Information spotting in huge repositories of scanned document images

Quoc Bao Dang

University of La Rochelle

Doctoral Thesis

Information spotting in huge repositories of scanned document images

Author:
Quoc Bao Dang

Supervisor:
Prof. Jean-Marc Ogier
Co-Supervisor:
Assoc. Prof. Mickaël Coustaty
Dr. Muhammad Muzzamil Luqman
Assoc. Prof. Cao De Tran

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Laboratory of Informatics, Image and Interaction
Faculty of Science and Technology
University of La Rochelle, France

6 April 2018

Composition du jury :

- M. CAO DE Tran, Maitre de conférences, HDR, Université de Can Tho, Vietnam (Co-Director)
- Mme EGLIN Véronique, Professeur, INSA de Lyon, France (le Président du jury)
- M. FALQUET Gilles, Professeur, Université de Genève, Switzerland (European evaluator)
- M. MARINAI Simone, Professeur, Université de Florence, Italy (European evaluator)
- M. OGIER Jean-Marc, Professeur, Université de la Rochelle, France (Director)
- M. RUSINOL Marcal, Chercheur, Université Autonome de Barcelone, Spain (Jury committee)
- Mme VINCENT Nicole, Professeur, Université Paris Descartes, France (Jury committee)

Résumé

Ce travail vise à développer un cadre générique qui est capable de produire des applications de localisation d'informations à partir d'une caméra (webcam, smartphone) dans des très grands dépôts d'images de documents numérisés et hétérogènes via des descripteurs locaux. Le système développé dans cette thèse repose sur une requête image (acquise via une caméra) et le système est capable de renvoyer le document qui correspond le mieux à la requête, et également d'indiquer la zone visée par la caméra. Ainsi, dans cette thèse, nous proposons d'abord un ensemble de descripteurs qui puissent être appliqués sur des contenus aux caractéristiques génériques (composés de textes et d'images) dédié aux systèmes de recherche et de localisation d'images de documents. Nos descripteurs proposés comprennent SRIF, PSRIF, DELTRIF et SSKSRIF qui sont construits à partir de l'organisation spatiale des points d'intérêts les plus proches autour d'un point-clé pivot. Tous ces points sont extraits à partir des centres de gravité des composantes connexes de l'images. A partir de ces points d'intérêts, des caractéristiques géométriques invariantes aux dégradations sont considérées pour construire nos descripteurs. SRIF et PSRIF sont calculés à partir d'un ensemble local des m points d'intérêts les plus proches autour d'un point d'intérêt pivot. Quant aux descripteurs DELTRIF et SSKSRIF, cette organisation spatiale est calculée via une triangulation de Delaunay formée à partir d'un ensemble de points d'intérêts extraits dans les images. Cette seconde version des descripteurs permet d'obtenir une description de forme locale sans paramètres. En outre, nous avons également étendu notre travail afin de le rendre compatible avec les descripteurs classiques de la littérature qui reposent sur l'utilisation de points d'intérêts dédiés, comme par exemple SURF ou SIFT, de sorte qu'ils puissent traiter la recherche et la localisation d'images de documents à contenu hétérogène.

La seconde contribution de cette thèse porte sur un système d'indexation de très grands volumes de données à partir d'un descripteur volumineux. Ces deux contraintes viennent peser lourd sur la mémoire du système d'indexation. En outre, une la très grande dimensionnalité des descripteurs peut amener à une réduction de la précision de l'indexation, réduction liée au problème de dimensionnalité. Nous proposons donc trois techniques d'indexation robustes, qui peuvent toutes être employés sans avoir besoin de stocker les descripteurs locaux dans la mémoire du système. Cela permet, in fine, d'économiser la mémoire et d'accélérer le temps de recherche de l'information, tout en s'abstrayant d'une validation de type distance. Pour cela, nous avons proposé trois méthodes s'appuyant sur des arbres de décisions : " randomized clustering tree indexing" qui hérite des propriétés des "kd-tree", "kmean-tree" et les "random forest" afin de sélectionner de manière aléatoire les K dimensions qui permettent de combiner la plus grande variance expliquée pour chaque nœud de l'arbre. Nous avons également proposé une version

pondérée de la distance euclidienne entre deux points de données, afin d'orienter celle-ci vers la dimension avec la variance la plus élevée. Enfin, pour améliorer la recherche de l'information, une fonction de hachage a été proposée en second lieu pour indexer et récupérer rapidement les contenus sans stocker les descripteurs dans la base de données. Nous avons également proposé une fonction de hachage étendue pour l'indexation de contenus hétérogènes provenant de plusieurs couches de l'image. Comme troisième contribution de cette thèse, nous avons proposé une méthode simple et robuste pour calculer l'orientation des régions obtenues par le détecteur MSER, afin que celui-ci puisse être combiné avec des descripteurs dédiés (par exemple SIFT, SURF, ORB, etc.). Comme la plupart de ces descripteurs visent à capturer des informations de voisinage autour d'une région donnée, nous avons proposé un moyen d'étendre les régions MSER en augmentant le rayon de chaque région. Cette stratégie peut également être appliquée à d'autres régions détectées afin de rendre les descripteurs plus distinctifs. Là encore, nous avons utilisé une méthode d'indexation basée sur une fonction de hachage étendue afin d'indexer des contenus hétérogènes aux caractéristiques multiples (textes, graphiques, etc.) à partir d'une décomposition des images en couches. Ce système est donc applicable pour les contenus uniformes (un seul type d'information), mais également pour plusieurs types d'entités à partir de plusieurs couches séparées. Enfin, afin d'évaluer les performances de nos contributions, et en nous fondant sur l'absence d'ensemble de données publiquement disponibles pour la localisation d'information hétérogène dans des images capturées par une caméra, nous avons construit trois jeux de données qui sont disponibles pour la communauté scientifique. Cet ensemble de données contient des parties d'images de documents acquises via une caméra en tant que requête. Il est composé de trois types d'informations: du texte, des contenus graphiques et enfin des contenus hétérogènes.

MOTS-CLÉS : Reconnaissance de formes, Spotting d'informations, Recherche de document à partir d'une caméra, Indexation automatique, Séparation texte/graphique, Extraction de caractéristiques.

Abstract

This work aims at developing a generic framework which is able to produce camera-based applications of information spotting in huge repositories of heterogeneous content document images via local descriptors. The targeted systems may take as input a portion of an image acquired as a query and the system is capable of returning focused portion of database image that match the query best.

We firstly propose a set of generic feature descriptors for camera-based document images retrieval and spotting systems. Our proposed descriptors comprise SRIF, PSRIF, DELTRIF and SSKSRIF that are built from spatial space information of nearest keypoints around a keypoints which are extracted from centroids of connected components. From these keypoints, the invariant geometrical features are considered to be taken into account for the descriptor. SRIF and PSRIF are computed from a local set of m nearest keypoints around a keypoint. While DELTRIF and SSKSRIF can fix the way to combine local shape description without using parameter via Delaunay triangulation formed from a set of keypoints extracted from a document image. Furthermore, we propose a framework to compute the descriptors based on spatial space of dedicated keypoints e.g SURF or SIFT or ORB so that they can deal with heterogeneous-content camera-based document image retrieval and spotting.

In practice, a large-scale indexing system with an enormous of descriptors put the burdens for memory when they are stored. In addition, high dimension of descriptors can make the accuracy of indexing reduce. We propose three robust indexing frameworks that can be employed without storing local descriptors in the memory for saving memory and speeding up retrieval time by discarding distance validating. The randomized clustering tree indexing inherits kd-tree, kmean-tree and random forest from the way to select K dimensions randomly combined with the highest variance dimension from each node of the tree. We also proposed the weighted Euclidean distance between two data points that is computed and oriented the highest variance dimension. The secondly proposed hashing relies on an indexing system that employs one simple hash table for indexing and retrieving without storing database descriptors. Besides, we propose an extended hashing based method for indexing multi-kinds of features coming from multi-layer of the image.

Along with proposed descriptors as well indexing frameworks, we proposed a simple robust way to compute shape orientation of MSER regions so that they can combine with dedicated descriptors (e.g SIFT, SURF, ORB and etc.) rotation invariantly. In the case that descriptors are able to capture neighborhood information around MSER regions, we propose a way to extend MSER regions by increasing the radius of each

region. This strategy can be also applied for other detected regions in order to make descriptors be more distinctive.

Moreover, we employed the extended hashing based method for indexing multi-kinds of features from multi-layer of images. This system are not only applied for uniform feature type but also multiple feature types from multi-layers separated.

Finally, in order to assess the performances of our contributions, and based on the assessment that no public dataset exists for camera-based document image retrieval and spotting systems, we built a new dataset which has been made freely and publicly available for the scientific community. This dataset contains portions of document images acquired via a camera as a query. It is composed of three kinds of information: textual content, graphical content and heterogeneous content.

KEYWORDS: Pattern recognition, Information spotting, Camera-based document image retrieval, Automatic indexing, Text/graphic separation, Feature extraction.

Acknowledgements

I would like to express my deep and sincere appreciation to my Ph.D. supervisors who are Prof. Jean-Marc Ogier, Assoc. Prof. Mickaël Coustaty, Dr. Muhammad Muzzamil Luqman and Assoc. Prof. Cao De Tran. They are kindly and wholeheartedly provide a lot of support to my work. They always make my research environment be free so that I can really focus on the works that I am interested in. In addition, with the wide knowledge and constructive advice of them, I am inspired with various ideas and new directions in order to solve the challenges. Without their guidance and help during my studying period, this research would be impossible.

I would also like to thank the CVC (Computer Vision Center Universitat Autònoma de Barcelona) members for having welcomed me as a European doctorate, especially Marçal Rusiñol who instructed and helped me during the time I worked there.

I also would like to thank all my lab fellows, who always have great ideas and willing to discuss and to give me exciting solutions about difficult problems in my research work. They help me a lot in academic fields as well non-academic aspects, especially give me a very cordial research environment. I would like to thank Ms. Marwa Mansri who helped me to construct the dataset and groundtruth. Furthermore, I wish to extend my warm thanks to all my friends who help me during my Ph.D. studying in France. I would not be able to overcome difficulties and have so many happy and memorable moments without their support and encouragement.

My special gratitude is to Project 165 of the Vietnamese government and L3i laboratory, University of La Rochelle for their financial support. I also thank Dong Thap University and Dong Thap Provincial Committee of the Party for giving me a golden opportunity to study in France.

Last but not least, I would like to give my most sincere gratitude to my family who always provide their support to anything I would like to do, and who understand any of my bad or good mood unconditionally. I also wish to express my special appreciation to my wife and my son, who accompanies me every day along the way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A huge amount of paper-based documents is produced in everyday life. These documents are generally printed or written on the papers e.g manuscripts, business contracts or letters, published books or magazines, handwritten notes or records, posters and so on. Besides the trend to move toward a paperless world in the digital era, many important and valuable paper-based documents need to be converted and stored as images which can be saved in electronic devices and can be exchanged or shared through a computer network. The explosion of these document images has created an enormous demand to access and manipulate the information contained in these images via robust systems. Therefore, the research of automatic extraction, classification, clustering and searching of information from such a large amount of data is worthwhile [7–16].

Information in these documents is heterogeneous and can be classified into two major groups: textual elements and graphical elements. The graphical elements can prevail in various forms such as symbols, logos, seals, signatures, photographs, etc. These graphical elements often provide more obvious and more compact in terms of conveying information compared to text. Yet, the text is a really convenient way to share information and tend to have a large proportion in both type-written documents and hand-written documents [7–9]. Traditional document images indexing and retrieval systems try to convert the document to an electronic representation which can be indexed automatically. A complete conversion being able to index both text and graphics is however difficult to implement. Indeed, documents images may contain many graphical components and hand-written text that are generally not able to be converted with a sufficient accuracy to provide a complete indexing system.

Document Image Retrieval (DIR) is a research domain, which belongs to the frontier between classic **Information Retrieval** (IR) and **content based image retrieval** (CBIR) [17]. Document Image Retrieval is the task to find information or similar

document images from a large dataset for a given user query. These approaches can be divided into two groups including recognition-based approaches and recognition-free approaches.

The former group which is often applied for type-written documents and needs to perform the recognition of whole documents and measures the similarity between documents at the symbolic level using Optical Character Recognition (OCR). Document images are firstly converted into text format using OCR, and then text retrieval techniques are applied for information retrieval step. For example, Viola et al. presented in [18] a system aiming at automatically forwarding incoming faxes to the correspondent recipient. OCR generally works quite well with type-written documents in which character font and size can be predefined, and generally in the context for which text and background can be easily distinguished. However, OCR-based approaches also have some drawbacks such as high computational cost, language dependency, and they are generally sensitive to image resolution, especially in the context of historical documents and documents captured by cameras. In these cases, employing recognition-based approaches cannot provide efficient results. Furthermore, one important drawback of these approaches is linked to the fact that OCR can difficultly deal with hand-written elements and graphical elements. Consequently, it is impossible to preserve document images as a full-text format by applying OCR on the whole documents, especially when the documents contain non-text elements that cannot be converted with sufficient accuracy. Either way, directly indexing converted document images using OCR is a very common task in many industrial process that produces many errors because of OCR's drawbacks and new research trends try to correct OCR errors without being actually able to satisfy the end-users needs [19].

The later group relies on the computation of features that are computed at a low level and that are generally based on its content. The similarity between two documents is then measured using those features summarizing the content of the document image (textual contents, graphical contents). Features can finally be globally or locally computed from the pixel information without using OCR-based methods.

These features are considered as a natural watermark for every document without embedding any special visual pattern. As it can be seen in the literature, local features have been widely used in computer vision and are considered as much more relevant than global approaches [20–22]. The local features generally require detecting Regions Of Interest (ROIs) that remain stables even under certain classes of transformations such as affine transformations and/or perspective distortions in order to cope with challenges of camera based document images. ROIs are generally determined through keypoints which are detected from stable corners or stable blobs [20, 21]). Then, for each detected

region, some radiometric features are locally computed to build the local feature. These features are generally inspired by the human visual system that can perform several image processing tasks in vastly superior manner comparing with computer vision system. For the past decades, the approaches in this group have been promisingly applied for textual documents and graphical documents, and they have been considered as an emerging research topic.

The explosion of the number of portable digital imaging devices has created a tremendous opportunity for **Camera Based Document Analysis and Recognition** (CBDAR). For example, some augmented reality tools appear to provide similar contents (e.g. newspapers and magazine articles) to the users by simply capturing an image with their smartphones or cameras [23–25]. Users have now access to a huge amount of content on the Internet and a big challenge is to offer some tools to link real documents to those captured using digital devices. A typical architecture of these kinds of system is shown in Figure 1.1.

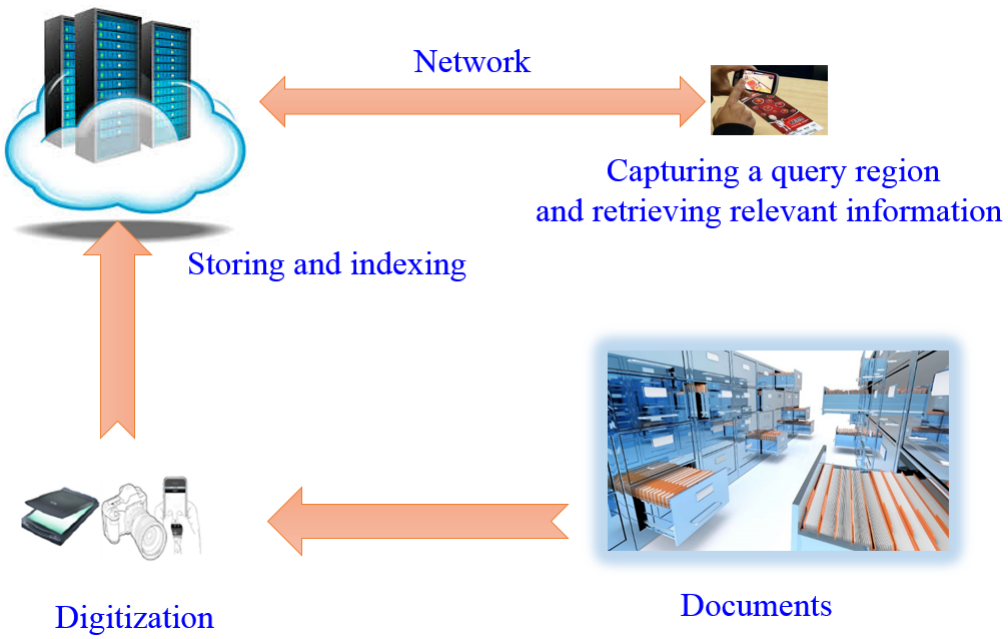

FIGURE 1.1: The architecture of typical camera-based document image retrieval system.

Camera-based document image retrieval systems take as input a part or the whole page document acquired as a query by a digital camera and retrieve document images that include the query [23–25]. These systems generally require to tackle many problems, as also listed by [26]:

1. Images captured with cameras usually have a low resolution.

2. A camera has much less control of lighting conditions on an object compared to a flatbed scanner, so uneven observed lighting can be due to both the physical environment and the response from the device.

3. Perspective distortion problems can occur as the capture device is not parallel to the imaging plane.

4. Since digital devices are designed to operate over a variety of distances, focus becomes a significant factor. At short distances and large apertures, even slight perspective changes can cause uneven focus.

5. The nature of mobile devices suggest that either the device or the target may be moving, which can cause motion blur problems.

6. Lastly, the acquisition of images from a camera generally results in the capture of a subpart of the original image. The retrieval process can be seen as a sub-matching process between the digitized image and the original one. This then consists of establishing a way of efficiently matching document images.

There have been various applications which can be developed in CBDAR field. For instance, we can deploy a file searching system on smartphones. With this system, users can retrieve the original electronic document online by capturing a snapshot from a paper based document even if the quality of this document is not good enough for us to read and users can also add comments or annotation or send feedbacks to the system. More importantly, this document retrieval system can help in reducing the cost of document management from the work of searching the original documents in huge repositories of scanned document images [27].
In addition, publishers can get information of what readers read when they take a snapshot of the article via this system. Another application can be applied for the visually impaired people [28]. They can use this system to listen to the audio version of the article by retrieving it from a snapshot. In this system, documents have to be stored along with the audio versions synthesized using text to speech. Google Goggles permits users to search a book via the captured book cover. This application can recognize some artwork and bring back related information of the artwork. Users can also search a product based on wine marks and spencer in this system [29]. Kooaba's Paperboy supplies an interactive storytelling system about print ads to users. This application can direct customers to a nearby store when they capture an ads. It also provides product ingredients and origins that are supplied by food makers [30].

More recently, a new kind of information retrieval systems have been proposed in the literature alongside to the classical recognition techniques: **Information spotting**.

It can be defined as the task of locating and retrieving specific information from a large document image dataset, based on a given user query, without explicitly recognizing the content of the query. For instance, word spotting in textual document images coarsely locates some regions where the query word is found [31–35]; symbol spotting techniques try to identify regions which are likely to contain a certain symbol within graphics-rich documents [9, 36, 37]; logo spotting on document images aims to find the position of a set of regions of interest which are likely to contain an instance of a certain queried logo [36, 38]; comic characters spotting attempts to detect and then retrieve and/or locate comic characters in comic documents where reader's queried character appears [39, 40];

In conclusion, using image features based on its content without embedding any special visual pattern for CBDAR system in the context of information spotting has many advantages such as keeping good document fidelity and dealing with image transformations and cameras' problems. Yet, there is a lack of local features which can deal with the context of heterogeneous content documents. Furthermore, in the context of huge document repositories, the high dimensionality of these descriptors rises two more constraints regarding the curse of dimensionality on the one hand, and the computation time when dealing with real time matching systems [41].

1.1 Objectives and Contributions

This thesis is a step forward to achieve the objective of joining the advantages of recognition-free approaches for CBDAR system in the context of information spotting. The thesis aims at producing a camera-based application of information spotting in huge repositories of heterogeneous content document images via local descriptors.

The contributions of this thesis are three-fold:

- The first contribution of this thesis relies on new methods for computing generic feature descriptors for camera-based document images retrieval and spotting systems
- The second contribution of this thesis is indexing frameworks for automatic indexing of document image repositories
- The third contribution of this thesis is a dataset and ground-truth which is used to evaluate camera-based document images retrieval and spotting systems

1.2 Thesis Organization

The organization of the chapters in this dissertation is as follows:

- In Chapter 1, we provide a preview of the whole thesis including the scope of the thesis, the problems and contributions.
- In Chapter 2, we firstly present a literature review of the state of the art of local features extraction that includes keypoint detectors and keypoint descriptors. Afterwards, we present a brief overview of the literature and of indexing approaches. This panorama is categorized into two main groups including tree-based approaches and hashing-based approaches.
- In Chapter 3, we present several novel schemes towards features computation for heterogeneous-content camera-based document image retrieval. The first one is SRIF (Scale and Rotation Invariant Features), which is computed based on geometrical constraints between pairs of nearest points around a keypoint. In addition, we propose four extensions based on SRIF. The second one is PSRIF (Polygon-shape-based Scale and Rotation Invariant Features), which is an extension to SRIF and which makes SRIF more discriminative even though it is computed from a small number of constraint points around the keypoint. The third one is DE-TRIF (Delaunay triangulation-based features), which relies on the geometrical constraints from each pair of adjacency triangles in Delaunay triangulation which is constructed from centroids of connected components. The last one is SSKSRIF (Scale Rotation Feature descriptor based on Spatial Space of Keypoints), which also relies on the geometrical constraints from each pair of adjacency triangles in Delaunay triangulation using similarity transformation. In addition, we propose a framework to compute descriptors based on spatial space of dedicated keypoints such as SIFT, SURF or ORB. This aims at enhancing proposed features can deal with the context of heterogeneous-content camera-based document image retrieval and spotting.
- In Chapter 4, we present a contribution towards the implementation of an indexing system for camera-based document image retrieval using local features. The first proposed indexing scheme is based on randomized hierarchical trees. The second proposed hashing relies on an indexing system that employs one simple hash table for indexing and retrieving descriptors without storing them. Besides, we propose an extended hashing based method for indexing multi-kinds of features coming from multi-layer of the image.

- In Chapter 5, we present a dataset and evaluation protocol for camera-based document image retrieval and spotting systems. This dataset is composed of three subparts: the wikibook dataset represents the images with textual content only; The Cartodialect dataset represents images with graphical content mainly; the tobacco dataset contains text plus graphical contents. Along with the dataset, we present the protocol that describes measurements to evaluate the accuracy and processing time of camera-based document image retrieval and spotting systems.

  Afterwards, we present the experimental evaluations of proposed approaches on the one hand, approaches in literature on the other hand on our proposed dataset. In addition result discussions are given in this chapter.

- Conclusions are given in Chapter 6. A summary of the main contributions has been given firstly. Afterwards, we discuss some possible future perspectives.

Chapter 2

State-of-the-art

2.1 Introduction

Camera-based document image retrieval system takes an image or a part of an image acquired by a digital camera and tries to retrieve the document image that includes the query [23–25] among a large repository of documents. This task is very challenging for the characterization of the mobile captured content because captured images can be affected by uneven lighting, low resolution, motion blur and perspective distortion problems [26]. Furthermore, the acquisition of images with a camera generally results in the capture of a subpart of the original image. The retrieval process can thus be seen as a sub-matching process between the partly digitized image and the original one. In order to implement efficient real time systems, it is required to structure the information by using reliable indexing methods.

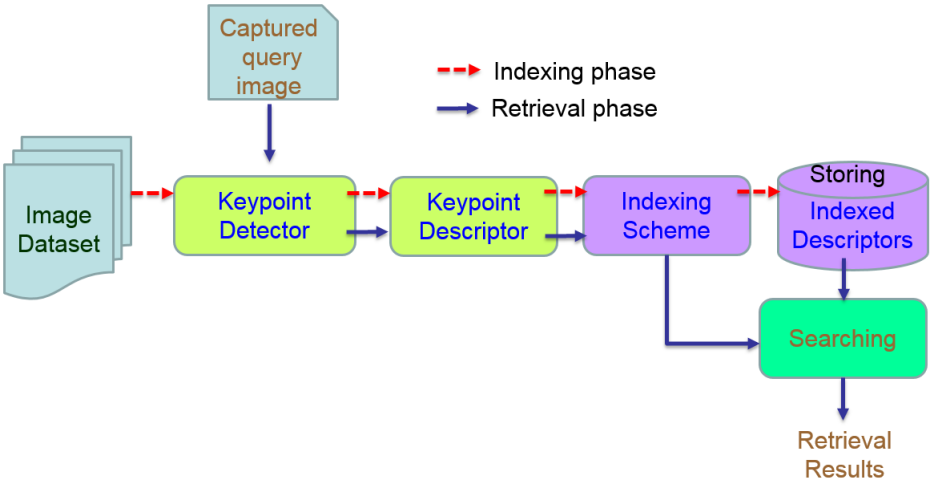

FIGURE 2.1: Camera-based document image retrieval using local feature.

From several decades, there have been many proposed methods using original document features (e.g. textual contents, graphical contents). These features are considered as a natural watermark for every document without embedding any special pattern. To compute these features, as it can be seen in the literature, local features have been widely used in computer vision and are considered as much more relevant than global approaches [20, 36, 38] especially for camera camera-based information spotting [22]. The local features based approaches need to detect Regions Of Interest (ROIs) that remain stables even under certain classes of transformations such as affine transformation and/or perspective distortion (e.g in order to cope with challenges of camera captured document images). ROIs are generally determined through keypoints which are detected from stable corners or stable blobs [20, 21]). Then, for each detected region, some radiometric features are computed locally to build the *local features* which are sometimes considered as bio-inspired.

The block diagram in Figure 2.1 gives an overview of a typical camera-based document image retrieval system using local features. There are two main phases respectively including the indexing phase and retrieval phase. Both of them share the feature extraction step, which comprises keypoint detector and descriptor. For the feature extraction and indexing phase, we usually have to choose suitable features and an adapted indexing method, respectively.

In this chapter, we firstly present a literature review on the state of the art concerning local features extraction including keypoint detectors and keypoint descriptors. Afterwards, we present a brief overview of the literature of indexing approaches which are categorized into two main groups including tree-based approaches and hashing-based techniques.

2.2 Keypoint detectors and keypoint descriptors

Keypoint detectors and keypoint descriptors correspond to the most widely used methods that computer vision community uses to characterize the content of an image, supposedly as good as human do. Generally, there are two ways which people may use to recognize a picture including global to local and vice versa. The first way is that people try to recognize what is the image and then discover more detail contents inside the image. The second way is that people look at some attractive points or regions in the image and discover relationship contents around them later towards understanding the image. This second way of considering human perceptual based systems is probably at the basis of keypoints based approaches that try to represent attractive points. From

a computer vision point of view, these attractive points are keypoints detectors and relationship contents around attractive points correspond keypoint descriptors.

Features proposed in the literature, for pattern recognition purposes, can be divided into two groups: global features and local features. The global ones try to globally summarize the information contained in the image, while the local ones tend to extract some regions of interest and describe the content of the image through the properties of these specific regions. In this section, we focus on the brief overview of the literature related to local feature detection and description.

Local features are normally relied two main steps: keypoint detection and keypoint description (as shown in Figure 2.2). Keypoint detectors try to notice Regions Of Interest (ROIs) that are considered as the anchor points/regions for local descriptors. These keypoints are generally stable under image transformations as well as viewing transformations. Local image contents of each keypoint are then used for computing local descriptors. Consequently, a set of descriptors allows characterizing the local patches of image content. An ideal local feature should adapt six major properties [20] such as: *repeatability, distinctiveness/informativeness, locality, quantity, accuracy and efficiency*

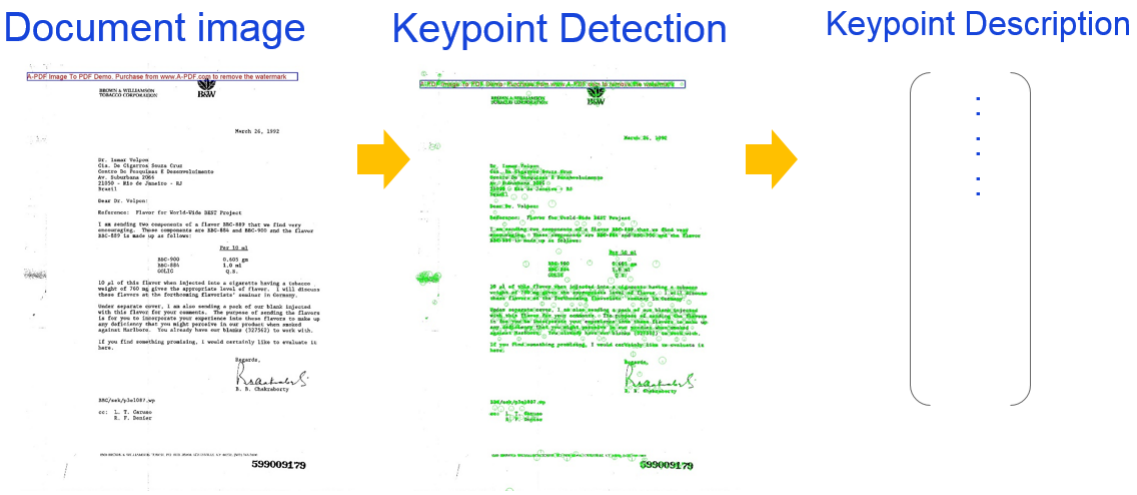

FIGURE 2.2: Local feature detection and description.

Repeatability requires a high percentage of the features detected in two images of the same object or scene even if they are captured under different viewing conditions.

Distinctiveness/informativeness refers to the fact that local features describe the intensity patterns underlying detected ROIs, that should contain distinctive information such that features can be well distinguished and matched.

Locality properties of feature descriptors help them to reduce the probability of occlusion and to preserve features relationships of spatial relations over simple geometric and photometric deformations.

Quantity ensures the number of detected features that should be sufficient so that a reasonable number of features are detected even on small objects. Yet, different applications need a particular optimal number of features.

Accuracy requires that the local information of detected features should be accurate not only in image location, as well as with respect to scale and possibly shape.

Efficiency makes sure that both the feature detection and description in a new image should be fast enough and can be applied for real-time applications.

2.2.1 Keypoint detectors

A keypoint may be composed of various types of corner, blob, maximum shapes, etc. In practice, a robust keypoint must be easy to find and ideally fast to compute. People usually set their sight on that the keypoint is at a good location to compute a feature descriptor. Therefore, the keypoint can be considered as the qualifier from which a local feature may be described.

Normally, a typical keypoint region that can be used for deriving a descriptor may have the following properties, respectively including the coordinates of the keypoint (x, y), the diameter of the meaningful surrounding region around the keypoint, the scale space s from which the keypoint is extracted in the image and the orientation of the keypoint relatively to the image coordinate system that helps descriptors to be built and normalized local features around the keypoint, especially when rotation invariant property is needed.

In this section, we review some common keypoint detectors that are grouped into two categories including corner-based detectors and blob-based detectors.

2.2.1.1 Corner-based detectors

Corner detection has been widely used in computer vision for keypoint detection. Corners can be found at various types of junctions, on highly textured surfaces, at occlusion boundaries, etc. In practical applications, the goal is to have a set of stable and repeatable corners. To detect corners, Harris corner detector [42] detects a point for which

there are two dominant and different edge directions in its neighborhood. FAST detector [1] detects a corner by checking a point which has different intensities with a set of pixels in a circular pattern.

Harris corner detector

Harris corner detector, proposed by Harris and Stephens [42], is based on the second moment matrix (the auto-correlation matrix). A corner is detected from the pixel where the intensity changes significantly in at least 2 directions. This detector is based on the idea of the Moravec's corner detector [43] and reduces the weaknesses of the Moravec's corner detector by using Gaussian window and Taylor's expansion. Indeed, the former uses a binary window function, the consequence of which is that it can not deal with noisy responses and also that it considers only a set of shifts at every 45 degree.

To detect Harris corners, Gaussian derivatives at each pixel of the input image are computed firstly. The next step consists in computing second moment matrix in a Gaussian window around each pixel. Lastly, the authors proposed corner response function called "cornerness" that reaches a maximum if the pixel is a corner.

From the input image $I(x, y)$, it basically finds the difference in intensity for a displacement of (u,v) in all directions. This difference is characterized by a kind of energy function, which is expressed as below:

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2 = \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}, \tag{2.1}$$

where $w(x, y)$ is a window function. It is a gaussian window (equation 2.2) which gives weights to pixels underneath.

$$w(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.2}$$

$I(x + u, y + v)$ is shifted intensity and $I(x, y)$ is intensity. M is the second moment matrix:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \tag{2.3}$$

here I_x and I_y are image derivatives in x and y directions respectively.

