Graph representation and mining applied in comic images retrieval

Thanh Nam Le

LA ROCHELLE UNIVERSITÉ

ÉCOLE DOCTORALE EUCLIDE

Laboratoire Informatique, Image et Interaction (L3i)

THÈSE présentée par :

Thanh Nam LE

soutenue le : **29 mars 2019**
pour obtenir le grade de : **Docteur de l'Université de La Rochelle**
Discipline : **Informatique et Applications**

Graph Representation and Mining Applied in Comic Images Retrieval

[Représentation par graphes et Fouille de graphes : Application à la recherche d'images de bandes dessinées par le contenu]

COMPOSITION DU JURY :

Sébastien ADAM	Professeur, Université de Rouen Normandie (France), Examinateur
Jean-Christophe BURIE	Professeur, Université de La Rochelle (France), Encadrant de thèse
Josep LLADÓS	Professeur, Université Autonome de Barcelone (Espagne), Rapporteur
Muhammad Muzzamil LUQMAN	Ph.D, Ingénieur de Recherche, Université de La Rochelle (France), Encadrant de thèse
Jean-Marc OGIER	Professeur, Université de La Rochelle (France), Directeur de thèse
Nicole VINCENT	Professeur, Université Paris Descartes (France), Rapporteur

Director	**Prof. Jean-Marc Ogier** Laboratoire Informatique, Image et Interaction (L3i) Université de La Rochelle (France)
Co-Directors	**Prof. Jean-Christophe Burie** Laboratoire Informatique, Image et Interaction (L3i) Université de La Rochelle (France) **Dr. Muhammad Muzzamil Luqman** Laboratoire Informatique, Image et Interaction (L3i) Université de La Rochelle (France)
Reporters	**Prof. Nicole Vincent** Laboratoire d'Informatique Paris Descartes (LIPADE) Université Paris-Descarte (France) **Prof. Josep Lladós** Centre de Visió per Computador (CVC) Universitat Autònoma de Barcelona (Spain)
Examiner	**Prof. Sébastien Adam** Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes (LITIS) Université de Rouen Normandie (France)

This document was typeset by the author using LaTeX 2ε.

The research described in this book was carried out at the Laboratoire Informatique, Image et Interaction (L3i) from the Universtité de La Rochelle, and at the Centre de Visió per Computador (CVC) from the Universitat Autònoma de Barcelona.

Acknowledgements

It would not have been possible to achieve this thesis and write this manuscript without the help and support of the kind people that helped me through such a long and arduous journey.

I would like to first express my gratitude to my three supervisors, for their company over the past years of Ph.D program, as I worked my way from writing the very first initial proposal to a complete study.

Thank you Professor Jean-Marc Ogier, who have given me the chance to participate in this interesting research project related to graph and comics. Such combination was really new to me at the beginning, when we first met in Hanoi. Later on, you were much occupied with the new demanding role of presidency, yet you always found time to help with improving the quality of my work, be it a paper, or a thesis manuscript, or a general advice.

Thank you Professor Jean-Christophe Burie, who helped me throughout my research work, giving me the valuable suggestions, and guided me in completing this thesis, especially in the final phases. Thank you for your great help in extending the funding for my research. I appreciate your meticulousness in revising all the writing, and how you helped me with all the administrative paperwork.

Thank you Muzzamil, for all detailed guidance and encouragements, for the countless hours of exchange we had together, which I'll never forget. You were not only a tutor, but also a good friend indeed. I thank you for your help, not only concerning research and scientific aspects, but also on other matters of life during my Ph.D. Yes, it is not exaggerated: your continuous encouragement and patience are much appreciated, as they pushed me to go through the crucial and rough stages of my thesis.

I'm also thankful to all the members of my PhD jury. In particular, I thank the reviewers, Professor Nicole Vincent and Professor Josep Llados for reviewing my thesis manuscript and providing their valuable remarks, and I thank Professor Sebastien Adam for spending his valuable time as my thesis examiner.

During my time working on this thesis, I had the great fortune of being able to travel to other institutions and meet excellent research teams there. It helped prevented isolated lab time of doing research in an echo chamber. Going to other places gave me the chance to get to know new minds, new ideas, new people and new culture. Thank you again Josep for receiving me as a visiting PhD student and for the kind guidance during my three months at CVC, UAB Barcelona. You were always willing to help and give best suggestions. Working with you was a great pleasure. I

would also like to thank Dr. Motoi Iwata and Professor Koichi Kise for receiving me twice at Osaka Prefecture University and for the great experience I had in Japan.

Thank you to all my friends, colleagues, and all the staffs at L3i for your support and for the years that I passed among you. I prefer to not mention names, since the list would be very long, and I don't want to miss anyone. You all were very kind to me, and I always felt so comfortable working in the lab, thanks to the hospitality the people here, despite a minor language barrier.

My parents and my wife deserve my deepest appreciation, for their love, encouragement, understanding, patience and support. They help me successfully complete this important phase of my life. And thank you my little son, for always being my unlimited source of joy and hope in life. This work is dedicated to you, Phillipe Gia-Minh.

Abstract

Graphs are powerful mathematical modeling tools used in various fields of Computer Science, and in Pattern Recognition in particular. In information retrieval tasks from image databases where content representation is based on graphs, the evaluation of similarity is based both on the appearance of spatial entities and on their mutual relationships. In this thesis we present a novel scheme of Attributed Relational Adjacency Graphs representation and mining, which has been applied in content-based retrieval of comic images. We first address the problem of graph representation of image and its applications in Pattern Recognition, with a focus on content-based image retrieval (CBIR) applications. The images used in this thesis are comics images, which have their inherent difficulties in applying content-based retrieval, such as their abstractness, partial occlusion, scale change and shape deformation due to viewpoint changes. We propose a graph representation that yields stable graphs and allow to retain high-level and structural information of objects of interest in comic images. Next, we extend the indexing and matching problem to graph structures representing the comic image, and apply it to the problem of retrieval. The graphs in the graph database representing the whole comic volume are then mined for frequent patterns (or frequent substructures). This step is to overcome the non-repeatability problem caused by the unavoidable errors introduced into the graph structure during the graph construction stage, which ultimately create a semantic gap between the graph and the content of the comic image. Finally, we demonstrate the effectiveness of the system with a database of annotated comic images. Experiments of performance measures is addressed to evaluate the performance of this CBIR system.

Keywords: *graph matching, graph based modeling, graph mining, frequent pattern mining, pattern recognition, deformable object recognition, content-based image retrieval, comic image retrieval*

Résumé

Les graphes sont de puissants outils de modélisation mathématique utilisés dans divers domaines de l'informatique, et en particulier en reconnaissance des formes. Dans une tâche de recherche d'informations dans de grandes bases de données images où la représentation du contenu est basée sur des graphes, l'évaluation de la similarité est basée à la fois sur l'apparence des entités spatiales et leurs relations mutuelles. Dans cette thèse, nous présentons un nouveau schéma de représentation et d'exploration de graphes d'adjacence relationnels attribués. Ce schéma a été appliqué pour rechercher des personnages dans les images de bandes dessinées.

Nous abordons d'abord le problème de la représentation sous forme d'un graphe de l'image et de ses applications en reconnaissance des formes, en mettant l'accent sur les applications de recherche d'images basées sur le contenu (CBIR). Les images utilisées dans cette thèse sont des images de bandes dessinées, qui possèdent des spécificités qui sont des freins pour les méthodes de recherche d'information par le contenu utilisées dans la littérature.

Le contenu, des bandes dessinées, tel que les objets et les personnages, est complexe. La représentation des personnages, par exemple, peut, d'une case à l'autre, varier énormément en taille et avec différents effets de perspective, selon la situation que l'auteur souhaite retranscrire. Les personnages peuvent ainsi être vus de face, de profil, de dos ou à l'envers; être visible totalement ou partiellement. Cette variabilité représente de réels défis pour les algorithmes de reconnaissance et d'indexation. Nous proposons ainsi une représentation qui permet d'obtenir des graphes stables et qui conserve des informations structurelles de haut niveau pour les objets d'intérêt dans les images de bandes dessinées.

Ensuite, nous étendons le problème d'indexation et d'appariement aux structures de graphes représentant les images d'une bande dessinée et nous l'appliquons au problème de la recherche d'information. Un album de bandes dessinées est ainsi transformé en une base de graphes, chaque graphe correspondant à la description d'une seule case. La stratégie utilisée pour retrouver un objet ou un personnage donné, consiste donc à rechercher des motifs fréquents (ou des sous-structures fréquentes) dans cette base de graphes. Cette étape nécessite de surmonter le problème de non-répétabilité provoqué par les erreurs introduites dans la structure du graphe pendant la phase de construction dues notamment à la variabilité des dessins. Il apparait donc un écart sémantique entre le graphe et le contenu de l'image de bande dessinée.

Nous démontrons l'efficacité du système avec une base de données d'images de bandes dessinées annotées. Des expériences de mesures de performance sont présentées

et permettent d'évaluer la performance de notre système de recherche d'information dans les images de bandes dessinées.

