color information processing to represent the color pixel values more compactly. This property benefits the development of faster and more performant algorithms; the challenge is to maintain the input information's consistency and not lose the primary characteristics of the color distribution. In the following subsections, we will show some of the representations commonly used to represent color information compactly.

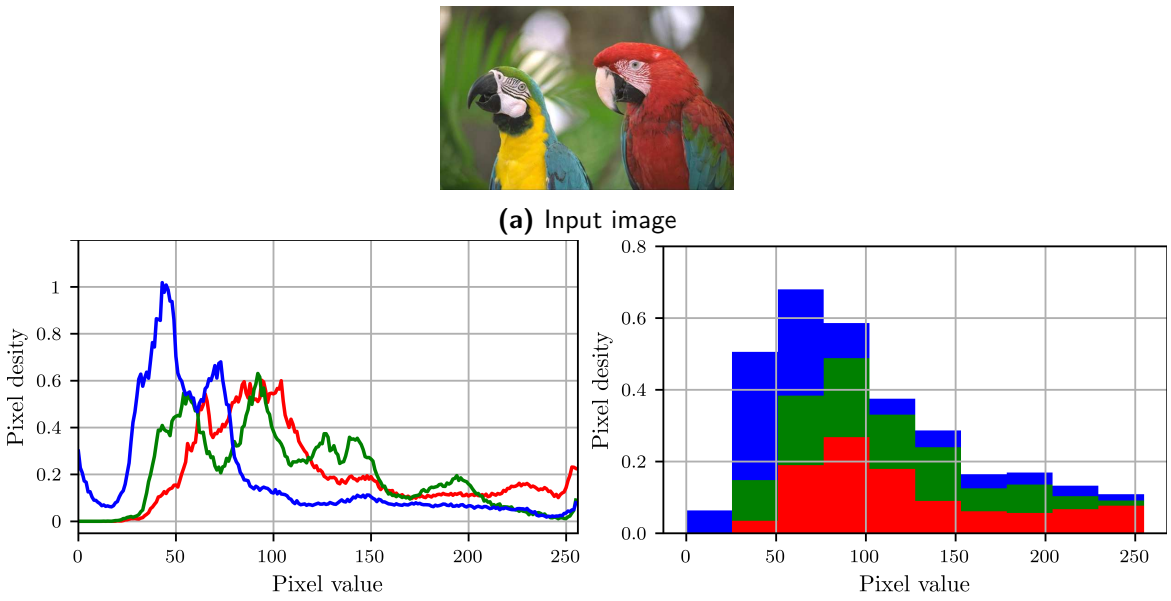
Single-channel Color Histogram

We can represent the global color distribution of an image employing a color image histogram. A histogram $\mathbf{h} = \{h_i\}$ is a standard statistical tool that approximates the distribution of numerical data. Such representation is done by a discrete function h_i that maps an integer vector to the set of non-negative reals [Scott, 2008]. These vectors represent bins (or their centers) in a fixed partition of the underlying feature space. The associated reals are a measure of the mass of the distribution that falls into the corresponding bin. For instance, the single-channel color histogram of a 2-d image given by the following expression

$$h_i = \sum_{j=1}^{n} \mathbb{1}_{B_i}(x_j) \quad \text{with} \tag{2.5}$$

$$\mathbb{1}_{B_i}(x) = \begin{cases} 1 & \text{if} \quad x \in B_i \\ 0 & \text{elsewhere} \end{cases}$$

as indicator function. The n pixels of the single-channel image are arranged in a one-dimensional vector $\{x_1, x_2, \cdots, x_n\}$; the set of possible color values is split into k intervals, so h_i is the number of pixels in an image that have a color value in the interval $[t_i, t_{i+1})$ denoted by B_i with $0 < i \leq k$.



(a) Input image

(b) Single-channel color pixel distribution

(c) Single-channel color pixel histogram

Figure 2.7: 1-d image color representation. 1-d pixel distribution and 10 bins 1-d pixel histogram.

Fig. 2.7 shows the one-dimensional representations of the color information of a natural image. In it, we first plot the distribution of each channel of the image (RGB in this case), and then we show the color distribution of the pixels compressed using

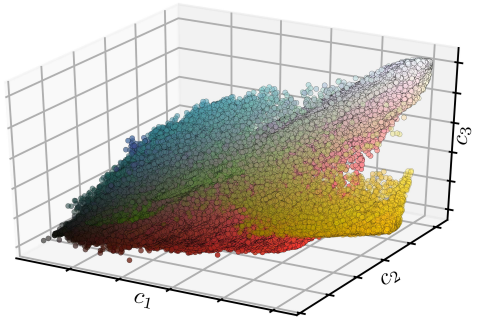
a 1-d histogram. We obtain the distributions by setting the number of bins equal to the maximum number of color values of an 8-bit image, that is, $b = 256$. The single-channel histograms split the color space into $k = 10$ bins. In both plots, the x-axis represents the channel color value, whereas the y-axis represents the number of pixels (density) normalized between 0 and 1. Lastly, the color of the plots corresponds to the color channel of the RGB space.

This representation's advantages are that it is invariant to the rotation or translation of the image and, to a lesser extent, it is also invariant to changes of point of view and scale changes. Besides, we can compact as much as we need the image color information by reducing the count intervals, that is, by selecting a small number of bins k. Despite the advantages offered by the color representation in simple channel histograms, analyzing the color channels individually does not provide a global idea of the color distribution. This effect occurs because only the blend of the three-channel values gives the final color of a pixel in RGB space. A better way to analyze color information is to consider all channels at the same time.

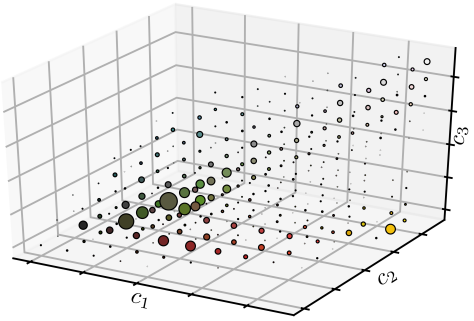
3-d Color Histogram

The 3-d color histogram copes with the drawback of the color representation in single-channel color histograms. Mainly, this representation is an extension of the single-channel histogram to three dimensions. The discrete function $h_{i,j,k}$ maps the 3-d integer vector of pixel color values $x_{i,j,k}$ into a Euclidean cube where each axis corresponds to a color dimension divided into k intervals. We count the number of pixels whose values fall into the bin $B_{i,j,k}$ with the indicator function as in Eq. (2.5).

(a) Input image



(b) 3-d image color pixel distribution



(c) 3-d color pixel histogram

Figure 2.8: 3-d image color representation. 3-d pixel distribution and ten bins 3-d pixel histogram.

We can see the 3-d distribution of the color of the pixels in a color image and its histogram of $k = 10$ bins in Fig. 2.8. In the figure, we can notice how the colors most present in the image stand out in the histogram, generating larger spheres in the 3-d cube. Also, the shape of the pixel distribution is maintained. A reflection we can make about the number of bits required to encode the image color as a 3-d distribution is that, when representing the colors in a histogram with a low number of bins, the information to manipulate is significantly less.

On the other hand, 3-d color histograms suffer from some drawbacks. Because histograms are fixed-size structures, they cannot achieve a good balance between expressiveness and efficiency. For example, applying a coarsely quantized histogram on images containing a large amount of color information would lead to losing this information. In the opposite case, for images that contain a small amount of color information, a finely quantized histogram is highly inefficient. In the example of the parrots' image (Fig. 2.8), we see that the 10-bin histogram is severe with the blue color of the feathers, causing it to almost disappear from the 3-d representation.

Color Signature

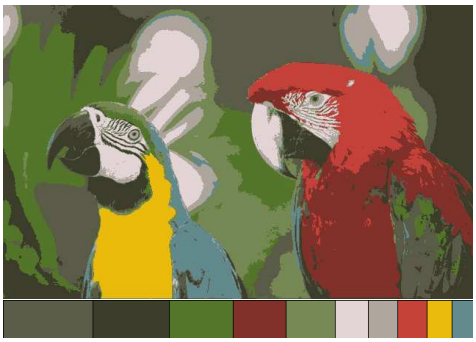
The color signature is another representation for the compression of color information. The signatures are a type of boosted histogram since they make adjustable the number of bins of each color dimension in the 3-d histogram. Two of the main strategies to obtain color image signatures are k-d tree or k-means algorithms.

A signature $\{s_j = (m_j, w_{m_j})$ denotes a set of feature clusters. Each cluster is represented by its mean (or mode) m_j and the fraction of the pixels w_{m_j} that belong to that cluster [Rubner and Tomasi, 2001]. The exciting thing about signatures is that they can adapt the number of clusters j according to the image's complexity. Therefore, images with a low or straightforward quantity of color have short signatures, while images with complex variations of color have long signatures.

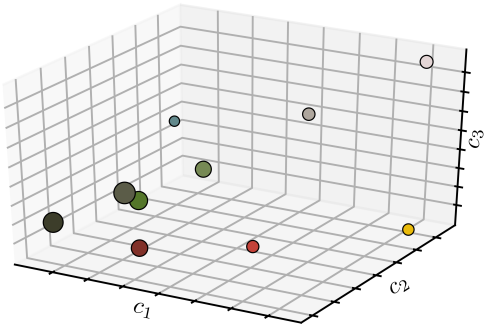
In Fig. 2.9, we depict the color signature of the parrot image and a rendering of the input image using the signature colors. Likewise, we plot the centroids of the $k = 10$ clusters obtained with the k-means algorithm in a 3-d space. The spheres' size in the 3-d plot represents the fraction of pixels w_{m_j} that belong to each cluster. Compared to the 3-d color histogram, the color signature better represents the original color information of the image, maintaining the high information compression rate without losing some colors of the original distribution, as is the case of the 3-d histogram with blue feathers mentioned above.

Finally, we show more examples of the different techniques to compact the color information reviewed in this section in Fig. 2.10. These examples include the images of Fig. 2.1, where we show three natural and one synthetic image. To calculate the histograms (single-channel and 3-d) and the color signature, we set the number of bins

(a) Input image



(b) Color signature



(c) 3-d representation of color ignature

Figure 2.9: Image color signature and the 3-d visualization of signature clusters.

(and clusters in the case of the signatures) at ten for visual comparison purposes.

We obtain some interesting observations from the analysis of the different compact representations of color. For example, with the synthetic image plots (first column of the figure), we find that a medium-fine 3-d histogram with ten bins is inefficient since many half-empty bins could be removed. On the other hand, the color signatures do a good job of compression, mainly respecting the shape and density of the original color distribution, for example, in the clownfish image (central column of the figure). However, there is a problem related to the global analysis and compression of color information. When there are objects in the image with a low number of pixels, and yet they are perceptually visible, there is no rendering technique that can preserve them. We can see this effect in the snow mountains' image; although the yellow and red color of the cable cars are perceptive, neither the 3-d histogram nor the color signature makes the colors appear.

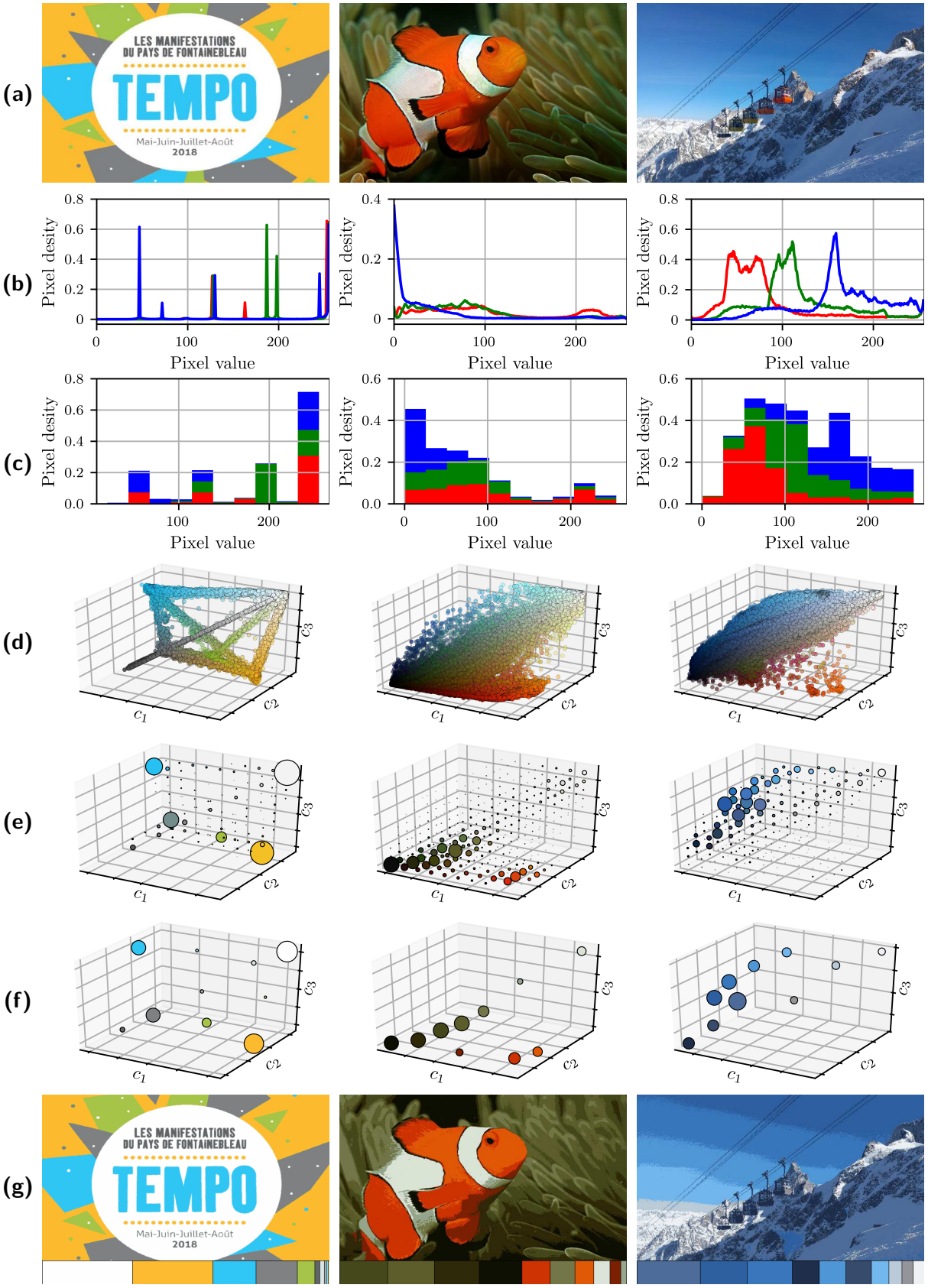


Figure 2.10: Compact representations of color information: **(a)** Input color image, **(b)** Single-channel color distribution, **(c)** Single-channel color histogram, **(d)** 3-d color distribution, **(e)** 3-d color histogram, **(f)** 3-d color signature, **(g)** Color signature clusters.

2.3 Texture

There is a disagreement in the definition of texture in the field of computer vision. It is possible to give a mathematical definition based on its statistical properties; however, these properties are very imprecise and restrictive to adapt to the diversity of existing textures.

The definition that we prefer in this work is based on an experimental finding: a texture is a field of the image that appears as a coherent and homogeneous domain; that is, it forms a whole for an observer. In fact, the texture coherence property placed in the context of being perceived as a homogeneous whole for the human eye is most often sought for image processing, either to isolate textures, segment and recognize regions.

We show some examples of natural textures in the Fig. 2.11. These images come from the reference work Brodatz and show the possible variety of textures commonly used to test different algorithms and methods of vision. Generally, we can classify such textures according to their origin as natural or artificial, the latter being those created by man. We can also classify them by the regularity of the pattern they display as real or stochastic. Finally, we can classify them according to the image proportion they cover into homogeneous, weakly-homogeneous, or inhomogeneous.

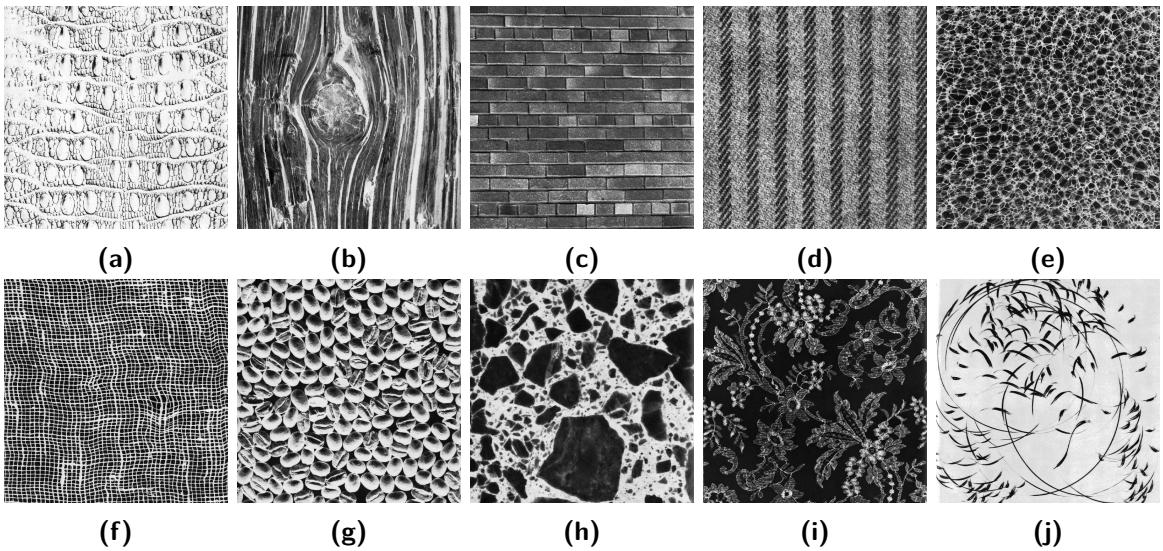


Figure 2.11: Examples of texture images and its classification: [**(a) (b) (e) (h)**] Natural textures, [**(c) (d) (f) (g) (i) (j)**] Human-made textures, [**(c) (d)**] Regular textures, [**(g) (h)**] Stochastic textures, [**(c) (d)**] homogeneous, [**(a) (b) (f) (g) (h)**] weakly-homogeneous, [**(i) (j)**] inhomogeneous.

2.4 Texture Characterization

The perception of textures is a key property of human vision. Although there is still no generalized definition, we can define texture using measures of coarseness, contrast, directionality, line similarity, regularity, and roughness. Therefore, the features that characterize texture attempt to capture the granularity and repetition of perceptually similar patterns of surfaces within a region of the image, such that a human observer perceives the region as homogeneous. Unlike color, texture information is not a purely pixel-level property. Texture implies the notion of spatial extent, that is, that the spatial variation of intensities of a group of pixels generates textures in the images.

There are numerous studies that review, compare and organize the work of texture analysis in different ways [Materka and Strzelecki, 1998], [Zhang and Tan, 2002], [Bharati et al., 2004],[Lukashevich and Sadykhov, 2012], [Humeau-Heurtier, 2019]. One possible organization is based on its operating principle, which classifies the texture characterization techniques into

1. Statistical methods
2. Structural methods
3. Model-based methods
4. Transform-based methods
5. Graph-based methods
6. Learning-based methods
7. Entropy-based methods

This section reviews five of the most widely used literature methods and their techniques for extracting texture features.

2.4.1 Statistical Methods

Statistical methods contemplate that textures are determined by how the gray levels are spatially distributed over the image pixels. In these methods, the gray level distribution of the image is represented by a histogram.

The first approach in this category is the histogram properties analysis [Aggarwal and K. Agrawal, 2012]. 1-d image histogram, brightness, and contrast are among the first-order statistics from which we can compute the central moments: mean, variance, skewness, and kurtosis. These properties provide information on the distribution of the gray levels of the image from a global point of view, taking into account individually the gray level of the pixels. However, they do not provide any information on how the gray level of a pixel at a given location statistically affects the gray level value of another pixel at a relative location from the reference pixel. The second-order statistical properties explore this option and describe the texture based on comparing two pixels'

intensity values. In this case, the Co-Occurrence matrix [Haralick et al., 1973] is the second-level histogram that maps the pixels' intensity distribution. Some of the texture features extracted from the second-order statistics are Angular Second Moment (ASM), contrast, dissimilarity, homogeneity (Inverse Difference Moment), entropy, maximum probability, mean, standard deviation, correlation, and energy.

Local Binary Patterns (LBP) [Ojala et al., 1996] are another technique for obtaining second-level histograms. This approach summarizes the spatial structure and local contrast of an image within a binary pattern, comparing the gray level of each pixel with its neighborhood. If the central pixel's intensity value is more significant than its neighbor, it is denoted by 1, otherwise 0. Subsequently, we construct a binary array following a consistent ordering of the neighboring values, which are transformed into a decimal number and stored in a new array. The process of thresholding, construction of binary strings, binary to decimal transformation, and storing of decimal output is performed for all pixels in the image, resulting in an LPB image. Finally, the second-level histogram for texture characterization is obtained from this resulting LBP image.

2.4.2 Structural Methods

The structural methods are based on the decomposition of the image in basic units, i.e., in elements, low-level primitives, or texels. Such units can be points, lines, regions, or shapes. The basic units and their spatial arrangement in the image are used to characterize the textures. These approaches consider that textures are patterns formed by replication, more or less regular, of a basic unit. The arrangement of the primitives allows obtaining geometric relationships and subsequently statistical properties that serve to characterize textures. Structural techniques aim to determine the texture primitives and define the location rules [Humeau-Heurtier, 2019].

Depending on the application, structural techniques differ according to the choice of primitives. Some of the commonly considered primitives are pixels [Lu and Fu, 1978], regions of uniform intensity [Tuceryan and Jain, 1993], line segments [Carlucci, 1972], or peaks in the gray level distribution [Ehrich and Foith, 1978]. For the recovery of these primitives, highly known approaches are generally used, such as the SIFT (Scale Invariant Feature Transform) operator for points characterization and the contour detectors, such as Sobel or Canny, for line and edge recovery. On the other hand, the primitive's measurements and statistics most commonly used are intensity, orientation, elongation, curvature, and compactness.

2.4.3 Model-based Methods

This group of methods stipulates that some mathematical models can describe the textures. We can subdivide this category mainly into two approaches: stochastics and fractals.

Stochastic methods for texture modeling are popular, in particular random field models. In this context, a texture model is a parametric family of spatially homogeneous random fields depending on a hyperparameters series [Winkler, 2003]. Inside such a family, a specific texture can be characterized by a unique set of hyperparameters that captures its characteristic features. According to the properties of the random fields, some of the models used for the characterization of texture are Markov Random Field (MRF) [Cross and Jain, 1983; Hassner and Sklansky, 1980], Gibbs Random Field (GRF) [Derin and Cole, 1986], Conditional Random Field (CRF), Gaussian Markov Random Field (GMRF) [Cohen et al., 1991].

Within the category of stochastic approaches, we found a group of techniques that use probabilistic approaches and mathematical morphology operators to model random textures [Serra, 1980], [Cord et al., 2010].

Fractal models consider textures as complex, chaotic systems, so they exhibit fractal behavior [Petrolekas and Mitra, 1993]. Textures, as fractal objects, have identical shapes and statistical characteristics at different scales. Fractal geometry relies on self-similarity across multiple scales and is measured with the fractal dimension [Keller et al., 1989]. Fractal model-based approaches aim to determine fractal dimension, find fractal geometry and calculate fractal measurements to describe textures in images.

2.4.4 Transform-based Methods

Transform methods map an image to a space within which the textures are characterizable. The peculiarity is that the new space coordinates allow the interpretation of the textures because they reflect the texture properties; for example, the log-polar coordinates in the Gabor transform reflect the periodicity and orientation of the textures in an image.

Within this category, one of the most notable methods for the extraction of texture features is Law's filter banks [Laws, 1979, 1980a,b]. There are also the approaches based on the Fourier transform [Ursani et al., 2007], where we use it to decompose the image into its frequency components. Following the same principle, there are the approaches based on Gabor decomposition [Gabor, 1946] and those based on wavelets [Arivazhagan and Ganesan, 2003], which analyzes the content of a texture in the frequency domain and the spatial domain. On the one hand, the Gabor filter is defined as a sinusoidal wave plane modulated by a Gaussian kernel, adapted in frequency, orientation, and bandwidth. For its part, the wavelet transform allows the analysis of the texture in the frequency and spatial domain employing the dilation and translation, respectively, of a mother wavelet.

2.4.5 Learning-based Methods

The extraction of texture features based on learning is relatively new concerning the other methods mentioned in this work. We can divide this category of approaches into two subclasses: visual dictionary methods and deep learning methods.

Visual dictionary methods are motivated by natural language processing algorithms. In this case, the aim is to generate a codebook or dictionary that contains essential geometric elements of the images, also called *textons*. In the document processing analogy, textons correspond to words; therefore, we can describe an image as the repetition (organized or not) of a set of textons.

There are different strategies for calculating textons [Zhu et al., 2005], for example, the approaches based on generative models, where an image is considered a linear combination of some base images. Such base images are represented by Gabor or Laplacian-of-Gaussian (LoG) functions and other wavelet transforms. Following generative models' principle, textons are the base functions learned from a large number of image patches. Other approaches to obtaining textons are based on discriminative modeling. In this case, the base functions are defined by rotated and scaled filters that form a family with which we convolve the image. The responses of the filters form a feature space in which it is possible to form clusters. Each cluster center then corresponds to a texton; therefore, to obtain a texton dictionary, we need to obtain from a group of training images the feature space and the cluster centers.

Models based on deep learning use Convolutional Neural Networks (CNN) to extract and represent image features. CNN methods consist of multiple locally connected layers which convolve kernels over the entire image. These approaches analyze the information of a group of images to generate a model [Lin and Maji, 2016]. The characteristics of the learned model are a function of the input images, which in the case of texture information, is expected to generalize the properties, such as granularity, frequency, and orientation of patterns in the training dataset.

2.5 Conclusion

This chapter has reviewed some key concepts for managing color and texture information in an image. Regarding color, we have seen the origin and function of models and color spaces. This information allows us to enter the complex color space that we will use throughout the following chapters to explore the relationship between color and texture. Besides, we show some of the possible representations to organize and work with color information: histograms (one-dimensional or 3-d) and color signatures.

We have briefed the different strategies to characterize the texture in an image regarding texture. This compilation tries to show the advantages and disadvantages of each of the approaches.

Considering this review of color and texture techniques used in image processing, we will focus on the study of Gabor filters as a tool for texture analysis. Firstly, because of its relationship with human perception, and secondly, it is not limited to analyzing homogeneous textures as other approaches.

Chapter 3

Similarity Measures of Distributions

Résumé

Il existe de nombreuses mesures de similarité qui, selon l'application, n'ont pas toujours un comportement optimal. Ce chapitre présente une analyse qualitative des mesures de similarité les plus utilisées dans la littérature et de la *Earth Mover's Distance* (EMD). L'EMD est une métrique basée sur la théorie du transport optimal avec des propriétés géométriques intéressantes pour comparer les distributions. Cependant, l'utilisation de cette mesure est limitée par rapport à d'autres mesures de similitude. La raison principale était, jusqu'à récemment, la complexité du calcul. Nous montrons la supériorité de l'EMD à travers trois expériences différentes. Premièrement, analyser la réponse des mesures dans le plus simple des cas; distributions synthétiques à une dimension. Deuxièmement, avec deux systèmes de récupération d'images, utilisant des fonctionnalités de couleur et de texture. Enfin, utiliser une technique de réduction dimensionnelle pour une représentation visuelle des textures. Nous montrons qu'aujourd'hui l'EMD est la mesure la plus adaptée de similarité de deux distributions.

Abstract

There are many similarity measures that, depending on the application, do not always have optimal behavior. This chapter presents a qualitative analysis of the similarity measures most used in the literature and the Earth Mover's Distance (EMD). The EMD is a metric based on optimal transport theory with interesting geometrical properties for comparing distributions. However, the use of this measure is limited in comparison

with other similarity measures. The main reason was, until recently, the computational complexity. We show the superiority of the EMD through three different experiments. First, analyzing the response of the measures in the simplest of cases; one-dimension synthetic distributions. Second, with two image retrieval systems, using color and texture features. Finally, using a dimensional reduction technique for a visual representation of the textures. We show that today the EMD is a measure that better reflects the similarity between two distributions.

3.1 Introduction

In image processing and computer vision, the comparison of distributions is a frequently used technique. Some applications where we use these measures are image retrieval, classification, and matching systems [Smeulders et al., 2000]. The distributions could represent low-level features like pixel's intensity level, color, texture, or higher-level features like objects. The comparison could be made using a unique feature, for example, the texture [Banerjee et al., 2018; Kwitt and Uhl, 2008], or combining features in a multidimensional distribution as the fusion of color and texture features [Liu et al., 2017]. In the field of medical imaging, comparing distributions are helpful to achieve image registration [So and Chung, 2017]. More general applications such as object tracking [Klein and Frintrop, 2011; Nejhum et al., 2008] and saliency modeling [Bylinskii et al., 2018] also recur to the distribution comparison. Regarding the number of computer vision applications that employ distributions, it is crucial to choose the right metric to measure the similarity between distributions.

The Earth Mover's Distance (EMD) [Rubner et al., 2000] is a dissimilarity measure inspired by the optimal transport theory. This measure is considered as true distance because it complies with the constraints of non-negativity, symmetry, the identity of indiscernibles, and triangle inequality [Peyré and Cuturi, 2018]. The superiority of the EMD over other measures has been demonstrated in several comparative analyses (see for example [Puzicha et al., 1999; Rubner et al., 2000]). Despite this superiority, in theory, this distance continues to be underused for the benefit of other measures in practice. The main reason is the high computational cost due to its iterative optimization process. However, nowadays, this should not be a problem thanks to the algorithmic advances to computing efficiently the EMD (see "Notes about EMD computation complexity" in section 3.4) and computer processors' progress. Although there are comparative studies (image retrieval scores, for example), in this chapter, we illustrate how other similarity measures dramatically fail even on straightforward tasks. We use a set of 1-d synthetic distributions and two simple image databases (color and texture-based) to compare a set of similarity measures through two image retrieval systems and a visual representation in low-dimensional spaces. Surprisingly, we show that no metric but the EMD yields to classify and give a coherent visual representation

of the images of the databases (see Figs. 3.7 and 3.8 in section 3.3.4). In this chapter, we want to emphasize the importance of having a true metric to measure the similarity between distributions.

In this chapter, we present a new qualitative study of some popular similarity measures. Our primary objective is to show that not all measures express the difference between distributions adequately. Also, we show that today the EMD is a competitive measure concerning computing time. Among the similarity measures that we compare are some of the most used bin-to-bin class methods; the histogram intersection and correlation [Nejhum et al., 2008], the Bhattacharya distance [So and Chung, 2017], the χ^2 statistic and, the Kullback-Leibler divergence [Klein and Frintrop, 2011].

This chapter is organized as follows: in section 3.2, we describe and discuss some properties of the bin-to-bin measures, and we expose the geometrical properties of the EMD. Then, in section 3.3, we show the performance of the different similarity measures with a one-dimensional test, with two image classifiers; one based on color (3-d case) and the other based on texture (2-d case) information and, with a dimensionality reduction using the multidimensional scaling (MDS) technique. Finally, in section 3.4, we close this chapter with some thoughts about EMD and optimal transport in image processing and computer vision.

3.2 Similarity Measures Review

Similarity Measures Notation. In many different science fields, there is a substantial number of ways to define the proximity between distributions. In language abuse, the use of synonyms such as *similarity*, *dissimilarity*, *divergence*, and *distance* complicates the interpretation of such a measure. Here we recall a coherent notation used throughout this chapter.

From the physical point of view, a ***distance*** is defined as a quantitative measurement of how far apart two entities are. Mathematically, a distance is a function $d : M \times M \to \mathbb{R}^+$. We say that d is a ***true distance*** if $\forall (x, y) \in M \times M$ it fulfills the following properties.

1. Non-negativity: $d(x, y) \geq 0$
2. Identity of indiscernibles: $d(x, y) = 0$ if and only if $x = y$
3. Symmetry: $d(x, y) = d(y, x)$
4. Triangle inequality: $d(x, y) \leq d(x, z) + d(z, y)$

From this definition, we can define other distances depending on which properties are (or not) fulfilled. For example, ***pseudo-distances*** do not fulfill the identity of

indiscernibles criterion, ***quasi-distances*** do not satisfy the symmetry property, ***semi-distances*** do not fulfill the triangle inequality condition, and ***divergences*** do not comply with the last two criteria [Khamsi, 2015].

According to the measure, the numerical result could represent the similarity or the dissimilarity between two distributions. The ***similarity*** and the ***dissimilarity*** represent, respectively, how alike or how different two distributions are. Namely, a similarity value is higher when the distributions are more alike, while a dissimilarity value is lower when the distributions are more alike. In this thesis, we use the term ***similarity*** to refer to how similar or dissimilar two distributions are. If distributions are close, they will have high similarity, and if distributions are far, they have low similarity.

3.2.1 Bin-to-Bin Similarity Measures

In computer vision, distributions describe and summarize different features of an image. These distributions are discretized by dividing their underlying support into consecutive and non-overlapping bins p_i to generate histograms. Let $\mathbf{p}$ be a histogram that represents some data distribution. In the histogram, each bin represents the mass of the distribution that falls into a specific range; the bins' values are non-negative reals numbers.

The bin-to-bin measures compare only the corresponding bins of two histograms. Namely, to compare the histograms $\mathbf{p} = \{p_i\}$ and $\mathbf{q} = \{q_i\}$, these techniques only measure the difference between the bins that are in the same interval of the feature space; that is, they only compare bins p_i and q_i $\forall i = \{1, \ldots, N\}$, where i is the histogram bin number and N is the total number of bins. Next, we summarize the bin-to-bin measures we compare.

The **histogram intersection** [Swain and Ballard, 1991] is expressed by a min function that returns the smallest mass of two input bins (see Eq. (3.1)). As a result, the measure gives the number of samples of $\mathbf{q}$ that have corresponding samples in the $\mathbf{p}$ distribution. According to the notation defined at the beginning of section 3.2, the histogram intersection is a dissimilarity measure.

$$d_{\cap}(\mathbf{p}, \mathbf{q}) = 1 - \frac{\sum_i \min(p_i, q_i)}{\sum_i q_i} \tag{3.1}$$

The **histogram correlation** gives a single coefficient which indicates the degree of relationship between two variables. Derived from Pearson's correlation coefficient, this measure is the covariance of the two variables divided by the product of their standard deviations. In Eq. (3.2), $\overline{\mathbf{p}}$ and $\overline{\mathbf{q}}$ are the histogram means. Since this measure is a pseudo-distance (the resulting coefficient is between -1 and 1), it expresses the

distributions' similarity.

$$d_C(\mathbf{p},\mathbf{q}) = \frac{\sum_i (p_i - \overline{\mathbf{p}})(q_i - \overline{\mathbf{q}})}{\sqrt{\sum_i (p_i - \overline{\mathbf{p}})^2 \sum_i (q_i - \overline{\mathbf{q}})^2}} \tag{3.2}$$

The χ^2 **statistic** comes from Pearson's statistical test for comparing discrete probability distributions. The calculation of this measure is relatively straightforward and intuitive. As depicted in Eq. (3.3), the measure is based on the difference between what is actually observed and what would be expected if there was truly no relationship between the distributions. From a practical point of view, it gives the dissimilarity between the two distributions.

$$d_{\chi^2}(\mathbf{p},\mathbf{q}) = \sum_i \frac{(p_i - q_i)^2}{q_i} \tag{3.3}$$

The **Bhattacharyya distance** [Bhattacharyya, 1946] is a pseudo-distance that is closely related to the Bhattacharyya coefficient. This coefficient, represented by $\sum_i \sqrt{p_i q_i}$ in Eq. (3.4), gives a geometric interpretation as the cosine of the angle between the distributions. We normalize the values of this measure between 0 and 1 to express the dissimilarity between the two distributions.

$$d_B(\mathbf{p},\mathbf{q}) = \sqrt{1 - \frac{1}{\sqrt{\overline{\mathbf{p}}\overline{\mathbf{q}} n^2}} \sum_i \sqrt{p_i q_i}} \tag{3.4}$$

The **Kullback-Leibler divergence** [Kullback and Leibler, 1951] measures the difference between two histograms from the information theory perspective. It gives the relative entropy of $\mathbf{p}$ with respect to $\mathbf{q}$ (see Eq. (3.5)). Although this measure is one of the most used to compare two distributions, it is not a true metric since it does not fulfill the symmetry and the triangle inequality properties described in section 3.2.

$$d_{KL}(\mathbf{p},\mathbf{q}) = \sum_i p_i \log \frac{p_i}{q_i} \tag{3.5}$$

We can find other measures in the literature that represent the similarity between distributions, for example, the **Lévy–Prokhorov metric** [Prokhorov, 1956] and the **total variation distance** [Bogachev and Kolesnikov, 2012], which is defined as

$$d_{TV}(\mathbf{p},\mathbf{q}) = \frac{1}{2} \sum_i |p_i - q_i|. \tag{3.6}$$

The Lévy–Prokhorov metric defines the distance between two probability measures on a metric space with its Borel sigma-algebra. However, the use of this metric is not very frequent in the area of computer vision because of the implementation complexity [Bogachev and Kolesnikov, 2012]. On the other hand, the total variation distance equals the optimal transport distance [Cuturi and Avis, 2011] in the simplified setup

when the cost function between bin i and bin j is $c_{ij} = 1,\ \forall i \neq j$ (see section 3.2.2 for definition of cost matrix). For countable sets, it is equal to the L_1 norm. Given these reasons, for the comparative purposes of this chapter, we only take into account the first five bin-to-bin measures defined before.

3.2.2 The Earth Mover's Distance

Earth Mover's Distance is the term used in the image processing community for optimal transport; in other areas, we also find this measure referred to as the Wasserstein distance [Gibbs and Su, 2002] or the Monge-Kantorovich problem [Bogachev and Kolesnikov, 2012; Kantorovich, 2006]. This concept lies in the study of the transportation theory, which aims to optimize transportation and allocation of resources. The main idea behind optimal transport is simple and very natural for the comparison of distributions. Let $\boldsymbol{\alpha} = \sum_{i=1}^{N} \alpha_i \delta_{x_i}$ and $\boldsymbol{\beta} = \sum_{j=1}^{M} \beta_j \delta_{y_j}$ be two discrete measures supported in $\{x_1, \ldots, x_N\} \in \mathcal{X}$ and $\{y_1, \ldots, y_M\} \in \mathcal{Y}$, where α_i and β_j are the weights of the histograms bins $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$; δ_{x_i} and δ_{y_j} are the Dirac functions at position x_i and y_i, respectively. Intuitively, the Dirac function represents a unit of mass that is concentrated at location x_i. This notation is equivalent to the one proposed in [Rubner et al., 2000] where δ_{x_i} is the central value in bin i while α_i represents the number of samples of the distribution that fall in the interval indexed by i.

The key elements to compute the optimal transport are the cost matrix $\mathbf{C} \in \mathbb{R}_+^{N \times M}$, which defines all pairwise costs c_{ij} between points i and j in the discrete measures α and β, and the flow matrix (optimal transport matrix) $\mathbf{F} \in \mathbb{R}_+^{N \times M}$, where f_{ij} describes the amount of mass flowing from bin i (or point x_i) towards bin j (or point x_j). Then the optimal transport problem consists in finding a total flow $\mathbf{F}$ that minimizes the overall cost defined as

$$W_{\mathbf{C}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min\langle \mathbf{C}, \mathbf{F} \rangle = \sum\nolimits_{ij} c_{ij} f_{ij} \tag{3.7}$$

Placing the optimal transport problem in terms of *suppliers* and *consumers*; for a supplier i, the objective is to supply α_i quantity of goods at some location δ_{x_i}. On the other hand, a consumer j, at some location δ_{y_j}, expects to receive at most β_j quantity of goods. Then, the optimal transport problem is subject to three constraints, $\forall i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, M\}$.

1. Mass transportation (positivity constraint): $f_{ij} \geq 0 : i \to j$.
2. Mass conservation (equality constraint): $\sum_j f_{ij} = \alpha_i$ and $\sum_i f_{ij} = \beta_j$.
3. Optimization constraint: $\sum_{ij} f_{ij} = \min\left(\sum_i \alpha_i, \sum_j \beta_j\right)$.

Then, we define the Earth Mover's Distance as the work $W_{\mathbf{C}}$ normalized by the

total flow.

$$d_{EMD}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\sum_{ij} c_{ij} f_{ij}}{\sum_{ij} f_{ij}} \tag{3.8}$$

The importance of the EMD is that it represents the distance between two discrete measures (distributions) in a natural way. Moreover, when we use a *ground distance* as the cost matrix $\mathbf{C}$, the EMD is a true distance. Peyré and Cuturi [2018] show the metric properties of the EMD. In the following section, we developed a series of experiments to show the advantages of EMD.

Ground distance matrix

The EMD finds the best match that minimizes the maximum transport cost; however, the ground distance design can significantly impact the total transportation cost.

Modifying the ground distance to represent the feature space's properties limits the effect that some signatures have on the EMD. The most traditional way to define the ground distance of the EMD is to take it as the Euclidean distance between two points. This configuration is suitable for points that live in a Euclidean feature space, for example, the color pixels of an image in the LAB color space. In such a case, the ground distance of two points (a, b) is defined as

$$d(a, b) = \sqrt{(\Delta L)^2 + (\Delta A)^2 + (\Delta B)^2 + \lambda((\Delta x)^2 + (\Delta y)^2)} \tag{3.9}$$

where the Deltas (Δ) define the difference between the values of each color channel (L, A, B) and the color value location (x, y) in the image space. The parameter λ weights the importance between the spatial information and the color information of the points. A representation of ground distance resulting from Eq. (3.9) is the squared Euclidean distance, which penalizes the further away points in the color space.

On the other hand, the Euclidean distance is not convenient to design the EMD's ground distance between texture signatures. Considering that texture information can be decomposed in M frequencies f and N orientations θ (see Chapter 4), a texture image can be seen as a vector within a hypercylinder in a $M \times N$ dimensional space. We can define then the ground distance as the L1 distance in a linear-polar (lin-polar) space or in a logarithmic-polar (log-polar) space; thus, the ground distance follows the properties of the texture space. In both cases, we map texture orientations on a polar axis, while for the frequencies, we can map them either on a linear axis or on a logarithmic axis. The ground distance between two texture signatures (f_1, θ_1), (f_2, θ_2) is then:

$$d((f_1, \theta_1), (f_2, \theta_2)) = |\Delta f| + \alpha|\Delta\theta| \tag{3.10}$$

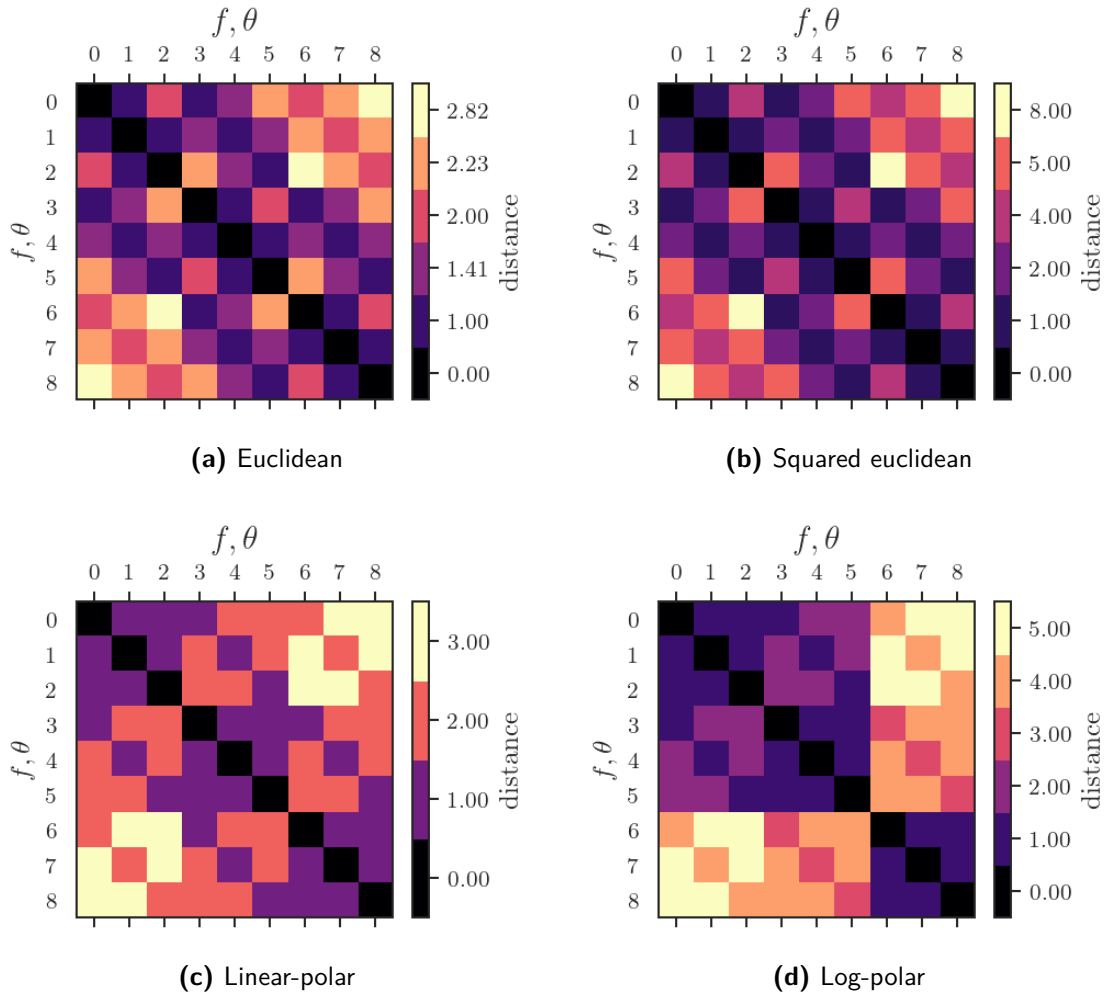
where $\Delta f = \arg(f_1) - \arg(f_2)$ for the linear space and $\Delta f = f_1 - f_2$ for the logarithmic space. Since the polar axis is cylindrical, there are two possible distances between a pair of points; we take the smaller of the two distances such that

$$\Delta\theta = \min(|\theta_1 - \theta_2|, n - |\theta_1 - \theta_2|)$$

Finally, the α parameter controls the relative importance between the frequency and the orientation of textures; we use $\alpha = 1$ in all of our experiments.

To better understand the effect of ground distance on EMD for measuring the similarity between textures, let us consider the following example. We have a filter bank formed by $M = 3$ frequencies separated at a frequency bandwidth $B_f = 1$ (1 octave) and $N = 3$ orientations separated at an angular bandwidth $B_\theta = 60°$. With this configuration, the filter bank contains $m \cdot n = 9$ different Gabor filters; consequently, we have the same number of Gabor responses per image channel. Then, the EMD's ground distance for this filter bank is a symmetric matrix of size $m \cdot n \times m \cdot n$ with zeros on the diagonal. Each cell (i, j) of the matrix represents the distance (L1 or L2) between two frequency and orientation settings (f, θ). For example, the row $i = 0$ represents the comparison between the combination (f_1, θ_1) of the first signature against all the configurations of the second signature.

We illustrate in Fig. 3.1 the ground distance matrix between two texture signatures subjects of $m = 3$ frequencies and $n = 3$ orientations using the four distances described in this section: L2-Euclidean distance, L2-Squared Euclidean distance, L1 lin-polar distance, and L1 log-polar distance. This figure shows how distances in linear-polar or logarithmic-polar space are more convenient with textures and Gabor filters since they consider the cylindrical axis of the angles.



(a) Euclidean **(b)** Squared euclidean

(c) Linear-polar **(d)** Log-polar

Figure 3.1: Visualizations of cost matrix of EMD.

3.3 Comparative Analysis of Similarity Measures

3.3.1 One-Dimensional Case Study

To compare the measures described in section 3.2 in the simplest scenario, we use a set of one-dimensional synthetic distributions. We create a source distribution and a series of target distributions (see Fig. 3.2). Both source and target distributions have 1000 samples and are random normal distributions. The unique difference between them is that the mean of each target distribution (μ) is increasing five units concerning the previous distribution. The first target distribution is indiscernible from the source distribution.

Since the distributions' mean value increases linearly, we expect that the similarity measure has an equivalent response, i.e., that the similarity decreases when the difference of the source and target means increases. In Fig. 3.3, we can see the response of the bin-to-bin measures and the EMD. Among the bin-to-bin measures, those that give a coefficient of dissimilarity (χ^2 statistic, Bhattacharyya pseudo-distance, and K-L

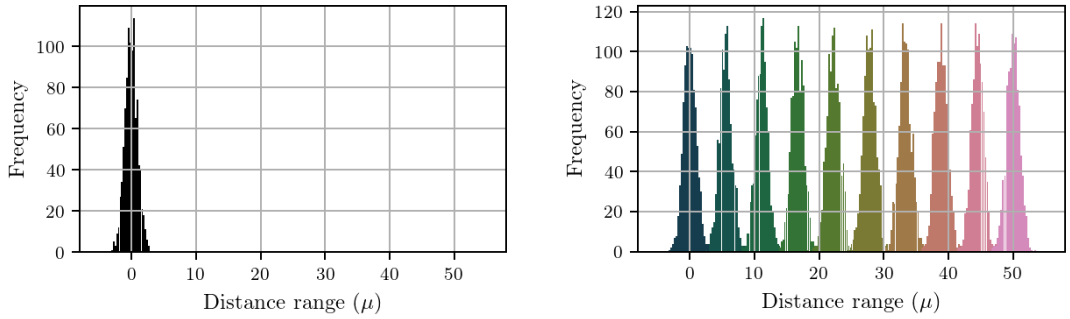
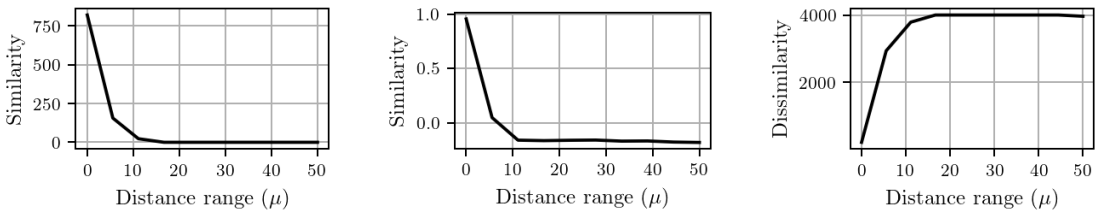
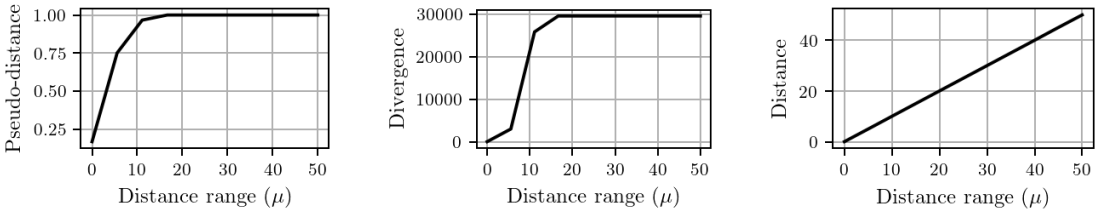


Figure 3.2: Source and target synthetic distributions.