Finally, let λ_1 and λ_2 be the eigenvalues of M. To reduce the computational complexity, the authors proposed the computation of score, which determines whether or not the window can contain a corner. The score combines the two eigenvalues as follows:

$$sroce = det(M) - k\ trace(M)\,, \tag{2.4}$$

where $det(M)$ is the determinant of the matrix M ($det(M) = \lambda_1\lambda_2$) and $trace(M)$ is the trace of the matrix M ($trace(M) = \lambda_1 + \lambda_2$). A typical value for k is 0.04 to 0.06. If the $|score|$ is small, which happens when λ_1 and λ_2 are small, the region is flat. If $score < 0$, which happens when $\lambda_1 >> \lambda_2$ or vice versa, the region is an edge. If score is large, which happens when λ_1 and λ_2 are large and $\lambda_1 \sim \lambda_2$, the region is a corner.

The Harris keypoints are invariant to rotation. This detector finds locations with the large gradient in all directions at a predefined scale that contains corners. However, this detector can not deal with scaling problem. In order to make this detector being able to deal with this challenge, Harris-Affine detector is an affine-invariant keypoint detector relying on Harris keypoints. It firstly detects feature points using the Harris-Laplace detector [44]. Then, it iteratively refines these regions to affine regions using the second moment matrix [44].

Good Features To Track (GFTT) was proposed in [45] by Shi-Tomasi, et al. This detector is also based on Harris corner detectors but relies on a small modification of the scoring function. It provided better results compared to Harris Corner Detector in tracking systems. The scoring function in Harris Corner Detector was given by: $score = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$. Instead of this, Shi-Tomasi proposed: $score = min(\lambda_1, \lambda_2)$. If $score$ is a greater than a threshold value, the point is considered as a corner.

FAST

FAST, which stands for Features from Accelerated Segment Test, was proposed by Edward Rosten and Tom Drummond in the paper "Machine learning for high-speed corner detection" [1]. It can be considered the development of SUSAN detector [46]. SUSAN computes the fraction of pixels within a circle neighborhood which have similar intensity to the center pixel, SUSAN corners can then be localized by thresholding this measure and selecting local minima. FAST relies on a connected set of pixels in a circular pattern to determine a corner. This connected region size is commonly 9 or 10 out of a possible 16 as shown in figure 2.3.

Let p be a pixel in the image and its intensity be I_p. A binary comparison with each pixel in a circular pattern against the center pixel using a threshold is done to determine p is a corner or not. The pixel p is a corner if there exists a set of n contiguous pixels in the circular pattern which are all brighter than $I_p + t$, or all darker than $I_p - t$, n was chosen to be 12.

Most of local binary descriptors employ FAST detectors. In order to avoid detecting edges, n must be larger than nine and the FAST with $n = 9$ (FAST-9) is usually used. As the FAST detector is not scale-invariant, in order to ensure approximate scale invariance,

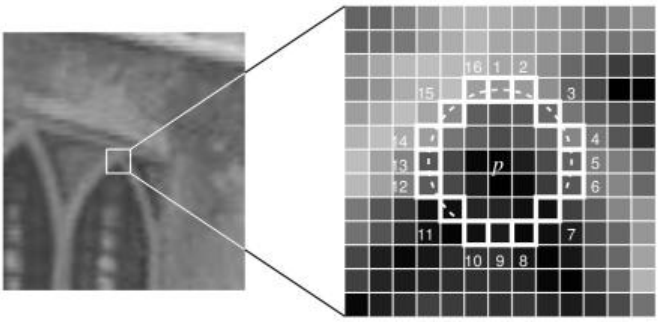

FIGURE 2.3: An example circular pattern of 16 pixels (extracted from [1])

feature points can be detected from an image pyramid, which can thus be considered the multi-scale FAST detector [47].

ORB

ORB [47], which stands for Oriented FAST and Rotated BRIEF, is basically a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance. It uses FAST to find keypoints, before applying Harris corner measure to find top N points among them. It also uses a pyramidal approach (multi-scale) to produce multi-scale features.

BRISK

BRISK detector, Binary Robust Invariant Scalable Keypoints, is proposed by Leutenegger et al. [48]. To localize the key-point, it uses the AGAST corner detector [49] which improves FAST detector by increasing the computation speed while maintaining the same detection performance. To deal with scaling problem, BRISK detects keypoints in a scale-space pyramid, performing non-maximal suppression and interpolation across all scales. Finally, scales and positions of the detected local regions are refined in a similar way to the DOG detector [50].

2.2.1.2 Blob-based detectors

Blob detection methods are aimed at detecting regions that diverge from surrounding regions in properties such as brightness or color, etc. Informally, a blob is a region of an image in which some properties are constant or approximately constant, or all the points in a blob can be considered similar to each other. After corner detectors, aiming to provide complementary information about regions that can not be obtained from edges or corners, many blob-based detectors have been developed.

DOG detector

Difference-of-Gaussian (DoG) keypoint, which was proposed in [50], is a method to efficiently detect stable key-point locations in scale space. This is also called SIFT detector. It includes three main steps. Firstly, a Gaussian-pyramid is constructed by doing gradually Gaussian-blur the input-image. Next, the Difference of Gaussian (DOG) pyramid is built by computing the difference of two consecutive Gaussian-blurred images in the Gaussian pyramid. Lastly, key-point locations in the DOG space are found from local maximums and local minimums in the DOG space and the locations and scales of these maximums and minimums. The advantage of Gaussian smoothing is that it helps in discarding noise that would be amplified and result in false DoG features.

From the input image $I(x, y)$, the Gaussian-blur image at the scale σ is a funnction $L(x, y, \sigma)$ which is computed as follows:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)\,, \tag{2.5}$$

where $*$ is the convolution operation, $I(x, y)$ is the original image and $G(x, y, \sigma)$ is the 2D Gaussian kernel at scale σ:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\,. \tag{2.6}$$

The difference-of-Gaussian function $D(x, y, \sigma)$ between two consecutive Gaussian-blurred images in the Gaussian pyramid σ and $k\sigma$ is computed using equation 2.7:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= I(x, y, k\sigma) - I(x, y, \sigma)\,. \end{aligned} \tag{2.7}$$

In the final step, a keypoint is selected in the DOG space when it is either larger or smaller than all its neighbors which are from over three neighborhood scales (3 x 3 x 3 neighborhood pixels) from the current scale and two nearby scales.

To filter unstable extrema with low contrast and edge responses, some authors also used Taylor series expansion and ratio between the largest magnitude eigenvalue with the smaller one from 2x2 Hessian matrix computed at the location and scale of the keypoint as two thresholds, respectively [50].

SURF detector

SURF detector is also called *Fast-Hessian detector*, as initially proposed by Bay et al. [51]. This detector detects keypoints from a multi-scale image set where the determinant of the Hessian matrix is at a maximum, by using integral images to calculate the

Gaussian partial derivatives and the Hessian Matrix. It is considered as a speeded-up version of SIFT detector.

The Hessian matrix $\mathcal{H}(p,\sigma)$ in a point $p(x,y)$ in an image I at scale σ is defined as follows:

$$\mathcal{H}(p,\sigma) = \begin{bmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{xy}(p,\sigma) & L_{yy}(p,\sigma) \end{bmatrix}, \tag{2.8}$$

where $L_{xx}(p,\sigma)$ is the convolution of the Gaussian second order derivative, as shown in equation 2.9 , and similarly for $L_{yy}(p,\sigma)$ and $L_{xy}(p,\sigma)$:

$$L_{xx}(p,\sigma) = I(p) * \frac{\partial^2}{\partial x^2} g(\sigma)\,. \tag{2.9}$$

To reduce the computational cost, SURF uses the approximation for $\mathcal{H}(p,\sigma)$ as following:

$$\mathcal{H}_{approx} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}. \tag{2.10}$$

Blob-like structures are then detected at the location where the determinant is maximum using equation 2.11:

$$det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - 0.9D_{xy}^2\,. \tag{2.11}$$

SURF applies the scale-space using Gaussian approximation filters at each level of filter size in the scale space to extract interest points from images, which is similar to SIFT algorithm. Yet, the SIFT algorithm iteratively reduces the image size, whereas the SURF algorithm uses the integral images, allowing up-scaling of the filter at a constant cost, and SIFT uses the approximated Laplacian of Gaussian (LoG) with DoG function whereas SURF uses an approximation LoG with a box filter based on non-maximum suppression. Non-maximum supression is scanned along the image gradient direction and set any non-maximum pixel to zero. The advantages of this approximation is linked to the fact that the convolution with box filter can be easily calculated from integral images and can be done in parallel for different scales.

Lastly, interest point detection is performed using the non-maximum suppression over three neighborhood scales (3 x 3 x 3 neighborhood pixels). The points that have the maximum of the determinant of the Hessian matrix are then regarded as the feature points.

MSER detector

Maximally Stable External Regions (MSER) detector was proposed by Matas et al. [52]. MSER regions are connected areas characterized by almost uniform intensity, surrounded by contrasting background. These regions are detected through a process consisting of

trying multiple thresholds and by selecting the regions for which one can be observed a maintain of unchanged shapes over a large set of thresholds.

Let a binary image I_t of an image I at threshold level t be defined as follows:

$$I_t(x) = \begin{cases} 1 & \text{if } I(x) \geq t \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.12}$$

It means that all the pixels below a given threshold are white and all those above or equal are black.

Centroid of connected components Centroids of connected components have qualities that are stable identically even under the perspective distortion, noise, and low resolution [25, 28, 53, 54]. In [53, 54], the authors use centroids of segmented words as keypoints that are used to build LLAH (Locally Likely Arrangement Hashing) descriptors for camera based textual document image retrieval. Similarly, Kise *et al.* [6] extracted centroids of letter connected components for LLAH in the system with camera-pen [6].

In summary, keypoint detectors detect distinguishable regions from an image. Detectors can be characterized by region type and invariance type. The region type represents the shape of a detected point or region such as corner or blob. The invariance type represents the transformations which the detector is robust to. The transformation can be a rotation, a similarity transformation or an affine transformation. Corresponding various applications, it is important to choose an appropriate detector with a specific invariance. An overview of keypoint detectors is shown in the Table 2.1. Normally, to deal with image scaling problem, some particular keypoints need to be detected in scale space (multi-scale) e.g ORB, BRISK, SIFT, SURF.

In next section 2.2.2, we will present a survey to examine a range of keypoint descriptor approaches.

2.2.2 Keypoint descriptors

In computer vision, various feature descriptors and metrics have been proposed along with many practical applications. For instance, cell biology and medical applications are typically interested in polygon shape descriptors. Augmented reality applications for mobile phones may be more interested in local binary descriptors. This section provides a survey and observations about a few representative feature descriptor methods. In practice, the feature descriptor methods are often modified and customized. The goal

TABLE 2.1: Keypoint detectors summation.

Keypoint detectors	Corner	Blob	Rotation invariant	Scale invariant	Affine invariant	Multi-scale detection
Harris	x		x			
FAST	x		x			
ORB(oFAST)	x		x	(x)		x
BRISK(AGAST)	x		x	x		x
SIFT(DoG)	(x)	x	x	x		x
SURF(Fast-Hessian)		x	x	x		x
MSER		x	x	x	x	
Centroid of CCs		(x)	x	x	x	

of this survey is to examine a range of feature descriptor approaches from each feature descriptor family of the taxonomy.

2.2.2.1 SIFT

Scale-Invariant Feature Transforms (SIFT), proposed in [50], is calculated on the gradient distribution of the region. Firstly, the Gaussian-smoothed image $L(x, y, \sigma)$ at the scale σ where the key-point is detected is applied (see Equation 2.5 and 2.6).

A gradient magnitude $m(x, y)$ and an orientation $\theta(x, y)$ of each point $I(x, y)$ in the region is computed:

$$m(x,y) = \sqrt{[I(x+1,y) - I(x-1,y)]^2 + [I(x,y+1) - I(x,y-1)]^2}\,, \tag{2.13}$$

$$\theta(x,y) = \tan^{-1}[I(x,y+1) - I(x,y-1)]/[I(x+1,y) - I(x-1,y)]\,. \tag{2.14}$$

Then, an orientation is assigned to each keypoint to achieve invariance to image rotation. The gradient magnitude of keypoints neighborhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction are calculated in that region (as shown in Figure 2.4). An orientation histogram with 36 bins covering 360 degrees is created. It is weighted by gradient magnitude and Gaussian-weighted circular window with σ equal to 1.5 times the scale of keypoint. The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. This may lead to the case that there are more than one key-point at the same position because there may be keypoints with the same location and scale, but different directions to be created at the position.

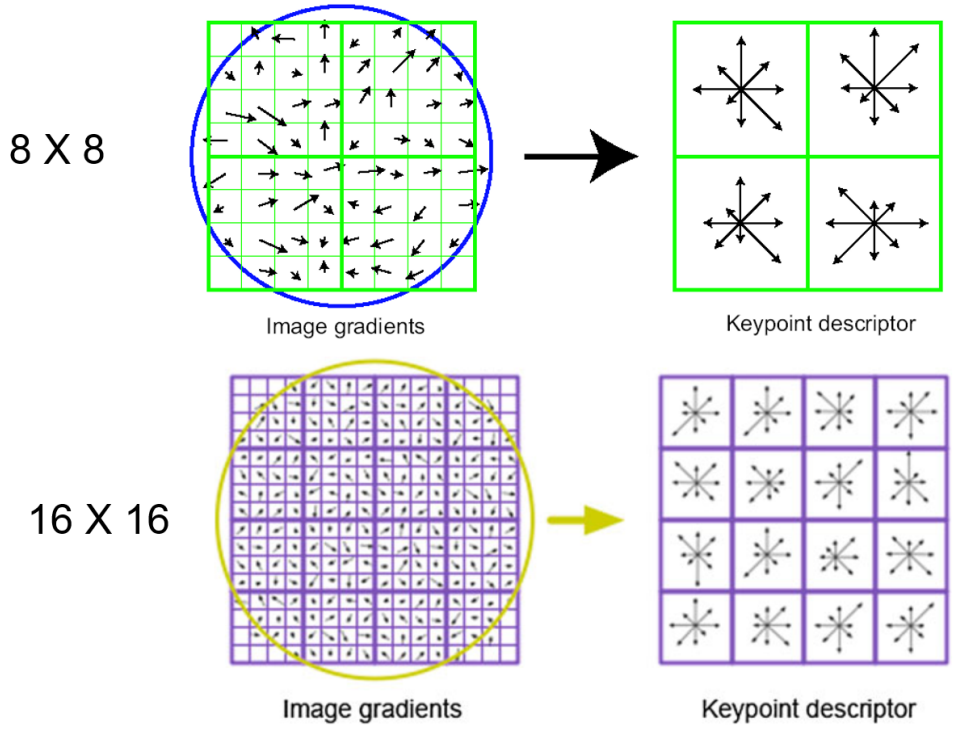

FIGURE 2.4: Example the computation of a SIFT descriptor in a region of size 8×8 and 16×16

Lastly, keypoint descriptor of each key-point is created. A 16x16 neighborhood around the keypoint is taken. It is divided into 16 sub-blocks of 4x4 size. For each sub-block, 8 bin orientation histogram is created. So a total of 128 bin values are available. It is represented as a vector to form keypoint descriptor. In addition to this, several measures are taken to achieve robustness against illumination changes, rotation etc.

2.2.2.2 SURF (Speeded Up Robust Features)

Speeded Up Robust Features (SURF) [51] estimate the orientation for the key-point by applying the Haar-wavelet response in two directions (horizontal and vertical) in a circular neighborhood of radius 6s, with s the scale where the key-point is detected (see Figure 2.5). A sliding orientation window (60 degrees) is used to find the dominant orientation by calculating the sum of all responses.

For feature description, SURF used a rectangular grid of 4×4 regions around the key-point to create the descriptor vector. Similar to SIFT, each region is also divided into 4×4 sub-regions. Then, the Haar-wavelet response is used to calculate each sub-region. Like SIFT, SURF uses a circularly symmetric Gaussian weighting factor for each Haar response. For each 4×4 sub-region, each feature vector contains four parts:

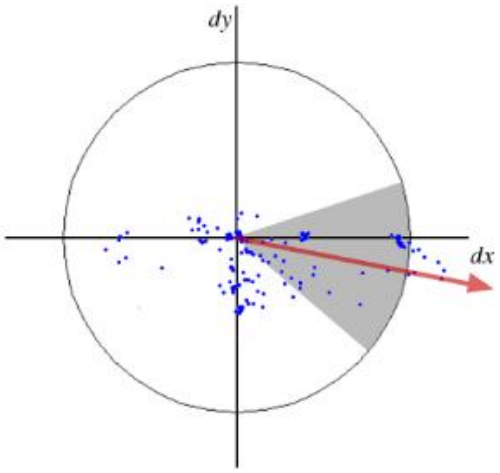

FIGURE 2.5: The orientation of key-point is calculated in a circular neighborhood using a sliding orientation window

$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ where, the wavelet responses d_x and d_y for each sub-region are summed, and the absolute value of the responses $|d_x|$ and $|d_y|$ provide polarity of the change in intensity. The final descriptor vector is $4 \times 4 \times 4$ dimension, which includes 4x4 regions with four parts per region [51].

2.2.2.3 Shape Context

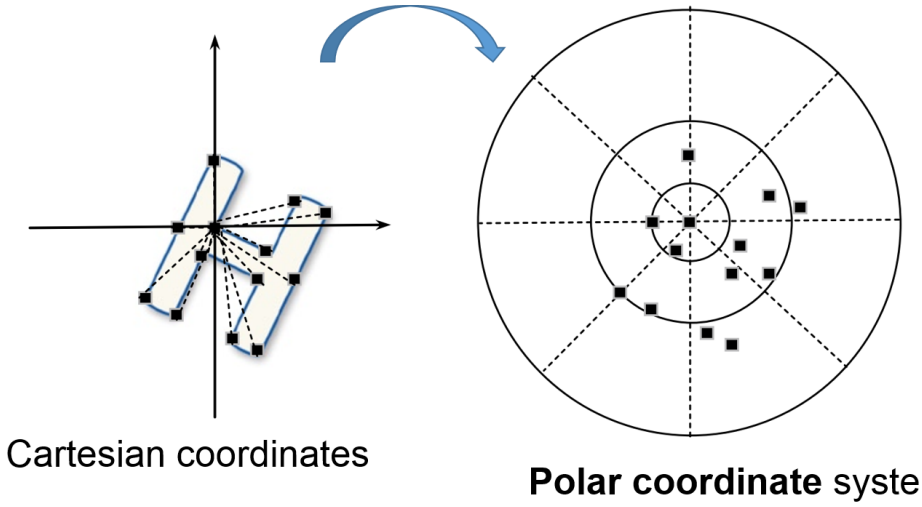

FIGURE 2.6: Example the computation of Shape Context descriptor.

The shape context descriptor was proposed by Belongie et al. in [55]. This descriptor allows measuring shape similarity by recovering point correspondences between the two shapes under analysis. In the first step, a set of interest points has to be selected from the object. Usually, a Canny edge detector is used and the edge elements

are sampled in order to obtain a fixed number of n points p_i per object. Given these n points, the shape context captures the distribution of points within the plane relative to the reference point origin that is chosen on the perimeter of the object as the origin. A histogram using log-polar coordinates counts the number of points inside each bin. For a reference point p_i of the shape, a histogram h_i of the coordinates of the nearby points q is computed as:

$$h_i(k) = \#\{q \neq p_i : q \in bin_{p_i}(k)\}\,. \tag{2.15}$$

The total bins is the number of bins for log r multiply the number of bins for θ(see Figure 2.6 for an example). To achieve scale invariance, all radial distances are normalized by the mean distance between the n^2 point pairs in the shape. For rotation invariance, the authors proposed using the tangent vector at each point as the positive x-axis instead of absolute axis for computing the associated shape context. This descriptor is invariant over scale, translation, rotation, occlusion, and noise.

2.2.2.4 Local Binary Patterns (LBP)

Local Binary Patterns descriptor was proposed in [56]. It is based on the basic idea that the local structure around each pixel (called the center pixel) is encoded by comparing it with its eight neighbors in a 3×3 neighborhood. By subtracting intensity of the center pixel with its eight neighbors, the resulting negative values are encoded with 0 and the others with 1; Finally, a binary number is obtained by concatenating all these binary codes in a clockwise direction starting from the top-left one and its corresponding decimal value is used for labeling. These derived binary numbers are referred to as Local Binary Patterns or LBP codes. An example of LBP is shown in figure 2.7.

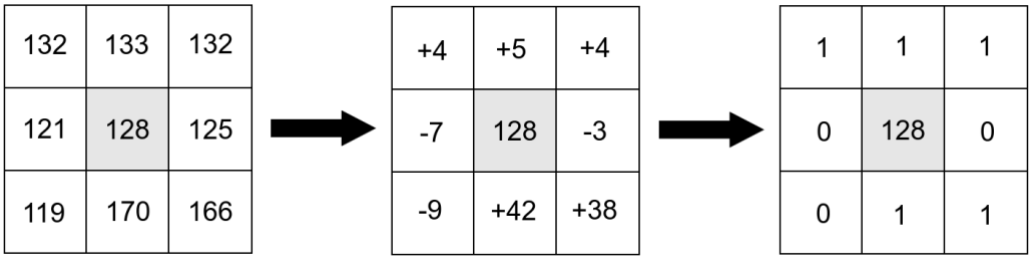

FIGURE 2.7: An example of LBP

LBP descriptor is robust against illumination changes and can be computed very quickly. The limitation of the basic LBP operator is that its small 3×3 neighborhood cannot capture dominant features with large scale structures. To deal with the texture at different scales, the author proposed some methods to generalize the analysis by using

the neighborhoods of different sizes which are defined as a set of sampling points evenly spaced on a circle which is centered at the pixel to be labeled, and the sampling points not falling within the pixels are interpolated using bilinear interpolation for any radius and any number of sampling points.

2.2.2.5 BRIEF (Binary Robust Independent Elementary Features)

The original BRIEF, which was proposed by Calonder [2], randomly selects n pairs of positions according to the Gaussian distribution and the center of an image patch are selected as the origin of the used coordinate system, as shown in figure 2.8.

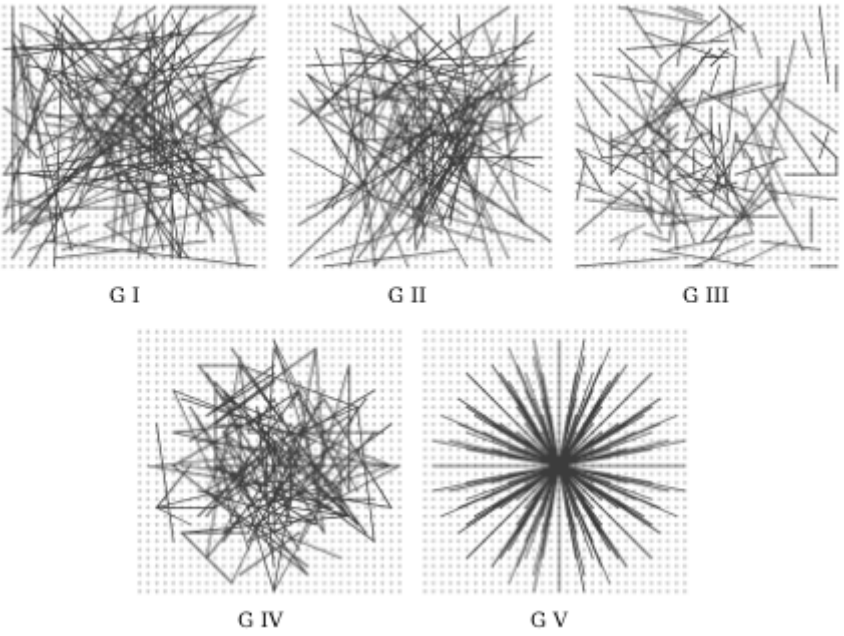

FIGURE 2.8: Different approaches to choosing the test locations of BRIEF by random sampling except the rightmost one, 128 tests in every image (extracted from [2]).

The descriptor is obtained by comparing the intensity of n pairs of pixels after applying a Gaussian smoothing which aims to reduce the noise sensitivity. The positions of the pixels are preselected randomly according to a Gaussian distribution around the center patch. The obtained descriptor is not invariant to scale and rotation changes.

BRIEF takes smoothened image patches and selects a set of $n_d(x, y)$ location pairs based on a unique pattern. Then some pixel intensity comparisons are done on these location pairs. For example, let first location pairs be p and q. If $I(p) < I(q)$, then its result is 1, else it is 0. This is applied for all the n_d location pairs to get a $n_d -$ *dimensional* bit string. Usually, n_d is set to be 256 as it comprises well between matching performance and efficiency.

One important point is that BRIEF doesn't provide a method to find the keypoint. The authors recommend using CenSurE (Center Surround Extrema) introduced in [57]. The main motivation behind the development of this detector is to achieve a full spatial resolution in a multiscale detector. BRIEF works even slightly better for CenSurE keypoints than for SURF keypoints.

2.2.2.6 BRISK (Binary Robust Invariant Scalable Keypoints)

BRISK uses the multi-scale AGAST as keypoint detector [48]. It searches for a maximum in scale space using the FAST score as a measure of saliency. BRISK uses sample points in concentric circles surrounding the feature which defines N locations equally spaced on concentric circles with the keypoint. When considering each sampling point p_i, a small patch around it is applied by using a Gaussian smoothing approach, the standard deviation of which (σ_i) is proportional to the distance between the points on the respective circle (shown in figure 2.9 (a)). In this pattern, distance sample point comparisons are classified into 2 subsets including short pairs $\mathcal{S}$ and long pairs $\mathcal{L}$. $\mathcal{S}$ contains pairs of sampling points for which their distance is below a threshold δ_{min} (see figure 2.9 (b)) and $\mathcal{L}$ contains pairs of sampling points for which their distance is above a threshold δ_{max} (see figure 2.9 (c)) . Long pairs are used to determine the orientation and short pairs are used for the intensity comparisons that build the BRISK descriptor. To compute the orientation of the keypoint k, BRISK uses local gradients between the sampling pairs which are defined by:

$$g(p_i, p_j) = (p_j - p_i).\frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{||p_j - p_i||^2}\,, \tag{2.16}$$

where (p_i, p_j) is one of the $N.(N-1)/2$ sampling-point pairs. The smoothed intensity values at these points which are $I(p_i, \sigma_i)$ and $I(p_j, \sigma_j)$ respectively.

The orientation is an average of all the local gradients between all the long pairs and then takes $arctan2(g_y/g_x)$ to determine the angle of the keypoint.

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{|\mathcal{L}|} \sum_{(p_i,p_j)\in\mathcal{L}} g(p_i, p_j)\,. \tag{2.17}$$

Beside scale-normalization, the angle $\alpha = arctan2(g_y, g_x)$ is then used to make BRISK descriptors rotation invariant by rotating the sampling pattern with α around the keypoint k before performing intensity comparisons. Finally, the BRISK descriptor assembles 512 bits which are resulting of the intensity comparisons on all the short pairs

in $\mathcal{S}$, where each bit b of the descriptor is computed as:

$$\forall (p_i^\alpha, p_j^\alpha) \in \mathcal{S}, \quad b = \begin{cases} 1, & if\ I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i) \\ 0, & otherwise\,. \end{cases} \tag{2.18}$$

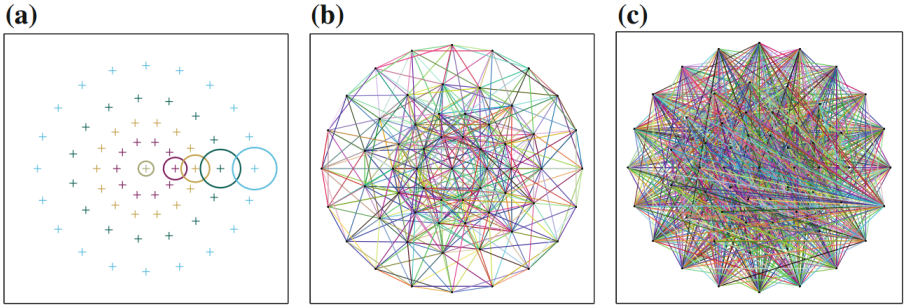

FIGURE 2.9: The 60 sampling pattern used in BRISK (a); the short pairs of sampling points used for constructing descriptor (b) and the long pairs of sampling points used for computing orientation (c) (extracted from [3]).

2.2.2.7 ORB

ORB, proposed in [47], was based on BRIEF descriptor and overcomes rotation invariance problems of BRIEF by computing an orientation component to FAST detector and adding it to BRIEF features. FAST is an ideal choice for finding key points that match visual features. Nevertheless, it does not produce a measure of the corner and lacks multi-scale features. To fill these gaps, ORB first employs a Harris corner measure to order FAST key points, then utilizes a scale pyramid of the image with each level producing certain FAST features. The orientation of FAST features is created by intensity centroid which is an offset between the intensity of a certain corner and its center. This offsetting makes the orientation, which is the vector between the feature location and the centroid [58]. From [58], the moments of a patch image I is defined as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y)\,. \tag{2.19}$$

The centroid is then defined from the moment as:

$$C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}})\,. \tag{2.20}$$

In oder to make C rotation invariant, ORB computes moments with only (x,y) in a circular region with radius r.

Let O be the corner's center, from the centroid C, the orientation $\overrightarrow{OC}$ of the patch can be simply determined by the angle θ, $\theta = atan2(m_{01}, m_{10})$. This keypoint orientation is consistent across views. Based on this, ORB applies machine learning approaches on training data automatically to reform BRIEF into rBRIEF [47]. These approaches supply the greedy search algorithm that selects 256 sampling pairs that have improvement in the variance and correlation. Finally, ORB descriptor is a result of the combination of oFAST keypoint detector and rBRIEF descriptor.

2.2.2.8 FREAK (Fast Retina Keypoint)

The approach adopted by FREAK is closely related to BRISK, differing only on the geometric pattern chosen to perform the binary tests and on the way those test pairs are selected [59]. FREAK uses a geometric pattern that mimics how the human retina works and its test pairs are selected through learning, similarly to the method used by ORB[47]. FREAK sampling pattern is a retinal sampling grid which is also circular with the difference of having a higher density of points near the center, as shown in figure 2.10 (a).

Similar to BRISK, FREAK sampling pattern uses different kernels size for every sample points. Yet, FREAK samples point more densely near the keypoint, which make the density of sampling points to drop exponentially as they are being far from the keypoint while the size of Gaussian kernels used to smooth intensities of the sampling points is increased exponentially with respect to the distance to the keypoint. Such a sampling pattern makes sampling points contain overlap information which is claimed to be more discriminative. In addition, FREAK uses the learning algorithm similarly to [47] to select point pairs from all possible pairs generated from the sampling points. Finally, 512 point pairs (as shown in figure 2.10 (b)) are selected for constructing FREAK descriptor.

Both BRISK and FREAK are based on the averaged local gradient computed from several point pairs in order to compute orientation. BRISK takes point pairs whose distances are larger than a certain threshold. While FREAK takes point pairs which are symmetric with respect to the center of the sampling pattern (see figure 2.10 (c)).

2.2.2.9 ALOHA

Using the power of integral image representation for images, ALOHA [60] is a feature which is fast to compute and efficient binary descriptor. It is based on a set of Haar-like

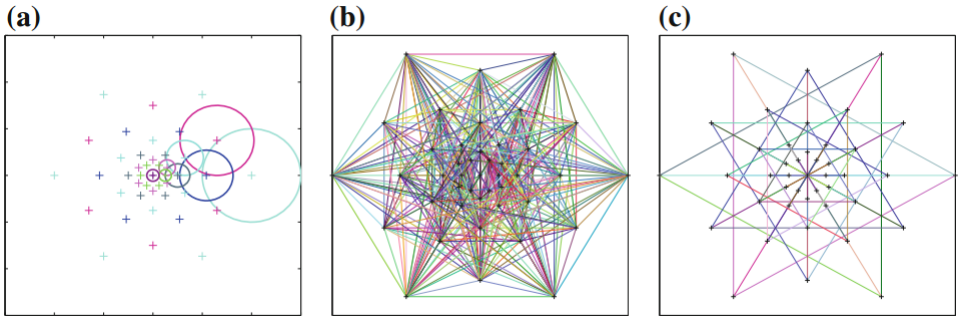

FIGURE 2.10: The 43 sampling pattern used in FREAK (a); the pairs of sampling points used for constructing descriptor (b) and the pairs of sampling points used for computing orientation (c) (extracted from [3]).

pixel patterns defined within an image patch and performs intensity difference tests to encode the image patch into a binary string. The authors define a test τ on patch P of size SxS as

$$\tau(P, X, Y) = \begin{cases} 1, & if \quad \overline{P_X} > \overline{P_Y} \\ 0, & otherwise\,, \end{cases} \tag{2.21}$$

where $\overline{P_X}$ and $\overline{P_Y}$ represent the mean intensities for two different pixel groups X and Y belonging to P. They define the set of 32 pixel patterns, each of which consists of two same size groups of pixel X and Y.