Mots clés: *mise en correspondance de graphes, modélisation par les graphes, fouille de graphes, fouille de motifs fréquents, reconnaissance des formes, reconnaissance d'objets déformables, recherche d'images par le contenu, recherche d'images de bandes dessinées.*

Contents

List of Tables

List of Figures

Chapter 1

General Introduction

This chapter provides a general overview of the thesis, including the thesis context and its particular application to comic content. We start by addressing the origin and the evolution of comic books, as well as their social impacts and market places. We then present the problem of abstract object (such as hand-drawn object) recognition, image retrieval, and particularly a narrower target which this thesis focuses on – comic images retrieval. We point out the necessity and importance of dedicated systems oriented to facilitate the retrieval of this special kind of document images. Graphs, as a powerful tool to represent the comic image content, is briefly introduced too. The motivation, objectives and contributions of this thesis are then described, followed by the organization of the manuscript provided at the end of the Chapter.

1.1 Comics and Society

Comic books are a form of graphic art combining text and images. This art form used to tell stories dates back to a long time ago, however comics as we know today was mainly created since a couple of centuries, and only became especially popular in the latter half of the 20th century. To put officially, the general concept of comics is *"a series of fragmented images, juxtaposed in a deliberate sequence to convey information, aesthetic value"* and into full-fledged stories [McCloud, 1993, Eisner, 1985].

While we can universally refer to this type of media as "comics", it is also known as *bandes dessinées* in Western Europe (particularly in France and Belgium), and *manga* in Japan. Despite having been originally presented as short stories and considered as "children literature" genre due to its nature, over time, comics have gained a great deal of interest from both creators, adult readers, and publishers [Christiansen, 2000]. Nowadays this type of media can be found "in book-length format, allowing long-arced narratives with complex storylines" [Lopes, 2009, p. xvi] spanning over multiple chapters and volumes, and thus can be considered as *graphic novels*. In France, *bandes dessinées* are even considered as "the ninth art" [Screech, 2005].

Comics have worldwide audience and have grown to become one of the most enjoyed and popular storytelling medium. In many countries, they represent an important part of the cultural heritage. People read comics easily and learn many things, so even children can learn about cultures and trends, among other things, through comic books even unconsciously.

1.1.1 Cultural Impact and Market Place

Magazine-style comics emerged as a mass medium in the US in the 1930s, and steadily gained their popularity. According to comprehensive reports with data gathered by [Com, 2016][1], the world's largest public repository of comic-book sales figures, featuring data from the 1930s to today about comic book and graphic novel circulation, and ICv2[2], at the height of their popularity in the 1960s, comic books already reached hundreds of thousands of readers every month. For example, in 1965 DC's Superman alone sold more than 800,000 paid circulation copies per year. In 2014, the top 300 comics reached a combined market size (digital and print copies) of 82 million units sold [Com, 2016]. The rise of graphic novels has truly reshaped the publishing industry.

The overall graphic novel market in the U.S. and Canada has been growing consistently, from $75M in 2000 to $245M in 2005 [Brenner, 2007]. A decade later, total comics and graphic novel sales to consumers in this market reached $1.09B in 2016[3], a 5% growth but representing $55M increase over the figure of 2015 [Com, 2016]. Note well that the rapid growth of movies and TV shows as spin-offs indicates a strong cultural impact and popularity of the characters in comic books' world.

In Europe, *bandes dessinées*, which are mainly created in France and Belgium also proved their popularity. For example, in 2014, there were 80,255 new releases and reissues, representing a 7.3% increase on the previous year, which brought a publishers' turnover of 2,652 million euros. Of the largest segments contributing to that figure, comics were in the fifth largest segment (9.3%) [Report, 2014]. In terms of exporting *bandes dessinées* products, French publishers enjoyed upward trend in the number of licences sold abroad. It is noteworthy that with the overall share of 28.2%, followed by that of children's and young adult books segment (25.3%), comics have proved to be the most successful segment in this exporting business (Spanish was the target language with the largest number of comic translations). Via exportation of cultural products such as books and comics, French still follows English as the second most commonly translated language albeit at a considerable distance [Report, 2014].

In Japan, manga publishing is a vibrant and substantial market, making up 30% of the entire publishing market [Brenner, 2007, p. xi]. According to a more recent

[1] http://www.comichron.com

[2] identified as the industry source on the business of "geek" culture, including comics and graphic novels and hobby games https://icv2.com/

[3] http://www.comichron.com/yearlycomicssales/industrywide/2016-industrywide.html

report, published in 2017 by "The All Japan Magazine and Book Publishers' and Editors' Association" (AJPEA), the sale of manga in Japan accounts for 445.4 billion Japanese Yen (approximately US$4B) in 2016. The market is stable compared to 2015 and 2014, but a large progression of the digital market can be observed as it almost double from 2014 to 2016[4].

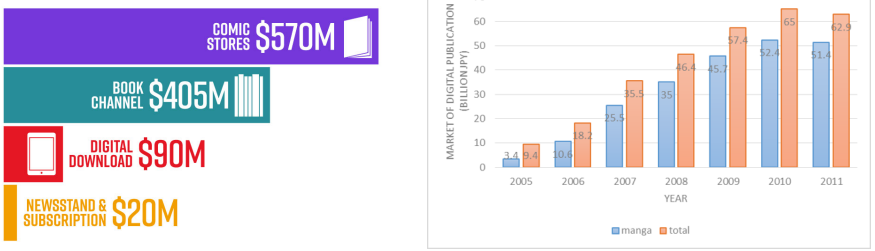
Similar to French culture with *bandes dessinées*, Japanese culture finds its way to the world through manga and *anime* (animation film) [Ingulsrud and Allen, 2009, Bouissou, 2006, Dolle-Weinkauff, 2006]. From the 1950s, postwar Japan started a massive production of Japanese comics, called "manga", bound as small, red books. One of these red book artists was Tezuka Osamu, who was also the great influencer to every manga artists after him, and now heralded as the grandfather of Japanese manga [Brenner, 2007, p. 6]. Although it was only in the 1980s when manga and anime were first reported as making major advances into the world market, the cultural impacts of manga on various countries outside Japan have been generally acknowledged. For example, [Thompson, 2007] illustrated manga's successful story in penetrating American markets by reaching a two-third of sales in graphic novels industry in the 2000s. In 2008, in the U.S. and Canada, the manga market alone was valued at $175M. In Europe and the Middle East the market is worth $250M [Figueiredo, 2018]. The interest toward manga mostly derived from the non-Japanese's perception on manga's distinctiveness compared to American and European comics.

1.1.2 Towards an Enhanced Experience of Comic Consumption

Besides a huge amount of paper-based comic books produced annually, comics nowadays are also commonly found in digitized format and are electronically distributed, thanks to the ubiquity of portable reading devices and the availability of Internet access. For the readers, comics in digital format can be read easily and literally anywhere, and for the publisher the cost of introducing comics to the public is lower. Figure 1.1(a) shows revenue separated by distribution channel in North American market (main market for American comics). While digital comics is still a small channel compared to traditional distribution channels, but in terms of absolute value (and recent growth), they already represent a significant amount. From Figure 1.1(b), we can see that in Japan, manga segment makes the major part regarding the digital publication industry.

Besides these obvious advantages in storage and in distribution, digital format of comics offers a lot more benefits. The opportunities and challenges of digital comics lie in how to take advantage of the added value provided by reading of comics on reading devices. However, we have not yet exploited their full potentials. For example, integrating technology to classic comics would facilitate the exploration of digital libraries, assist translators [Borodo, 2014], augmented reading [Singh et al., 2004], speech playback for the visually impaired [Brandon, 2014], story analysis, and

[4] http://www.ajpea.or.jp/information/20170224/index.html



(a) (b)

Figure 1.1: (a) North American comic sales report 2016, by channel. Source: Comichron Comic 2016 Sales Report (see footnote 3). (b) Market of digital publication over the years (in JPY). Manga segment contributes the major part to the total number. Source: AJPEA (see footnote 4).

so on. Several companies have started to develop their own "added-value" features to improve the reading experience: Marvel with AR (augmented reality)[5], Avecomics[6] (from Aquafadas company) proposes zooming and transition effects to better suit readers who read comics on screens. Sequencity [7], a project from Actialuna, a start-up in Paris, focuses on developing a virtual bookstore environment specially dedicated to comics on tablets, allowing searching by metadata, integrating communication systems with book sellers, and providing personal reviews, recommendations), etc. The connectivity to the Internet makes it also possible to enrich comic books with additional information from the web, allowing the reader to get extra content relating to the comic book.

Nevertheless, in changing the medium of comics, the process of conversion and adaptation is not that straightforward. For example, considering the layout of the page, its size and its arrangement of panels and speech balloons matter to the author and readers. As opposed to books or movies, changing the presentation of a comic book image might very well change its original artistic dimension, resulting in a potential betrayal of the author's intention, or worse, a totally incorrect path of panel navigation. Dealing with this special kind of document image introduced a lot of specific problems related to its characteristics. It has pulled a significant amount of research interest in enriching the reading experience of this specific format of comics. Figure 1.2 shows an interactive system as an example of such enriched reading experience. We will revisit state-of-the-art research about comics in Computer Science in Chapter 3 and discuss the open problems.

[5] http://marvel.com/ar
[6] https://www.avecomics.com/fr_fr/
[7] https://www.sequencity.leclerc/en-FR

(a) (b)

Figure 1.2: Conceptual illustration showing a few examples of capability of an enhanced reading experience could be like. (a) Eye tracking and instant translation. (b) Character gallery, augmented media (extra sounds added), browse by characters, provide characters' side information, etc.