(a) Histogram intersection **(b)** Histogram correlation **(c)** χ^2 Statistic



(d) Bhattacharyya **(e)** Kullback-Leibler **(f)** EMD

Figure 3.3: Distances between the source and target distributions.

divergence) rapidly saturate and stick to a maximum value; while for those that give a coefficient of similarity (histogram correlation and intersection), their value falls rapidly to zero. We can interpret these behaviors as follows. When the bins p_i, q_i do not have any mass in common, the bin-to-bin measures fail to consider the mutual distance of the bins. They could consider that the distributions are precisely at the same distance (there is no difference between them), or that the distributions are entirely dissimilar. The only measure that presents a convenient behavior with the increasing difference of the means is the EMD. This is due to taking into account the *ground distance* $\mathbf{C}$ of the matching bins (see above, Eq. (3.8)). One can argue that for applications like image retrieval finding the most similar distribution is sufficient to find the alike image or texture, whereas the ordering on the other is irrelevant. In the following experiment, we show that this intuition is incorrect and that even in an overly simple case, the bin-to-bin measures are not the best choice.

3.3.2 Image Retrieval Systems

We develop two image retrieval systems as a second comparison test, the first based on color information and the second based on texture information. For the classifiers, we use different databases. The first one contains 24 different color images of superhero toys[1]. It has 12 classes with two samples per class. The first two rows in Fig. 3.4 show some examples of the superhero toys and their variations (change of the angle of the toy or the addition of accessories). The second database [Kylberg, 2011] comprises images belonging to different surfaces and materials (see the last two rows in Fig. 3.4). The database contains 28 different classes; it contains different patches per class.

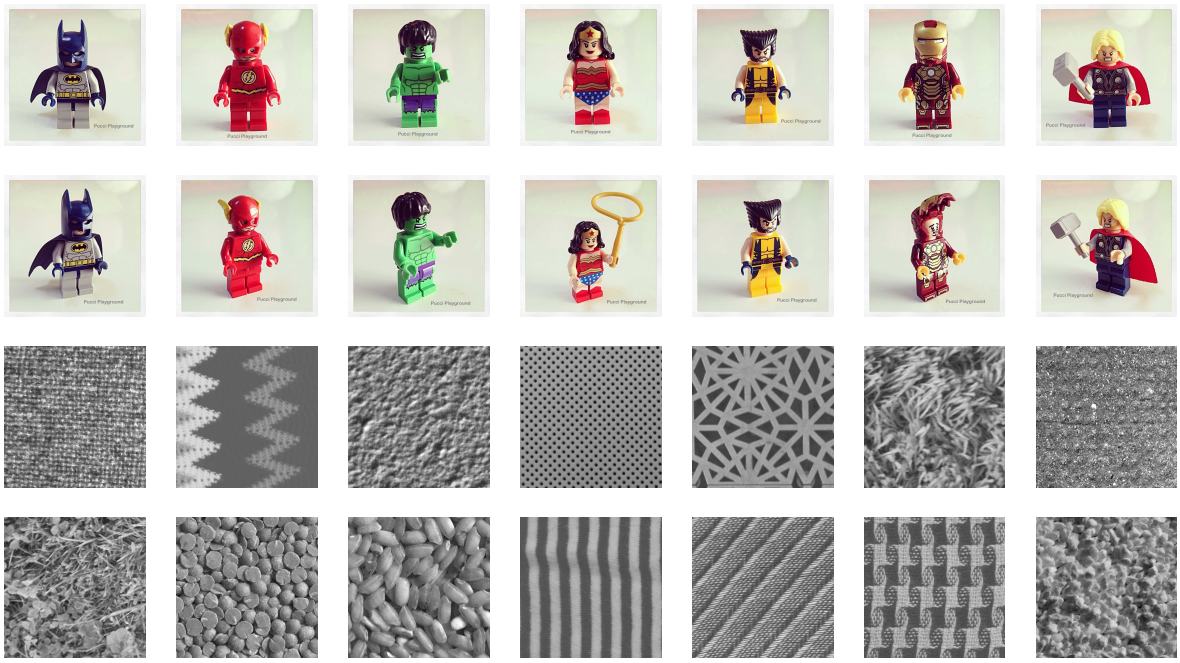


Figure 3.4: Some samples from the color and texture databases.

We compare the performance of six out of the eight measures described in section 3.2 in the following way. First, we divide the database samples into *model images* and *query images*; each class only has one model and one query image. We select an image from the query set and compare its color/texture distribution (source distribution) with all model images' color/texture distribution (target distributions). Then, we order the images from the most similar to the most dissimilar image. We repeat this process for all the images in the query set[2].

Image color distribution. We use 3-d histograms to represent the distribution of color pixels. Since the superhero images are very simple and do not present significant challenges, i.e., the images possess a very distinctive color palette and do not present textures or important changes in lighting, any similarity measure should be sufficient

[1] CC super hero's images courtesy of Christopher Chong on Flickr.

[2] The image classification systems (color-based and texture-based) and the datasets used for this chapter are available at https://github.com/CVMethods/image_clasifier.git

to perform the image retrieval. However, image retrieval systems are sensitive to the representation and quantification of the color image pixels. We show this effect by varying the color space and the color quantization level in the image classifier. We represent the images in the RGB, HSL, and LAB color spaces for the color space. We represent the color space in histograms of 8, 16, and 32 bins per channel for the color quantization level.

Image texture distribution. We use a family of Gabor filters as texture descriptors to obtain a distribution (2-d histograms) that models the images' texture. This type of filter models the human visual cortex's behavior [Daugman, 1988], so they are very useful in computer vision applications to represent textures [Lee, 1996]. The mother wavelet

$$g(x, y, \omega, \theta) = \frac{\omega^2}{4\pi\kappa^2} \mathsf{e}^{-\frac{\omega^2}{8\kappa^2}\left(4\hat{x}^2+\hat{y}^2\right)} \cdot \left[\mathsf{e}^{i\kappa\hat{x}} - \mathsf{e}^{\frac{\kappa^2}{2}}\right] \tag{3.11}$$

represents the family of Gabor descriptors, where $\hat{x} = x\cos\theta + y\sin\theta$ and $\hat{y} = -x\sin\theta + y\cos\theta$. The 2-d texture histograms are a function of the Gabor responses' energies according to the frequency ω, the orientation θ and the constant κ in the Gabor wavelet. The 2-d texture histograms are a function of the Gabor responses' energies according to the frequency $\kappa \approx \pi$ and six bins means that there are six different frequencies to a bandwidth of one octave and six different orientations. The energy of a Gabor response is given by

$$E_{\omega,\theta} = \sum\nolimits_{x,y} |W_{x,y,\omega,\theta}|^2, \tag{3.12}$$

where $W_{x,y,\omega,\theta}$ is the response of the convolution of a 2-d wavelet with frequency ω and orientation θ with an image.

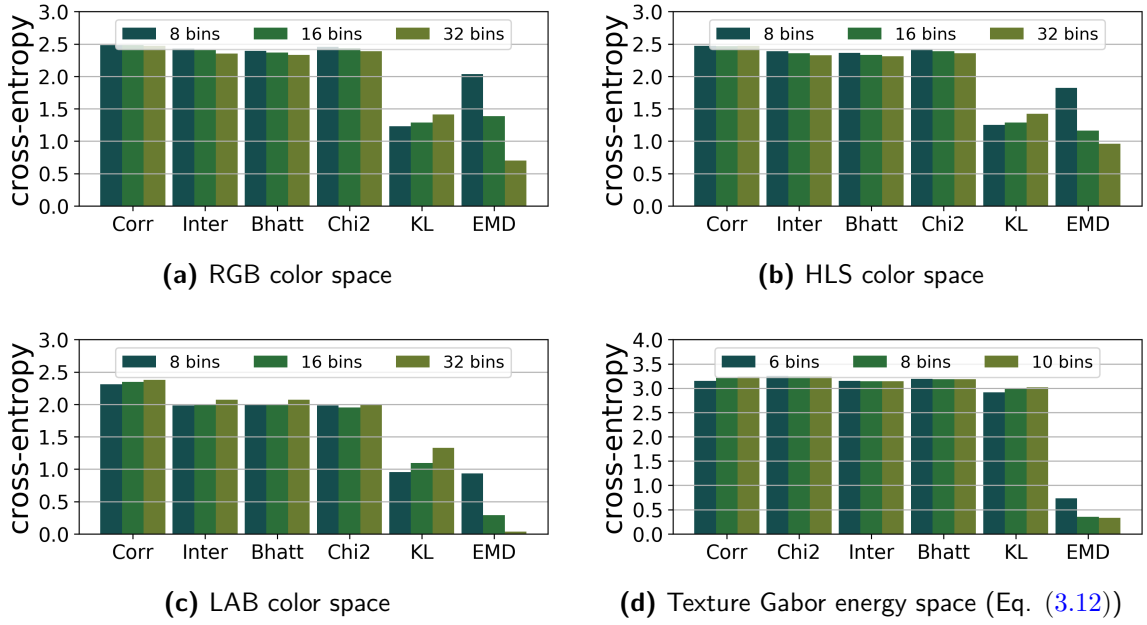
Image Retrieval Systems Evaluation

We create a comparison benchmark for the six similarity measures. First, we normalize the distances given by the different methods between 0 and 1, where a value close to 0 means the most similar distribution to the source distribution and 1 the most dissimilar. Then we transform the normalized distances into probabilities using a softmax function $SM(\mathbf{d}) = \frac{\mathsf{e}^{\mathbf{d}}}{\sum_i \mathsf{e}^{d_i}}$. The vector $\mathbf{d} = \{d_i\}$, $\forall i = \{1, \ldots, M\}$, represents the distances between the query image to the M images in the database. Considering the softmax function of the distances vector as a classification probability $SM(\mathbf{d}) = \hat{\mathbf{y}}$, we compute the cross-entropy [Bishop, 2006] considering the classification ground truth $\mathbf{y}$ such as,

$$H(\mathbf{y}, \hat{\mathbf{y}}) = -\sum\nolimits_i y_i \log \hat{y}_i \tag{3.13}$$

where y_i is the ground truth probability vector, and $\hat{y}_i$ the probability vector of the prediction.

We can interpret the cross-entropy value as the image classifier's confidence level for some given metric, feature space (color or texture), and histogram size. When this value is very close to zero, it indicates a perfect classification of the query image. In Fig. 3.5, we note the superiority of the EMD over the other measures in both color and texture-based classifiers.



(a) RGB color space

(b) HLS color space

(c) LAB color space

(d) Texture Gabor energy space (Eq. (3.12))

Figure 3.5: Cross entropy value of image retrieval systems (color and texture) using different similarity measures (lower is better).

With the image retrieval systems, we can highlight interesting aspects of the EMD and the use of bin-to-bin measures in the comparison of distributions. First, we see the importance of selecting the color space and the compression level of the feature space (histogram size). The effect of discretization in the bin-to-bin measures is counter-intuitive because the error increases slightly when the number of bins increases. The explanation could be a poorer intersection of mass distributions. In the case of EMD, increasing the number of bins improves the classification result. Besides, as we expected, in the color-based classifier, the calculation of the EMD using the LAB color space performs better than with the HLS or the RGB. This effect is because the LAB color space models the color human perception in the Euclidean space; therefore, the *ground distance* between two colors is easily calculated with the L_2 norm. On the other hand, in the texture-based classifier, increasing the number of bins beyond eight bins does not improve the classification considerably. This behavior occurs because the histograms with eight frequencies and eight orientations represent sufficiently well the image textures.

3.3.3 Texture Projection Quality Evaluation

We use the multidimensional scaling (MDS) technique [Kruskal, 1964] as the last evaluation test for the similarity measures on the texture dataset. The MDS allows to geometrically represent n textures using a set of n points $\{x_1, \ldots, x_n\}$ in a reduced Euclidean space $\mathbb{R}^d$ so that the distances between the points $\hat{d}_{ij} = ||x_i - x_j||_2$ correspond as much as possible to the values of dissimilarity d_{ij} between the texture distributions. To evaluate the quality of the projection, we use the stress value proposed in [Kruskal, 1964].

$$S = \sqrt{\frac{\sum_{ij}(\hat{d}_{ij} - d_{ij})^2}{\sum_{ij} \hat{d}_{ij}^2}} \tag{3.14}$$

The stress coefficient is a positive value that indicates how well the distances given by the measures are preserved in the new low-dimensional space, i.e., the lower the level of S, the better the representation of texture in a low-dimensional space (2-d in our case). Fig. 3.6 shows how the lowest stress is obtained using the EMD. This result is because the MDS technique interprets the distances of the entrance towards distances in a low dimension space. Given that the EMD is the only measure that is a true metric, not only is the stress level low, but the visual projection is following the frequency ω and orientation θ used in the Gabor filters (Eq. (3.11)) that model the distribution of the textures (see Fig. 3.14).

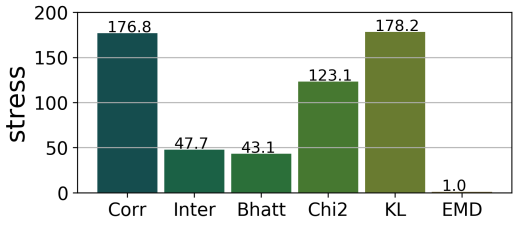


Figure 3.6: Stress value of the MDS projections using the six principal similarity measures.

3.3.4 Color and Texture Retrieved Images: Some Cases of Study

Color-based Image Retrieval

The Figs. 3.7 and 3.8 present the result of the classification of the images of two superhero toys. The query image is displayed in the upper left. The rows of the image arrangement represent the different measurements used in the retrieval, while the columns show the most similar image from left to right in descending order. The numerical values of the distances are below each image; these values are not normalized,

nor are they on the same scale. Notice that some values are sorted in decreasing whereas some others in increasing order. The various distances are increasing, the similarity measures (correlation, intersection) are decreasing.

The two examples here show how, under certain disturbances in color distribution, bin-to-bin measurements cannot identify the correct result. For the Wonderwoman toy, the fact that the query image has an extra accessory modifies the color signature of the image, while in the case of the superman toy, the color signature is very close to those toys that contain red and blue colors. These two images were obtained using the LAB color space and 32 bins for the pixels' color distribution.

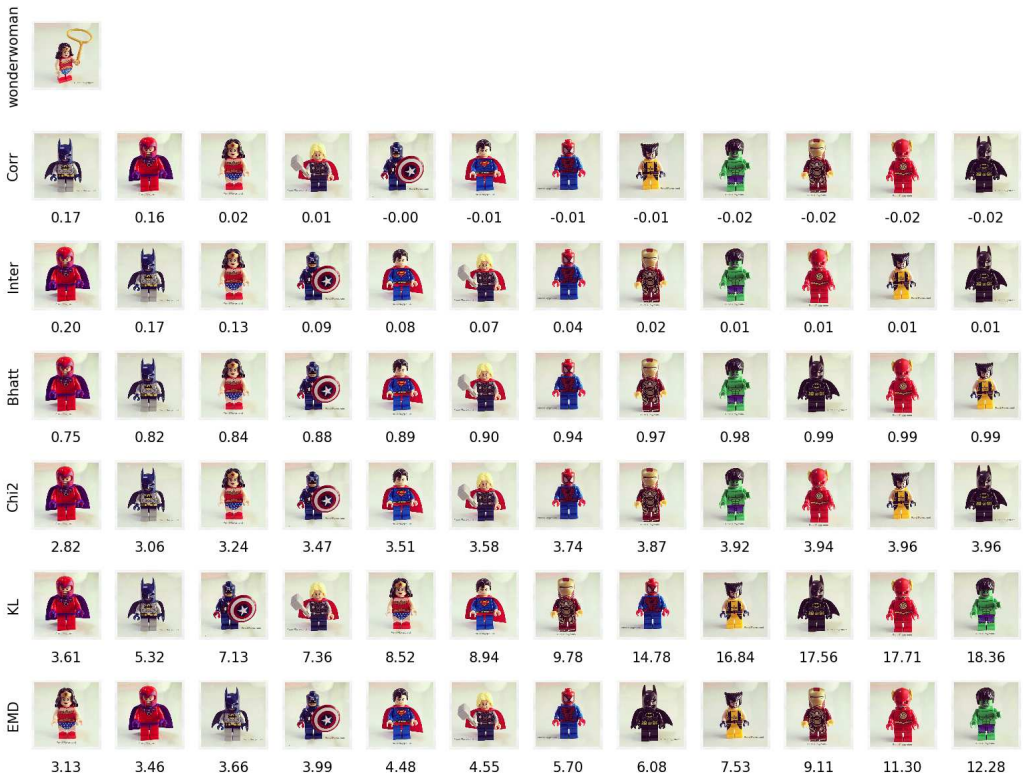


Figure 3.7: Wonderwoman toy image retrieval example

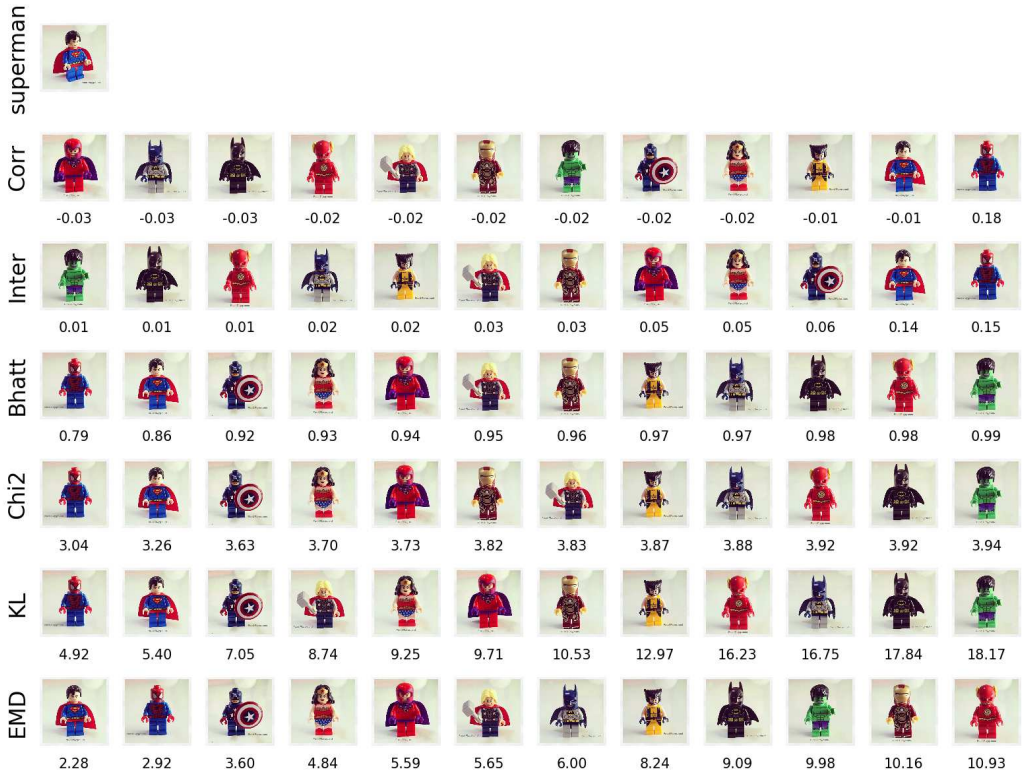


Figure 3.8: Superman toy image retrieval example

Texture Projection Visual Evaluation

The following images serve as a visual tool for the comparative evaluation of the different measures analyzed in this chapter. As we described in section 3.3.3, the MDS technique allows to projecting the textures in a low dimensional space using the distances given by the similarity measures. This representation is carried out in a two-dimensional Euclidean space in our case. In the figures, we can notice that the MDS technique has problems representing the textures coherently when the input measures are not a metric, i.e., for the bin-to-bin measures. The axis in Figs. 3.9 to 3.13 do not correspond with the input space of the textures. According to the coarseness, we can speculate that there is a tendency to rank the textures in some arbitrary direction.

However, in the case of EMD, we can observe that since this measure uses a ground distance to calculate the similarity, we can define the cost matrix $\mathbf{C}_{ij} = c(x_i, y_j)$ of Eq. 3.7 to be the L1-distance as

$$c(x_i, y_i) = d((\omega_i, \theta_i), (\omega_j, \theta_j)) = \alpha|\Delta\omega| + |\Delta\theta| \tag{3.15}$$

where $|\Delta\omega| = \omega_i - \omega_j$, $\Delta\theta = min(|\theta_i - \theta_j|, \theta_{max} - |\theta_i - \theta_j|)$, and α is a constant that regulates the importance between the orientation and the coarseness of textures. In such a way that it represents the 2-d texture distributions into a log-polar space. We can distinguish this effect in Fig. 3.14 where we can see how the orientation and

the frequency organize the textures into a log-polar coordinate space forming a circle. The texture orientation is represented along the circular axis; on the other hand, the texture frequency follows the axis going from the outside to the circle center. The lower frequency images remain at the edge of the circle, and those with high frequency (and low directionality) are grouped in the center. This behavior is not observed with any of the other measures and is reflected in the stress value of Fig. 3.6.

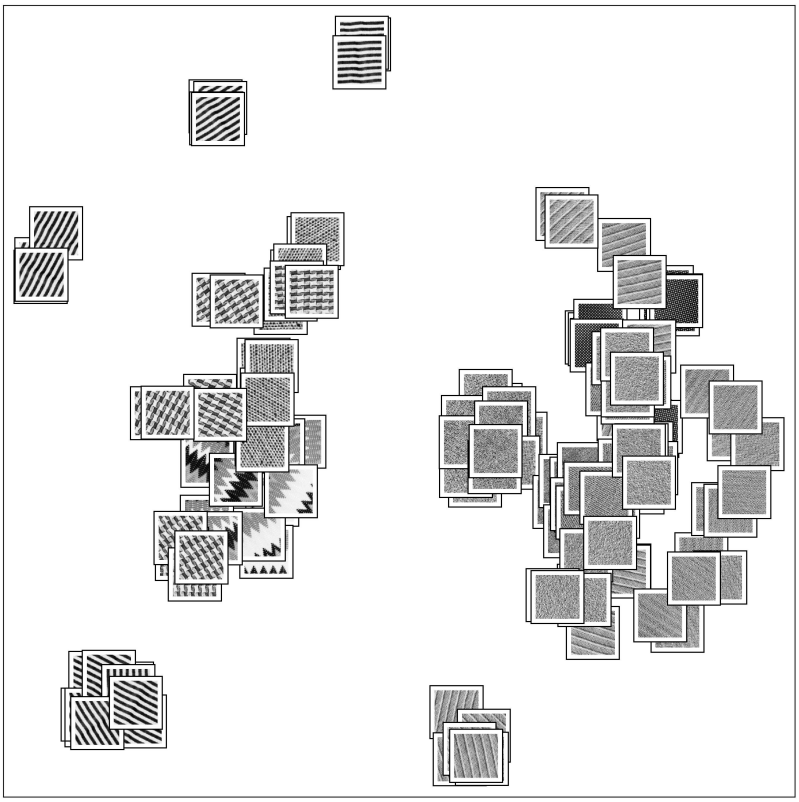


Figure 3.9: MDS texture projection using the histogram intersection

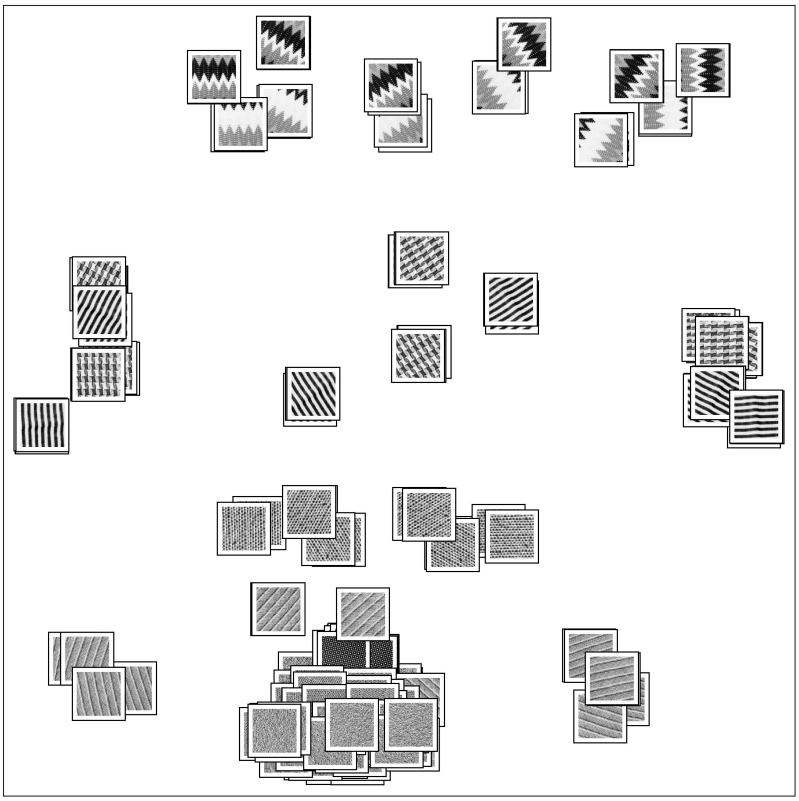


Figure 3.10: MDS texture projection using the histogram correlation

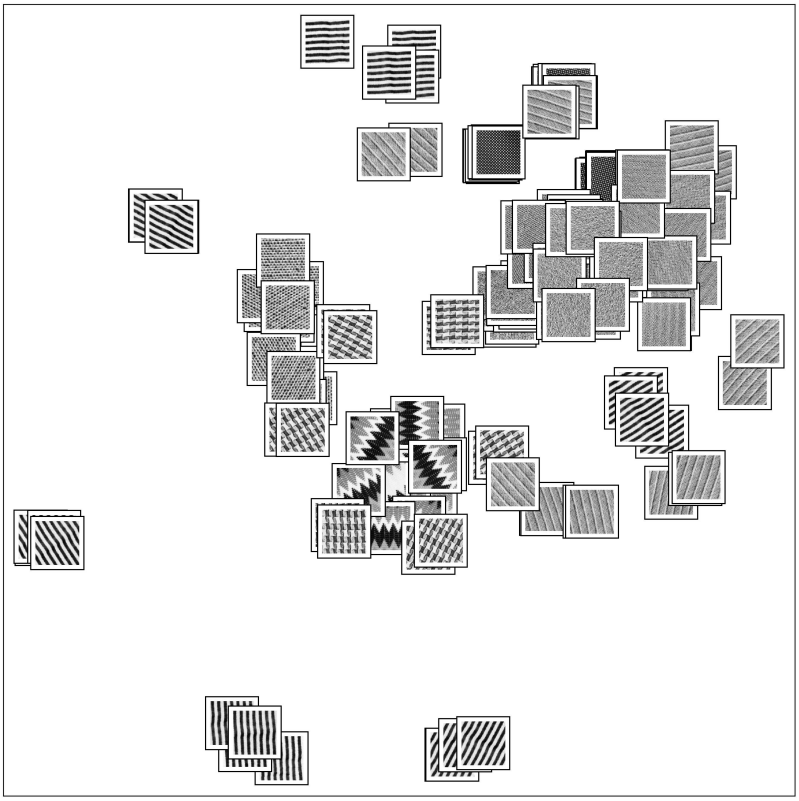


Figure 3.11: MDS texture projection using the χ^2 statistic

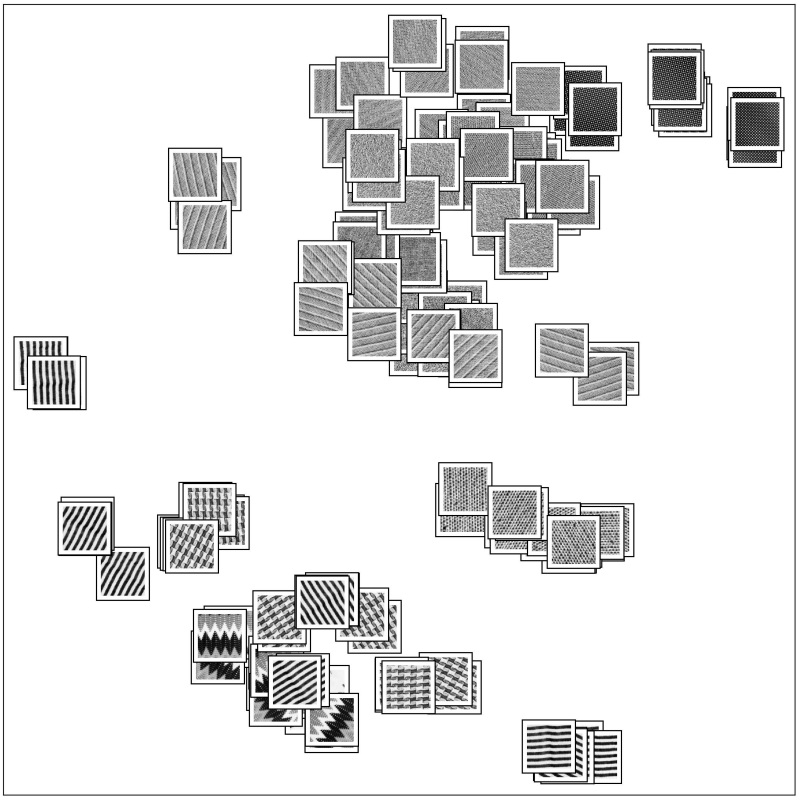


Figure 3.12: MDS texture projection using the Bhattacharyya distance

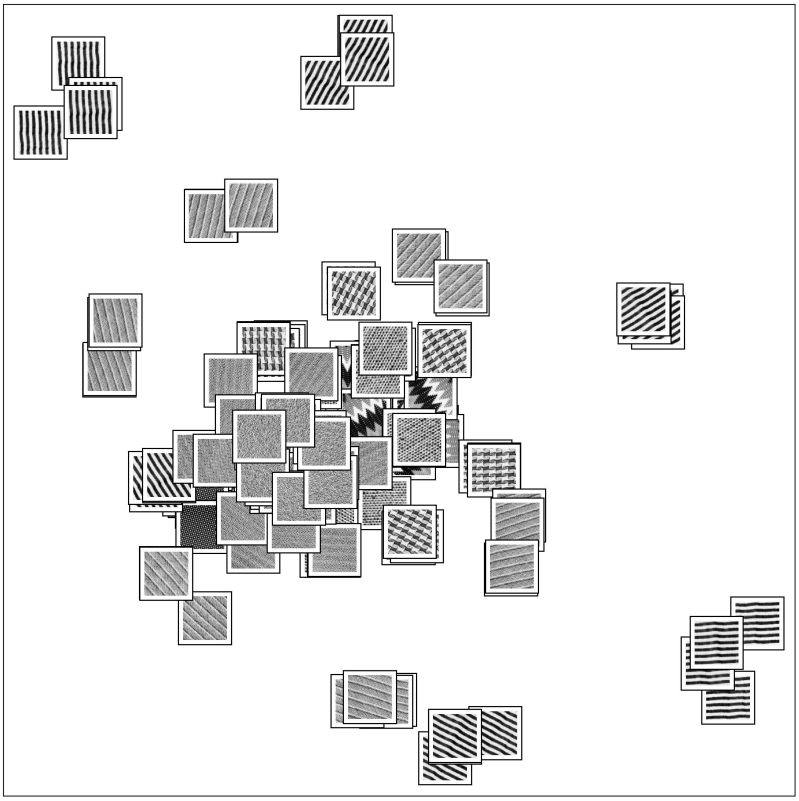


Figure 3.13: MDS texture projection using the Kullback-Leibler divergence

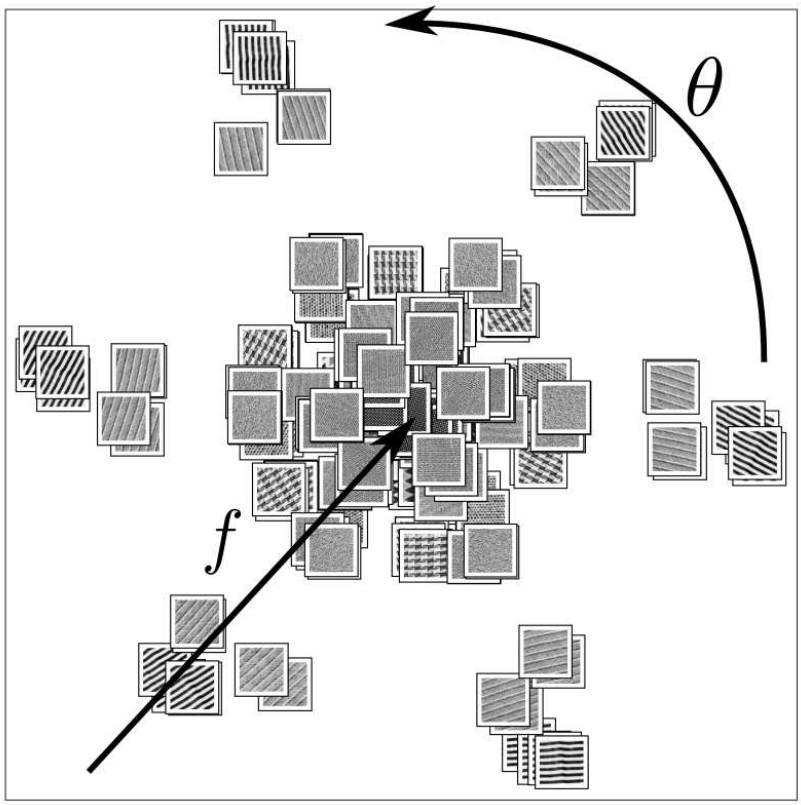


Figure 3.14: MDS texture projection using the Earth Mover's Distance

3.4 Conclusion

In this chapter, we compare some of the few popular, bin-to-bin similarity measures with the EMD. We measure their performance in three tests: a one-dimensional analysis with synthetic distributions, two image classifiers (color and texture-based), and a visual projection using the MDS technique and the stress as the comparison value. The objective is to show that such measures highly used in the literature to develop complex tasks are not the best choice since they fail even in the most straightforward conditions. We illustrate that the EMD is a true metric [Peyré and Cuturi, 2018] that naturally expresses dissimilarity between distributions.

Results. The experiments of the previous sections show the superiority of the EMD to represent the similarity between distributions. First, the one-dimensional case shows how the bin-to-bin measures saturate (or fall to zero) as soon as the probabilities have an empty intersection (see Fig. 3.2). As for the image retrieval systems, we can see that by correctly choosing the feature image space and a good compression resolution of the distributions (LAB color space with 32 bins in the color-based system and the Gabor energy with eight bins in the texture-based system), the EMD performs the best classification result. However, this is not the case with the other measures because they are not a true distance. Representing the textures in the Euclidean space using the MDS technique shows another advantage of the EMD. The use of the ground distance $\mathbf{C}$ in the optimal transport calculation makes it possible to transfer 2-d texture histograms to a logarithmic-polar space, making the stress value relatively low.

Notes about EMD computation complexity. We believe that EMD is a depreciated metric only because of its excessive calculation time. In the examples developed before, we calculate the EMD using the iterative process of linear programming. Despite this, the calculation is fast enough to develop the image classifier. In comparison with the first EMD algorithm [Rubner et al., 2000], the computer processors' progress allows to use the same algorithm and be competitive with the bin-to-bin measures. Moreover, a solution to the excessive complexity time and memory consumption is the regularized distances, also called Sinkhorn distances [Cuturi, 2013]. This entropy-based regularization accelerates the computing time, giving a close approximation of the EMD. The regularization of distances allows for creating parallelizable algorithms.

Chapter 4

Spectral Image Decomposition

Résumé

Ce chapitre utilise la théorie du signal et la transformée de Fourier pour générer une famille optimisée de filtres de Gabor. Le but est de récupérer autant d'informations que possible sur des textures d'une image dans le domaine fréquentiel sans affecter la localisation des informations. Nous présentons une analyse de la fonction de Gabor conforme au principe d'incertitude de Heisenberg. La méthodologie présentée dans ce chapitre générera des banques de filtres Gabor entièrement personnalisées.

Abstract

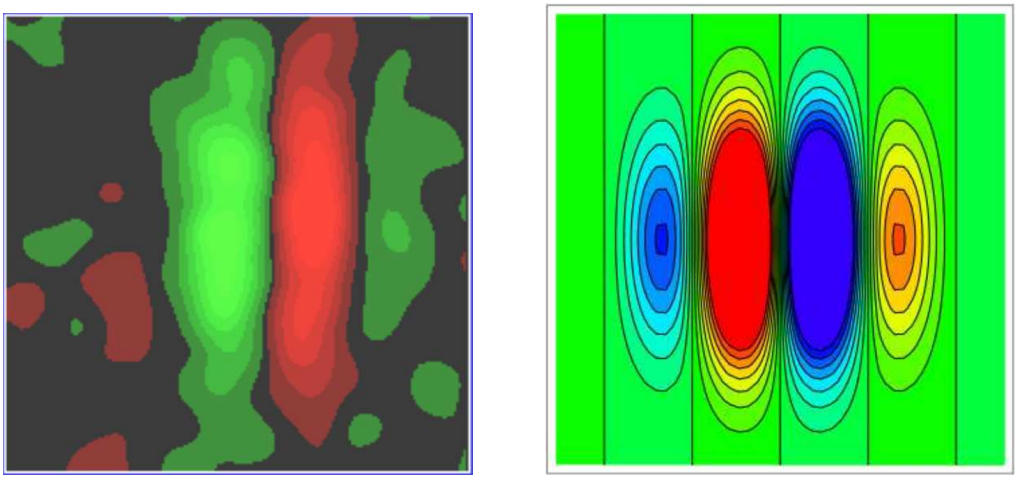
This chapter uses signal theory and Fourier transform to generate an optimized family of Gabor filters. The goal is to retrieve as much information as possible about textures from an image in the frequency domain without affecting the information location. We present an analysis of the Gabor function that complies with the Heisenberg uncertainty principle. The methodology presented in this chapter will generate fully customized Gabor filter banks.

4.1 Introduction

The study and understanding of human vision have contributed to computer vision. Through neuropsychology, mathematical interpretations of the visual system have been developed, particularly the first area of the primary visual cortex, the so-called V1.

Novel experimental techniques [DeAngelis et al., 1995] have made it possible to observe the activity of V1 and the modules involved in visual processes. It is known

that the Receptive Field (RF) is in charge of formatting the optical signals for their interpretation. Physically, RF is the area of a visual neuron that responds to specific light stimuli. The RF response is known as the Receptive Profile (RP), and it can be positive and exciting or negative and exciting. Fig. 4.1a shows the level curves of a neuron's receptor profile, showing positive responses in green and negative responses in red. Mathematically, the RP is a function $\varphi : D \to \mathbb{R}$ defined in the RF domain D that measures the neuron's response $\varphi(x, y)$ (as positive or negative) to the stimulations at the point (x, y) [Petitot, 2008]. The transfer function of a neuron $\varphi(x, y)$ can be considered as a filter, e.g., the Gabor filter, that correctly replicates the behavior of the V1 receptive field (see Fig. 4.1b).



(a) Experimental level curves of RP **(b)** Gabor model of a RP

Figure 4.1: Receptive field of a simple visual neuron. Images from [Petitot, 2008].

In chapter 3 we use the Gabor filter to extract global energy from homogeneous, i.e., stationary textures. This energy serves as the characteristic signature of the texture present in the image. In this chapter, motivated by its relationship with the human perception process, we delve into the study of Gabor filters. In particular, we are interested in its space-frequency properties to extract local texture features in natural images.

We carry out an analysis of the Gabor filters first from the point of view of signal theory to expose their properties and limits. Then, starting from the representation of the filters in 1-d, we propose a formulation of the filters that allow us to fully customize a family of 2-d filters depending on the application. The reformulation of Gabor filters allows dealing and take advantage of the aliasing and of the DC[1] component.

Using a Gabor filter family, we generate a feature bank that contains local information about the texture. As natural images contain color information, the proposed framework allows obtaining local texture features taking into account the luminance

[1]The term originates in electronics, where DC refers to a direct current voltage

and the chrominance of an image. This strategy's novelty is that we consider color as a complex signal where we can measure the space-frequency distribution of a color texture.

4.2 The Gabor Filter as a Measurement Tool

This section presents a reminder of the signal theory applied to image processing for feature extraction. First, we show the reasons for restricting signal analysis in two predefined domains: time and frequency (for the 1-d case) and space and frequency (for the 2-d case). We especially recall Gabor filter theory and properties by showing how these filters are related to Heisenberg's uncertainty principle.

As we mentioned before, we use Gabor filters as a tool to measure the information that characterizes a signal. Since the signal carries relevant information at different points and scales (either in the spatial or space-frequency domain), a well-known strategy used in the literature is to create a filter bank that covers most of the spectrum to be able to reconstruct the original signal.

4.2.1 Signals in Two Domains

Bound by the Fourier transform, there are two equivalent representations of a signal (one-dimensional (1-d) or two-dimensional (2-d)). The first represents the signal as a function of time, while the second represents it as a function of frequency. These two representations carry the same information but in different ways; besides, we can go from one to another via the Fourier transform (or the inverse Fourier transform), making these special interest descriptions. We define the pair of 1-d Fourier transforms (FT) as follows

$$\begin{aligned} H(f) &= \mathcal{F}\{h(t)\} = \int_{-\infty}^{\infty} h(t)\mathsf{e}^{-\mathsf{j}2\pi ft}dt, \\ h(t) &= \mathcal{F}^{-1}\{H(f)\} = \int_{-\infty}^{\infty} H(f)\mathsf{e}^{\mathsf{j}2\pi ft}df, \end{aligned} \tag{4.1}$$

while for the 2-d case, we define the FT as

$$\begin{aligned} H(u,v) &= \mathcal{F}\{h(x,y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y)\mathsf{e}^{-\mathsf{j}2\pi(ux+vy)}dxdy, \\ h(x,y) &= \mathcal{F}^{-1}\{H(u,v)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(u,v)\mathsf{e}^{\mathsf{j}2\pi(ux+vy)}dudv. \end{aligned} \tag{4.2}$$

It is evident that the function $h(t)$ (resp. $h(x,y)$) is located in both domains; however, it is also well known that no signal with compact support can have a finite Fourier transform and vice versa [Bracewell, 1999], there is a particular uncertainty in the time and frequency locations of $h(t)$ (resp. space and frequency locations of $h(x,y)$). We developed and demonstrated the principle that defines this uncertainty (for signal processing and image processing) in the following section.

4.2.2 The Uncertainty Principle in Image Processing

The uncertainty principle is one of the most famous ideas in quantum mechanics. An early incarnation of the uncertainty principle appeared in a 1927 paper by the German physicist Heisenberg. The uncertainty principle says that we cannot measure the position (x) and momentum (p) of a particle with absolute precision. The more accurately we know one of these values, the less accurately we know the other.

However, quantum mechanics' uncertainty principle is just a particular case of a more general compromise in simple everyday life phenomena involving waves. The central idea is connected with the interrelation between frequency and duration. For example, in the case of sound waves, if we want to identify the frequency of a musical note, the shorter the sound lasts in time, the less specific we can be about the exact frequency of the sound to find, it would be necessary to listen to the sound for a longer time, in which case the locality measure loses its sense. In the language of signal processing, we can say that a short signal correlates highly with a wide range of frequencies, and only wide signals correlate with a short range of frequencies. Formally this is expressed as

$$\Delta t \Delta \omega \geq \frac{1}{2}, \tag{4.3}$$

where Δt is the duration of the signal in the time domain and $\Delta\omega$ is the bandwidth of the signal in the frequency domain [Petrou and Sevilla, 2006]. The uncertainty principle then states: the spectral bandwidth product multiplied with the signal's time duration cannot be less than a particular minimum value. Considering the signal bandwidth in terms of frequency as $\Delta\upsilon$ where $\omega = 2\pi\upsilon$, the uncertainty principle is stated as

$$\Delta t \Delta \upsilon \geq \frac{1}{4\pi}. \tag{4.4}$$

The Heisenberg uncertainty principle can be mathematically proved in signal processing and image processing by the Parseval's identity, where **Parseval's theorem** states that

$$\int_{-\infty}^{\infty} h(t)^2 dt = \int_{-\infty}^{\infty} |H(\upsilon)|^2 d\upsilon, \tag{4.5}$$

where $h(t)$ is a function and $H(\upsilon)$ its the Fourier transform.

The **energy content** of the signal described by $h(t)$ is defined as:

$$E_\infty \equiv \int_{-\infty}^{\infty} h(t)^2 dt. \tag{4.6}$$

From the Parseval's identity this may be written as:

$$E_\infty = \int_{-\infty}^{\infty} |H(\upsilon)|^2 d\upsilon. \tag{4.7}$$

By setting $\Delta t = t - t_0$, the **time dispersion** of the signal is given by

$$(\Delta t)^2 \equiv \frac{1}{E_\infty} \int_{-\infty}^{\infty} (t - t_0)^2 h(t)^2 dt, \tag{4.8}$$

where t_0 is the **center of gravity** of the signal defined by

$$t_0 \equiv \frac{1}{E_\infty} \int_{-\infty}^{\infty} t h(t)^2 dt, \tag{4.9}$$

and where if we shift the origin of t so that $t_0 = 0$, then

$$(\Delta t)^2 = \frac{1}{E_\infty} \int_{-\infty}^{\infty} t^2 h(t)^2 dt. \tag{4.10}$$

In an analogous way for $\Delta \upsilon = \upsilon - f$, the **spectral bandwidth** of the signal is given by

$$(\Delta \upsilon)^2 \equiv \frac{1}{E_\infty} \int_{-\infty}^{\infty} (\upsilon - f)^2 |H(\upsilon)|^2 d\upsilon, \tag{4.11}$$

where f is the **spectral center of gravity** of the signal defined by

$$f \equiv \frac{2\pi}{E_\infty} \int_{-\infty}^{\infty} f |H(\upsilon)|^2 d\upsilon, \tag{4.12}$$

and if we consider $f = 0$, then Eq. (4.11) becomes

$$(\Delta \upsilon)^2 = \frac{1}{E_\infty} \int_{-\infty}^{\infty} f^2 |H(\upsilon)|^2 d\upsilon. \tag{4.13}$$

If $h'(t)$ is the derivative of the function, its Fourier transform is $\mathrm{j}2\pi f H(\upsilon)$. By applying the Parseval's identity (using the left and right terms in Eq. (4.5)) to the Fourier pair $h'(t) \longleftrightarrow \mathrm{j}2\pi f H(\upsilon)$ we obtain

$$4\pi^2 \int_{-\infty}^{\infty} f^2 |H(\upsilon)|^2 d\upsilon = \int_{-\infty}^{\infty} h'(t)^2 dt. \tag{4.14}$$

By substituting it in Eq. (4.13), we have:

$$(\Delta \upsilon)^2 = \frac{1}{4\pi^2 E_\infty} \int_{-\infty}^{\infty} h'(t)^2 dt. \tag{4.15}$$

We use Eqs. (4.10) and (4.15) to calculate:

$$(\Delta t)^2 (\Delta \upsilon)^2 = \frac{1}{4\pi^2 E_\infty^2} \int_{-\infty}^{\infty} t^2 h(t)^2 dt \int_{-\infty}^{\infty} h'(t)^2 dt \tag{4.16}$$

Applying the Schwartz's inequality for the integrals on the right-hand side of Eq. (4.16) we obtain

$$\int_{-\infty}^{\infty} t h(t)^2 dt \int_{-\infty}^{\infty} h'(t)^2 dt \geq \left| \int_{-\infty}^{\infty} t h(t) h'(t)^2 dt \right|^2. \tag{4.17}$$

We may integrate by parts the integral on the right-hand side of Eq. (4.17) such as

$$\int_{-\infty}^{\infty} th(t)h'(t)^2 dt = \frac{1}{2} th(t)^2 \Big|_{-\infty}^{\infty} - \frac{1}{2}\int_{-\infty}^{\infty} h(t)^2 dt. \tag{4.18}$$

If $\lim_{t\to\infty} th(t)^2 = 0$, the first term on the right-hand side of (4.18) vanishes and from Eq. (4.6) we have

$$\int_{-\infty}^{\infty} th(t)h'(t)dt = -\frac{1}{2}E_{\infty}. \tag{4.19}$$

If we use this into Eq. (4.17) and then into Eq. (4.16) we obtain

$$(\Delta t)^2(\Delta \upsilon)^2 \geq \frac{1}{16\pi^2}. \tag{4.20}$$

This is the mathematical statement of the uncertainty principle in signal processing [Petrou and Sevilla, 2006].

4.2.3 1-d Gabor Filters

The uncertainty principle shows that time and frequency are two fundamental domains and physically measurable quantities, but still idealizations if we consider one from the other's perspective. Frequency is a simple waveform in the time domain, but to be sharply defined in the frequency domain, it must be infinite in the time domain, i.e., a waveform that has always existed and will remain forever. In everyday life, it is complicated to find phenomena with these characteristics; it is more common to find signals that have properties from both domains; certainly, they have some frequency characteristics, but they also have a starting point, and after some time, these signals begin to fade away. This phenomenon motivated Dennis Gabor to represent signals simultaneously in time and frequency through the Gabor Elementary Function (GEF) [Gabor, 1946]. The function represents the minimal quantum of information, i.e., the minimal amount of simultaneous information in time and frequency. In other words, it occupies the minimal area, given by a rectangle, in the time-frequency plane.

The Gabor function is derived form the uncertainty principle, therefore, it has a shape for which the product $\Delta t \Delta \upsilon$ assumes the smallest possible value. In other words, the Gabor function is the one that transforms inequality of Eq. (4.4) into the equality $\Delta t \Delta \upsilon = \frac{1}{4\pi}$. Then, the Gabor function is defined as the modulation product of a harmonic oscillation (a sinusoidal wave) of any frequency with a pulse of the form of a probability function (a Gaussian function) [Gabor, 1946] and is represented as

$$g(t) = \mathrm{e}^{-\alpha^2(t-t_0)^2}\mathrm{e}^{\mathrm{j}2\pi f t+\phi} \tag{4.21}$$

where α express the *spread* and t_0 denotes the centroid of the Gaussian function, f is the frequency of the sinusoidal wave, and ϕ defines the phase shift of the oscillation.

The Fourier transform of Eq. (4.21) defines the representation of the Gabor function in the frequency domain $G(\upsilon) = \mathcal{F}\{g(t)\}$ with the following analytical form

$$G(\upsilon) = \sqrt{\frac{\pi}{\alpha^2}} \mathrm{e}^{-\left(\frac{\pi}{\alpha}\right)^2 (\upsilon - f)^2} \mathrm{e}^{-\mathrm{j}2\pi t_0 (\upsilon - f) + \phi} \tag{4.22}$$

The Eqs. (4.21) and (4.22) show straightforwardly that the center of gravity t_0 is equal to Eq. (4.9) and the spectral center of gravity f is equal to Eq. (4.12), i.e., the Gabor functions follow Heisenberg's uncertainty principle.