An original patch P of size S x S is divided into four equal subparts at 1-level, and then each of which is divided recursively. At the final step, original patch P and 4 patches at 1-level are tested with 32 patterns, and 16 patches at level 2-level are tested with 6 first patterns. Therefore, a $(1 + 4) \times 32 + 16 \times 6 = 256$-dimensional descriptor is defined. Disadvantages of ALOHA is that this descriptor is not scale and rotation invariant.

2.2.2.10 LDA-HASH

LDA-HASH [61] is based on the idea that how to projects the SIFT descriptors [50] into the Hamming space in order to encode them as short binary strings. Firstly, LDA-Hash extracts SIFT descriptors from the image. Then, they are projected to the more discriminant space using Linear Discriminant Analysis(LDA) or Difference of Covariances (DIF). Finally, the projected descriptors are thresholded in order to obtain binary descriptors.

2.2.2.11 BGP

BGP (Binary Gabor pattern) was proposed by Zhang et al. [62]. While LBP [56] encodes the local structure around each pixel by comparing it with its eight neighbors in a $3x3$ neighborhood, BGP encodes local image patch p with a radius R centered on the pixel by convolving J Gabor filters. Firstly J Gabor filters that share the same parameters except the parameter for orientation are computed at local image patch p with each orientation π/J incrementally . Then, the resulting negative values are encoded with 0 and the others with 1. Finally, these J bits are re-encoded by the approach called rotation invariant binary Gabor pattern, that performs a circular bitwise right shift on the J bits J times and takes the maximum one. BGP descriptor is rotation invariant and robust for texture classification.

2.2.2.12 D-BRIEF

D-BRIEF which was proposed in [63] is closely related to [61] LDAHash but faster than LDAHash in time computation thanks to learning stage of a set of discriminative orthogonal projections from patch intensities directly instead of using SIFT descriptors. From a real-valued vector made of the intensities of an image patch, this approach applies a set of projections (as shown in equation 2.22) and then thresholds the results in order to compute D-BRIEF descriptor.

$$\forall_{i\in 1,\dots,N} \quad b_i = sign(w_i^T x + t_i)\,, \tag{2.22}$$

where the b_i is the N bits of the descriptor, the w_i are the projections, the t_i is the thresholds, and x is the image patch in vector form. With only 32 bits per descriptor, D-Brief is robust in terms of accuracy and its efficiency is also due to the fact that it also significantly reduces the memory. Yet, like BRIEF, D-BRIEF is not adapted to scale and rotation changes. To deal with these constraints, in practice, the authors proposed extracting feature points using FAST [1] with 54 views (using 18 rotated views at 3 scales) and computed database of D-BRIEF descriptors for these feature points. The alternative approach is to estimate the scale and orientation of the feature points, and compute the descriptors on the rectified patches, which was used in ORB [47].

2.2.2.13 Binboost

Binboost descriptor [64] computes binary descriptor from an image patch p by the learning based approach named Adaboost. It constructs a binary vector $C(p) = [C_1(p), ..., C_D(p)]$

where,

$$C_D(p) = sgn(b_d^T h_d(p)), \tag{2.23}$$

$h_d(p)$ is K weak learners weighted by the vector $b_d^T = [b_{d,1}...b_{d,K}]^T$.

The weak learners consider the orientations of intensity gradients over image regions R and are parameterized by a rectangular region R over the image patch p, an orientation e, and by a threshold T which are defined as follows:

$$h(p, R, e, T) = \begin{cases} 1, & if \quad \phi_{R,e}(p) \quad \leq \quad T \\ -1, & otherwise \end{cases} \tag{2.24}$$

with

$$\phi_{R,e}(p) = \sum_{m \in R} \xi_e(p, m) \quad / \sum_{e' \in \phi, m \in R} \xi_{e'}(p, m)\,, \tag{2.25}$$

$$\xi_e(p, m) = max(0, cos(e - o(p, m))\,, \tag{2.26}$$

where $o(p, m)$ is the orientation of the image gradient in p at location m. From the number of quantization bins q, the orientation e is quantized to take values in $\phi = 0, 2\pi/q, 4\pi/q, ..., (q-1)2\pi/q$, which can be computed using integral images efficiently [64].

2.2.2.14 Geometrical descriptors

Geometrical features are based on geometrical constraints between keypoints which can be extracted from centroids of connected components such as letters or words. In this part, popular geometrical descriptors for textual documents retrieval are presented.

Locally Likely Arrangement Hashing (LLAH) LLAH is a feature descriptor developed by Nakai et al. [53, 65–67] in order to propose camera-based textual document retrieval and recognition. LLAH considers the centroid of each word connected component as a keypoint; this kind of keypoint can be generally obtained stably, even under perspective distortion, noise, and low resolution. A detailed description of the methods allowing to obtain centroid of each word connected component can be found in [53].

To compute LLAH descriptors from each keypoint P, the n nearest neighbor points around keypoint P are selected and organized following clockwise order. All possible combinations of m points among n are examined ($m < n$) and are considered as a good basis for determining stable local features. Depending on the way to compute features

from these arrangement combinations of m points, the LLAH descriptor is able to deal with various distortions that can appear in camera-based approaches.

The simplest version of LLAH is computed using three keypoints A, B, C (see Figure 2.11 for an example) and the formula in the equation (2.27). In this specific case, the system will be able to retrieve similar content being robust to image scaling, translation and rotation. These invariants are called similarity invariants [67].

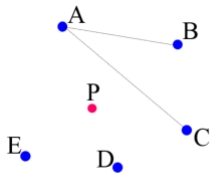

FIGURE 2.11: Three selected points A, B, C around one keypoint P

$$\frac{AC}{AB}\,. \tag{2.27}$$

When $k = 4$, from 4 points (e.g. points A, B, C, D from Figure 2.11), LLAH vector becomes robust to affine transformation with the affine invariant defined as follows [65]:

$$\frac{S(A,C,D)}{S(A,B,C)}\,, \tag{2.28}$$

where S(A,B,C) is the area of a triangle with apexes A, B, C.

Finally, the most advanced version of LLAH is able to be robust to perspective transformations which are very common when capturing a document with a camera. To do so, five keypoints are needed (points A, B, C, D, E from Figure 2.11 for instance), and a cross-ratio is computed as presented in (4.9)[65]:

$$\frac{S(A,B,C) * S(A,D,E)}{S(A,B,D) * S(A,C,E)}\,. \tag{2.29}$$

In order to reduce the sensibility of the system to keypoint extraction errors, multiple LLAH descriptors are computed for each keypoint. As all the possible combinations of m points among n are examined, C_n^m LLAH vectors have to be built from each keypoint. As a consequence, the more LLAH descriptors are built, the more processing time and memory consumption is required by the system. Thus, n and m need to be suitably set depending on each system.

The first testing of LLAH was evaluated on a dataset of 10000 scientific paper documents. Query images were captured covering entire pages with a 6.3 megapixels digital camera (CANON EOS 300D). The results were impressive in terms of accuracy, time and scalability [53]. In order to improve LLAH features, Takeda *et al.* [68] proposed an extension of the LLAH feature by adding some additional features which are based on the rank of k area ratios of the extracted word regions. In the work presented in [69], in order to consider the case of the capture of a portion of a document, they also proposed the method which can improve the LLAH features by adding additional features by ranking words regions based on their area.

Word shape coding Lu et al. [70] proposed word shape coding method for scanned Latin textual document retrieval system. This method converts each document into a word shape vector whose values are composed of word shape code and word frequency information of word images in the document. Firstly, the document is preprocessed in order to remove the noise and small connected components. Then, each word connected component is encoded with a word shape code and the number of intersections between character strokes within a word image and the middle line of text. The word shape code is formed from local extrema points which are classified into three categories as a function of their positions relatively to the x-line and baseline of text lines. The points which are far above the x-line and which correspond to a maximum are encoded with the category "3". The maximum or minimum points which are between the x-line and baseline are encoded with the category "2". The minimum points far below the baseline are encoded with the category "1".
This word shape coding method is used for document retrieval system in which the query is a scanned image of a document page. Yet this method is not able to deal with very small portions of documents captured by camera or with heterogeneous-content document images in which the base lines of text are various.

Layout Context A document image retrieval system with camera phones was proposed by Liu and Doermann in 2007 [28]. They proposed the features called "Layout Context" descriptor. The "Layout Context" features rely on the geometrical location of words' bounding boxes of a document image. Beginning at the center of a word and looking for the most visible n neighbors, the "Layout Context" of a word w is proposed. The visibility is defined by the angle of the view. According to the authors, the top n visible neighbors are rotation invariant and two view angles that a neighbor word occupies are also not subject to rotation. From the center of w, the coordinate system origin is established with X-axis parallel to the baseline of w and the width of w is used to define the unit metric. Under this coordinate, the coordinates of n most visible neighbors

are invariant to similarity transformation. The "Layout Context" descriptor is robust against perspective distortion, occlusion, uneven lighting and even crinkled pages. The experimental results showed that the system is able to identify even a small patch of the document image, captured by a camera phone, in a known set of documents. Drawbacks of this system are that it is quite slow to find every candidate page [28], and it is not able to deal with heterogeneous-content document images in which the base lines of text vary.

n-Word length Another document image retrieval system with camera phones was proposed by Hull *et al.* [25]; this relies on an augmented reality system. The authors built a local descriptor for text based on statistical analysis of n-word length next to a word in both vertical and horizontal direction. This descriptor is computed from each word position that is considered as keypoint. From this coordinate, the descriptor is established by counting the word length of n succeeding words from both directions. The word length is the number of connected components inside the word segment. This descriptor can distinguish an image of a patch of text from a collection of thousands of examples. Supplementary, this descriptor can be indexed efficiently by hashing methods. Nevertheless, it works well only with queries which are captured under portrait direction (which is similar to documents stored in database) and the retrieval accuracy dramatically drops down with documents containing noises that cause incorrect word lengths [25].

As presented above, various metrics and taxonomies are used to build local feature descriptors after having detected keypoints. Aiming to give a general framework to describe these various design approaches used for feature descriptor, we present a view of descriptor taxonomy categorization in Section 2.2.3.

2.2.3 Descriptor taxonomy classification

Keypoints descriptors generally describe a region around a point of interest (keypoint) and they have all been created in order to fit various approaches and goals. Generally, feature description attempts to find desirable ways to describe image content so that it is understandable for computer vision by learning to mimic how human visual system to recognize images. Thanks to the human brain, the human visual system can perform a number of image processing tasks much quicker than a computer vision system.

In our point of view, there are four main steps (see in Figure 2.12) to recognize the image in the brain. The first step is *retina* step from which electrical signals are relayed

to the brain via the optic nerve. The second step is called *brain's first step* during which human visual system discriminates and firstly responds to captured information, in a scale and rotationally invariant manner. It tends to look for features relationships among contrast variances along with psychophysical gestalt [71]. The third step is called *brain's second step*. During this stage, the perceptual system forms a percept from dependent of parts such as lines, circle, triangles and etc. The final step is called *brain's semantic step* from which the human brain recognizes the whole content of the image (e.g. what the picture is) based on semantic features and context of the image.

FIGURE 2.12: Descriptor classification inspired by human visual system.

In computer vision, many image enhancement methodologies have been proposed based on several models of the human visual system [72, 73]. To be inspired by this, we confer a point of view of descriptor taxonomy based on the human visual system. It is hoped that this can provide general information about descriptor taxonomies for those who employ or improve descriptors; or develop new ones.

2.2.3.1 Retina inspired descritors

Descriptors belonging to this group are close to how human retina discriminates and responds to image signals. They are normally based on frequency domains (e.g., Gabor and HAAR wavelets) or based on retina patterns (e.g., FREAK pattern). These descriptors are built from the way that the density of the receptor cells is greater in the center and decreases with distance from the center. Example descriptors in this group are FREAK, ALOHA, BGP and D-BRIEF.

2.2.3.2 Brain's first step inspired descriptors

Descriptors belonging to this group are similar to how the first step of the brain recognizes the images. They are generally based on gradients, intensity variances in the images. Example descriptors in this group are SIFT, SURF, LBP, BRIEF, BRISK, ORB, LDA-HASH and Binboost.

2.2.3.3 Brain's second step inspired descriptors

Descriptors belonging to this group are similar to how the second step of the brain recognizes the images. They are generally based on shapes and topologies that are established from information around a keypoint. Generally, spatial information of neighborhood keypoints (e.g. geometrics constrains) is used to compute invariant values for the descriptors e.g, Shape Context, LLAH.

2.2.3.4 Brain's semantic inspired descriptors

An analogy between image and text document in semantic granularity was introduced in [74], which shows that pixels are equated to letters, patches to words, patch pairs to phrases, objects to sentences. Semantic inspired descriptors are similar to a human-interpretable the semantic information containing in a document image. These descriptors represent semantics of the document, such as a set of keywords, or a text description. They represent the ultimate goal of annotation, indexing, high-level concept detection, or more generally automatic generation of semantic descriptors.

Most of the automatic indexing methods try to learn a correspondence model between local descriptors and semantic inspired descriptors. The system is able to produce some semantic inspired descriptors from a given set of training descriptors thanks to the learnt model, e.g. proposed methods in [75–81].

2.3 Indexing

A large-scale camera-based information spotting systems using local descriptors generally consists in extracting keypoints with their descriptors and to store them into a database (the block diagram illustrates a generic system in Fig. 2.1). The problem of such system is that it generally contains billion of descriptors, which corresponds to a large amount of memory and looking for the nearest neighbors can take a lot of time. In this case, linear searching is impossible to be applied in a reasonable time. Therefore, approximate nearest neighbor search approaches need to be developed. These approaches are also very notable in many other applications [82] such as information retrieval [83–85], pattern recognition[86, 87], image and video databases [88–90], databases and data mining [91], machine learning [92] etc. Approximate nearest neighbor search algorithms are known to provide large speedups with only minor loss in accuracy. They play an important role in the real-time required application e.g. camera based document image retrieval and spotting systems.

TABLE 2.2: Keypoint descriptors summation.

Feature Descriptor	Retina	Brain's first step	Brain's second step	Brain's semantic step	Robustness	Drawback
FREAK	x				fast, brightness, contrast, rotation without needed keypoint angle , scale,viewpoint,blur	large scale indexing
ALOHA	x				illumination	large scale indexing
BGP	x				rotationt	
D-BRIEF	x				low dimension	
SIFT		x			brightness, contrast, rotation, scale, affine transforms, noise, captures discriminative information via image gradient	real-time application
SURF		x			scale, rotation, illumination, noise , captures discriminative information via image gradient	
LBP		x			brightness, contrast	scale and rotation invariant
BRIEF		x			brightness, contrast	scale and rotation invariant
BRISK		x			brightness, contrast, rotation, scale	large scale indexing
ORB		x			fast,brightness, contrast, rotation, limited scale	large scale indexing
LDA-HASH		x			shortening SIFT	time processing, lost informations of SIFT
BinBoost		x			low dimension, high accuracy	
Shape Context			x		scale, rotation, occlusion, and noise	
LLAH			x		Fast,low dimesion	non-text elements, inadequate keypoints, many descriptors for one keypoint

The nearest neighbor search in a metric space can be defined as follows: given a set of n points $P = p_1, .., p_n$ in a metric space M and a query point $q \in M$, aiming to find the element $NN(q, P) \in P$ that is the closest to q based on a metric distance $d : M \times M \rightarrow \mathbb{R}$ such that: $NN(q, P) = \arg\min_x d(q, x) \quad \forall x \in P$. The distance function d satisfies four characteristics including $d(x, x) = 0$, non-negativity, symmetry and triangle inequality [93].

The nearest neighbor search methods need to find systematic indexing approaches which are fast, accurate and scalable sufficiently. For fast criterion indexing systems need to build a structure that permits to structure/organize the description such as tree data structure [94] or Locality Sensitive Hashing [95] or etc such that the operation $NN(q, P)$ can be carried out efficiently. For scalable criterion, indexing systems require very little memory, enabling their use on standard hardware or even on hand-held devices. For accurate criterion, indexing systems need to learn distance metric so that they are able to determine correct nearest neighbor of query descriptor vector.

In this section, we review popular nearest neighbor search approaches which are classified into two categories: tree based approaches and Hashing based approaches.

2.3.1 Tree based approaches

One of the well-known nearest neighbor algorithms is kd-tree algorithm, proposed in [94]. It is efficient in low dimensions. Yet, the performance degrades quickly in high dimensions. Kd-tree (short for k-dimensional tree) is a binary tree which is used for sorting and searching a set of n data points with k-dimension. In kd-tree, the root node contains entire n points and the other nodes represents a subpart of data points from their parent node based a partitioning at the parent node. This partitioning can be thought of as implicitly generating a splitting hyperplane that divides the space into two parts. Points on the left of this hyperplane are represented by the left subtree of that node and points on right of the hyperplane are represented by the right subtree. Because of recursive partition, the leaf node contains only small subset of the data points, which is used to determine nearest neighbor point of a query point. Two examples kd-tree are shown in figure 2.13.

In [94], partition data points of each non-leaf node is chosen in the following way: the dimension with highest variance is chosen and the median value of it is used for partitioning data points into two children nodes. This process is done recursively until the number of data points of a node is less than a minimum value and it is returned as leaf node. To search for approximate nearest neighbor, query data point q is put into the root and traversed down the tree until reaching the leaf node. Then, a linearly search for

nearest neighbor among all data points in leaf node is performed. The problem of kd-tree

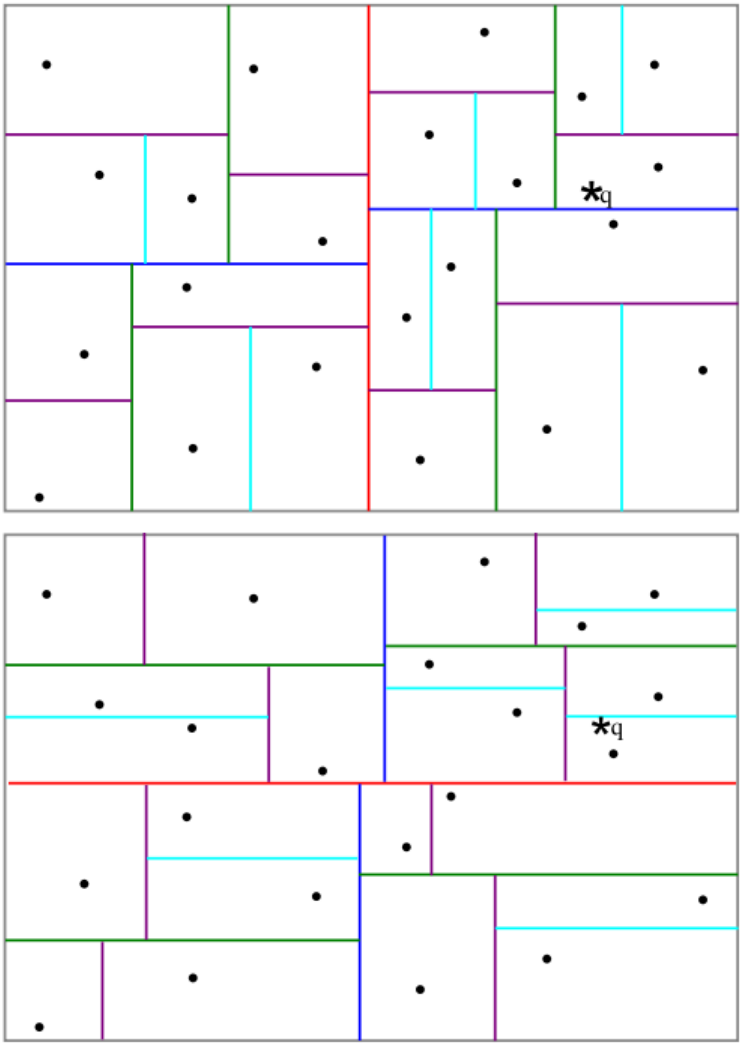

FIGURE 2.13: Example of randomized kd-trees in $\mathbb{R}^2$. In the first tree, the nearest neighbor of the query point does not lie in the same cell of the leaf node. Yet, it lies in the same cell of the leaf not, in the second tree (extracted from [4]).

is that the nearest neighbor may not be in the found leaf. To overcome this, a process of backtrack searching needs to be done iteratively so that other nodes are searched for better candidates. Arya et al. [96] propose the modified k-d tree for approximate matching. They also introduce a bounded on the accuracy searching using the notation called ε–*approximate* nearest neighbor which is defined as follows: a data point p in kd-tree is an ε–*approximate* nearest neighbor of a query point q if $d(p,q) \leq (1+\varepsilon)d(p^*,q)$ where p^* is the true nearest neighbor. The authors also propose priority search which uses a priority queue to speed up the search in a tree by visiting tree nodes in the priority queue following sequence of their distance from the query point. This distance is the minimum distance between q and any point on the node.

A similar kd-tree based algorithm, proposed by Beis and Lowe [97], use a stopping criterion based on examining a fixed number of leaf nodes instead of using ε–*approximate* cutoff. The authors use the Best-Bin-First (BBF) algorithm which is similar to priority search in [96] and employed successfully for multiple hierarchical clustering tree [4, 98] in which each cluster center is randomly chosen as one of the input data points instead of being the mean of the cluster elements (similar to [99, 100]). BBF starts to search in the tree by traversing from the root to the closest leaf. This follows at each inner node the branch with the closest cluster center to the query point and pushes all unexplored branches along the path to a priority queue. After the initial tree traversal, the algorithm resumes traversing the tree with the top branch in the queue. The priority queue is sorted in increasing distance from the query point to the boundary of the branch being added to the queue until the number of examined points excess than the maximum expected threshold.

Another solution is the use of multiple randomized kd-trees, proposed by Silpa-Anan and Hartley [101]. In this randomized k-d tree, the splitting dimension is chosen randomly among the top of highest variance dimensions and split value is randomly chosen using a point close to the median. The conjunction of these trees creates an overlapping partition of the feature space, which helps the trees to overcome incorrect nearest neighbor which may be affected by quantization (see figure 2.13). This robustness is especially important in high-dimensions where points will be more likely to lie close to a boundary due to the "curse of dimensionality".

Fukunaga and Narendra [102] propose a nearest-neighbor matching which is performed with a hierarchical tree structure constructed by clustering the data points into k disjoint groups and then recursively practicing the same for each of the groups. The authors also propose using the branch and bound tree-search method for searching nearest neighbor of a query point from the tree. Brin proposes a similar tree, called GNAT (Geometric Near-neighbor Access Tree) [99]. Instead of computing the cluster mean points as the cluster centers they use some of the data points (e.g. $p_1, p_2, ..., p_k$ from P) which are randomly chosen in fairly far apart manner. This tree is defined in a general metric space by integrating minimum distance and maximum distance for each pair of cluster center points $(p_i, p_j)i, j \in k$ which is defined to be min/max distance from cluster center points p_i to any point in the cluster of p_j. This information is used for pruning branches which do not contain the nearest neighbor in searching phase. The searching objective is to find all points with distance $\leq r$ to a given query point q. Yet this algorithm can lead to the case that it has to search too many nodes when pruning criteria do not occur.

Nister and Stewenius [103] propose the vocabulary tree, which is searched by accessing a single leaf of a hierarchical k-means tree. The most significant property of this approach is that the tree directly defines the quantization that is fully integrated with the indexing. This approach organizes local descriptors of database images in a tree using hierarchical k-means clustering. Then inverted files are stored at each node with scores in the off-line phase.

In the on-line phase, each descriptor vector of a given query image is propagated down the tree by at each level comparing the descriptor vector to the k candidate cluster centers (represented by k children in the tree) and choosing the closest one, which is performed recursively until reaching the leaf node. Then, generating a score for the given query image is based on Term Frequency-Inverse Document Frequency. Finally, the images in the database are found based on best match score. As a result, a vocabulary tree with k clusters of depth D has k^D leaf nodes, or visual words, at the bottom of the tree. The structure of the vocabulary tree (in the 128-dimensional SIFT space) is visualized as a nested set of Voronoi cells as shown in figure 2.14. In addition,

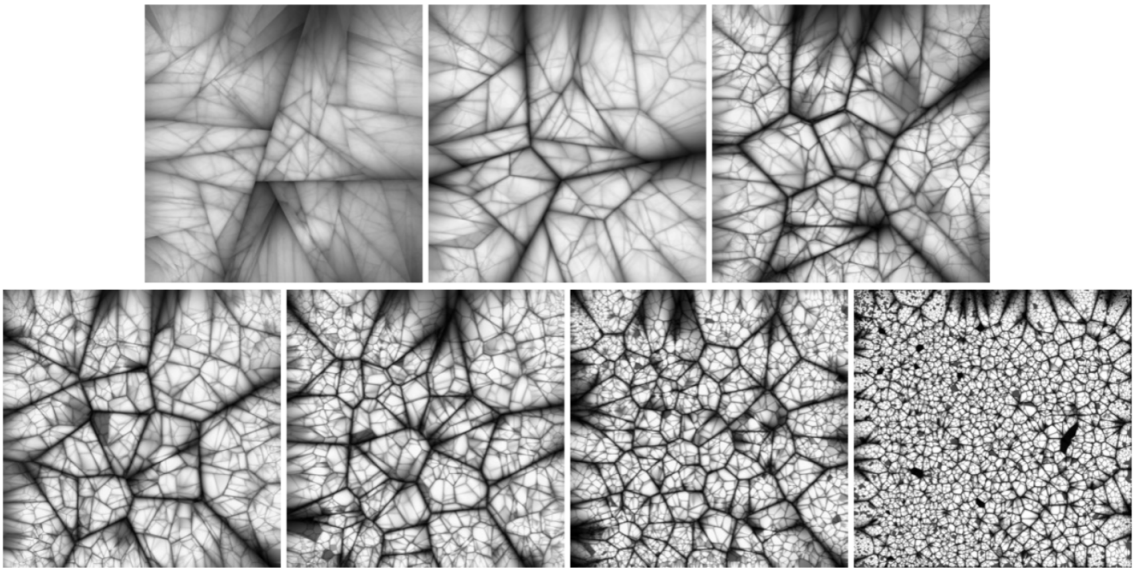

FIGURE 2.14: Vocabulary trees of varying branching factor and depth. Starting from top left, the sizes of tree are 2^{20}, 4^{10}, 10^6, 32^4, 100^3,1000^2(extrated from [5])

.

the authors propose using an inverted file for storing the scores efficiently. An inverted file index is associated to each node of the vocabulary tree (the inverted file of inner nodes is the concatenation of it's children's inverted files). The authors also propose the method to add an image to the database that requires the following steps: Firstly, image feature descriptors are computed; then, each descriptor vector is dropped down from the root of the tree and quantized into a path down the tree. The inverted files at each node store the id-numbers of the images in which a particular node occurs and the

term frequency of that image. These indexes are updated to the relevant inverted files [103].

In the next section , Section 2.3.2, we present other common indexing methods based on hashing techniques which have been widely studied and applied for approximate nearest neighbor search.

2.3.2 Hashing based approaches

Hashing is one of the popular indexing methods which is based on the idea of transforming the data item to a low-dimensional representation, or equivalently a short code consisting of a sequence of bits. The application of hashing to approximate nearest neighbor search includes two ways: indexing data items using hash tables that are formed by storing the items with the same code in a hash bucket; and approximating the distance using the one computed with short codes.

One of the well-known hashing based nearest neighbor approach is locality sensitive hashing (LSH) [95]. The main idea of LSH is to hash the points in the way that the probability of collision is much higher for objects which are close to each other than for those which are far apart. In retrieval phase, nearest neighbors of the query point can be found by hashing the query point and retrieving elements stored in buckets containing the query point. An example of LSH system is shown in figure 2.15

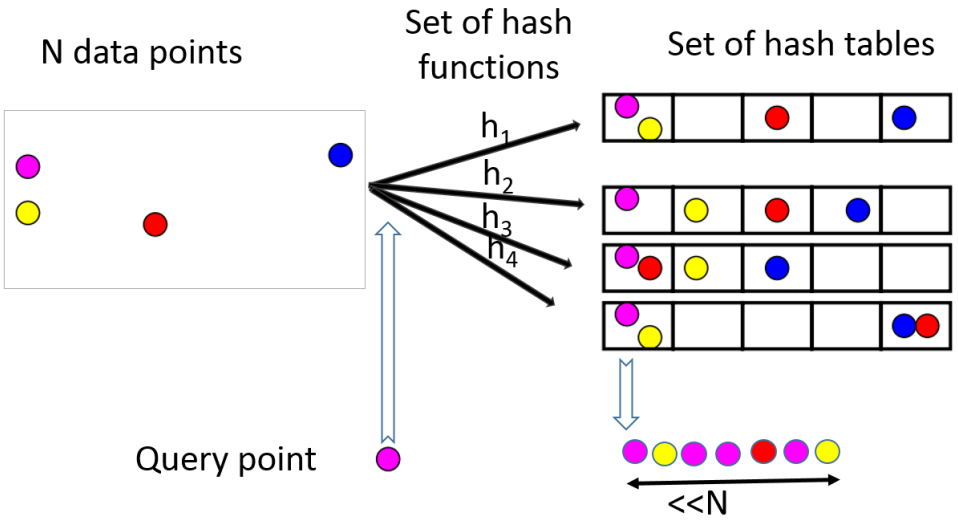

FIGURE 2.15: An example of LSH system for data in $\mathbb{R}^2$

There are variants of LSH. An improved version of LSH algorithm is proposed by Gionis et al. [104, 105]. This algorithm transforms each point $p \in P$ into a binary vector by embedding P into a Hamming space where distances between points in the

original space are preserved. Hash functions are created by selecting a subset of the bits that satisfy the desired locality-sensitive properties. The algorithm builds a set of l such hash functions, each of which selects k bits from the binary vector randomly. The two parameters (k and l) enable to select an appropriate trade-off between accuracy and running time. The bigger they are the more accurate the system can get and the more running time the system needs as a result. This LSH based algorithm requires using a large number of hash functions and a long with a set of hash tables. Each hash table uses a hash function randomly chosen from LSH family and contains the dataset points hashed by using its hash function. For a query point q, nearest neighbors of q are found by looping over all hash table and retrieve the points from the bucket in it with validation based on distance from q. Another LSH method is Multi-probe LSH, propose in [106]. It improves the high storage costs by reducing the number of hash tables. There is a LSH method that does not require tuning of parameters and can adapt better to the data is LSH Forest (introduce in[107]).

Kise et al. propose a simple representation method for approximate searching of local feature vectors [108]. The main ideas of this method is that each PCA-SIFT feature vector (proposed in [109]) is binarized into a simple bit representation and a hashing method that is able to be accessed very fast with less memory via a hash table. This indexing system can work without storing feature vector and re-checking the distance between database point and query point. In the retrieval phase, an approximate search with query perturbation is proposed in order to find its approximate nearest neighbors efficiently.

The binarized function for PCA-SIFT vector $v = (v_1, v_2, ..., v_n)$ is defined as follows:

$$b_i = \begin{cases} 1 & \text{if } v_i \geq 0 \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.30}$$

This produces a bit vector $b = (b_1, b_2, ..., b_n)$ where the first ($d \leq n$) elements are employed for indexing using hash function:

$$H_{index} = (\sum_{i=1}^{d} b_i 2^{(i-1)}) \bmod H_{size}\,. \tag{2.31}$$

The query perturbation is applied because the query vector is also transformed into the bit vector using the same threshold of zero for each dimension. Some dimensions of the query vector having values close to 0 are flipped before searching in the hash table.

Similarly, instead of using the threshold of zero, the later system proposed in [110] use θ_j as the threshold, where θ_j is the median of each dimension v_j of feature vectors in the database.

2.4 Conclusion

Camera-based information spotting systems need to deal with not only heterogeneous-content document images that may contain both textual and graphical elements but also real-time retrieval and spotting requests. To archive these two goals by using local features, we should choose robust descriptors as well as systematic indexing approaches.

Concerning local features, both keypoint detectors and keypoint descriptors play an important role and need to be well combined in such a way that it can detect and describe not only texts but also graphics in document images robustly and can overcome challenges of images captured by cameras in real-time. Because of these demands, camera-based information spotting systems should employ local features that are rotation and scale invariant and/or robust to affine and perspective transformations. In addition, to deal with potential problems of document images captured by cameras, descriptors should be robust to brightness, contrast, noise, illumination and blur.

Regarding indexing and retrieval, an ideal approximate nearest neighbor search algorithms need to be fast, accurate and sufficiently scalable. In some application, the quick adding/removing a data point into/from the indexing systems without reconstructing the index structures, or the indexing methods without storing database points which can help the systems to reduce a mount of memory is also concerned.

In tree based techniques, partition methods to build the trees and the traversal algorithms to search in the trees are very important. Both can effect the accuracy and the searching speed. Furthermore, using multiple randomized kd-trees helps in improving the accuracy search but it can take more time processing and need more memory space.