1.2 Content-Based Image Retrieval

Efficient image browsing, searching, and retrieval tools are required by users from various domains. Just as all other types of document, for comics in digital format, a frequent and crucial task consists in retrieving specific content. In the context where the supporting metadata are far from adequate, when we mention "retrieving," we focus on content-based retrieval. Searching through large collections for a specific scene or character is not easy, as the reader has to perform exhaustive searches, with clues based only on his memory and impression of the content. Larger archives, i.e. the content owned by publishers, retailers, or online reading sites, have always been growing and became massive collections [Matsui et al., 2015]. As mentioned in the above Section 1.1.2, to create an enhanced reading environment, the capability of searching and browsing comics should be as powerful as possible, yet for now the content-based retrieval support is still rather inadequate.

Content-based Image Retrieval, commonly referred to in short as CBIR, is the automatic retrieval of digital images from image databases using their visual content. Given an input query image and an image database, the goal of a CBIR system is to find in the image database the most relevant images to the query one.

The capability to quickly recognize an object from a large visual memory is one of the most noteworthy features of human visual system. What is truly outstanding is that it can deal with objects that are represented in a highly abstract way, or objects that are only partly similar to the objects in memory. With a little effort, we can intuitively classify and recognize a large variety of objects, such as the face of

a person, a category of an animal, a shape of a character, a road sign, a car in the street, an image, etc., even when they are in significant appearance alterations due to occlusion, deformation, changes in pose, viewing angle, or lighting condition. The aim of pattern recognition as a field of computer science is to propose algorithms that can imitate the capability of human perception and recognition process.

The challenges in developing CBIR systems lie in the inherent difficulties of the object recognition problem. Besides, recognizing an object of interest does not only involve the object itself, but also other factors such as illumination, relative position, cluttered background in the target image. General robust object recognition is still one of the fundamental challenges for current artificial vision systems. The main reasons do not seem to result from a lack of research in this domain: object recognition has always been a central topic in computer science, and has been studied for more than five decades, resulting in a vast amount of research on the subject [Andreopoulos and Tsotsos, 2013]. Despite the recent advances in applying Deep Learning to the recognition task which yielded highly impressive and unprecedented results, the mechanism that allow us to naturally and efficiently recognize distorted objects is not fully understood yet. But it is generally agreed upon that the brain does not directly compare the interested object with all the objects in the visual memory, it instead *abstracts* the information so as to only retrieve the objects which have the most meaningful features matched.

Depending on how to represent an object, i.e. using global or local features, a recognition system can be broadly classified as a global or a local approach. Global methods extract features on the whole object at once, while local methods extract features on each local primitive (i.e. interest points, edges, regions, etc.). Global approaches have the advantage of requiring less computation compared to the local ones. They however, may likely fail in many practical applications due to partial occlusion or clutter. Local methods, based on the extraction and representation of local features, overcome the global ones. Probably one of the most widely used local methods is Lowe's SIFT algorithm [Lowe, 1999], which is invariant to translation, scale and rotation, and robust to affine transformations. A SIFT (Scale-Invariant Feature Transform) descriptor represents a histogram of local gradient information around extrema in a pyramid of Difference of Gaussian (DoG) images (i.e. in scale space).

Traditional CBIR systems use low level features like color, texture, shape and spatial location of objects to index and retrieve images from databases. Recent methods in CBIR usually rely on the Bag-of-Visual-Words (BoVW) model. The idea, borrowed from the Bag-of-Words (BoW) in text document processing, is to build a visual codebook from all the feature points in a training image dataset. Each image is then represented by a signature, which is a histogram of quantized visual features-words from the codebook. Image features are considered as independent and orderless. The traditional BoW model does not embed spatial layout of local features in the image signature.

In recent years, a paradigm shift has changed the focus of CBIR research from generic CBIR towards application-oriented, domain-specific technique that would

(a) Query Sample

(b) To find the character in these panels

(c) Expected result

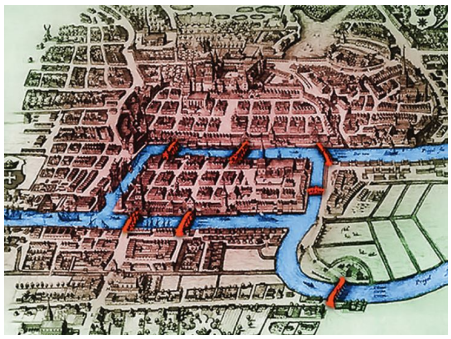
Figure 1.3: An example of object recognition. The interested object here is a comic character (a), which will be search in the panels in the corresponding comic titles, two of the panels containing him are shown here.

have greater impact. CBIR specifically designed for comics is one such domain of content-based image retrieval, where the retrieval of specific comical images from large volumes of diverse drawing styles has great potential applications in entertaining, publishing, archiving, and research. For example, Figure 1.3 illustrates detecting the query character in two panels and the corresponding result provided by a dedicated CBIR system.

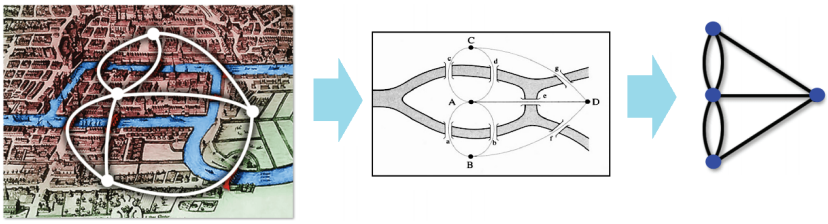
Note that when referring to the object detection problem, there is an ambiguity that needs to be clarified. Object detection may implicitly include object localization, i.e. finding the object's position and size. However, object detection may also refer to answer the binary decision of whether or not an object instance is present in the retrieved image. In this thesis we consider the detection problem as the latter meaning.

1.3 Graph As A Representation Tool

Graph theory is an important mathematical field. It almost certainly began in 18th century when the Swiss mathematician Leonard Euler solved the problem of the Seven Bridges of Königsberg [Alexanderson, 2006]. The city of Königsberg in East Prussia (now Kaliningrad) occupied two sides of the Pregel river, including two large islands. The two sides of the city and the islands were connected by seven bridges at various sections of the river. The problem posed was this: could a continuous tour through the city that would cross each of the seven bridges exactly once then come back to the starting point exist. Although it had been long thought to be impossible, Euler was the first to demonstrate mathematically that no such tour was possible. He gave an abstract model of the problem by representing land areas as points and bridges as arcs linking connected pairs of points (Figure 1.4). This puzzle is well-known, and the illustrations of Euler's graph-based representation are often reproduced in popular books on mathematics. The abstract description of the problem was the introduction



(a) The Map of Königsberg



(b) Graph representation of Königsberg bridge problem

Figure 1.4: Graph Representation to the Seven Bridges Problem

to the graph notion, and its solution is often referred to as the first theorem that laid the foundations of graph theory and prefigured the idea of topology.

The question how to represent objects in a formal way has always been the cornerstone in the whole discipline of pattern recognition. There are two major ways to address this problem: the *statistical* approach and the *structural* approach.

In the statistical approach, objects are represented by feature vectors, i.e., an object is formally represented as a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ of n measurements. A lot of systems employ feature vectors as numerical/statistical representations of objects or pattern, as the choice of vectors is often motivated by their ease of handling in vector space. For example, computing the sum, the product, the mean, or the distance of two entities becomes well-defined and can be efficiently computed in vector spaces. The convenience and low computational complexity of algorithms that use feature vectors as their input have resulted in an algorithmic wealth of tools for pattern recognition, document analysis, and related fields.

However, other than statistical methods, the interested objects can also be represented by a structural representation. While representing objects or patterns by feature vectors $\mathbf{x} \in \mathbb{R}^n$ offers a number of useful properties, it is not without shortcomings. First, it sacrifices the spatial relationships of the features. There is no direct possibility to describe relationships that might exist between different parts of an object. Second, it lacks flexibility, as vectors always represent a predefined set of

features, all vectors in a given application have to preserve the same length regardless of the size or complexity of the corresponding objects.

The structural approach, on the other hand, is based on symbolic data structures, such as strings, trees, or graphs, out of which graphs are the most general one. The above-mentioned drawbacks of feature vectors, namely the size constraint and the lacking ability of representing relationships, can be overcome by graph-based representations. In fact, graphs are not only able to describe properties of an object, but also relationships among different parts of the underlying object by means of edges (Figure 1.5). These relationships can be of various nature (spatial, temporal, conceptual, etc.). Moreover, graphs are not constrained to a fixed size, i.e. the number of nodes and edges can be adapted to the size and the complexity of each individual object under consideration.

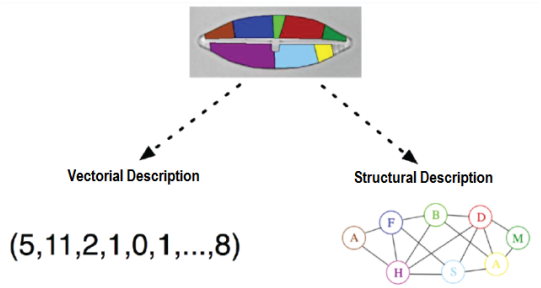


Figure 1.5: Two major approaches for describing objects in image: statistical approach and structural approach.

Indeed, graphs are useful and powerful mathematical abstraction tools to model simple and complex objects in terms of components and their relations. Due to the ability to represent properties of entities and their relations at the same time, graphs have found a trend of widespread applications in document analysis and in pattern recognition over the years. In document analysis area, and in pattern recognition as a broader domain, we see the popularity of attributed graphs. In attributed graphs, the nodes represent local features of the object's components and the edges represent geometric relationships (spatial and temporal relationships) between them.