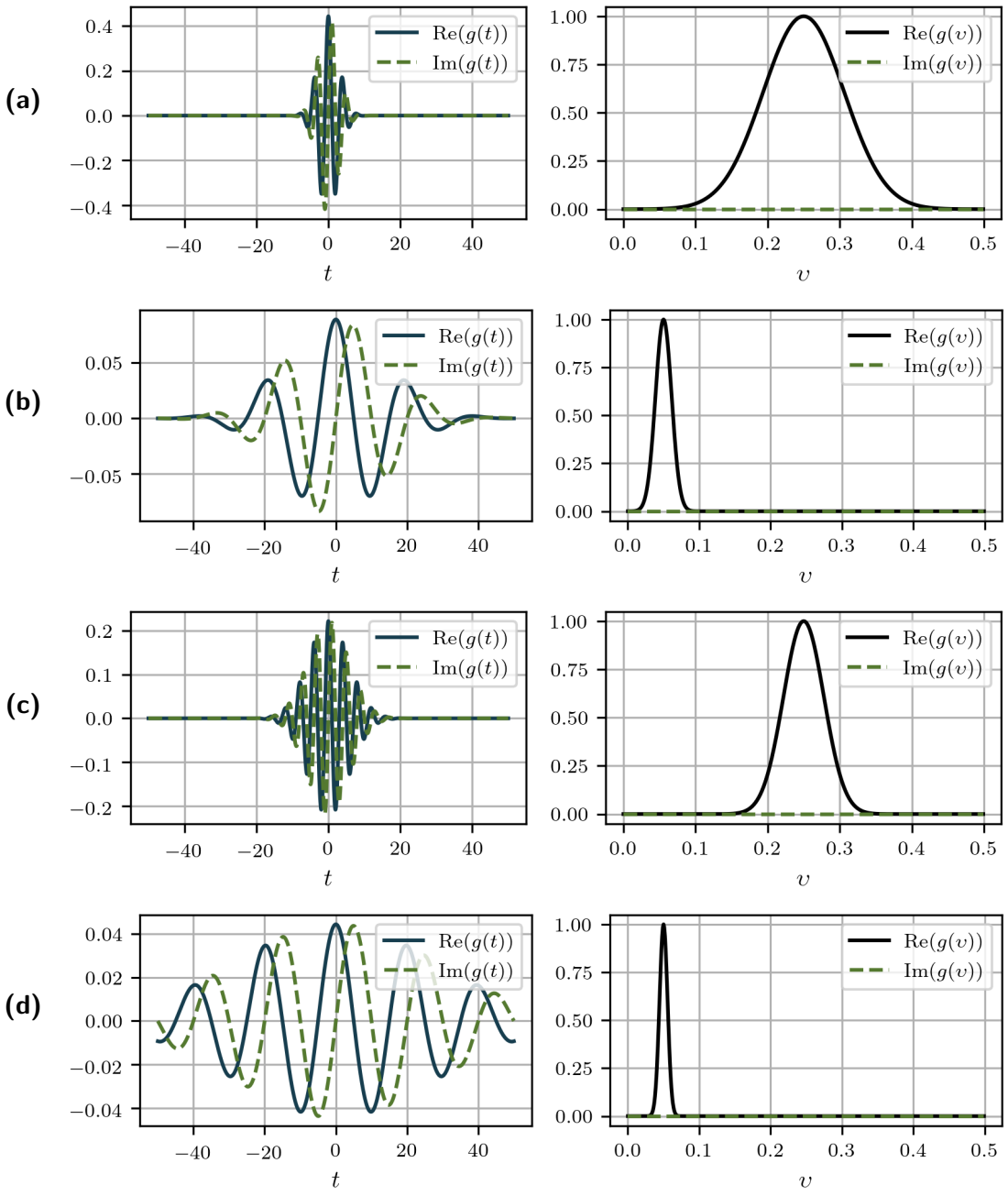


Figure 4.2: Visualization of the uncertainty principle in 1-d Gabor filters. First column: filters on the time domain, Second column: filters on the frequency domain. **(a)** $f = 1/4$, $\gamma = 1$; **(b)** $f = 1/20$, $\gamma = 1$; **(c)** $f = 1/4$, $\gamma = 2$; **(d)** $f = 1/20$, $\gamma = 2$.

Filter normalization

We can define the Gabor filter more appropriately by taking the following justifications. First, we must remember that we use the Gabor function as a linear filter to analyze a signal. Under this condition, the temporal analysis of the signal is carried out using the convolution operator. Considering that the Gabor function is concentrated near the time instant t_0 and that a convolution centered at the origin is preferable, we consider $t_0 = 0$. Since there is no evidence that any specific phase would be more beneficial than any other [Liu et al., 2005], another parameter that we can omit is the phase shift ϕ. Moreover, for the functions to be similar at all locations, the phase shift should depend on the location t_0, and thus, the phase shift can be removed from the origin-centered filter ($\phi = 0$). Then, the Gabor filter function in its compact form is defined as

$$\begin{aligned} g(t) &= \mathrm{e}^{-\alpha^2 t^2} \mathrm{e}^{\mathrm{j}2\pi f t} \\ G(\upsilon) &= \sqrt{\frac{\pi}{\alpha^2}} \mathrm{e}^{-\left(\frac{\pi}{\alpha}\right)^2 (\upsilon - f)^2} \end{aligned} \tag{4.23}$$

We can normalize the Gabor filter depending on the application we will use it. However, in this thesis, we use the general normalization based on the multi-domain representation property of the function following the subsequent conditions [Boukerroui et al., 2004]:

1. Maximum condition:

$$\max |G(\upsilon)| = 1 \tag{4.24}$$

2. Constant spectra condition:

$$\int_{-\infty}^{\infty} |g(t)| dt = 1 \tag{4.25}$$

From Eq. (4.22), it is evident that the maximal response of the Gabor filter in the frequency domain is a function of $\sqrt{\pi/\alpha^2}$, therefore, its inverse

$$\sqrt{\frac{\alpha^2}{\pi}} \tag{4.26}$$

can be used as the normalization factor in the time domain to fulfill the two conditions mentioned above. Then using the normalization factor (4.26), the normalized Gabor filter is defined as

$$\begin{aligned} g(t) &= \sqrt{\frac{\alpha^2}{\pi}} \mathrm{e}^{-\alpha^2 t^2} \mathrm{e}^{\mathrm{j}2\pi f t} \\ G(\upsilon) &= \mathrm{e}^{-\left(\frac{\pi}{\alpha}\right)^2 (\upsilon - f)^2} \end{aligned} \tag{4.27}$$

Fig. 4.3 shows a Gabor filter in the time and frequency domain before and after

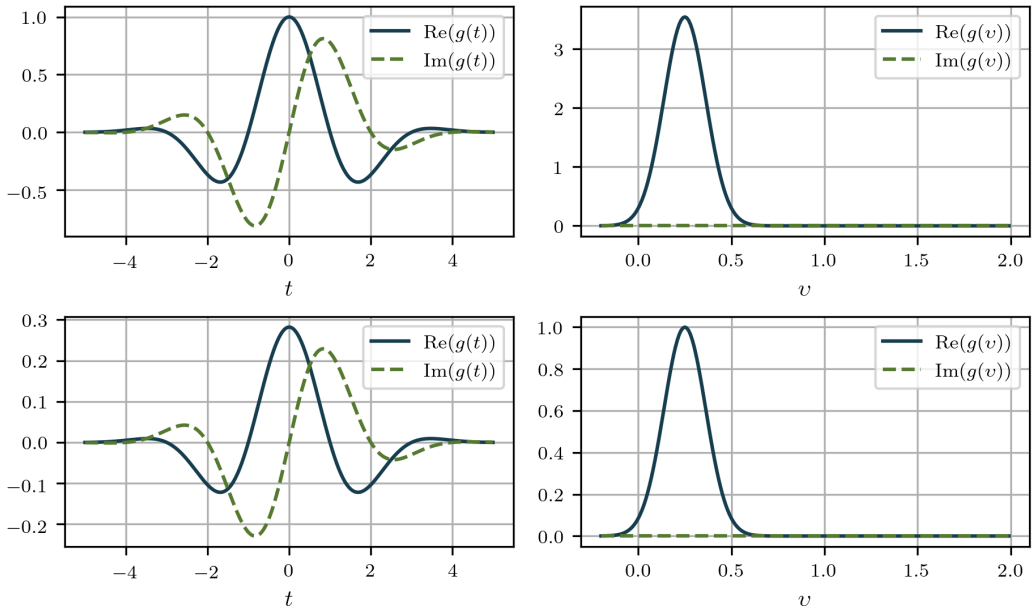


Figure 4.3: 1-d Gabor filter in the time domain (first column) and in the frequency domain (second column). From top to bottom: non-normalized and normalized Gabor filter with [$f = 1/4$, $\alpha = 0.5$, $t_0 = 0$].

normalization following the two conditions described above. At this point, it is important to note that normalization is an essential step in the multi-spectral analysis and the feature extraction of a signal.

Frequency filter spacing

Our main interest in Gabor filters is the multi-spectral analysis of a function. We accomplish this by using multiple Gabor functions as filters that are tuned on several frequencies f_m. This group of filters is known as a *Gabor filter bank*. The separation between the filter bank is defined through the half-response spatial frequency bandwidth B_f measured between two central frequencies $f_1 < f_2$ [Granlund, 1978]. This bandwidth is measured in octaves and we can express it as

$$B_f = \log_2 \left(\frac{f_2}{f_1} \right). \tag{4.28}$$

The frequency bandwidth Eq. (4.28) shows that the central frequencies f_m must have a logarithmic relationship to maintain a homogeneous spacing between the filters. The scaling factor $k = 2^{B_f}$ gives the logarithmic relationship, so the frequency of each filter (f) in this case corresponds to the scale information. Then we can write the central frequencies as

$$f_m = k^{-m} f_{max}, \quad m = \{1, \cdots, M\} \tag{4.29}$$

where f_m is the mth frequency, f_{max} is the maximal desired frequency, $k > 1$ is the

frequency scaling factor, and M is the total number of frequencies of the filter bank.

The octave spacing between two adjacent filters is an interesting property of the Gabor filters; however, the filters denoted by the Eqs. (4.27) have a spread that only depends on the parameter α, regardless of its central frequency f. This trait means that when implementing the Gabor function in a filter bank at different frequencies to obtain a multi-spectral decomposition of a signal, all of the filters will have the same spread in the frequency domain. We can see this effect in Fig. 4.4a, where we show a filter bank with five central frequencies and an adjacent filter's spacing of one octave, that is, $M = 5$ and $B_f = 1$.

Frequency crossing point

The fact we choose the filter bank's central frequencies f_m to have a constant separation causes two adjacent Gabor functions to intersect at a particular point on the frequency axis. For example, in a filter bank formed with two Gabor functions with central frequencies f_1 and f_2, the low cut-off frequency of the function at f_1 coincides with the high cut-off frequency of the function at f_2. Generally, in the literature, the crossing point c_1, corresponds to the points where the Gabor function has decreased half of its maximum value, i.e., $c_1 = 0.5$ [Granlund, 1978]. However, by setting the crossing point to half of the maximum value, the filter bank does not cover the input signal's entire spectrum. Consequently, the filter bank will not respond (or the response will be minimal) to artifacts oscillating between central frequencies.

We obtain the mathematical expression of this crossing point c_1 by defining a frequency interval Δf, representing the distance between points where the function $G(\upsilon)$ intersects adjacent functions at frequencies $\upsilon = f \pm \frac{\Delta f}{2}$ (see Subfig. 4.4b). The Gabor function has a peculiarity; its analytical form in the frequency domain is completely defined by the Fourier transform of the normalized Gaussian function Eq. (4.27).

$$G(\upsilon) = w(\upsilon) = \mathsf{e}^{-\left(\frac{\pi}{\alpha}\right)^2(\upsilon - f)^2} \tag{4.30}$$

therefore, evaluating Eq. (4.30) at $\upsilon = f + \frac{\Delta f}{2}$

$$G\left(f + \frac{\Delta f}{2}\right) = \mathsf{e}^{-\left(\frac{\pi}{\alpha}\right)^2\left(\frac{\Delta f}{2}\right)^2} = c_1 G(f) \tag{4.31}$$

we obtain the expression of the half-frequency interval

$$\frac{\Delta f}{2} = \frac{\alpha}{\pi}\sqrt{\ln\left(\frac{1}{c_1}\right)} \tag{4.32}$$

from which we obtain that the crossing point is defined as

$$c_1 = \mathsf{e}^{-\left(\frac{\alpha}{\pi}\right)^2\left(\frac{\Delta f}{2}\right)^2} \tag{4.33}$$

This expression allows us to control the intersection point of two adjacent filters of the filter bank. Modifying the crossing point allows the generation of filter banks that cover (almost) the entire frequency spectrum, translating into a more faithful decomposition of the input signal.

Effective and Adaptable Gaussian Envelope

The full bandwidth B_f (expressed in octaves) of a Gabor filter with center frequency f and cut-off frequency interval Δf is defined as [Daugman, 1985].

$$B_f = \log_2\left(\frac{f + \frac{\Delta f}{2}}{f - \frac{\Delta f}{2}}\right) \tag{4.34}$$

It is clear that using the half-frequency interval Eq. (4.32) into the frequency bandwidth Eq. (4.34), we find an expression that introduces a relationship between the frequency bandwidth B_f, the central frequency f, and the spread of the Gabor filter α.

$$B_f = \log_2\left(\frac{\frac{f}{\alpha}\pi + \sqrt{\ln\left(\frac{1}{c_1}\right)}}{\frac{f}{\alpha}\pi - \sqrt{\ln\left(\frac{1}{c_1}\right)}}\right) \tag{4.35}$$

The relationship is noticeable through the ratio

$$\gamma = \frac{f}{\alpha} \tag{4.36}$$

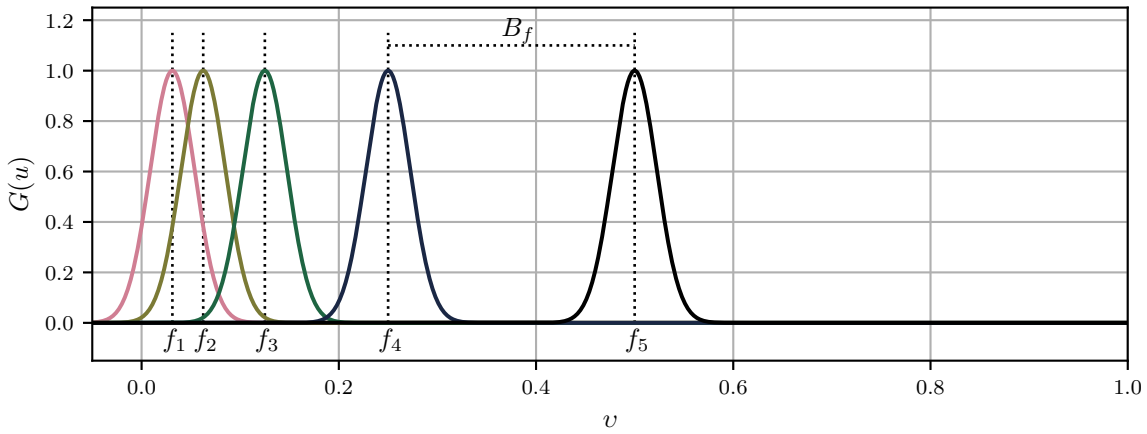
The ratio given in Eq. (4.36) allows generating Gabor filters of variable size as a function of the center frequency. For this, we must remember that the window size of a Gabor function is denoted by the effective width of a Gaussian function, which in the time domain has a form of

$$w(t) = \mathsf{e}^{-\frac{(t-t_0)^2}{2\sigma^2}} \tag{4.37}$$

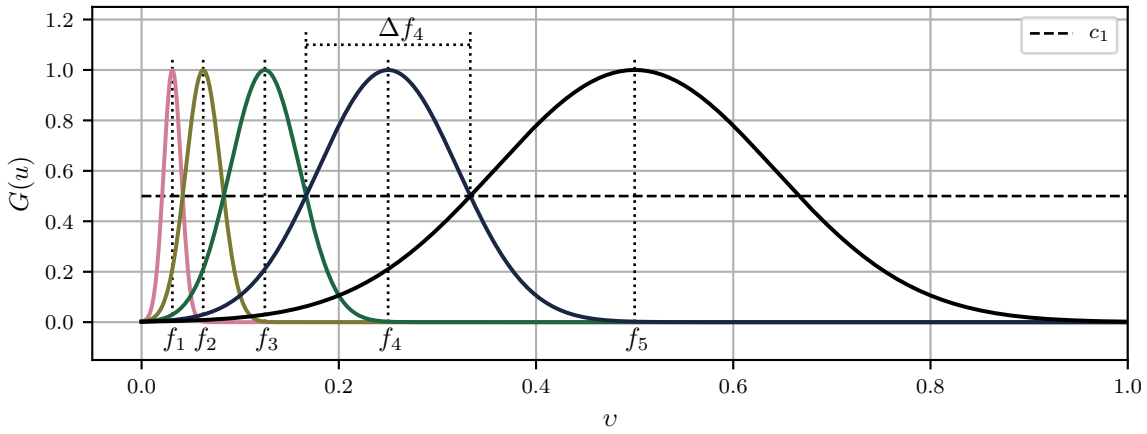
The Gaussian window Eq. (4.37) is infinite in its extent, so it is characterized by its locality t_0 and standard deviation σ, implicit in the Gabor function parameter α as $\alpha^2 = 1/2\sigma^2$. By setting the standard deviation dependent on the frequency ratio and the central frequency, we find that

$$\sigma = \frac{\gamma}{\sqrt{2}f} \tag{4.38}$$

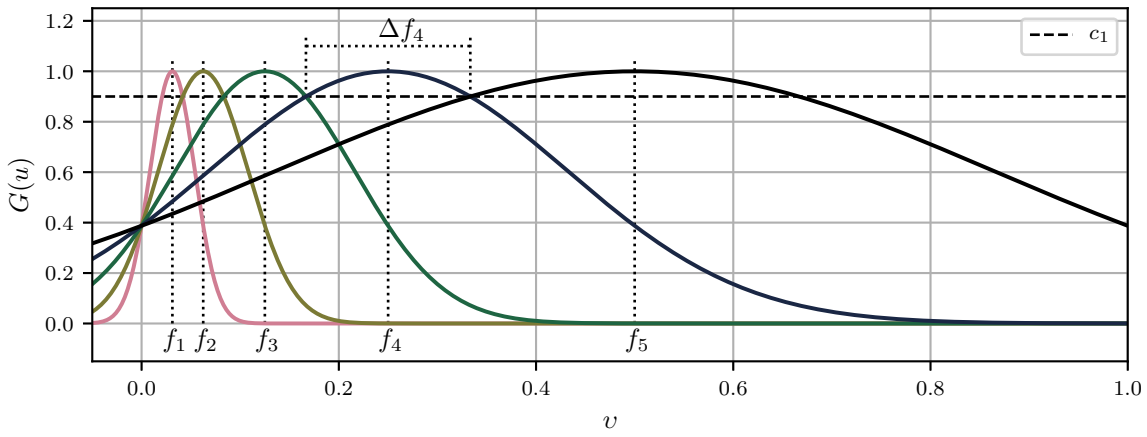
makes the Gabor's window adaptive as a function of the frequency.



(a)



(b)



(c)

Figure 4.4: Filter spacing and crossing point effect represented on a bank of filters in the frequency domain: **(a)** Separation of filters in octaves without crossing point between adjacent filters $[B_f = 1, \alpha = 0.1, c_1 = \text{n/a}]$, **(b)** High and low cut-off frequency points given by Δf $[B_f = 1, \alpha = f/\gamma, c_1 = 0.5]$, **(c)** Filter bank behavior after changing the crossing point $[B_f = 1, \alpha = f/\gamma, c_1 = 0.9]$.

In addition to adapting the filter size, with Eq. (4.38), we make the Gabor filter effective; that is, we make the filter envelope correspond to the time (resp. spatial) support where the function's values are significant. We use the empirical *three-sigma* rule [Pukelsheim, 1994], a conventional heuristic that expresses that nearly 99.7% of the Gaussian distribution's energy lies within three standard deviations of the mean. Therefore, we define the shortest interval of the function that includes most of the energy as

$$\kappa = \{x | x \in [-3\sigma, 3\sigma]\} \tag{4.39}$$

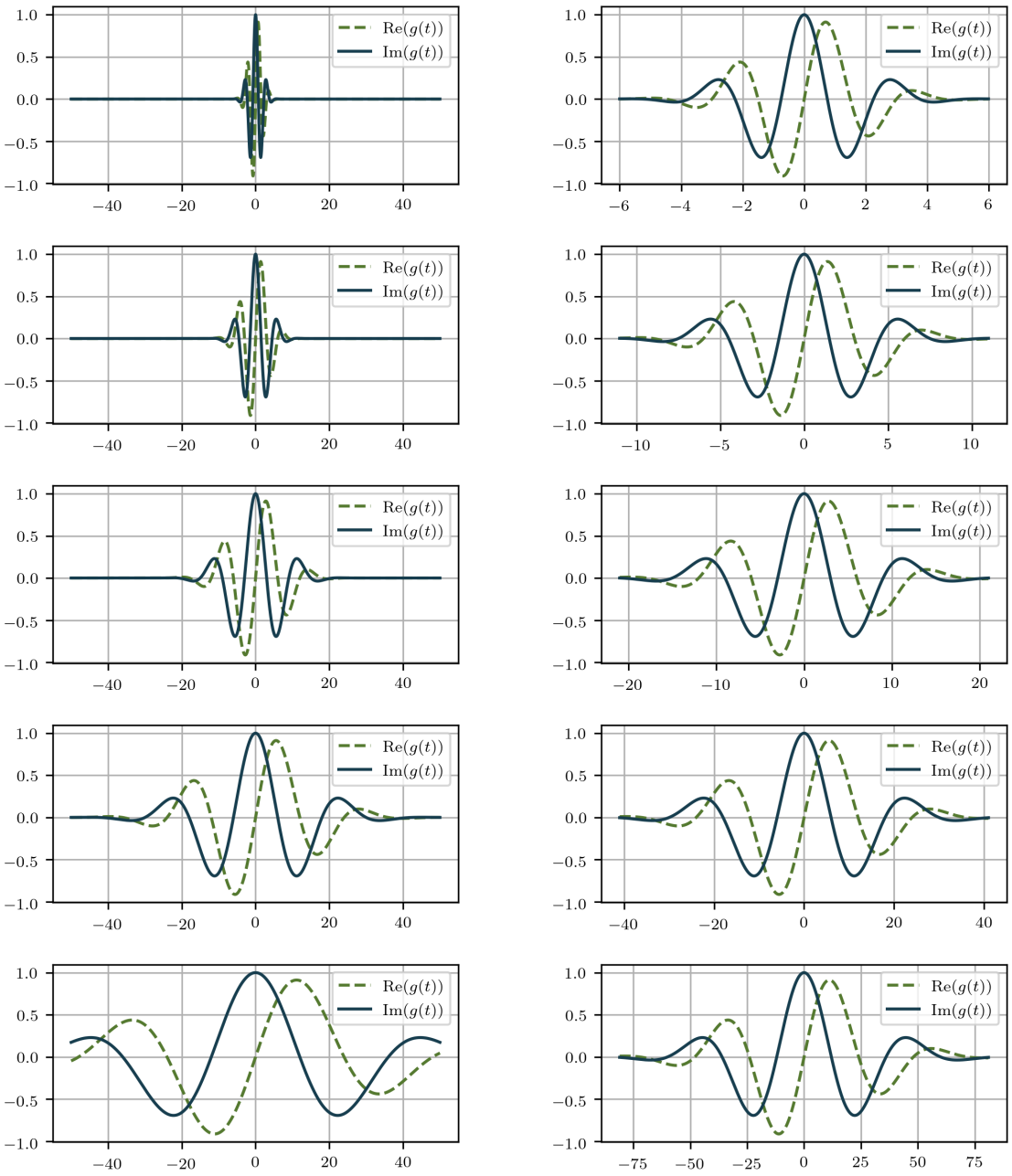
The fact that the bank filters have the same width at all frequencies is not a problem, nor is it a requirement to analyze a signal with the Gabor function. However, making the filter width dependent on its frequency implies a multi-resolution analysis since the filters behave like a scaled version of each other. The advantages of an effective adaptative envelope are related to the computation time and the loss of information when we filter a signal with a Gabor filterbank. Figure 4.5 illustrates the advantages of using a filter bank with adaptive and effective support (Fig. 4.5b) versus a conventional filter bank (Fig. 4.5a). More precisely, in the case of constant envelope width, for the example $\kappa = \{x | x \in [-50, 50]\}$ (see Fig. 4.5a), the computation time of the responses is the same for all filter frequencies. In contrast, with the adaptive envelope, the calculation time is reduced for high frequencies since the envelope width is smaller (compare images on the first column of Fig. 4.5). Moreover, there is a risk of losing information from the original signal if the chosen envelope is not large enough for low frequencies to fit in (compare images on the last column of Fig. 4.5). Using the adaptative and effective envelope width, we recover as much energy as possible from the signal by optimizing the response's computation time for each frequency f of the filter.

Optimized 1-d Gabor function

Gathering the different modifications of Gabor functions developed in the previous sections, we can define an optimized 1-d Gabor function as follows

$$\begin{aligned} g(t) &= \frac{f}{\gamma\sqrt{\pi}} \mathsf{e}^{-\left(\frac{f}{\gamma}\right)^2 t^2} \mathsf{e}^{\mathsf{j}2\pi f t} \\ G(v) &= \mathsf{e}^{-\left(\frac{\gamma\pi}{f}\right)^2 (v-f)^2} \end{aligned} \tag{4.40}$$

This Gabor function representation allows generating filter banks normalized by the maximum spectrum condition and homogeneously distributed in the frequency domain. Besides, a filter bank generated with the Gabor function Eq. (4.40) integrates the frequency crossing point allowing a quasi-total and almost flat coverage of the frequency spectrum, occupying the most relevant part of the filter using an adaptive window. A



(a) $\kappa = \{x | x \in [\pm 50]\}$ **(b)** $\kappa = \{x | x \in [\pm 3\sigma]\}$

Figure 4.5: Visual representation of the effective Gaussian envelope adaptation with filter bank with $M = 5$ frequencies in the space domain. **(a)** Filter bank with no control over the envelope, **(b)** Filter bank with control over the envelope's effective width.

possible disadvantageous effect of this approach is the ripple between the filter bank's filters; however, we must remember that the filters proposed here are normalized concerning the size of the support (Gaussian window) and the central frequency of each filter. Therefore, even though a filter responds to different frequencies, textures close to the filter's center frequency are weighted. Additionally, the ripple effect occurs more frequently when decreasing the bandwidth of the filter bank. Moreover, this effect can be reduced by applying a halfwave rectification or, more generally called a thresholding

[Petkov, 1995], [Grigorescu et al., 2003] [Kruizinga and Petkov, 1999].

Fig. 4.4 shows three examples of filter banks in the frequency domain. Particulary, Fig. 4.4a shows a bank without the relationship between the effective width and the central frequency of the filter, whereas the Figs. 4.4b and 4.4c show the interdependence between α, B_f, f and c_1 and the behavior of the bank with a different crossing point.

4.2.4 2-d Gabor Filters

The generalization of the Gabor function's theory from 1-d to 2-d is straightforward. First, we replace the time variable t with the pair of spatial coordinates (x, y) and the frequency variable f with the pair of frequency variables (u, v). Then, as for the 1-d case, the 2-d Gabor functions follows the Heisenberg principle where the uncertainty measures for the spatial and spatial-frequency domains are expressed in terms of Δx, Δy, Δu, and Δv, for which it holds that

$$\Delta x \Delta u \geq \frac{1}{4\pi}, \quad \Delta y \Delta v \geq \frac{1}{4\pi} \text{and} \quad \Delta x \Delta y \Delta u \Delta v \geq \frac{1}{16\pi^2} \tag{4.41}$$

The 2-d Gabor function is represented by the modulated product of a harmonic oscillation with a pulse in the form of a probability function. The harmonic oscillation is represented by a complex exponential on any spatial frequency and any orientation; the pulse is represented by an elliptical Gaussian ellipse on any orientation. For simplicity, we assume that the orientation of the Gaussian and the harmonic modulation are the same and, therefore, define a compact form of the 2-d Gabor Elementary Function (GEF) in the space domain applying the given simplifications as follows

$$\begin{aligned} g(x, y) &= \mathsf{e}^{-\left(\alpha^2 x_r^2 + \beta^2 y_r^2\right)} \mathsf{e}^{\mathsf{j}2\pi f x_r} \\ x_r &= x\cos\theta + y\sin\theta \\ y_r &= -x\sin\theta + y\cos\theta \end{aligned} \tag{4.42}$$

We obtain the analytical expression for the 2-d GEF in the spatial-frequency domain from the Fourier transform of Eq. (4.42), $G(u, v) = \mathcal{F}\{g(x, y)\}$, given by

$$\begin{aligned} G(u, v) &= \frac{\pi}{\alpha\beta} \mathsf{e}^{-\pi^2\left(\frac{(u_r - f)^2}{\alpha^2} + \frac{v_r^2}{\beta^2}\right)} \\ u_r &= u\cos\theta + v\sin\theta \\ v_r &= -u\sin\theta + v\cos\theta \end{aligned} \tag{4.43}$$

We can normalize the two above expressions following the same reasoning as in the 1-d case. We apply the maximum value condition Eq. (4.24) and the constant spectrum condition Eq. (4.25) described in section 4.2.3 to get $\max|G(u, v)| = 1$ and $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|g(x, y)|dxdy = 1$ for a filter on any frequency f and orientation θ. Under

these circumstances, the normalization constant is defined by

$$\frac{\alpha\beta}{\pi} \tag{4.44}$$

which applied to Eqs. (4.42) and (4.43), gives us the normalized Gabor function in both 2-d domains.

$$\begin{aligned} g(x,y) &= \frac{\alpha\beta}{\pi} \mathsf{e}^{-\left(\alpha^2 x_r^2 + \beta^2 y_r^2\right)} \mathsf{e}^{\mathsf{j}2\pi f x_r} \\ G(u,v) &= \mathsf{e}^{-\pi^2\left(\frac{(u_r-f)^2}{\alpha^2} + \frac{v_r^2}{\beta^2}\right)} \end{aligned} \tag{4.45}$$

Orientation filter spacing

The Gabor function defined by Eqs. (4.45) do not cover most of the spectrum when we use them to build a filter bank at different frequencies and orientations (see Fig. 4.6a); therefore, it does not help reconstruct a signal and the extraction of features. To obtain a more encompassing filter bank, it is evident that we need to include a relationship between the sharpness of the Gaussian window and the central frequency.

The sharpness of the Gaussian function, unlike the 1-d case, now includes two variables (α, β) that affect the effective width of the Gabor filter envelope. Such an envelope can have an elliptical shape, where α controls the length of the major axis and β controls the length of the minor axis.

The analysis of the frequency separation between adjacent filters of a bank viewed in the section 4.2.3 is also valid in the 2-d case. Thus, the full bandwidth of half the frequency response, B_f, represents the separation between the center frequencies; the interval Δf represents the distance between the points where $G(u,v)$ intersects adjacent functions (see Subfig. 4.4b). Finally, the full frequency bandwidth through the ratio γ and the frequency crossing point c_1 allows adapting the size of the major axis of the envelope α depending on the center frequency f.

We can do a similar analysis for the minor axis of Gabor's envelope. First, notice that insertion of the orientation variable θ in the 2-d case implies the existence of an angular separation B_θ between the centers of the filters in a bank (see Fig. 4.6a). This angular bandwidth can be defined by the total number of orientations N in a filter bank such that

$$B_\theta = \frac{\pi}{N} \tag{4.46}$$

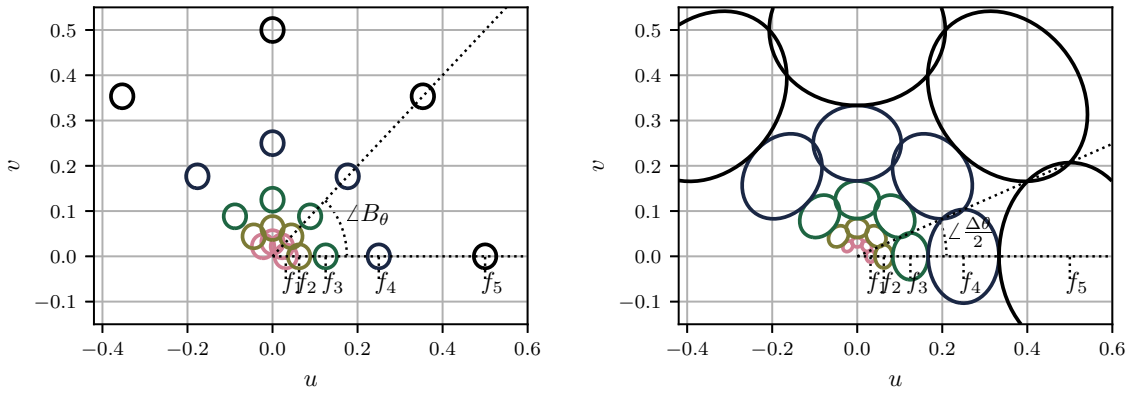
and therefore, we can obtain the the filter bank's orientation angles as

$$\theta_n = n\frac{\pi}{N}, \quad n = \{0, \cdots, N-1\} \tag{4.47}$$

We propose to vary the length of β as a function of the central frequency and the angular bandwidth through an angular interval $\Delta\theta$, which is the distance along β where

$G(u, v)$ intersects adjacent functions (see Subfig. 4.6b).

$$\frac{\Delta\theta}{2} = \frac{\beta}{\pi}\sqrt{\ln\left(\frac{1}{c_2}\right)} \tag{4.48}$$



(a) **(b)**

Figure 4.6: Filter spacing and crossing point effect represented on a 2-d filter bank in the frequentcy domain: **(a)** Filters separation without crossing points between adjacent filters $[B_f = 1, B_\theta = 45°, \alpha = \beta = 0.1, c_1 = c_2 = \text{n/a}]$, **(b)** Filters separation with crossing points between adjacent filters $[B_f = 1, B_\theta = 45°, \alpha = f/\gamma, \beta = f/\eta, c_1 = c_2 = 0.9]$. High and low cut-off frequency points given by Δf and $\Delta\theta$.

We know that for a filter whose center frequency is f and whose cut-off angular interval is $\Delta\theta$, the full orientation bandwidth B_θ expressed in radians is defined as [Daugman, 1985].

$$B_\theta = 2\tan^{-1}\left(\frac{\Delta\theta}{2f}\right) \tag{4.49}$$

It is clear that using Eq. (4.48) in Eq. (4.49), we find the expression that relates the frequency bandwidth to the central frequency and the length of the Gaussian minor axis.

$$B_\theta = 2\tan^{-1}\left(\frac{\beta}{\pi f}\sqrt{\ln\left(\frac{1}{c_2}\right)}\right) \tag{4.50}$$

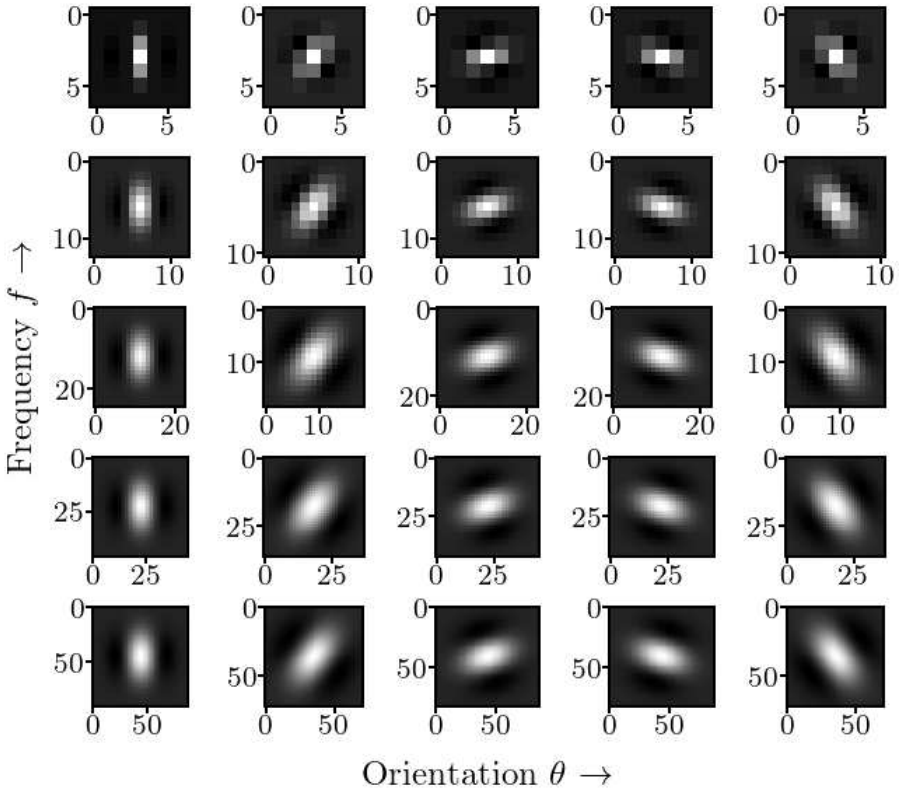
Taking the above relationships permits to write the 2-d GEF to use it into a bank as follows.

$$\begin{aligned} g(x, y) &= \frac{f^2}{\gamma\eta\pi}\mathrm{e}^{-\left(\frac{f^2}{\gamma^2}x_r^2 + \frac{f^2}{\eta^2}y_r^2\right)}\mathrm{e}^{\mathrm{j}2\pi f x_r} \\ G(u, v) &= \mathrm{e}^{-\left(\frac{\pi}{f}\right)^2\left(\gamma^2(u_r - f)^2 + \eta^2 v_r^2\right)} \end{aligned} \tag{4.51}$$

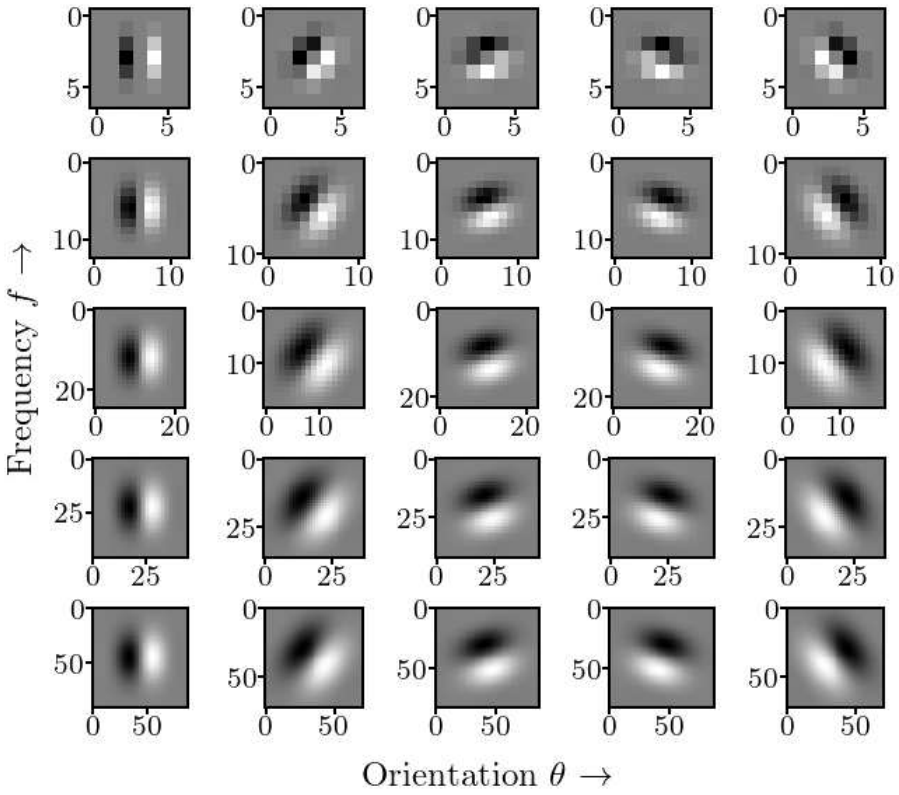
where now the length α of each filter in the bank will be determined based on the ratio $\gamma = \frac{f}{\alpha}$ and the frequency crossing point between adjacent filters c_1; and the length β

will be determined based on the ratio $\eta = \frac{f}{\beta}$ and the angular crossing point between adjacent filters c_2.

Fig. 4.6 shows the octave spacing and the orientation bandwidth for a bank of filters in the frequency domain. Particulary, Fig. 4.6a shows a bank without the relationship between the effective width and the central frequency of the filter, whereas Fig. 4.6b show the interdependence between α, β, B_f, B_θ, f and the crossing points c_1, c_2. Finally, Fig. 4.7 shows a family of optimized Gabor filters in 2-d (real and imaginary part) using $M = 5$ frequencies and $N = 5$ orientations. In the array of filters, the frequency of analisys increases from bottom to top and the orientation from left to right.



(a) Real part



(b) Imaginary part

Figure 4.7: Custom designed Gabor filter bank. The design parameters are: max/min period $[1/f_{min} = 70, 1/f_{max} = 4]$, crossing points (frequency and angular) $[c_1 = c_2 = 0.9]$, bandwidths (frequency and angular) $[B_f = 1, B_\theta = 35°]$, standard devaitions $[\sigma = 3]$.

4.3 Conclusion

In this chapter, we have presented a space-frequency analysis for creating a filter. This study aims to create a family of filters capable of measuring the texture information in an image under a perceptual approach. The human eye (perception and stimulus) is sensitive to the local contrast of textures, that is, to their amplitude. Therefore, we are interested in measuring this amplitude and its location correctly through a filter bank.

We use the Gabor function, which follows the Heisenberg uncertainty principle, to design an optimal filter bank. Given its ability to measure the texture's energy, both in the spatial and frequency domains, the filter configuration that we propose in this chapter allows us to build an adaptive filter bank. The filter bank is adaptable because we can customize it to privilege the precision of measuring the amplitude or the locality of a texture. Also, the proposed filter family is efficient since the Gaussian support is variable as a function of the central frequency of analysis, accelerating the convolution with the image.

On the other hand, each Gabor function of the filter bank is normalized concerning the central frequency of analysis, taking into account the size of the analysis window (Gaussian function). This characteristic makes it possible to attenuate the ripple and aliasing effects characteristic of the classic Gabor function.

The bank of filters proposed in this chapter achieves an almost total and uniform coverage of the frequency spectrum due to the modification of the frequency and angular crossover points. We must then uniformly cover the entire spectrum. Modifying the Gabor function to create a filter bank is an effort to conceive filters capable of measuring the amplitude and locality of a texture's different spectral components.

In the following chapters, we show this filter bank's use for the spectral decomposition of an image. This decomposition allows measuring the texture's information favoring the locality of the textures (without losing the amplitude), which makes this bank of filters a texture measurement tool.

Chapter 5

Color Texture Analysis Based on Spectral Decomposition

Résumé

Dans ce chapitre, nous présentons la décomposition spectrale d'une image couleur utilisant le filtre de Gabor. Nous utilisons la théorie des fonctions de Gabor développée au chapitre 4 pour extraire les caractéristiques de texture locale d'une image en couleur. La stratégie principale consiste à transformer l'image d'entrée d'un espace couleur réel à trois canaux en une représentation couleur complexe à deux canaux. Ensuite, nous utilisons une banque de filtres Gabor sur chaque canal de l'image pour extraire les informations de texture générées par les variations de couleur et d'illumination de l'image.

Abstract

In this chapter, we present the spectral decomposition of a color image employing the Gabor filter. We use the Gabor functions theory developed in chapter 4 to extract local features of a texture color. The primary strategy involves transforming the input image from a three-channel real color space into a two-channel complex color representation. Then, we use a bank of Gabor filters on each channel of the image to extract the texture information generated by the variations of color and illumination in the image.

5.1 Introduction

Gabor filters have long been used for analyzing textures and extracting corresponding image features. Their adaptability and customization, depending on the application and the relationship with the human visual system [Daugman, 1985], have made this technique one of the most relevant for analyzing textures in an image.

The use of Gabor filters for image texture analysis is highly dependent on the final application. Some of the most recognized works in the literature date back to the late 90s, where this technique was a hot research topic for image texture analysis. However, regarding the works present in the literature, we can separate the methods taking into account the nature of the extracted features. The first group uses Gabor filters to extract a global texture descriptor (Gabor signature). Generally, this strategy is suitable for applications where the images contain homogeneous textures, and it is sought to make the classification of images or an image retrieval system based on the content, as we can see in chapter 3. The second group is characterized by using Gabor filters to obtain local texture features present in an image. Such a strategy is suitable for image segmentation tasks. This chapter addresses the second case straightforwardly and comprehensively, delving into the spectral decomposition of color images to obtain texture features generated by the changes in illumination and (or) color.

We take advantage of the Gabor function's dual-domain (spatial and frequency) representation capability to create a bank of filters $G = \{g_{f,\theta}(x, y)\}$ and obtain the spectral decomposition of an input image $I(x, y)$ through the convolution operation of each of the filters such that

$$r_{f,\theta}(x, y) = I(x, y) * g_{f,\theta}(x, y) \tag{5.1}$$

represents the filter response at different central frequencies f (scales) and orientations θ. Given the complex form of Gabor filters Eq. (4.51) defined in chapter 4, the filter response $r_{f,\theta}(x, y)$ has a real and an imaginary part, here denoted as $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$, respectively.

The linear transformation of an image using Eq. (5.1), produces considerable information about the image's textures. The efficient manipulation of this information is the basis for extracting appropriate (local or global) texture features. Although the image's convolution by a filter bank is a common denominator in techniques based on signal processing, in the literature, we find various options to create more separable texture features (see Fig. 5.1). In general, these methods differ in the type of output they use to measure the image's textural information and the post-processing techniques to refine the Gabor responses. Among the possible Gabor filter responses to measure the texture information, some of the most used in the literature are

1. The amplitude of the response (magnitude or Gabor energy) [Bovik et al., 1990].

$$|r_{f,\theta}(x,y)| = \sqrt{\mathrm{Re}(r_{f,\theta}(x,y))^2 + \mathrm{Im}(r_{f,\theta}(x,y))^2} \tag{5.2}$$

2. The phase of the response [Palm and Lehmann, 2002].

$$\arg(r_{f,\theta}(x,y)) = \arctan 2\left(\frac{\mathrm{Im}(r_{f,\theta}(x,y))}{\mathrm{Re}(r_{f,\theta}(x,y))}\right) \tag{5.3}$$

3. The real component of the response [Jain and Farrokhnia, 1991].

$$\mathrm{Re}(r_{f,\theta}(x,y)) \tag{5.4}$$

4. The square amplitude of the response (Gabor local power spectrum) [Grigorescu et al., 2002].

$$|r_{f,\theta}(x,y)|^2 = \mathrm{Re}(r_{f,\theta}(x,y))^2 + \mathrm{Im}(r_{f,\theta}(x,y))^2 \tag{5.5}$$

while the most common post-processing techniques for the filter outputs consist of a non-linear transformation followed by smoothing using a rectangular or Gaussian window [Randen and Husoy, 1999], [Clausi and Ed Jernigan, 2000]. The application of non-linearity favors the activation of the textured areas in the images, while the smoothing favors the location of the energy obtained with the filter, avoiding the loss of information from the natural contours of the image. Figure 5.1 illustrates the stages (boxes with continuous black lining) and the input/outputs (boxes with black dotted lining) of the scheme mentioned above, referring to the extraction of Gabor-based texture features.

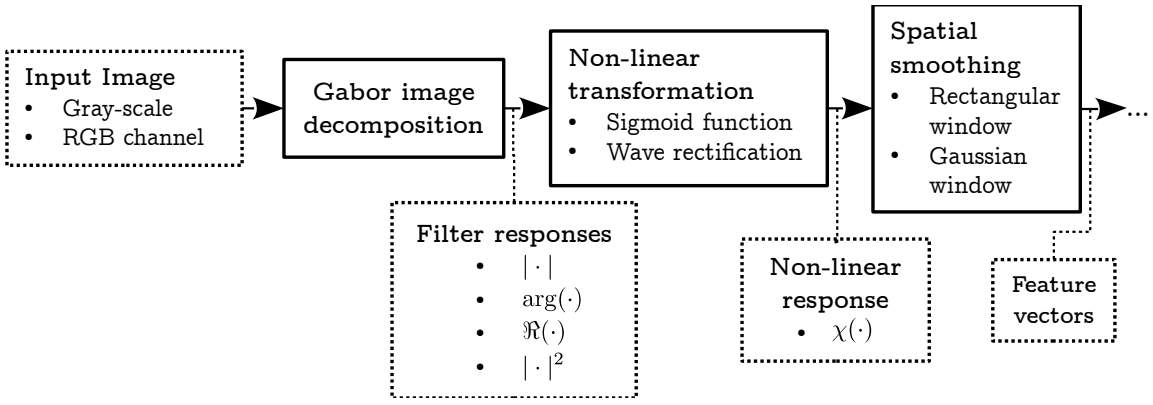


Figure 5.1: Pipeline of classic techniques for extraction of texture features using the Gabor filters.

5.1.1 Texture Features for Color Images

Most of the research work on texture has been done using gray-scale images and homogeneous textures; see for example [Jain and Farrokhnia, 1991], [Liu and Wechsler, 2003], [Liu et al., 2005], [Al-Kadi, 2017]. Consequently, the simplest way to obtain texture features from color images is to transform them into a gray-scale image. This strategy favors the acceleration of feature calculation because we work with scalar values instead of vectors. However, despite the good results in images with homogeneous gray-scale textures, reducing channels for a natural-color image with non-homogeneous textures does not ensure the generation of representative texture features. This outcome is primarily because luminance variations and variations in chromaticity generate the non-homogeneous textures in a color image. Moreover, the real-world scenes are in color and contain non-homogeneous textures. For example, in the case of a texture image in the RGB color space, which its gray-scale transformation represents the levels of red, green, and blue [Artusi et al., 2016] such as

$$L = 0.299R + 0.587G + 0.114B; \tag{5.6}$$

if the image contains isoluminant colors (colors with the same luminance value), the transformation L leads to minimization or loss, in the worst case, of textures generated by the color changes.

Notwithstanding, we find a large number of methods that propose the characterization of textures in color images. Such methods generally use two strategies for the analysis of color textures [Mäenpää and Pietikäinen, 2004], [Qazi et al., 2011]:

- process color and texture information separately
- process color and texture as a joint phenomenon

The first category methods assume that the spatial variations that form textures and color distributions of the image are independent cues (see for example [Permuter et al., 2006]). We differ from this point of view, and we consider that color and texture information in an image is a joint phenomenon based on the idea that textural segmentation occurs based on the distribution of simple properties of texture elements, for example, the brightness, color, size, and the slopes of contours and other elemental descriptors of a texture [Werner and Chalupa, 2004].