In hashing based approaches, the number of hash tables and the hash function play an important role. The hash function should be chosen so that the probability of collision is much higher for data points which are close to each other than for those which are far apart. This can be designed without or with exploring the data distribution and learning to hash. Because of this, when the dataset has a large amount of collision data points, the searching cost can be high because of the inner loop in the hash bucket for a long sequential list. On the other hand, the way to encode or transform data may effect the hash function and cause collision of the hash table.

Finally, some indexing structures which allow to re-check distance between query point and database point false matches can occur when searching. Indeed, indexing systems may return a good nearest neighbor which is not the best nearest neighbor. This decision can be done by defining a fixed distance thresholding or adaptive distance

thresholding or distance ratio thresholding or contrario matching criterion [111]. Thus, we should choose the suitable matching criterion depending on the objective of each system.

Chapter 3

Proposed features for Heterogeneous-Content Camera-based Document Image Retrieval and spotting system

3.1 Introduction

Local features are very tolerant to illumination changes, perspective distortions, image blur, image zoom, and so on [20]. Yet, when dealing with less textured documents, these local features are generally not discriminative enough to permit the calculation of relevant and stable descriptors [112]. In the context of huge document repositories, the high dimensionality of these descriptors arises two more constraints respectively regarding the curse of dimensionality on the one hand, and the computation time when dealing with real time matching systems on the other hand [41].

In this chapter, our proposed features based on spatial space of connected components are presented in section 3.2. Afterward, a framework to compute these features based on spatial space of keypoints is presented.

3.2 Features based on spatial space of connected components

In this section, we present several novel schemes towards features computation for heterogeneous-content camera-based document image retrieval. The first one is SRIF (Scale and Rotation Invariant Features), which is computed based on geometrical constraints between pairs of nearest points around a keypoint. In addition, we propose four extensions based on SRIF. The second one is PSRIF (Polygon-shape-based Scale and Rotation Invariant Features), which is an extension to SRIF and which makes SRIF more discriminative even though it is computed from a small number of constraint points around the keypoint. The third one is DETRIF (Delaunay triangulation-based features), which relies on the geometrical constraints from each pair of adjacency triangles in Delaunay triangulation which is constructed from centroids of connected components. The last one is SSKSRIF (Scale Rotation Feature descriptor based on Spatial Space of Keypoints), which also relies on the geometrical constraints from each pair of adjacency triangles in Delaunay triangulation using similarity transformation. In addition, we propose a framework to compute descriptors based on spatial space of dedicated keypoints such as SIFT, SURF and ORB. This aims at enhancing proposed features can deal with the context of heterogeneous-content camera-based document image retrieval and spotting.

3.2.1 Scale and Rotation Invariant Features (SRIF)

Firstly, SRIF extracts centroids of word connected components as keypoints (as shown in Figure 3.2). We can definitely employ centroids of letters as keypoints if needed. Then, SRIF feature vectors are extracted from each keypoint. It relies on the idea of using pairs of nearest constraint points around a keypoint (see Figure 3.1). Let P be a keypoint, P_i and P_j be two points coplanar with P (e.g. in 2D). $|\overrightarrow{PP_i}|$ and $|\overrightarrow{PP_j}|$ denote the length of the two vectors $\overrightarrow{PP_i}$ and $\overrightarrow{PP_j}$, respectively, and θ_{ij} is the angle between these two vectors. It is obvious that the three values θ_{ij}, $L_{max_{ij}} = max(\frac{|\overrightarrow{PP_i}|}{|\overrightarrow{PP_j}|}, \frac{|\overrightarrow{PP_j}|}{|\overrightarrow{PP_i}|})$, and $L_{min_{ij}} = \min(\frac{|\overrightarrow{PP_i}|}{|\overrightarrow{PP_j}|}, \frac{|\overrightarrow{PP_j}|}{|\overrightarrow{PP_i}|})$ are scale and rotation invariant [113].

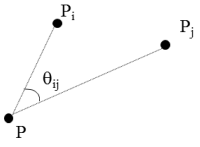

FIGURE 3.1: Constraint between two points around one keypoint P.

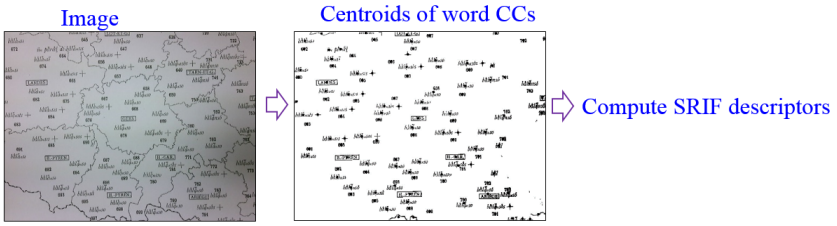

FIGURE 3.2: Centroids of word connected components as keypoints

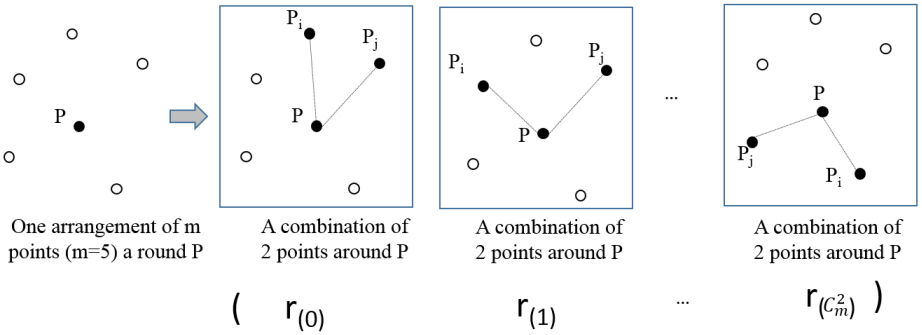

FIGURE 3.3: The arrangement of m points (m=5) and the sequence of new invariants (SRIF) calculated from all possible combinations of 2 points among m points.

Based on these scale and rotation invariant constraints between three points (as shown in Figure 3.3), we propose two scale and rotation invariant ratios used for SRIF:

$$\theta_{ij} * L_{max_{ij}}, \tag{3.1}$$

$$\theta_{ij} * L_{min_{ij}} \tag{3.2}$$

From each keypoint P, n nearest neighbor points around P are selected and organized clockwise (e.g. $n = 6$). The nearest neighbor points are determined by using the Euclidean distance and they are selected from the keypoint list. After this, all possible combination of m points among n are examined with $m < n$ (e.g. $m = 5$ in Figure. 3.3), which aims at dealing with keypoint extraction errors. This combination strategy leads to the result that there are C_n^m SRIF descriptors being computed at each keypoint position.

Then, from one arrangement combination of m points, the SRIF vector r is calculated based on a sequence of scale and rotation invariants computed from all possible combinations of 2 points (constrained to P) among m points. Finally, each value of the SRIF vector, r_i (feature i^{th}), is computed using invariant values: $\theta_{ij} * L_{max_{ij}}$ as presented in equations 3.1. Experimentally, using $L_{max_{ij}}$ is better than using $L_{min_{ij}}$ because the

values of $L_{min_{ij}}$ belongs to (0..1) which are too small and need to be suitably normalized. As a result, the dimension of SRIF is C_m^2.

Data: set of keypoints KpS of document image
Result: SRIF descriptors list DL

```
Dl ← null;
for each P ∈ KpS do
    P_nS ← n nearest neighbor points in KpS around P;
    sort P_nS in order clockwise around P;
    for each P_mS ∈ all possible combination of m points among P_nS do
        select the starting point in P_mS ;        /* in order to make SRIF
        rotation invariant */
        i ← 0;
        for each P_2S ∈ all possible combination of 2 points among P_mS do
            r_i ← SRIF ;        /* SRIF is computed using equation 3.1 */
            i++ ;
        end
        add r to DL;
    end
end
```

Algorithm 1: computation of SRIF descriptor.

As SRIF feature vector is computed from m nearest neighbor points which are organized following a clockwise order. To deal with rotation invariance, each of the m points need to be used as a starting point by examining all cyclic permutations in the retrieval phase. The analysis of these cyclic permutations is necessary because the feature vector of the retrieval algorithm may not match with the feature vector in the feature vector repository storage (indexing) algorithm, due to rotations of camera-captured images. This takes more retrieval time because of the fact that the lookup in the hash table is done m times.

To overcome this problem, similar to the work from [53], we apply the method that could select the same starting points in both the repository (indexing) and the query processes. This point is chosen by selecting the point from which the maximum invariant is obtained by combining it with clockwise succeeding points. In the case when there are two or more equivalent maximum values, succeeding clockwise invariant values of the starting point are used for comparison.

To make SRIF descriptors more distinctive as well as robuster, we propose four extensions for SRIF as follows:

Area of connected component (CC) ratio: This descriptor is built based on two basic properties established from 2 keypoints around P that are the product between angle and area ratio of two CCs, which are also distinctive and affine (not subject to image translation, scale, rotation, aspect ration and shear transformations), which is described in more detail as below:

Let S_i and S_j denote the area of two CCs corresponding to P_i and P_j respectively (as in Figure 3.1). Let us define $S_{max_{ij}} = max(S_i/S_j, S_j/S_i)$ and $S_{min_{ij}} = min(S_i/S_j, S_j/S_i)$ that are scale and rotation invariant. Consequently, SRIF descriptors using area of connected component (CC) ratio are computed from invariant feature relied on $\theta_{ij}.S_{max_{ij}}$ or $\theta_{ij}.S_{mix_{ij}}$.

Combined distance ratio and area of CC ratio: In order to make SRIF descriptors more distinctive, we propose to combine the ratio between areas of two CCs, distance ratio between two vectors and the angle that are computed from a pair of keypoints around P. This descriptor ensures that each matched value must be invariant in both distance ratio and area of CC ratio. For each of these constraints, two separate values are established in the SRIF descriptor from each pair of keypoints around P. One value uses $\theta_{ij}.L_{min_{ij}}$, and the other uses $\theta_{ij}.S_{min_{ij}}$. As a result, combined SRIF descriptors have double the dimensionality compared to SRIF without this combination.

A new normalized formulation for SRIF descriptors: The keypoint positions can be a little changed because CCs can be affected by camera's effects. For the purpose that it can tolerate errors in keypoint extraction, we propose a method to normalize SRIF descriptors as follows:

We apply the square root of ratios including distance ratio and connected components ratio for establishing new SRIF descriptors, e.g. $\theta_{ij} * sqrt(L_{max_{ij}})$ or $\theta_{ij} * sqrt(S_{max_{ij}})$. The growth of the square root function is very slow and what is more, the output of this function is always a positive value ensuring order relation, thus normalized SRIF descriptors are more precise with a displacement of keypoints that can be affected by camera's problems.

A new method for computing rotation invariant descriptors: Regarding computation of rotation invariant descriptors where the method requires choosing a starting point in clockwise direction ([53]), we propose a new method which does not require selection of a starting point but has the same starting points in both the storage and the retrieval processes. The proposed method is to use the order of n nearest neighbor points from the keypoint based on their distance from the keypoint P. This order is rotation invariant, so it is not necessary to re-organize (or re-order) them. When two points are equidistant from P, the area of CCs where they are extracted is compared.

3.2.2 Polygon-shape-based Scale and Rotation Invariant Features (PSRIF)

The main idea of computing geometrical descriptors (e.g SRIF and LLAH) from a keypoint is to find a local set of m nearest keypoints around a keypoint. From these points, the invariant geometrical features are considered to be taken into account in the descriptor. To deal with error keypoints, m combination from n nearest keypoints are considered, based on the approach already presented in Section 3.2.1. It is assumed that in the case for which there are $n - m$ error keypoints among n points, at least one correct local set of m nearest keypoints can be found for computing descriptors. When n is chosen, the bigger m is, the more discriminating feature vectors are. However, there are more erroneous keypoints, in this case, and vectors' dimension is higher. When the amount of text is small or when camera pen is used, instead of increasing the value of m, extension features should be used. It makes feature vectors more discriminating and makes the retrieval system work better in this case. In addition, retrieval time and scalability of the system are more efficient with less computational complexity.

In this section, we first present two methods that are used by LLAH as additional features and which are based on the areas of connected components. Afterwards, we present how our new extension features for SRIF are computed (PSRIF).

Areas of connected components based features of LLAH In order to increase the discriminative power of LLAH and to deal with small portions of a document captured by a camera, *Nakai et al.* [69] proposed an extension of the LLAH. It is based on magnitude relation among areas of connected components. They use ranks of connected components' areas as extension features, illustrated in Figure 3.4 on the left. This ranking is based on the fact that the largest connected component tends to have the largest area under a certain degree of change in condition and that magnitude relation among areas of connected components hardly changes when images are captured. According to the authors, these extension features are affine invariant. In [68], similarly, Takeda *et al.* proposed an extension of the LLAH by using features based on the rank of m area ratios of the extracted word regions (as shown in Figure 3.4 on the right). These extension features are perspective invariant [68].

The two main drawbacks of areas of connected components based extension features are that there are equal area connected components among them and that areas of connected components can be affected by uneven lighting which is quite common in case of camera capture. This can lead to incorrect ranks of connected components' areas when they are compared with their original images. We propose a new approach to add

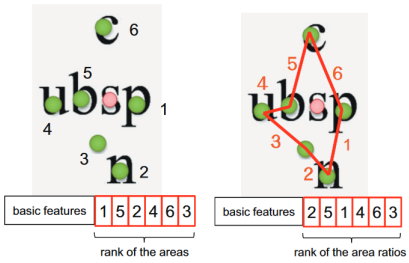

FIGURE 3.4: Extension features of LLAH [6].

extension features for SRIF and applicable to the LLAH as well.

Data: set of keypoints KpS of document image
Result: PSRIF descriptors list DL

$Dl \leftarrow null$;
for *each* $P \in KpS$ **do**
  $P_nS \leftarrow n$ nearest neighbor points around P;
  sort P_nS in order clockwise around P;
  for *each* $P_mS \in$ *all possible combination of* m *points among* P_nS **do**
    select *the starting point* in P_mS ; `/* in order to make the descriptor rotation invariant */`
    $i \leftarrow 0$;
    for *each* $P_2S \in$ *all possible combination of 2 points among* P_mS **do**
      $r_i \leftarrow SRIF$; `/* SRIF is computed using equation 3.1 */`
      $i++$;
    end
    add m extension features to vector r;
    add r to DL;
  end
end

Algorithm 2: PSRIF computation.

PSRIF uses the angles and the edges of the polygon which is formed from m keypoints around a keypoint (as shown in Figure 3.5). From the starting point and following clockwise order, let $\overrightarrow{e_i}$ and $\overrightarrow{e_{i\_next}}$ be two succeeding edges from the polygon. It is obvious that $(\overrightarrow{e_i}, \overrightarrow{e_{i\_next}})max(\frac{|\overrightarrow{e_i}|}{|\overrightarrow{e_{i\_next}}|}, \frac{|\overrightarrow{e_{i\_next}}|}{|\overrightarrow{e_i}|})$ is scale invariant and rotation invariant. From the polygon, there are such m invariant values that are used as m extension features for SRIF. As a result, PSRIF vector with extension features has C_m^2 basic SRIF features + m extension features.

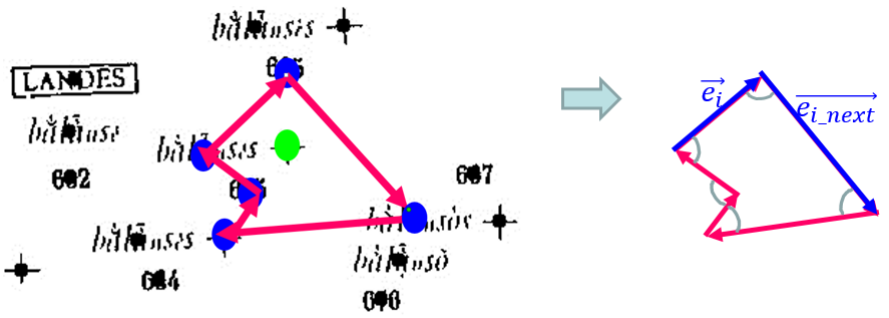

FIGURE 3.5: Extension features for SRIF based on the polygon formed from $m = 5$ keypoints around one keypoint.

3.2.3 Delaunay Triangulation-based Features (DETRIF)

When using SRIF, two vital parameters that establish combinations of nearest keypoints for computing descriptors need to be set (e.g. n and m). We aim to propose a new descriptor which can be employed without parameters controlling the selection of feature points. Our idea is to use a stable structure of the feature points and then to build descriptors from this structure so that it can cope with portions of a document captured by a camera. Because of this, we choose Delaunay triangulation to form the stable structure for the feature points and then DETRIF descriptors are built from this structure.

Delaunay triangulation has three main properties [114]:

- Given a set of points, there always exists a Delaunay triangulation except when all the points are aligned.
- The Delaunay triangulation maximizes the minimum angle of each triangle in the triangulation.
- When a subset of four or more points can be placed on the same circle (e.g. the vertices of a rectangle), the Delaunay triangulation of the points is not unique.

From these properties, we will always be able to compute a Delaunay triangulation from centroids of word CCs in documents. Because unstable cases (e.g. aligned points) will never occur in the whole page, they may only occur locally and create local instabilities.

To take advantage of Delaunay triangulation, we propose a way to combine the local Delaunay triangulation in order to build new descriptors named DETRIF. DETRIF is computed based on the geometrical constraints from the Delaunay triangulation which itself is constructed from centroids of connected components. As the Delaunay

triangulation is invariant to similarity transformations and not to perspective, invariant values of DETRIF are extracted from geometrical constraints on each pair of adjacent triangles in Delaunay triangulation. Therefore, this feature is tolerant to perspective distortion [115, 116].

DETRIF considers centroids of word connected components as feature points from which Delaunay triangulation is constructed. An example of DETRIF computation on a map is illustrated in Figure 3.6.

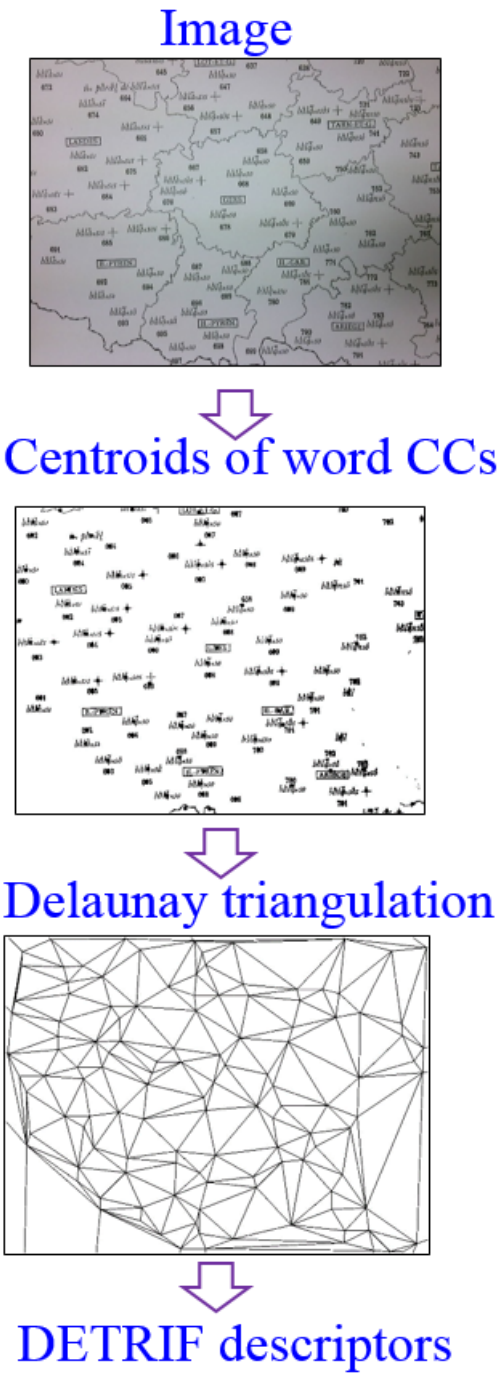

FIGURE 3.6: The main steps to build DETRIF descriptors.

The pseudo-code below describes how DETRIF descriptors are built (for a document image), from a Delaunay triangulation.

Data: Delaunay triangulation DTr computed from the set of keypoints of document image
Result: DTRIF descriptors list DL

```
DL ← null;
for each triangle tr ∈ DTr do
    for each vertex v ∈ tr do
        if exist adjacent triangle of tr then
            for each vertex X_i connect to edge v and not belong to tr neither to
            the adjacent triangle of tr do
                build DETRIF descriptor f at X_i ;
                add f to DL;
            end
        end
    end
end
```

Algorithm 3: Computation of DETRIF descriptor.

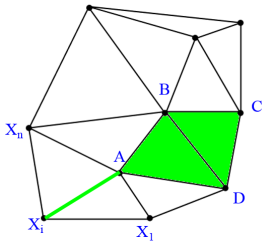

FIGURE 3.7: Adjacent triangles (ABD, BDC) and vertexes connected to vertex A $(X_0, X_1, ...X_n)$.

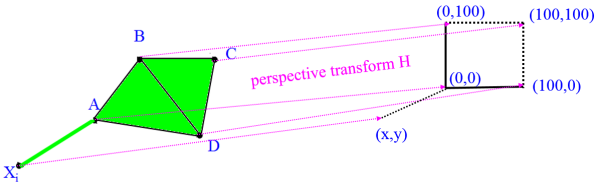

FIGURE 3.8: DETRIF descriptor extraction from each vertex X_i

In order to reduce the sensitivity of the system to keypoint extraction errors, one DETRIF descriptor is built for each vertex X_i. As we can see in figure 3.8, the constraint between X_i and two triangles $\triangle ABD$ and $\triangle BCD$ is used to build one DETRIF

descriptor. This process can be summarized in two steps:

Firstly, we find the perspective transformation H that transforms four points $ABCD$ on the left of Figure 3.8 into a normalized coordinate system on the right of Figure 3.8. Then, transformation H is applied on X_i in order to obtain the (x,y) coordinates in the new normalized space. Aiming to get positive features for indexing DETRIF descriptors with a hash table, the point (x,y) is transformed into the polar coordinate system. As a result, the point (x,y) becomes the point (r,φ) in the polar coordinate system, where $r \in \mathbb{R}$, and $\varphi \in (0..360^o)$. These two invariant values are used to build DETRIF descriptor. In order to make DETRIF descriptor more distinctive, we also use geometric constraints from Figure 3.8. Finally, DETRIF descriptors f are built by using the invariant values that includes r, φ, $\frac{S(B,C,D)}{S(A,B,D)}$, $\angle ABC$, $\angle BCD$, $\angle CDA$, $\angle DAB$, $\angle X_iAB$, $\angle X_iBC$, $\angle X_iCD$, $\angle X_iAD$.
Where, $\angle ABC$ is the angle between $\overrightarrow{AB}$ and $\overrightarrow{BC}$, so $\angle ABC \in (0..\pi)$

3.2.4 Scale Rotation Feature descriptor based on Spatial Space of Keypoints (SSKSRIF)

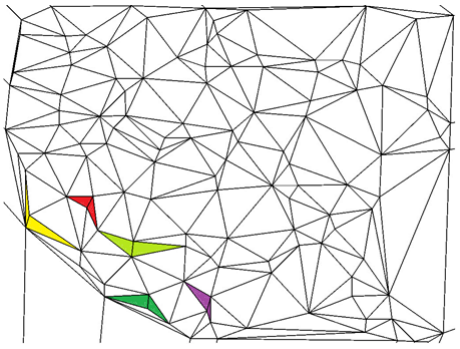

FIGURE 3.9: Example of concave quadrangles in a Delaunay triangulation.

DETRIF works well only when the quadrangles formed from two adjacency triangles are convex. However, if concave quadrangles occur, the perspective transformation of the quadrangle into the normalized space does not exist, which makes the accuracy of DETRIF reduced (as shown in Figure 3.9). Therefore, to tackle this problem, we propose to build a new scale rotation feature descriptors based on spatial space of keypoints by using the transformation shown in Figure 3.10. This transformation relies on the method from [117], in which the authors proposed a geometric hashing from a set of quadpoints. Starting from four points defining a unity square (a quadpoint), they defined a coordinate space from two of them and use the rest two points for the indexing and the retrieval process of astronomical images.

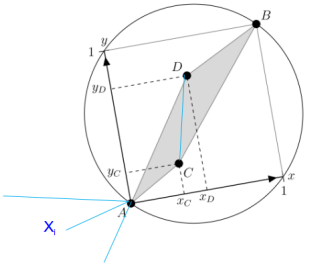

FIGURE 3.10: SSKSRIF descriptor extraction from each vertex X_i

Instead of using all combination of 4 points, we propose a method to build SSKSRIF descriptors base on quadrangle formed from two adjacency triangles in Delaunay triangulation graph of sampled keypoints. The computation of SSKSRIF descriptors is similar to DETRIF but the invariant features used for SSKSRIF descriptor f at X_i rely on a new transformation which is described in Figure 3.10. The descriptors f are built by using the invariant values r_{X_i},φ_{X_i},r_C,φ_C,r_D, and φ_D, where, r_* and φ_* are the polar coordinates of points X_i, C and D. To normalize features for hashing, each r_* in polar coordinate system is scaled-up to β times (e.g. β=15). This normalization aims to scale-up values which are less than or equal 1 such as r_C and r_D in order reduce collision when hashing.

In order to make SSKSRIF descriptors more distinctive we also use geometric constraints including $\frac{S(D,B,C)}{S(A,D,C)}$, $(\overrightarrow{e_i},\overrightarrow{e_{i\_next}})max(\frac{|\overrightarrow{e_i}|}{|\overrightarrow{e_{i\_next}}|},\frac{|\overrightarrow{e_{i\_next}}|}{|\overrightarrow{e_i}|})$, where $\overrightarrow{e_i}$ and $\overrightarrow{e_{i\_next}}$ are two succeeding edges from the polygon $ADBC$.

To make the descriptors rotation invariant, the starting point is set from A and the other points are determined from the point following A by clockwise order around the centroid of quadrangle ADBC (e.g. point D in Figure. 3.10).

3.3 Proposed framework to compute proposed features based on spatial space of keypoints

The descriptors based on spatial space of connected components work well with textual documents because they are computed from spatial organization connected components such as centroids of words or letters. However, they may fail when the number of connected components is insufficient or when dealing with graphical objects in which connected components are not stable when they are captured by cameras.

In order to enhance the proposed features (Section 3.2) in the context of heterogeneous-content camera-based document image retrieval and spotting, we propose to compute

our features with dedicated keypoint detectors such as corner-based or blob-based keypoints presented in Chapter 2. As presented before, our descriptors are only computed from spatial information of keypoints without capturing local textures or pixels in images. But, using existing well-known keypoints detectors allow proposed descriptors to be able to deal with various types of document contents like graphical contents, less texture contents, blur, lighting effects and etc.

Yet, when computing such descriptors, we need to tackle major issues including how to sampling stable keypoints from a set of keypoints extracted from an image for computing descriptors. It is not useful for geometrical descriptors if there are too many unstable or unrepeatable keypoints between the original image and the captured image. These unstable keypoints lead to the fact that there are too many incorrect created descriptors.

In this section, we propose a framework to compute our proposed features from spatial space of keypoints. Firstly, keypoints are extracted from an image using SIFT, SURF or ORB detector. Then, stable keypoints are sampled in order to compute descriptors from local connections between these keypoints.

3.3.1 The proposed method for sampling stable keypoints

Sampling stable keypoints for building geometrical descriptors is a really important step. This is due to the fact that the redundancy of sampled keypoints does not only make geometrical descriptors less robust but also it becomes a burden for the indexing and the retrieval processes. On the other hand, the lack of stable keypoints can lead to a lack of necessary descriptors being built and lead to some incorrect geometrical descriptors.

From a set of keypoints extracted from an image, we aim at sampling keypoints which are stable and repeatable under different image transformations such as similarity transform, affine transform and perspective transform. Generally, the important property of the stable keypoints is the response property that represents the quality of a keypoint and can be used for prioritizing and sampling. According to the research in [118], the minimum distance between two keypoints is also essential. When this distance is very near, geometrical features for such keypoints were not discriminative. Thus, it is necessary to set up a threshold for the minimum distance between keypoints in order to avoid sampling dense keypoints.

We propose to prioritize keypoints by selecting those with the strongest response on the one hand, and to use a threshold on the distance between keypoints in order to summarize a dense keypoints region by only the best ones using a selection process. Another good property of this selection is that sampled keypoints are sparsely distributed.

On the one hand, the average repeatability of keypoints depends on textures of image, image scale, detectors, capturing devices used and so on [119]. Instead of using a threshold to define the maximum number of keypoints to retain, as in [118], we suggest using a relative threshold (percentage) among the whole set of detected keypoints. This strategy helps in avoiding insufficient sampled stable keypoints at a region where the query image is captured. Especially it is also useful when database images are very large like posters or maps.

In addition, sampled keypoints have to adapt the minimum distance threshold between selected keypoints. To ensure the sampling is fair in database image as well as query image adapted the minimum distance threshold, we propose computing the distance from two keypoints in the unit coordinate. The transformation is done by scaling-down keypoints coordinate relying on the size of the image. This strategy can help the keypoint selection to be fairly when two images are different in the scale space.

After maximum the percentage threshold is set up, we can sample by local sampling or global sampling:

The global strategy sampling sorts all keypoints in the descending order of keypoint response firstly, and then one by one keypoint from the sorted keypoint list is sampled if the distance from the keypoint candidate to its nearest neighbor in selected list is greater than the minimum distance threshold otherwise it is discarded. This process will stop when the number of selected keypoints is over the maximum keypoint threshold.

The local strategy used for sampling the set of detected keypoints relies on the use of a quadtree approach where each leaf node contains the keypoints from a specific subregion in the image. With this strategy, we can easily define the criteria to use from which the partition will be stopped. This criteria could then be a minimum quantity of keypoints, and/or a minimum region area, and/or a minimum keypoint density. This process is done recursively by partitioning keypoints into region quadtree. When a leaf node is returned, we similarly apply our sampling strategy with a smaller set of keypoints that corresponds to those available in the sub-region of the image.

3.3.2 The algorithm for building descriptors

The general idea to compute the proposed spatial descriptors from any keypoints detector is described in Algorithm 4.

Data: document image I
Result: Descriptors list dL
1 $dL \leftarrow null$;
2 blur image I using a Gaussian filter;
3 $keypointList \leftarrow$ extracted keypoints from I;
4 separate $keypointList$ into separated scale levels $keypointList_s$;
5 **for** *each scale level* s **do**
6 $\quad keypointSampled_s \leftarrow$ sample stable keypoints from $keypointList_s$;
7 $\quad$ build descriptors ds from $keypointSampled_s$;
8 $\quad$ add ds to dL;
9 **end**

Algorithm 4: Computation of descriptor from scale-space of keypoints.

The main challenges, when building a geometrical descriptor based on keypoint is to select stable keypoints. stable keypoints have two properties: they are repeatable in two different scales from the pyramid scale-space and they are stable towards rotation transformations. The rotation challenge has been reported in [118, 120]. In order to make detected keypoints more stable, even if the image is captured by cameras, we need to reduce noise before the detection step. This is done in step 2 of Algorithm 4 using a Gaussian filter.

In addition, in the context of queries captured from documents with heterogeneous content, textures and resolutions of images may vary. Defining automatically a threshold that is able to be used in all possible conditions is then utopian. In this case, the descriptors could be built from a non-uniform pyramid scale in which the number of inlier keypoints between two scale levels is sufficient enough for computing geometrical descriptors (e.g. the method proposed in [118])

3.4 Conclusion

This section has presented the proposed descriptors (SRIF, PSRIF, DETRIF and SSKSRIF), which are promising for camera-based heterogeneous-content document image retrieval. These descriptors are built using some geometrical constraints between nearest keypoints

around a selected keypoint. This can then be considered as a local shape descriptor of keypoints extracted from document images.

The main advantage of DETRIF and SSKSRIF is that it can fix the way to combine local shape description without using parameter thanks to Delaunay triangulation from a set of keypoints in an image. As DETRIF and SSKSRIF are computed from quadrangles formed by two adjacency triangles, this can lead to other matched structures of keypoints around a keypoints to be ignored. By contrast, the combination of local shape description by parameters from SRIF also has its flexibility. The n nearest neighbors define how large region around a keypoints is examined, and each combination of m keypoint among n is used to build a descriptor. In the case $m = 4$, this strategy can not only describe the matched quadrangle in DETRIF, but also it can describe other matched structures that are formed from m points. However, when the value of n is large, one can notice that more and more combinations of m points can be examined, but the impact of unstable keypoints is more and more significant and then produce descriptors spoiling the indexing structure. Finally, PSRIF is an enhancement of SRIF with adding additional features in order to enhance the discriminating power of SRIF even when the number of combined keypoints becomes small.

Another advantage of the proposed descriptors is that they are not built at a pixel level. They can then be computed in a very quickly in the case that the number of keypoints is not too large. In addition, when the dimension of feature vectors is small, they can be indexed efficiently by simply using a hash table structure without needing to store them in the retrieval step. This indexing scheme can then deal with large scale of database documents and it allows adding new documents into the database without rebuilding the structure of the indexing system.