Thanks to the improvement of computer capacities and prolific research in algorithms concerning graph theory, structural representations have gained popularity in the field of Pattern Recognition over the years. The fact that graphs are naturally well suited to model objects in terms of linked components makes them very attractive for various recognition applications, e.g. general object recognition (e.g., face recognition, street recognition, rigid and non-rigid object recognition), document analysis (e.g. handwriting recognition, sketch recognition, digit and symbols recognition), biometric identification (e.g. fingerprint recognition), video analysis (e.g. gesture recognition) etc. In such domains, the recognition problem is translated into the task of graph matching, or evaluating the similarity between graphs or subgraphs [Bunke, 2000].

One drawback of graphs, when compared to feature vectors, is the significant increase of the complexity of many algorithms. Once the graphs have been constructed, an important question arising in the context of pattern recognition is how to perform efficiently graph matching, which, depending on the specific formulation, can be difficult or considered as NP-hard problem. The main issue of graph matching is the problem of computation time and memory space. This is a serious problem when applying graph matching to pattern recognition, as the computational complexity increases with the size of the graph. For example, the comparison of two feature vectors for identity can be accomplished in linear time with respect to the length of the two vectors. On the other hand, for the analogous operation on general graphs, i.e. testing two graphs for isomorphism (readers can refer to Chapter 2, Section 2.4 for the formal definition of graph isomorphism), while we can avoid the worst scenario in some quasi-isomorphic cases, in general, comparing graphs is relied on exponential algorithms.

In the following Section, we will introduce the research context, i.e. the characteristics of comic image that suggest us to employ the advantage of using structural methods (graph-based representation and matching) for the specific task of content-based comic image retrieval.

1.4 Research Context and Motivation

The popularity of storing and distributing comic books electronically, together with the trend of reading them on various types of devices, has made the task of comics analysis an interesting research problem. However the goal to revolutionize comic reading experience is extremely ambitious, as there are a lot of work to be done to reach that level. In this thesis, we focus on an important task: content-based retrieval.

The research about comics is challenging because of the nature of this medium. Comics contains a mixture of drawings and text. In fact, to fully analyze comics, besides image processing tools, we even need to resort to natural language processing, cognitive science, or the understanding of comics making. A lot of researches have been carried out aiming at understanding the layout structure and the graphic content. However the results are still not yet universally applicable, largely due to the huge variety in expression styles and in the page arrangement. Besides, there is also a lack of annotated databases to do research on the topic. A high level analysis is also necessary to understand events, emotions, storytelling, etc. In a sense, comics research could be related to classic computer vision such as natural image and video analysis, but the availability of drawings with labeled information is much less, compared to natural images.

However there exist methods which do not need extensive training or no training, i.e. matching methods. Such methods usually consist of the following stages: (i) Feature extraction and representation: objects of interest can be represented by either all or a subset of extracted features, (ii) Feature matching: this stage computes distances between two vectors (in statistical approach), or dissimilarity between struc-

tures (graph or trees in case of structural approach), and (iii) Verification: to decide whether the object of interest is present in the retrieved instance (comic panel, or panel region, or comic page) by comparing with a threshold.

One way to support the querying process is to rely on textual information. This text-based approach dates back to the 1970s, where the images are manually annotated by text descriptors, which are then used by some database management system (DBMS) to perform image retrieval. In the world of comics, such kind of extra textual information could be OCR-ed text, or metadata, e.g. summary, annotations, categorization tags, etc. that were manually input. The text-based has certain drawbacks. First, it is not typically suitable for large-scale information input because of the huge amount of human labor required for the annotation process. Second, the text queries do not take image content into consideration. The semantic meaning of a comic image, which the users expect in the search results, may be very different from the text in the metadata which is in turn used for indexing (semantic gap). Third, the accuracy of the annotation process depends on the subjectivity of human perception [Sethi et al., 2001]. Thus, to overcome these disadvantages in text-based retrieval system, a Content-Based Image Retrieval (CBIR) system dedicated for comic images would easily make the searching or browsing experience much more efficient, intuitive, and enjoyable.

Classically, in a CBIR system, visual features are extracted from the images, arranged and stored as a signature. For retrieval, a similarity function is computed to compare the distance between the query's signature to those of the images in the collection. The results are ranked according to the similarity. To improve the retrieval quality, user's interaction with the system, called relevance feedback can be added [Chum et al., 2011]. These techniques are generally considered as a standard setup in the context of searching visually similar images.

Unlike naturalistic image, the characteristics of comic images introduce a lot of challenges associated with content-based retrieval. A full comic page can be considered as a complex document image with various combinations of lines, text, stylistic text, curves and sketch strokes. The great variation in size, shape and pose of the characters, the abstractness in expression, and the severe occlusion problem account for the main difficulties in recognizing comic characters and understanding the drawing content.

In this thesis, we introduce an approximate solution to graph matching problems for the application we are interested in, which is retrieval of comic images, and for that purpose the computation of exact solutions is hard or (generally) impossible for the retrieval. Graph matching and more generally graph comparison is the main operation in the process of pattern recognition using a graph-based approach. Graph matching is the process of finding a correspondence between vertices and edges of two graphs that satisfies a certain number of constraints ensuring that substructures in one graph are mapped to similar substructures in the other.

We tackle the problem by breaking it down to smaller different tasks, as in [Matsui et al., 2015], where the authors addressed several challenges associated with content-

based *manga* (Japanese comics) retrieval, but they can be generalized to comics as well:

- *Description problem.* The visual characteristics of comic images are very different from those of natural images. A typical comic image is usually comprised of black lines on a flat white background, or homogeneous colored regions for color comic pages. Comic images often do not have varying gradient intensities, thus traditional local-feature characterizations may not be suited to describe this kind of images.
- *Retrieval and localization problem.* A comic page comprises several panels (usually bounded boxes, separated by gutters). It is necessary to retrieve not only an image, but also a part of the image. This is a combined problem of retrieval and localization.
- *Efficient querying and indexing problem.* Suppose that a user wants to retrieve instances of a given character (or even more specific, a part of it). We clearly have neither annotations for all specific part, nor man power to realize such a labor-intensive task. While verbose description is still too abstract, we deem that for users, having access to some form of drawing, such as example images (query-by-example or QBE) or sketches of their own, is an efficient method to narrow down the semantic gap in querying.

1.4.1 Objectives

The thesis addresses issues of image representation for object recognition and categorization from images. We try to address the retrieval problem by representing comic images by region adjacency graphs (RAGs) to capture the topological structure of the drawing content, which in turn may yield better retrieval results of comics. The contents of comic images exhibit complex characteristics. It is thus essential that the characteristics of comic images be taken into account when designing a dedicated CBIR system. A successful recognition approach should be able to deal with these difficulties which are commonly found in comic images:

- **Poor texture information.** As opposed to natural images, drawings are poor on texture information or not textured at all, and it is therefore difficult to extract local features of the target object from the scene. This makes it necessary to resort to methods which take into account larger and more abstract parts of the object, instead of local signature of keypoints. Some features like SIFT do not carry enough information in a comic image.
- **Occlusion.** Occlusion problem is inevitable in retrieving drawing content. While global features can be applied in the retrieval process, their capability is only to a limited extent, as they are not designed to handle occlusion problem.
- **Scale variation.** In object recognition from natural images, the scale variation problem has partially been solved by the power of different features, e.g. SIFT-like features, since these features are designed with scale-invariant characteristic

in mind (and they are obtained through processing the image over different scale spaces). In sketch recognition however, the traditional approach of assigning a local scale value to a feature is not possible due to the lack of texture. Thus, an efficient method is needed to cope with these problems.

- **Deformation problem.** Successful approaches must be able to deal with the object deformation problem. Hand-drawn images usually undergo non rigid deformations, thus we cannot model the relation between the query and its potential matches by simply relying on global transformations (rigid, isometry, affine transformation etc.). The deformation problem usually requires a flexible matching mechanism between structural features, e.g. graphs which represent the data as sets of nodes and edges between the nodes.
- **Inter-class and intra-class variations.** Inter-class and intra-class variations are the central issues of any pattern recognition problem, they characterize the difficulty of the problem. In sketch recognition, intra-class variations can even be larger than inter-class variations.

1.5 Contributions

In this section, we present the scientific contributions of this thesis. The use of graph-related techniques in the field of pattern recognition have been studied intensively, but there were no methods dedicated to content-based comic image retrieval. Regarding the popularity of digital comics and its huge potentials described in Section 1.1, we focus on a specific problem: how to construct expressive graphs and how to efficiently retrieve the comic characters by looking for the distinguishable subgraphs.

- First, we decide to focus our interest on how to represent comic images in a CBIR system. We extract the graphical structure and local features of each panel to build Attributed Region Adjacency graphs. One contribution here is the proposal to separate the attributed RAGs into different layers of attributes to solve the non-repeatability from exact matching problem. This will be discussed in-depth in Chapter 4 and 5.
- Inspired from the well-established fact in research that finding frequent patterns plays an essential role in mining associations, correlations, and other interesting relationships among data, we then propose to use the mining of frequent subgraphs (or frequent patterns) from the graphs representing the whole comic database. Frequent patterns are patterns (e.g., set of items in lists, sub-structures, or sub-sequences) that appear frequently in a data set. This contribution emphasizes the importance of frequent, repeated patterns found in the graphs representing the comic images. For the CBIR purpose, we formulate the recognition problem as a graph mining and subgraph matching problem: for a query graph, each graph in the graph repository is determined if there is a match by checking the number of common frequent subgraphs they share, as well as the associated attributes (node and edge attributes).