There are various techniques to joint color and texture information to characterize natural color textures in this regard. A popular option is to get unichromatic texture features from each color channel of the image using, for example, Gabor filters. Taking the RGB color space as reference, the filter responses represent the texture features of each primary color red, green, and blue independently, i.e., in principle, this strategy does not involve the correlation between RGB band colors. This strategy might be

corrected using the opponent color model based on the human color vision theory [Jain and Healey, 1998]. In such a case, each unichromatic feature vector (RGB-feature) is multiplied and normalized by the feature vector of its opponent color to include the correlation between color channels [Palm et al., 2000]. This method manages to gather the information of color and texture under a frame of human color perception. However, the normalization and multiplication of the unichromatic texture feature vectors imply extra post-processing steps in the features extraction pipeline.

One way to avoid the post-processing stage after the image Gabor decomposition is to first transform the color image in a color space that handles the coupling between the color channels rather than separating them as individual components of the color space. The quaternion framework [Sangwine and Ell, 2000] provides this possibility of coupled color representation. It encodes the color value of each pixel in a pure quaternion, where the real component is set to zero, and the three imaginary components represent the color band, such as $I(x,y) = R(x,y)\mathsf{i} + G(x,y)\mathsf{j} + B(x,y)\mathsf{k}$. This 3-component vector representation yields a system with well-defined mathematical operations, such as Quaternion Fourier Transform, that makes the Gabor image decomposition possible through the Quaternion Gabor Filters (QBF) [Subakan and Vemuri, 2009]. However, when using quaternion values, the non-existing commutativity must be considered; the QGF does not support any physic interpretation of what is measured.

Another alternative to this problem is representing the image in one of the two-channel color spaces, previously defined in chapter 2, where one channel contains the luminance information and the other the chrominance information of the image. We can obtain such a representation from the non-linear color spaces like LAB or LUV and HSV or HSL perceptual color spaces. The representation in the form of luminance-chrominance concentrates the color information in a complex channel, which is compatible with the multispectral Gabor decomposition. In both cases, the choice of a pertinent color space for the texture's characterization is necessary [Qazi et al., 2011].

The methodology we present in this chapter mainly follows the stages shown in the diagram of figure 5.1. We introduce some modifications to exploit the color and texture information in the same framework. The modifications proposed to this scheme are transforming the input image from the RGB color space to one of the luminance-chrominance spaces (or complex two-channel color spaces) described in chapter 2. Then, the non-linear transformation was replaced by a morphological opening followed by an adaptive Gaussian smoothing to highlight the amplitude of the filter responses. Under this configuration, we obtain a spectral decomposition of the image that considers the textures generated by changes in lighting and those generated by color changes. Figure 5.2 shows the stages we perform for the extraction of local features.

Later in the chapter, we apply the Gabor feature space for the segmentation of natural color images.

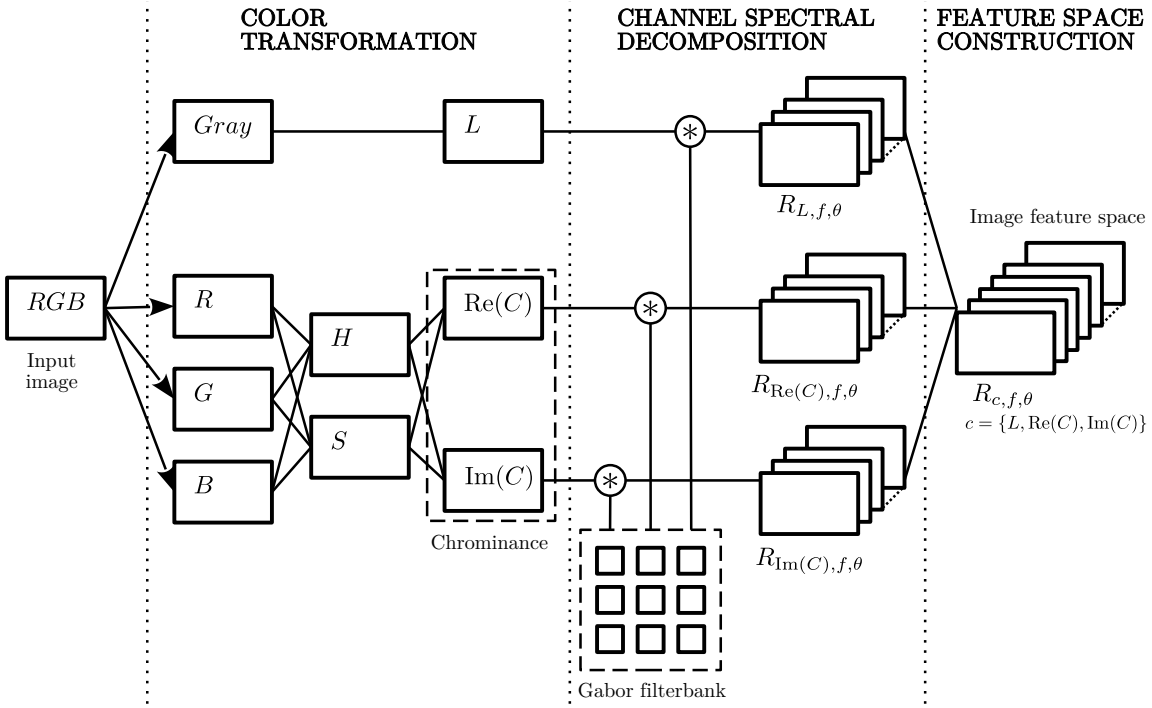


Figure 5.2: The proposed methodology for the computation of Gabor features in color images.

5.2 Gabor-filter-based Texture Feature Space

5.2.1 Color Image Transformation

The first stage in creating the feature space is transforming the input image from the RGB color space to the two-channel luminance-chrominance space. The representation in two channels, one real and the other complex, of a color image allows us to separate the intervention of luminance and colors in the generation of textures in an image. To help visualize such a joint phenomenon, we create a synthetic image that reflects the complexity of natural color images (see Fig. 5.3).

Synthetic Image Description

The synthetic image we create (see Fig. 5.3) contains seven different regions with spatial variations (textures) generated by alternating various colors at different frequencies. Each tile of the image is perceived as a whole; this is perceptually a constant region. The colors alternate along different directions in the chrominance phase (Fig. 5.4). The last image region has two frequency components.

We generate the input image with the sign function of a 2-d sinusoidal signal mul-

tiplied by the color values of each region in the RGB color space such that

$$I(x,y) = sgn(\sin 2\pi(f_x x_r + f_y y_r)) \cdot [R,G,B] \tag{5.7}$$
$$x_r = x\cos\theta + y\sin\theta$$
$$y_r = -x\sin\theta + y\cos\theta$$

We control the image texture by varying the frequencies f_x, f_y and the orientation angle θ of the coordinate plane (x,y), where $\theta = 0°$ means vertical variations and $\theta = 90°$ horizontal variations. The proposed image has a size of 320×1400 pixels. The seven characteristic regions of the image are spread over 1400 pixels wide, so each region is about 320×200 pixels; that is, the color/texture distribution in the image changes every 200 pixels (on the x-axis). The following expressions define the spatial image variations.

$$f_x = \begin{cases} 0 & 0 \le x \le 200 \\ 1/64 & 201 \le x \le 400 \\ 1/32 & 401 \le x \le 600 \\ 1/16 & 601 \le x \le 800 \\ 1/8 & 801 \le x \le 1000 \\ 1/4 & 1001 \le x \le 1200 \\ 1/8 & 1201 \le x \le 1400 \end{cases} \quad ; \quad f_y = \begin{cases} 0 & 0 \le x \le 200 \\ 0 & 201 \le x \le 400 \\ 0 & 401 \le x \le 600 \\ 0 & 601 \le x \le 800 \\ 0 & 801 \le x \le 1000 \\ 0 & 1001 \le x \le 1200 \\ 1/32 & 1201 \le x \le 1400 \end{cases}$$

For comprehension purposes, we use colors easily identified in the RGB space (primary colors) or in the HSV space (perceptual colors) to generate the image textures. Figure 5.3 depicts the resulting synthetic image.

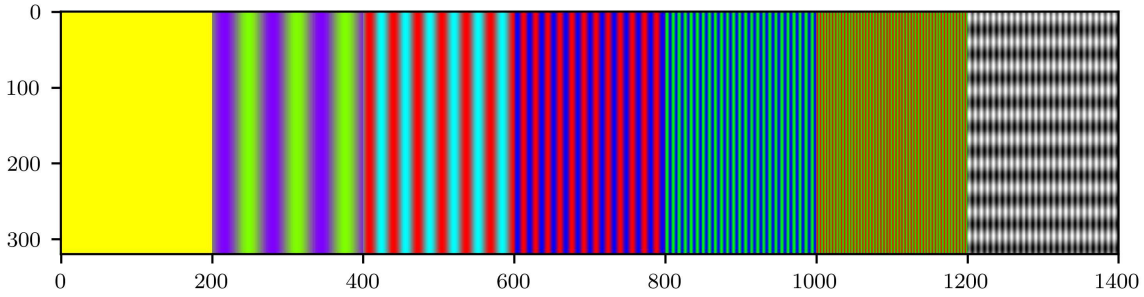


Figure 5.3: Synthetic color textured image.

The 2-d sinusoidal modulations generate textures in the image at different and well-known frequencies. These modulations change the colors of the regions generating a texture of oriented lines. The regions of the synthetic image have the following color and texture characteristics.

Region 1. Textureless zone: This region does not contain spatial variations, i.e., it has only a solid color. The color of the region is yellow (arbitrarily chosen).

Region 2. Lowest frequency textured zone with colors on the imaginary plane: This region is described by the vertical texture generated by variations between purple and green lime. Such colors are found in the imaginary axis of the chrominance plane. The colors in this region change every 64 pixels (along the x-axis).

Region 3. Textured zone with colors on the real plane: This region contains a vertical texture generated by the variations between red and cyan. Such colors are found in the real axis of the chrominance plane. The colors in this region change every 32 pixels (along the x-axis).

Region 4. Textured zone with two primary colors: The variations between red and blue generate the horizontal texture of this region. The colors in this region change every 16 pixels (along the x-axis).

Region 5. Textured zone with two primary colors: The variations between blue and green generate the horizontal texture of this region. The colors in this region change every 8 pixels (along the x-axis).

Region 6. Textured zone with two primary colors: The variations between green and red generate the horizontal texture of this region. The colors change every 4 pixels (along the x-axis).

Region 7. Colorless mixed textures zone: This region contains two textures, both of them formed by the variations between black and white, i.e., there is no color information. Moreover, the textures change in frequency and orientation; the pixels of the horizontal texture change of color every 4 pixels along the x-axis (highest frequency), while the pixel values of the vertical texture changes every 16 pixels along the y-axis (same frequency as region 4).

We summarize the colors and frequency of each zone in Table 5.1. In the table, we expose the RGB and HSV values of the texture-forming colors as well as the frequency and orientation of each section.

Graphical Display of the Synthetic Image Color Distribution

The choice of texture-forming colors comes from the interest in visualizing the color spectrum of the image more graphically. The two-channel color spaces, described in chapter 2, encode color information in a complex chrominance channel. Therefore, we can interpret the chrominance values as points within a complex plane (see Fig. 5.4).

	Region						
	1	2	3	4	5	6	7
Color 1							
Name	Yellow	Purple	Red	Red	Blue	Green	Black
RGB values	[255, 255, 0]	[128, 0, 255]	[255, 0, 0]	[255, 0, 0]	[0, 0, 255]	[0, 255, 0]	[0, 0, 0]
HSV values	[60, 100, 100]	[270, 100, 100]	[0, 100, 100]	[0, 100, 100]	[240, 100, 100]	[120, 100, 100]	[0, 0, 0]
Color 2							
Name	-	Green lime	Cyan	Blue	Green	Red	White
RGB values	-	[128, 255, 0]	[0, 255, 255]	[0, 0, 255]	[0, 255, 0]	[255, 0, 0]	[255, 255, 255]
HSV values	-	[90, 100, 100]	[180, 100, 100]	[240, 100, 100]	[120, 100, 100]	[0, 100, 100]	[0, 0, 100]
Texture							
Freq.	-	1/64	1/32	1/16	1/8	1/4	1/8 1/32
Angle	-	90°	90°	90°	90°	90°	0° 90°

Table 5.1: Specifications of the color and texture settings for each of the regions within the synthetic image.

In the case of LAB/LUV spaces, the values of chrominance are defined in Cartesian coordinates as

$$C = A + \mathrm{i}B, \tag{5.8}$$

where channel A values represent the real axis coordinates, and channel B values represent the imaginary axis coordinates.

HSV/HSL color spaces encode chrominance values as points in polar coordinates

$$C = S\mathrm{e}^{\mathrm{i}H}, \tag{5.9}$$

where channel H values are the angle expressed in radians and channel S values are the distance from the origin to the color points.

Considering these two chrominance configurations, the one defined by hue and saturation provides a more straightforward geometric interpretation of chrominance. Originally hue is a cylindrical dimension representing the color tints in the chromatic circle as angles, while saturation, which represents the purity of color, places achromatic tints in the center of the circle and pure colors on the circular edge of the disk. In figure 5.4, we show the synthetic image color distribution plot using the HS dimensions in the 2-d chrominance complex plane. The hue H is expressed in degrees, and saturation S is normalized between 0 and 255. This representation complements the description of our synthetic test image Fig. 5.3, showing the variation between colors that generate textures. We can notice in the chroma circle of figure 5.4 the transition between the red-green-blue primary colors at 0°, 120° and 240° respectively; the transition between purple and lime green on the imaginary axis with a hue of 90° and 270° respectively; the transition between red at 0° and cyan at 180° passing through the real axis of the plane and finally; the yellow color with a hue value of 60°.

Under this color space configuration, the colors of an image are represented in the complex chrominance plane. However, in this plane, the variations due to luminance information do not appear. The brightness of the colors is captured in the luminance

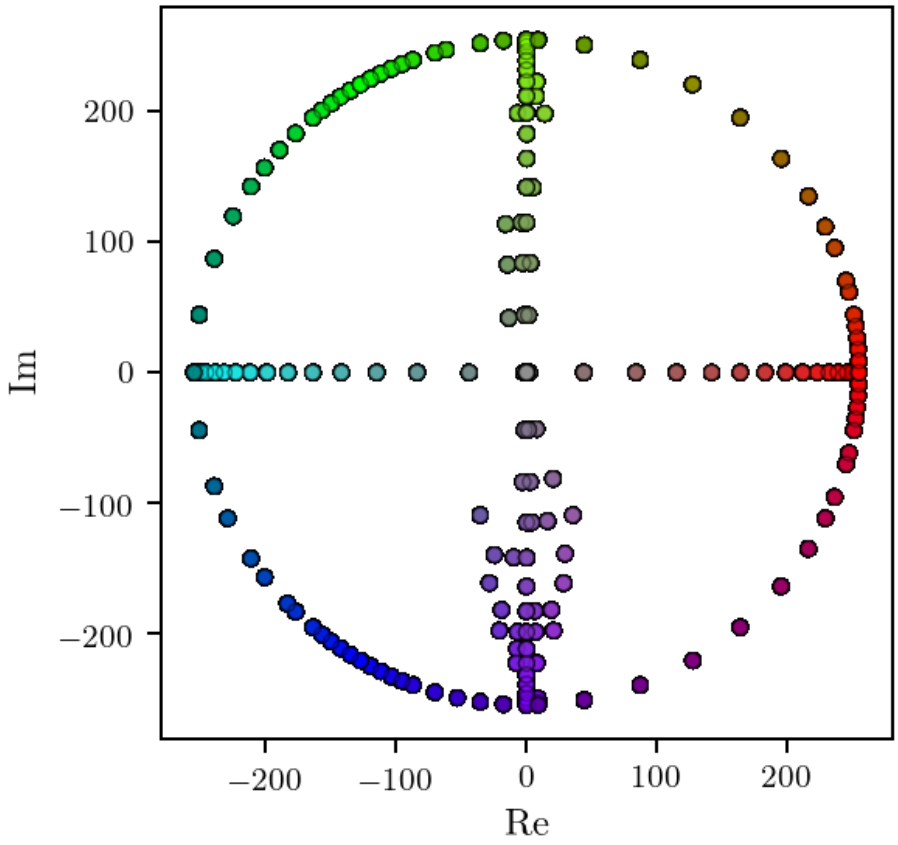


Figure 5.4: Synthetic image chrominance distribution represented in the color complex plane.

channel. Depending on the color space that we use to build the two-channel color representation, we can obtain the luminance channel differently. In the LAB/LUV and HSL color spaces, the luminance channel is directly represented by the L dimension values. In the HSV color space, we use the image transformation from RGB to grayscale following the Eq. (5.6) to obtain the luminance channel.

Fig. 5.5 shows the three dimensions of the two-channel representation of the synthetic image ($L(x, y)$, $\mathrm{Re}(C(x, y))$, $\mathrm{Im}(C(x, y))$) in grayscale. In this representation, we use the HSV color space as a basis to obtain the luminance and chrominance values, so the L channel is obtained with Eq. (5.6) and the C channel with Eq. (5.9). In Fig. 5.5, we can see how the different regions show more or less important values depending on the channel in which they are. For example, the horizontal and vertical spatial variations of region 7 (between 1200 and 1400 column pixels) are only visible in the luminance channel $L(x, y)$ (see left image in 5.5). We observe this same effect in the chrominance channels, for example, with the vertical textures of region 3. The spatial variations of this region are made up by the alternation of colors that live only in the real plane of chrominance (red and cyan), so the texture is only visible in channel $\mathrm{Re}(C(x, y))$ (see column pixels between 400 and 600 of the central image and the image to the right of Subfig. 5.5).

We can also visualize the color variations and their influence on the generation of textures by plotting a horizontal line using a row of pixels' intensity values for each dimension of the two-channel space. In Fig. 5.5b, we show the variations generated by changes in color and (or) lightness in a 1-d plot. Taking the area without texture (region 1), the horizontal line between pixels 0 and 200 remains constant in all three channels due to the absence of texture. However, in region 2, which corresponds to the low-frequency texture formed by the colors at 90° and 270° in the chroma circle, we see that variations are only present in the imaginary channel of chrominance $\mathrm{Im}(C(x, y))$. In regions 4, 5, and 6, since the texture-forming colors are two of the three primary colors (red, green, blue), the variations are visible in all three luminance-chrominance representation channels. Finally, in the last region (colorless mixed textures zone), we can see that the variations are only present in the channel that describes the luminance $L(x, y)$.

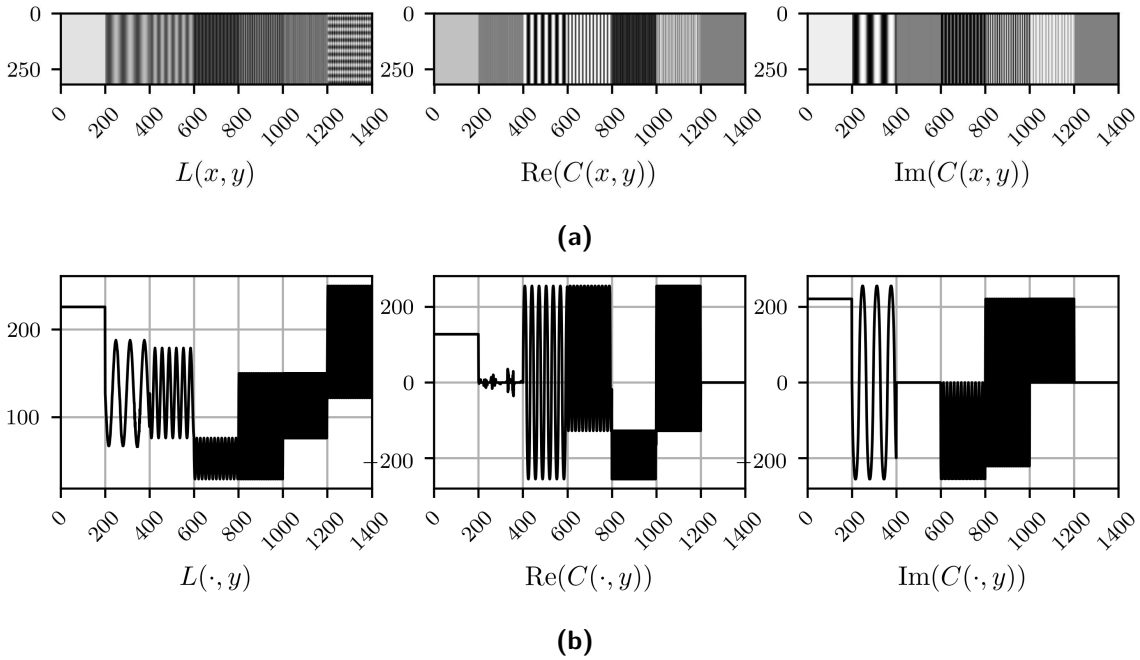


Figure 5.5: Illustration of the proposed synthetic image: **(a)** Luminance and chrominance decomposition; **(b)** Numerical values on a horizontal line cut through the three channels.

5.2.2 Spectral Image Decomposition

The first stage of our methodology for color-textured images' characterization consists of representing the input image in a two-channel luminance and chrominance space. Once we represent the image in this space, we obtain the spectral decomposition of each image dimension using a family of optimized Gabor filters (shown in Fig. 5.6). Following this strategy, we measure the spatial variations generated by luminance and chrominance individually.

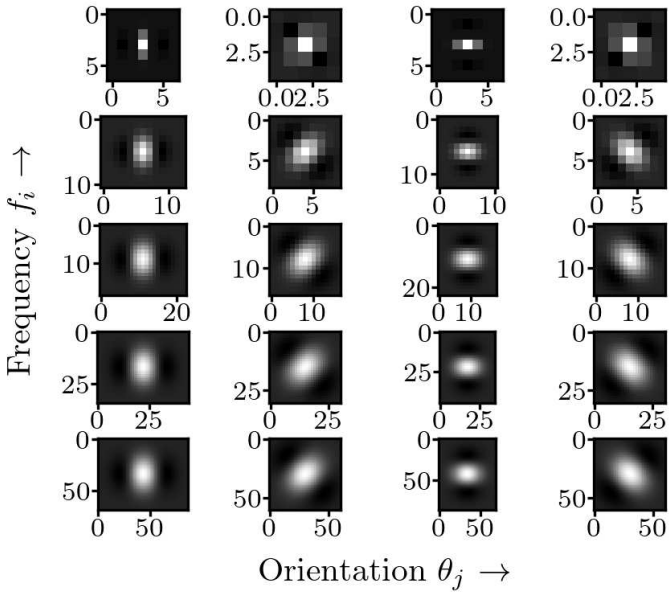


Figure 5.6: Gabor filter bank with filters at five frequencies f and four orientations θ used to model the synthetic image textures.

Among the different strategies to model the texture information with the Gabor filters listed above, we used the amplitude filter response Eq. (5.2) such that

$$e_{c,f,\theta}(x,y) = |r_{c,f,\theta}(x,y)|, \tag{5.10}$$

where $|r_{f,\theta}(x,y)| = \sqrt{\mathrm{Re}(r_{f,\theta}(x,y))^2 + \mathrm{Im}(r_{f,\theta}(x,y))^2}$ represents the image energy captured by a Gabor filter set at frequency f and orientation θ for every color channel $c = \{L, \mathrm{Re}(C), \mathrm{Im}(C)\}$.

The responses generated by Eq. (5.10) represent the raw Gabor responses, that is, without any post-processing. Although the Gabor filters we use are optimized (see chapter 4 for more details on filter optimization) to capture the most information by reducing the trade-off between the spatial-frequency domains, the raw responses lack homogeneity, especially at high frequencies, where the filter support is smaller. Fig. 5.7 shows the raw Gabor responses at different frequencies and orientations of the channels of the luminance-chrominance space.

The raw response images are organized as a matrix arrangement that follows the same structure as the Gabor family of filters. The rows in the array represent the responses to the bank's different frequencies (the frequency increases from bottom to top). The array columns represent the responses to the bank's different orientations; from left to right, θ varies between 0° and 180° at equidistant intervals.

In the raw Gabor response array images, the zones are brightened more or less depending on the information of the regions. A high level of brightness indicates more energy recovered by the Gabor filter at that frequency and orientation. For example, analyzing the responses of the first column of the luminance channel arrangement, they

show how the areas light up as the filter frequency changes. The textures' frequency in the synthetic image increases from left to right, while the filter bank's frequency increases from the bottom up, which generates this staircase effect of illuminated areas in the first column of images in Subfig. 5.7a.

This same effect is seen when analyzing region 7 of the synthetic image (the region with mixed textures at different frequencies and orientations between column pixels 1200 and 1400). The textures of such a region are formed by alternating black and white pixels, so all their information is in the luminance channel $L(x, y)$. The vertical texture of the region has frequency $f = 1/8$ and orientation $\theta = 0°$ while the horizontal texture has frequency $f = 1/32$ and orientation $\theta = 90°$. Under this configuration, the response images that reflect the texture information are the image in row 1, column 3 (from bottom to top and left to right) for the horizontal texture and; the response image in row 4, column 1 for the vertical texture. Note that in row 4 of the responses array (corresponding to the frequency $f = 1/8$), we see that region 5 is illuminated with the same intensity as region 7, which indicates that both zones contain textures at that frequency. This information is consistent with the description of the regions of our synthetic image.

Another interesting thing visible in the Gabor responses is that region 1 of the synthetic image (region without texture between pixels 0 and 200) does not illuminate any channel or any frequency or orientation of the filters. This effect occurs because Gabor filters only retrieve the texture information of the image. Color information is implicit in the chrominance channel. We see this reflected in the responses of the real and imaginary channels of the chrominance Subfigs. 5.7b, 5.7c.

Refinement of Gabor responses

Although the raw Gabor responses are a good starting point for describing textures, these responses exhibit a lack of spatial homogeneity generated by the trade-off between the spatial-frequency domains of the Gabor filters (see chapter 4.2, section 4.2.2 for more details about the uncertainty principle in Gabor filters and the optimization of Gabor filters proposed in this thesis). Clearly, this behavior is more visible at high frequencies, where the support of Gabor filters is smaller. For example, in the responses of the imaginary channel of the chrominance corresponding to the frequency $f = 1/8$ and orientation $\theta = 0°$ (response image at row 2 column 1 in Subfig. 5.7c), we see how the energy found by the filter for the region 4 (between pixels 600 and 800) is not homogeneous, and the lines texture effect keeps appearing.

In the literature, there are various strategies to homogenize the filter responses; one of them very recurrent is the application of a non-linear transformation [Jain and Farrokhnia, 1991]. This non-linear transformation acts as the bounded sigmoid activation function used in artificial neural networks. The non-linear transformation

modulates the sinusoidal signals of the image to square signals, so it behaves like a blob detector. The downside to this approach is that we need to define an empirical parameter that functions as a threshold for the transformation.

We propose to use a morphological opening to achieve homogenization of the Gabor responses. The opening of each Gabor response is defined as

$$\widehat{e}_{c,f,\theta} = \gamma_B(e_{c,f,\theta}), \tag{5.11}$$

where B is a structuring element and $\gamma_B = e_{c,f,\theta} \circ B$ is the morphological opening. Our approach's advantage is that the structural element's size is defined individually for each Gabor energy by the central period $T = 1/f$ of each filter. For example, the radius of a disk-shaped structuring element for the Gabor energies obtained with a filter with a center frequency $f = 1/8$ is 8 pixels. Gabor responses after morphological opening appear in the Fig. 5.8.

Finally, after the morphological opening, we apply an adaptive Gaussian smoothing to localize the Gabor energy response. This smoothing is defined as

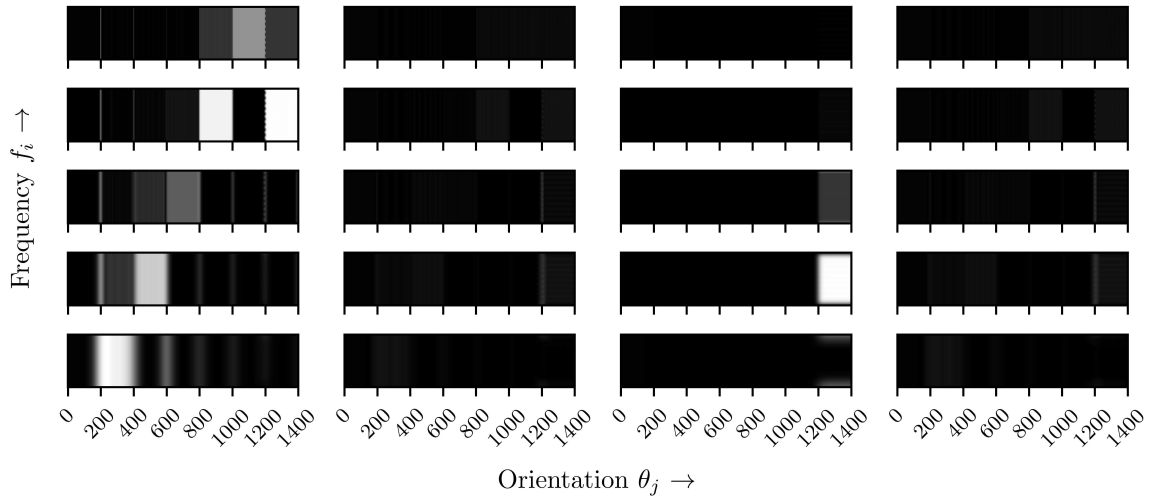
$$\widetilde{e}_{c,f,\theta}(x, y) = W(x, y)_\sigma * \widehat{e}_{c,f,\theta}(x, y), \tag{5.12}$$

where the scale parameter σ of the Gaussian window $W(x, y)_\sigma$ is the maximum value between the standard deviations of the Gabor filter support in the x and y axis.

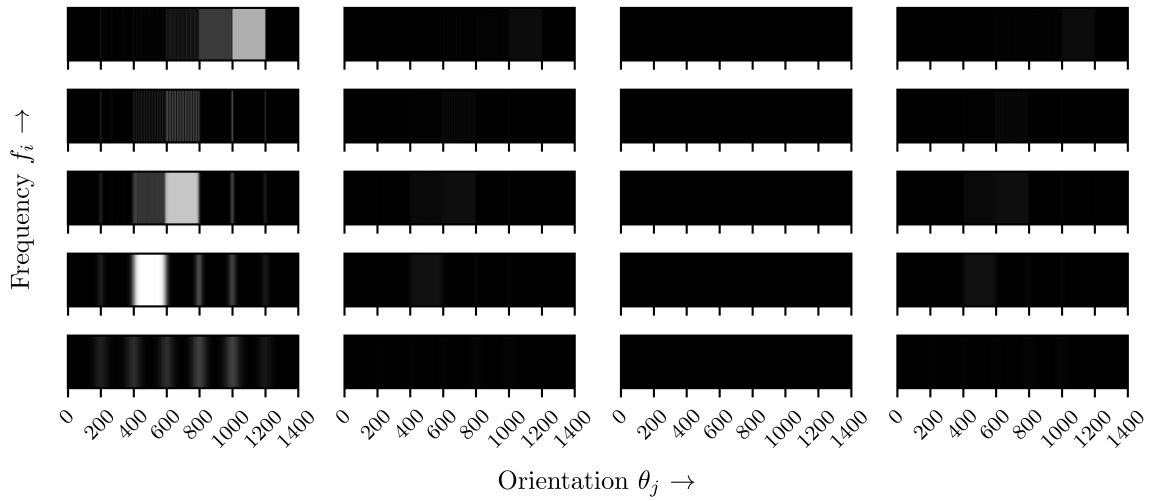
$$\sigma = \max(\sigma_x, \sigma_y) \tag{5.13}$$

From the analysis of Gabor filters presented in chapter 4.2, we know that the standard deviations σ_x, σ_y (Eq. (4.38)) of a Gabor filter are defined by its center frequency f and the frequency and angular point crossing points (Eqs. (4.33), (4.48)) defined at the moment of filter bank design. Fig. 5.9 shows the Gabor responses after opening and adaptive Gaussian smoothing.

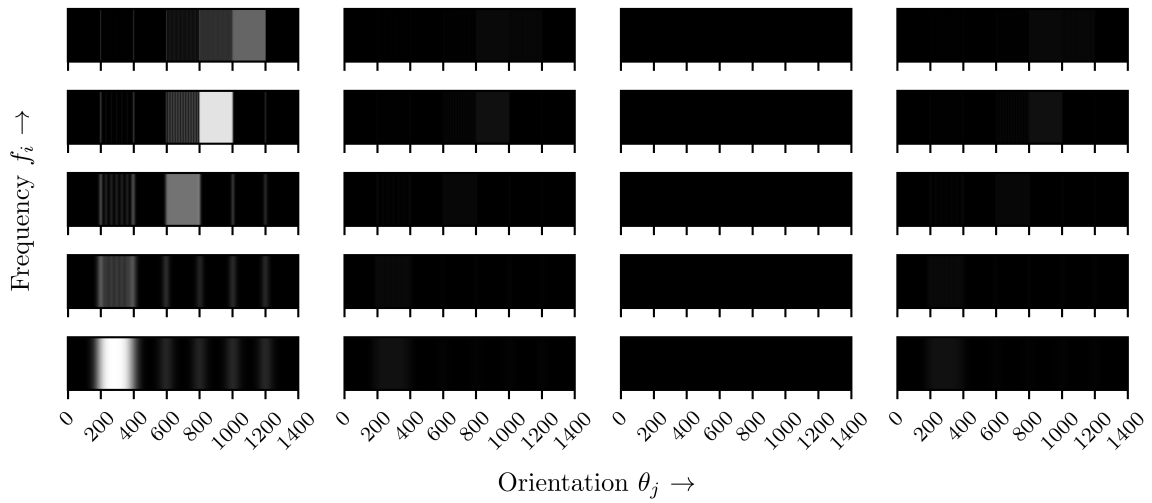
Recapitulating, Figs. 5.7, 5.8 , and 5.9 show the Power Spectral Density (PSD) decomposition of the synthetic image Fig. 5.3. In particular, Fig. 5.7 shows the raw filter responses; Fig. 5.8 shows the filter bank responses after Gaussian smoothing, and finally, Fig. 5.9 shows the array of responses after performing morphological opening. The bright areas indicate the place in the image space containing perceptual information (color or texture) in these three figures. The intensity level of the zones in the response images indicates the energy recovered by the Gabor filter at a specific orientation and frequency.



(a) $e_{L,f,\theta}(x, y)$

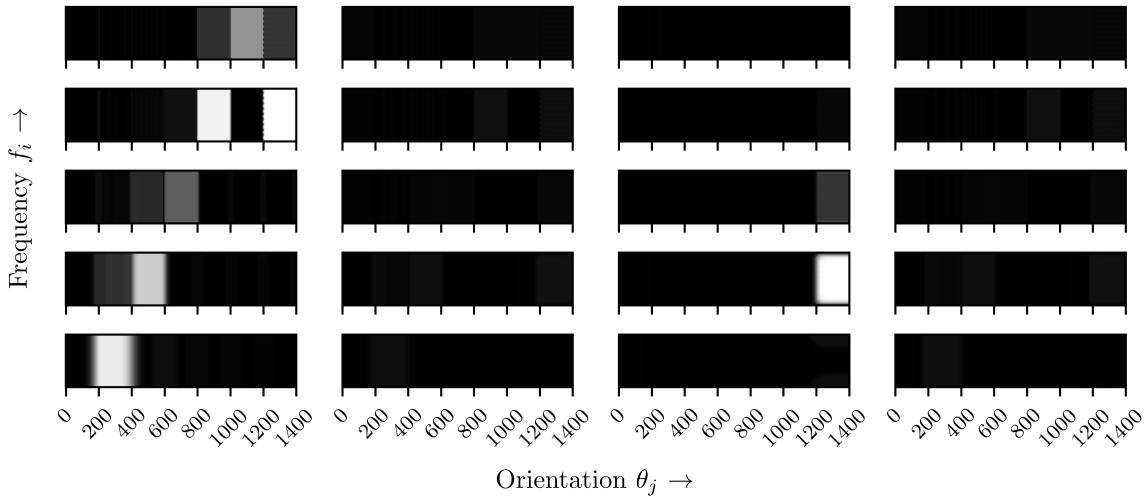


(b) $e_{\mathrm{Re}(C),f,\theta}(x, y)$

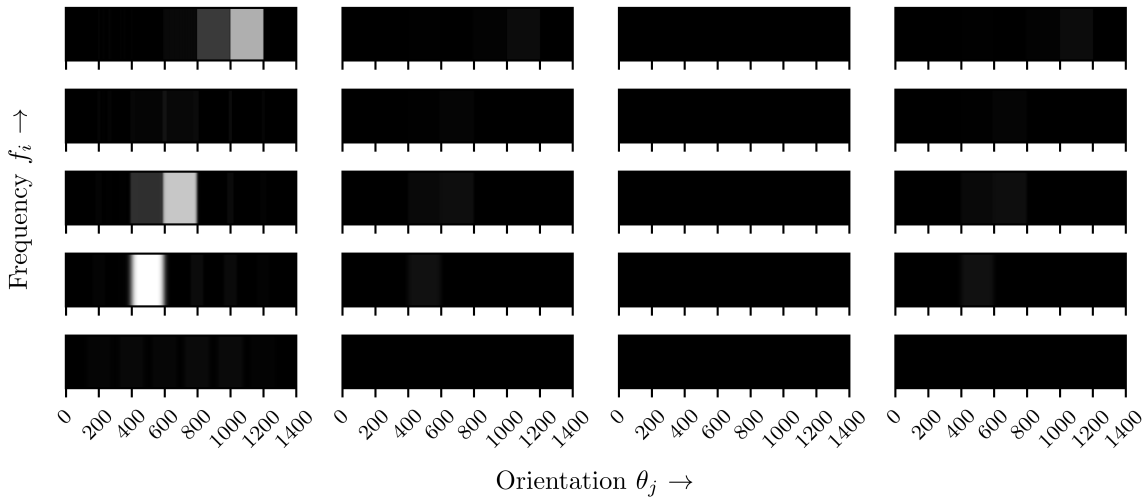


(c) $e_{\mathrm{Im}(C),f,\theta}(x, y)$

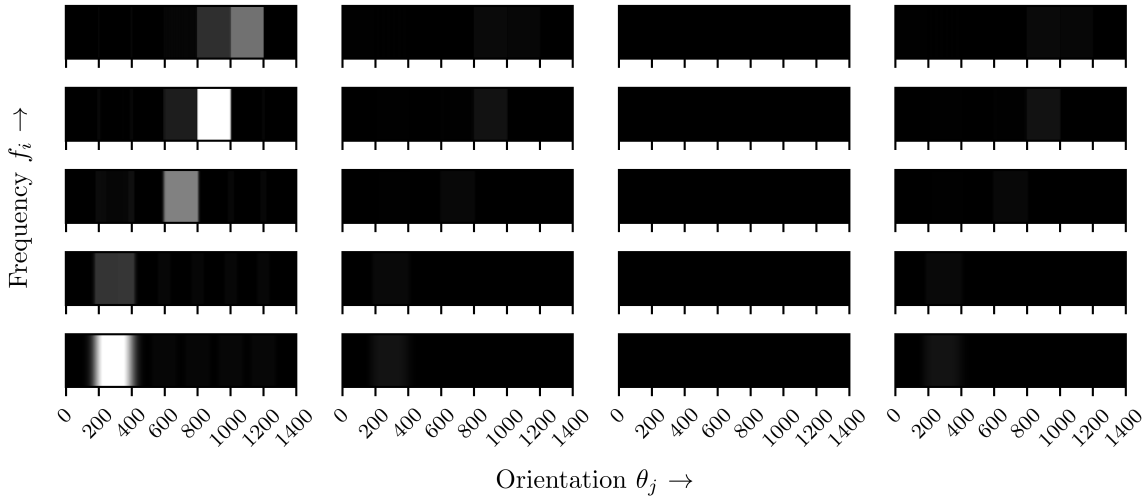
Figure 5.7: Gabor responses of the synthetic image obtained with a filter bank of 5 frequencies and 4 orientations. **(a)** Luminance channel responses, **(b)** Real part of chrominance channel responses, **(c)** Imaginary part of chrominance channel responses.



(a) $\widehat{e}_{L,f,\theta}(x,y)$

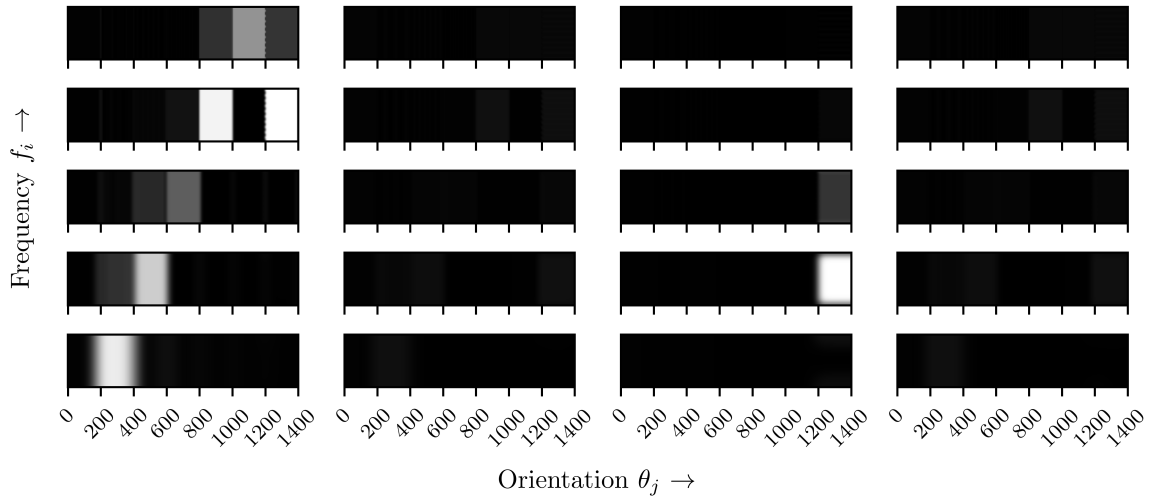


(b) $\widehat{e}_{\mathrm{Re}(C),f,\theta}(x,y)$

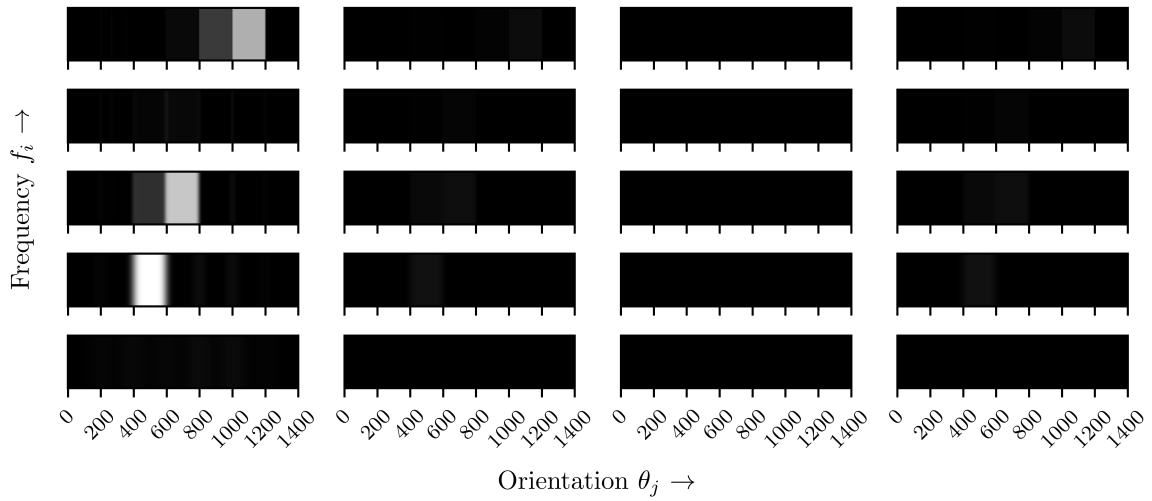


(c) $\widehat{e}_{\mathrm{Im}(C),f,\theta}(x,y)$

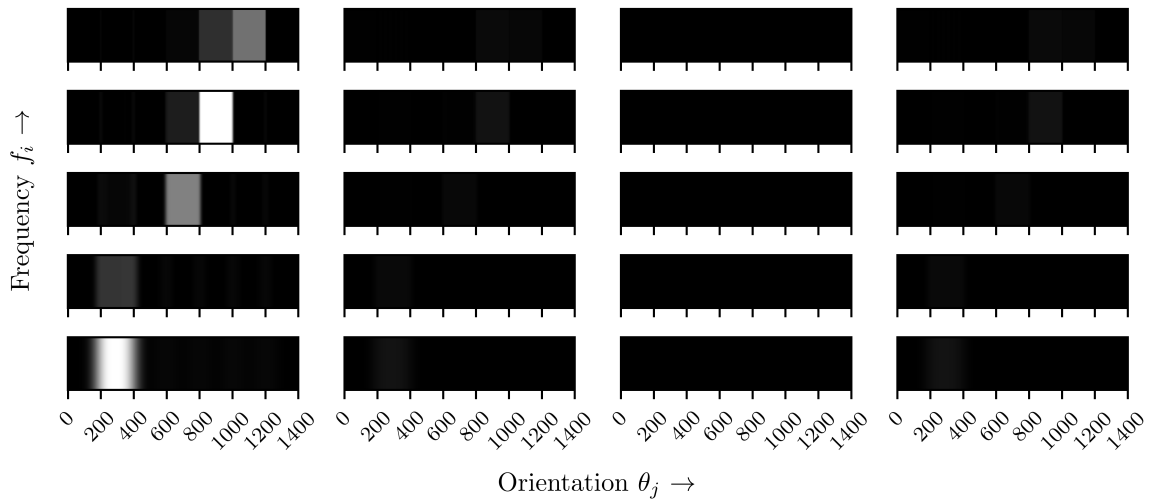
Figure 5.8: Gabor responses after the morphological opening of the synthetic image. **(a)** Luminance channel responses, **(b)** Real part of chrominance channel responses, **(c)** Imaginary part of chrominance channel responses.



(a) $\widetilde{e}_{L,f,\theta}(x,y)$



(b) $\widetilde{e}_{\mathrm{Re}(C),f,\theta}(x,y)$



(c) $\widetilde{e}_{\mathrm{Im}(C),f,\theta}(x,y)$

Figure 5.9: Gabor responses after morphological openning and Gaussian smoothing of the synthetic image. **(a)** luminance channel responses, **(b)** real part of chrominance channel responses, **(c)** imaginary part of chrominance channel responses.

5.3 Gabor Filter-based Feature Space Validation

In this section, we integrate the feature space obtained with Gabor filters within a clustering framework. We hypothesize that if the color/texture features represent the variety of information in the images (synthetic and natural), we can obtain a consistent segmentation with this perceptual information. We then use the segmentation results to qualitatively and quantitatively evaluate our Gabor feature space.

Before clustering, we adapt the feature space to prepare it for the clustering methods.

Data organization The feature space

$$X(x, y) = \widetilde{e}_{c,f,\theta}(x, y) \tag{5.14}$$

is a log-polar space given the logarithmic scale of the M frequencies and the N orientations of the Gabor filter bank. Then, the feature space is composed of $3 \times M \times N$ Gabor responses, where the constant 3 corresponds to the number of channels of the luminance-chrominance color space. We arrange the data to obtain a two-dimensional array of size $P \times D$, where $P = H \times W$ is the number of samples or pixels and $D = 3 \times M \times N$ is the number of features or dimensions of the data.

Spatial information integration Gabor's color and texture features do not include spatial information. We enter such information by adding two more dimensions to the feature space X. These two features are the positional coordinates (x, y) of each pixel in the image.

Data standardization We standardize each feature in the X matrix to have a mean of zero and a constant variance. We perform this operation to avoid the dominance of some features over others due to numerical differences of units of magnitude.

Dimensionality reduction We reduce the feature space's dimensions from $D = 3 \times M \times N$ to 5 using the linear transformation technique of principal component analysis (PCA). With the PCA we identify patterns in the feature space based on the correlation between features. We choose 5 as the new subspace dimension based on the idea that three of these dimensions contain Gabor's color and texture information, and the remaining two dimensions contain the spatial information. In addition, a low number of dimensions speeds up the calculation of clustering algorithms.

5.3.1 Qualitative Evaluation

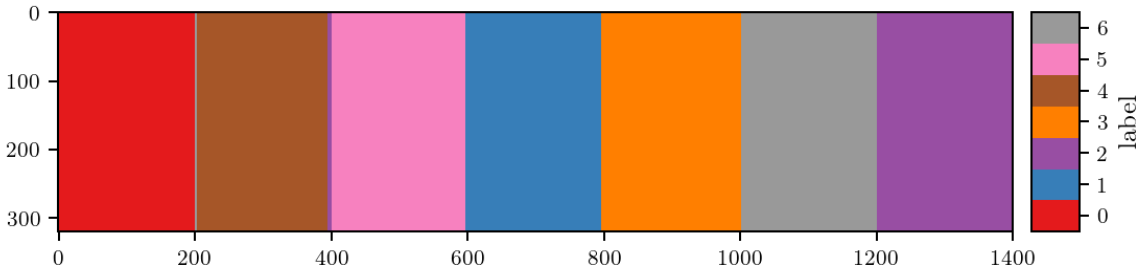
We qualitatively validate the spectral decomposition of images based on Gabor filters for the generation of color texture features first, in fully controlled conditions using the synthetic image and later, with a semi-controlled set up using handmade texture mosaics.

In both cases, we use the k-means algorithm as a clustering method on the feature space X (after spatial information integration, data standardization, and dimensionality reduction) setting the number of clusters manually. The grouping technique acts as a segmentation method, with which we validate our methodology.

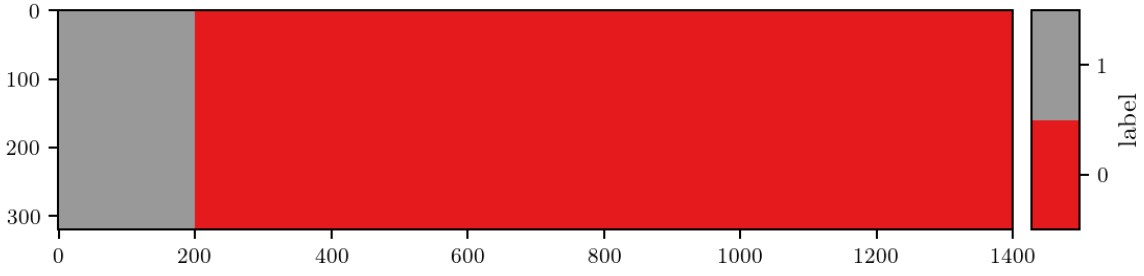
Synthetic Image Segmentation

Looking at the synthetic image (Fig. 5.3), there are several coherent ways for a human observer to segment it. The two most apparent possibilities are to segment the image into 7 clusters, where each cluster represents a region of the image and; segment the image only into 2 clusters, where one cluster groups the regions with texture and the other the flat region.