One main issue, that appears with all the possible camera-based information spotting techniques, is related to the ability to compute a relevant descriptor (from the indexing and retrieval point of view) when the number of connected components is becoming low in queries or when many keypoint extraction errors arise caused by camera's distortions (related to the resolution of the camera and the size of the text spotted). We then decided to compute our descriptor using centroids of connected components. In this specific case, results appear to be better only for documents composed of separated connected components. One can explain this specificity by the fact that based on geometrical constraints between nearest points around a keypoint, our method is impacted by the stability of the keypoints extraction method on the one hand, and by the fact that a minimal size of text has to be present in the document. When these two specificities are gathered, performances are very good and this happens to be a frequent case when dealing with daily life documents.

Finally, in order to deal with the previous limitation, and in order to highlight the genericity of the spatial representation of keypoints presented in our work, we proposed to adapt this framework to apply geometric descriptor on well-knowns keypoints detectors from the literature. This genericity rises some challenges when dealing with heterogeneous-content camera-based document image retrieval and spotting. Firstly, stable keypoint selection needs to be carried fairly from different scale level and sufficient enough. Secondly, the method used to search sets of m keypoints in an area among the n nearest neighbor keypoints is also important for the computation of the descriptor. This can be explained by the fact the bigger n is, the more correct set of m points can be found. However, more useless descriptors will be built from the extracted outliers, which can generate noise in the indexing and retrieval process. Lastly, to be able to deal with errors issuing from the keypoint detector step, we recommend using a not too large value of m in the descriptor computation step. This is why we proposed to add another kind of features (based on color, textures, ..) to enhance the discrimination power.

Chapter 4

Proposed indexing systems for Camera-based Document Image Retrieval and spotting systems

4.1 Introduction

The camera-based document image retrieval systems which can consider large-scale images need an efficient and accurate searching method. It can include billion of descriptors when employing local features. Thus, the nearest neighbor search is one of the burdens of computation and memory. In this case, linear searching is impossible to be applied in a reasonable time. Therefore, approximate nearest neighbor search approaches need to be developed. These approaches are also very notable in many other applications [82] such as information retrieval [83–85], pattern recognition[86, 87], image and video databases [88–90], databases and data mining [91], machine learning [92] etc.

The nearest neighbor search methods need to find systematic indexing approaches which are sufficiently fast, accurate and scalable. Considering the speed criterion, the fastest data structures actually used in the literature uses a structure that permits to organize the descriptors based on tree data structure [94] or Locality Sensitive Hashing [95]. Concerning scalability criterion, an indexing system should only require very little memory to enable their use on standard hardware or even on hand-held devices. Finally, accurate means that the indexing systems need to learn distance metrics so that they are able to determine correct nearest neighbors of a query descriptor vector.

To the best of our knowledge, no exact nearest neighbors algorithm is able to deal with those three issues nowadays, and that is why approximate nearest neighbors search algorithms were developed [4, 94, 98, 101, 103]. Approximate means they do not provide the whole set of results, which results in a large speedup, with only minor loss in the final accuracy. They have been widely discussed in the literature, and they play an important role in real-time computer vision applications *e.g.* camera-based document image retrieval and spotting systems. This chapter presents this category of techniques and proposes a new indexing process for approximate nearest neighbors search for computer vision purposes.

The motivation of this chapter is to build nearest neighbor search methods which are fast, accurate and scalable without storing database local descriptors. Our proposed indexing approach includes three methods. The first one is based on randomized clustering trees. The second one relies on a hashing indexing and retrieval approaches for the features proposed in Chapter 3 including SRIF, PSRIF, DETRIF and SSKSRIF. The third one is based on an idea which extends the hashing based method for indexing multi-kinds of features from multi-layer of images when text-graphic separation is needed and which permits to index and retrieve multi kinds of features from multi-layers representation.

4.2 Randomized hierarchical trees

Our proposed approach is inspired by hierarchical tree proposed by Muja et al. [4, 100]. In [100], the authors propose the method using k-mean tree and Best-Bin-First (BBF) searching method in order to index vector space descriptors. To construct the hierarchical trees, k-means clustering is used for splitting the data points at each level into K distinct regions. Recently, the other version called the priority search k-means tree for indexing binary descriptor is proposed [4]. In this tree [4], each cluster center is randomly chosen as one of the input data points instead of being based on the mean of the cluster elements and BBF is applied as well in the searching phase.

Based on the ideas presented in these two works, we propose a new approach using randomized hierarchical trees. To construct the hierarchical k-means tree, k-means clustering is used for splitting the data points at each level into 2 distinct groups. Instead of using the entire dimensions, only a small number of dimensions is chosen randomly and they are combined with the dimension with the highest variance which are computed along all dimensions and the maximum variance is selected. This combination is based on a mixture of randomized kd-tree, hierarchical k-means tree and random forest. Random forest is one of the most successful ensemble methods, proposed by Breiman [121]. It

was firstly proposed to solve the classification problem. Later, it was extended to handle regression and other applications. Recently, there have been a lot of applications which employ the random forest as the basic data structure and indexing [122–124] in computer vision fields.

The proposed randomized hierarchical trees perform a hierarchical decomposition of the descriptor space by clustering the input dataset successively. In these trees, every non-leaf node contains two cluster centers and the leaf nodes contain only one input point which is able to be matched to a query data point. At the outset step, the randomized hierarchical tree's root node contains all data points in the dataset. The randomized hierarchical tree is constructed by partitioning these data points at each level into two distinct group using k-Means++ [125]. We choose this clustering method because k-mean uses the random seeding which will inevitably merge clusters together, and the algorithm will never be able to split them apart, while k-means++ avoids this problem altogether via the careful seeding method. This partitioning of the trees is applied to the data points in each group recursively. The recursive decomposition is stopped when the group contains only one data point or all data points belong to the same cluster. The algorithm to build the randomized hierarchical tree is described in Algorithm 5.

Data: feature dataset D in $\mathbb{R}^n$

Result: hierarchical tree.

if $|D| = 1$ **then**
  create a leaf node with *pointId* and *documentId* of the data point in D;
else
  $d_{highestOfVariance} \longleftarrow$ seclect the highest variance dimension in D;
  $d_{randoms} \longleftarrow$ select K dimension from n dimension randomly such that $d_{randoms} \neq d_{highestOfVariance}$;
  cluster the data points in the node into 2 clusters C_{left} and C_{right} based on $d_{randoms}$ and $d_{highestOfVariance}$ using Kmean++;
  partition data points of the node into two child nodes based on distance to each cluster as follow: if data point belongs to C_{left} node it is partitioned into $leftchild$ node otherwise it is placed into $rightchild$ node;
  store C_{left} in the $leftchild$ node, C_{right} in the $rightchild$ node ; `/* along with encode of selected dimensions */`
  recursively apply the algorithm to the data points in each cluster C_{left} and C_{right};
end

Algorithm 5: Building the randomized hierarchical tree.

The clustering process relies on the highest variance dimension and K dimensions chosen randomly from n dimension ($K << n$). As a result, this process is carried out on the data points of nodes whose dimensions are reduced to K+1 dimension. The highest variance dimension makes data points in the node well-separating into two clusters while K sampling dimensions make the partition in each tree different. Aiming to keep the highest variance dimension not to be affected too much by K sampling dimensions, we propose to calculate the Euclidean distance between two data points in a weighted way for the highest variance dimension as follows.

We denote $Q = (q_h, q_{s1}, ..., q_{sK})$ and $P = (p_h, p_{s1}, ..., p_{sK})$ as two data points; h is highest variance dimension and $s1..sK$ are K sampling dimensions. The weighted Euclidean distance between two data points is defined as follows:

$$d(q,p) = \sqrt{\gamma(q_h - p_h)^2 + (q_{s1} - p_{s1})^2 ... (q_{sk} - p_{sk})^2}, \tag{4.1}$$

where $\gamma \geqq K$. This equation is also used to calculate distances from a data/query point to each cluster.

As can be seen from the Algorithm 5, the computation complexity of this process can be expressed by the recurrence relation (as shown in Equation 4.2), where N is the size of database descriptors and $C (C << N)$ is the complexity for step 4 to 8 the Algorithm 5(e.g number of iterations when clustering). As a result, the complexity of the Algorithm 5 is O(NlogN) from solving the Equation 4.2 [126].

$$T(N) = 2T(N/2) + cN \tag{4.2}$$

Searching for the approximate nearest neighbor of a query image can be seen as a navigating process throughout the tree by validating different properties. Using the query data point q, the idea is to put it at the root node of the tree and to traverse it down following the nearest distance between q and the two cluster centroids at each step of the tree. This process is done recursively until reaching a leaf node, i.e. the class

of the point. The algorithm is described in Algorithm 6.

Data: hierarchical tree T, query feature point q
Result: *pointId* and *documentId* of the nearest neighbor of q

Starting from the root node of T ;
ambiguousCheck ⟵ 0 ;
while *not reaching a leaf node* **do**
    d_{left} ⟵ distance from q to C_{left} ;
    d_{right} ⟵distance from q to C_{right} ;
    distanceRatio ⟵ min(d_{left},d_{right})/max(d_{left},d_{right});
    if *(distanceRatio>ratioThreshold)* **then**
        ambiguousCheck++;
    end
    if *(ambiguousCheck>threshold)* **then**
        return NULL ;
    end
    if *($d_{left} < d_{right}$)* **then**
        traverse the left child node;
    else
        the right child node;
    end
end
goodNearest ⟵ check distance of data point in the leaf node (p) with q;
if *(goodNearest< goodNearestThreshold)* **then**
    return *pointId* and *documentId* from the leaf node;
else
    return NULL;
end

Algorithm 6: Searching in the randomized hierarchical tree.

When traversing, we also employ a pruning rule based on a number of ambiguous searches. The idea is based on distance ratio matching method presented in [50]. In this method, matches are rejected if the ratio between the nearest neighbor distance and the second nearest neighbor distance is greater than a threshold (e.g 0.6 or 0.8), which helps to eliminate many false matches while discarding very little correct matches. In our tree, a query point with *ambiguousCheck* >threshold, the searching process will be eliminated with an empty result. This strategy can also help to speed up the searching time.

In order to reduce the memory which is used for storing descriptor database, the leaf nodes of the tree only store information of *pointId* and *documentId* of descriptor database. To ensure the nearest neighbor result is well-matched enough, data points are encoded in two float numbers using Euclidean norm and generalized mean (using equation 4.3 and equation 4.4) of a data point. To validate the best nearest neighbor at the leaf node based, two these encoded values are used. Thus, the indexing system only needs to store these two encoded values along with each database points instead of storing all dimensions of them being used in kd-trees or k-means tree. This strategy helps the system to save amount of memory with only minor loss in the final accuracy.

$$\|\mathbf{x}\| = \sqrt{x_1^2 + ... + x_n^2}, \tag{4.3}$$

$$M_p(x) = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{4.4}$$

Let q be query descriptor and p is database descriptor, *goodNearest* between p and q is defined as in equation 4.5 when p and q are near together and can be the nearest neighbor of each other.

$$goodNearest_{p,q} = abs(\|\mathbf{p}\| - \|\mathbf{q}\|) * abs(M_p(p) - M_p(q). \tag{4.5}$$

In addition, we build N randomized hierarchical trees for traversing q throughout these trees. If *pointId* and *documentId* returned results are the same from each tree, it is considered as being the best nearest neighbor otherwise the best nearest one is returned from the results of all the trees using equation 4.5 . Based on the experiments we led, we observe that N=2 is good enough. From the Algorithm 6, the computation complexity of searching process can be expressed by the recurrence relation (as shown in Equation 4.6), where N is the size of database descriptors. Therefore, the complexity of the Algorithm 6 is O(logN) from solving the Equation 4.6 [126].

$$T(N) = T(N/2) + 1 \tag{4.6}$$

4.3 Hashing based indexing and retrieval approaches for SRIF, PSRIF, DETRIF and SSKSRIF

In this section, we present the hashing-based indexing approach which is applied for SRIF, PSRIF, DETRIF features. This approach is similar to LLAH [127], thus it is carried out for LLAH testing as well. In this approach, feature vectors (called r) can be

indexed and retrieved very quickly using a hash table without need of storing feature vectors in the hash table [53], which make the system be efficient in terms of scalability. Furthermore, this indexing scheme allows adding new documents into the database without rebuilding all the database structure of indexes. Fig. 4.1 presents the hashing strategy.

4.3.1 Storing in the hash table

For each document image in the database, one descriptor type among proposed descriptors (SRIF, PSRIF, DETRIF and SSKSRIF) is extracted. One of the main issues when dealing with large numbers of feature vectors is the lack of what makes features of the descriptor more distinctive when both the fractional part and the integer part have been normalized. Similar to [66], our indexing system relies on the use of integer feature vectors r, that are discretized and normalized by Equation 4.7.

$$r_i = trunc(r_i) * 2 + round(r_i - trunc(r_i)), \tag{4.7}$$

In order to apply the indexing and retrieval with a hash table, a hash function is defined as follows [66]:

$$H_{index} = (\sum_{i=0}^{d-1} r_i q^i) \bmod H_{size}, \tag{4.8}$$

where d is the number of dimensions of vector r, q is the level of quantization constant (e.g. $q = 17$), H_{size} is the size of hash table.

The use of this technique is an incremental process. If a user wants to add a document into the database, the system firstly extracts keypoints. Then for each keypoint, descriptors are computed and indexed.

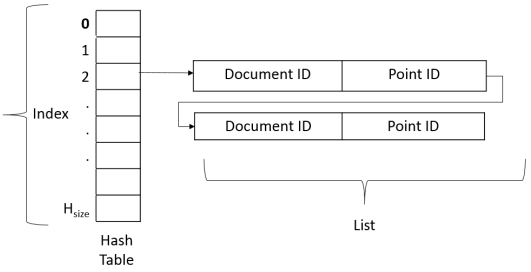

FIGURE 4.1: The hash table structure.

4.3.2 Retrieval

Starting from a query image captured with a camera, keypoints are firstly extracted (like in the indexing phase). Then for each keypoint, all descriptor vectors are computed and looked up in the indexing system, the hash table (using the hash function, equation 4.8), in order to get the list of document IDs related to each keypoint (Figures 4.1). In the next step, for each document in this retrieval result list, the number of votes in the voting table corresponding to each list element is incremented.

In addition, to discard confusion votes in a document, only the first vote is kept for each query's point and for each document's point. This makes sure that each query's point only matches with one document's points and vice versa. After getting the voting result, the top-t documents with the largest number of votes are selected as candidate results.

4.4 Extended hashing based method for indexing multi-kinds of features from multi-layer of images

As presented in chapter 2 and chapter 3 about local features in camera-based document image spotting and retrieval systems, some of them work well with graphical documents while the others work well with textual documents. In practice, documents may contain heterogeneous-information such as texts, symbols, logos, pictures, signatures, etc.

In this section, we present a system which can be applied for heterogeneous-content documents using text-graphic separation and combination of various features. The indexing and retrieval task is carried out with a homogeneous hash table, which is a combination of two hashing based methods from [127] and [110].

4.4.1 Text-graphic separation

Text/graphics separation is a process which consists in segmenting a document image into two layers, one containing text and the other containing graphics. Many different approaches have been proposed in order to tackle this problem.

Dhar and Chanda [128] introduced a method for the extraction and the recognition of symbol features from topographic maps. The method commences by separating a map into different color layers and then it recognizes the features in each layer on the basis of symbol-specific geometrical and morphological attributes.

Furthermore, connected component (CC) analysis has been used for this separation. For instance, Karl Tombre [129] proposed a size-histogram analysis from the bounding boxes of all CCs. By a correct threshold selection, obtained dynamically from the histogram, large graphical components are discarded, and smaller graphics and text components are kept.

Hohn [130] also used density of CC, more specifically a ratio between the area of the convex hull and the number of pixels in CC. To remove the dashed and dotted lines, the CCs are filtered by their density if their density is lower than a threshold. Furthermore, they used a diameter ratio that corresponds to the ratio between the minimum diameter value and the maximum diameter value among all the connected components. They also used a combined threshold region for the density and the ratio of maximum and minimum diameter, extended by an analysis of neighboring components to recognize text with large variation in style, size and orientations.

4.4.2 The architecture of the extended hashing based system

The proposed system includes 3 main steps: text/graphics separation, feature extraction and indexing/retrieval. An example architecture is shown in Figure 4.2.

Text/graphic separation for our dataset:

Our dataset, called CartoDialect dataset, includes French linguistic maps, and is composed of 400 images with a resolution of 9800 x 11768 pixels. Each map contains the phonetic symbols which describe the pronunciation of a word in different regions of France. All maps contain the same graphical elements which are region borders. Moreover, text density in each map is very sparse.

In both the indexing phase and the retrieval phase, the input document image is separated into 2 layers. For complex map data, the linguistic maps of France, attributes of CC are used for separating the image into 2 layers. Layer 1 contains CCs related to texts, and layer 2 contains CCs related to graphics. It can be seen from Fig 4.2.

In order to extract word connected components(CCs), the image is converted into binary image firstly, using the Otsu's method [131] or the adaptive thresholding [132]. Aiming to makes CCs belonging to a word touch together, then, the binary image is blurred using the Gaussian filter whose parameters are determined based on an estimated character size (the square root of a mode value of areas of CCs). Then, an adaptive-threshold is applied on the blurred image in order to determine boundary contours of word CCs . Finally, all word CCs are extracted. Consequently, dashed and dotted lines

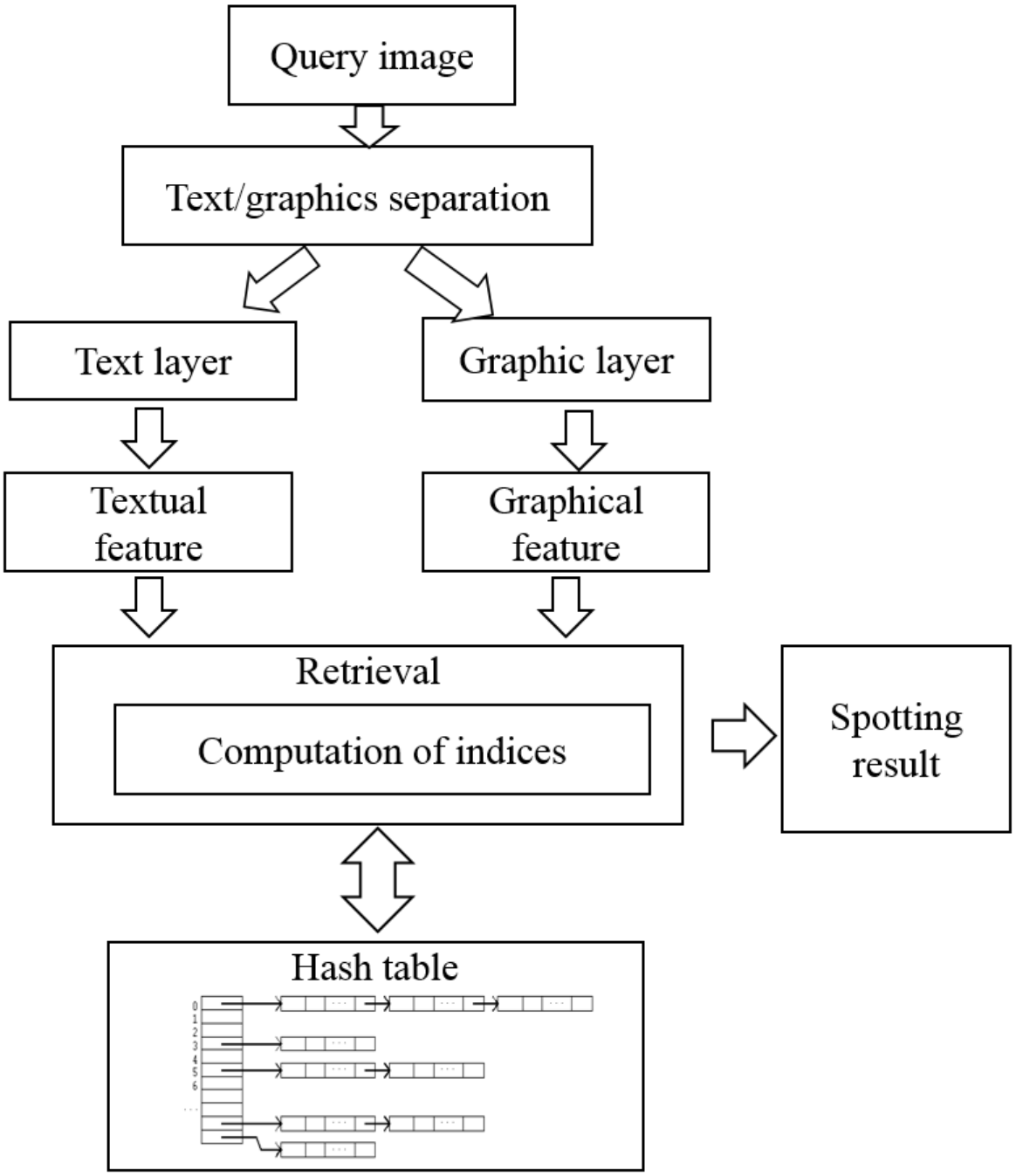

FIGURE 4.2: The architecture of a hashing based system for indexing multi-kinds of features from multi-layer of images.

are also joined to large CCs or long CCs which is a base for separating.

For classifying CCs, we use the attributes of CC, such as the area, the bounding box area and the maximum diameter so that large CCs and long CCs can be extracted into the graphics layer (see Fig 4.5 for an example). Those whose attributes are bigger than thresholds are considered to be large CCs. The thresholds are determined by choosing mean value multiplied by 2 (for each attribute) because almost CCs with attributes

FIGURE 4.3: An example query image.

belong to $[0...2 * mean]$ are text and noise. Thus, very small CCs which are noise need to be discarded. Lastly, the rest of CCs are extracted into text layer (Fig 4.4 illustrates the text layer extraction for an example query image).

FIGURE 4.4: An example of text layer separation result.

Feature extraction:

From the text layer, textual feature vectors are extracted using one of the descriptors proposed in Chapter 3. These features are computed from spatial organization of connected components. From graphics layer, we extract feature vectors which can deal with graphical elements by capturing pixels level from image patches e.g. SIFT [50] or SURF [51].

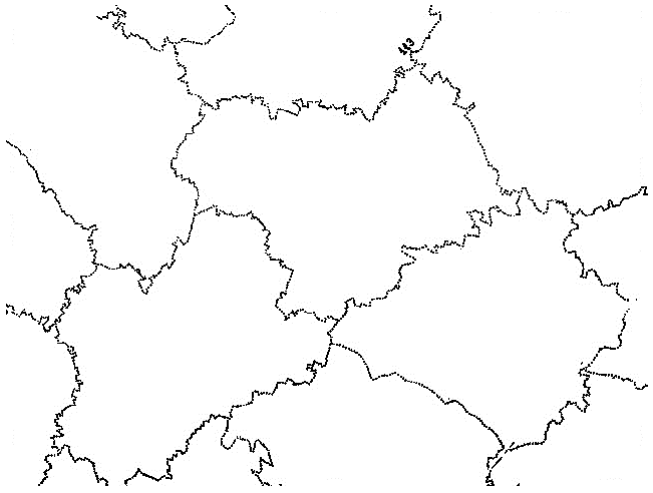

FIGURE 4.5: An example of graphic layer separation result.

4.4.3 Indexing phase

The indexing phase relies on the computation of a dedicated hash function which can hash each features vector into each integer value for determining the storing bucket position in the hash table. Firstly, all maps in the database need to be separated into text layer and graphic layer and then each layer is extracted with corresponding features in order to be indexed in a hash table via a hash function. This process is carried out off-line.

Aiming at reducing the required amount of memory, feature vectors are not stored in the hash table, only point IDs where key points are extracted need to be stored. Because one hash table is used for indexing both textual descriptors as well graphic descriptors, feature type needs to be stored along with document ID and point ID where the feature vector is extracted. The structure of the hash table is shown in Fig 4.6.

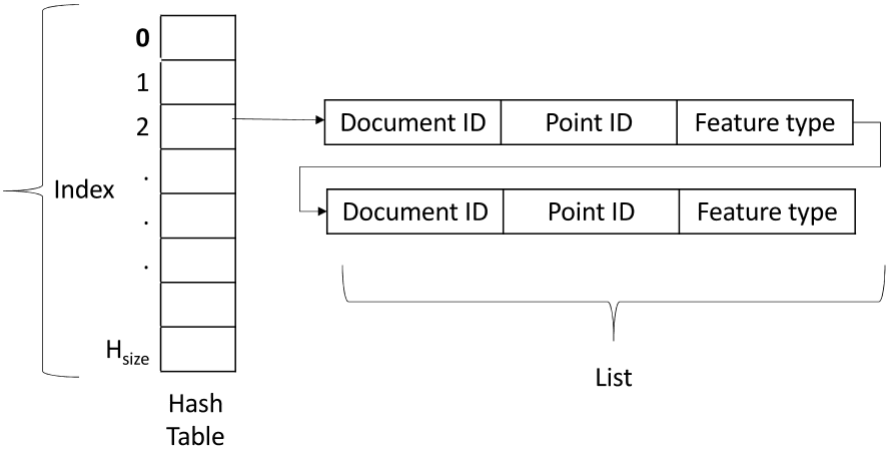

FIGURE 4.6: Structure of hash table.

From text layers, textual feature vectors are extracted and indexed using a hash function (Eq 4.10). From graphics layers, graphical features are extracted using and trained PCA space firstly. Then they are projected into PCA space to obtain K dimension vectors u (e.g. $K = 36$), which does not only help to reduce dimension but also to be hashed by the hash function straightforwardly by applying binarization.

Binary vector r is defined by using equation (4.9). It is noted that this transformation is only applied for float-type descriptors e.g. SIFT or SURF.

$$r_i = \begin{cases} 1 & \text{if } u_i \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

The hash function (Eq. 4.10) is also applied in order to index r. The index H_{index} of the hash table is calculated by the following hash function (Eq. 4.10):

$$H_{index} = (\sum_{i=0}^{n-1} r_i.k^i) \bmod H_{size}, \tag{4.10}$$

where n is the number of dimensions of r, k is the level of quantization, and H_{size} is the size of hash table. In the case that r is a binary vector, k is set equal to 2, and the hash function is the way to convert r into decimal number.

4.4.4 Retrieval phase

The retrieval phase is outlined in Fig 4.2, in which the query image also needs to be separated into two layers: text layer and graphic layer firstly. And then, from each layer is extracted corresponding textual features and/or graphical features on which the hash function (Eq. 4.10) is then applied to search for nearest neighbors stored in the hash table.

In addition, for the graphical layer, each feature vector's dimension needs to be reduced by projecting into PCA space and converted into binary vector similar to the indexing phase. Indeed, this binarization stage is required in order to ensure that two nearest feature vectors in metric space will get near distance in hamming space. However, two nearest feature vectors may be a little different from binary code and hash index as well.

To solve this problem, each converted binary vector needs to be expanded into several binary vectors in order to find its nearest neighbors as follows [110]. Let $e \gtrsim 0$ be a tolerant threshold, the idea is to change the way to binary feature dimensions that are less then e by bit flipping. From reduced dimension feature vector u, if $|u_i| \leq e$,

where $e \gtrsim 0$ is a tolerant threshold, then we use not only the bit vector with r_i but also with $1 - r_i$. Firstly, no bit flip is searched to obtain the result. Next, one-bit flip is applied, and its value is increased up to b at the final step. This strategy is applied to a limit b dimensions which satisfy the tolerance e. If the number of dimensions that satisfy the tolerance e exceeds the limit b, the strategy stops.

However, this query binary vector expansion strategy can lead to confusion voting problem because one query vector descriptor has many matched vectors in descriptor database when retrieving it. To tackle this issue, we propose a method that can select only one good nearest neighbor of the query vector from a list of potential nearest neighbor vectors. In our method, data points also need to be encoded in two float numbers using Euclidean norm and generalized mean (using equation 4.3 and equation 4.4) of a data point. The distance between two vector descriptors q and p is measured by using equation 4.5. From the list of potential nearest neighbor vectors of the query descriptor, if there is only one element in the list we select this element, otherwise ratio between two nearest neighbors of each query descriptor is considered. The best nearest neighbor is considered to be correct and taken into account for the voting process if the ratio between the nearest and the second nearest descriptor is equal or less than the threshold (0.64).

For the voting process, when searching in the hash table, each feature type only votes for $PointID$ and $DocumentID$ that belongs the same feature type of query descriptor. Finally, the document with a majority of votes is returned as the result.

4.5 Conclusion

We have presented three proposed indexing approaches that can be applied for camera-based document image retrieval, spotting systems indexing and local descriptors retrieval. All the methods can be employed without storing local descriptors in the memory for saving memory and speeding up retrieval time by discarding distance validating.

Randomized clustering trees are built by recursively partitioning data point at each node using hierarchical clustering that selects K dimension randomly (K <<total of dimension n e.g $K <= \sqrt{n}$) combining with the highest variance dimension. This strategy helps the system to reduce clustering processing time as well as the memory for storing the tree structures compared with k-mean tree which uses entire of dimension. Similar to random forest [121], the strategy sampling K dimension randomly applied at each node of the tree can deal with high dimension data without reducing dimension. However, each tree is split recursively until the leaf nodes contain only one data point,

which lead to the fact that depth of the proposed tree may be larger than the k-mean tree's depth.

By using a simple hash table and robust hash function, the hashing index system for SRIF, PSRIF and DETRIF is fast, accurate and scalable. Furthermore, it allows adding new documents into the database without rebuilding all the database structure of indexes.

Finally, multi-features indexing system is a composite between the proposed hashing index and the hash-based approach for floating value descriptors. Float-type descriptors need to be transformed into binary vectors by using PCA and median threshold value for each dimension. In the retrieval phase, each query descriptor needs to be expanded into several approximate binary vectors in order to search its nearest neighbors. This solution can lead to the result of many approximate nearest neighbors. Furthermore, transforming feature vector into binary bits after reducing dimension leads to the fact that many data points have the same binary vector although they are not nearest together. Consequently, the system can get many incorrect matching results in the retrieval phase but it can get full good matching points.

Chapter 5

Datasets and experimental results

5.1 Introduction

In this chapter, we present the datasets and the ground-truth that we have generated and which were used for the evaluation of our approaches. The proposed datasets were designed to evaluate the accuracy of the camera-based information spotting systems presented in this thesis at two different levels: the feature descriptors part and the indexing part. To validate our proposed approaches as well as to compare them with the state of the art techniques, we evaluated both retrieval and spotting accuracies on the one hand, and the retrieval time needed on the other hand. To the best of our knowledge, we can not find some publicly available datasets for camera-based information spotting in heterogeneous-content document images in the literature, and we thus decided to create a new dataset which has been made freely and publicly available for the scientific community. This dataset is comprised of three subparts, each of them being dedicated to a specific kind of information: The wikibook dataset represents the images with textual content only; The cartodialect dataset[1] represents images with graphical content mainly; The tobacco dataset contains text plus graphical content [133, 134].

5.2 Dataset and Ground-truth Generation

This section provides a detailed insight on the three datasets, their ground-truth and the methodoly used for capturing the videos.

[1] http://navidomass.univ-lr.fr/SRIFDataset/

5.2.1 Datasets

For the **WikiBook dataset,** we chose a book named LaTex from wikibooks[2] including 700 A4-sized pages, which were converted into JPEG format at 300 dpi resolution.

The **CartoDialect dataset** includes French linguistic maps, and is composed 400 images with a resolution of 9800 x 11768 pixels. Each map contains phonetic symbols which describe the pronunciation of a word in different regions of France. All maps contain the same graphical elements which are region borders, while the text density varies widely from map to map. The idea is then to evaluate the proposed methods on documents containing mainly graphical parts.

Finally, the **Tobacco dataset** includes 1291 documents containing heterogeneous-content such as text, handwriting, logos, tables and signatures.

The table 5.1 gives an overview of the dataset characteristics that have been generated.

TABLE 5.1: Dataset details

Dataset name	# of documents	Resolution	# of videos	# of tested frames
WikiBook	700	2480 x 3508	1630	24450
CartoDialect	400	9800 x 11768	2400	36000
Tobacco	1291	1696 x 2689	3191	47865

5.2.2 Ground truth generation

Camera-based document image retrieval and spotting systems consist in providing the focused portion of images from the learning database that match the best to the query. In these systems, the query generally corresponds to a subpart of a document which is captured by a camera. As mentioned earlier, to the best of our knowledge, no public dataset following those requirements is publicly available. We then decided to create one with its ground truth. Ground truth is a really important point as it allows automatic verification of the liability of a system and its accuracy.