- We present a system for retrieving comic images in a Query-by-Example (QBE) model. The last contribution of this thesis to CBIR shows how our proposed approach can indeed retrieve a character and how the retrieval can be improved by involving the context of the neighboring comic panels of the retrieved results.

1.6 Thesis Organization

The rest of the thesis is organized as follows:

In Chapter 2, we present related notations, definitions, concepts, and theoretical basis that have been used in this thesis.

In Chapter 3, we present a literature review on the state-of-the-art methods in the field of structural pattern recognition, with a focus on specific object detection and content-based image retrieval techniques, as we consider it to be the fundamental background of this thesis. Those methods concern graph representation of images, exact graph matching, error tolerant graph matching, distance between graphs, graph mining, graph indexing. We also present a review of the state-of-the-art methods for the analysis and retrieval of comic images as well. We end this chapter by summarizing the contributions of our work in light of the limitations of existing methods in applying to comic CBIR.

In Chapter 4, we introduce our approach in content-based retrieval of comic images by graph representation and graph retrieval. It relies on transforming the drawing content of the comic page images into attributed graphs with different features, this allows the integration of different types of local features, as well as the topological relation between parts that compose objects of interest in comics.

In Chapter 5 we present qualitative and quantitative evaluations of our contributions on our own datasets. In this chapter, we also present the SSGCI competition ("Subgraph Spotting in Graph Representations of Comic book Images") and its results [Le et al., 2018], where the effectiveness of using frequent graph mining method is compared with other methods proposed by the participants.

Finally, Chapter 6 concludes the presented work, and introduces some perspectives.

Chapter 2

Definitions and Notations

In this Chapter we present important definitions, concepts, terms and notations that were used throughout this thesis in a formalized way. These include common concepts of graph, subgraph, and attributed graph. This is followed by a section defining important features of graphs and a section introducing concepts on the representation and processing of graphs. We also present the notations frequently used in graph mining to provide background for the techniques presented in rest of the thesis.

2.1 Terminology on Graphs

2.1.1 Graph

From a mathematical point of view, a graph is a collection of points and lines connecting some subset of them. The points of a graph are most commonly known as graph *vertices*, or *nodes*. Similarly, the lines connecting the nodes of a graph are most commonly known as graph *edges*, but are also called *links* or *arcs*.

Formally, we can denote a graph by $G = (V, E)$, where V is the vertex set and E is the edge set:

Definition 2.1. *A graph $G = (V, E)$, where V is a set of vertices, E is a set of edges such that $E \subseteq V \times V$* $\square$

2.1.2 Subgraph

A subgraph G_s is a graph whose set of vertices V_s and set of edges E_s form subsets of the sets V and E of graph G. A subgraph G_s of graph G is said to be induced (or full) if, for any pair of vertices u_i and u_j of Gs, an edge $e\,(u_i, u_j)$ (or e_{ij} in short) connecting u_i and u_j is an edge of G_s if and only if $e\,(u_i, u_j)$ is an edge of G. In other

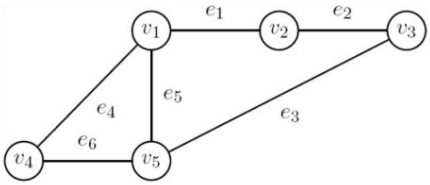


Figure 2.1: Example of a graph with nodes and edges

words, G_s is an induced subgraph of G if it has exactly the edges that appear in G over the same vertices set, i.e., $E_s = E \cap (V_s \times V_s)$

Definition 2.2. *A subgraph G_s of graph G has a set of vertices V_s and a set of edges E_s where $V_s \subseteq V$ and $E_s \subseteq E \cap (V_s \times V_s)$* □

2.1.3 Attributed Graph

Non-attributed graphs are graphs which are only based on their neighborhood structures defined by edges, e.g. molecular graphs, where the structural formula is considered as the representation of a chemical bonds. There are no attributes in the edges or the vertices of the graph.

On the other hand, *attributed*, or *labeled* graphs (AG) can have attributes on edges, vertices, or both of them to represent the objects (for example, in terms of shape, color, coordinate, size, etc.) and the characteristic of their relation.

Definition 2.3. *Let A_V and A_E denote the domains of possible values for attributed nodes and edges. An attributed graph AG over (A_V, A_E) is a four-tuple $AG = \left(V, E, \mu^V, \mu^E\right)$, where: V is the set of nodes; $E \subseteq V \times V$ is the set of edges; $\mu^V : V \to A_V$ is a function assigning attributes to nodes; and $\mu^E : E \to A_E$ is a function assigning attributes to edges* □

2.2 Representation of Graphs

For a finite graph G with n nodes, G can be represented in the form of an $n \times n$ symmetric matrix called *adjacency matrix* $\mathbf{A} = [a_{ij}]_{n \times n}$, where $a_{ij} = 1$ denotes there is an edge between the $i-$th and the $j-$th nodes and $a_{ij} = 0$ otherwise.

A graph $G = (V, E)$ is said to be undirected when each edge e_{ij} of the set E has no direction: $e(u_i, u_j) = e(u_j, u_i)$, i.e., an edge from node u_i to node u_j is not distinguished from an edge from node u_j to node u_i. In contrast, in directed graphs, a direction is assigned to each edge e_{ij}, i.e. in general the two directions are distinct: $e(u_i, u_j) \neq e(u_j, u_i)$.

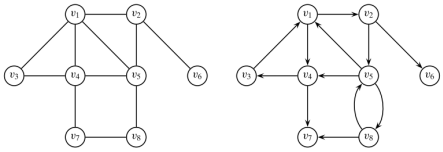


Figure 2.2: Two graphs with the same set of nodes and (almost) the same set of edges. One is an undirected graph (on the left), while the other one (on the right) is a directed graph, resulting in the different corresponding adjacency matrices representing them.

Depending on the application, for each edge, a weight (usually a positive value) may or may not be associated, indicating some certain information about the relationship within the corresponding pair of nodes.

Introducing the weight to each edge in a graph will result in a weighted graph. In this case the adjacency matrix becomes a weight matrix $\mathbf{W} = [a_{ij}]_{n\times n}$ (instead of a binary matrix), with $w_{ij} > 0$ indicating the edge weight between the $i-$th and the $j-$th vertices and $w_{ij} = 0$ indicating no edge there. For an undirected graph, both the adjacency matrix and the weight matrix are symmetric. If the graph is directed, the adjacency matrix $\mathbf{A}$ is not symmetric, as $e(u_i, u_j) \neq e(u_j, u_i)$. Figure 2.2 shows an example of directed and undirected graphs.

2.3 Important Features of Graphs

We list here some important topological attributes of a graph. These attributes are local if they apply to only a single node (or an edge), and global if they refer to the entire graph.

2.3.1 Graph Order

Definition 2.4. *Of the given graph G, **graph order** $|V|$ refers to the number of vertices, and **graph size** $|E|$ refers to the number of edges of G.*

2.3.2 Connectivity and Clique

Definition 2.5. A graph G is connected if there exists a path between any two distinct vertices of G. Otherwise, the graph G is disconnected.

Definition 2.6. A clique in a graph G is a subset S of $V(G)$ such that every two nodes in S are adjacent.

2.3.3 Degree and Neighborhood

The set of all neighbors of a node v in a graph G, is denoted by $N(v)$. The number of neighbors of v is called the degree of v and it is denoted by $deg(v)$.

A node v is an isolated node if $deg(v) = 0$, which means that v is without any neighbors. A node of degree one ($deg(v) = 1$) is called a leaf or a pendant node.

The *minimum degree* of a graph G is $\delta(G) = \min\{\deg(v) : v \in V(G)\}$ and the *maximum degree* of a graph G is denoted by $\Delta(G) = \max\{\deg(v) : v \in V(G)\}$

We have already defined the degree of a node v_i as the number of its neighbors. Given a graph G, which is represented by an adjacency matrix $\mathbf{A}$, a more general definition of node degree $deg(v_i)$ that holds when the graph is weighted is as:

$$deg(v_i) = \sum_j \mathbf{A}(i,j)$$

The degree is a local attribute of each node. One of the simplest global attribute is the average degree:

$$\mu_{\deg} = \frac{1}{n}\sum_{i=1}^{n} \deg(v_i)$$

Shortest Path The shortest path between two vertices in a graph is a path such that the sum of the weights of its constituent edges is minimized.

Distance The *distance* $d(u_i, u_j)$ between two vertices u_i and u_j of a finite graph is the length of the shortest path among the paths connecting them (this is also known as the geodesic distance). If no such path exists (i.e., if the vertices are in different connected components), then the distance is defined as infinite. There may be more than one shortest path between two vertices.

Eccentricity The eccentricity of a node v_i is the maximum distance from v_i to any other node in the graph:

$$\varepsilon(u_i, u_j) = \max_j \{d(v_i, v_j)\}$$

Radius and Diameter The radius of a connected graph, denoted $r(G)$, is the minimum eccentricity of any node in the graph:

$$r(G) = \min_i \{\varepsilon(v_i)\} = \min_i \left\{ \max_j \{d(v_i, v_j)\} \right\}$$

For a disconnected graph, the *diameter* is the maximum eccentricity over all the connected components of the graph. The diameter of a graph G is sensitive to outliers. A more robust notion is *effective diameter*, defined as the minimum number of hops for which a large fraction, e.g. 90%, of all connected pairs of nodes can reach each other.