We apply these conditions to set the cluster's number of the k-means algorithm ($k = 7$ and $k = 2$) and obtain a segmentation of the synthetic image coherent with the human perception. The segmentation results are depicted in Fig. 5.10. These segmentation results show that the Gabor multi-spectral analysis captures well the color and texture information. Also, the segmentation shows that both features (color and texture) are perceptually relevant in the image segmentation task.



(a) 7 clusters segmentation



(b) 2 clusters segmentation

Figure 5.10: Synthetic image k-means segmentation results.

DTD Mosaic Image Segmentation

We go a step further and test our feature space under slightly more complex conditions; we created a series of color texture mosaics using images from the Describable Textures Dataset (DTD) [Cimpoi et al., 2014] as a base. The DTD is a collection of homogeneous nature textures with 47 annotated classes. To create the mosaics, we take 5 images of the same (or similar) class and put them together in a collage. The five classes we take the images are *lined-banded-zigzagged*, *cracked*, *braided*, *dotted*, and *striped-veined-scaly*. The texture collage we propose is relatively standard, with a circular patch in the center of the image that is superimposed on four square patches.

We apply the clustering algorithm on each mosaic created, setting $k = 5$ clusters. The segmentation results are shown in Fig. 5.11. In this case, the clustering algorithm results are coherent with the input image; however, the segmentation's precision and quality are lower than in the synthetic image. This result is mainly due to the complexity of the natural textures in the mosaics.

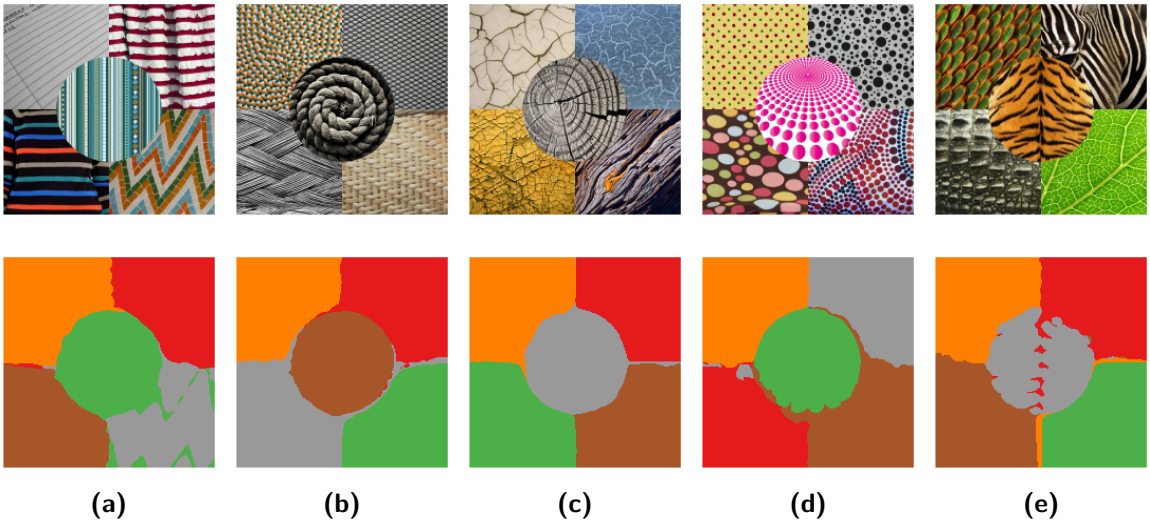


Figure 5.11: DTD mosaics k-means segmentation results.

5.3.2 Quantitative Evaluation

We also quantitatively evaluate the quality of the feature space developed in this chapter. For this, we use a database of natural images that have a ground truth generated by humans. The database and its characteristics are described below.

Berkeley Segmentation Image Data Set

The Berkeley Database for Segmentation (BSDS) is one of the gold standards for segmentation results [Martin et al., 2001]. The BSDS comprises images from the Corel database selected under a simple criterion: choose images of complex and natural scenes containing at least one distinguishable object. Under this criterion, selected

images contain multiple cues for human segmentation, for example, low-level cues such as coherence of brightness, texture, color, and contour continuity; mid-level cues such as symmetry, convexity, and area of the regions; as well as high-level cues based on the semantics of the image objects.

There are two versions of this database. The first one (BSDS300) contains 300 images, while the second (BSDS500) contains 500 images. Each image in the database contains between 5 and 11 human-made segmentations. The instruction given to the observers to naturally break the scene is simple:

> Divide each image into pieces, where each piece represents a distinguished thing in the image. It is important that all of the pieces have approximately equal importance. The number of things in each image is up to you. Something between 2 and 30 should be reasonable for any of our images [Martin et al., 2001].

Following these instructions, most segmentations meet the criterion of the number of segments; however, we can also find exceptions with more than 50 segmented things.

Finally, both databases (BSDS300 and BSDS500) contain segmentations of gray level and color images. Since in this chapter we analyze the color textures, we mainly use the BSDS500 color images with their respective segmentations to evaluate our segmentation results.

Scores

We use the human-generated segmentations of the BSDS500 as ground truth (GT), applying the precision-recall framework of Martin et al. [2004]. The precision is the fraction of detections that are true positives rather than false positives, while the recall is the fraction of true positives that are detected rather than missed. This evaluation framework is generally applied to evaluate contour detection algorithms. Therefore, applied in the image segmentation task, the framework involves evaluating the boundaries of the segmentation resulting regions [1], considering the detected boundaries pixels as a two-classes classification problem (contour and non-contour pixels). Under this configuration, precision is translated as the number of pixels correctly labeled as belonging to the contour class (true positives) divided by the total number of pixels labeled as contours (the sum of true positives and false positives). The recall in this context is defined as the number of true positives divided by the sum of true positives and the pixels which were not labeled as contours but should have been (false negatives). The following mathematical expressions define precision and recall.

$$\text{precision} = \frac{tp}{tp + fp} \tag{5.15}$$

[1] Note that for evaluating the resulting regions of our algorithm, we do not carry out any post-processing of the resulting boundaries to correct, for example, the slightly moved contours.

$$\text{recall} = \frac{tp}{tp + fn} \tag{5.16}$$

A simple metric that captures the trade-off between precision and recall is the F-measure, which is defined as the harmonic mean between the two scores.

$$\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{5.17}$$

Experiments Set up

For the numerical evaluation, we perform the segmentation of the BSDS500 images using different clustering algorithms. In addition, we perform the segmentation in the feature space obtained with different luminance-chrominance color spaces, particularly those derived from the LAB, HSL, and HSV spaces. This series of experiments allows us to evaluate other aspects of our Gabor-based feature space, for example, the behavior of the feature space on different clustering algorithms (and vice-versa) and the performance of each clustering method in the image segmentation task. Other secondary items that we also analyze are the effect of the initial color space in the transformation to the two-channel color space, the effect of choosing the number of clusters to detect, and the computation time of the clustering algorithms.

Clustering method vs. former luminance-chrominance color space. Within the wide range of clustering algorithms that exist in the literature [Omran et al., 2007] [Sathya and Manavalan, 2011], we performed the segmentation of the BSDS images using four different clustering techniques: k-means, fast k-means, Birch, and Gaussian mixture. The choice of these techniques depends on the characteristics of the input data: a high dimensional space with a large number of observations. Such input data comes from the multi-spectral decomposition of the images using Gabor filters on the luminance-chrominance channels of the images. We then compare the performance of the different clustering algorithms on the luminance-chrominance feature spaces from the HSV, HSL, and LAB color spaces reviewed in chapter 2.

Figs. 5.12 and 5.13 show the segmentation results of two BSDS500 images. In both cases, the images contain wild animals in their natural environment, which implies that the color and texture of the fur/feathers create a mimicry that helps the animals to blend in with the scene. Animal mimicry makes segmentation a challenging problem. Despite this, we see how some configurations of color space and clustering algorithms manage to classify the pixels based on the perceptual information of color and texture encoded on the Gabor features. Particularly in these two segmentation examples, we could say that the color space that best represents the color and texture information of the image is the HSL, while the clustering method that best uses the multi-spectral features is the Gaussian mixture.

For the segmentation results shown in Figs. 5.12 and 5.13, we use $k = 4$ as the target number of clusters to find in the image. In addition, for visualization purposes, we display the resulting clusters with the mean color of the pixels of the original image within the segmentation regions.

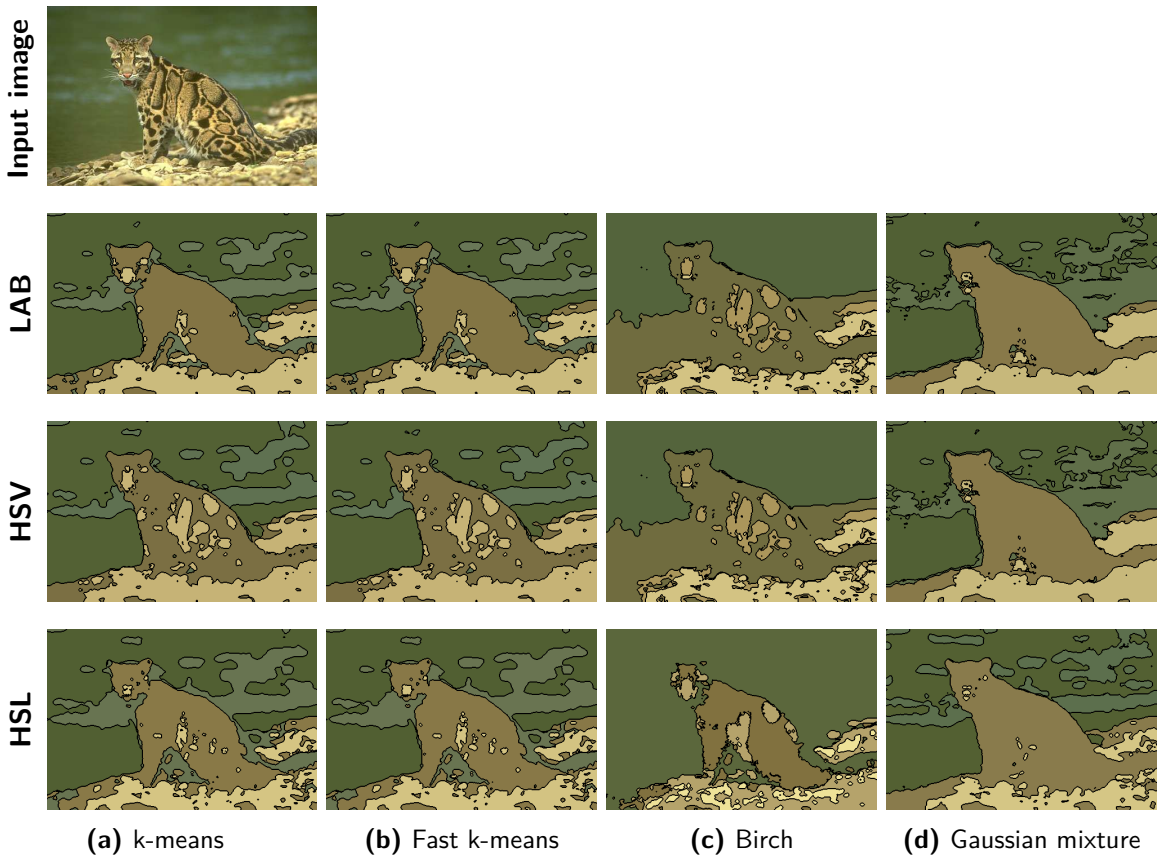


Figure 5.12: Importance of the color space and the clustering algorithm in the image segmentation task using the feature space based on Gabor filters. BSDS lynx image segmentation results.

Number of clusters in the data. Determining the number of segments to detect when using clustering algorithms as an image segmentation technique is a frequent problem. The four algorithms we present here for image segmentation need the k parameter to specify the number of clusters to find. Although there are techniques to estimate the number of regions in the image, these strategies involve one more stage of processing, which is reflected in the final segmentation calculation time. To show the influence of the number of segments k in clustering algorithms, we use the GT of the BSDS500. In the database, each image contains between $n = 5$ and $n = 11$ human-made segmentations $\mathcal{S}_i$, i.e., $\mathcal{S} = \{\mathcal{S}_i \mid i = 1, 2, \cdots, n\}$. Moreover, each human segmentation s_i could contain between $k = 5$ and $k = 100$ segments. We manually set k at fixed values of 3 and 4 segments (since most of images contains 3 or 4 objects), but also we set the number of segments as the number of maximum and minimum segments of the GT, i.e., $k_{max} = \max(\text{card}(\mathcal{S}))$ and $k_{min} = \min(\text{card}(\mathcal{S}))$.

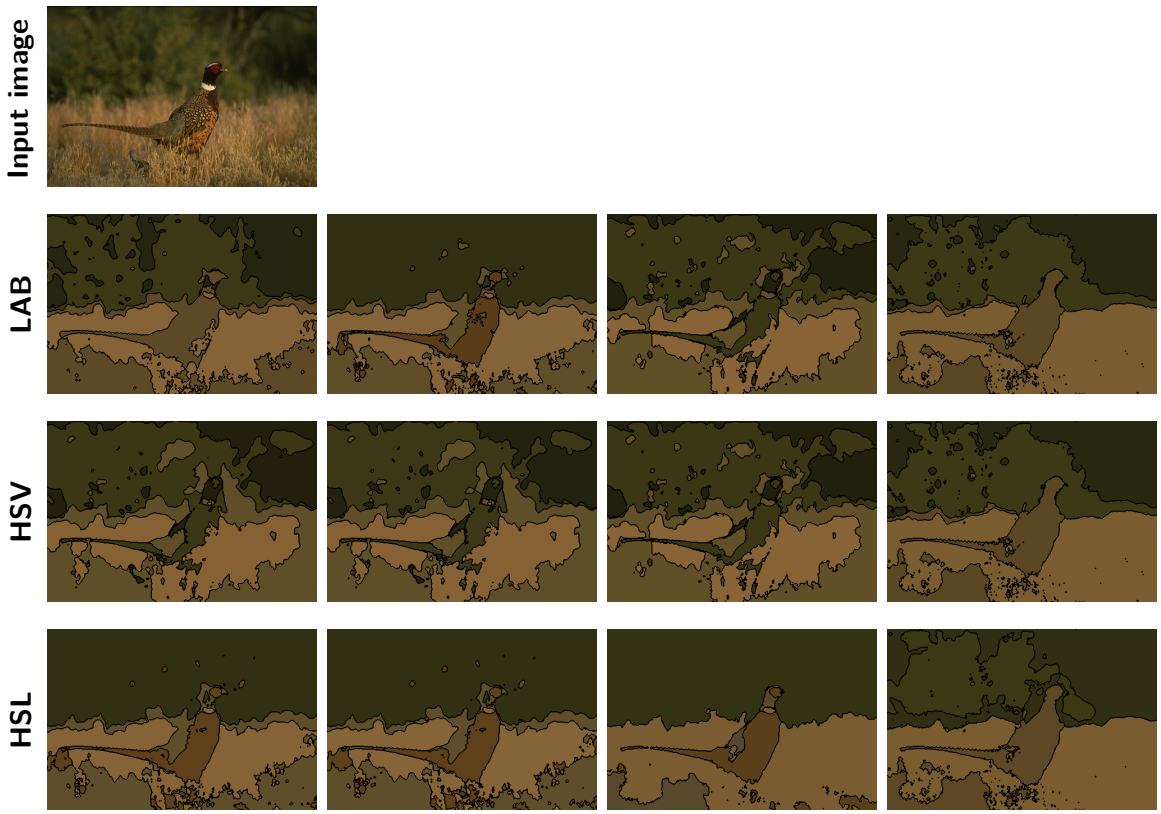


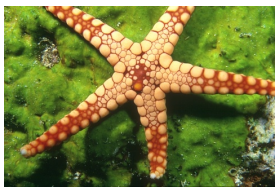
Figure 5.13: Importance of the color space and the clustering algorithm in the image segmentation task using the feature space based on Gabor filters. BSDS pheasant image segmentation results.

Fig. 5.14 shows the segmentation result of a BSDS image by manually setting k. In this particular case of the starfish image, the maximum number of annotated segments by a human observer is 92, while the minimum number of annotated segments is 6. The clustering algorithm used for this experiment is the Gaussian mixture in using the feature space from the LAB color space[2].

The image segmentation results show various phenomena. The first one is the big difference between human-made segmentations; the annotation with 92 regions implies a very detailed segmentation, while the annotation with 6 regions does not reflect the most basic segmentation of the image: two objects, the starfish and the background. Both of these extrema, despite their substantial difference, are considered as ground truth.

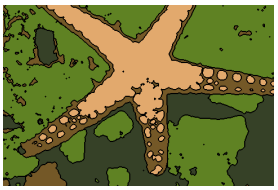
This behavior is of vital importance in evaluating a segmentation algorithm since the scores are a function of the GT. In the case of the BSDS500, increasing k means finding more regions that coincide with the GT regions, so the recall score increases; however, the precision of the regions found is very low. On the other hand, by decreasing k

[2]We chose to show the clustering segmentation results using the LAB color space because it best illustrates the influence of the number of segments in clustering algorithms. The HSV/HLS color space suffers less from this influence.

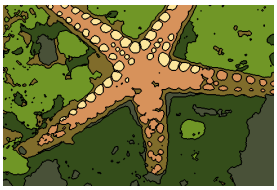


(a) Input image

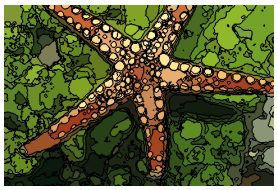
(b) $k = 3$



(c) $k = 4$



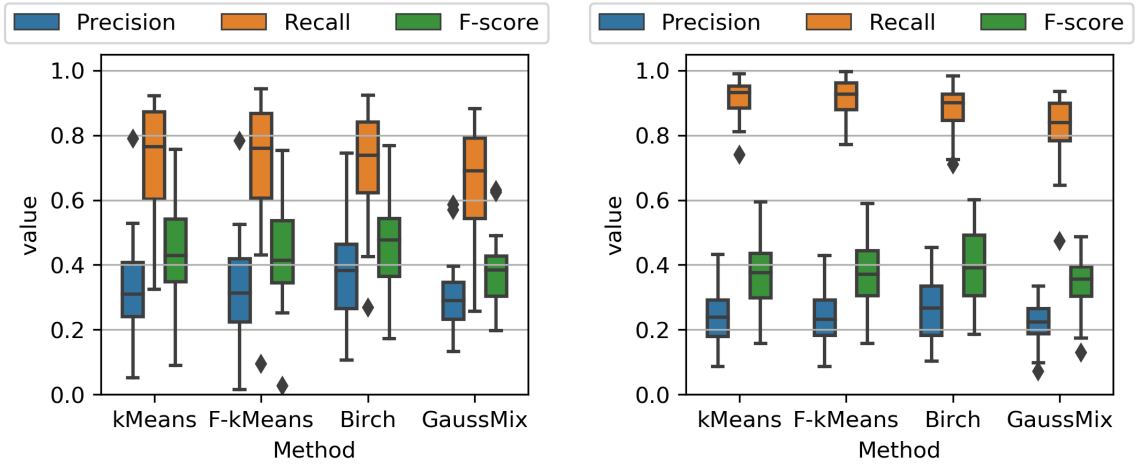
(d) k_{min}



(e) k_{max}

Figure 5.14: Effect of the choice of the number of clusters k in clustering algorithms as segmentation methods.

we detect fewer regions, which favors the precision score but affects the recall. Fig. 5.15 shows this phenomenon with the boxplots of the precision and recall scores of the different clustering methods using $k_{max} = \max(\text{card}(\mathcal{S}))$ and $k_{min} = \min(\text{card}(\mathcal{S}))$. Therefore, the optimal number of clusters k should keep a balance between maximum data compression using a single cluster and maximum precision when assigning each pixel to its own cluster.



(a) k_{min} **(b)** k_{max}

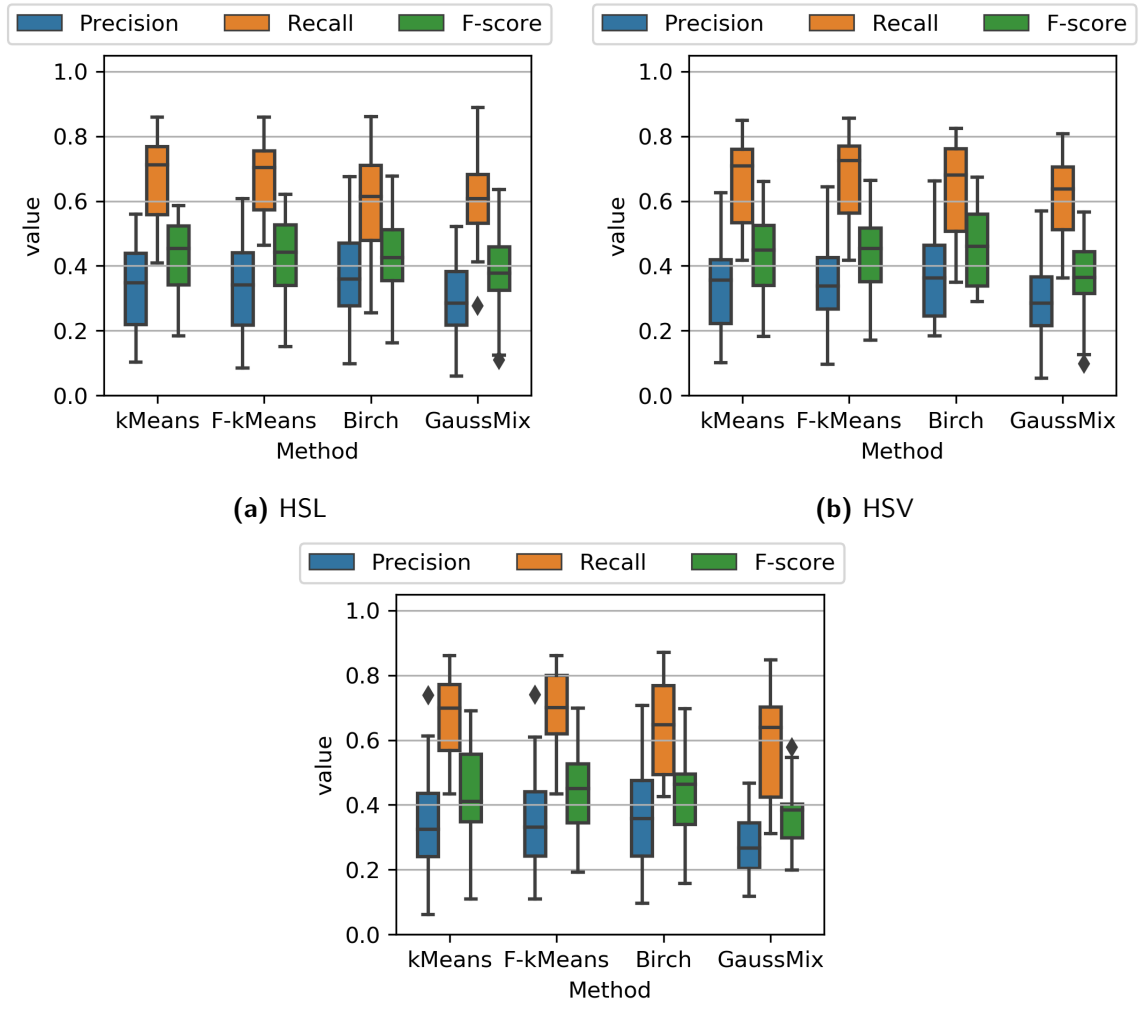
Figure 5.15: Precision-recall boxplot segmentation results.

Results

The two previous experiments show that the segmentation of real images is a complex task in which the result depends on various parameters such as the configuration of the input space, the number of segments, etc. Here we show some segmentation results (see

Fig. 5.17) using the feature space derived from the HSV-based luminance-chrominance color space and the four segmentation algorithms presented in this section.

The segmentation results are obtained by setting $k = 4$ for all images is the BSDS500 test set, where the choice of $k = 4$ comes from the idea thar most of the BSDS500 images contain at least 4 perceptually identifiable objects. Fig. 5.16 shows the precision and recall boxplots of each segmentation algorithm using the three different color spaces (HSV, HSL, and LAB).



(a) HSL **(b)** HSV

(c) LAB

Figure 5.16: Boxplots of precision and recall scores of the different clustering methods and the different color spaces. For the three plots, the number of clusters was set constant at $k = 4$.

5.3.3 High-level Texture Features

This section addresses the methodology for the extraction of high-level local texture features. To this end, we base the study of image textures on the Gabor filters. We obtain a spectral decomposition of the image through the convolution of the image with

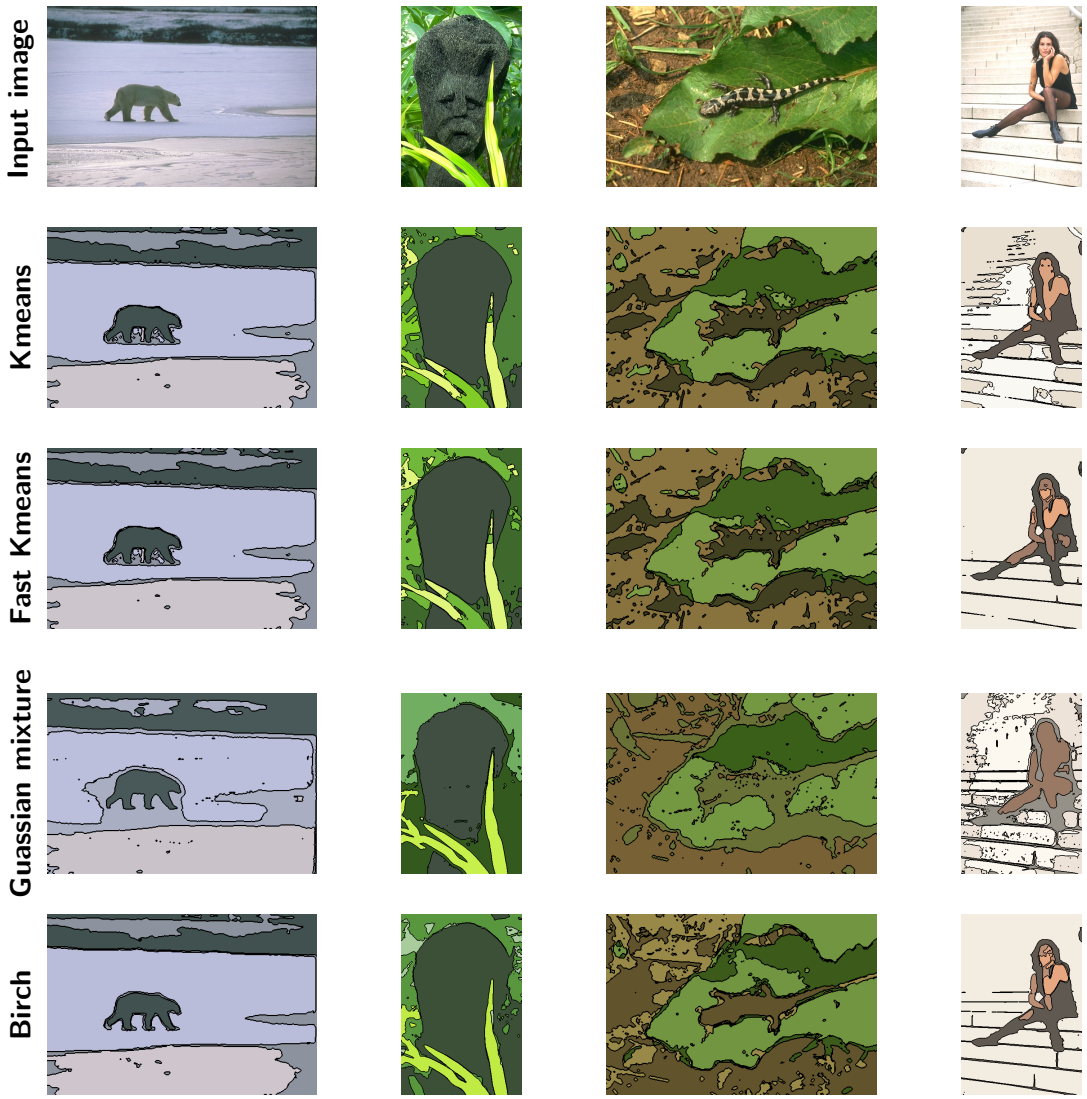


Figure 5.17: Segmentation results using different segmentation algorithms and HSV color space. The number of segments to find is fixed at $k = 4$ for all images.

the filter bank. The spectral image decomposition allows us to obtain the following high-level features, which we define and discuss below:

- Fundamental Frequency,
- Dominant Orientation,
- Maximal Response,
- Orientation Entropy,
- Orientability,
- Texturability and,
- Perceptual Window,

- Mean Color, and
- Principal Colors[3].

Fundamental Frequency

As we mentioned earlier, a texture is generated by contrast variations at a particular frequency or with a specific, repeating pattern. Although a texture may contain variations at multiple frequencies, there is only one that stands out and is more perceptive to the human eye. We call this *fundamental frequency*.

The fundamental frequency is a concept commonly used in music, acoustics, signal theory, and speech analysis [Benward, 2014], [Sigmund, 2013]. This is defined as the lowest frequency of a harmonic series representing periodic parts of a speech signal. To our knowledge, this concept has not been applied under the exact definition for image processing and texture analysis. The closest approach is that of Kamarainen et al. [2002b], who defines it as the frequency within the Gabor filter bank frequencies that gives the maximum response for each filter bank orientation. They use the fundamental frequency as a feature to characterize and recognize objects [Kamarainen et al., 2002a]; however, it is prone to failure when there are multiple objects of the same size in the image or when the objects' shapes are not precise.

We propose to obtain the fundamental frequency of textures from the image spectral decomposition $\widetilde{e}_{c,f,\theta}$ Eq. (5.12), following the definition by signal theory. The first step is to obtain the filter responses for each frequency, taking into account all the filter bank orientations. We do this procedure for each channel of the image $c = \{L, \mathrm{Re}(C), \mathrm{Im}(C)\}$ as

$$\widetilde{e}_{c,f}(x,y) = \sum_{\theta} \widetilde{e}_{c,f,\theta}(x,y). \tag{5.18}$$

From the reduced Gabor space Eq. (5.18), we can calculate the fundamental frequency $\widehat{f}_c(x,y)$ of each channel of the image as

$$\widetilde{f}_c(x,y) = \arg\max_f \left(\widetilde{e}_{c,f}(x,y)\right) \mid \widetilde{e}_{c,f}(x,y) > \frac{\max_f \left(\widetilde{e}_{c,f}(x,y)\right)}{2} \tag{5.19}$$

where $\max_f \left(\widetilde{e}_{c,f}(x,y)\right)/2$ is a threshold value that filters out the small responses generated low-level frequencies or zones without texture. The corresponding frequency to such zones is set to the zero frequency (f_0), given by the image's DC component.

Finally, the fundamental frequency for the complete image (for all three channels) is obtained as

$$\widetilde{f}(x,y) = \max_c \left(\widetilde{f}_c(x,y)\right) \tag{5.20}$$

[3]Note that the three last high-level texture features are highly related; therefore, their descriptions are joint below.

We show the fundamental frequency of the different areas of the synthetic image in Fig. 5.18. We can see how our approach recovers each zone's lowest frequency within the filter center frequencies in the figure. This effect is most visible in image zone 7 (between pixels 1200 and 1400), which contains a texture that varies at two different frequencies, $f = 1/8$ and $f = 1/32$ (see table 5.1 and Fig. 5.3). The lowest frequency in this zone is $f = 1/32$, which corresponds to the found fundamental frequency. On the other hand, the fundamental frequency for zone 1 (the yellow textureless zone between pixels 0 and 200) corresponds to the frequency zero f_0, which we obtain by filtering the image with a low-level filter such as a Gaussian filter larger than the lowest frequency filter in Gabor's filter bank.

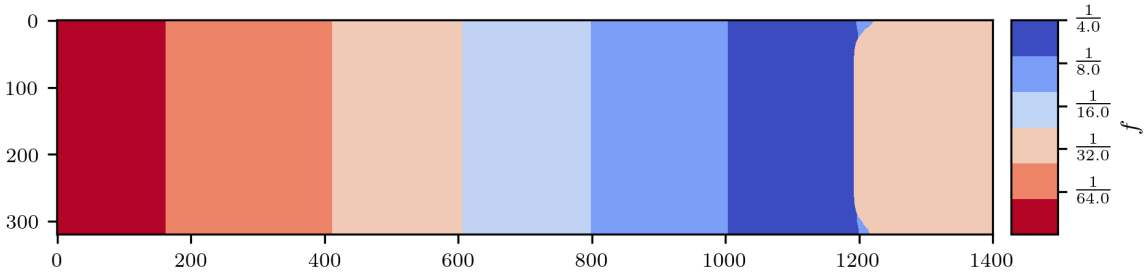


Figure 5.18: Fundamental frequency in the synthetic test image.

Dominant Orientation

Similarly, as in the frequency dimension, a texture can be generated at different orientations; however, there is an orientation in which spatial variations stand out more to the human eye. We call this orientation the *dominant orientation*.

To obtain the dominant orientation of a texture, we need first to obtain the Gabor responses along the three image channels.

$$\widetilde{e}_{f,\theta}(x,y) = \sum_{c} \widetilde{e}_{c,f,\theta}(x,y) \tag{5.21}$$

Then, we define the dominant orientation as

$$\widetilde{\theta}(x,y) = \arg\max_{\theta} \left(\widetilde{e}_{f,\theta}(x,y) \right) \mid f = \widetilde{f} \tag{5.22}$$

The dominant orientation denotes the angle within the Gabor filter bank's orientations at the fundamental frequency that allows recovering the highest Gabor response from the image after convolution.

We show the dominant orientations of our synthetic test image in Fig. 5.19. We see the relationship between the dominant orientation and the fundamental frequency in the values retrieved for the first zone of the synthetic image (between pixels 0 and 200). Since it does not contain any texture, its fundamental frequency is the frequency

zero; therefore, the dominant orientation is random. The rest of the zones (from pixel 200 to 1400) contains textures created by vertical lines, i.e., at an angle of 0°; however, in the last zone, which contains two textures, we recover 90° as dominant orientation since the fundamental frequency is that of the texture created with horizontal lines.

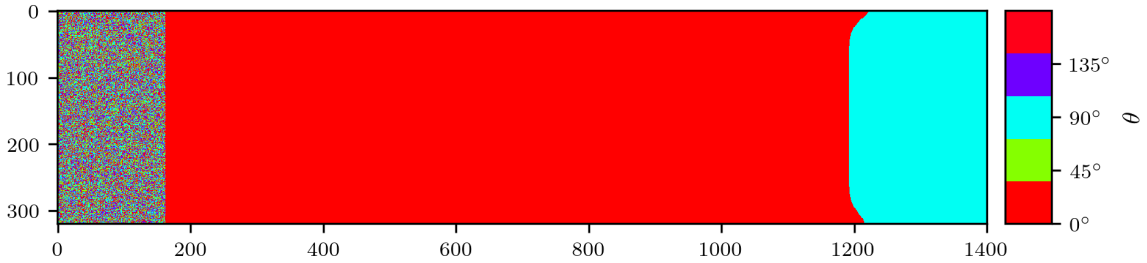


Figure 5.19: Dominant orientation in the synthetic test image.

Maximal Response

The *maximal response* is a feature that reflects the contribution of the various components of the color information (luminance and chrominance) and the texture (frequency and orientation). To correctly capture such information, we first retrieve the maximum Gabor response along with the frequency and orientation axis for each color channel, and later we add the maximum contributions of each channel of the complex color space. The expression that denotes the maximum response of the filter is

$$\widetilde{e}_{max}(x,y) = \sum_{c} \max_{f,\theta} \left(\widetilde{e}_{c,f,\theta}(x,y) \right) \tag{5.23}$$

We can see the maximal response of the synthetic image in Fig. 5.20. This feature highlights sudden dynamic changes (significant Gabor responses) and shadows textureless zones in the image. Note that the maximal response of the last zone of the synthetic image (zone with two textures) is less than the other textured areas; this is because this texture's energy is distributed between Gabor's responses with $f = 1/8, \theta = 0°$ and $f = 1/32, \theta = 90°$.

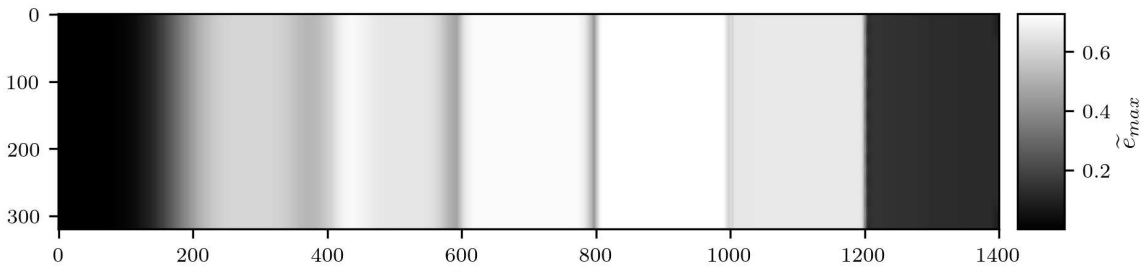


Figure 5.20: Maximum filter response of the synthetic test image.

Orientation Entropy

Initially, the concept of entropy is a measure from physics adapted to the information theory to calculate the amount of information stored in a particular signal. We adapt this measure to compute the randomness of the texture's orientation in the Gabor responses distribution. We call this feature *orientation entropy*.

We obtain the orientation entropy h by multiplying a probability vector by its logarithm. Since we use the image in a complex color space, we first obtain the entropy for each image channel as follows:

$$\widetilde{h}_c(x,y) = -\sum_{\theta} \overline{e}_{c,\theta}(x,y) \log\left(\overline{e}_{c,\theta}(x,y)\right). \tag{5.24}$$

We obtain the probability $\overline{e}_{c,\theta}$ by adding the Gabor responses along the frequency axis and dividing it by the total Gabor response along the three image channels as

$$\overline{e}_{c,\theta}(x,y) = \frac{\sum_f \widetilde{e}_{c,f,\theta}(x,y)}{\sum_{c,f,\theta} \widetilde{e}_{c,f,\theta}(x,y)}. \tag{5.25}$$

Each value of the vector indicates the probability that the Gabor filter response corresponds to a given orientation. If the vector values are of similar magnitude, the vector has no defined orientation; that is, the value response is likely to come from any filter. On the other hand, if the probability vector values are unequal, the response is very likely to belong to a well-defined orientation zone.

The orientation entropy can be normalized since we know the number of orientation angles N in the filter bank, meaning that the maximum entropy is given by

$$h_{max} = -\log\left(\frac{1}{N}\right). \tag{5.26}$$

The min value between the three image channels' normalized entropies gives the image's total orientation entropy:

$$\widetilde{h}(x,y) = \min_c \left(\widetilde{h}_c(x,y)\right). \tag{5.27}$$

This feature is helpful to identify the isotropic zones on the image. A high entropy value indicates a zone with random orientation, whereas a low entropy value indicates a zone with a well-defined orientation (anisotropic texture). We depict the orientation entropy of the synthetic image in Fig. 5.21.

Orientability

This feature is a visual property for understanding and interpreting the texture information of an image. The *orientability* is a composite feature that allows us to enhance

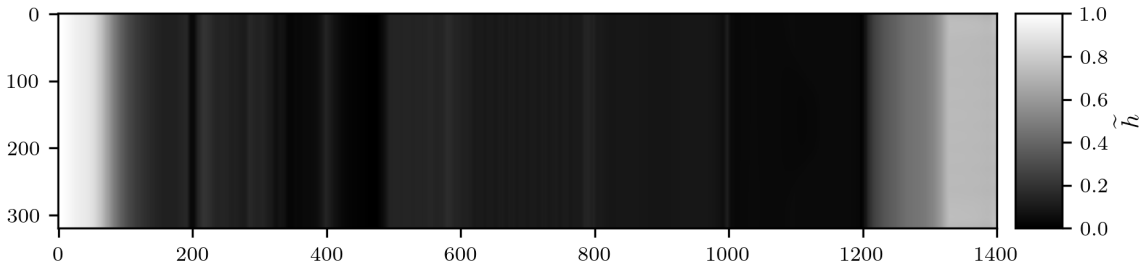


Figure 5.21: Orientation entropy of the synthetic test image.

those texture zones with a well-defined orientation angle and hide the isotropic ones.

We obtain the orientability by weighing the dominant orientation Eq. (5.22) by the orientation entropy Eq. (5.27). We depict this feature in a three-channel image. The first channel is the luminance dimension of the input image $L(x, y)$. This channel serves as a canvas to put the color given the dominant orientation $\widetilde{\theta}$, which can be represented into a cyclic colormap such as the HSV. The last channel is an alpha channel given by the opposite of the orientation entropy $\widetilde{h}$. This last channel is a transparency channel that acts as a weight for the dominant orientation. That is,

$$\text{channel } 1 = L(x, y) \tag{5.28}$$

$$\text{channel } 2 = \widetilde{\theta}(x, y) \tag{5.29}$$

$$\text{alpha channel} = 1 - \widetilde{h}(x, y) \tag{5.30}$$

We can see the orientability of the synthetic image in Fig. 5.22. This figure shows how the dominant orientation of the textureless zones (zone 1) and the zone with two textures (zone 7) are less saturated than the rest of the zones.

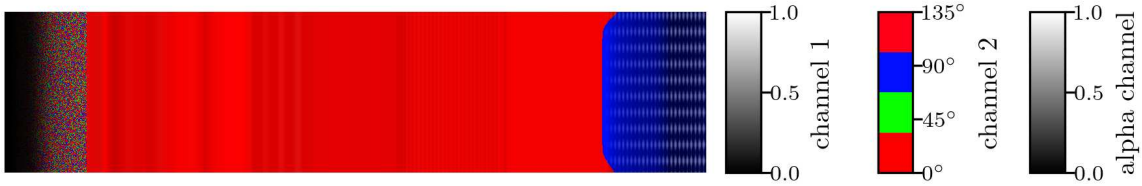


Figure 5.22: Orientability of the synthetic test image.

Texturability

The *texturality* is a visual feature that joins the fundamental frequency Eq. (5.20) and the maximal Gabor response Eq. (5.23). We represent this feature in a three-channel image. We map the fundamental frequency values $\widetilde{f}$ to a diverging color map and use it as channel 2 of the composed image. Then, the alpha channel of the fundamental frequency is given by the Gabor filter's maximal response $\widetilde{e}_{max}(x, y)$. Finally, we use

the input image's luminance channel $L(x, y)$ as a canvas to show the composite feature. That is

$$\text{channel } 1 = L(x, y), \tag{5.31}$$

$$\text{channel } 2 = \widetilde{f}(x, y), \tag{5.32}$$

$$\text{alpha channel} = \widetilde{e}_{max}(x, y). \tag{5.33}$$

Fig. 5.23 shows the texturality feature computed for the synthetic texture image. We can see how the fundamental frequency colors are shadowed by the maximal response, specifically at the textureless zone (zone 1) and the zone with two textures (zone 7).

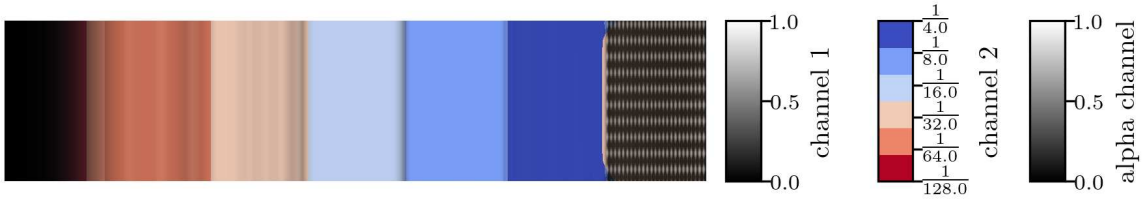


Figure 5.23: Texturality of the synthetic test image.

Perceptual Window, Mean Color, and Principal Colors

The Gabor function analysis in chapter 4.2 for designing an optimized Gabor filter bank includes the computation of an adaptative Gaussian envelope (cf. Eq. (4.39)). The adaptative support is a function of the Gabor function's central frequency. We state that the filter support κ of the fundamental frequency $\widetilde{f}(x, y)$ contains the most representative information about the period of the texture of each pixel in the image. Since we know each pixel's fundamental frequency, we can recover their *perceptual window*, which is the minimum window that describes a texture.

The perceptual window is given by the inverse of the fundamental frequency, such that

$$\widetilde{T}(x, y) = \frac{1}{\widetilde{f}(x, y)} \tag{5.34}$$

The perceptual window allows us to compute some other features, including the mean color and the two principal texture color-former of each window. The computation is straightforward. We take the mean value of the color pixel values inside the perceptual window to obtain the mean color. For the texture-forming colors, we apply a PCA to compute the two principal components in the window.

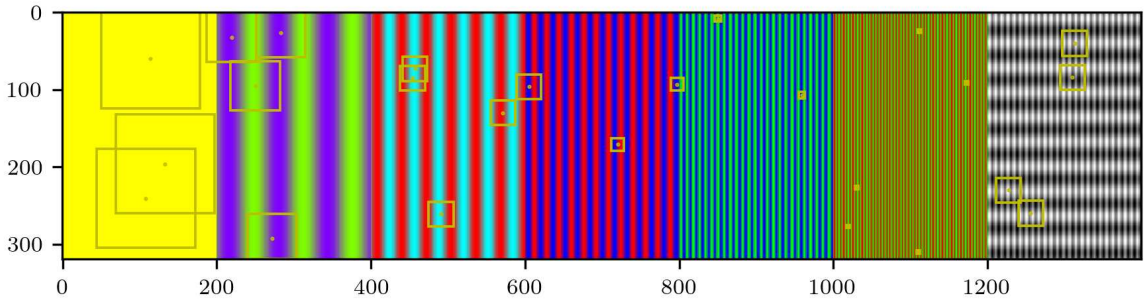
Fig. 5.24 shows the set of features resulting from the perceptual window. Subfig. 5.24b shows the window of some pixels (chosen randomly); in this subfigure, we see

how the perceptual window covers at least one period of the texture. In the two-textures zone, the perceptual window size corresponds to the period of the texture generated with horizontal contrast changes, while the yellow zone (without texture) has the largest window.

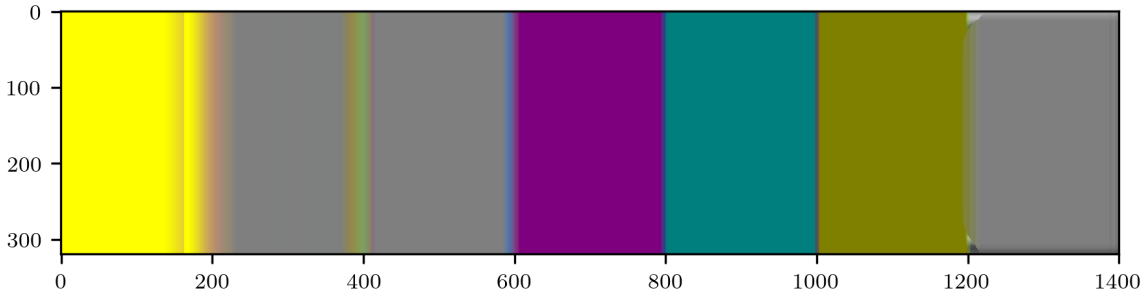
We can see the mean colors of each textured region of the synthetic image in Subfig. 5.24b. We remember that we created the synthetic image using primary colors of the chromatic circle color representation. Therefore, we can find the mean colors of each region in the complex chromatic circle. For example, for zones 2 and 3, where we combine the colors of the imaginary and real axis of the complex chromatic circle (red and cyan for zone 2, and green and violet for zone 3), the mean color is the color at the center of the complex chromatic circle, that is, gray. For better comprehension, we invite the reader to analyze Subfig. 5.24b together with table 5.1 and Fig. 5.4.

Finally, looking at Subfigs. 5.24c and 5.24d, we can corroborate that we can recover the synthetic image's texture-forming colors from the multi-spectral image decomposition proposed in this document (see table 5.1 for the reference of the texture-forming colors).

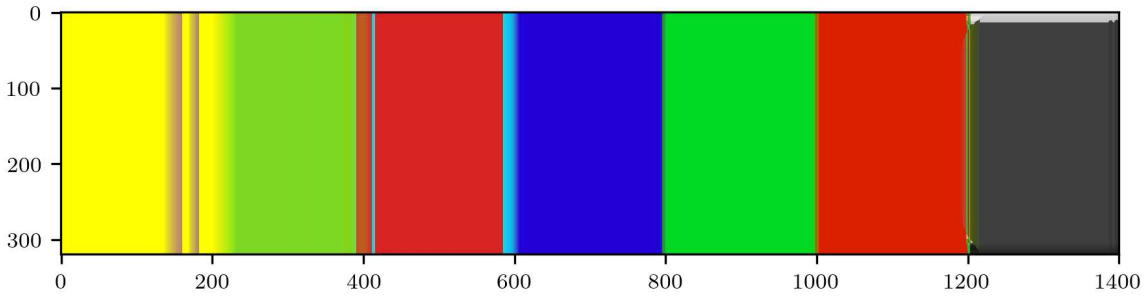
In the Figs. 5.25 and 5.26, we show the different high-level texture features presented in this section calculated for a natural image of the BSDS. In that particular example, for the black-and-white striped zebras, the expected mean color is grey, and the two texture-forming colors are black and white. Note in Fig. 5.26 how the two colors are not purely black and white when the stripes' period does not correspond exactly to the window.



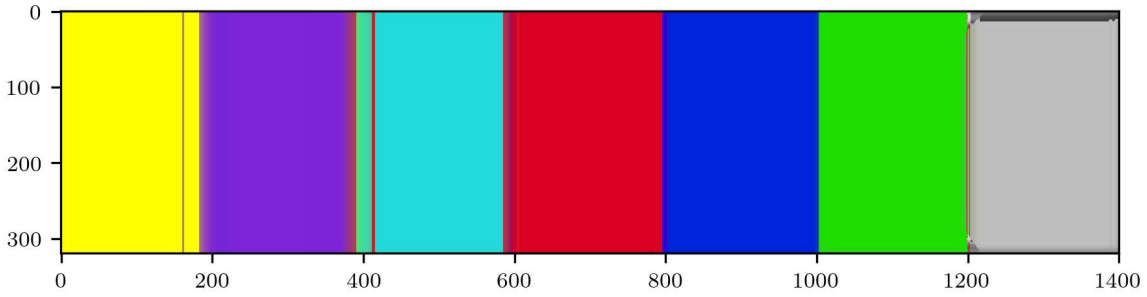
(a) Examples of perceptual windows



(b) Perceptual window's mean color



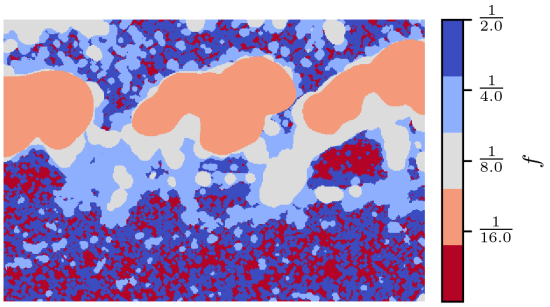
(c) Perceptual window's first texture-forming color



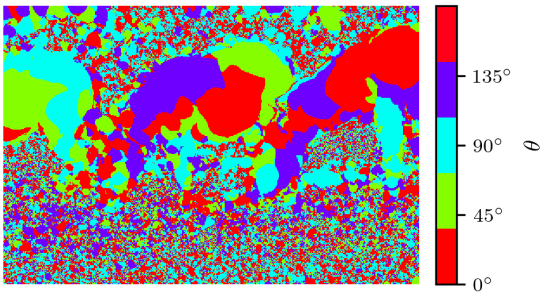
(d) Perceptual window's second texture-forming color

Figure 5.24: Synthetic image's high-level texture features derived from the fundamental frequency.