Concerning our datasets, the ground truth has been built as follows. The document in WikiBook and Tobacco datasets have been divided into four regions (top left, top right, bottom left and bottom right) as presented in figure 5.1. As documents from the CartoDialect dataset are of a larger size of 9800 x 11768 pixels, they were divided into 6 regions (top left, top right, middle left, middle right, bottom left and bottom right)

[2]http://upload.wikimedia.org/wikipedia/commons/2/2d/LaTeX.pdf

(see figure 5.2). These regions have been used for validating the correct spotting in the retrieval phase (to ensure that the system being evaluated is not only retrieving the good document, but is also looking at the good location.

Starting from those partitions, one video was recorded at each region (excepted for blank regions). Documents were captured without rotations. The IPEVO VZ-1 HD document camera [3] was used for recording the videos and it was fixed at about 15cm above the surface of the captured document with a resolution of 1024x768.

For each video in each dataset, we selected the first 15 frames of the video. To validate the robustness of the systems towards rotation issues, we also rotated each frame by an angle of 0, 30, 60, 90, and 180 degrees. The number of captured videos is shown in Table 5.1. As many pages in the WikiBook dataset have large blank areas or with not enough text, the number of recorded videos in this dataset does not correspond to the number of frames multiplied by 4. These datasets and their ground truths are publicly available for academic research purposes[4].

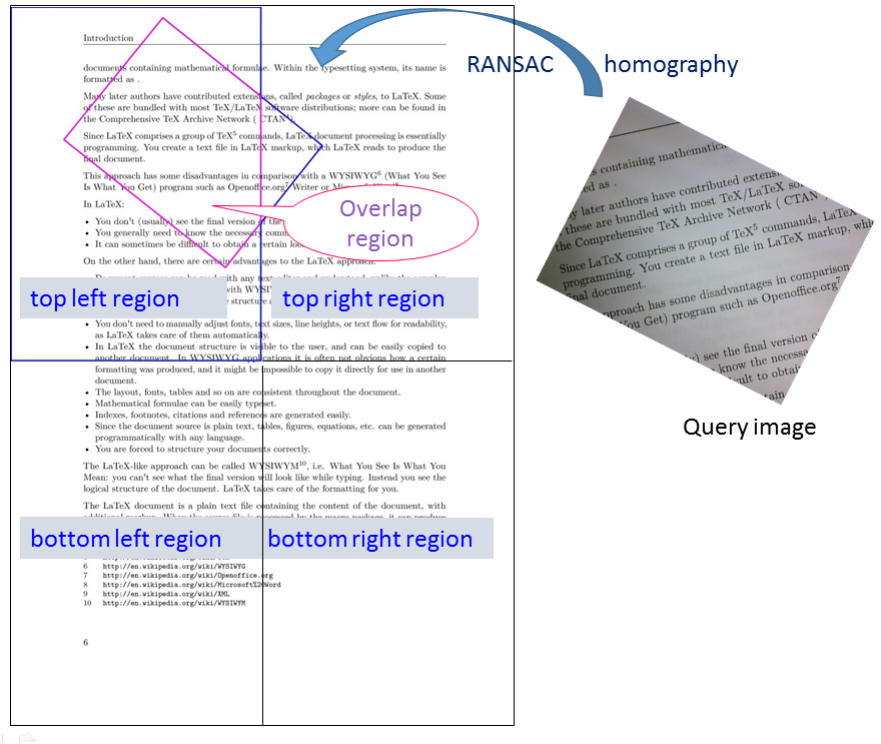

FIGURE 5.1: Captured video from a document at four regions, the overlap between spotting region result and captured region from a query image in WikiBook dataset.

[3] http://www.ipevo.com/prods/ipevo-vz-1-hd-vga-usb-document-camera
[4] The dataset can be downloaded at http://navidomass.univ-lr.fr/SRIFDataset/

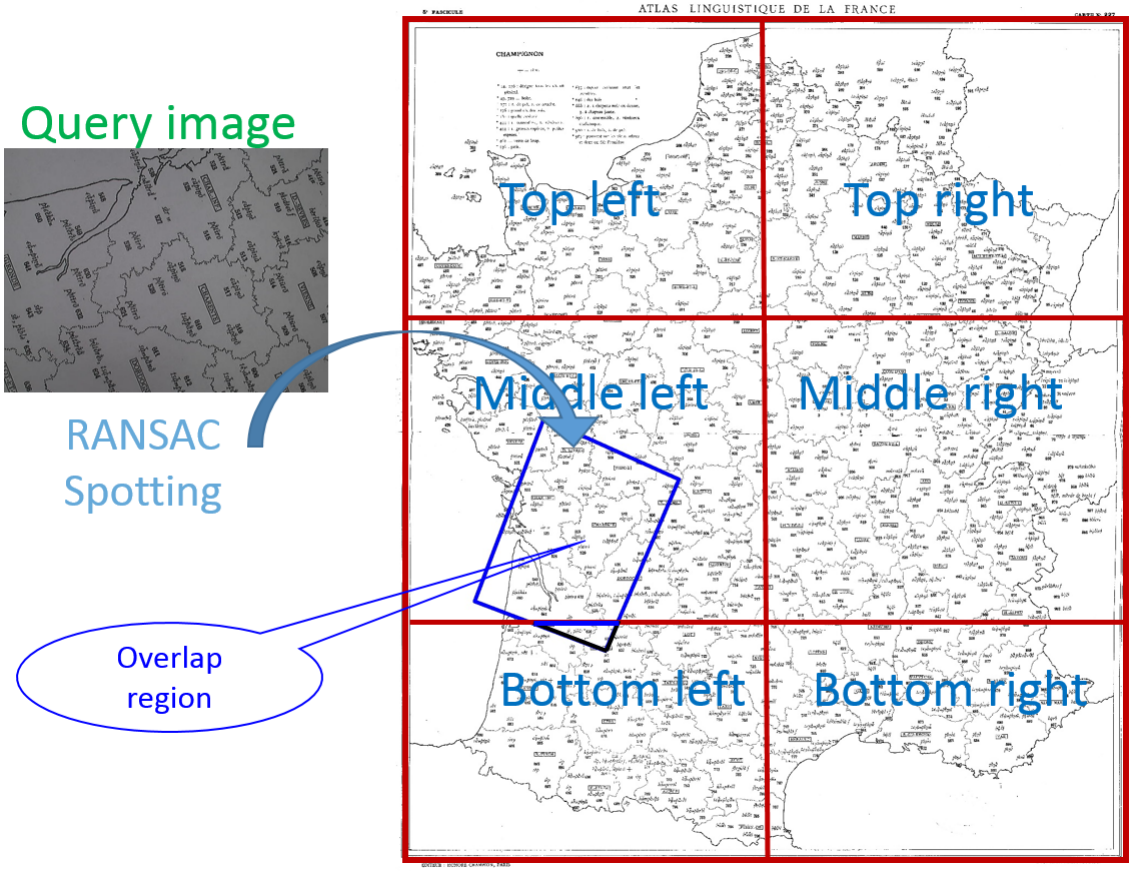

FIGURE 5.2: Captured video from a document at six regions, the overlap between spotting region result and captured region from a query image in CartoDialect dataset.

5.3 Experimental and Evaluation Protocol

Based on the dataset and ground-truth we designed, we decided to set an experimental and evaluation protocol in order to make our results as objective as possible. In the framework of a camera-based information spotting technologies, we decided to measure two main criteria including the retrieval accuracy and the time needed to get the answer. All methods were tested at the video level which means that we evaluated the retrieval accuracy for each video and not for each frame of the videos. We then decided to extract the first 15 frames, and to rotate them before the retrieval phase for testing the rotation invariance of systems. If the number of correct retrieved frames was greater than 50%, the video was considered as successful. Otherwise, it was considered as failed. The choice of this threshold was defined in order to ensure that at least the majority of the video frames was right. Finally, the videos retrieval accuracy is computed using the ratio between the number of correctly retrieved videos and the total number of videos for each dataset.

Even if the global evaluation was made at a video level, it relies on the fact to be able to evaluate if a frame was correctly retrieved as well as correct spotting. This leads to saying that it must be made clear whether or not there is a correct perspective transformation between query keypoints and each document keypoints.

To validate the correctness of a region, we firstly applied RANSAC [135] so that we can obtain the spotted region of query image in the document that has been identified in the previous step, through perspective transformation. Next, the overlap ratio between the ground-truthed region (where query image was captured) and the spotted region is computed. The frame is considered as a correct retrieved result if the overlapping between two regions is higher than 60% of the area of the spotted region. Otherwise, it is considered as an incorrect result. An example of the overlap region validation is shown in Fig. 5.1.

5.4 Experimental results of proposed descriptors compared with LLAH descriptors

In order to evaluate proposed descriptors and compare them to LLAH, we have tested 3 SRIF-based methods, DETRIF, SSKSRIF, and 3 LLAH-based methods as shown in Table 5.2. All of them were tested on spatial organization of connected components and spatial space of SURF keypoints.

In addition, the hashing based indexing and retrieval approaches (presented in 4.3) was employed. The parameters were set to $H_{size} = 10^{17}$, $t = 5$ for selecting top t of best candidate retrieval results, $\alpha = 60$ for validating the overlap spotting result. We implemented our method and LLAH on a 64 GB RAM Linux machine running in C extended C++ environment with a single thread.

TABLE 5.2: Tested Methods

Method	Description
SRIF	SRIF using distance based scale & rotation ratios as invariant
SRIF-CC	SRIF using ratio of CC's areas as invariant
SRIF-Combined	SRIF Combined distance ratio and area of CC ratio
DETRIF	Delaunay Triangulation-based Features
SSKSRIF	Scale Rotation Feature descriptor based on Spatial Space of Keypoints
LLAH-Affine	LLAH using affine invariant
LLAH-Cross-Ratio	LLAH using cross ratio invariant
LLAH-Similarity	LLAH using similarity invariant

For SRIF, we applied a maximum ratio selection from the distance ratio and the CC's area ratio. Besides, we also applied the square root of these ratios e.g.

$\theta_{ij}.sqrt(L_{max_{ij}})$ or $\theta_{ij}.sqrt(S_{max_{ij}})$ in order to make SRIF descriptors more tolerant to keypoint extraction's errors.

5.4.1 Computation on spatial organization of connected components

For these experiments, all the tested methods shared the same keypoint extraction approach. For Wikibook and Tobacco dataset, we extracted centroids of character connected components as keypoints. Because the texts in CartoDialect dataset are phonetic symbols, letter connected components are not clearly separated. So, we extracted centroids of word connected components as keypoints in this dataset. In addition, proposed text/graphics separation was used before extracting connected components (CC). This method uses attributes of connected components for filtering out big CCs as graphics elements which are discarded.

All SRIF-based methods and LLAH-based methods were tested with and without adding additional features by ranking CC based on their area [69]. The two ways of order of keypoints are also tested. The first one uses a clockwise ordering in the browsing of keypoints around a selected point with choosing the starting poin, and the second one uses the nearest neighbor points order without choosing the starting point. Furthermore, they were tested with additional features by using Polygon-shape-based Scale and Rotation Invariant Features (as proposed in 3.2.2).

5.4.1.1 WikiBook dataset's experimental result

The Table 5.3 shows that the retrieval accuracy of SRIF and LLAH-Similarity were the highest in both approaches (applying clockwise order and NN order) with approximately 94% of accuracy. The second highest retrieval accuracy was SRIF-Combined with a value of 86.5%, while the retrieval accuracy of LLAH-Cross-Ratio was the lowest one. The best mean retrieval time was obtained with the SRIF-Combined with 0.17 second/query.

As shown in Table 5.4, when applying PSRIF as extension features, SRIF-based methods and LLAH-based methods got better results in terms of videos accuracy retrieval. Besides, this extension could speed-up retrieval time of some methods including LLAH-Affine, LLAH-Cross-Ratio.

Table 5.3: Experimental results on WikiBook dataset based on spatial space of word connected components

Method	Order	n	m	Add	Videos Retrieval Accuracy						s/q
					0^o	30^o	60^o	90^o	180^o	Avg	
LLAH-Affine	NN	8	6	y	40.0%	40.3%	37.6%	39.3%	39.0%	39.2%	0.60
LLAH-Cross-Ratio	NN	9	7	y	5.9%	4.9%	4.1%	5.8%	6.3%	5.4%	1.43
LLAH-Similarity	NN	8	6	y	96.1%	89.3%	88.4%	96.1%	96.3%	93.2%	0.40
SRIF	NN	8	6	y	96.4%	90.1%	88.7%	96.1%	96.4%	93.5%	0.32
SRIF-CC	NN	8	6	y	74.3%	66.7%	62.3%	73.8%	74.4%	70.3%	0.32
SRIF-Combined	NN	6	5	n	83.2%	75.0%	73.1%	82.8%	83.1%	79.4%	**0.17**
LLAH-Affine	Clockwise	8	6	y	48.7%	37.6%	33.3%	48.4%	47.3%	43.1%	0.73
LLAH-Cross-Ratio	Clockwise	9	7	y	5.0%	2.0%	4.0%	5.1%	5.2%	4.3%	1.84
LLAH-Similarity	Clockwise	8	6	y	97.7%	91.4%	90.3%	97.6%	97.7%	**94.9**%	0.57
SRIF	Clockwise	8	6	y	97.7%	91.5%	90.2%	97.6%	97.7%	**94.9**%	0.37
SRIF-CC	Clockwise	8	6	y	76.5%	70.1%	64.6%	76.5%	77.2%	73.0%	0.39
SRIF-Combined	Clockwise	6	5	n	90.4%	81.2%	79.7%	90.7%	90.7%	86.5%	**0.17**
DETRIF					77.5%	71.4%	71.5%	73.4%	71.5%	73.6%	0.37
SSKSRIF					88.2%	86.5%	86.5%	86.9%	87.6%	87.1%	0.35

Table 5.4: Experimental results on WikiBook dataset by applying PSRIF as extension features

Method	Order	n	m	Add	Videos Retrieval Accuracy						s/q
					0^o	30^o	60^o	90^o	180^o	Avg	
LLAH-Affine	Clockwise	7	5	PSRIF	97.1%	89.8%	88.6%	96.8%	96.9%	93.8%	0.5
LLAH-Cross-Ratio	Clockwise	8	6	PSRIF	98.2%	92.4%	92.1%	98.3%	98.2%	95.8%	0.9
LLAH-Similarity	Clockwise	7	5	PSRIF	98.7%	96.0%	95.5%	98.8%	98.5%	97.5%	0.5
SRIF	Clockwise	7	5	PSRIF	98.8%	96.1%	95.6%	98.8%	98.7%	**97.6**%	0.45
SRIF-CC	Clockwise	7	5	PSRIF	97.8%	93.4%	92.5%	97.9%	97.8%	95.8%	**0.4**
SRIF-Combined	Clockwise	8	6	PSRIF	97.1%	94.3%	94.1%	97.3%	97.4%	96.04%	**0.4**

5.4.1.2 CartoDialect dataset's experimental results

It can be seen from the Table 5.5 that the best retrieval accuracy methods were SRIF and LLAH-Similarity in both approaches (applying clockwise order and NN order) with approximately 96% of accuracy. The second highest retrieval accuracy was LLAH-Affine with 92.6 % by applying NN order, which was approximately 11 % higher than with the clockwise order. The retrieval accuracy of LLAH-Cross-Ratio was the lowest. The best retrieval time was obtained by SRIF-Combined with approximately 0.24 second/query.

TABLE 5.5: The experimental results on CartoDialect dataset based on spatial space of word connected components .

Method	Order	n	m	Add	Videos Retrieval Accuracy						s/q
					0^o	30^o	60^o	90^o	180^o	Avg	
LLAH-Affine	NN	8	6	n	93.7%	91.6%	90.1%	94.1%	93.6%	92.6%	0.42
LLAH-Cross-Ratio	NN	9	7	y	36.0%	29.0%	28.0%	36.0%	36.0%	33.0%	0.71
LLAH-Similarity	NN	8	6	n	97.0%	95.9%	94.9%	96.8%	96.8%	96.3%	0.36
SRIF	NN	8	6	n	97.5%	96.2%	95.2%	97.5%	97.4%	**96.8%**	0.33
SRIF-CC	NN	8	6	n	79.1%	46.4%	45.9%	78.7%	79.2%	65.9%	0.30
SRIF-Combined	NN	7	5	n	89.6%	74.6%	72.6%	90.0%	89.5%	83.3%	**0.24**
LLAH-Affine	Clockwise	8	6	n	89.8%	87.0%	85.3%	89.9%	90.0%	88.4%	0.50
LLAH-Cross-Ratio	Clockwise	9	7	y	30.0%	23.0%	22.0%	30.0%	30.0%	27.0%	1.06
LLAH-Similarity	Clockwise	8	6	n	96.7%	95.6%	95.5%	96.9%	96.7%	96.3%	0.45
SRIF	Clockwise	8	6	n	96.8%	95.6%	95.7%	96.9%	96.8%	96.4%	0.33
SRIF-CC	Clockwise	8	6	n	69.2%	36.5%	36.7%	69.4%	69.2%	56.2%	0.39
SRIF-Combined	Clockwise	7	5	n	88.5%	72.9%	71.1%	88.2%	88.5%	81.8%	0.26
DETRIF					95.8%	94.5%	94.5%	94.2%	93.6%	94.5%	0.38
SSKSRIF					97.5%	97.2%	96.5%	97.6%	96.8%	97.1%	0.37

We can see in Table 5.6 that when applying PSRIF as extension features, SRIF-based methods and LLAH-based methods got better results in terms of videos accuracy retrieval and retrieval times of them were faster.

5.4.1.3 Tobacco dataset's experimental results

The results from this dataset are shown in the Table 5.7. It can be seen that retrieval accuracy of SRIF was the highest one in both approaches applying clockwise order and NN order, and SRIF applying clockwise order obtained an average accuracy of

Table 5.6: Experimental results on CartoDialect dataset by applying PSRIF as extension features

Method	Order	n	m	Add	Videos Retrieval Accuracy						s/q
					0^o	30^o	60^o	90^o	180^o	Avg	
LLAH-Affine	Clockwise	7	5	PSRIF	95.2%	90.2%	88.3%	95.2%	95.0%	92.7%	0.31
LLAH-Cross-Ratio	Clockwise	8	6	PSRIF	94.3%	90.0%	88.5%	94.3%	94.2%	92.2%	0.5
LLAH-Similarity	Clockwise	7	5	PSRIF	97.7%	96.2%	95.8%	97.1%	97.2%	96.8%	0.27
SRIF	Clockwise	6	4	PSRIF	98.2%	97.5%	96.8%	98.3%	98.3%	**97.8**%	0.27
SRIF-CC	Clockwise	6	4	PSRIF	96.9%	92.6%	92.5%	96.7%	96.9%	95.1%	**0.2**
SRIF-Combined	Clockwise	6	4	PSRIF	97.0%	94.0%	93.1%	97.2%	97.1%	95.6%	**0.2**

86.8%. The second highest retrieval accuracy was obtained with LLAH-Similarity, which was a little lower than SRIF. Retrieval accuracy of SRIF-CC was higher than LLAH-Affine, and the retrieval accuracy of LLAH-Cross-Ratio was the lowest. Finally, the best retrieval time was obtained by SRIF-Combined with an average time consumption of 0.26 second/query.

Table 5.7: The experimental results on Tobacco dataset based on spatial space of word connected components.

Method	Order	n	m	Add	Videos Retrieval Accuracy						s/q
					0^o	30^o	60^o	90^o	180^o	Avg	
LLAH-Affine	NN	8	6	y	70.6%	56.2%	56.4%	70.3%	70.6%	64.8%	0.75
LLAH-Cross-Ratio	NN	9	7	y	16.5%	12.1%	11.6%	16.3%	16.4%	14.6%	1.58
LLAH-Similarity	NN	8	6	y	87.8%	81.5%	81.6%	88.2%	87.9%	85.4%	0.52
SRIF	NN	8	6	y	88.0%	81.8%	81.9%	88.0%	88.2%	85.6%	0.37
SRIF-CC	NN	8	6	y	75.5%	64.2%	64.1%	75.4%	75.3%	70.9%	0.43
SRIF-Combined	NN	7	6	n	71.2%	55.6%	56.0%	71.4%	71.1%	65.1%	**0.23**
LLAH-Affine	Clockwise	8	6	y	66.8%	44.4%	45.2%	66.5%	66.4%	57.9%	0.87
LLAH-Cross-Ratio	Clockwise	9	7	y	14.1%	10.5%	10.1%	14.2%	14.1%	12.6%	2.10
LLAH-Similarity	Clockwise	8	6	y	86.8%	80.1%	79.8%	86.8%	86.6%	84.0%	0.68
SRIF	Clockwise	8	6	y	89.4%	83.0%	82.9%	89.6%	89.3%	**86.8**%	0.50
SRIF-CC	Clockwise	8	6	y	78.9%	68.0%	67.6%	78.5%	78.6%	74.3%	0.45
SRIF-Combined	Clockwise	7	6	n	69.9%	53.5%	53.7%	69.1%	69.0%	63.0%	0.25
DETRIF					70.8%	60.5%	60.5%	65.2%	61.6%	63.7%	0.38
SSKSRIF					80.2%	72.9%	70.0%	75.0%	71.7%	73.9%	0.37

We can see in Table 5.8, when applying PSRIF as extension features, LLAH-Affine, LLAH-Cross-Ratio,SRIF-CC and SRIF-Combined got better results in terms of videos accuracy retrieval, retrieval time of them were also faster. Especially, videos accuracy retrieval of LLAH-Cross-Ratio increased from the lowest one to the best one.

TABLE 5.8: Experimental results on Tobacco dataset by applying PSRIF as extension features

Method	Order	n	m	Add	Videos Retrieval Accuracy						s/q
					0^o	30^o	60^o	90^o	180^o	Avg	
LLAH-Affine	Clockwise	7	5	PSRIF	88.5%	77.3%	77.6%	88.6%	88.6%	84.1%	0.7
LLAH-Cross-Ratio	Clockwise	7	5	PSRIF	90.1%	79.8%	80.4%	90.2%	90.4%	**86.1%**	1.0
LLAH-Similarity	Clockwise	8	6	PSRIF	87.1%	79.1%	79.0%	87.1%	87.0%	83.8%	0.6
SRIF	Clockwise	7	5	PSRIF	88.1%	79.0%	78.9%	88.0%	87.7%	84.3%	0.5
SRIF-CC	Clockwise	8	6	PSRIF	82.4%	66.6%	65.5%	82.3%	82.3%	75.8%	0.7
SRIF-Combined	Clockwise	6	4	PSRIF	70.8%	61.1%	60.7%	70.6%	70.5%	66.7%	**0.4**

5.4.1.4 Discussion

It is clear that SRIF using distance ratio can get better accuracy than LLAH-Affine and LLAH-Cross-Ratio on the three datasets in the case they are not combined with PSRIF. This demonstrates that using invariant values from each 2 constraint points (in SRIF) is better than from 4, 5 constraint points(in LLAH). This is due to the fact that keypoint extraction errors always occur by camera's effects and the various noises related to camera-capture of document images. As a result, the more constraint points we combine the more errors keypoints we can have. This leads to the result that there are more incorrect descriptors when the number of constraint points is large.

Moreover, LLAH-Affine uses invariants which correspond to an area-ratio from two triangles, and this is the reason why it does not work well with keypoints extracted from centroids of letter CCs. In this specific case, 3 points can easily be aligned (e.g. three letters from a word). That is why the accuracy increases when testing with the CartoDialect dataset.

Regarding the invariance towards rotation effects, we observed that the automatic rotation did not affect much the retrieval accuracy results when the keypoints are extracted from word CCs or when frames are rotated by an angle of 90 and 180 degrees. It only affects the retrieval accuracy results when keypoints are extracted from letter CCs or when frames are rotated by other angles. This can be explained by the fact that

when rotating images, it becomes more difficult to make letter connected components separated, which leads to keypoint extraction errors. The drawback of features based

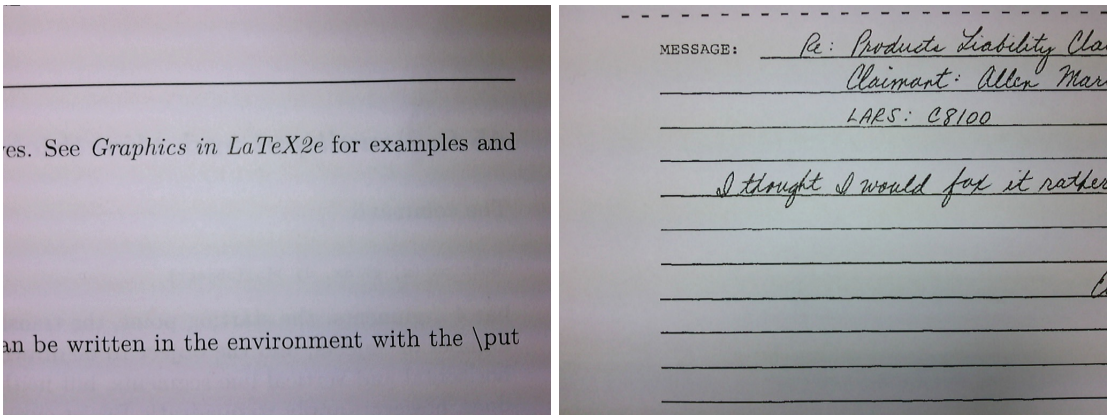

FIGURE 5.3: Insufficient text query examples.

on geometrical constraints between nearest points around a keypoint like LLAH, SRIF or DETRIF is that they need a stable keypoint extractor and enough text in order to work correctly. There are many queries in which the number of word CCs is not enough, which is the reason for which the global performances (retrieval accuracy) of all methods get closed to 100% (two example images are shown in Fig. 5.3).

Both SRIF and LLAH are based on geometrical constraints between m nearest points around a keypoint and use combinations of m points among n points to cope with keypoint extraction errors caused by the challenging conditions of camera capture. The bigger m is, the more discriminating feature vectors are. However, there are more erroneous keypoints, in this case, and vectors' dimension is higher. Therefore, instead of increasing the value of m, extension features are used. It makes feature vectors more discriminating and makes the retrieval system work better when the amount of text is small. In addition, time and scalability of the system is more efficient but computational complexity does not increase.

PSRIF extension features got much better results (accuracy performance) than the ranking of CCs based extension features. This can be explained by the fact that the polygons formed from the keypoints get a better discriminating power. It helps the extension features to better discriminate keypoints and then reduce the confusion.
One can also see from the results that LLAH and SRIF got a higher accuracy retrieval score with the proposed extension of features although they were computed with the small number of nearest points (4 or 5 points) from the neighborhood of the keypoints. This is very useful when the regions captured by the query images are very small, like for instance for queries captured using a camera pens.

Another characteristic that was observed based on the experimental results is that LLAH-Affine and LLAH-Cross-Ratio descriptors enhance their discriminative power when they are combined with PSRIF extension features. This results from the fact that these features are computed using the ratio between the areas of the triangles and they are not relying on the center keypoint. We can then come to the conclusion that in the case where many errors occur at the keypoint detector stage, these descriptors become more discriminative when the number of surrounding keypoints is large enough. However, these two descriptors need additional features when the number of surrounding keypoints becomes small and the discrimination power reduces.

Although DETRIF and SSKSRIF need more time to build the Delaunay triangulation structure compared to LLAH and SRIF, they still remain faster when the number of feature points is not too large in each query. This can be explained by the fact that the Delaunay triangulation of a set S of N points in the plane can be computed in $O(NlogN)$. Then, if S and N are points in the plane, if they are not all collinears, and if K denotes the number of points in S that lie on the boundary of the convex hull of S. Then any triangulation of P has $2N - 2 - K$ triangles and $3N - 3 - K$ edges. The computational complexity of building DETRIF descriptors depends on the time to find the adjacency vertexes and triangle. So, the computational complexity of building DETRIF descriptors is $O(N)$ which is similar to the computational complexity of LLAH.

5.4.2 Computation on spatial organization of dedicated keypoints

In this section, we present the experimental results of PSRIF, SSKSRIF methods and LLAH-based methods computed on keypoints obtained using the SURF, SIFT or ORB keypoints detectors on the three datasets we have created. Furthermore, all LLAH-based methods were employed using additional features proposed in Section 3.2.2 (Polygon-shape-based Scale and Rotation Invariant Features).

SURF and SIFT keypoints were extracted using the default parameters in Opencv. All tested descriptors were built following Algorithm 9 (in Section 3.3). For ORB detector, we set the maximum number of key points threshold equaling to the number of connected components $\times$ 5 and used FAST score to detect corners.

The parameters which are set for tested methods are shown in Table 5.9. For parameter n, m (n is nearest neighbor points, m is one combination among n points), we tested with various values and only the best results are reported in this thesis. For quantization's level q, we set it based on dimension d of descriptors in order to to avoid collisions from the hash function $H_{index} \in [0, H_{size}]$.

TABLE 5.9: Parameters for tested methods with spatial space of dedicated keypoints

Method	n	m	q
SSKSRIF			31
PSRIF	6	4	31
LLAH-Similarity+PSRIF	7	5	11
LLAH-Affine+PSRIF	7	5	31
LLAH-Cross Ratio+PSRIF	8	6	21

For the indexing and the retrieval phases, we applied the same approach and setting that in Section 5.4 and the parameters for sampling stable keypoints are given in Table 5.10.

TABLE 5.10: Sampled keypoints parameters

Dataset	Detector	Maximum percentage of keypoint	Minimum distance	Bluring size
WikiBook	SURF	15%	0.00015	7x7
CartoDialect	SURF	15%	0.00015	9x9
Tobacco	SURF	15%	0.00015	9x9
WikiBook	SIFT	30%	0.00015	9x9
CartoDialect	SIFT	30%	0.00015	9x9
Tobacco	SIFT	30%	0.00015	9x9
WikiBook	ORB	50%	0.00015	9x9
CartoDialect	ORB	40%	0.025	9x9
Tobacco	ORB	50%	0.00015	9x9

Finally, in order to assess the robustness of the system towards scale and rotation variations, each query was tested with rotation levels by an angle of 0, 10, 30, 45, and 90 degrees.

5.4.2.1 WikiBook dataset's experimental results

The results obtained on the WikiBook dataset with SURF keypoints detector are shown in Table 5.11. When considering the average video retrieval accuracy, PSRIF got the highest results with 82.7%, and the rest of the methods were ranked as follows: SSKSRIF, LLAH-Similarity+PSRIF LLAH-Affine+PSRIF, LLAH-Cross Ratio+PSRIF. The results of all methods show that video retrieval accuracy decreased when the query images are rotated. The worst case was when the query images were rotated by 45 degrees. Concerning the average time processing needed for the retrieval phase, PSRIF was the best with 0.9 second/query, and the rest of the methods were ranked as follows: SSKSRIF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF and LLAH-Cross Ratio+PSRIF.

TABLE 5.11: The results on WikiBook dataset with SURF keypoints

Method	Videos Retrieval Accuracy						s/q
	0^o	10^o	30^o	45^o	90^o	Avg	
SSKSRIF	96.1%	93.9%	70.1%	58.4%	94.7%	82.6%	1.0
PSRIF	95.3%	93.8%	68.9%	60.4%	95.3%	**82.7%**	**0.9**
LLAH-Similarity+PSRIF	91.7%	88.0%	59.9%	51.7%	91.9%	76.6%	1.1
LLAH-Affine+PSRIF	87.2%	81.1%	42.7%	32.8%	87.4%	66.2%	1.2
LLAH-Cross Ratio+PSRIF	84.1%	75.6%	27.3%	20.4%	84.1%	58.3%	2.1

The results obtained on the WikiBook dataset with the ORB keypoint detector are shown in Table 5.12. In this configuration, SSKSRIF got the highest results with 93.2% for the average video retrieval accuracy, and the other methods were ranked as follows: PSRIF, LLAH-Similarity+PSRIF LLAH-Affine+PSRIF, LLAH-Cross Ratio+PSRIF. The results of all the methods show that the video retrieval accuracy decreased slightly when the query images were rotated. Finally, from the processing time point of view, PSRIF and SSKSRIF were the best with 0.6 second/query, and the other methods were ranked as follows: LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF and LLAH-Cross Ratio+PSRIF.

TABLE 5.12: The results on WikiBook dataset with ORB keypoints

Method	Videos Retrieval Accuracy						s/q
	0^o	10^o	30^o	45^o	90^o	Avg	
SSKSRIF	95.2%	93.7%	90.0%	91.5%	94.7%	**93.2%**	**0.6**
PSRIF	89.8%	86.1%	81.1%	84.1%	89.6%	86.1%	**0.6**
LLAH-Similarity+PSRIF	88.5%	83.1%	76.3%	78.8%	88.4%	83.0%	1.0
LLAH-Affine+PSRIF	68.8%	56.1%	45.4%	50.5%	69.6%	58.0%	1.4
LLAH-Cross Ratio+PSRIF	29.1%	14.4%	11.4%	13.4%	23.2%	18.3%	2.7

The results being tested on WikiBook dataset with SIFT keypoints are shown in Table 5.13. Concerning the average of video retrieval accuracy, SSKSRIF got the highest results with 90.5%, and the rest methods were ranked as follows: LLAH-Similarity+PSRIF, PSRIF LLAH-Affine+PSRIF, LLAH-Cross Ratio+PSRIF. The results of all the methods show that video retrieval accuracy decreased slightly when query image was rotated. Concerning the average of the retrieval time, SSKSRIF was the best with 1.5 second/query, and the rest of the methods were ranked as follows: PSRIF, LLAH-Affine+PSRIF, LLAH-Similarity+PSRIF and LLAH-Cross Ratio+PSRIF.