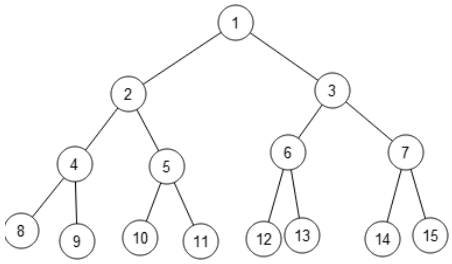


Figure 2.3: Example of graph tree

2.3.4 Path and Cycle

A *path* is in an undirected graph G defined as a sequence of nodes $(v_0, v_1, \ldots, v_k)$ such that each pair (v_i, v_{i+1}) is an edge in $E(G)$ $(i \in (0, 1, \ldots, k-1))$. A path is called *simple* if all its nodes are distinct.

A *cycle* is in an undirected graph G defined as a sequence of nodes $(v_0, v_1, \ldots, v_k)$ such that the set of edges $E(G)$ contains $e(v_i, v_{i+1})$ and the edge $e(v_k, v_0)$, where $i \in (0, 1, \ldots, k-1)$. In other words, a cycle is a closed path starting and finishing with the same node.

Tree: A tree T is a connected graph which do not contain any cycles, i.e. there is only exactly one path connecting any two nodes. An example is given in Figure 2.3. A tree can be rooted or unrooted. A rooted tree is a tree in which one node has been distinguished as the root.

2.3.5 Graph Construction

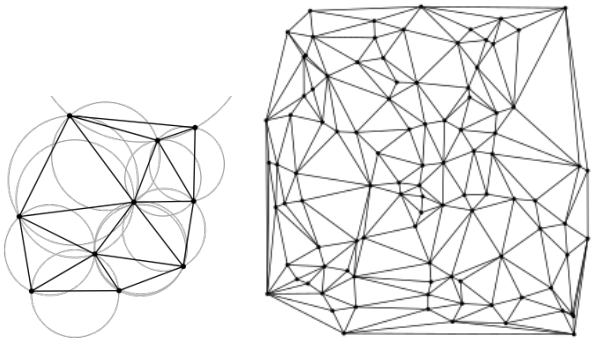
For a given image, generating a corresponding graph which represents the image content generally follows these two steps:

Construct the nodes: The nodes of the graphs often represent one of the following types of information:

- Regions of interest (ROI)
- Points of interest: they represent a rich set of local points that are robust to geometric transformations of the image.

Construct the edges: Once we get the nodes constructed, the edges of the graph are often determined as follow to represent the relationship between the elements which the nodes represent:

- **Proximity:** proximity graphs are constructed by connecting the neighboring nodes based on thresholding techniques (i.e. distances between the nodes).



(a) (b)

Figure 2.4: (a) Grid of triangles in a Delaunay triangulation (DT). (b) The DT of a random set of 100 points in a plane.

- **Adjacency:** graphs constructed this way are usually by segmentation of the original image, where the nodes represent the segmented regions and the edges are formed by adjacency relations between the regions.
- **Fully-connected graph:** a graph in which every pair of nodes is connected by an edge. This kind of graph is sometimes used in inexact matching methods restricted to finding rigid transformations.
- **Delaunay Grid:** (or Delauney triangulation (DT) for a given set P of discrete points in a plane) is particular way of joining P to make a triangular mesh, so that no point in P is inside the circumcircle of any triangle in $DT(P)$, illustrated in Figure 2.4. Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation, and they tend to avoid skinny triangles.Delaunay triangulations are widely used in scientific computing in many diverse applications, as the geometric properties of the Delaunay triangulation make it useful.

2.4 Graph Matching and Graph Isomorphism

Graph isomorphism: the criteria for exact graph matching is that there must be a bijective correspondence (i.e. one-to-one correspondence) between the vertices of the two graphs that preserves edges of both graphs – implying that the numbers of vertices and edges of the two graphs must be the same.

The mapping between the vertices of the two graphs must be edge preserving, i.e. if two vertices in the first graph are linked by an edge, they are mapped to two vertices in the second graph that are linked by an edge as well. This condition must be held in both directions, and the mapping must be bijective. Therefore, a one-to-one

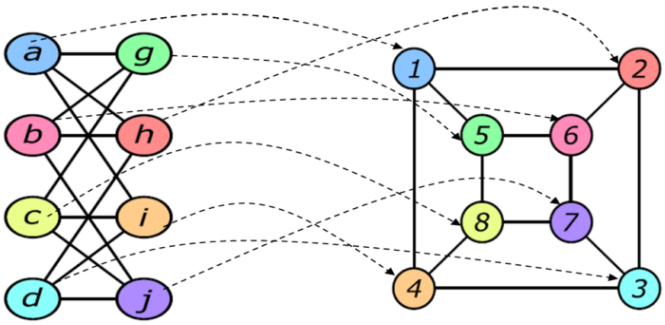


Figure 2.5: Illustration of exact graph matching (node and edge features are not used). There exists an isomorphism mapping f between these two graphs: $f : \{a, b, c, d, g, h, i, j\} \to \{1, 2, 3, 4, 5, 6, 7, 8\}$, $f(a) = 1$, $f(b) = 6$, $f(c) = 8$, $f(d) = 3$, $f(g) = 5$, $f(h) = 2$, $f(i) = 4$, $f(j) = 7$.

correspondence must be found between each vertex of the first graph and each vertex of the second graph. In the case of attributed graphs, attributes have to be identical.

At first glance, the use of adjacency matrices seems rather straightforward: to determine if the two graphs are isomorphic, we can check if their adjacency matrices are equal or if there is some transformation to convert one matrix into the other. In fact it is not that straightforward, because there is not a universal or intuitive matrix transformation to detect isomorphism, as a graph can be represented in many different ways depending on how the nodes (and edges) are enumerated [Washio and Motoda, 2003]. Figure 2.5 illustrates this properties of graph isomorphism. The two visualized graphs shown here look totally different, their corresponding matrices shown below are totally different too, and there is no mathematically defined transformation between them to show the isomorphism:

$$\begin{pmatrix} 0&0&0&0&1&1&1&0\\ 0&0&0&0&1&1&0&1\\ 0&0&0&0&1&0&1&1\\ 0&0&0&0&0&1&1&1\\ 1&1&1&0&0&0&0&0\\ 1&1&0&1&0&0&0&0\\ 1&0&1&1&0&0&0&0\\ 0&1&1&1&0&0&0&0 \end{pmatrix} \to \begin{pmatrix} 0&1&0&1&1&0&0&0\\ 1&0&1&0&0&1&0&0\\ 0&1&0&1&0&0&1&0\\ 1&0&1&0&0&0&0&1\\ 1&0&0&0&0&0&1&1\\ 0&1&0&0&1&0&1&0\\ 0&0&1&0&0&1&0&1\\ 0&0&0&1&1&0&1&0 \end{pmatrix}$$

However, there exists an exact matching between these two graphs, because there is a mapping between the nodes.

Definition 2.7. Let G_1 and G_2 be the two attributed graphs $G_1 = \left(V_1, E_1, \mu_1^{V_1}, \mu_1^{E_1}\right)$ and $G_2 = \left(V_2, E_2, \mu_2^{V_2}, \mu_2^{E_2}\right)$, a bijective function $f : V_1 \to V_2$ which maps each node

$u_i \in V$ to a node $v_k \in V_2$ is a graph isomorphism from G_1 to G_2 if it satisfies these condition:

$$\forall u_i \in V_1, \quad \mu_1(u_i) = \mu_2(f(u_i))$$
$$\forall (u_i, u_j) \in V_1 \times V_1, \quad e(u_i, u_j) \in E_1 \Leftrightarrow \mu_1^{E_1}(f(u_i), f(u_j)) \in E_2$$
$$\forall e(u_i, u_j) \in E_1, \quad \mu_1^{E_1}(e(u_i, u_j)) = \mu_2^{E_2}(e(f(u_i), f(u_j)))$$

Subgraph isomorphism is a form of graph isomorphism, which requires an isomorphism to hold between one of the graphs and a subgraph of the other. In practice, this is useful for searching objects in larger scenes.

Monomorphism: Monomorphism, also known as partial or induced subgraph isomorphism, drops the condition that the mapping should be edge-preserving. It requires that each vertex of the source graph is mapped to a distinct vertex of the target graph, and each edge of the source graph has a corresponding edge in the target graph. However, it allows additional vertices and edges in the target graph.

In practice, most exact graph matching methods are mainly based on the exploitation of the adjacency matrix. This matrix is an $n \times n$ matrix $\mathbf{A} = (a_{ij})$, in which the entry $a_{ij} = 1$ if there is an edge from vertex i to vertex j, and $a_{ij} = 0$ otherwise.

Maximum Common Subgraph (MCS)

Two graphs may share a subpart. That subpart can be as small as a common single node. The largest part of two graphs that is identical in terms of structure, is referred to as the maximum common subgraph.

Maximum Common Subgraph is the problem of mapping a subgraph of the source graph to an isomorphic subgraph of the target graph. Usually, the goal is to find the largest subgraph for which such a mapping exists.