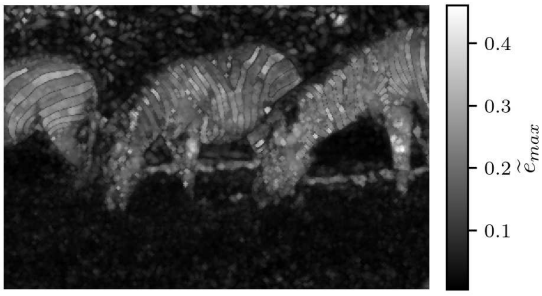
(a) Input image



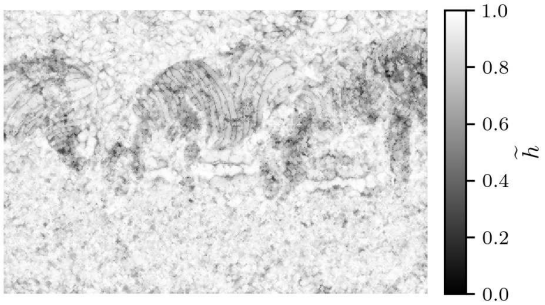
(b) Fundamental frequency



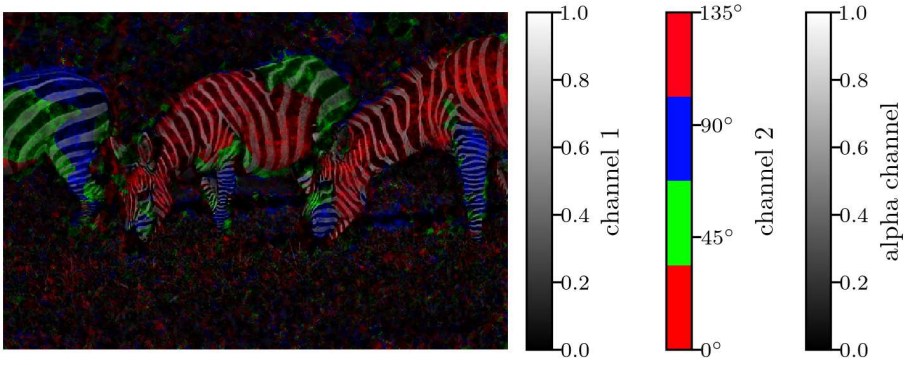
(c) Dominant orientation



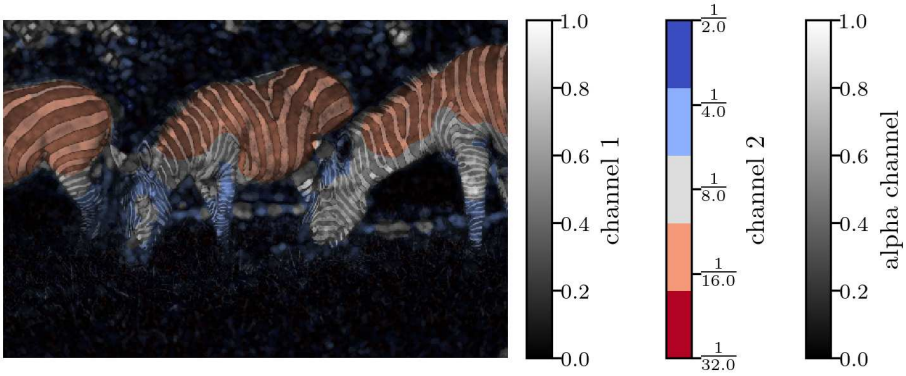
(d) Maximum filter response



(e) Entropy



(f) Orientability



(g) Texturality

Figure 5.25: High-level texture feautures computed form a natural image.

(a) Perceptual window's mean color

(b) Perceptual window's first texture color-former

(c) Perceptual window's second texture color-former

Figure 5.26: Natural image's high-level texture features derived from the fundamental frequency.

5.4 Conclusion

This chapter has shown a methodology for constructing a feature space that considers color and texture information. Our model's basis is the optimized Gabor filters and the multi-spectral decoding of the image in a color space that reflects the luminance and chrominance. We validate, qualitatively and quantitatively, the proposed model through a series of experiments where we use some clustering algorithms as a strategy for the segmentation of synthetic and natural images.

Although they do not yield the best scores in the precision and recall framework of boundary detection, the segmentation results obtained in this chapter show that the feature space captures the perceptual information of the image.

Using clustering algorithms as a technique for color image segmentation tasks has several disadvantages. One of them is the need to define the number of clusters in which the image is segmented. Also, the features space's high dimensionality means that not all clustering methods are compatible for implementation.

Finally, in section 5.3.3, we propose a set of high-level features based on the Gabor energies recovered from the image's real and complex channels that also show the richness of our features space. Despite the results obtained, the calculation of high-level texture features presents some limitations; in particular, the mean and the two texture-forming colors are not yet very precise; when the frequency does not correspond to perceptual window size, the mean color fluctuates. Also, using the PCA locally is costly, and for the moment, is a prohibitive factor for time-critical applications. We do not use the high-level features in the following parts of the manuscript for further image analysis nor segmentation; however, the interesting results obtained merit further investigation, probably for their use in a specific application. This topic appears as one of the perspectives of this work.

Chapter 6

Perceptual Object Segmentation Model

Résumé

Ce chapitre utilise des concepts développés tout au long de la thèse pour générer un modèle de segmentation d'image non supervisée. Le modèle est basé sur la décomposition multispectrale de l'image transformée en un espace colorimétrique luminance-chrominance. Nous représentons l'image sous forme de graphes, et avec la métrique EMD, nous générons un gradient perceptuel à partir duquel nous obtenons les limites perceptives de l'image. Les contours et segmentations résultant de la méthodologie proposée sont comparés aux différents travaux présents dans l'état de l'art.

Abstract

This chapter uses concepts developed throughout the thesis to generate a model for unsupervised image segmentation. The model is based on the multi-spectral decomposition of the image transformed into a luminance-chrominance color space. We represent the image as a graph, and with the EMD metric, we generate a perceptual gradient from which we obtain the perceptual boundaries of the image. The contours and segmentations resulting from the proposed methodology are compared with the different works present in the state-of-the-art.

6.1 Introduction

In chapters 4 and 5 we have introduced Gabor filters' theoretical aspects and their use in a complex color space. Using the filter family as a measurement tool, we create a feature space that exploits the color and texture information of the images and their relationship. This feature space can be seen as the complex spectral decomposition of an image.

In this chapter, we propose a workflow to use in an image segmentation task. We have seen that it is possible to obtain unsupervised segmentation using some clustering methods on the proposed feature space (see chapter 5). However, clustering methods have the limitation of needing some a priori information, for example, the number of clusters (objects) in the image. We propose, then, to obtain a coherent segmentation of the image using the fewest possible parameters using the spectral decomposition on the image.

We present a framework that obtains a segmentation of an image from the perceptual gradient of the objects. The overall idea of this framework can be seen in the diagram of Fig. 6.1. First, we represent the image as a graph (which can be pixel-based or region-based). Next, we calculate the graph edges weights using the concept of optimal transport through the EMD (see section 3.2.2), which is a measure that reflects the true distance between two distributions. Finally, the representation of the image as an edge-weighted graph allows us to apply straightforwardly various graph-based segmentation techniques or recover the perceptual boundaries of the image in the form of a gradient image, on which we can apply some edge-based segmentation techniques.

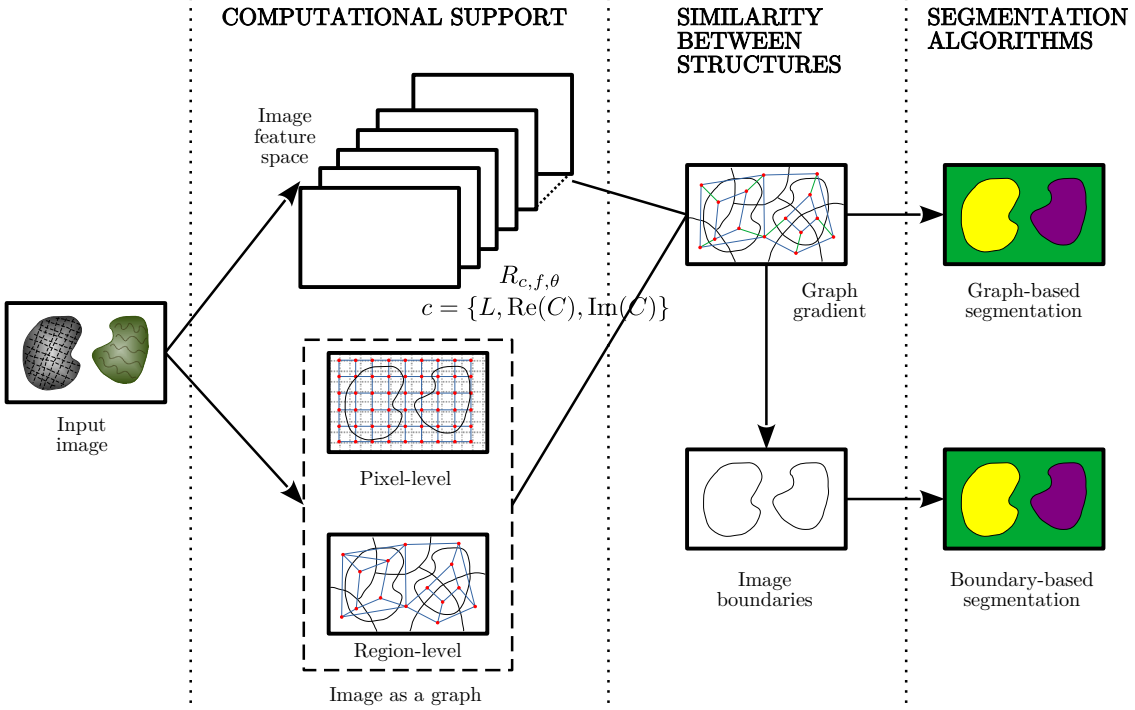


Figure 6.1: General pipeline for extraction of perceptual image boundaries and unsupervised image segmentation.

6.2 Related Work

Edge detection is a fundamental problem of computer vision that has been intensively studied since the early 1970s [Fram and Deutsch, 1975; Hueckel, 1971]. The main idea behind traditional approaches to contour detection is to model edges as discontinuities in the brightness channel of an image. This idea gave way to the creation of mask-based operators such as Sobel [Sobel and Feldman, 1990], Roberts [Roberts, 1963], Gradient [Maître, 2003] and Prewitt [Prewitt, 1970], which quantify the presence of an edge through the convolution of a gray level image with local derivative filters. Other techniques, such as the edege detector of Marr and Hildreth [1980], define edges as the zero crossings of the Laplacian of a Gaussian (LoG). The Canny operator [Canny, 1986] is one of the most popular approaches within traditional methods. This operator follows the same operation principle of the previous methods, adding a non-maximum suppression stage and hysteresis thresholding.

Despite their efficiency in controlled environments or synthetic images, traditional methods suffer from identifying contours in natural images. The edges of natural images can be present at different scales, and the colors and textures of the scene can generate edges that are perceptually significant to the human eye. Ideally, a contour detection method is intended to simultaneously exploit the brightness, color, and texture properties of an image so that it can handle the boundary perimeters defined by the brightness steps and the regions with consistent color and (or) texture.

One of the relatively recent strategies for identifying perceptual contours is to use the local energy response of an image. The operation principle is simple: generate features from the responses of an image produced by a family of filters at different scales and orientations. The idea has been exploited from different points of view. For example, the use of the Difference of Gaussians (DoG) and its Hilbert transform [Morrone and Owens, 1987; Morrone et al., 1988] to generate a family of filters. Inspired by Gabor's work, this group of filters comply with the Parseval principle and generate an exact quadrature pair (even and odd symmetry cells).

The *Probability-of-boundary* (Pb) detector [Malik et al., 2001] is one of the principal exponents of using a filter bank formed by the DoG and its Hilbert transform. The contour detector only uses the brightness and texture information to obtain the Pb. The brightness information is processed following the intervening contour framework [Leung and Malik, 1998], which consists of obtaining the image's quadrature energy, also called oriented energy (OE). The texture information is analyzed using so-called textons [Malik et al., 1999]. Since each cue (brightness and texture) has a domain of applicability, hence different units of magnitude, they introduce a gating operator based on a neighborhood's texturedness at a pixel. The operator gives, as a result, a local measure that indicates how much two nearby pixels are to belong to the same region. Later in that work, they use spectral graph theory (normalized cut algorithm [Jianbo Shi and Malik, 2000]) to segment the image into coherent texture and brightness regions.

This contemporary method, developed by the UC Berkeley research group, was the basis for many other techniques for contour detection and segmentation of natural images widely used today. Most of these works bring substantial improvements to the Pb detector. For example, the seminal papers [Martin et al., 2002] and [Martin et al., 2004] bring together previous works related to Pb and obtain a feature space of four image characteristics: localized OE, Brightness Gradient (BG), Color Gradient (CG), and localized Texture Gradient (TG). To cope with the difference in units of the magnitude of the cues, they use a logistic regression classifier to combine oriented energy, brightness, color, and texture. The proposed supervised method optimizes each feature's weights, formulating it as a two-class classification problem, where they learn the rules for combining cues from the ground truth data of the Berkeley Segmentation Dataset (BSDS) [Martin et al., 2001].

On the other hand, Ren [2008] showed that the Pb detector improves when using features of the image calculated at different scales. However, a better version of the Pb detector, which has dominated the state of the art scores for several years, is obtained by combining local and global contours [Maire et al., 2008]. The local contours are represented by the multiscale oriented signal mPb, while the global contours are represented by the oriented signal from the spectral partition sPb. The final detector that combines both signals is the globalized Probability-of-boundary gPb, which learns

the local and global part's weights through an ascending gradient, taking as reference the BSDS evaluation score.

The Berkeley research group laid the foundation for contour detection and natural image segmentation, providing the database and tools for the comparison and quantitative evaluation of the different approaches. Furthermore, Pb has motivated the development of state-of-the-art methods that use different strategies to obtain image features and contour detection. Such methods can use supervised approaches, avoiding careful filter design, computation of texture and brightness gradients, and hand-crafted features. We can also find semi-supervised methods, which generally replace the Pb detector with a supervised detector to apply later a pre-processing chain similar to that applied in the Berkeley group methods to refine the detection.

The set of the most popular supervised methods for contour detection, led by researchers from the University of Pasadena California and colleagues, is based on the calculation of features on channels of integrals of the image [Dollar et al., 2009]. Some examples of these integral channels are image color and gray channels, image responses to linear filters (e.g., Gabor filters, DoG), non-linear image transformations (e.g., Canny edges, gradient magnitude, hysteresis threshold), among others. The calculation of features on the integral channels follows the object detection framework of Viola and Jones [2004], obtaining first-order and higher-order features such as Haar-like features. Following this principle, a pool of features is obtained by randomly choosing both the integral channel and a rectangle where the features are calculated, allowing the acceleration of the computation of features and boosting learning techniques.

Some of the supervised edge detectors that use the integral channel features as input are the Boosted Energy Learning (BEL) [Dollar et al., 2006], which attempts to learn an edge classifier in the form of a probabilistic boosting tree from the thousands of simple features calculated in image patches. On the other hand, Sketch tokens [Lim et al., 2013] uses the same features as input to a random forest classifier. The peculiarity of this second method is that the classes of the classifier are the so-called sketch tokens; mid-level information patches that represent complex shapes such as joints, corners, vertical and horizontal edges, calculated from the contours of the ground truth. The Structural Edge (SE) detector [Dollár and Zitnick, 2013] and its different versions [Dollár and Zitnick, 2015] take these strategies to another level, learning not only the integral input features but also the output space, using structured-output decision forests. The Oriented Edge Forest (OEF) detector [Hallman and Fowlkes, 2015] outperforms existing supervised methods using a decision forest that analyzes local patches and outputs probability distributions over the space of oriented edges passing through the patch. On the other hand, Kivinen et al. [2014] propose a fully supervised method (DeepNet) that does not use the framework of the integral features; instead, it calculates complex-cell like covariance features from multiple scales and semantic levels, which depend on the squared response of a filter to the input image. The Object-

Contextual Representations (OCR) [Yuan et al., 2021] learns object regions and the relation between each pixel and each object region, augmenting the representation pixels with the object-contextual representation. The Convolutional Oriented Boundaries (COB) [Maninis et al., 2018] produces image contours and region hierarchies starting from generic image classification. Finally, Kelm et al. [2019] propose the RefineContourNet, a model based on the ResNet architecture that surpases the state-of-the-art BSDS500 benchmark.

Another group of approaches to contour detection are those based on sparse local coding. Such techniques are said semi-supervised because they contain two main stages, one unsupervised and the other supervised. The first stage consists of obtaining a generic representation (without information of the contours) from the image's information in an unsupervised way. The second stage consists of transforming the sparse representation of the image into a classification task, wherein the case of contours detection is a two-classes supervised problem to label the pixels as a contour or no contour. Some renowned works under this approach are the detector proposed by Mairal et al. [2008] and the Sparse Code Gradients (SCG) detector [Ren and Bo, 2012]. Both works use K-SVD as a dictionary learning algorithm and Orthogonal Matching Pursuit for efficient optimization and sparse coding of each pixel. At the end of the process, they use SVM as a linear classifier on the feature vectors resulting from the reconstruction error with each dictionary for pixel classification. The main difference between these detectors is that the SCG adopts the same scheme as the Pb detector, replacing the brightness, color, and texture gradients with sparse code gradients. Finally, the Sparse Code Transfer (SCT) detector [Maire et al., 2014] improves on the detector of Mairal et al. [2008] using a larger number of dictionaries at different scales and layers in addition to the multipath sparse coding technique, which rectifies the initial sparse codes to reconstruct the contours with an extra transfer dictionary. The main disadvantage of these semi-supervised methods is the computational time of both processes, dictionary calculation, and learning.

There is a fine line between contour detection and image segmentation. In this sense, the Pb contour detector has also influenced image segmentation. The Ultrametric Contour Maps (UCM) [Arbeláez et al., 2009] uses the gPb to define a measure of dissimilarity (ultrametric distance) between pairs of adjacent regions defined by a hierarchical segmentation operator (HSO). This technique is refined by adding a supplementary pre-processing stage using the oriented watershed transform (OWT), giving rise to the gPb-owt-ucm hierarchical segmentation method [Arbeláez et al., 2009]. The extensive qualitative and quantitative comparison of these techniques can be consulted in [Arbeláez et al., 2011].

The Pb contour detector has driven (directly or indirectly) 50 years of research work around perceptual contours detection in natural images and their segmentation. Like the Canny operator, the Pb operator has become a reference work. The importance

of this method is that it provides a reasonable basis that considers human perception principles, using operators that have a physical sense.

6.3 Image as a Graph

Graphs are mathematical structures that have been applied to almost all fields of engineering. Historically, Euler used these structures to solve a problem related to the optimal crossing of people across bridges. The success of these structures in fields such as electricity and chemistry contributed to creating a standard nomenclature, giving way to the Graph theory.

Fundamentally, a graph is a helpful structure for modeling pairwise relations between objects. In this section, we present the notation and the commonly encountered graphs in image processing applications.

6.3.1 Graph Notations and Definitions

This section introduces some critical definitions that will be used throughout the chapter related to graphs and related structures.

Definition 6.1 (Graph)**.** A graph $\mathcal{G}$ is defined by the (assumed finite) sets (V, E) in which $\mathsf{E} \subset \mathsf{V} \times \mathsf{V}$. The elements of $\mathsf{v} \in \mathsf{V}$ are called *vertices* and the elements of $\mathsf{e} \in \mathsf{E}$ are called edges. Since the edges are subsets of two nodes, we can write them as $\mathsf{e}_{i,j}$, $\{i, j\}$ or $\{\mathsf{v}_i, \mathsf{v}_j\} \quad \forall i, j \in \mathsf{V}$.

Definition 6.2 (Subgraph)**.** A subgraph $\mathcal{G}' = (\mathsf{V}', \mathsf{E}')$ is a (partial) graph of $\mathcal{G} = (\mathsf{V}, \mathsf{E})$ if $\mathsf{V}' \subseteq \mathsf{V}$, $\mathsf{E}' \subseteq E$ and $\mathsf{e}_{i,j} \in E' \Rightarrow \mathsf{v}_i, \mathsf{v}_j \in \mathsf{V}'$.

Definition 6.3 (Edge-weighted graph)**.** Given a graph $\mathcal{G} = (\mathsf{V}, \mathsf{E})$, egde weighting is a function $\omega : \mathsf{E} \to \mathbb{R}$. The weight of an edge incident to two vertices is denoted by $\omega(\mathsf{v}_i, \mathsf{v}_j)$, $\omega(\mathsf{e}_{i,j})$ or simply as $\omega_{i,j}$. We denote an edge-weighted graph as $(\mathcal{G}, \omega)$.

Definition 6.4 (Node-weighted graph)**.** Given a graph $\mathcal{G} = (\mathsf{V}, \mathsf{E})$, vertex weighting is a function $\hat{\omega} : \mathsf{V} \to \mathbb{R}$. The weight of a vertex is denoted by $\hat{\omega}(\mathsf{v}_i)$ or simply as $\hat{\omega}_i$. We denote a node-weighted graph as $(\mathcal{G}, \hat{\omega})$.

Definition 6.5 (Adjacency)**.** Given an edge $\mathsf{e}_{i,j}$ that connects $\{\mathsf{v}_i, \mathsf{v}_j\}$, the two vertices $\{\mathsf{v}_i, \mathsf{v}_j\}$ contained in the edge are said to be *adjacent* or *neighbors*. In the same way two edges that share a vertex are *adjacent*.

Definition 6.6 (Neighborhood)**.** Given a graph $\mathcal{G} = (\mathsf{V}, \mathsf{E})$, a neighbourhood N_i is the subgraph of $\mathcal{G}$ containing all adjacent vertices of v_i.

Definition 6.7 (Adjacency matrix)**.** The adjacency matrix of a graph $\mathcal{G} = (\mathsf{V}, \mathsf{E})$ is a $|\mathsf{V}| \times |\mathsf{V}|$ matrix $A_{\mathcal{G}}$ that indicates whether pairs of vertices are adjacent or not. For undirected graphs, it is a symmetric $(0, 1)$-matrix with zeros on its diagonal such that

$$A_{\mathcal{G}} = (A_{ij})_{(i,j)\in\{1,\cdots,n\}^2} \quad \text{where } n = |\mathsf{V}| \text{ is the number of nodes in } \mathcal{G} \text{ and}$$

$$A_{ij} = \begin{cases} 1 & \text{if } i, j \text{ are adjacent in } \mathcal{G}, \\ 0 & \text{elsewhere.} \end{cases}$$

The adjacency matrix may be transformed into a weighted adjacency matrix $W_{\mathcal{G}}$ such that:

$$W_{\mathcal{G}} = (W_{ij})_{(i,j)\in\{1,\cdots,n\}^2} \quad \text{where } n = |\mathsf{V}| \text{ is the number of nodes in } \mathcal{G} \text{ and}$$

$$W_{ij} = \begin{cases} \omega_{ij} & \text{if } A_{ij} = 1, \text{ and} \\ 0 & \text{if } A_{ij} = 0. \end{cases}$$

Definition 6.8 (Affinity matrix)**.** The affinity matrix A (also called similarity matrix) of a graph $\mathcal{G} = (\mathsf{V}, \mathsf{E})$ is a $|\mathsf{V}| \times |\mathsf{V}|$ matrix that indicates how affine or similar a pair of vertices are. For undirected graphs, it is a symmetric matrix with zeros on its diagonal such that

$$A = (A_{ij})_{(i,j)\in\{1,\cdots,n\}^2} \quad \text{with}$$

$$A_{ij} = \begin{cases} s(i,j) & \text{if } i, j \text{ are adjacent in } \mathcal{G}, \\ 0 & \text{elsewhere.} \end{cases}$$

where $n = |\mathsf{V}|$ is the number of nodes in $\mathcal{G}$ and $s(i, j)$ is some strictly positive similarity function between the points i, j.

Definition 6.9 (Degree matrix)**.** The Degree matrix $D = (D_{ij})$ is a diagonal matrix that measures the *degree* at each node $\mathsf{v} \in \mathsf{V}$ of a graph $\mathcal{G}$ such that

$$D_{ii} = \sum_{\{j|(i,j)\in\mathsf{E}\}} s(i, j)$$

Definition 6.10 (Laplacian matrix)**.** Given a graph $\mathcal{G}$ with $n = |\mathsf{V}|$ vertices, its Laplacian matrix $L = (L_{ij})_{(i,j)\in\{1,\cdots,n\}^2}$ is defined as

$$L = D - A,$$

where D is the degree matrix and A is the affinity matrix of the graph.

Definition 6.11 (Connected graph)**.** A graph $\mathcal{G}$ is *connected* when there is a path from v_i to v_j in $\mathcal{G}$, for every $\mathsf{v}_i, \mathsf{v}_j \in \mathsf{V}$; otherwise, we say $\mathcal{G}$ is *disconnected*.

Definition 6.12 (Spanning Tree (ST))**.** A graph is said acyclic when it does not contain any cycle on three or more vertices. Acyclic graphs are also called *forests*. A connected acyclic graph $\mathcal{T}$ is called a *tree*. When $\mathcal{G}$ is a connected graph, a subgraph $\mathcal{ST}$ is called a *spanning tree* if $\mathcal{ST}$ is both a spanning subgraph of $\mathcal{G}$ and a tree.

In a graph $\mathcal{G}$ there can be many spanning trees; $\widehat{\mathcal{ST}}$ denotes the set of all possible spanning trees $\mathcal{ST}$ of $\mathcal{G}$.

Definition 6.13 (Minimum Spanning Tree (MST))**.** Given an edge-weighted graph $\mathcal{G} = (\mathsf{V}, \mathsf{E})$, the cost of the spanning tree $\mathcal{ST} = (\mathsf{V}', \mathsf{E}')$ is the sum of the weights of all the edges in the tree. The minimum spanning tree of $\mathcal{G}$, denoted as $\mathrm{MST}(\mathcal{G})$ or simply $\mathcal{MST}_{\mathcal{G}}$, is the spanning tree where the cost is minimum among all the spanning trees, that is,

$$\mathrm{MST}(\mathcal{G}) = \underset{\mathcal{ST} \in \widehat{\mathcal{ST}}}{\operatorname{argmin}} \left(\sum_{\mathsf{e}_{i,j} \in \mathsf{E}'} \omega_{ij} \right).$$

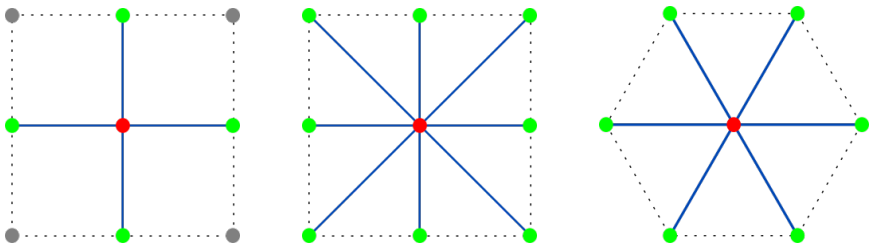
Definition 6.14 (Graph cut)**.** A *cut* $\mathcal{C}_{\mathcal{G}} = (\mathsf{S}, \mathsf{T})$ is a partition of the vertices V of a graph $\mathcal{G} = (\mathsf{V}, \mathsf{E})$ into two disjoint subsets S, T. The *cut-set* of a cut $\mathcal{C}_{\mathcal{G}} = (\mathsf{S}, \mathsf{T})$ is denoted as the set $\{(\mathsf{s}, \mathsf{t}) \in \mathsf{E} \mid \mathsf{s} \in \mathsf{S}, \mathsf{t} \in \mathsf{T}\}$ of edges that have one endpoint in S and the other endpoint in T.

Pixel-based graph representation

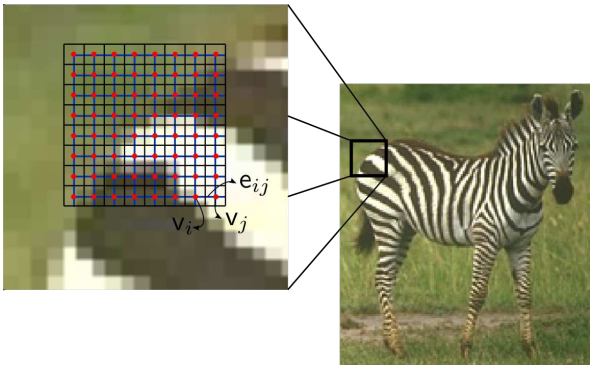
Considering a digital image as a 2-d grid of pixels, where the intensity (or color) values are mapped to the spatial coordinates (x, y), we can use the graph theory to represent all the pixels as a dense graph. In the graph notation $\mathcal{G} = (\mathsf{V}, \mathsf{E})$, each node $\mathsf{v}_i \in \mathsf{V}$ corresponds to a pixel in the image, and the edges $\mathsf{e} \in \mathsf{E}$ correspond to the junctions between adjacent pixels.

There are several strategies to join the nodes of a graph. The types of graphs that we can form are a function of such linking strategies. For example, the complete graph connects each pair of different nodes with a single edge. The epsilon-graph connects a pair of nodes if they are within an epsilon distance. The k-nearest neighbors' graph (knn-graph) connects a central node to another node only if the distance between them is among the k smallest distances from the central node to other nodes. Lastly, the adjacency graph connects only a pair of nodes if they are neighbors or adjacent. At a pixel level, this last graph is referred to as a *Pixel Adjacency Graph* (PAG).

We can define the adjacency level of the pixels to generate a specific pixel-based graph. In our applications, we mainly use the 4-neighbor adjacency system. This configuration and other adjacency systems based are illustrated in Fig. 6.2.



(a) 4-neighborhood **(b)** 8-neighborhood **(c)** 6-neighborhood



(d) Example of a 4-n graph on a real image

Figure 6.2: Most common k-nearest neighbors adjacency systems.

The representation of an image as a graph opens the possibility of new methods for data processing; however, a recurring problem is the need to satisfy the compromise between algorithmic complexity and precision. In most algorithms, complexity is a function of the number of nodes and edges in the graph, so the adjacency system plays an essential role in the speed of the methods applied to an image graph. One way to reduce the number of nodes (and consequently the number of edges) is to use graphs on the image's elements of greater size.

Region-based graph representation

To build this type of graph, we must first separate the image into regions, preferably into regions that are coherent with the image's perceptual information. Subsequently, the graph nodes represent the image regions while the graph edges follow the same strategies described above to connect the regions. The primary type of graph that we consider in this work is the *Region Adjacency Graph* (RAG).

Pixels are the smallest elements in the image. The grouping of these elements into coherent regions generates the so-called superpixels. In the following section, we introduce some of the most used methods for generating these regions, exposing their main characteristics.

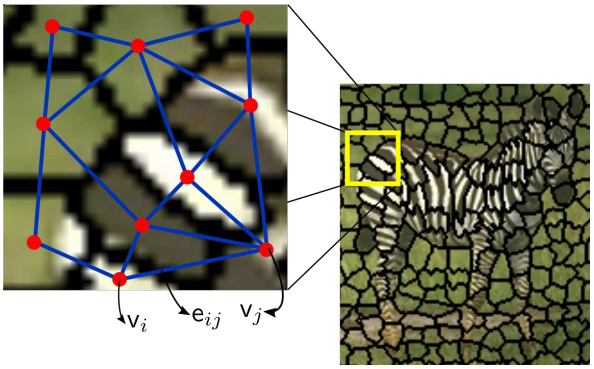


Figure 6.3: Example of a Region Adjacency Graph on a real image.

6.3.2 Superpixels

Pixels are a consequence of the discrete representation of the intensity or color of an image; therefore, pixels are not entities that naturally reflect the perceptual information in an image. Moreover, the number of pixels on an image is too high (even in moderate resolutions), making the optimization at pixel level difficult. The superpixels are locally coherent and preserve most of the structure necessary for image processing algorithms.

The term *superpixels*, introduced first by Ren and Malik [2003], describe the resulting regions of an over-segmentation image process. However, Stutz et al. [2018] gathers a series of requirements from different state-of-the-art works to differentiate superpixels from other regions generated by over-segmentation algorithms.

- **Partition**. Superpixels should define a partition of the image. They should be disjoint and assign a label to every pixel.
- **Connectivity**. Superpixels represent a connected set of pixels.
- Boundary Adherence. Superpixels must preserve image boundaries.
- **Compactness, Regularity, and Smoothness**. Superpixels should be compact (closed and bounded), placed regularly, and exhibit smooth boundaries.
- **Efficiency**. Superpixels should be generated efficiently.
- **Controllable number of superpixels**. The number of superpixels should be controllable.

Besides, according to the followed strategy to obtain the regions, Stutz et al. [2018] propose a classification for superpixels techniques. We present four superpixel techniques; each one of them represents a category of the classification.

Felzenszwalb's Superpixels

This technique belongs to the category of graph-based superpixels, representing the region generation problem in terms of an edge-weighted undirected graph.

This method treats the image as an undirected graph and produces the image partition based on edge-weights (computed as color differences or similarities) [Felzenszwalb and Huttenlocher, 2004]. We can obtain the weight of an edge ω_{ij} in two ways:

1. $\omega(\mathsf{v}_i, \mathsf{v}_j) = |I(p_i) - I(p_j)|$, the absolute intensity difference between the pixels connected by an edge;

2. $\omega(\mathsf{v}_i, \mathsf{v}_j) = |X(p_i) - X(p_j)|$, the L2 (Euclidean) distance between two corresponding pixels in the feature space.

For color images, it is possible to use option 1, considering each color channel as an individual intensity channel, and then combine the tree weights. Otherwise, the feature space of option 2 is defined by $X = (x, y, r, g, b)$, where (x, y) is the location of the pixels in the image and (r, g, b) is the color value of the pixels. This method uses a Gaussian filter to smooth the image before the edge weights computation to compensate for digitization artifacts.

With this method, we cannot directly control the number of resulting superpixels. We must use the Gaussian scale parameter to modify the size and shape of the superpixels. However, the actual size and number of segments can vary greatly, depending on local contrast.

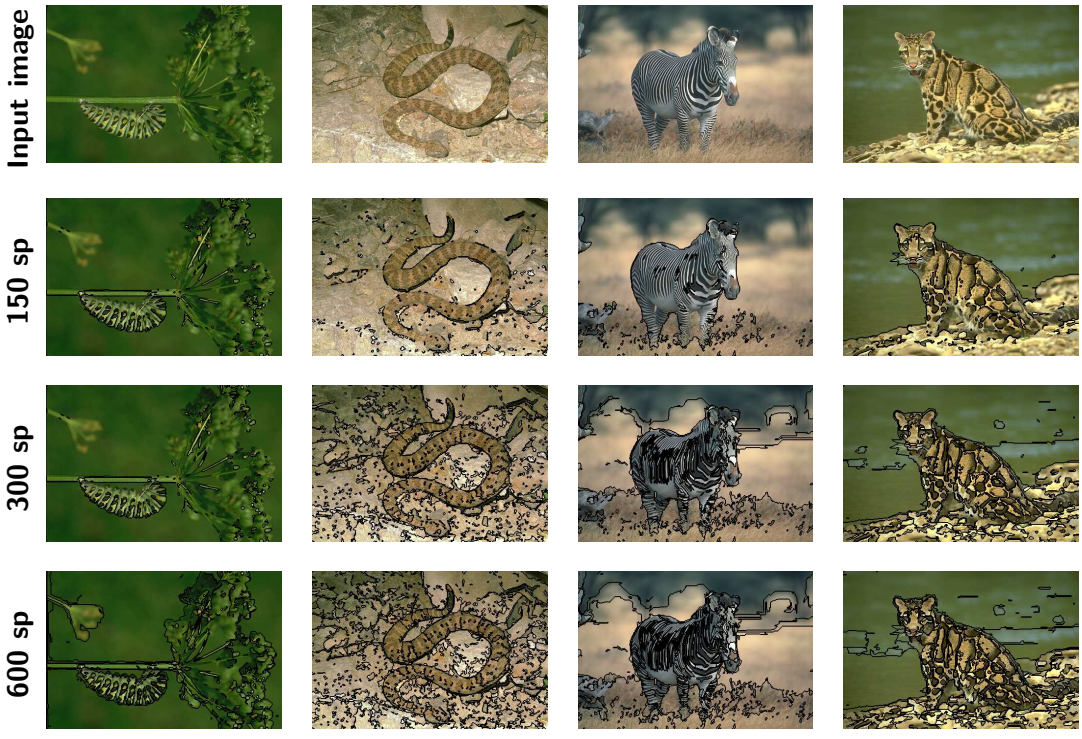


Figure 6.4: Examples of superpixels using Felzenszwalb's technique.

Fig. 6.4 shows the superpixels generated by adjusting the scale parameter to obtain approximately 150, 300, and 600 superpixels (sp). The examples show that Felzenszwalb's superpixel technique acts more like an over-segmentation algorithm since neither the size nor the regions' shape is close to being homogeneous.

Quick shift superpixels

The Quick Shift (QS) [Vedaldi and Soatto, 2008] is a density-based method for superpixel computation. This technique performs a so-called mode-seeking algorithm [Yizong Cheng, Aug./1995] for locating the maxima of a density function. It locates the maxima or the modes of a density function given discrete data sampled from that function through a mean shift procedure.

This superpixel technique is an iterative method that does not offer control over the number of superpixels or their compactness; therefore, it is also categorized as an over-segmentation algorithm.

The QS algorithm computes a hierarchical segmentation of the image at multiple scales simultaneously. The parameters to define the size and shape of the regions depend on the scale of the local density approximation and the level in the produced hierarchical segmentation. Additionally, we can also control the trade-off between color-space proximity and image-space proximity.

Figure 6.5: Examples of Quick Shift superpixels.

We illustrate in Fig. 6.5 four images and the superpixels obtained with the QS algorithm setting the parameters to obtain 150, 300, and 600 superpixels.

Watershed superpixels

The watershed-based superpixel techniques take the watershed segmentation algorithm proposed by [Meyer, 1992] as a calculation basis. Initially, the watershed method is an over-segmentation technique, i.e., one can control the number of regions employing the number of markers, but we can not control its compactness. In this context, high compactness means that the superpixels are of approximately equal size and more or less regularly shaped in the absence of strong image gradients. Some watershed-based superpixel algorithms, such as the Compact watershed algorithm [Neubert and Protzel, 2014] or the waterpixels algorithm [Machairas et al., 2015], upgrade the original algorithm adding the compactness.

Initially, these algorithms implement a seeded watershed segmentation (also called marker-controlled watershed). Markers can be determined manually or automatically using, for example, the local minima of the image gradient or the local maxima of the distance function to the background. Therefore, instead of taking a color image as input, watershed requires a grayscale gradient image, where bright pixels denote a boundary between regions.

Once the markers and the gradient are given, the algorithm views the image as a landscape, where bright pixels of the gradient forms high peaks. This landscape is then flooded from the given markers until separate flood basins meet at the peaks. Each distinct basin forms a different image segment.

Fig. 6.6 shows the superpixels obtained with the compact watershed algorithm [Neubert and Protzel, 2014] defining the number of superpixels to find at 150, 300, and 600.

SLIC superpixels

The Simple Linear Iterative Clustering (SLIC) algorithm is part of the superpixel clustering-based methods. The methods in this category group pixels into clusters (superpixels) and iteratively refine such clusters until some convergence criterion is met. In the case of the SLIC, the centers of the clusters are randomly initialized and then associate each pixel to the closest central pixel. The central clusters are subsequently adjusted iteratively until the error converges.

The SLIC algorithm allows controlling the number of superpixels in the image, the compactness of the superpixels, and the adherence to the object boundaries. However, this method cannot capture the global properties of the image [Stutz et al., 2018]. Fig. 6.7 shows, in the same way as the previous algorithms, the method's results demanding 150, 300, and 600 superpixels.

In this work, we choose the SLIC algorithm to generate the superpixels of the images. This method is one of the most competent in terms of customization and calculation time. Furthermore, it is possible to use the image in the LAB color space

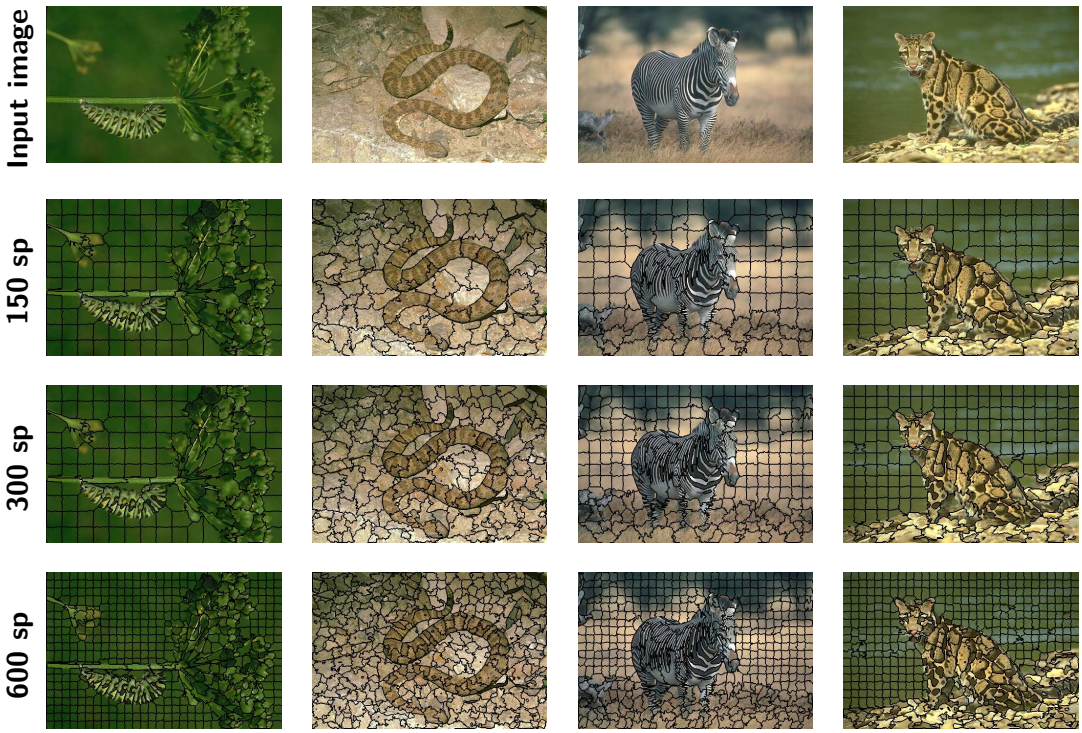


Figure 6.6: Examples of superpixels obtained with the Compact Watershed algorithm.

Figure 6.7: Examples of superpixels generated with SLIC algorithm.

to form superpixels according to the image's perceptual colors. For more details on the implementation and performance of the different superpixel algorithms, we invite the reader to review the work of Wang et al. [2017].

6.4 Graph-based Image Gradient and Segmentation

In this section, we develop the methodology to obtain a perceptual gradient on an image graph. Such an image gradient is said perceptual as it is based on the multispectral decomposition of an image using Gabor filters. We use EMD as a measure of similarity between the graph's nodes to define the edges' weight. Specifications and adaptations of the EMD are also developed in this section. Subsequently, we use the resulting gradients to segment the image using well-known state-of-the-art techniques.

6.4.1 Earth Mover's Distance for Non-normalized Distributions

In chapter 3, we have seen the utility of the EMD as a true metric for measuring similarity between distributions. In image retrieval systems based on color information, the EMD can measure tiny changes in the normalized color distributions of superhero images. On the other hand, with the texture patches, the EMD can capture, reflect and measure the importance of the logarithmic (frequency) and polar (orientation) axis of the texture signature (see Fig. 3.14 in section 3.3.4). This chapter uses EMD to measure the similarity between two elements of an image: pixels or superpixels. By measuring the similarity of elements in the Gabor-filter-based feature space, we locally measure changes in color and texture in the image.

The EMD [Rubner et al., 2000], as we used it to measure color and texture distributions in chapter 3 (cf. Eq. (3.8)), is a true metric only when used with normalized distributions (histograms or signatures), i.e., distributions with total mass equal to one. This fact has no impact when the distributions represent global information of an image (color or texture). However, the distributions of the image elements (pixels or superpixels) contain only a portion of the information; for example, in the case of our feature space, Gabor's energy.

Normalizing the individual pixels or superpixels' texture signatures leads to Gabor energy information loss, so the classic EMD cannot be applied. We use instead the EMD proposed by [Pele and Werman, 2008], which is defined as

$$d_{\widehat{EMD}}(P,Q) = \left(\min_{\{f_{ij}\}} \sum_{i,j} f_{ij}c_{ij}\right) + \left|\sum_i P_i - \sum_j Q_j\right| \times \alpha \max_{i,j}\{c_{ij}\} \tag{6.1}$$

such that, $\forall i \in \{1,\ldots,N\}$, $j \in \{1,\ldots,M\}$, the $\widehat{EMD}$ is subject to the following constraints:

1. $f_{ij} \geq 0$,

2. $\sum_j f_{ij} \leq P_i$,

3. $\sum_i f_{ij} \leq Q_j$, and

4. $\sum_{i,j} f_{ij} = \min\left(\sum_i P_i, \sum_j Q_j\right)$.

In Eq. (6.1), f_{ij} denotes the amount of mass transported from the ith supplier to the jth consumer, whereas c_{ij} denotes the transport cost from the ith supplier in P to the jth consumer in Q. The variable α denotes a penalty for extra mass transportation. If we want the resulting distance to be a metric, it should be at least half the diameter of the space (maximum possible distance between any two points). If we want partial matching we can set it to zero (but then the resulting distance is not guaranteed to be a metric). In our experiments we set this value to 1, which means the maximum value in the distance matrix (ground distance matrix) is used.

6.4.2 Graph Image Gradients

As we described earlier, we can use pixels or superpixels as support to represent an image as a graph. Fig. 6.8 shows a natural-color image, its superpixels obtained with the SLIC technique (approximately 2500 regions), and the region adjacency graph obtained on the superpixels. In this sense, each node of the graph is positioned at the centroid of each superpixel in the image. This strategy allows reducing the EMD calculation time between nodes of the graph; but also, this action generates a tradeoff between the final gradient and the regions generated by the SLIC algorithm. The pixel adjacency graph of the image is not displayed (see Fig. 6.2d instead as a PAG reference); however, we used a 4 neighboring node adjacency system (4nn) for its creation.

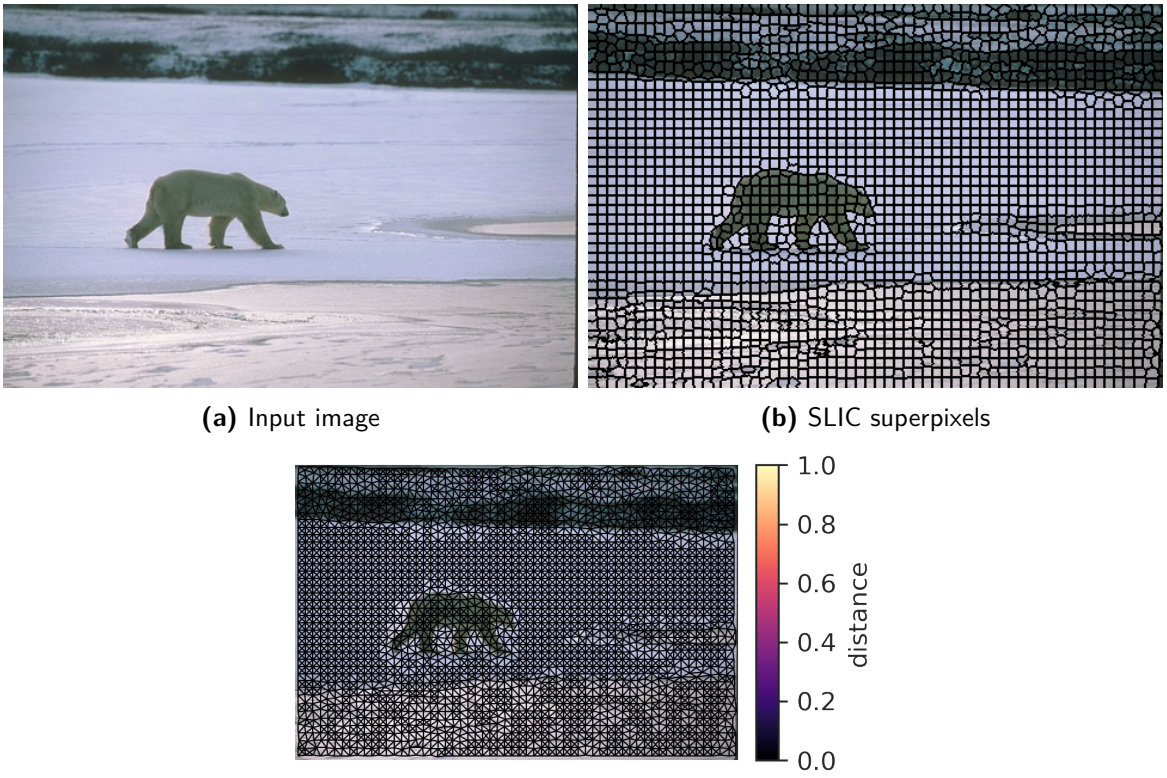
The weight of the edges between nodes of both computational supports is given by the EMD for non-normalized distributions such as

$$\mathsf{e}_{ij} \leftarrow \omega(\mathsf{v}_i, \mathsf{v}_j) = d_{\widehat{EMD}}(\widetilde{e}_{c,f,\theta}(i), \widetilde{e}_{c,f,\theta}(j)), \tag{6.2}$$

where $\widetilde{e}_{c,f,\theta}(i)$ and $\widetilde{e}_{c,f,\theta}(j)$ are the Power Spectral Density of the image at the pixel (or superpixel centroid) i, j in the image space coordinates (x, y).