5.4.2.2 CartoDialect dataset's experimental results

For the CartoDialect dataset, mainly composed of graphical parts, we firstly tested with the SURF keypoint detector and the results are shown in Table 5.14. Concerning average of video retrieval accuracy, SSKSRIF got the highest results with 77.2%, and

TABLE 5.13: The results on WikiBook dataset dataset with SIFT keypoints

Method	Videos Retrieval Accuracy						s/q
	0^o	10^o	30^o	45^o	90^o	Avg	
SSKSRIF	91.8%	91.6%	90.1%	88.5%	90.6%	**90.5%**	**1.5**
PSRIF	64.1%	64.0%	56.7%	55.5%	63.6%	60.78%	2.0
LLAH-Similarity+PSRIF	69.0%	70.1%	64.7%	63.0%	69.9%	67.3%	2.1
LLAH-Affine+PSRIF	58.1%	58.7%	51.5%	50.0%	58.8%	55.4%	2.0
LLAH-Cross Ratio+PSRIF	46.3%	46.4%	37.0%	36.3%	46.2%	42.4%	3.0

the other methods were ranked as follows: PSRIF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF, and LLAH-Cross Ratio+PSRIF. The results of all methods show that video retrieval accuracy decreased when query image was rotated. The worst case was when query image was rotated through 45 degrees. Concerning the average time needed for the retrieval step, PSRIF was the best with 0.6 second/query, followed by: SSKSRIF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF, LLAH-Cross Ratio+PSRIF.

TABLE 5.14: The results on CartoDialect dataset with SURF keypoints

Method	Videos Retrieval Accuracy						s/q
	0^o	10^o	30^o	45^o	90^o	Avg	
SSKSRIF	89.8%	83.6%	63.8%	59.2%	89.7%	**77.2%**	0.8
PSRIF	84.4%	76.3%	51.5%	45.7%	84.0%	68.3%	**0.6**
LLAH-Similarity+PSRIF	83.7%	75.3%	49.3%	42.5%	83.5%	66.8%	0.9
LLAH-Affine+PSRIF	77.5%	65.0%	25.1%	19.7%	77.3%	52.9%	1.1
LLAH-Cross Ratio+PSRIF	70.5%	68.1%	20.0%	14.1%	70.3%	51.5%	1.8

When using the ORB keypoint detector, we obtained the results presented in Table 5.15. SSKSRIF got the highest video retrieval accuracy with 40.1%, and the other methods were ranked as follows: PSRIF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF, and LLAH-Cross Ratio+PSRIF. The results of all methods show that video retrieval accuracy decreased when the query image is rotated. Concerning the retrieval time, PSRIF was the best with 1.1 second/query, followed by SSKSRIF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF, LLAH-Cross Ratio+PSRIF.

TABLE 5.15: The results on CartoDialect dataset with ORB keypoints

Method	Videos Retrieval Accuracy						s/q
	0^o	10^o	30^o	45^o	90^o	Avg	
SSKSRIF	51.8%	36.2%	29.1%	34.2%	49.6%	**40.1%**	1.2
PSRIF	24.8%	10.5%	8.4%	11.3%	24.1%	15.8%	**1.1**
LLAH-Similarity+PSRIF	22.2%	8.1%	6.3%	9.4%	22.1%	13.6%	1.7
LLAH-Affine+PSRIF	15.3%	3.3%	3.6%	4.6%	14.9%	8.3%	1.9
LLAH-Cross Ratio+PSRIF	7.0%	1.5%	1.0%	1.7%	7.5%	3.7%	3.0

Finally, the results with the SIFT keypoint detector are shown in Table 5.16. The global video retrieval accuracy shows that SSKSRIF got the highest results with 66.1%, and the other methods were ranked as follows: PSRIF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF, and LLAH-Cross Ratio+PSRIF. The results of all methods show that video retrieval accuracy decreased slightly when the query image is rotated. For the retrieval time, PSRIF was the best with 1.0 second/query, followed by SSKSRIF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF, LLAH-Cross Ratio+PSRIF.

TABLE 5.16: The results on CartoDialect dataset with SIFT keypoints

Method	Videos Retrieval Accuracy						s/q
	0^o	10^o	30^o	45^o	90^o	Avg	
SSKSRIF	71.1%	65.1%	62.8%	60.8%	71.8%	**66.1%**	1.3
PSRIF	70.5%	63.1%	61.3%	58.7%	69.7%	64.6%	**1.0**
LLAH-Similarity+PSRIF	57.5%	48.5%	46.6%	44.2%	56.2%	50.6%	1.4
LLAH-Affine+PSRIF	41.5%	33.4%	30.6%	26.5%	41.5%	34.7%	1.5
LLAH-Cross Ratio+PSRIF	36.4%	30.3%	27.0%	24.2%	36.2%	30.8%	2.1

5.4.2.3 Tobacco dataset's experimental results

We finally tested our work, using the well-known keypoint detectors from the literature on a heterogeneous corpus, the Tobacco dataset. The results are once again given using the SURF Keypoint detector, then the ORB keypoint detector and finally the SIFT one.

Results obtained with the SURF keypoint detector are shown in Table 5.17. The best accuracy was obtained by PSRIF with 90.0%, and the rest methods were ranked as follows: SSKSRIF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF and LLAH-Cross Ratio+PSRIF. The results of all methods show that video retrieval accuracy decreased when the query image was rotated. The worst case was when query images were rotated by 45 degrees. From the processing time point of view, PSRIF was the best with 0.6 second/query, and the rest methods were ranked as follows: SSKSRIFF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF, and LLAH-Cross Ratio+PSRIF.

TABLE 5.17: The results on Tobacco dataset with SURF keypoints

Method	Videos Retrieval Accuracy						s/q
	0^o	10^o	30^o	45^o	90^o	Avg	
SSKSRIF	95.7%	95.0%	83.1%	79.7%	95.1%	89.7%	0.8
PSRIF	95.0%	93.5%	84.3%	82.8%	94.7%	**90.0%**	**0.6**
LLAH-Similarity+PSRIF	91.7%	89.2%	75.7%	73.4%	91.7%	84.3%	0.9
LLAH-Affine+PSRIF	81.5%	75.1%	55.6%	52.5%	81.4%	69.2%	1.1
LLAH-Cross Ratio+PSRIF	79.1%	73.5%	52.3%	48.1%	77.3%	66.0%	1.4

With the ORB keypoint detector, the best average accuracy of video retrieval was obtained with SSKSRIF (75.4%), and the others results (ranked as follows: PSRIF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF and LLAH-Cross Ratio+PSRIF) are given in Table 5.17. The results of all methods show that video retrieval accuracy decreased slightly when the query image was rotated. Even if the SSKSRIF was the best, PSRIF still remains the fastest method (0.5 second/query), followed by SSKSRIFF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF, and LLAH-Cross Ratio+PSRIF.

TABLE 5.18: The results on Tobacco dataset with ORB keypoints

Method	Videos Retrieval Accuracy						s/q
	0^o	10^o	30^o	45^o	90^o	Avg	
SSKSRIF	80.8%	72.1%	70.4%	73.2%	80.5%	**75.4%**	0.6
PSRIF	74.3%	63.6%	61.6%	67.0%	74.2%	68.14%	**0.5**
LLAH-Similarity+PSRIF	65.2%	55.9%	54.0%	58.6%	65.7%	59.8%	0.9
LLAH-Affine+PSRIF	44.7%	27.1%	24.4%	30.7%	44.6%	34.3%	1.2
LLAH-Cross Ratio+PSRIF	17.7%	6%	5%	7%	17.3%	10.6%	2.4

Finally, for the tobacco dataset, we present the results obtained with the SIFT keypoint detector in Table 5.19. Again, SSKSRIF got the highest accuracy results with 91.1%, and the rest methods were ranked as follows: PSRIF, LLAH-Similarity+PSRIF, LLAH-Affine+PSRIF and LLAH-Cross Ratio+PSRIF. The results of all methods show that video retrieval accuracy decreased slightly when the query image was rotated. The result is the same concerning the processing time associated with the retrieval process with PSRIF which was the best with an average computation time of 0.8 second/query, followed by LLAH-Similarity+PSRIF, SSKSRIFF, LLAH-Affine+PSRIF, and LLAH-Cross Ratio+PSRIF.

TABLE 5.19: The results on Tobacco dataset with SIFT keypoints

Method	Videos Retrieval Accuracy						s/q
	0^o	10^o	30^o	45^o	90^o	Avg	
SSKSRIF	92.4%	91.9%	90.5%	90.6%	90.5%	**91.1%**	1.3
PSRIF	88.2%	88.0%	86.2%	86.4%	88.1%	87.3%	**0.8**
LLAH-Similarity+PSRIF	82.9%	82.9%	81.1%	80.7%	83.3%	82.1%	1.1
LLAH-Affine+PSRIF	64.2%	64.6%	62.8%	61.8%	64.2%	63.5%	1.3
LLAH-Cross Ratio+PSRIF	36.6%	36.6%	33.5%	35.0%	36.6%	35.6%	2.4

5.4.2.4 Discussion

The experimental results presented in the previous section, and computed on three different datasets, show that SRIF and SSKSRIF gave globally better results in terms of retrieval accuracy as well as retrieval time compared to LLAH based methods. This can

be explained by the fact that the invariant features of SRIF (or SSKSRIF) are computed relatively to the keypoint chosen as a reference. This enhances the discriminative power of SRIF (or SSKSRIF) even if the number of nearest points around the center point is small. Yet, invariant features of LLAH are not computed relatively to the reference keypoint, which may lead to the fact that one invariant feature may take into account many descriptors. For instance, the figure 5.4 shows an example of computation of the SRIF features from two points A and B for each keypoint P and Q. One can see that while LLAH features from four points A,B,C, and D for each keypoint P and Q are the same, the SRIF descriptors will be different.

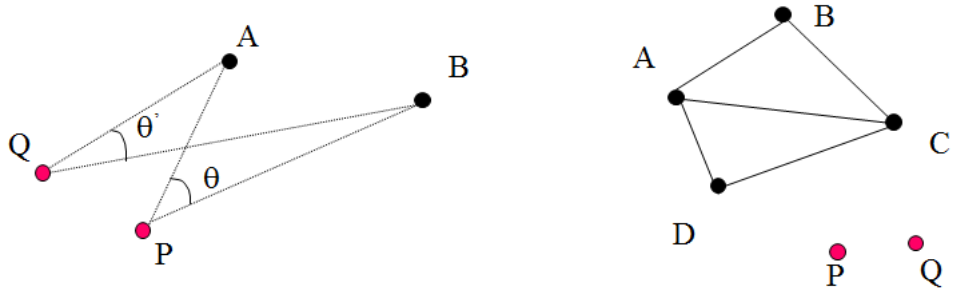

FIGURE 5.4: Distinctive SRIF features computed at P and Q (on the left) and the same LLAH features computed at P and Q (on the right).

The reason for which the retrieval accuracy results on CartoDialect dataset was not high is due to the fact that the spatial organization of sampled keypoints is not distinctive. These sampled keypoints come from the names or the identification number of the regions and the borders that are the same in all map and are highlighted in bold format. Thus, spatial information form these keypoints are not distinctive enough for descriptors to be built.

The essential challenge when building geometrical descriptors based on dedicated keypoint is the way to chose a stable keypoint detector that can be stable in two far scales from the SIFT or SURF pyramid scales, and that can be stable toward rotation transformations. In addition, in the context of heterogeneous content documents, the textures and the resolution of images are various. Thus, it is not easy to fix a threshold that can be adapted to all documents. In this case, the descriptors should be built from a non-uniform pyramid scale, e.g. the method proposed in [118].

Besides, the repeatability of keypoints detectors reduces when the image is rotated [118, 120], and this demonstrates why the accuracy of all descriptors decreased when the captured query is rotated. For example, the worst case is at 45 degrees rotation with SURF keypoint, because this is the most different angle between database document images and captured query.

5.5 Experimental results of dedicated descriptors combined with dedicated detectors

In this section, we present experimental results of popular dedicated descriptors combined with popular dedicated detectors.

For SIFT and SURF descriptors which are quite high dimensional descriptors, we apply Principal Components Analysis (PCA) to reduce the number of dimensions to 36-dimension vectors. As these features vectors are composed of a very large number of values, processing them means using of a lot of memory space and computation time. Valenzuela *et al.* [136] introduced a method using PCA to reduce the number of dimensions of SIFT and SURF vectors. PCA is used in the case that there is plenty of numeric variables (observed variables) and it is desired to find a lower number of principal components, that will be responsible for higher variance in the observed variables. These principal components can be used as predictor variables in the subsequent analysis. Valenzuela's experiments show that it is feasible to have an accurate low-dimensional feature vector after applying PCA.

For the systems using SIFT and SURF, reduced dimension by applying PCA, they were indexed using FLANN framework as described in [100]. The constructed index consists of a set of randomized kd-trees that are built by partitioning database descriptors. These kd-trees are searched in parallel in order to find nearest neighbors matching in high-dimensional spaces of descriptors.

For the system using binary descriptors such as ORB, BRISK, and FREAK, the binary feature vectors are indexed by LSH, whose index uses multi-probe LSH method from [137]. This indexing method is built on the well-known LSH technique and intelligently probes multiple buckets which are likely to contain query results in a hash table. It is more time and space efficient than ordinary LSH methods.

In the retrieval process, to filter the bad matching pairs we applied two methods. The first method is applied for searching only one nearest neighbor (NN) of each query descriptor. In this case, the matching is considered to be correct if the distance between the nearest and the query descriptor is equal or less than the threshold (10, 100, 70 for SURF, SIFT and binary descriptors) and it is taken into account for the voting process. The second one uses distance ratio between two nearest neighbors of each query descriptor [50]. It is considered to be correct and taken into account for the voting process if the ratio between the nearest and the second nearest descriptor is equal or less than the threshold (0.8).

Because of the large resolution, all maps (in CartoDialect dataset) are resized by a scale factor of 0.4 for reducing the resolution. To use the indexing framework (FLANN, LSH) for SIFT, SURF, ORB, we employ libraries integrated with OpenCV library. Our systems were implemented on a 64 GB RAM Linux machine running in C extended C++ environment with a single thread.

To set the orientation for a MSER region, firstly the ellipse that fits the region's contour is computed by the algorithm in [138] by using OpenCV. In this method, the orientation of the region which is fit by th ellipse region belongs to 0 to 180 degree. AIn order to enhance this direction from 0 to 180 degree to 0 to 360 degree, the orientation of the region is set following the ellipse's direction if numbers of contour points belonging to this direction are larger than the other side. Otherwise, it is set opposite the ellipse's direction. Furthermore, in order to capture neighborhood information around MSER regions, we proposed a way to extend MSER regions by increasing or decreasing the radius of each region. This strategy can be also used for other detected regions in order to make descriptors more distinctive.

5.5.1 Wikibook dataset's experimental results

The table 5.20 shows the results obtained on the WikiBook dataset. In this experimental results, SIFT descriptor combined with BRISK detector gave the highest videos retrieval accuracy with the average of 98.6%. Concerning the videos retrieval accuracy of SURF descriptors, combining them with MSER detector gave highest accuracy results in average with 95.5%. The retrieval time of FLANN LSH indexing binary descriptors was slower than the retrieval time of FLANN kd-trees indexing SIFT and SURF descriptors.

5.5.2 CartoDialect dataset's experimental results

For the CartoDialect dataset, the results are shown in the table 5.21. As we can see in the table 5.21, SIFT descriptors being computed with SIFT detector gave the best videos retrieval accuracy with the average of 98.94%. Concerning the videos retrieval accuracy of SURF descriptor, combining with MSER detector gave highest accuracy results in average with 98.5%. The retrieval time of FLANN LSH indexing binary descriptors was also slower than the retrieval time of FLANN kd-trees indexing SIFT and SURF descriptors.

.

TABLE 5.20: Experimental results on Wikibook dataset with popular dedicated detectors and descriptors

Descriptor	Detector	Size	NN	Videos Retrieval Accuracy						s/q
				0^o	30^o	60^o	90^o	180^o	Avg	
SIFT	SIFT	×1	1	73.7%	71.2%	71.0%	73.4%	73.2%	72.5%	1.28
SIFT	SIFT	×1	2	88.4%	84.6%	83.9%	87.7%	87.7%	86.4%	1.21
SIFT	SURF	×0.5	2	98.6%	93.5%	93.2%	98.6%	98.5%	96.4%	2.76
SIFT	BRISK	×1	2	98.7%	98.8%	98.4%	98.7%	98.5%	**98.6%**	1.4
SIFT	ORB	×1	2	98.3%	97.7%	97.1%	98.5%	98.3%	97.9%	1.5
SIFT	MSER	×1	2	96.6%	94.2%	93.8%	96.4%	96.5%	95.5%	0.83
SURF	SURF	×1	1	92.5%	10%	10%	92.6%	92.6%	59.5%	3.06
SURF	SURF	×1	2	94.6%	11%	11.0%	94.2%	93.8%	60.9%	3.1
SURF	SIFT	×5	2	55.2%	5%	5%	46%	43.8%	31%	1.2
SURF	BRISK	×5	2	94.0%	69.8%	69.9%	93.2%	93.2%	84.2%	1.8
SURF	ORB	×1	2	96.9%	83.0%	80.0%	96.1%	96.0%	90.4%	1.3
SURF	MSER	×4	2	93.7%	83.3%	81.7%	93.6%	92.8%	89.0%	0.47
ORB	ORB	×1	1	81.2%	79.5%	79.0%	80.0%	80.0%	79.9%	6.81
ORB	MSER	×1	2	43.7%	40.2%	40.1%	43.1%	43.2%	42.0%	5.0
BRISK	MSER	×1.5	2	75%	72.1%	72.0%	75.1%	75%	73.8%	33.0
FREAK	MSER	×1.5	2	78.5%	76.2%	76.0%	78.5%	78.4%	77.5%	42.0

TABLE 5.21: Experimental results on CartoDialect dataset with popular dedicated detectors and descriptors

Descriptor	Detector	Size	NN	Videos Retrieval Accuracy						s/q
				0^o	30^o	60^o	90^o	180^o	Avg	
SIFT	SIFT	×1	1	99.0%	98.8%	98.7%	99.0%	99.0%	98.9%	1.5
SIFT	SIFT	×1	2	99.0%	98.9%	98.8%	99.0%	99.0%	**98.94%**	1.6
SIFT	SURF	×0.5	2	99.0%	98.7%	98.7%	99.0%	99.0%	98.88%	8.2
SIFT	BRISK	×1	2	97.1%	94.9%	94.5%	97.2%	96.8%	96.1%	2.1
SIFT	ORB	×1	2	98.5%	97.5%	96.9%	98.2%	98.3%	97.8%	1.3
SIFT	MSER	×0.5	2	99.0%	98.7%	98.8%	99.0%	99.0%	98.6%	1.1
SURF	SURF	×1	1	92.6%	35.25%	35.1%	92.7%	91.9%	69.5%	5.4
SURF	SURF	×1	2	98.9%	38.1%	32.4%	98.5%	98.2%	73.2%	5.4
SURF	SIFT	×5	2	89.8%	53.8%	48.2%	83.1%	76.7%	70.3%	1.6
SURF	BRISK	×5	2	98.2%	83.9%	82.5%	98.2%	98.1%	92.1%	3.0
SURF	ORB	×1	2	98.7%	93.3%	91.5%	98.7%	98.4%	96.1%	1.3
SURF	MSER	×2	2	98.9%	98.2%	98.0%	98.9%	98.9%	98.5%	1.2
ORB	ORB	×1	1	92.7%	90.1%	89.5%	92.5%	92.5%	91.4%	11.3
ORB	MSER	×2	2	89.5%	70.2%	70.5%	89.1%	89.0%	81.6%	20.0
BRISK	MSER	×1	2	98.1%	94.6%	94.8%	98.1%	98.2%	96.7%	14.0
FREAK	MSER	×1	2	97.9%	94.0%	94.1%	97.7%	97.8%	96.3%	61.0

5.5.3 Tobacco dataset's experimental results

Finally, the results obtained from Tobacco dataset are shown in the table 5.22. It can be seen that SIFT descriptor combining with BRISK detector gave the highest Videos retrieval accuracy with the average of 98.5%. Concerning the videos retrieval accuracy of SURF descriptor, combining with ORB detector gave highest accuracy results in average with 97.6%. The retrieval time of FLANN LSH indexing for binary descriptors was also slower than the retrieval time of FLANN kd-trees indexing for SIFT and SURF descriptors.

TABLE 5.22: Experimental results on Tobacco dataset with popular dedicated detectors and descriptors

Descriptor	Detector	Size	NN	Videos Retrieval Accuracy						s/q
				0^o	30^o	60^o	90^o	180^o	Avg	
SIFT	SIFT	×1	1	91.2%	90.5%	90.0%	91.3%	91.3%	90.8%	2.1
SIFT	SIFT	×1	2	98.3%	98.8%	98.9%	98.4%	98.4%	**98.5%**	2.1
SIFT	SURF	×0.5	2	98.6%	98.1%	98.3%	98.5%	98.4%	98.4%	2.6
SIFT	BRISK	×1	2	97.4%	96.7%	96.4%	97.2%	97.4%	97.0%	2.3
SIFT	ORB	×1	2	96.3%	94.9%	95.0%	96.3%	96.3%	95.7%	1.7
SIFT	MSER	×0.5	2	94.8%	93.8%	93.8%	94.9%	95.0%	94.4%	0.4
SURF	SURF	×1	1	95.6%	89.7%	89.0%	95.4%	95.5%	93.0%	3.8
SURF	SURF	×1	2	98.3%	92.1%	91.3%	98.3%	98.0%	95.6%	3.8
SURF	SIFT	×5	2	97.6%	94.4%	92.6%	96.8%	95.6%	95.4%	2.2
SURF	BRISK	×4	2	95.8%	81.0%	79.5%	95.8%	95.5%	89.5%	2.3
SURF	ORB	×1	2	98.1%	97.4%	96.6%	98.0%	98.0%	97.6%	1.4
SURF	MSER	×2	2	93.7%	83.3%	82.1%	93.6%	92.8%	89.1%	0.5
ORB	ORB	×1	1	95.1%	89.5%	89.4%	95.1%	95.0%	92.8%	4.8
ORB	MSER	×1.5	2	95.3%	92.0%	92.0%	95.3%	95.2%	93.9%	12.7
BRISK	MSER	×1.5	2	82.2%	60.0%	61.5%	81.7%	82.1%	73.6%	7.0
FREAK	MSER	×1.5	2	92.0%	75.5%	76.0%	92.1%	92.0%	85.5%	15.0

5.5.4 Discussion

Experimental results obtained on three datasets show that the way to select the best nearest neighbors of query descriptors in the post-retrieval from the indexing system is able to affect the retrieval accuracy. Using distance ratio threshold between two nearest neighbors of each query descriptor to be able to make the retrieval accuracy be better than using the distance threshold for one nearest neighbor. This is due to the fact that using distance ratio threshold helps the retrieval phase and makes it able to filter the bad matching pairs which almost come from repeatable texture regions that correspond to the same text such as the same letters, symmetric or rotated letters and etc. Consequently, these bad matching pairs lead to confusion votes when they are not discarded when using the distance threshold for one nearest neighbor. For example, a

descriptor from a letter 'e' at a position can vote for another letter 'e' which appears at different positions in a document.

SIFT descriptor can work well when it is combined with various detectors. In some case e.g Wikibook dataset, SIFT descriptor gave better results by combining with SURF or BRISK or MSER detector. Furthermore, this descriptor is distinctive and accurate even though it is computed on regions which are reduced of a half of size. In addition, it can deal with rotation problem very well. By contrast, the accuracy of SURF descriptor rapidly decreases when query image rotated by an angle of 30 or 60 degrees. This problem of SURF is also reported in [139]. Overall, the retrieval accuracy of binary descriptors is lower than the retrieval accuracy of SIFT or SURF descriptors on three datasets.

5.6 Experimental results of proposed indexing methods

In this section, we present experimental results of proposed random trees indexing and hash-based indexing approaches compared with popular tree-based indexing approaches including randomized kd-tree, hierarchical trees and kmean trees [4, 100]. To extract local descriptors we used SIFT descriptors computed from SIFT keypoint [50] and the parameters were set by default in Opencv.

For our proposed random tree, the parameters were set as follows: the number of trees = 2; the numbers of random dimension K=1 for both Wikibook dataset and Tobacco dataset; K=6 for CartoDialect dataset; $\gamma = K$; ratioThreshold=0.8; ambiguousCheck=7; SIFT descriptors were not applied PCA to reduce the dimension. For proposed hashing index, SIFT descriptors are applied Principal Components Analysis (PCA) to reduce the number of dimensions to 36-dimension vectors. the other parameters were set as follows: the tolerance e=1.5; the maximum number of expansion bit b=10.

For tree-based indexing approaches [4, 100], we also set numbers of trees with 2 and default values for other parameters in Opencv and SIFT descriptors were not applied PCA to reduce the dimension. In the retrieval process, to filter the bad matching pairs we used distance ratio between two nearest neighbors of each query descriptor [50]. It is considered to be correct if the ratio between the nearest and the second nearest descriptor is equal or less than the threshold (0.8) and taken into account for the voting process.

5.6.1 Wikibook dataset's experimental results

The experimental results obtained on the Wikibook dataset are shown in the Table 5.23. It can be seen in the Table 5.23, concerning the average of video retrieval accuracy, Flann kd trees got the highest results with 88.3%, and the rest methods were ranked as follows: Flann kmean trees, flann hierarchical trees, proposed hashing and proposed random trees. Concerning the average of time retrieval, Flann kmean trees method is the best with 0.25 second/query, and the rest methods are ranked as follows: proposed hashing, Flann hierarchical trees, proposed random trees and Flann kd trees.

TABLE 5.23: Experimental results of tree based indexing methods on Wikibook dataset

Method	q scaling	Videos Retrieval Accuracy						s/q
		0^o	30^o	60^o	90^o	180^o	Avg	
Proposed random trees	1.0	50.0%	33.4%	33.0%	49.2%	50.0%	43.1%	1.6
Proposed random trees	0.5	79.6%	61.9%	62.0%	78.5%	71.4%	72.4%	0.3
Proposed hashing	1.0	40.9%	20.9%	20.7%	39.9%	40.1%	32.5%	1.2
Proposed hashing	0.5	64.5%	47.3%	46.5%	64.2%	61.8%	56.8%	0.3
Flann hierarchical trees	1.0	85.2%	80.4%	79.6%	85.3%	84.9%	83.0%	1.4
Flann hierarchical trees	0.5	85.0%	81.2%	80.6%	85.2%	85.0%	83.4%	0.38
Flann kmean trees	1.0	86.4%	83.2%	82.2%	86.5%	85.7%	84.8%	1.2
Flann kmean trees	0.5	86.6%	83.9%	83.3%	86.9%	86.8%	85.9%	**0.25**
Flann kd trees	1.0	89.2%	86.1%	85.4%	89.3%	88.9%	87.7%	1.8
Flann kd trees	0.5	89.3%	87.3%	86.6%	89.7%	88.9%	**88.3%**	0.35

5.6.2 CartoDialect dataset's experimental results

The Table 5.24 shows the experimental results on the CartoDialect dataset. As we can see from the Table 5.24, Flann kd trees got the highest results with 98.4%, and the rest methods were ranked as follows: Flann kmean trees, Flann hierarchical trees, proposed random trees and proposed hashing in terms of average of video retrieval accuracy. Proposed hashing method is the best with 0.19 second/query, and the rest methods are ranked as follows: Flann kmean trees, Flann hierarchical trees, Flann kd trees and proposed random trees in terms of average of video retrieval accuracy.

5.6.3 Tobacco dataset's experimental results

Finally, Tobacco dataset's experimental results are shown in the Table 5.25. It can be seen that Flann kd trees got the highest results with 97.8%, and the rest methods were ranked as follows: Flann kmean trees, Flann hierarchical trees, proposed random trees and proposed hashing in terms of average of video retrieval accuracy. Concerning

TABLE 5.24: Experimental results of tree based indexing methods on CartoDialect dataset

Method	q scaling	Videos Retrieval Accuracy						s/q
		0^o	30^o	60^o	90^o	180^o	Avg	
Proposed random trees	1.0	97.4%	94.1%	93.4%	97.3%	97.1%	95.8%	2.0
Proposed random trees	0.5	97.0%	94.0%	93.6%	97.0%	95.7%	95.7%	0.38
Proposed hashing	1.0	95.6%	92.4%	91.0%	96.0%	95.8%	94.1%	0.9
Proposed hashing	0.5	94.6%	91.0%	89.6%	94.8%	94.2%	92.8%	**0.19**
Flann hierarchical trees	1.0	97.9%	96.9%	96.5%	97.9%	97.7%	97.3%	1.5
Flann hierarchical trees	0.5	96.8%	95.0%	94.7%	96.7%	96.6%	95.9%	0.35
Flann kmean trees	1.0	98.0%	97.5%	97.5%	97.7%	98.2%	97.7%	0.95
Flann kmean trees	0.5	97.5%	96.0%	95.7%	97.6%	97.6%	96.8%	0.2
Flann kd trees	1.0	98.6%	98.3%	98.2%	98.5%	98.5%	**98.4**%	1.7
Flann kd trees	0.5	98.2%	97.5%	97.6%	98.2%	98.2%	97.9%	0.32

average of time retrieval, Flann kmean trees method is the best with 0.27 second/query, and the rest methods are ranked as follows: proposed hashing, proposed random trees, Flann hierarchical trees and Flann kd trees.

TABLE 5.25: Experimental results of tree based indexing methods on Tobacco dataset

Method	q scaling	Videos Retrieval Accuracy						s/q
		0^o	30^o	60^o	90^o	180^o	Avg	
Proposed random trees	1.0	97.2%	97.5%	97.4%	97.2%	97.0%	97.2%	1.7
Proposed random trees	0.5	96.6%	96.5%	96.6%	96.5%	96.5%	96.5%	0.4
Proposed hashing	1.0	97.1%	96.5%	96.9%	96.8%	96.8%	96.8%	1.2
Proposed hashing	0.5	95.8%	95.2%	95.2%	95.6%	95.8%	95.5%	0.3
Flann hierarchical trees	1.0	97.2%	97.5%	97.6%	97.0%	97.1%	97.2%	1.9
Flann hierarchical trees	0.5	96.6%	96.9%	96.8%	96.4%	96.4%	96.6%	0.46
Flann kmean trees	1.0	97.3%	97.6%	97.9%	97.3%	97.0%	97.4%	1.0
Flann kmean trees	0.5	96.8%	97.3%	97.2%	97.0%	96.7%	97.0%	**0.27**
Flann kd tree trees	1.0	97.8%	98.2%	98.3%	97.6%	97.5%	**97.8**%	2.5
Flann kd tree trees	0.5	97.3%	97.8%	97.9%	97.2%	97.0%	97.4%	0.56

5.6.4 Discussion

Experimental results demonstrate that proposed random trees and hashing indexing methods could approximately reach the accuracy of the stage art methods on Tobacco dataset although database descriptors are not stored in memory. Besides, the retrieval time of proposed random trees indexing was faster than kd-trees method in two datasets among three datasets and the retrieval time of proposed hashing indexing was the fastest on CartoDialect dataset. When filtering the bad matching pairs by using distance ratio

between two nearest neighbors of each query descriptor could not be checked with proposed indexing approaches, this could lead to confusion voting for retrieval documents and contributed to a reduction of the retrieval accuracy of proposed indexing approach.

Concerning retrieval accuracy, kd trees got the highest accuracy, which shows that partition based on the highest variance dimension is very robust. Regarding time retrieval, kmean trees method got the fastest results in two datasets including Wikibook and Tobacco dataset and proposed hashing method was the best with CartoDialect dataset. This is due to the fact that branching of kmean trees was set 32 branches, which makes the depth of each kmean tree is shorter than the kd-trees' depth. The retrieval via traversing the tree can, therefore, be done faster. Furthermore, Kmean trees method was little better than hierarchical trees method in terms of accuracy and retrieval time.

5.7 Experimental results of the extended hashing based method for indexing multi-kinds of features from multi-layer of images

In this section, we present the experimental results obtained with the proposed extended version of an hashing-based method for indexing multi-kinds of features from multi-layers documents on the CartoDialect dataset (presented in Section 4.4). As mentioned before, the maps from this dataset are composed of two main layers of information (the textual layer and the graphical layer), we decided to use different approaches on each layer.