Definition 2.8. *Maximum Common Subgraph (MCS): Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. A graph $G_s = (V_s, E_s)$ is said to be a common subgraph of G_1 and G_2 if there exists subgraph isomorphism from G_s to G_1 and from G_s to G_2. The largest possible graph H (i.e. the graph that has many edges as possible), which is isomorphic to both a subgraph of G_1 and a subgraph of G_2, is called the maximum common subgraph, or MCS, of G_1 and G_2.*□

Graph Edit Distance

Given two graphs, the source graph $g_1 = \left(V_1, E_1, \mu_1^V, \mu_1^E\right)$ and the target graph $g_2 = \left(V_2, E_2, \mu_2^V, \mu_2^E\right)$, Graph Edit Distance (GED) measures the distances between two graphs g_1 and g_2 by the amount of distortion needed to transform g_1 into g_2 using some edit operations. The set of elementary graph edit operations typically includes:

- node insertion to introduce a single new labeled node to a graph

Figure 2.6: Illustration of Graph Edit Distance: one possible editing path to transform g_1 to g_2 by the taking the following operations in order from left to right: $3 \times \{$*edge deletion*$\}$, $1 \times \{$*node deletion*$\}$, $1 \times \{$*node insertion*$\}$, $2 \times \{$*edge insertion*$\}$, and $1 \times \{$*node substitution*$\}$.

- node deletion to remove a single (often disconnected) node from a graph
- node substitution to change the label of a given node
- edge insertion to introduce a new colored edge between a pair of nodes
- edge deletion to remove a single edge between a pair of nodes
- edge substitution to change the label of a given edge

The basic distortion operations of graph edit distance can cope with arbitrary labels on both nodes and edges, as well as with directed or undirected edges. Graph edit distance is therefore one of the most flexible dissimilarity models available for graphs.

Definition 2.9. Graph Edit Distance: Let $g_1 = \left(V_1, E_1, \mu_1^V, \mu_1^E\right)$ be the source and $g_2 = \left(V_2, E_2, \mu_2^V, \mu_2^E\right)$ the target graph. The graph edit distance $d_{\lambda_{\min}}$, or $d_{\lambda_{\min}}$ for short, between g_1 and g_2 is defined by

$$GED(g_1, g_2) = d_{\lambda_{\min}} = \min_{(e_1,\ldots,e_k)\in\mathcal{P}(g_1,g_2)} \sum_{i=1}^{k} c(e_i)$$

where $\mathcal{P}(g_1, g_2)$ denotes the set of all complete edit paths transforming g_1 into (a graph isomorphic to) g_2, $c(e) \geqslant 0$ is the cost of each graph edit operation e, and $\lambda_{\min}$ refers to the edit path with the minimal cost possible in $\mathcal{P}(g_1, g_2)$.

There might be two (or more) edit paths with equal minimal cost, thus the minimal cost edit path $\lambda_{\min}$ is not necessarily unique.

Theoretically, it is possible to extend a complete edit path with any number of additional insertion steps, followed by their corresponding deletion steps. In that way, the size of the set of possible edit paths $\mathcal{P}(g_1, g_2)$ is infinite. However in practice, the cost function c is often with the following constraints so that only a finite number of edit paths have to be evaluated to find which one has the minimum cost edit path among all valid paths:

- Non-negativity, i.e. $c(e) \geqslant 0$ for all edit operation e
- $c(e) > 0$ for all node and edge deletions and insertions e

- Triangle inequality, i.e.

$$c(u \to w) \leqslant c(u \to v) + c(v \to w)$$
$$c(u \to \varepsilon) \leqslant c(u \to v) + c(v \to \varepsilon)$$
$$c(\varepsilon \to v) \leqslant c(\varepsilon \to u) + c(u \to v)$$

Given the above conditions on the edit cost function, it is guaranteed that adding edit operations to an edit path containing operations on nodes or edges, which are neither involved in $g1$ nor in $g2$, will never decrease the overall edit cost of the edit path. Consequently, we only have to consider the $|V_1|$ node deletions, the $|V_2|$ node insertions, and the $|V_1| \times |V_2|$ possible node substitutions to find the minimum cost edit path (as in Definition 2.9) among all possible edit paths $\mathcal{P}(g_1, g_2)$. In other words, the size of the possible set of edit path $\mathcal{P}(g_1, g_2)$ is bounded by a finite number of edit paths.

We will revisit GED in further details in reviewing State-of-the-art in Chapter 3.

2.4.1 Graph Kernel

Definition 2.10. (Graph Kernel) Let G be a set of graphs. Function $\kappa : \Gamma \times \Gamma \to \mathbb{R}$ (where κ denotes the graph domain containing graphs in G) is called a graph kernel if there exists a possibly infinite-dimensional Hilbert space $\mathcal{F}$ and a mapping $\phi : \Gamma \to \mathcal{F}$ such that $\kappa(g, g') = \langle \phi(g), \phi(g') \rangle$ for all $g, g' \in \Gamma$ where $\langle ., . \rangle$ denotes a dot function in $\mathcal{F}$.

In other words, kernels κ can be thought of as a dot product $\langle ., . \rangle$ in some feature space F, and thus, instead of mapping patterns from the original pattern space (graph space κ in this context) to the feature space F and computing their dot product there, one can simply evaluate the value of the kernel function in κ. The following theorem gives a good intuition what kernel functions are [Shawe-Taylor and Cristianini, 2004]:

Theorem 2.4.1. Let $\kappa : X \times X \to \mathbb{R}$ be a valid kernel on a pattern space X, then there exists a possibly infinite-dimensional Hilbert space F and a mapping $\phi : X \to F$ such that $\kappa(x_i, x_j) = \langle \phi(x), \phi(x') \rangle$, $\forall x, x' \in X$ where $\langle ., . \rangle$ denotes the dot product in F.

Definition 2.11. (Positive Definite Kernel) Given a pattern domain X , a kernel function $\kappa : X \times X \to \mathbb{R}$ is a symmetric function, *i.e.*, $\kappa(x_i, x_j) = \kappa(x_j, x_i)$, mapping pairs of patterns $(x_i, x_j) \in X$ to real numbers. A kernel function κ is called positive definite if and only if, for all $n \in \mathbb{N}$,

$$\sum_{i,j=1}^{n} c_i c_j \kappa(x_i, x_j) \geqslant 0$$

for all $\{c_1, c_2, \ldots, c_N\} \subseteq \mathbb{R}$, and any choice of n objects $\{x_1, x_2, \ldots, x_N\} \subseteq X$. [Shawe-Taylor and Cristianini, 2004]

2.4.2 Graph Embedding

Definition 2.12. (Graph Embedding) Given a graph domain G, if $T = \{g_1, g_2, \ldots, g_N\} \subseteq G$ is a set with N graphs and $P = \{p_1, p_2, \ldots, p_n\} \subseteq T$ is a *prototype set* with $n \leqslant N$ graphs, the mapping $\Phi_n^P : G \to \mathbb{R}^n$ is defined as the function:

$$\Phi_n^P(g) = (d(g, p_1), \ldots, d(g, p_n))$$

where $d : G \times G \to \mathbb{R}$ is an appropriately defined graph edit distance.

By means of this definition we obtain a vector space where each axis is associated with a prototype graph $p_i \in P$ and the coordinate values of an embedded graph g are the distances of g to the elements in P. In this way, we can transform any graph g from the set T, as well as any other graph from G, into a vector of real numbers. The graphs which are selected as prototypes before, have a zero distance in conversion.

2.5 Graph Mining

The objective of Frequent Subgraph Mining (FSM) is to find, from a given set of graphs, all the frequent subgraphs, whose occurrence counts or relative frequency of appearance (in percentage) exceed a specified value called .

The occurrence count for a subgraph is usually referred to as its *support*, and the threshold can be referred to as the *support threshold*. A subgraph g is considered to be frequent if it appear a number of times greater than some predefined *minimum support* σ threshold.

The support of g may be computed using either *transaction-based* counting or *occurrence-based* counting. Transaction-based counting is only applicable to graph transaction based FSM. While occurrence-based counting may be applied to either transaction based FSM or single graph based FSM. It is however typically used with single graph based FSM, for example, to identify and count repeated patterns in a large single graph (i.e. a graph representing relationships in a social network).

In transaction-based counting the support is defined by the number of graph transactions that g occurs in, one count per transaction regardless of whether g occurs once or more than once in a particular graph transaction.

Definition 2.13. Given a database $D = \{G_1, G_2, ..., G_N\}$ consisting of a collection of N graph transactions, and a support threshold σ $(0 < \sigma \leq 1)$. The set of graph transactions where a subgraph g occurs is then defined by $\Omega_{\mathbf{D}}(g) = \{G_i \subset \mathbf{D} \,|g \subseteq G_i\}$, and the support of g is defined by:

$$\sup_{\mathbf{D}}(g) = \frac{|\Omega_{\mathbf{D}}(g)|}{N}$$

where $|\Omega_{\mathbf{D}}(g)|$ denotes the cardinality of $\Omega_{\mathbf{D}}(g)$ and N is the number of graphs (transactions) in $\mathbf{D}$. A subgraph g is therefore said to be frequent if and only if $\sup_{\mathbf{D}}(g) \geq \sigma$. The threshold σ can be given in percentage value as well.

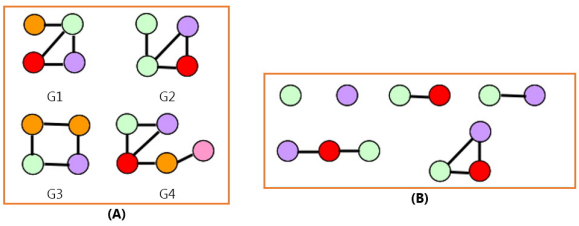


Figure 2.7: A simplified example of mining frequent subgraphs in graph data set: (A) dataset containing 4 graphs, (B) frequent subgraphs which occur at least 3 times ($minSup_{\text{abs}} = 3$ or $minSup_{\%} = 75\%$)

Figure 2.7 shows a simplified example of the frequent subgraphs.