In color images, we obtain a weighted edge graph for each channel c of the complex luminance-chrominance color space. Fig. 6.9 shows the weighted graphs of each channel of a natural image based on pixels and superpixels.

The gradients obtained with this methodology follow the color and texture variations of the authentic images. Furthermore, an essential aspect of our methodology is that the image's decomposition using the optimized Gabor filters and the complex color space respects Parseval's identity. In other words, the sum of all Gabor responses (in frequency axis, angular axis, and per color channel) is equal to 1. This property is



(a) Input image **(b)** SLIC superpixels

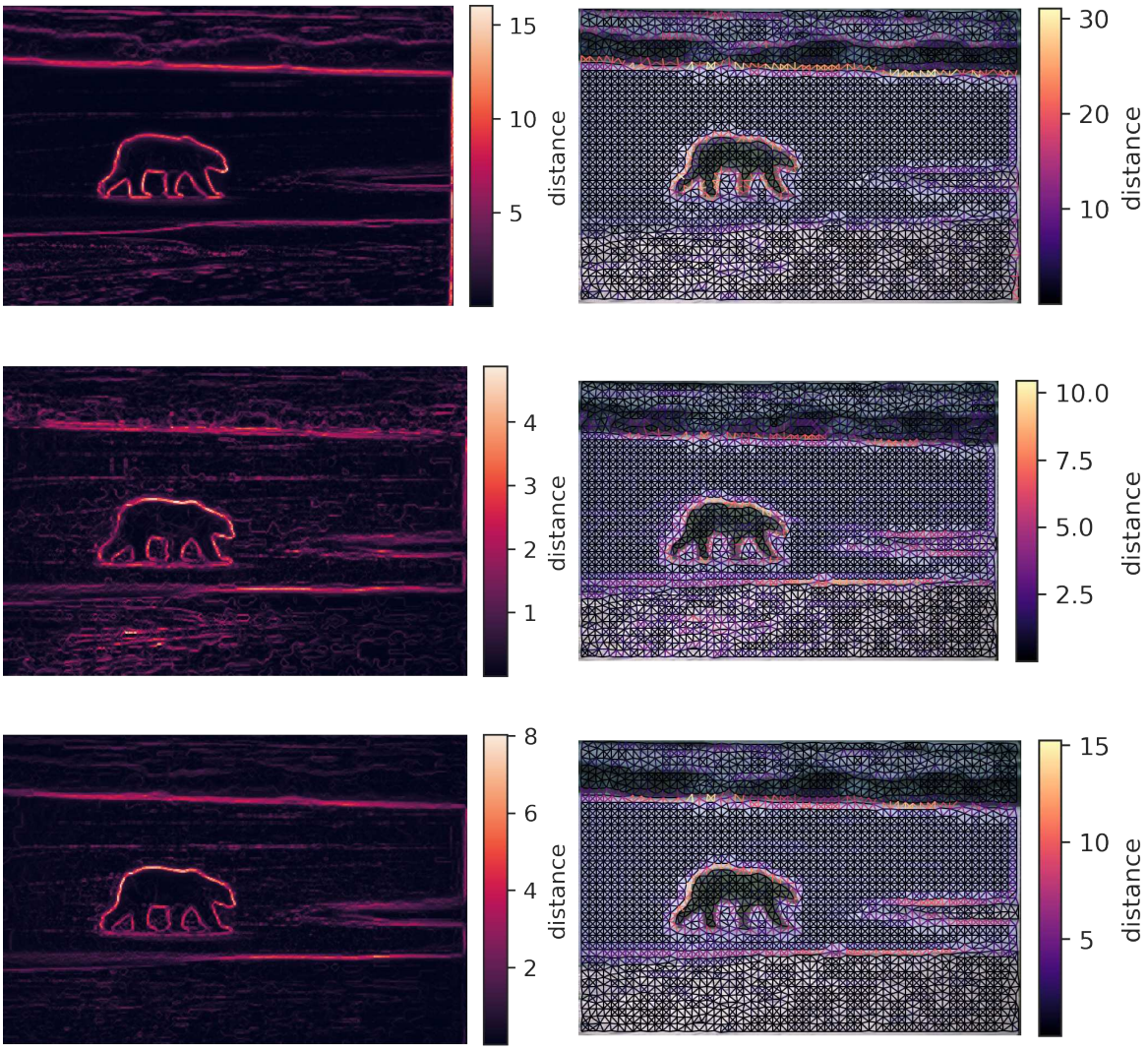
(c) Unweighted superpixel RAG

Figure 6.8: Computational support for graph-based image gradient.

reflected in the weights of the edges of the luminance and chrominance graphs. Generally, the luminance channel is the one that recovers most of the energy, while the color energy is divided into the real and imaginary parts of the chrominance. In the gradients of the second row of images of Fig. 6.9, the maximum distance between two nodes in the luminance channel is approximately 30, while in the real and imaginary part of the chrominance, the distance varies between 10 and 14. This property allows us to add the weights of the edges of each channel's graphs to obtain a single graph gradient. We can see the total gradients in the pixel and superpixel support in Fig. 6.10.

6.4.3 Image Segmentation Based on Image Gradients

The perceptual gradients obtained with EMD contain the necessary structure to segment images using graph-based segmentation methods. We use the adjacency matrix of the graph for this purpose. Here we present the methodology and results of three different segmentation methods on the weighted graphs on the superpixel support; pixel-level graphs are not considered due to the high computation time.



(a) Pixel-based graph gradient **(b)** Region-based graph gradient

Figure 6.9: Graph gradient images. First row: Luminance channel, second row: Chrominance (real part), and third row: Chrominance (imaginary part). The column names indicate the respective graph structure used to compute the gradient.

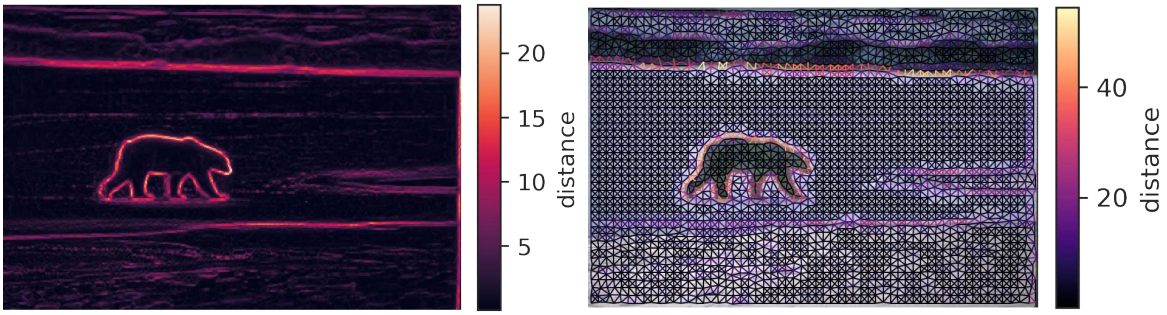


Figure 6.10: Total graph gradient images.

MST threshold graph-cut

This image segmentation method consists of obtaining the minimum spanning tree (MTS) of the total edge-weighted graph (with normalized values between 0 and 1) and

then perform a cut over the graph. Since in the MST, there is only a single path that joins all the nodes of the graph, we can make one or more graph cuts to separate the image into regions. We define the edges that are removed by the cut setting a threshold value over the weights of the MST edges. This threshold value choice is not always obvious, and this value may differ for each image.

We propose obtaining a threshold value by fitting a probability density function to the distribution of the MST edges values. We choose the log-normal distribution as the density function to fit.

We hypothesize that the areas with the same color and (or) texture information behave as flat areas in the generated feature space; therefore, the edges connecting these areas have a considerable EMD value and behave as outliers within the probability function. The threshold value is defined then by the log-normal distribution point where the edge weights reach the quantile of order $q = 0.9$. Under this setting, we cut all the MST edges above the quantile and keep 90% of the graph's edges. The segmentation of the image is given by the connected components of the resulting graph.

Taking the example of the polar bear image, Fig. 6.11 shows the MST graph of the image and the weight distribution of the MST edges (black bars), and the logarithmic distribution (red line) that matches the weighted histogram. The red arrow on the plot shows the threshold value for cutting the graph. The resulting segmentation is shown in Subfig. 6.11c.

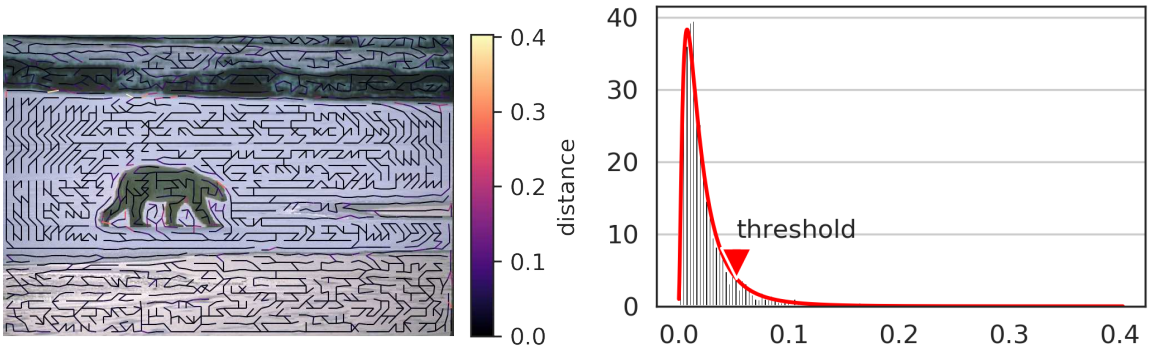
An advantage of this method is the computational speed to obtain an image segmentation. Furthermore, selecting the threshold value for the graph cut is independent of the user and self-adapting for each image. However, the choice of the quantile q is important; a very low value leads to cutting edges within flat areas of color and (or) texture, generating an over-segmentation of the image; on the other hand, a high threshold value leads to preserving most of the edges so that most of the image regions remain connected.

Spectral clustering

Spectral clustering is a technique used to cluster elements of a graph [Ng et al., 2001]. Considering the RAG weighted with the EMD over the Gabor feature space, we used this technique to group the superpixels into perceptual regions consistent with the color and texture information.

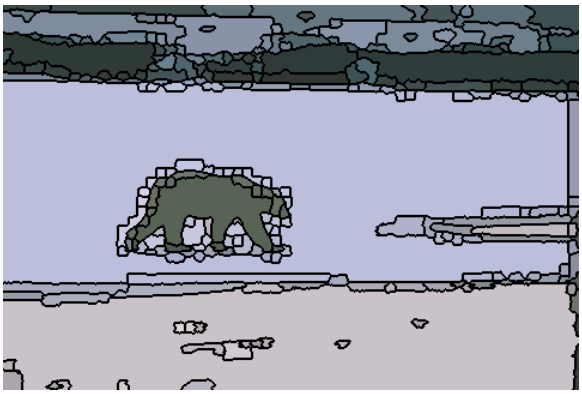
This technique transfers the data from a given domain to the spectral domain using the eigen-decomposition method. The general process of spectral clustering is mainly divided into the following three stages:

- Preprocessing: Construct a similarity matrix.
- Decomposition: Compute the Laplacian graph's eigenvectors to embed the data points in a low-dimensional space (spectral embedding).



(a) edge-weighted MTS

(b) MTS weights distribution



(c) Threshold graph-cut segmentation result

Figure 6.11: Stages of threshold graph-cut segmentation technique.

- Grouping: Assign the points in k clusters based on the new representation using the k-means algorithm.

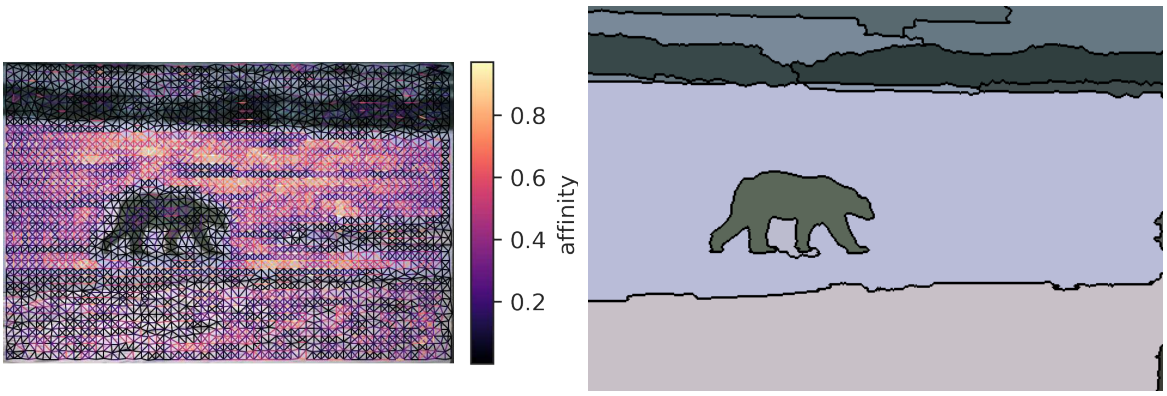
Detailing the stages of spectral clustering, we obtain the affinity matrix $A = (A_{ij})$ from the graph based on superpixels. The elements of the matrix are given by the function

$$A_{ij} = \begin{cases} \exp\left(-\alpha \frac{\omega_{ij}}{\sigma(W)}\right) & \text{if } i, j \text{ are adjacent in } \mathcal{G} \\ 0 \quad \text{elsewhere} \end{cases} \tag{6.3}$$

where ω_{ij} is the measure of distance between nodes of the graph (given by the EMD between non-normalized distributions); $\sigma(W)$ is the total standard deviation of the weight matrix W that serves a global optimization parameter for the affinity matrix computation and; α is a human-defined constant which controls how rapidly the affinity A_{ij} falls off with the distance between i and j.

The spectral decomposition is done using the Laplacian matrix, which depends on the affinity matrix A and the matrix of degree D. We follow the algorithm of Ng et al. [2001], which proposes to use the normalized Laplacian matrix $L = D^{-1/2}AD^{-1/2}$. This Laplacian matrix form is known as the non-normalized one. Once the k largest eigenvectors of L have been found and normalized, we apply the k-means algorithm on the data points of this reduced space of the graph and assign each node to a cluster.

Fig. 6.12 shows the similarity graph constructed from the EMD distances between texture signatures. Since the similarity function $s(i, j)$ is inverse to the distance $\omega(i, j)$ between nodes, we see that the edges' value tends to 0 (black edges) when the regions are dissimilar, and when the regions are similar, the edges' value tends to 1 (white edges). Subfig. 6.12b shows the polar bear segmentation result using $k = \min(\mathcal{S})$ (which for this image $k = 5$) and $\alpha = 6$.



(a) Affinity graph **(b)** Spectral clustering segmentation result

Figure 6.12: Gradient-based spectral clustering segmentation.

Normalized graph-cut

The Normalized cuts technique is a variation of the spectral clustering technique to cluster graphs. Initially proposed by Jianbo Shi and Malik [2000], this method performs the same spectral clustering steps described above with some minor differences. As for the affinity matrix, it remains the same. However, the Laplacian matrix here is given as $L_{Ncut} = D^{-1}L$ so the eigen-decomposition problem is different as well.

Furthermore, the normalized cut is a recursive method that finds a bipartition of the graph to maximize $Ncut$. This process is repeated considering the stability of the cut and if $Ncut$ is below the preset value. This condition implies that, unlike spectral clustering, we do not need to introduce a specific number of regions k to find in the image. We find more details about this method in Jianbo Shi and Malik [2000]. We show the segmentation result of the polar bear image using the normalized cut method in Fig. 6.13.

Comparison of the Different Graph-based Segmentation Methods

We evaluate the quality of the segmentation given by the three segmentation methods based on graphs: MST threshold graph-cut, Spectral clustering, and Normalized graph-cut. For this, we use the Berkeley image database test set, and the F-measure obtained from the precision and recall scores described in chapter 5.

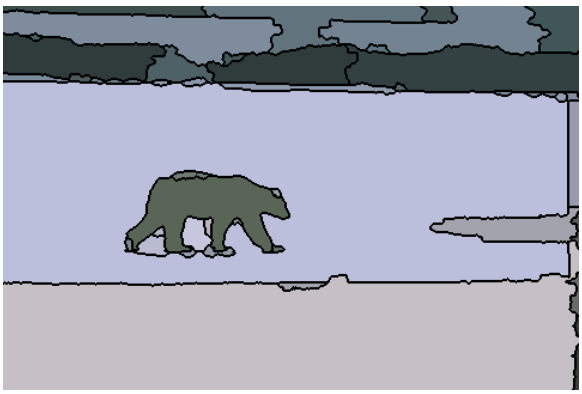
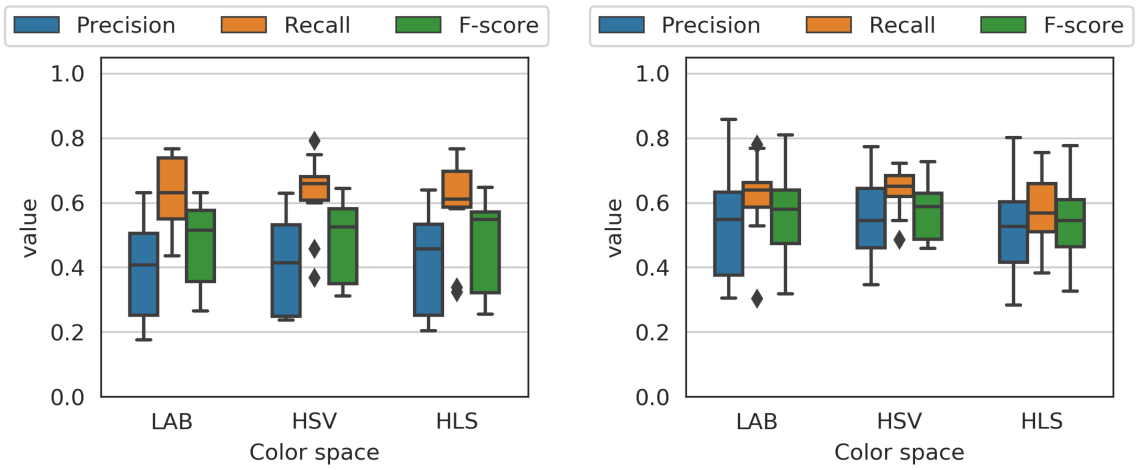


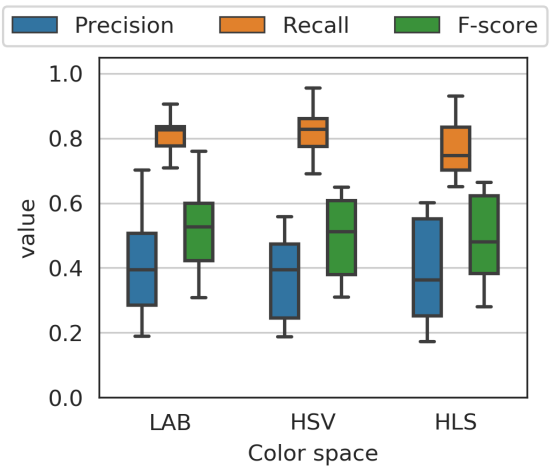
Figure 6.13: Normalized cut segmentation result.

In Fig. 6.14, we show the resulting scores applying the segmentation methods to graphs built in different feature spaces. In particular, we vary the input color space to construct the luminance-chrominance representation of the image.



(a) Threshold graph-cut

(b) Spectral clustering



(c) Normalized graph-cut

Figure 6.14: Segmentation scores of the different graph-based segmentation methods.

6.5 Image Contour Detection and Segmentation

The graph-based segmentation methods presented above showed exciting results in the segmentation task of natural color images. However, these methods require a series of parameters that are often not so easy to define, for example, the number of k regions to segment, the α parameter for constructing the affinity matrix, and the stop point of the normalized cut algorithm. Although there are various literature methods to optimize and automate the choice of such parameters, this implies a longer calculation time to obtain a segmentation. The peculiarity of these methods is that they depend on an over-segmentation in superpixels before calculating the graph's gradient. Although this step reduces the number of nodes and, consequently, the number of edges and EMD to be calculated, the contours in the image's final segmentation are part of the superpixel method's contours. In other words, if the boundaries of the regions thrown by the SLIC algorithm do not correspond to the contours of the objects in the images, these borders will not appear in the final segmentation.

This section presents the methodology to obtain the objects' perceptual contours in an image directly using the information from the edge-weighted graph instead of using graph-based segmentation methods. The main advantage of this approach is that the image's contours can be obtained at the pixel and superpixel levels. We call our contour detector Gabor-filter-based Complex Color (GCC) detector.

The procedure for obtaining the contours is straightforward. The similarity between the image elements (pixels or superpixels) is calculated by the EMD and stored in the graph's edges. So, we obtain the perceptual contours for pixel-level graphs with the transformation of the edge-weighted graph into a node-weighted graph. In such a case, each node's value (pixel) is the maximum value between the weights of all the edges connected to the node. In the case of superpixels, we transform the edge-weighted graph into a perceptual contour image by assigning the weight of the edge between two regions to the pixels that form the common border between these superpixels. Fig. 6.15 shows some examples of the contours found with this methodology in images of the BSDS500 using the graphs at the pixel and superpixel levels.

6.5.1 Contour-based Image Segmentation: Hierarchical Watershed

Image segmentation is a complementary problem to contour detection. Some methods, such as the UCM [Arbeláez et al., 2009], generate a hierarchical partition of the image from image contours. In a partition hierarchy, the image is represented as a sequence of coarse to fine partitions that satisfy the principle of causality of Koenderink [1984], that is, any partition is a refinement of the previous one in the sequence [Perret et al., 2018].

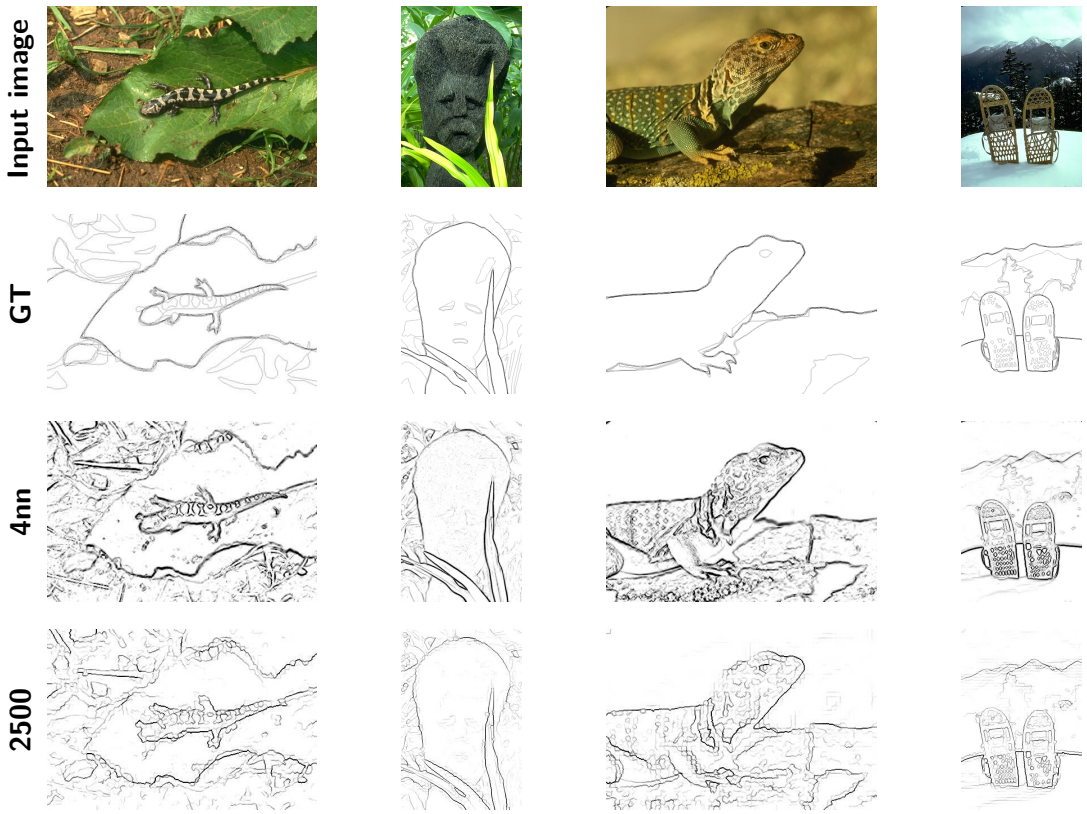


Figure 6.15: Some examples of image contours obtained with the Gabor-filted-based Complex Color (GCC) feature space on the support of pixels (4nn) and superpixels (2500) graph.

We only use the gradient information resulting from the edge-weighted graph's transformation to a boundary image (described in the previous section) in a classic morphological approach for image segmentation, the so-called watershed. This approach complies with the principle of causality, so defining a hierarchy of watersheds as a sequence of watershed segmentation of an image is possible.

One of the first authors to study the properties and relationship between the partition hierarchy and the watershed operator for image segmentation are [Najman and Schmitt, 1996]. In general terms, this method constructs a watershed by a flooding process; that is, the gradient image, seen as a topographic surface, is pierced at its minima and progressively submerged in water. The water fills the catchment basins of the minima and forms lakes. When the water of two lakes meets, the saddle point height determines the saliency of the corresponding watershed arch. The sequence of watershed segmentation is obtained by sequentially removing the minima from the gradient image according to regional attributes related to size and the contrast of the components, such as dynamics, area, or volume. In Fig. 6.16, we show the results of hierarchical watershed segmentation using the volume attribute on the GCC4nn and GCC2500 image gradients.

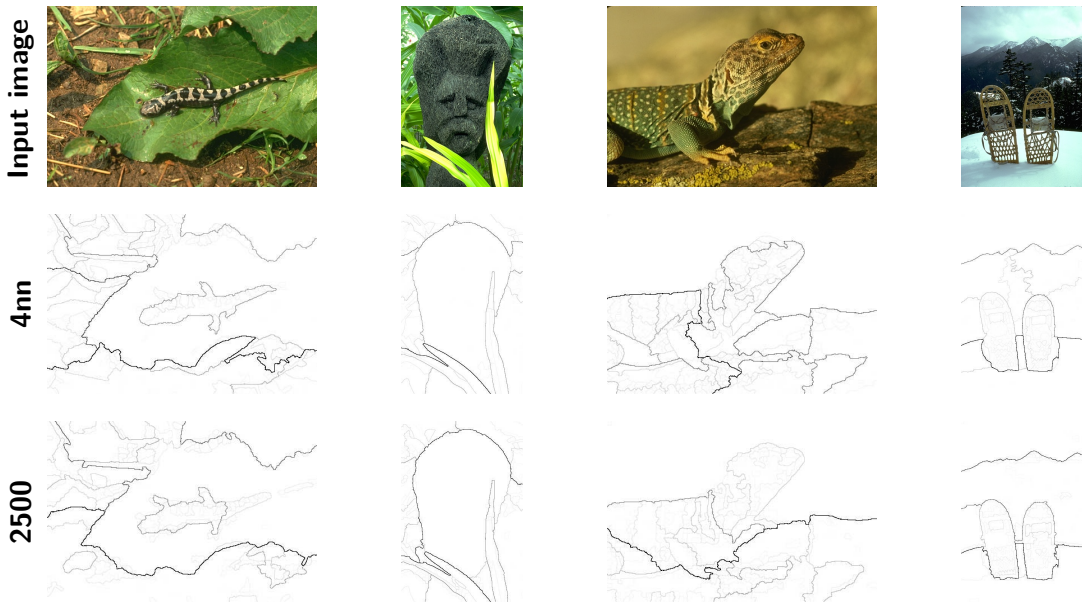


Figure 6.16: Some examples of image segmentations obtained with the hirarchical watershed method on the Gabor-filter-based Complex Color (GCC) boundaries in the pixels level support (4nn) and super-pixels (2500) support.

6.6 Comparison with the State of the Art

In section 6.2, we review the different methods for detecting contours and the segmentation of natural images. This section classifies state-of-the-art methods based on the input features and the techniques used for contour detection and image segmentation. Table 6.1 organizes the various characteristics of these methods, which allows us to position our contour detector (GCC) w.r.t. the existing works.

We recall that one of the thesis's objectives is to generate algorithms for the detection of objects in complex environments to implement them in the context of UAV tasks. In this sense, the characteristics that we look for in a contour detector are:

- The independence of databases to train learning models,
- The simplicity and a low number of parameters, and
- The possibility for real-time implementation.

Taking this into account, table 6.1 separates state-of-the-art methods into three groups: non-supervised (N-S), semi-supervised (S-S), and fully supervised (F-S) approaches. Our contour detector is positioned into the group of non-supervised methods (the first column of the table together with the Pb and the PMI methods).

Edge detector	Input features	Main approach	Segmentation technique
GCC (Ours)	Gabor lum-chr gradients	Gradients dissimilarity	Hierarchical watershed
Pb	BG, TG	Gradients dissimilarity	Normalized cuts
PMI	LUV color channels	Pixel mutual information	Spectral clustering
SCG	Gray, color, and depth channels	Sparse coding + SVM	-
SCT	RGB color channels	Sparse coding + SVM	-
$\widehat{\text{Pb}}$	BG, TG	Logistic regression	Normalized cuts
gPb	$\widehat{\text{OE}}$, BG, CG, $\widehat{\text{TG}}$	Logistic regression	UCM + OWT
BEL	Integral channel features	Gradient boosting	-
Sketch tokens	Integral channel features	Random forest	-
SE	Integral channel features	Random forest	-
OEF	Integral channel features	Random forest	-
DeepNet	Covariance-like features	Deep NN	-
ISCRA	BG, CG, TG, SIFT, Shape features, Boundary features	Cascading classifier	Region merging
COB	$\widehat{\text{OE}}$, BG, CG, $\widehat{\text{TG}}$	Convolutional NN	-

Table 6.1: Principal characteristics of the state-of-the-art works for boundary detection and image segmentation. First block: Non-supervised methods, Second block: Semi-supervised methods, and Third block: Supervised methods.

6.6.1 Scores

We use the benchmark for contour detection and image segmentation of the BSDS. This benchmark uses the precision and recall scores described in chapter 5 (cf. subsection 5.3.2). In addition to these scores, the BSDS benchmark uses the following measures and tools to compare results in contour detection and image segmentation.

Optimal Dataset Scale (ODS), Optimal Image Scale (OIS)

A hierarchical segmentation method applied on an image provides a hierarchy (H, λ), where the successive ultrametric levels $(\lambda_1, \cdots, \lambda_N)$ correspond to a series of nested segmentations $(S_1, \cdots, S_N)$. To properly evaluate it, one has to compute the score (S_i, GT) for any level λ_i of each image's hierarchy. We can then either retain the best λ-level on the overall dataset, and the corresponding score is the Optimal Dataset Scale (ODS), or retain the best level λ_i for each image and average the best individual scores for all images, which correspond to the Optimal Image Scale (OIS). By definition, the ODS is inferior or equal to the OIS. For a gradient image, the levels λ_i correspond to the threshold values at which the image contours are evaluated.

Precision-Recall curve and Average Precision (AP)

A precision-recall curve (PRC) is a graph that shows the relationship between precision (positive predictive value) in the x-axis, and recall (sensitivity), in the y-axis, for every possible cut-off. The cut-offs for a gradient image are the threshold values at which the image contours are evaluated. Every point on the PRC represents a chosen cut-off; therefore, what we can see in the graph is the precision and the recall we get when choosing a cut-off. The average precision (AP) on the full recall range is equivalent to the area under the precision-recall curve.

Segmentation Covering (SC)

The Segmentation Covering is a measure introduced by Arbeláez et al. [2009] that we can see as the generalization of the classic overlap measure between two regions R and R' defined as

$$O(R, R') = \frac{R \cap R'}{R \cup R'} \tag{6.4}$$

The Segmentation Covering (SC) extends the overlap measure so that the covering of a segmentation S by a segmentation S' is defined as

$$SC(S' \to S) = \frac{1}{N} \sum_{R \in S} |R| \cdot \max_{R' \in S'} O(R, R') \tag{6.5}$$

where N denotes the total number of pixels in the image and O denotes the overlap between two regions R and R'.

In the case of a family of multiple image ground-truth segmentations $\{GT_i\}$, the covering of a machine segmentation S is defined by first covering S individually for each human segmentation GT_i, and then averaging over the different humans. If the machine segmentation explains all of the human data, it achieves a perfect covering score.

Probabilistic Rand Index (PRI)

The Rand Index has initially been introduced for clusterings evaluation. It operates by comparing the compatibility of assignments between pairs of points in the compared clusters. The Rand Index between a machine segmentation S and a ground-truth GT is the sum of the number of pixels pairs with the same labels in S and GT, and of those with different labels in the two segmentations, divided by the total number of pixels pairs. The Probabilistic Rand Index (PRI) [Unnikrishnan et al., 2005] is a variant introduced for the case when multiple ground truths are available. If we consider a set of ground-truth segmentations $\{GT_k\}$, the PRI is given by:

$$PRI(S, \{GT_k\}) = \frac{1}{T} \sum_{i<j} [c_{ij} p_{ij} + (1 - c_{ij})(1 - p_{ij})] \tag{6.6}$$

where c_{ij} is the event the pixels i and j have the same label and p_{ij} its probability. T is the total number of pixel pairs.

Variation of Information (VI)

The Variation of Information (VI) is also a measure that has been introduced to compare clusterings [Meilă, 2003]. It measures the distance between two segmentations relatively to their average conditional entropies given by:

$$VI(S, S') = H(S) + H(S') - 2I(S, S') \tag{6.7}$$

where H and I represent the entropies and mutual information between two data clusterings S and S' respectively. In our case, these clusterings are the test and ground-truth segmentations.

	BSDS300			**BSDS500**		
	ODS	OIS	AP	ODS	OIS	AP
Human	0.79	0.79	-	0.80	0.80	-
Ours	**0.65**	**0.67**	**0.62**	**0.66**	**0.68**	**0.62**
Mean Shift [Comaniciu and Meer, 2002]	0.63	0.66	0.54	0.64	**0.68**	0.56
EGB [Felzenszwalb and Huttenlocher, 2004]	0.58	0.62	0.53	0.61	0.64	0.56
NCuts [Cour et al., 2005]	0.62	0.66	0.43	0.64	**0.68**	0.45
Canny [Canny, 1986]	0.58	0.62	0.58	0.60	0.63	0.58
Pb [Malik et al., 2001]	**0.65**	-	-	-	-	-
mPb [Maire et al., 2008]	0.67	-	-	-	-	-
sPb [Maire et al., 2008]	0.68	-	-	-	-	-
gPb [Maire et al., 2008]	0.70	0.72	0.66	0.71	0.74	0.65
gPb-owt-ucm [Arbeláez et al., 2009]	**0.73**	**0.76**	**0.73**	**0.73**	**0.76**	**0.73**

Table 6.2: BSDS image boundary detection scores.

	BSDS500						
	SC (↑)			PRI (↑)		VI (↓)	
	ODS	OIS	Best	ODS	OIS	ODS	OIS
Human	0.72	0.72	-	0.88	0.88	1.17	1.17
Ours	0.56	0.60	0.67	0.81	0.83	1.79	1.57
gPb-owt-ucm [Arbeláez et al., 2009]	**0.59**	**0.65**	**0.74**	**0.83**	**0.86**	**1.69**	**1.48**
Mean Shift [Comaniciu and Meer, 2002]	0.54	0.58	0.66	0.79	0.81	1.85	1.64
EGB [Felzenszwalb and Huttenlocher, 2004]	0.52	0.57	0.69	0.80	0.82	2.21	1.87
Ncuts [Jianbo Shi and Malik, 2000]	0.45	0.53	0.67	0.78	0.80	2.23	1.89

Table 6.3: BSDS image segmentation scores.

6.6.2 Results

To provide a basis for comparing the GCC boundaries, we use the region merging (EGB), Mean Shift, and Multiscale NCuts segmentation methods and the Canny and

(Probability-boundary) Pb edge detectors reviewed in section 6.2. We evaluate each method using the boundary-based precision-recall framework. On the other hand, we use the Variation of Information, Probabilistic Rand Index, and segment covering criteria discussed above to compare the GCC hierarchical segmentations. The BSDS serves as ground truth for both the boundary and region quality measures since the human-drawn boundaries are closed and work as segmentations.

We report in Table 6.2 the boundary detection scores on the BSDS300 and the BSDS500. Besides, Fig. 6.17 displays the precision-recall curves of different methods only for the BSDS500. Table 6.3 presents region benchmarks on the BSDS500.

For a family of machine segmentations $\{S_i\}$, associated with different scales of a hierarchical algorithm or different sets of parameters, we report three scores about the ground-truth covering by segments in $\{S_i\}$. These correspond to selecting covering regions from the segmentation at a universal fixed scale (ODS), a fixed scale per image (OIS), or from any level of the hierarchy or collection $\{S_i\}$ (Best). We also report the Probabilistic Rand Index and Variation of Information benchmarks. While the relative ranking of segmentation algorithms remains fairly consistent across different benchmark criteria, the boundary benchmark (table 6.2 and Fig. 6.17) appears most capable of discriminating performance.

According to the results obtained with our method and the state-of-the-art methods (see Table 6.1), we can underline some reflections. First, given the characteristics of our method (calculation of intensity, color, and texture gradients), the comparison method is the Pb [Malik et al., 2001], which we matched in BSDS300 and surpassed in BSDS500. We can also see that our method's performance is better than other unsupervised methods such as Felz-Hutt [Felzenszwalb and Huttenlocher, 2004], Mean Shift [Comaniciu and Meer, 2002], or Ncuts [Jianbo Shi and Malik, 2000]. This fact is due to the use of the intensity, color, and texture feature combined.

Accordingly, compared to the Pb boosted methods (mPb, sPb, gPb, gPb-owt-ucm [Arbeláez et al., 2009]), it is clear that we do not outperform their scorings; however, our method does not require supervised intermediate stages to optimize the weights of the brightness, color, and texture cues. Furthermore, our method also does not use contour refinement techniques to correct for slightly shifted contours. Therefore, we can assume that our method can increase the contour detection score in the same proportion as the Pb-enhanced methods.

Finally, in relation with the benchmark scores obtained with ANN methods, such methods obviously outperform all these results, achieving or even overpassing the human scores (see for example the blue PR curve in Fig. 6.17 corresponding to the COB method of Kelm et al. [2019]); however, they require huge datasets and annotations to train. We can then consider that our method can serve as an input to ANN architectures to aid model generalization and probably reduce the need for annotated data. We discuss this in more detail in the Perspectives section.

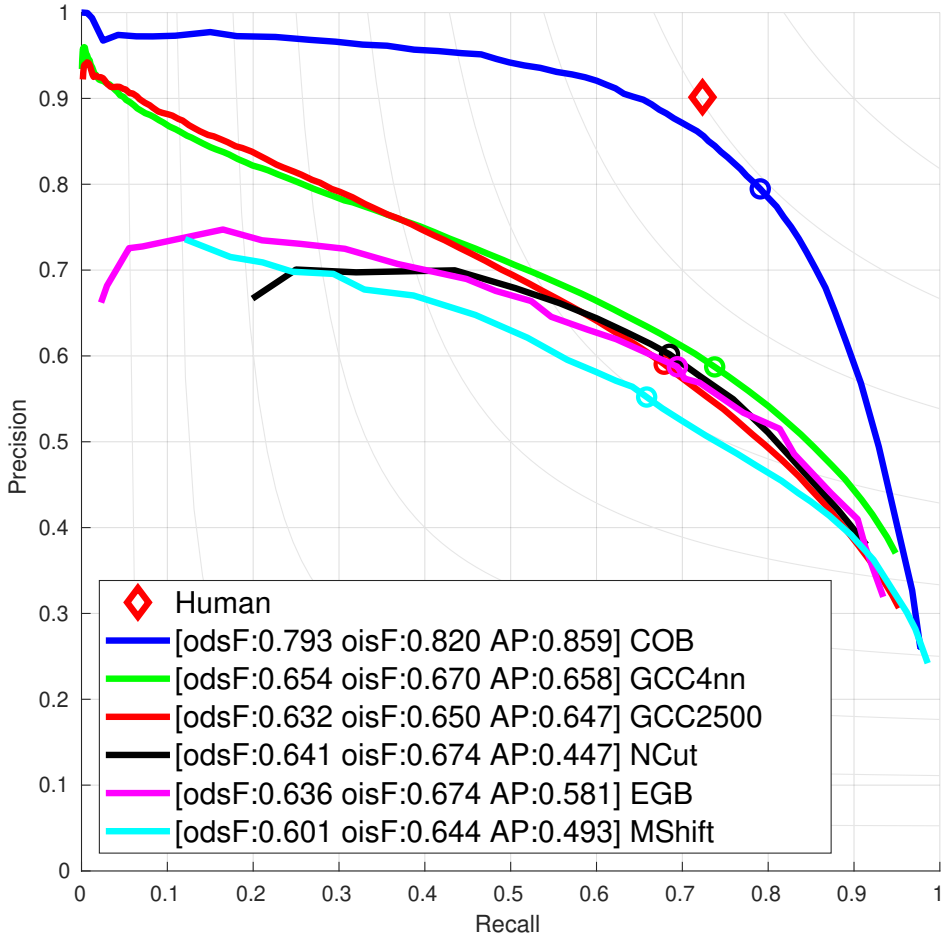


Figure 6.17: Precision-recall plot of different contour detectors.

6.7 Conclusions

This chapter presented the methodology for the segmentation of natural images based on the Gabor Complex Color feature space. We show the diversity of segmentation techniques using such feature space. First, with the segmentation methods based on graphs and second, with the boundary detector and the hierarchical segmentation by watershed.

The scores obtained from the BSDS benchmark show that our algorithms for the detection of contours and the segmentation of images are competitive, taking into account the characteristics of the input features and the methods used for the processing of the images: dissimilarity gradients in the complex color Gabor space and completely unsupervised algorithms for image segmentation.

Conclusion and Perspectives

Summary of Our Main Contributions

This thesis deals with the study of low-level image information concerning human perception for scene understanding. In particular, we study intensity, color, and texture primitives. We validate our methodology on applications that present similar characteristics to the conditions encountered in drone vision-based tasks.

The algorithms proposed in this thesis managed to overcome some of the difficulties present in today's most widely used methods for image segmentation and object detection. Our algorithms do not need to rely on an a priori model, which is reflected in the independence of parameter definition and annotated databases. This methodology benefits the stages of a scene understanding system, which can be integrated into UAVs to develop vision-based tasks.

Throughout this thesis, we encounter different challenges linked to the nature of the application problems we seek to address. Specifically, one of the problems of vision-based application areas is real-time implementation. We did not explore this functionality in more detail; however, the proposed algorithms have elements that we can optimize; for example, the image convolution with filter bank using parallel computation or the calculation of optimal transport metric using a regularization technique.

We further develop the conclusions of this work in the following list.

- We present a framework for landing target detection, one of the main characteristics of visual tasks with drones. This detector uses the intensity information to obtain contours at different scales. The algorithm for detection operates in an unsupervised manner, reducing the number of correct operation parameters in different complex scenarios. In particular, our framework exploits the multiscale image contours in a perceptual approach, taking into account the Helmholtz

principle and the laws of organization of the Gestalt.

- We present a complete study of the color and texture properties present in images, and we evaluated the different options for representation and characterization of color and texture during the analysis of this information.

- We present two image retrieval systems, one based on color information and the other based on texture information. These systems served to test some concepts such as the optimal transport as a metric of similarity between distributions, the spectral analysis properties to represent an image's textures, and the importance of the color spaces.

- Motivated by the lack of a general analysis of the Gabor function optimized for the study of textures in images, we delved into the concepts of signal theory to propose a framework for the generation of a smooth Gabor filter bank: using Parseval's identity, we obtained a transfer function that is closest to one almost everywhere (in 1-d - frequency, and in 2-d - frequency and angle). That allows us to measure the energetic density spectrum accurately and use a true metric to measure the distance between two textures. Previous works on wavelet texture analysis are only approximative, either greyscale or color, and do not use a true distance. The most frequent measure is the Kullback–Leibler, which is a divergence.

- We present the complex multispectral decomposition of a natural image to analyze color, texture, and the relationship between this information in natural images. This decomposition results from a space-frequency study of Gabor filters and the study of color spaces of an image through its luminance and chrominance. Such research results in the Gabor-filter-based Complex Color (GCC) feature space that captures the interaction of perceptual color and texture information of an image.

- We present Gabor-filter-based Complex Color (GCC) feature space's utility, characteristics, and potential by implementing various unsupervised algorithms for natural image segmentation such as clustering algorithms (k-means, Gaussian Mixture, and Birch) and graph-based algorithms (Spectral clustering, MST threshold, and Normalized cuts). In addition to the application of these segmentation methods, we use the feature space to construct a series of high-level texture features, including fundamental frequency, dominant orientation, main texture-forming colors, among others.

  Furthermore, we showed that our methodology allows us to obtain the boundaries of the image in a perceptual way. We show that our methodology outperforms

the BSDS benchmark score of state-of-the-art unsupervised methods for contour detection (Pb, Canny, Mean-shift, Felz-Hutt).

- All the frameworks presented in this document were implemented using open-access libraries in order to make them public. The algorithms presented were coded in python to use the different libraries and frameworks (OpenCV, NumPy, pandas, scikit-learn, scipy, etc.) existing in this language to work with images and public facts. This part of the thesis represents a significant programming effort hidden behind the results shown throughout this document.

- Finally, this thesis belongs to a group of works that maintain traditional computer vision methods as the basis. Although nowadays it is possible to segment images with an accuracy close to that of a human using supervised algorithms and convolutional neural networks, we believe that it is possible to increase the performance, reliability, and explanation of such methods by combining them with systems based on physical phenomena of the vision. Even though AI solutions offer solutions with unprecedented accuracy scores their most criticised drawback today is their lack of explainability. We will comment on that in the Perspectives section below.

Perspectives

We can think of several promising perspectives both in terms of methodology and applications.

- In chapter 1, for the target detection system, there are different ways to improve the system. On the part of the methodology, it is possible to add more features of the image, such as the information of the color and the texture developed in part two of the thesis. This strategy would make the system more robust, providing the possibility of creating markers with specific color and texture patterns. On the implementation side, this system can achieve the analysis and detection of targets in real-time by migrating the python code to some programming language, such as `C++`, which allows the efficient parallelization of functions.

- As for the image search systems presented in chapter 3 it is possible to integrate both features (color and texture) in a single system that allows the search for natural color images. Moreover, in terms of implementation, it is possible to speed up the calculation time of the EMD by implementing a regularization of the measure.

- The Gabor filter we propose achieves a sense of optimality regarding the trade-off between space and frequency. However, this filter bank is non-orthogonal, i.e., the

filter family may introduce redundant information. This feature does not affect our contour detection application as the EMD manages to handle the redundant information introduced mainly by the DC component of the signal. If the objective is the perfect reconstruction of a signal keeping the space-frequency trade-off, a clue to follow is the study of the logarithmic Gabor function (log-Gabor) [Field, 1987], which naturally eliminates the DC component by the logarithmic transformation of the Gabor domain [Boukerroui et al., 2004].

- In chapter 6, we present a brief review of the state-of-the-art methods for superpixel computation. These methods obtain superpixels using intensity and (or) color information. We have a feature space (Gabor-filter-based) that represents the texture and color information of an image in which we can use a metric (EMD). It is natural then to think of an extension of the SLIC algorithm based on this space for the generation of texture superpixels.

- In chapter 6 we obtain the graph gradients for the luminance and chrominance channels of the complex color space. We can use these gradients in conjunction with the ground truth of the BSDS to learn (in a supervised manner) the weight of each color channel and see its perceptual importance in the segmentation task.

- Regarding the AI techniques for segmentation, we obtained results below the scores obtained with DL techniques; nevertheless, we do not use any model. A possibility is to use the proposed Gabor filter bank at the input of a DL network and obtain a model that will use perceptually relevant features. This network could be smaller, better regularized, and less greedy (trainable with less data).

- Using the feature space generated from the smooth filter bank over the complex color space and the optimal transport, we can use most of the morphological algorithms transparently on images containing color and texture; for example, a controlled watershed or MST on natural images. This clue has been explored on a superpixel basis; however, because of the computational time, the implementation on a pixel basis was not achieved. After optimization of the code, it might be possible to perform it on a pixel basis.

- Finally, we consider that it is possible to generate a computational tool for the interactive segmentation of natural images. We can do it using the hierarchical watershed segmentation given by the perceptual boundaries obtained with the Gabor-filter Complex Color (GCC) detector presented in this thesis.

Bibliography

H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes. *International Journal of Computer Vision*, 126(9):961–972, Sept. 2018. ISSN 1573-1405. doi: 10.1007/s11263-018-1070-x.

R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, June 1994. ISSN 0162-8828. doi: 10.1109/34.295913.

N. Aggarwal and R. K. Agrawal. First and Second Order Statistics Features for Classification of Magnetic Resonance Brain Images. *Journal of Signal and Information Processing*, 03(02):146–153, 2012. ISSN 2159-4465, 2159-4481. doi: 10.4236/jsip.2012.32019.

S. Ahmed, H. Balasubramanian, S. Stumpf, C. Morrison, A. Sellen, and M. Grayson. Investigating the intelligibility of a computer vision system for blind users. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, pages 419–429, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 978-1-4503-7118-6. doi: 10.1145/3377325.3377508.

O. S. Al-Kadi. A gabor filter texture analysis approach for histopathological brain tumor subtype discrimination. *arXiv e-prints*, page arXiv:1704.05122, Apr. 2017.

A. Al-Kaff, Q. Meng, D. Martín, A. de la Escalera, and J. M. Armingol. Monocular vision-based obstacle detection/avoidance for unmanned aerial vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 92–97, June 2016. doi: 10.1109/IVS.2016.7535370.

O. Araar, N. Aouf, and I. Vitanov. Vision Based Autonomous Landing of Multirotor UAV on Moving Platform. *Journal of Intelligent & Robotic Systems*, 85(2):369–384, Feb. 2017. ISSN 0921-0296, 1573-0409. doi: 10.1007/s10846-016-0399-z.

P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2294–2301, June 2009. doi: 10.1109/CVPR.2009.5206707.

P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, May 2011. ISSN 1939-3539. doi: 10.1109/TPAMI.2010.161.

S. Arivazhagan and L. Ganesan. Texture classification using wavelet transform. *Pattern Recognition Letters*, 24(9):1513–1521, June 2003. ISSN 0167-8655. doi: 10.1016/S0167-8655(02)00390-2.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *arXiv:1701.07875 [cs, stat]*, Dec. 2017.

A. Artusi, F. Banterle, T. O. Aydın, D. Panozzo, and O. Sorkine-Hornung. *Image Content Retargeting: Maintaining Color, Tone, and Spatial Consistency*. CRC Press, Aug. 2016. ISBN 978-1-315-35533-7.

F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954. ISSN 1939-1471(Electronic),0033-295X(Print). doi: 10.1037/h0054663.

V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, Dec. 2017. ISSN 1939-3539. doi: 10.1109/TPAMI.2016.2644615.