For the textual layer, the geometrical version (spatial space) of the PSRIF descriptors was computed on the connected components (at the character level). We set parameters as following: n=6 and m=4.

For the graphical layer, we used the SIFT detector and descriptor, and we used the default parameters from OpenCV. We reduced the dimension of the SIFT descriptors using a PCA, and we binarized the vectors before the indexing and the retrieving processes. In addition, the query of this layer was scaled down with the factor 0.5 in order to speed up the retrieval time.

The table 5.26 shows the experimental results of the extended hashing based method for indexing multi-kinds of features from multi-layer of images on the CartoDialect dataset. It can be seen that the proposed system could improve the retrieval accuracy results in some specific cases like when the queries where rotated by an angle of 30 and 60 degrees although the average retrieval accuracy of it was a little lower than

the average retrieval accuracy of PSRIF with only textual layer. The drawback of the proposed indexing system is that it needs more time for processing and retrieving.

Because all maps have similar graphics layer which contains department's borders, it can lead to the case that there are some matched of SIFT descriptors (extracted from graphics layer) falling in different maps with captured map, which contributes to a reduction of the accuracy of PSRIF. The SIFT PCA method (historically dedicated to graphical objects) reached higher accuracy in some cases. Another explanation of these results is that the flipping bit strategy can lead to a wrong matching between the query feature vectors and the feature vector from the learning database. This may effect the final voting process and the homographic transformation using RANSAC algorithm.

TABLE 5.26: Experimental results on CartoDialect dataset of extended hashing based method indexing multi-kinds of features from multi-layers

Descriptors	Layers	Videos Retrieval Accuracy						s/q
		0^o	30^o	60^o	90^o	180^o	Avg	
PSRIF	Textual	98.2%	97.5%	96.8%	98.3%	98.3%	**97.8%**	**0.27**
PSRIF+SIFT PCA	Textual+Graphical	97.5%	98.2%	97.7%	97.5%	97.5%	97.68%	0.34

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we have proposed camera-based systems of information spotting, in huge repositories of heterogeneous document images, via local descriptors. We have proposed a set of generic feature descriptors for camera-based document image retrieval and spotting systems. We have also proposed indexing frameworks for automatic indexing of document image repositories. The dataset and ground-truth which were created for evaluating the camera-based document images retrieval and spotting systems have been made publicly available.

The feature descriptors that have been proposed in this thesis (SRIF, PSRIF, DETRIF and SSKSRIF) are promising for camera-based heterogeneous-content document image retrieval. These descriptors are built from spatial space information of nearest keypoints around a keypoint which is extracted from centroids of connected components. SRIF and PSRIF are computed from a local set of m nearest keypoints around a keypoint. From these keypoints, the invariant geometrical features are considered to be taken into account in the descriptor. To deal with erroneous keypoints, m combination from n nearest keypoints are computed. DETRIF and SSKSRIF can fix the way to combine local shape description without using any parameter via Delaunay triangulation (which is formed from a set of keypoints extracted from the document image). Since they are not built relying on image pixel level, they can be computed very fast if the number of keypoints is not too large. In addition, with low dimension, they can be indexed efficiently with a simple hash-based indexing method. In this thesis, we have also proposed a framework to compute the descriptors based on spatial space of dedicated keypoints (e.g SURF, SIFT or ORB) so that they can be used for retrieval and spotting in heterogeneous-content camera-based document image repositories. We

have also proposed a method to sample stable keypoints, which can adapt to the slight variations in scale.

Since the indexing plays an important role in camera-based heterogeneous-content document image retrieval and spotting systems, that use local descriptors, in this thesis we have proposed three robust indexing frameworks that can be employed without storing local descriptors in memory. This reduces the memory consumption and optimizes the retrieval time by discarding distance validation. The randomized clustering tree indexing inherits the properties of the kd-tree, kmean-tree and random forest, for selecting K dimensions randomly combined with the highest variance dimension from each node of the tree. We have also proposed a new weighted Euclidean distance between two local descriptors; in which the highest variance dimension is weighted.

Along with proposed descriptors as well indexing frameworks, we have proposed a simple robust way to compute shape orientation of MSER regions. This permits to combine them with dedicated descriptors like SIFT, SURF, ORB, BRISK, FREAK etc., maintaining the rotation invariance. In case when the descriptors are able to capture neighborhood information around MSER regions we also propose a way to extend MSER regions by increasing the radius of each region. This strategy can be used for other detected regions in order to make descriptors more distinctive. Moreover, we have employed the extended hashing-based method for indexing multi-kinds of features from multi-layers of image, where each kind of the feature is adapted to the content in the corresponding layer of the image. This strategy is not only applied for uniform feature type but also for multiple feature types from multi-layers of the image.

Concerning evaluation of camera-based document image retrieval and spotting systems, we have built a new dataset which has been made freely and publicly available for the scientific community. This dataset is comprised of three subparts that represent the three different evaluation contexts of content in document images: the textual content, graphical content heterogeneous content.

6.2 Future Work

Although our proposed methods have achieved reasonable results for camera-based heterogeneous-content document image retrieval and spotting using local descriptors, there is still a long way heading to the perfect solution for camera-based document image retrieval and spotting problems. The future directions of our research are discussed as follows.

The proposed descriptors do not capture local texture at dedicated keypoint and this decreases the accuracy when the scale level or rotation angle of the captured query is too different from original database image or captured query. A generic combination between spatial keypoint based descriptors and textures descriptor can be applied, which can make these descriptors more distinctive via not only local texture information but also neighborhood spatial information. Furthermore, how to develop brain semantics' inspired descriptors needs to be studied.

Instead of separating the image into multiple layers, the direction that employs multiple detectors combined with multiple descriptors without separating the image into multiple layers is interesting to be considered. This is because some image separation techniques require a very robust separation approaches in order to achieve the same accuracy for the original image and the captured query. Besides, image separation into multiple layers can lead to insufficient information in layers that affect the spatial keypoints based descriptors and the texture descriptors.

Although the best float-type descriptors like SIFT or SURF outperform binary descriptors in terms of accuracy, the performance gap is not huge. Binary descriptors are preferable in some applications which have a strict requirement on running time and memory. This is because that using binary descriptors usually requires less memory to store descriptors than using float-type descriptors. Yet, matching binary descriptors can be executed extremely fast in the modern computers by machine instructions but it is still too slow when we have to match millions of them, which often happens in image retrieval and specially for large-scale image retrieval. In these cases, matching binary descriptors by using LSH indexing is too slow compared to matching float-type descriptors with a tree-based approximate nearest neighbor indexing algorithms. Thus, for fast nearest neighbor searching of binary descriptors, it still needs more efficient indexing to deal with. Because of this, extending our proposed randomized clustering tree indexing for binary descriptors or an indexing system that can combine hashing-based and tree-based techniques, needs to be studied and developed. Furthermore, in the case of very large datasets, approaches of distributing the database of local descriptors to multiple machines in a computing cluster and performing the nearest neighbor search in parellel, using the cluster, should be employed.

Appendix A

List of Publications

This thesis has led to the following publications:

National Conference and Workshop Papers

Quoc Bao Dang, Muhammad Muzzamil Luqman, Mickaël Coustaty, Nibal Nayef, Jean-Marc Ogier, and Cao De Tran. A multi-layer separation based system for camera-based complex map image retrieval, In *CORIA-CIFED*, pages 359-362, 2014.

International Conference and Workshop Papers

Quoc Bao Dang, Muhammad Muzzamil Luqman, Mickaël Coustaty, Nibal Nayef, Jean-Marc Ogier, and Cao De Tran. A system for camera-based complex map image retrieval using a multi-layer approach. Short paper published in *IAPR International workshop on Document Analysis System* (DAS), 2014.

Quoc Bao Dang, Muhammad Muzzamil Luqman, Mickaël Coustaty, Nibal Nayef, Jean-Marc Ogier, and Cao De Tran. A multi-layer approach for camera-based complex map image retrieval and spotting system. In *Image Processing Theory, Tools and Applications (IPTA)*, page 1-6, IEEE, 2014.

Quoc Bao Dang, Viet Phuong Le, Muhammad Muzzamil Luqman, Mickaël Coustaty, Cao De Tran, and Jean-Marc Ogier. A System for Camera-Based Retrieval of Heterogeneous-Content Complex Linguistic Map. In *International Workshop on Graphics Recognition (GREC)*, pages 86-99, Springer, 2015.

Quoc Bao Dang, Muhammad Muzzamil Luqman, Mickaël Coustaty, Cao De Tran, and Jean-Marc Ogier. SRIF: Scale and Rotation Invariant Features for camera-based document image retrieval. In *Document Analysis and Recognition (ICDAR) 13th International Conference on*, pages 601-605, IEEE, 2015.

Quoc Bao Dang, Viet Phuong Le, Muhammad Muzzamil Luqman, Mickaël Coustaty, Cao De Tran, and Jean-Marc Ogier. Camera-based document image retrieval system using local features - comparing SRIF with LLAH, SIFT, SURF and ORB. In *Document Analysis and Recognition (ICDAR) 13th International Conference on*, pages 1211-1215, IEEE, 2015.

Viet Phuong Le, Quoc Bao Dang, and Cao De Tran. Logo spotting on document images using local features. In *Proceedings of the Sixth International Symposium on Information and Communication Technology*, pages 252-259, ACM, 2015.

Quoc Bao Dang, Marçal Rusiñol, Mickaël Coustaty, Muhammad Muzzamil Luqman, Cao De Tran, and Jean-Marc Ogier. Delaunay Triangulation-Based Features for Camera-Based Document Image Retrieval System. In *Document Analysis Systems (DAS) 12th IAPR Workshop on*, pages 1-6, IEEE, 2016.

Quoc Bao Dang, Mickaël Coustaty, Muhammad Muzzamil Luqman, Silvia Gally, Paule-Annick Davoine, Jean-Marc Ogier, and Jean-Christophe Burie. Camera-based document image spotting system for complex linguistic maps. In *Systems, Man, and Cybernetics (SMC) International Conference on*, page 003246-003251, IEEE, 2016.

Quoc Bao Dang, Mickaël Coustaty, Muhammad Muzzamil Luqman, Cao De Tran, and Jean-Marc Ogier. Polygon-shape-based Scale and Rotation Invariant Features for camera-based document image retrieval. In *Pattern Recognition (ICPR) 23rd International Conference on*, pages 2434-2439, IEEE, 2016.

Quoc Bao Dang, Mickaël Coustaty, Muhammad Muzzamil Luqman, Cao De Tran, and Jean-Marc Ogier. A randomized hierarchical trees indexing approach for camera-based information spotting. In *Pattern Recognition (ICPR) 24th International Conference on*, IEEE, 2018 (accepted).

National Journal Paper

Quoc Bao Dang and Cao De Tran. Camera-based document image retrieval and spotting system. *Information technology researches and Applications in Vietnamese Mekong Delta*, Internal Journal of Can Tho University, 2016.

International Journal Paper

Quoc Bao Dang, Mickaël Coustaty, Muhammad Muzzamil Luqman, Cao De Tran, and Jean-Marc Ogier. New Spatial-Organization-Based Scale and Rotation Invariant Features for Heterogeneous-Content Camera-Based Document Image Retrieval. In *Journal of Pattern Recognition Letters*, 2018 (under review, 2nd revision).

Bibliography

[1] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV), 2006*, pages 430–443. Springer, 2006.

[2] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision (ECCV), 2010*, pages 778–792. Springer, 2010.

[3] Bin Fan, Zhenhua Wang, and Fuchao Wu. *Local Image Descriptor: Modern Approaches*. Springer, 2015.

[4] Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2227–2240, 2014.

[5] Grant Schindler, Matthew Brown, and Richard Szeliski. City-scale location recognition. In *Computer Vision and Pattern Recognition (CVPR) 2007. IEEE Conference on*, pages 1–7. IEEE, 2007.

[6] Koichi Kise, Megumi Chikano, Kazumasa Iwata, Masakazu Iwamura, Seiichi Uchida, and Shinichiro Omachi. Expansion of queries and databases for improving the retrieval accuracy of document portions: an application to a camera-pen system. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (DAS) 2010*, pages 309–316. ACM, 2010.

[7] David Doermann. The indexing and retrieval of document images: A survey. *Computer Vision and Image Understanding*, pages 287–298, 1998.

[8] Mandar Mitra and BB Chaudhuri. Information retrieval from documents: A survey. *Information retrieval*, pages 141–163, 2000.

[9] Marçal Rusiñol and Josep Lladós. *Symbol spotting in digital libraries: Focused retrieval over graphic-rich document collections*. Springer Science & Business Media, 2010.

[10] Kai Kunze, Katsuma Tanaka, Masakazu Iwamura, and Koichi Kise. Annotate me: supporting active reading using real-time document image retrieval on mobile devices. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, pages 231–234. ACM, 2013.

[11] KC Santosh and Laurent Wendling. Graphical symbol recognition. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 2015.

[12] KC Santosh. Complex and composite graphical symbol recognition and retrieval: A quick review. In *International Conference on Recent Trends in Image Processing and Pattern Recognition*, pages 3–15. Springer, 2016.

[13] H Waruna H Premachandra, Chinthaka Premachandra, Chandana Dinesh Parape, and Hiroharu Kawanaka. Speed-up ellipse enclosing character detection approach for large-size document images by parallel scanning and hough transform. *International Journal of Machine Learning and Cybernetics*, pages 371–378, 2017.

[14] JianGuo Wang, Joshua Zhexue Huang, Jiafeng Guo, and Yanyan Lan. Query ranking model for search engine query recommendation. *International Journal of Machine Learning and Cybernetics*, pages 1019–1038, 2017.

[15] Ju-Xiang Zhou, Xiao-dong Liu, Tian-wei Xu, Jian-hou Gan, and Wan-quan Liu. A new fusion approach for content based image retrieval with color histogram and local directional pattern. *International Journal of Machine Learning and Cybernetics*, pages 1–13.

[16] Kazem Taghva. Name identification and extraction with formal concept analysis. *International Journal of Machine Learning and Cybernetics*, pages 171–178, 2017.

[17] Simone Marinai, Emanuele Marino, Francesca Cesarini, and Giovanni Soda. A general system for the retrieval of document images from digital libraries. In *Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on*, pages 150–173. IEEE, 2004.

[18] Paul Viola, James Rinker, and Martin Law. Automatic fax routing. In *International Workshop on Document Analysis Systems*, pages 484–495. Springer, 2004.

[19] Guillaume Chiron, Antoine Doucet, Mickael Coustaty, Muriel Visani, and Jean-Philippe Moreux. Impact of ocr errors on the use of digital libraries: Towards a better access to information. In *Joint Conference on Digital Libraries (JCDL), 2017 ACM/IEEE Joint Conference on*, pages 1–4. IEEE, 2017.

[20] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, pages 177–280, 2008.

[21] Jing Li and Nigel M Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, pages 1771–1787, 2008.

[22] Qiong Liu, Don Kimber, Chunyuan Liao, Lynn Wilcox, et al. High accuracy and language independent document retrieval with a fast invariant transform. In *IEEE International Conference on Multimedia and Expo (ICME) 2009*, pages 386–389. IEEE, 2009.

[23] Qiong Liu and Chunyuan Liao. Paperui. In *International Workshop on Camera-Based Document Analysis and Recognition (CBDAR)*, pages 83–100. Springer, 2012.

[24] Kazutaka Takeda, Koichi Kise, and Masakazu Iwamura. Real-time document image retrieval on a smartphone. In *Document Analysis Systems (DAS) 2012. 10th IAPR International Workshop on*, pages 225–229. IEEE, 2012.

[25] Jonathan J Hull, Berna Erol, Jamey Graham, Qifa Ke, Hidenobu Kishi, Jorge Moraleda, and Daniel G Van Olst. Paper-based augmented reality. In *Artificial Reality and Telexistence, 17th International Conference on*, pages 205–209. IEEE, 2007.

[26] Jian Liang, David Doermann, and Huiping Li. Camera-based analysis of text and documents: a survey. *International Journal of Document Analysis and Recognition (IJDAR)*, pages 84–104, 2005.

[27] Electronic content management. URL https://www.imagenetconsulting.com/products/electronic-content-management/#facts.

[28] Xu Liu and David Doermann. Mobile retriever-finding document with a snapshot. In *International Workshop on Camera-Based Document Analysis and Recognition (CBDAR)*, pages 29–34, 2007.

[29] Google goggles in action. URL http://www.google.com/mobile/.

[30] Kooaba. URL http://kooaba.com/.

[31] Marçal Rusiñol and Josep Lladós. Word and symbol spotting using spatial organization of local descriptors. In *Document Analysis Systems (DAS), 2018. The Eighth IAPR International Workshop on*, pages 489–496. IEEE, 2008.

[32] Chew Lim Tan, Xi Zhang, and Linlin Li. Image based retrieval and keyword spotting in documents. In *Handbook of Document Image Processing and Recognition*, pages 805–842. Springer, 2014.

[33] Wilkinson Tomas and Brun Anders. Semantic and verbatim word spotting using deep neural networks. In *International Conference on Frontiers in Handwriting Recognition (ICFHR), October 23-26, 2016, Shenzhen, China.*, 2016.

[34] Youssef Elfakir, Ghizlane Khaissidi, Mostafa Mrabti, Driss Chenouni, and Mounim El Yacoubi. Word spotting in handwritten arabic documents using bag-of-descriptors. 2016.

[35] C Thontadari and CJ Prabhakar. Scale space co-occurrence hog features for word spotting in handwritten document images. *International Journal of Computer Vision and Image Processing (IJCVIP)*, pages 71–86, 2016.

[36] Marçal Rusinol, Dimosthenis Karatzas, and Josep Lladós. Spotting graphical symbols in camera-acquired documents in real time. *Proceedings of the Tenth IAPR International Workshop on Graphics Recognition (GREC), 2013*, 2013.

[37] Muhammad Muzzamil Luqman, Jean-Yves Ramel, Josep Lladós, and Thierry Brouard. Subgraph spotting through explicit graph embedding: An application to content spotting in graphic document images. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 870–874. IEEE, 2011.

[38] Viet Phuong Le, Muriel Visani, Cao De Tran, and Jean-Marc Ogier. Improving logo spotting and matching for document categorization by a post-filter based on homography. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 270–274. IEEE, 2013.

[39] Hideaki Yanagisawa, Daisuke Ishii, and Hiroshi Watanabe. Face detection for comic images with deformable part model. In *Proceedings of The Institute of Image Electronics Engineers of Japan (IIEEJ) Image Electronics and Visual Computing Workshop*, 2014.

[40] Thanh-Nam Le, Muhammad Muzzamil Luqman, Jean-Christophe Burie, and Jean-Marc Ogier. Retrieval of comic book images using context relevance information. In *Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding*, page 12. ACM, 2016.

[41] Sandy Martedi, Hideaki Uchiyama, and Hideo Saito. Clickable augmented documents. In *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, pages 162–166. IEEE, 2010.

[42] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, page 50. Manchester, UK, 1988.

[43] Hans P Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI 1977, 1977.

[44] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, pages 63–86, 2004.

[45] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition (CVPR), 1994. Proceedings., IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[46] Stephen M Smith and J Michael Brady. Susan—a new approach to low level image processing. *International journal of computer vision*, pages 45–78, 1997.

[47] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[48] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV) 2011*, pages 2548–2555. IEEE, 2011.

[49] Elmar Mair, Gregory D Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Proceedings of the 11th European Conference on Computer Vision (ECCV) 2010*, pages 183–196. Springer, 2010.

[50] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, pages 91–110, 2004.

[51] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). In *Computer vision and image understanding*, pages 346–359. 2008.

[52] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, pages 761–767, 2004.

[53] Tomohiro Nakai, Koichi Kise, and Masakazu Iwamura. Camera based document image retrieval with more time and memory efficient llah. *International Workshop on Camera-Based Document Analysis and Recognition (CBDAR)*, pages 21–28, 2007.

[54] Tomohiro Nakai, Koichi Kise, and Masakazu Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *International Workshop on Document Analysis Systems (DAS) 2006*, pages 541–552. Springer, 2006.

[55] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 509–522, 2002.

[56] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, pages 971–987, 2002.

[57] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. In *European Conference on Computer Vision*, pages 102–115. Springer, 2008.

[58] Paul L Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, pages 291–307, 1999.

[59] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. Ieee, 2012.

[60] Sajib Saha and Vincent Démoulin. Aloha: An efficient binary descriptor based on haar features. In *2012 19th IEEE International Conference on Image Processing (ICIP)*, pages 2345–2348. IEEE, 2012.

[61] M. M. Bronstein C. Strecha, A. M. Bronstein and Pascal Fua. LDAHash: Improved Matching with Smaller Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[62] Lin Zhang, Zhiqiang Zhou, and Hongyu Li. Binary gabor pattern: An efficient and robust descriptor for texture classification. In *19th IEEE International Conference on Image Processing (ICIP)2012*, pages 81–84. IEEE, 2012.

[63] Tomasz Trzcinski and Vincent Lepetit. Efficient discriminative projections for compact binary descriptors. In *European Conference on Computer Vision (ECCV) 2012*, pages 228–242. Springer, 2012.

[64] Tomasz Trzcinski, Mario Christoudias, Pascal Fua, and Vincent Lepetit. Boosting binary keypoint descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*, pages 2874–2881, 2013.

[65] Tomohiro Nakai, Koichi Kise, and Masakazu Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *Proceedings of International Workshop on Document Analysis Systems(DAS)*, pages 541–552. Springer, 2006.

[66] Tomohiro Nakai, Koichi Kise, and Masakazu Iwamura. Hashing with local combinations of feature points and its application to camera-based document image retrieval. *International Workshop on Camera-Based Document Analysis and Recognition (CBDAR), 2005*, pages 87–94, 2005.

[67] Masakazu Iwamura, Tomohiro Nakai, and Koichi Kise. Improvement of retrieval speed and required amount of memory for geometric hashing by combining local invariants. In *Proceedings 18th British Machine Vision Conference (BMVC) 2007, year = 2007, month = sep, pages = 1010–1019.*

[68] Kazutaka Takeda, Koichi Kise, and Masakazu Iwamura. Real-time document image retrieval for a 10 million pages database with a memory efficient and stability improved llah. In *International Conference on Document Analysis and Recognition (ICDAR), 2011*, pages 1054–1058, September 2011.

[69] Tomohiro Nakai, Koichi Kise, and Masakazu Iwamura. Real-time retrieval for images of documents in various languages using a web camera. In *Document Analysis and Recognition (ICDAR) 2009. 10th International Conference on*, pages 146–150. IEEE, 2009.

[70] Shijian Lu and Chew Lim Tan. Retrieval of machine-printed latin documents through word shape coding. *Pattern Recognition*, pages 1799–1809, 2008.

[71] Agnes Desolneux, Lionel Moisan, and Jean-Michel Morel. *From gestalt theory to image analysis: a probabilistic approach.* Springer Science & Business Media, 2007.

[72] Karen A Panetta, Eric J Wharton, and Sos S Agaian. Human visual system-based image enhancement and logarithmic contrast measure. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, pages 174–188, 2008.

[73] Azeddine Beghdadi, M-C Larabi, Abdesselam Bouzerdoum, and Khan M Iftekharuddin. A survey of perceptual image processing methods. *Signal Processing: Image Communication*, pages 811–831, 2013.

[74] Qing-Fang Zheng, Wei-Qiang Wang, and Wen Gao. Effective and efficient object-based image retrieval using visual phrases. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 77–80. ACM, 2006.

[75] Sebastian Nowozin and Christoph H Lampert. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, pages 185–365, 2011.

[76] Matthew B Blaschko and Christoph H Lampert. Learning to localize objects with structured output regression. In *European conference on computer vision*, pages 2–15. Springer, 2008.

[77] Zhuowen Tu. Auto-context and its application to high-level vision tasks. In *Computer Vision and Pattern Recognition (CVPR), 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[78] Peter Kontschieder, Samuel Rota Bulo, Horst Bischof, and Marcello Pelillo. Structured class-labels in random forests for semantic image labelling. In *2011 International Conference on Computer Vision*, pages 2190–2197. IEEE, 2011.

[79] Yiqing Yang, Zhouyuan Li, Li Zhang, Christopher Murphy, Jim Ver Hoeve, and Hongrui Jiang. Local label descriptor for example based semantic image labeling. In *European Conference on Computer Vision*, pages 361–375. Springer, 2012.

[80] Matthew Maestri, Jeffrey Odel, and Jay Hegdé. Semantic descriptor ranking: a quantitative method for evaluating qualitative verbal reports of visual cognition in the laboratory or the clinic. *Frontiers in psychology*, 2014.

[81] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[82] Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of computing*, pages 321–350, 2012.

[83] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, page 391, 1990.

[84] Christos Faloutsos and Douglas W Oard. A survey of information retrieval and filtering methods. Technical report, 1998.

[85] Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.

[86] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, pages 21–27, 1967.

[87] Richard O Duda, Peter E Hart, and David G Stork. Pattern classification and scene analysis 2nd ed. 1995.

[88] Christos Faloutsos, Ron Barber, Myron Flickner, Jim Hafner, Wayne Niblack, Dragutin Petkovic, and William Equitz. Efficient and effective querying by image content. *Journal of intelligent information systems*, pages 231–262, 1994.

[89] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, et al. Query by image and video content: The qbic system. *computer*, pages 23–32, 1995.

[90] Arnold Smeulders and Ramesh Jain. *Image databases and multi-media search.* World Scientific, 1998.

[91] Trevor Hastie and Robert Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE transactions on pattern analysis and machine intelligence*, pages 607–616, 1996.

[92] Scott Cost and Steven Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine learning*, pages 57–78, 1993.

[93] Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko. *Similarity search: the metric space approach.* Springer Science & Business Media, 2006.

[94] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, pages 209–226, 1977.

[95] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

[96] Sunil Arya, David M Mount, Nathan Netanyahu, Ruth Silverman, and Angela Y Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. 1994.

[97] Jeffrey S Beis and David G Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Computer Vision and Pattern Recognition (CVPR), 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1000–1006. IEEE, 1997.

[98] Marius Muja and David G Lowe. Fast matching of binary features. In *Computer and Robot Vision (CRV), 2012 Ninth Conference on*, pages 404–410. IEEE, 2012.

[99] Sergey Brin. Near neighbor search in large metric spaces. 1995.

[100] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *The 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, page 2, 2009.

[101] Chanop Silpa-Anan and Richard Hartley. Optimised kd-trees for fast image descriptor matching. In *Computer Vision and Pattern Recognition (CVPR), 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[102] Keinosuke Fukunaga and Patrenahalli M. Narendra. A branch and bound algorithm for computing k-nearest neighbors. *IEEE transactions on computers*, pages 750–753, 1975.

[103] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition (CVPR), 2006 IEEE computer society conference on*, pages 2161–2168. IEEE, 2006.

[104] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Proceedings of Very Large Databases (VLDB)*, pages 518–529, 1999.

[105] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science (FOCS), 2006. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.

[106] Q Lv, W Josephson, Z Wang, M Charikar, and K Li. Efficient indexing for high-dimensional similarity search. In *Proceedings of Very Large Data Bases (VLDB)*, pages 950–961, 2007.

[107] Mayank Bawa, Tyson Condie, and Prasanna Ganesan. Lsh forest: self-tuning indexes for similarity search. In *Proceedings of the 14th international conference on World Wide Web*, pages 651–660. ACM, 2005.

[108] Koichi Kise, Kazuto Noguchi, and Masakazu Iwamura. Simple representation and approximate search of feature vectors for large-scale object recognition. In *The British Machine Vision Conference (BMVC)*, pages 1–10, 2007.

[109] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, pages II–II. IEEE, 2004.

[110] Tomohiro Sakata, Nobuaki Matozaki, Koichi Kise, and Masakazu Iwamura. Osaka prefecture university at trecvid 2011. *Proceedings of TRECVID. NIST*, 2011.

[111] Julien Rabin, Julie Delon, and Yann Gousseau. A contrario matching of sift-like descriptors. In *Pattern Recognition, 2008. 19th International Conference on International Conference on Pattern Recognition (ICPR 2008)*, pages 1–4. IEEE, 2008.

[112] Hideaki Uchiyama and Hideo Saito. Augmenting text document by on-line learning of local arrangement of keypoints. In *Mixed and augmented reality (ISMAR) 2009. 8th IEEE international symposium on*, pages 95–98. IEEE, 2009.

[113] Su Yang. Symbol recognition via statistical integration of pixel-level constraint histograms: A new descriptor. *IEEE transactions on pattern analysis and machine intelligence*, pages 278–281, 2005.

[114] Der-Tsai Lee and Arthur K Lin. Generalized delaunay triangulation for planar graphs. *Discrete & Computational Geometry*, pages 201–217, 1986.

[115] Georgios D Evangelidis and Christian Bauckhage. Efficient subframe video alignment using short descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 2371–2386, 2013.

[116] Paul McIlroy, Shahram Izadi, and Andrew Fitzgibbon. Kinectrack: Agile 6-dof tracking using a projected dot pattern. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pages 23–29. IEEE, 2012.

[117] Dustin Lang, DW Hogg, K Mierle, et al. Blind astrometric calibration of arbitrary astronomical images. *Astrometry. net*, pages 1–55, 2009.

[118] Hideaki Uchiyama and Eric Marchand. Toward augmenting everything: Detecting and tracking geometrical features on planar objects. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 17–25. IEEE, 2011.

[119] Ives Rey-Otero, Mauricio Delbracio, and Jean-Michel Morel. Comparing feature detectors: A bias in the repeatability criteria, and how to correct it. *arXiv preprint arXiv:1409.2465*, 2014.

[120] Pablo Alcantarilla, Adrien Bartoli, and Andrew Davison. Kaze features. pages 214–227, 2012.

[121] Leo Breiman. Random forests. *Machine learning*, pages 5–32, 2001.

[122] Frank Moosmann, Bill Triggs, Frederic Jurie, et al. Fast discriminative visual codebooks using randomized clustering forests. In *(NIPS)*, page 4, 2006.

[123] Gang Yu, Junsong Yuan, and Zicheng Liu. Unsupervised random forest indexing for fast action search. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 865–872. IEEE, 2011.

[124] Hao Fu, Qian Zhang, and Guoping Qiu. Random forest for image annotation. In *European Conference on Computer Vision*, pages 86–99. Springer, 2012.

[125] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-Society for Industrial and Applied Mathematics(SIAM) symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[126] Salvador Roura. An improved master theorem for divide-and-conquer recurrences. In *International Colloquium on Automata, Languages, and Programming*, pages 449–459. Springer, 1997.

[127] Tomohiro Nakai, Koichi Kise, and Masakazu Iwamura. Camera based document image retrieval with more time and memory efficient llah. *Proc. Camera Based Document Analysis and Recognition (CBDAR)*, pages 21–28, 2007.

[128] Deeptendu Bikash Dhar and Bhabatosh Chanda. Extraction and recognition of geographical features from paper maps. *International Journal of Document Analysis and Recognition (IJDAR)*, pages 232–245, 2006.

[129] Karl Tombre, Salvatore Tabbone, Loïc Pélissier, Bart Lamiroy, and Philippe Dosch. Text/graphics separation revisited. In *Proceedings of the 5th International Workshop on Document Analysis Systems*, DAS 2002, pages 200–211. Springer, 2002.

[130] Winfried Hohn. Detecting arbitrarily oriented text labels in early maps. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 424–432. Springer, 2013.

[131] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, pages 62–66, 1979.

[132] Pierre D Wellner. Adaptive thresholding for the digitaldesk. *Xerox, Electronic Pre Collation (EPC) 1993-110*, pages 1–19, 1993.

[133] G. Agam, S. Argamon, O. Frieder, D. Grossman, and D. Lewis. *The Complex Document Image Processing (CDIP) Test Collection Project*. Illinois Institute of Technology, 2006. URL http://ir.iit.edu/projects/CDIP.html.

[134] *The Legacy Tobacco Document Library (LTDL)*. University of California, San Francisco, 2007. URL http://legacy.library.ucsf.edu/.

[135] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, pages 381–395, 1981.

[136] Ricardo Eugenio Gonzalez Valenzuela, William Robson Schwartz, and Helio Pedrini. Dimensionality reduction through pca over sift and surf descriptors. In *Cybernetic Intelligent Systems (CIS), 2012 IEEE 11th International Conference on*, pages 58–63. IEEE, 2012.

[137] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe lsh: efficient indexing for high-dimensional similarity search. In *Proceedings of the 33rd international conference on Very large data bases*, pages 950–961. VLDB Endowment, 2007.

[138] Andrew W Fitzgibbon, Robert B Fisher, et al. A buyer's guide to conic fitting. *DAI Research paper*, 1996.

[139] Pablo Ricaurte, Carmen Chilán, Cristhian A Aguilera-Carrasco, Boris X Vintimilla, and Angel D Sappa. Feature point descriptors: Infrared and visible spectra. *Sensors*, pages 3690–3701, 2014.