M. Bai and R. Urtasun. Deep Watershed Transform for Instance Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2858–2866, July 2017. doi: 10.1109/CVPR.2017.305.

P. Banerjee, A. K. Bhunia, A. Bhattacharyya, P. P. Roy, and S. Murala. Local Neighborhood Intensity Pattern–A new texture feature descriptor for image retrieval. *Expert Systems with Applications*, 113:100–115, Dec. 2018. ISSN 0957-4174. doi: 10.1016/j.eswa.2018.06.044.

Baquedano, A. Automatic Drone Navigation Based on Computer Vision Applied to Drone Landing. Project Report, ESIEE Paris, June 2017.

B. Benward. *Music in Theory and Practice Volume 1*. McGraw-Hill Higher Education, 2014.

B. Berlin and P. Kay. *Basic Color Terms: Their Universality and Evolution*. University of California Press, 1991. ISBN 978-0-520-07635-8.

S. Beucher and F. Meyer. The Morphological Approach to Segmentation: The Watershed Transformation. In *Mathematical Morphology in Image Processing*. CRC Press, 1st edition edition, 1993. ISBN 978-1-315-21461-0.

M. H. Bharati, J. J. Liu, and J. F. MacGregor. Image texture analysis: Methods and comparisons. *Chemometrics and Intelligent Laboratory Systems*, 72(1):57–71, June 2004. ISSN 0169-7439. doi: 10.1016/j.chemolab.2004.02.005.

A. Bhattacharyya. On a Measure of Divergence between Two Multinomial Populations. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 7(4):401–406, 1946. ISSN 0036-4452.

C. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag, New York, 2006. ISBN 978-0-387-31073-2.

V. I. Bogachev and A. V. Kolesnikov. The Monge-Kantorovich problem: Achievements, connections, and perspectives. *Russian Mathematical Surveys*, 67(5):785, 2012. ISSN 0036-0279. doi: 10.1070/RM2012v067n05ABEH004808.

D. Boukerroui, J. A. Noble, and M. Brady. On the Choice of Band-Pass Quadrature Filters. *Journal of Mathematical Imaging and Vision*, 21(1):53–80, July 2004. ISSN 1573-7683. doi: 10.1023/B:JMIV.0000026557.50965.09.

A. Bovik, M. Clark, and W. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 (1):55–73, Jan. 1990. ISSN 1939-3539. doi: 10.1109/34.41384.

R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw Hill Higher Education, Boston, 3rd revised edition edition, July 1999. ISBN 978-0-07-303938-1.

D. Bradley and G. Roth. G.: Adaptive thresholding using the integral image. *ACM J. Graph. Tools*, pages 13–21, 2007.

F. Bremond. *Scene Understanding: Perception, Multi-Sensor Fusion, Spatio-Temporal Reasoning and Activity Recognition*. Thesis, Université Nice Sophia Antipolis, July 2007.

M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017. ISSN 1558-0792. doi: 10.1109/MSP.2017.2693418.

D. R. Bull. *Communicating Pictures: A Course in Image and Video Coding*. Academic Press, 2014.

Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. ISSN 0162-8828. doi: 10.1109/TPAMI.2018.2815601.

B. Caldairou, N. Passat, P. A. Habas, C. Studholme, and F. Rousseau. A non-local fuzzy segmentation method: Application to brain MRI. *Pattern Recognition*, 44(9): 1916–1927, Sept. 2011. ISSN 0031-3203. doi: 10.1016/j.patcog.2010.06.006.

J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov. 1986. ISSN 0162-8828. doi: 10.1109/TPAMI.1986.4767851.

L. Carlucci. A formal system for texture languages. *Pattern Recognition*, 4(1):53–72, Jan. 1972. ISSN 0031-3203. doi: 10.1016/0031-3203(72)90019-2.

A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. C. Cervera. A Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles. *Journal of Sensors*, 2017:3296874:1–3296874:13, 2017.

T. Chabardès, P. Dokládal, M. Faessel, and M. Bilodeau. A parallel, O(N) algorithm for unbiased, thin watershed. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2569–2573, Sept. 2016. doi: 10.1109/ICIP.2016.7532823.

A. Chaurasia and E. Culurciello. LinkNet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, Dec. 2017. doi: 10.1109/VCIP.2017.8305148.

L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam. MaskLab: Instance Segmentation by Refining Object Detection with Semantic and Direction Features. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, June 2018. doi: 10.1109/CVPR.2018.00422.

C. CIE. Commission internationale de l'eclairage proceedings, 1931. *Cambridge University, Cambridge*, 1932.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and a. A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

D. A. Clausi and M. Ed Jernigan. Designing Gabor filters for optimal texture separability. *Pattern Recognition*, 33(11):1835–1849, Nov. 2000. ISSN 0031-3203. doi: 10.1016/S0031-3203(99)00181-8.

F. S. Cohen, Z. Fan, and S. Attali. Automated inspection of textile fabrics using textural models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):803–808, Aug. 1991. ISSN 1939-3539. doi: 10.1109/34.85670.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5): 603–619, May 2002. ISSN 1939-3539. doi: 10.1109/34.1000236.

A. Cord, F. Bach, and D. Jeulin. Texture classification by statistical learning from morphological image processing: Application to metallic surfaces. *Journal of Microscopy*, 239(2):159–166, 2010. ISSN 1365-2818. doi: 10.1111/j.1365-2818.2010.03365.x.

T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *2005 IEEE Computer Society Conference on Computer Vision and*

Pattern Recognition (CVPR'05), volume 2, pages 1124–1131 vol. 2, June 2005. doi: 10.1109/CVPR.2005.332.

J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1362–1374, Aug. 2009. ISSN 1939-3539. doi: 10.1109/TPAMI.2008.173.

G. R. Cross and A. K. Jain. Markov Random Field Texture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(1):25–39, Jan. 1983. ISSN 1939-3539. doi: 10.1109/TPAMI.1983.4767341.

M. Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 2292–2300, USA, 2013. Curran Associates Inc.

M. Cuturi and D. Avis. Ground Metric Learning. *arXiv e-prints*, page arXiv:1110.2306, Oct. 2011.

J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7):1160–1169, July 1985. doi: 10.1364/JOSAA.2.001160.

J. G. Daugman. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, July 1988. ISSN 0096-3518. doi: 10.1109/29.1644.

G. C. DeAngelis, I. Ohzawa, and R. D. Freeman. Receptive-field dynamics in the central visual pathways. *Trends in Neurosciences*, 18(10):451–458, Oct. 1995. ISSN 0166-2236. doi: 10.1016/0166-2236(95)94496-R.

E. Dejnozkova and P. Dokladal. A parallel algorithm for solving the Eikonal equation. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, volume 3, pages III–325, Apr. 2003a. doi: 10.1109/ICASSP.2003.1199473.

E. Dejnozkova and P. Dokladal. A multiprocessor architecture for PDE-based applications. In *2003 International Conference on Visual Information Engineering VIE 2003*, pages 145–148, July 2003b. doi: 10.1049/cp:20030508.

E. Dejnozkova and P. Dokladal. Asynchronous multi-core architecture for level set methods. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages V–1, May 2004. doi: 10.1109/ICASSP.2004.1327032.

H. Derin and W. S. Cole. Segmentation of textured images using Gibbs random fields. *Computer Vision, Graphics, and Image Processing*, 35(1):72–98, July 1986. ISSN 0734-189X. doi: 10.1016/0734-189X(86)90126-X.

A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis: A Probabilistic Approach*. Interdisciplinary Applied Mathematics. Springer-Verlag, New York, 2008. ISBN 978-0-387-72635-9.

C. Doersch. Tutorial on Variational Autoencoders. *arXiv:1606.05908 [cs, stat]*, Jan. 2021.

P. Dollár and C. L. Zitnick. Structured Forests for Fast Edge Detection. In *2013 IEEE International Conference on Computer Vision*, pages 1841–1848, Dec. 2013. doi: 10.1109/ICCV.2013.231.

P. Dollár and C. L. Zitnick. Fast Edge Detection Using Structured Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570, Aug. 2015. ISSN 1939-3539. doi: 10.1109/TPAMI.2014.2377715.

P. Dollar, Zhuowen Tu, and S. Belongie. Supervised Learning of Edges and Object Boundaries. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1964–1971, June 2006. doi: 10.1109/CVPR.2006.298.

P. Dollar, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. In *Procedings of the British Machine Vision Conference 2009*, pages 91.1–91.11, London, 2009. British Machine Vision Association. ISBN 978-1-901725-39-1. doi: 10.5244/C.23.91.

A. Douglas and P. Kerr. Color and color spaces. (8):1–40, Nov. 2005.

S. Drouyer. *3D Topography by Image Segmentation Approach : Application to Scanning Electron Microscopy*. PhD thesis, Dec. 2017.

S. Drouyer, S. Beucher, M. Bilodeau, M. Moreaud, and L. Sorbier. Sparse Stereo Disparity Map Densification Using Hierarchical Image Segmentation. In J. Angulo, S. Velasco-Forero, and F. Meyer, editors, *Mathematical Morphology and Its Applications to Signal and Image Processing*, Lecture Notes in Computer Science, pages 172–184, Cham, 2017. Springer International Publishing. ISBN 978-3-319-57240-6. doi: 10.1007/978-3-319-57240-6_14.

J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57, Jan. 1973. ISSN 0022-0280. doi: 10.1080/01969727308546046.

R. W. Ehrich and J. P. Foith. A view of texture topology and texture description. *Computer Graphics and Image Processing*, 8(2):174–202, Oct. 1978. ISSN 0146-664X. doi: 10.1016/0146-664X(78)90048-5.

M. D. Fairchild. *Color Appearance Models*. John Wiley & Sons, July 2005. ISBN 978-0-470-01269-7.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, Sept. 2004. ISSN 0920-5691, 1573-1405. doi: 10.1023/B:VISI.0000022288.19776.77.

M. Fiala. Designing Highly Reliable Fiducial Markers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1317–1324, July 2010. ISSN 1939-3539. doi: 10.1109/TPAMI.2009.146.

D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *JOSA A*, 4(12):2379–2394, Dec. 1987. ISSN 1520-8532. doi: 10.1364/JOSAA.4.002379.

A. Fischer and C. Igel. An Introduction to Restricted Boltzmann Machines. In L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Lecture Notes in Computer Science, pages 14–36, Berlin, Heidelberg, 2012. Springer. ISBN 978-3-642-33275-3. doi: 10.1007/978-3-642-33275-3_2.

N. Forcadel, C. Le Guyader, and C. Gout. Generalized fast marching method: Applications to image segmentation. *Numerical Algorithms*, 48(1):189–211, July 2008. ISSN 1572-9265. doi: 10.1007/s11075-008-9183-x.

C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, May 2014. doi: 10.1109/ICRA.2014.6906584.

J. R. Fram and E. S. Deutsch. On the Quantitative Evaluation of Edge Detection Schemes and their Comparison with Human Performance. *IEEE Transactions on Computers*, C-24(6):616–628, June 1975. ISSN 1557-9956. doi: 10.1109/T-C.1975.224274.

N. Franceschini, F. Ruffier, J. Serres, and S. Viollet. Optic flow based visual guidance: From flying insects to miniature aerial vehicles. In T. M. Lam, editor, *Aerial Vehicles*. IntechOpen, Rijeka, 2009.

B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science (New York, N.Y.)*, 315(5814):972–976, Feb. 2017. ISSN 1095-9203. doi: 10.1126/science.1136800.

K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4): 193–202, Apr. 1980. ISSN 1432-0770. doi: 10.1007/BF00344251.

H. Furukawa. Deep Learning for End-to-End Automatic Target Recognition from Synthetic Aperture Radar Imagery. *arXiv:1801.08558 [cs]*, Jan. 2018.

D. Gabor. Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, 93(26):429–441, Nov. 1946. doi: 10.1049/ji-3-2.1946.0074.

N. Gageik, P. Benz, and S. Montenegro. Obstacle Detection and Collision Avoidance for a UAV With Complementary Low-Cost Sensors. *IEEE Access*, 3:599–609, 2015. doi: 10.1109/ACCESS.2015.2432455.

S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, June 2014. ISSN 0031-3203. doi: 10.1016/j.patcog.2014.01.005.

A. Gaszczak, T. P. Breckon, and J. Han. Real-time people and vehicle detection from UAV imagery. In *Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques*, volume 7878, page 78780B. International Society for Optics and Photonics, Jan. 2011. doi: 10.1117/12.876663.

A. L. Gibbs and F. E. Su. On Choosing and Bounding Probability Metrics. *International Statistical Review*, 70:419–435, Dec. 2002. doi: 10.1111/j.1751-5823.2002.tb00178.x.

E. B. Goldstein. *Sensation and Perception*. Cengage Learning, Feb. 2009. ISBN 978-0-495-60149-4.

R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2008. ISBN 978-0-13-168728-8.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 27, 2014.

L. Grady. Random Walks for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, Nov. 2006. ISSN 1939-3539. doi: 10.1109/TPAMI.2006.233.

L. Grady and E. Schwartz. Isoperimetric graph partitioning for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):469–475, Mar. 2006. ISSN 1939-3539. doi: 10.1109/TPAMI.2006.57.

G. H. Granlund. In search of a general picture processing operator. *Computer Graphics and Image Processing*, 8(2):155–173, Oct. 1978. ISSN 0146-664X. doi: 10.1016/0146-664X(78)90047-3.

C. Grigorescu, N. Petkov, and M. A. Westenberg. Contour detection based on non-classical receptive field inhibition. *IEEE Transactions on Image Processing*, 12(7):729–739, July 2003. ISSN 1941-0042. doi: 10.1109/TIP.2003.814250.

S. Grigorescu, N. Petkov, and P. Kruizinga. Comparison of texture features based on Gabor filters. *IEEE Transactions on Image Processing*, 11(10):1160–1167, Oct. 2002. ISSN 1941-0042. doi: 10.1109/TIP.2002.804262.

S. Hallman and C. C. Fowlkes. Oriented edge forests for boundary detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1732–1740, June 2015. doi: 10.1109/CVPR.2015.7298782.

R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, Apr. 1950. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1950.tb00463.x.

R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, Nov. 1973. ISSN 2168-2909. doi: 10.1109/TSMC.1973.4309314.

M. Hassner and J. Sklansky. The use of Markov Random Fields as models of texture. *Computer Graphics and Image Processing*, 12(4):357–370, Apr. 1980. ISSN 0146-664X. doi: 10.1016/0146-664X(80)90019-2.

D. He, Y. Qiao, S. Chan, and N. Guizani. Flight Security and Safety of Drones in Airborne Fog Computing Systems. *IEEE Communications Magazine*, 56(5):66–71, May 2018. ISSN 1558-1896. doi: 10.1109/MCOM.2018.1700916.

K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. doi: 10.1109/CVPR.2016.90.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, Feb. 2020. ISSN 1939-3539. doi: 10.1109/TPAMI.2018.2844175.

E. Hering. *Zur lehre vom lichtsinne*. C. Gerold's Sohn, Hamburg, 1878.

S. L. Horowitz and T. Pavlidis. Picture Segmentation by a Tree Traversal Algorithm. *Journal of the ACM*, 23(2):368–388, Apr. 1976. ISSN 0004-5411. doi: 10.1145/321941.321956.

S. Hrabar and G. S. Sukhatme. A comparison of two camera configurations for optic-flow based navigation of a UAV through urban canyons. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2673–2680 vol.3, Sept. 2004. doi: 10.1109/IROS.2004.1389812.

A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera. In *Robotics Research*, Springer Tracts in Advanced Robotics, pages 235–252. Springer, Cham, 2017. ISBN 978-3-319-29362-2 978-3-319-29363-9.

M. H. Hueckel. An Operator Which Locates Edges in Digitized Pictures. *Journal of the ACM*, 18(1):113–125, Jan. 1971. ISSN 0004-5411. doi: 10.1145/321623.321635.

A. Humeau-Heurtier. Texture Feature Extraction Methods: A Survey. *IEEE Access*, 7:8975–9000, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2890743.

N. Ikonomatakis, K. Plataniotis, M. Zervakis, and A. Venetsanopoulos. Region growing and region merging image segmentation. In *Proceedings of 13th International Conference on Digital Signal Processing*, volume 1, pages 299–302 vol.1, July 1997. doi: 10.1109/ICDSP.1997.628077.

Internest. Autonomous flights in complex environments. http://internest.fr/fr/.

A. Jain and G. Healey. A multiscale representation including opponent color features for texture recognition. *IEEE Transactions on Image Processing*, 7(1):124–128, Jan. 1998. ISSN 1941-0042. doi: 10.1109/83.650858.

A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, Jan. 1991. ISSN 0031-3203. doi: 10.1016/0031-3203(91)90143-S.

J. Janai, F. Güney, A. Behl, and A. Geiger. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*, 12(1–3):1–308, 2020. ISSN 1572-2740. doi: 10.1561/0600000079.

Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug. 2000. ISSN 1939-3539. doi: 10.1109/34.868688.

J. Kamarainen, V. Kyrki, and H. Kalviainen. Noise tolerant object recognition using Gabor filtering. In *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, volume 2, pages 1349–1352 vol.2, July 2002a. doi: 10.1109/ICDSP.2002.1028344.

J. Kamarainen, V. Kyrki, and H. Kalviainen. Fundamental frequency Gabor filters for object recognition. In *Object Recognition Supported by User Interaction for Service Robots*, volume 1, pages 628–631 vol.1, Aug. 2002b. doi: 10.1109/ICPR.2002.1044822.

L. V. Kantorovich. On a Problem of Monge. *Journal of Mathematical Sciences*, 133 (4):1383–1383, Mar. 2006. ISSN 1573-8795. doi: 10.1007/s10958-006-0050-9.

M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, Jan. 1988. ISSN 1573-1405. doi: 10.1007/BF00133570.

P. Kay and T. Regier. Resolving the question of color naming universals. *Proceedings of the National Academy of Sciences*, 100(15):9085–9089, July 2003. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1532837100.

Q. Ke, J. Liu, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. Chapter 5 - Computer Vision for Human–Machine Interaction. In M. Leo and G. M. Farinella, editors, *Computer Vision for Assistive Healthcare*, Computer Vision and Pattern Recognition, pages 127–145. Academic Press, Jan. 2018. ISBN 978-0-12-813445-0.

J. M. Keller, S. Chen, and R. M. Crownover. Texture description and segmentation through fractal geometry. *Computer Vision, Graphics, and Image Processing*, 45(2): 150–166, Feb. 1989. ISSN 0734-189X. doi: 10.1016/0734-189X(89)90130-8.

A. P. Kelm, V. S. Rao, and U. Zölzer. Object Contour and Edge Detection with RefineContourNet. In M. Vento and G. Percannella, editors, *Computer Analysis of Images and Patterns*, Lecture Notes in Computer Science, pages 246–258, Cham, 2019. Springer International Publishing. ISBN 978-3-030-29888-3. doi: 10.1007/978-3-030-29888-3_20.

D. A. Kerr. Chromaticity and Chrominance in color definition. (3):1–4, 2003.

M. A. Khamsi. Generalized metric spaces: A survey. *Journal of Fixed Point Theory and Applications*, 17(3):455–475, Sept. 2015. ISSN 1661-7746. doi: 10.1007/s11784-015-0232-5.

J. Kivinen, C. Williams, and N. Heess. Visual Boundary Prediction: A Deep Neural Prediction Network and Quality Dissection. In *Artificial Intelligence and Statistics*, pages 512–521. PMLR, Apr. 2014.

D. A. Klein and S. Frintrop. Center-surround divergence of feature statistics for salient object detection. In *2011 International Conference on Computer Vision*, pages 2214–2219, Nov. 2011. doi: 10.1109/ICCV.2011.6126499.

J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50(5):363–370, Aug. 1984. ISSN 1432-0770. doi: 10.1007/BF00336961.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

P. Kruizinga and N. Petkov. Nonlinear operator for oriented texture. *IEEE Transactions on Image Processing*, 8(10):1395–1407, Oct. 1999. ISSN 1941-0042. doi: 10.1109/83.791965.

J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2):115–129, June 1964. ISSN 1860-0980. doi: 10.1007/BF02289694.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, Mar. 1951. ISSN 0003-4851, 2168-8990. doi: 10.1214/aoms/1177729694.

R. Kwitt and A. Uhl. Image similarity measurement by Kullback-Leibler divergences between complex wavelet subband statistics for texture retrieval. In *2008 15th IEEE International Conference on Image Processing*, pages 933–936, Oct. 2008. doi: 10.1109/ICIP.2008.4711909.

G. Kylberg. The Kylberg Texture Dataset v. 1.0. External Report (Blue Series) 35, Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden, Sept. 2011.

C. Kyrkou, S. Timotheou, P. Kolios, T. Theocharides, and C. Panayiotou. Drones: Augmenting Our Quality of Life. *IEEE Potentials*, 38(1):30–36, Jan. 2019. ISSN 1558-1772. doi: 10.1109/MPOT.2018.2850386.

S. Lacroix and F. Caballero. Autonomous detection of safe landing areas for an UAV from monocular images. In *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

S. Lange, N. Sünderhauf, and P. Protzel. Autonomous Landing for a Multirotor UAV Using Vision. In *In SIMPAR 2008 Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots*, pages 482–491, 2008.

K. I. Laws. Texture energy measures. In *Proc. Image Understanding Workshop*, pages 47–51, 1979.

K. I. Laws. Rapid Texture Identification. In *Image Processing for Missile Guidance*, volume 0238, pages 376–381. International Society for Optics and Photonics, Dec. 1980a. doi: 10.1117/12.959169.

K. I. Laws. Textured image segmentation. Technical report, University of Southern California Los Angeles Image Processing INST, 1980b.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov. 1998. ISSN 1558-2256. doi: 10.1109/5.726791.

J. Lee, J. Wang, D. Crandall, S. Šabanović, and G. Fox. Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles. In *2017 First IEEE International Conference on Robotic Computing (IRC)*, pages 36–43, Apr. 2017. doi: 10.1109/IRC.2017.77.

T. S. Lee. Image Representation Using 2D Gabor Wavelets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):959–971, Oct. 1996. ISSN 0162-8828. doi: 10.1109/34.541406.

T. Leung and J. Malik. Contour continuity in region based image segmentation. In H. Burkhardt and B. Neumann, editors, *Computer Vision — ECCV'98*, Lecture Notes in Computer Science, pages 544–559, Berlin, Heidelberg, 1998. Springer. ISBN 978-3-540-69354-3. doi: 10.1007/BFb0055689.

C. H. Li and C. K. Lee. Minimum cross entropy thresholding. *Pattern Recognition*, 26 (4):617–625, Apr. 1993. ISSN 0031-3203. doi: 10.1016/0031-3203(93)90115-D.

J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman. Multi-target detection and tracking from a single camera in Unmanned Aerial Vehicles (UAVs). In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4992–4997, Oct. 2016. doi: 10.1109/IROS.2016.7759733.

J. J. Lim, C. L. Zitnick, and P. Dollár. Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3158–3165, June 2013. doi: 10.1109/CVPR.2013.406.

T. Lin and S. Maji. Visualizing and Understanding Deep Texture Representations. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2791–2799, June 2016. doi: 10.1109/CVPR.2016.305.

T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, July 2017. doi: 10.1109/CVPR.2017.106.

C. Liu and H. Wechsler. Independent component analysis of Gabor features for face recognition. *IEEE Transactions on Neural Networks*, 14(4):919–928, July 2003. ISSN 1941-0093. doi: 10.1109/TNN.2003.813829.

C.-L. Liu, M. Koga, and H. Fujisawa. Gabor feature extraction for character recognition: Comparison with gradient feature. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 121–125 Vol. 1, Aug. 2005. doi: 10.1109/ICDAR.2005.119.

P. Liu, J.-M. Guo, K. Chamnongthai, and H. Prasetyo. Fusion of color histogram and LBP-based features for texture image retrieval and classification. *Information Sciences*, 390:95–111, June 2017. ISSN 0020-0255. doi: 10.1016/j.ins.2017.01.025.

J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

C.-T. Lu, D. Chen, and Y. Kou. Multivariate spatial outlier detection. *International Journal on Artificial Intelligence Tools*, 13(04):801–811, Dec. 2004. ISSN 0218-2130. doi: 10.1142/S021821300400182X.

S. Y. Lu and K. S. Fu. A syntactic approach to texture analysis. *Computer Graphics and Image Processing*, 7(3):303–330, June 1978. ISSN 0146-664X. doi: 10.1016/S0146-664X(78)80001-X.

M. Lukashevich and R. Sadykhov. Texture analysis: Algorithm for texture features computation. In *2012 IV International Conference "Problems of Cybernetics and Informatics" (PCI)*, pages 1–3, Sept. 2012. doi: 10.1109/ICPCI.2012.6486307.

V. Machairas, M. Faessel, D. Cárdenas-Peña, T. Chabardes, T. Walter, and E. Decencière. Waterpixels. *IEEE Transactions on Image Processing*, 24(11):3707–3716, Nov. 2015. ISSN 1941-0042. doi: 10.1109/TIP.2015.2451011.

T. Mäenpää and M. Pietikäinen. Classification with color and texture: Jointly or separately? *Pattern Recognition*, 37(8):1629–1640, Aug. 2004. ISSN 0031-3203. doi: 10.1016/j.patcog.2003.11.011.

J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative Sparse Image Models for Class-Specific Edge Detection and Image Interpretation. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, Lecture Notes in Computer Science, pages 43–56, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-88690-7. doi: 10.1007/978-3-540-88690-7_4.

M. Maire, P. Arbeláez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587420.

M. Maire, S. X. Yu, and P. Perona. Reconstructive Sparse Code Transfer for Contour Detection and Semantic Labeling. *arXiv:1410.4521 [cs]*, Oct. 2014.

H. Maître. La détection des contours dans les images. In *Le traitement des images*, Traitement du signal, page 366 p. Hermes Science Publications, Aug. 2003. ISBN 2-7462-0584-X.

A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial Autoencoders. In *International Conference on Learning Representations*, 2016.

E. N. Malamas, E. G. M. Petrakis, M. Zervakis, L. Petit, and J. Legat. A survey on industrial vision systems, applications, tools. *Image Vis. Comput.*, 21:171–188, 2003.

J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, Contours and Regions: Cue Integration in Image Segmentation. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, page 918, USA, Sept. 1999. IEEE Computer Society. ISBN 978-0-7695-0164-2.

J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and Texture Analysis for Image Segmentation. *International Journal of Computer Vision*, 43(1):7–27, June 2001. ISSN 1573-1405. doi: 10.1023/A:1011174803800.

K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. V. Gool. Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):819–833, Apr. 2018. ISSN 1939-3539. doi: 10.1109/TPAMI.2017.2700300.

D. Marr and E. Hildreth. Theory of edge detection. *Proc. R. Soc. Lond. B*, 207(1167): 187–217, Feb. 1980. ISSN 0080-4649, 2053-9193. doi: 10.1098/rspb.1980.0020.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423 vol.2, July 2001. doi: 10.1109/ICCV.2001.937655.

D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using brightness and texture. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, pages 1279–1286, Cambridge, MA, USA, Jan. 2002. MIT Press.

D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, May 2004. ISSN 1939-3539. doi: 10.1109/TPAMI.2004.1273918.

A. Materka and M. Strzelecki. Texture analysis methods – a review. Technical report, INSTITUTE OF ELECTRONICS, TECHNICAL UNIVERSITY OF LODZ, 1998.

M. Meilă. Comparing Clusterings by the Variation of Information. In B. Schölkopf and M. K. Warmuth, editors, *Learning Theory and Kernel Machines*, Lecture Notes in Computer Science, pages 173–187, Berlin, Heidelberg, 2003. Springer. ISBN 978-3-540-45167-9. doi: 10.1007/978-3-540-45167-9_14.

F. Meyer. Color image segmentation. In *1992 International Conference on Image Processing and Its Applications*, pages 303–306, Apr. 1992.

F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, Sept. 1990. ISSN 1047-3203. doi: 10.1016/1047-3203(90)90014-M.

S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image Segmentation Using Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3059968.

M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs, stat]*, Nov. 2014.

M. A. Moreno-Armendáriz and H. Calvo. Visual SLAM and Obstacle Avoidance in Real Time for Mobile Robots Navigation. In *2014 International Conference on Mechatronics, Electronics and Automotive Engineering*, pages 44–49, Nov. 2014. doi: 10.1109/ICMEAE.2014.12.

M. C. Morrone and R. A. Owens. Feature detection from local energy. *Pattern Recognition Letters*, 6(5):303–313, Dec. 1987. ISSN 0167-8655. doi: 10.1016/0167-8655(87)90013-4.

M. C. Morrone, D. C. Burr, and H. B. Barlow. Feature detection in human vision: A phase-dependent energy model. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 235(1280):221–245, Dec. 1988. doi: 10.1098/rspb.1988.0073.

D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, July 1989. ISSN 1097-0312. doi: 10.1002/cpa.3160420503.

L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Proceedings. International Symposium on Mixed and Augmented Reality*, pages 27–36, Oct. 2002. doi: 10.1109/ISMAR.2002.1115065.

L. Najman and M. Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18 (12):1163–1173, Dec. 1996. ISSN 1939-3539. doi: 10.1109/34.546254.

S. M. S. Nejhum, J. Ho, and M.-H. Yang. Visual tracking with histograms and articulating blocks. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587575.

P. Neubert and P. Protzel. Compact Watershed and Preemptive SLIC: On Improving Trade-offs of Superpixel Segmentation Algorithms. In *2014 22nd International Conference on Pattern Recognition*, pages 996–1001, Aug. 2014. doi: 10.1109/ICPR.2014.181.

I. Newton. *Opticks: A Treatise of the Reflections, Refractions, Inflections, and Colours of Light*. 1704.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 849–856, Cambridge, MA, USA, Jan. 2001. MIT Press.

W. Niblack. *An Introduction to Digital Image Processing*. Prentice-Hall, 1986. ISBN 978-0-13-480674-7.

H. Noh, S. Hong, and B. Han. Learning Deconvolution Network for Semantic Segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1520–1528, Dec. 2015. doi: 10.1109/ICCV.2015.178.

T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, Jan. 1996. ISSN 0031-3203. doi: 10.1016/0031-3203(95)00067-4.

M. G. Omran, A. P. Engelbrecht, and A. Salman. An overview of clustering methods. *Intelligent Data Analysis*, 11(6):583–605, 2007.

S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, Nov. 1988. ISSN 0021-9991. doi: 10.1016/0021-9991(88)90002-2.

N. A. Othman and I. Aydin. A new IoT combined body detection of people by using computer vision for security application. In *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 108–112, Sept. 2017. doi: 10.1109/CICN.2017.8319366.

N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan. 1979. ISSN 0018-9472. doi: 10.1109/TSMC.1979.4310076.

R. P. Padhy, F. Xia, S. K. Choudhury, P. K. Sa, and S. Bakshi. Monocular Vision Aided Autonomous UAV Navigation in Indoor Corridor Environments. *IEEE Transactions on Sustainable Computing*, pages 1–1, 2018. doi: 10.1109/TSUSC.2018.2810952.

C. Palm and T. M. Lehmann. Classification of color textures by Gabor filtering. *Machine Graphics & Vision International Journal*, 11(2/3):195–219, Sept. 2002. ISSN 1230-0535.

C. Palm, D. Keysers, T. Lehmann, and K. Spitzer. Gabor filtering of complex hue/saturation images for color texture classification. In *Proc. JCIS*, pages 45–49. Citeseer, 2000.

O. Pele and M. Werman. A Linear Time Histogram Metric for Improved SIFT Matching. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, Lecture Notes in Computer Science, pages 495–508, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-88690-7. doi: 10.1007/978-3-540-88690-7_37.

H. Permuter, J. Francos, and I. Jermyn. A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, Apr. 2006. ISSN 0031-3203. doi: 10.1016/j.patcog.2005.10.028.

B. Perret, J. Cousty, S. J. F. Guimarães, and D. S. Maia. Evaluation of Hierarchical Watersheds. *IEEE Transactions on Image Processing*, 27(4):1676–1688, Apr. 2018. ISSN 1941-0042. doi: 10.1109/TIP.2017.2779604.

J. Petitot. *Neurogéométrie de la vision: modèles mathématiques et physiques des architectures fonctionnelles*. Editions Ecole Polytechnique, 2008. ISBN 978-2-7302-1507-7.

N. Petkov. Biologically motivated computationally intensive approaches to image pattern recognition. *Future Generation Computer Systems*, 11(4):451–465, Aug. 1995. ISSN 0167-739X. doi: 10.1016/0167-739X(95)00015-K.

M. G. Petrolekas and S. Mitra. Fractal model for digital image texture analysis. In *Applications of Digital Image Processing XV*, volume 1771, pages 292–298. International Society for Optics and Photonics, Jan. 1993. doi: 10.1117/12.139073.

M. M. P. Petrou and P. G. Sevilla. *Image Processing: Dealing with Texture*. Wiley, Mar. 2006. ISBN 978-0-470-02628-1.

G. Peyré and M. Cuturi. Computational Optimal Transport. *arXiv:1803.00567 [stat]*, Mar. 2018.

R. Preetha and G. R. Suresh. Performance Analysis of Fuzzy C Means Algorithm in Automated Detection of Brain Tumor. In *2014 World Congress on Computing and Communication Technologies*, pages 30–33, Feb. 2014. doi: 10.1109/WCCCT.2014.26.

J. M. Prewitt. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19, 1970.

Y. Prokhorov. Convergence of Random Processes and Limit Theorems in Probability Theory. *Theory of Probability & Its Applications*, 1(2):157–214, 1956. doi: 10.1137/1101016.

F. Pukelsheim. The Three Sigma Rule. *The American Statistician*, 48(2):88–91, 1994. doi: 10.1080/00031305.1994.10476030.

J. Puzicha, J. M. Buhmann, Y. Rubner, and C. Tomasi. Empirical evaluation of dissimilarity measures for color and texture. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1165–1172 vol.2, Sept. 1999. doi: 10.1109/ICCV.1999.790412.

I.-U.-H. Qazi, O. Alata, J.-C. Burie, A. Moussa, and C. Fernandez-Maloigne. Choice of a pertinent color space for color texture characterization using parametric spectral analysis. *Pattern Recognition*, 44(1):16–31, Jan. 2011. ISSN 0031-3203. doi: 10.1016/j.patcog.2010.07.007.

A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]*, Jan. 2016.

T. Randen and J. Husoy. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, Apr. 1999. ISSN 1939-3539. doi: 10.1109/34.761261.

I. S. Reed and X. Yu. Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(10):1760–1770, Oct. 1990. ISSN 0096-3518. doi: 10.1109/29.60107.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017. ISSN 1939-3539. doi: 10.1109/TPAMI.2016.2577031.

X. Ren. Multi-scale Improves Boundary Detection in Natural Images. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, Lecture Notes in Computer Science, pages 533–545, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-88690-7. doi: 10.1007/978-3-540-88690-7_40.

X. Ren and L. Bo. Discriminatively trained sparse code gradients for contour detection. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 584–592, Red Hook, NY, USA, Dec. 2012. Curran Associates Inc.

X. Ren and J. Malik. Learning a classification model for segmentation. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 10–17 vol.1, Oct. 2003. doi: 10.1109/ICCV.2003.1238308.

T. Ridler and S. Calvard. Picture Thresholding Using an Iterative Selection Method. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(8):630–632, 1978.

L. G. Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

J. B. T. M. Roerdink and A. Meijster. The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamenta Informaticae*, 41(1,2):187–228, Jan. 2000. ISSN 0169-2968. doi: 10.3233/FI-2000-411207.

O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28.

Y. Rubner and C. Tomasi. *Perceptual Metrics for Image Database Navigation*. The Springer International Series in Engineering and Computer Science. Springer US, 2001. ISBN 978-0-7923-7219-6.

Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance As a Metric for Image Retrieval. *Int. J. Comput. Vision*, 40(2):99–121, Nov. 2000. ISSN 0920-5691. doi: 10.1023/A:1026543900054.

S. Saini and K. Arora. A study analysis on the different image segmentation techniques. *International Journal of Information & Computation Technology*, 4(14):1445–1452, 2014.

R. Salakhutdinov and G. Hinton. Deep Boltzmann Machines. In *Artificial Intelligence and Statistics*, pages 448–455. PMLR, Apr. 2009.

S. Sangwine and T. Ell. Colour image filters based on hypercomplex convolution. *IEE Proceedings - Vision, Image and Signal Processing*, 147(2):89–93, Apr. 2000. ISSN 1350-245X. doi: 10.1049/ip-vis:20000211.

D. Sankowski and J. Nowakowski. *Computer Vision in Robotics and Industrial Applications*, volume 3. World Scientific, 2014.

B. Sathya and R. Manavalan. Image segmentation by clustering methods: Performance analysis. *International Journal of Computer Applications*, 29(11):27–32, 2011.

J. Sauvola and M. Pietikäinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, Feb. 2000. ISSN 0031-3203. doi: 10.1016/S0031-3203(99)00055-2.

D. Scaramuzza and F. Fraundorfer. Visual Odometry [Tutorial]. *IEEE Robotics Automation Magazine*, 18(4):80–92, Dec. 2011. ISSN 1070-9932. doi: 10.1109/MRA.2011.943233.

D. Scott. Histograms: Theory and Practice. pages 47–94. May 2008. ISBN 978-0-471-54770-9.

S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528, June 2006. doi: 10.1109/CVPR.2006.19.

J. Serra. The Boolean model and random sets. *Computer Graphics and Image Processing*, 12(2):99–126, Feb. 1980. ISSN 0146-664X. doi: 10.1016/0146-664X(80)90006-4.

M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *J. Electronic Imaging*, 13(1):146–168, July 2010.

M. Sigmund. Statistical Analysis of Fundamental Frequency Based Features in Speech under Stress. *Information Technology and Control*, 42(3):286–291, Sept. 2013. ISSN 2335-884X. doi: 10.5755/j01.itc.42.3.3895.

K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, Apr. 2015.

A. Singandhupe and H. M. La. A Review of SLAM Techniques and Security in Autonomous Driving. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 602–607, Feb. 2019. doi: 10.1109/IRC.2019.00122.

A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, Dec. 2000. ISSN 0162-8828. doi: 10.1109/34.895972.

R. W. K. So and A. C. S. Chung. A novel learning-based dissimilarity metric for rigid and non-rigid medical image registration by using Bhattacharyya Distances. *Pattern Recognition*, 62:161–174, Feb. 2017. ISSN 0031-3203. doi: 10.1016/j.patcog.2016.09.004.

I. Sobel and G. Feldman. An Isotropic 3×3 image gradient operator. 1990. doi: 10.13140/RG.2.1.1912.4965.

M. V. Srinivasan and R. L. Gregory. How Bees Exploit Optic Flow: Behavioural Experiments and Neural Models [and Discussion]. *Philosophical Transactions: Biological Sciences*, 337(1281):253–259, 1992. ISSN 0962-8436.

D. Steinley. K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, 59(1):1–34, 2006. ISSN 2044-8317. doi: 10.1348/000711005X48266.

D. Stutz, A. Hermans, and B. Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018. ISSN 1077-3142. doi: 10.1016/j.cviu.2017.03.007.

Ö. N. Subakan and B. C. Vemuri. Color Image Segmentation in a Quaternion Framework. *Energy minimization methods in computer vision and pattern recognition. International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 5681(2009):401–414, Jan. 2009. doi: 10.1007/978-3-642-03641-5_30.

F. Sultana, A. Sufian, and P. Dutta. Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey. *Knowledge-Based Systems*, 201-202:106062, Aug. 2020. ISSN 0950-7051. doi: 10.1016/j.knosys.2020.106062.

M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, Nov. 1991. ISSN 1573-1405. doi: 10.1007/BF00130487.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. *arXiv:1409.4842 [cs]*, Sept. 2014.

T. Taketomi, H. Uchiyama, and S. Ikeda. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16, June 2017. ISSN 1882-6695. doi: 10.1186/s41074-017-0027-2.

S. Thrun and A. Bü. Integrating Grid-based and Topological Maps for Mobile Robot Navigation. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, pages 944–950, Portland, Oregon, 1996. AAAI Press. ISBN 978-0-262-51091-2.

T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka. Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue. *IEEE Robotics Automation Magazine*, 19(3):46–56, Sept. 2012. ISSN 1070-9932. doi: 10.1109/MRA.2012.2206473.

J. Treboux, D. Genoud, and R. Ingold. Decision Tree Ensemble Vs. N.N. Deep Learning: Efficiency Comparison For A Small Image Dataset. In *2018 International Workshop on Big Data and Information Security (IWBIS)*, pages 25–30, May 2018. doi: 10.1109/IWBIS.2018.8471704.

M. Tuceryan and A. K. Jain. Texture analysis. In *Handbook of Pattern Recognition and Computer Vision*, pages 235–276. WORLD SCIENTIFIC, Aug. 1993. ISBN 978-981-02-1136-3.

R. Unnikrishnan, C. Pantofaru, and M. Hebert. A Measure for Objective Evaluation of Image Segmentation Algorithms. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pages 34–34, Sept. 2005. doi: 10.1109/CVPR.2005.390.

A. A. Ursani, K. Kpalma, and J. Ronsin. Texture features based on Fourier transform and Gabor filters: An empirical comparison. In *2007 International Conference on Machine Vision*, pages 67–72, Dec. 2007. doi: 10.1109/ICMV.2007.4469275.

A. Vedaldi and S. Soatto. Quick Shift and Kernel Methods for Mode Seeking. In *Computer Vision – ECCV 2008*, Lecture Notes in Computer Science, pages 705–718. Springer, Berlin, Heidelberg, Oct. 2008. ISBN 978-3-540-88692-1 978-3-540-88693-8. doi: 10.1007/978-3-540-88693-8_52.

P. Viola and M. J. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004. ISSN 1573-1405. doi: 10.1023/B:VISI.0000013087.49260.fb.

J. W. von Goethe. *Goethe's Theory of Colours*. Nov. 2015.

H. Von Helmholtz. *Handbuch der physiologischen Optik*, volume 9. Voss, Hamburg, 1867.

M. Wang, X. Liu, Y. Gao, X. Ma, and N. Q. Soomro. Superpixel segmentation: A benchmark. *Signal Processing: Image Communication*, 56:28–39, Aug. 2017. ISSN 0923-5965. doi: 10.1016/j.image.2017.04.007.

J. S. Werner and L. M. Chalupa. *The Visual Neurosciences*. MIT Press, 2004. ISBN 978-0-262-03308-4.

M. Wertheimer. FormsUntersuchungen zur Lehre von der Gestalt II. *Psycologische Forschung*, 4:301–350, 1923.

V. Wiley and T. Lucas. Computer vision and image processing: A paper review. *International journal of artificial intelligence*, 2:29–36, 2018.

G. Winkler. Random fields and texture models. In *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*, pages 231–242. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-642-55760-6.

A. Witkin. Scale-space filtering: A new approach to multi-scale description. In *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 150–153, Mar. 1984. doi: 10.1109/ICASSP.1984.1172729.

W. D. Wright. Professor wright's paper from the golden jubilee book: The historical and experimental background to the 1931 CIE system of colorimetry. In *Colorimetry*, pages 9–23. John Wiley & Sons, Ltd, 2007. ISBN 978-0-470-17563-7.

Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, Nov. 1993. ISSN 1939-3539. doi: 10.1109/34.244673.

X. Xia and B. Kulis. W-Net: A Deep Model for Fully Unsupervised Image Segmentation. *arXiv:1711.08506 [cs]*, Nov. 2017.

B. Xu, W. Wang, G. Falzon, P. Kwan, L. Guo, G. Chen, A. Tait, and D. Schneider. Automated cattle counting using Mask R-CNN in quadcopter vision system. *Computers and Electronics in Agriculture*, 171:105300, Apr. 2020. ISSN 0168-1699. doi: 10.1016/j.compag.2020.105300.

H. Yao, Q. Yu, X. Xing, F. He, and J. Ma. Deep-learning-based moving target detection for unmanned air vehicles. In *2017 36th Chinese Control Conference (CCC)*, pages 11459–11463, July 2017. doi: 10.23919/ChiCC.2017.8029186.

J.-C. Yen, F.-J. Chang, and S. Chang. A new criterion for automatic multilevel thresholding. *IEEE Transactions on Image Processing*, 4(3):370–378, Mar. 1995. ISSN 1057-7149. doi: 10.1109/83.366472.

Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, Aug./1995. ISSN 01628828. doi: 10.1109/34.400568.

T. Young. II. The Bakerian Lecture. On the theory of light and colours. *Philosophical transactions of the Royal Society of London*, (92):12–48, 1802.

Y. Yuan, X. Chen, X. Chen, and J. Wang. Segmentation Transformer: Object-Contextual Representations for Semantic Segmentation. *arXiv:1909.11065 [cs]*, Apr. 2021.

C. Zahn. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, C-20(1):68–86, Jan. 1971. ISSN 1557-9956. doi: 10.1109/T-C.1971.223083.

N. M. Zaitoun and M. J. Aqel. Survey on Image Segmentation Techniques. *Procedia Computer Science*, 65:797–806, Jan. 2015. ISSN 1877-0509. doi: 10.1016/j.procs.2015.09.027.

J. Zhang and T. Tan. Brief review of invariant texture analysis methods. *Pattern Recognition*, 35(3):735–747, Mar. 2002. ISSN 0031-3203. doi: 10.1016/S0031-3203(01)00074-7.

R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering*, 3(5):616–630, Oct. 2017. ISSN 2095-8099. doi: 10.1016/J.ENG.2017.05.015.

P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling. Vision Meets Drones: Past, Present and Future. *arXiv:2001.06303 [cs]*, July 2020.

S.-C. Zhu, C.-e. Guo, Y. Wang, and Z. Xu. What are Textons? *International Journal of Computer Vision*, 62(1):121–143, Apr. 2005. ISSN 1573-1405. doi: 10.1023/B:VISI.0000046592.70770.61.

S. W. Zucker. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5(3):382–399, Sept. 1976. ISSN 0146-664X. doi: 10.1016/S0146-664X(76)80014-7.

Scientific Contributions and Publications

International Conferences

- Bazán E., Dokládal P., Dokládalová E. *Quantitative Analysis of Similarity Measures of Distributions*. The British Machine Vision Conference (BMVC), Cardiff, U.K., Sep. 2019.
- Bazán E., Dokládal P., Dokládalová E. *Unsupervised Perception Model for UAVs Landing Target Detection and Recognition*. Advanced Concepts for Intelligent Vision Systems. ACIVS, Sep. 2018.
- Bazán E., Dokládal P., Dokládalová E. *Non-supervised perceptual model for target recognition in UAVs*. Reconnaissance des Formes, Image, Apprentissage et Perception RFIAP, Jun. 2018.

Communications without Review

- Bazán E., Dokládal P., Dokládalová E. *Learning Perceptual Importance of Color and Texture for Unsupervised Segmentation*. French-German Doctoral Workshop, Kaiserslautern, Germany, Oct. 2019.
- Bazán E., Dokládal P., Dokládalová E. *The Optimal Transport for Image Segmentation*. French-German Doctoral Workshop, Fontainebleau, France, Nov. 2018.

Oral Presentations

- *Vision Methods for Aerial Vehicles' Autonomous Navigation*. Oral communication presented at the Optics Research Center, León, Guanajuato, México, Feb. 2019.
- *Non supervised perceptual model for target recognition in UAVs and Quantitative Analysis of Similarity Measures of Distributions*. Oral communication presented at the doctoral day (journée de doctorants de deuxième année JD2A), Paris, France, 2019.

RÉSUMÉ

Ce travail de thèse porte sur l'extraction de caractéristiques et de primitives de bas niveau à partir des informations perceptuelles de l'image pour comprendre des scènes. Motivés par les besoins et les problèmes de la navigation basée sur la vision des véhicules aériens sans pilote (UAV), nous proposons de nouvelles méthodes en nous concentrant sur les problèmes de compréhension de l'image. Ce travail explore trois informations principales dans une image : l'intensité, la couleur et la texture.

Dans le premier chapitre du manuscrit, nous travaillons sur les informations d'intensité à travers les contours de l'image. Nous combinons ces informations avec des concepts issus de la perception humaine, tels que le principe de Helmholtz et les lois de la Gestalt, pour proposer un cadre non supervisé pour la détection et l'identification des objets. Nous validons cette méthodologie dans la dernière étape de la navigation par drone, juste avant l'atterrissage.

Dans les chapitres suivants du manuscrit, nous explorons les informations de couleur et de texture contenues dans les images. Tout d'abord, nous présentons une analyse de la couleur et de la texture en tant que distributions globales d'une image. Cette approche nous amène à étudier la théorie du transport optimal et ses propriétés comme véritable métrique de comparaison des distributions de couleur et de texture. Nous passons en revue et comparons les mesures de similarité les plus populaires entre les distributions pour montrer l'importance d'une métrique avec les propriétés correctes, telles que la non-négativité et la symétrie. Nous validons ces concepts dans deux systèmes de récupération d'images basés sur la similitude de la distribution des couleurs et de la distribution de l'énergie des textures. Enfin, nous construisons une représentation d'image qui exploite la relation entre les informations de couleur et de texture. La représentation de l'image résulte de la décomposition spectrale de l'image, que l'on obtient par convolution avec une famille de filtres de Gabor. Nous présentons en détail les améliorations apportées au filtre Gabor et les propriétés des espaces colorimétriques complexes. Nous validons notre méthodologie avec une série d'algorithmes de détection des limites et de segmentation basés sur l'espace des caractéristiques perceptuelles calculé.

MOTS CLÉS

traitement d'image, primitives de bas niveau, perception humaine, détection, segmentation, méthodes non supervisées, compréhension de scène, apprentissage automatique, drone.

ABSTRACT

This thesis work deals with extracting features and low-level primitives from perceptual image information to understand scenes. Motivated by the needs and problems in Unmanned Aerial Vehicles (UAVs) vision-based navigation, we propose novel methods focusing on image understanding problems. This work explores three main pieces of information in an image : intensity, color, and texture.

In the first chapter of the manuscript, we work with the intensity information through image contours. We combine this information with human perception concepts, such as the Helmholtz principle and the Gestalt laws, to propose an unsupervised framework for object detection and identification. We validate this methodology in the last stage of the drone navigation, just before the landing.

In the following chapters of the manuscript, we explore the color and texture information contained in the images. First, we present an analysis of color and texture as global distributions of an image. This approach leads us to study the Optimal Transport theory and its properties as a true metric for color and texture distributions comparison. We review and compare the most popular similarity measures between distributions to show the importance of a metric with the correct properties such as non-negativity and symmetry. We validate such concepts in two image retrieval systems based on the similarity of color distribution and texture energy distribution. Finally, we build an image representation that exploits the relationship between color and texture information. The image representation results from the image's spectral decomposition, which we obtain by the convolution with a family of Gabor filters. We present in detail the improvements to the Gabor filter and the properties of the complex color spaces. We validate our methodology with a series of segmentation and boundary detection algorithms based on the computed perceptual feature space.

KEYWORDS

Image Processing, Low-level Primitives, Human Perception, Detection, Segmentation, Unsupervised Methods, Scene Understanding, Machine Learning, UAV.