2.6 Conclusion

In this chapter, we have provided the notations and definitions for the terms relating to graphs as well as operations on graphs that are used in the thesis. The next chapter will present a state-of-the-art of the methods based on graphs for pattern recognition.

Chapter 3

State-of-the-art

In this Chapter, we first review the existing work in the literature concerning graph-based methods applied in pattern recognition, including graph representation, graph matching, graph mining, and graph indexing. Since graph-based methods have a long history and a huge number of matching algorithms in a wide variety of application in Computer Science, an exhaustive literature review of the algorithms is not possible. We intend to only provide the appropriate context and background needed for the approaches addressed in this thesis.

We then review the Content-based Image Retrieval (CBIR) systems, which is also a well-studied area in Computer Science. We review the widely used methods, including their used features and the performance, and since this thesis address the application of structural method (graph-based method), we present the work related to CBIR in group of non-structural methods and structural methods to highlight the difference in those approaches.

Finally we present a (non-exhaustive) survey on the research dedicated to comic images in computer science including comic analysis, understanding, and retrieval. Besides, we also focus on recent work, which has not been discussed in previous surveys. By organizing the chapter in this order, we formulate a clear flow of literature review to highlight the strength and flexibility of graphs as a tool for the CBIR task, and show how using graph tools for CBIR can be a contribution to the current research on challenging type of image such as comics.

3.1 Graph-based Methods in Pattern Recognition

Graph, among different representation forms, is a commonly used way to represent data. Generally, graphs are adopted in application domains where the relations among data are worth taking into consideration. In such domains, graphs are used to represent objects in terms of nodes and edges, and the recognition problem often turns

into the task of graph matching, i.e. searching a transformation of one graph into another. For example, they were successfully applied in chemical components analysis [Ralaivola et al., 2005], semi-structured data retrieval [Schenker et al., 2004], etc. One major advantage gained by graph-based models is the presence of structural information embedded in the graphs. Another advantage is that, once the data are well represented by graphs, a specific task in Pattern Recognition may benefit from the well-studied and prolific algorithms for graph manipulation and graph analysis in the literature.

Beginning the late 1970s, the use of graphs in Pattern Recognition gained popularity and obtained a growing attention from the scientific community ever since. This is due to the technological advancement, offering new computer generations with high computational power, allowing the use of graph-based algorithms which have high complexity in most of the cases.

Graphs have been successfully applied in the domain of Computer Vision and Pattern Recognition [Conte et al., 2004, Torresani et al., 2008], as well as in various fields of computer science, since they provide a universal modeling tool which allows the description of structured data. The prolific research work in this field clearly indicate that, object or image representation by means of graphs has a number of advantages over feature vectors. For example, graphs were successfully used in shape retrieval [Huet et al., 1999], object recognition [Kubicka et al., 1990] or face recognition [Wiskott et al., 1997].

In a graph-based representation, nodes and their attributes describe objects (or part of objects) while edges represent interrelationships between the objects, thus one common approach is the partitioning of the image into disjoint regions which can be seen as a graph. The local and spatial features are respectively nodes and edges: local features describe intrinsic properties of regions (such as shape, colors, texture), while spatial features provide topological information about neighborhood.

Graph based techniques for Pattern Recognition aim to solve mainly two major problems. The first one is to find an optimal way to represent objects by graphs. The second problem is to find the appropriate method to compare and/or classify the objects represented by graphs. As the staring point of applying the mathematical concept of graph to recognition problems, graph construction aims at encoding the topological and geometrical information as complete and accurate as possible. In the following pages, we first present efficient and noteworthy graph matching algorithms.

3.1.1 Graph Matching

In Pattern Recognition or CBIR applications, there is usually a fundamental requirement to compare objects. As objects are represented by graphs, methods to compare graphs, i.e. methods to do graph matching (and in general, graph comparison), have become of first interest.

One way to compare graphs is to pairwise compare of nodes or edges, which is possible in quadratic time, but this approach neglects the structural information.

Therefore, graph matching is generally considered the process of finding a correspondence between vertices and edges of two graphs that satisfies a certain number of constraints, ensuring that substructures in one graph are mapped to similar substructures in the other. This approach is called structural in the sense of using the structure of the patterns to compare them.

To initiate the graph matching topic, it is worth mentioning that a comprehensive survey of the technical achievements over the last 30 years is provided in [Conte et al., 2004]. Matching problems are broadly divided into two categories. The first category contains exact matching problems that require a strict correspondence among the two objects being matched. The second category consists of error-tolerant matching problems, where a matching can occur even if the two graphs being compared are structurally different to some extent. We review efficient algorithms for graph matching, including exact and inexact methods.

Exact Graph Matching and Graph Isomorphism

Graph isomorphism or exact graph matching is an edge-preserving mapping from the set of nodes of one graph to that of another graph. Edge-preserving requires that adjacent nodes in the query graph are mapped to adjacent nodes in the target graph. Graph isomorphism represents the most strict form of graph matching, in which the edge-preserving must be satisfied in both directions (in case of directed graphs) and that the mapping is a bijective (one-to-one) correspondence (See Definition 2.7). The output of an exact matching algorithm indicates whether or not two graphs in consideration are isomorphic, or whether a (sub)graph is found in another graph (in the case of subgraph isomorphism). Note that, in this type of problems, node and edge attributes in the query graph and the target graph have to be identical, therefore exact graph matching methods can only be efficiently applied on non-attributed graphs or attributed graphs whose attributes are symbolic.

As mentioned in Section 2.4, the use of adjacency matrices,although straightforward, does not lend itself to isomorphism detection, because a graph can be represented in many different ways depending on how the nodes (and edges) are enumerated [Washio and Motoda, 2003]. With respect to isomorphism testing it is therefore desirable to adopt a consistent labeling strategy that ensures that any two identical graphs are labeled in the same way regardless of the order in which vertexes and edges are presented (i.e., a canonical labeling strategy, a canonical labeling strategy is a labeling algorithm that defines a unique code or sequence for a given graph).

Most of the algorithms for exact graph matching are tree-based search methods with backtracking. Ullmann's algorithm is one of the most popular algorithms used in exact graph matching, despite its age [Ullmann, 1976]. This algorithm targets to find all the isomorphisms between a given graph $G_1 = (V_1, E_1)$ and subgraphs of another graph $G_2 = (V_2, E_2)$ by creating a mapping matrix M with $|V_1|$ rows and $|V_2|$ columns. Each element m_{ij} in M takes value of 0 or 1 : $m_{ij} \in \{0, 1\}$ where $m_{ij} = 1$ means the $i-$th node of G_1 corresponds to the $j-$th node of G_2 so each row contains only one element of value of 1 and each column has no or maximum one element with

value 1. Let $A^{G_1} = \left[a_{ij}^{G_1}\right]$ and $A^{G_2} = \left[a_{ij}^{G_2}\right]$ be adjacency matrices of G_1 and G_2, respectively. The idea of this algorithm based on adjacency matrices is that, if we can find a permutation matrix $\mathbf{M}$ which permutes rows and columns of G_2, resulting in a matrix $\mathbf{C}$:

$$\mathbf{C} = \mathbf{M}\left(\mathbf{M}\mathbf{A}^{G_2}\right)^T$$

and the following expression is true:

$$\left(a_{ij}^{G_1} = 1\right) \Rightarrow \left(c_{ij} = 1\right), \forall i, j \in \{1, 2, .., |V_1|\}$$

then $\mathbf{M}$ specifies an isomorphism between G_1 and G_2, i.e. if $A^{G_1} = \mathbf{C}$ for a certain $\mathbf{M}$, then the two graphs are isomorphic. The brute-force algorithm exhaustively evaluates every possible matrices $\mathbf{M}$.

Tree-based search is also used to find the maximum common subgraph isomorphism between two graphs. In general the maximum common subgraph problem is related to the maximum clique (*i.e.* a fully connected subgraph) one. A typical example is the work of [Bron and Kerbosch, 1973], who use tree search approach to find the maximum clique in an association graph, which represents node-to-node correspondences between two graphs.

More recent algorithms, which are still based on tree search, are the generations of VF algorithms by Cordella *et al.*: VF [Cordella et al., 1999], VF2 [Cordella et al., 2004] and most recently, VF3 [Carletti et al., 2017]. They define a heuristic approach that is based on a depth-first strategy with a set of rules, which significantly prunes the search space. They have shown that this heuristic is faster to compute, and leads to improvement over Ullmann's algorithm. The VF2 algorithm [Cordella et al., 2004] reduces the memory requirement from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ with respect to the number of nodes in the graphs.

Other exact matching algorithms have been proposed besides tree search-based algorithms, [McKay et al., 1981, Bunke and Messmer, 1997, Foggia et al., 2001, Messmer and Bunke, 1999]. Table 3.1 gives a general overview and compares the different approaches of popular exact graph matching algorithms. For example, the *Nauty* algorithm [McKay et al., 1981] is based on the idea of converting the graphs to be matched into a canonical form, and the isomorphism check of two graphs is performed by checking the equality of their canonical forms. This algorithm is considered the fastest isomorphism algorithm available as in a comparison study [Foggia et al., 2001]. However, it deals only with isomorphism problems, and the construction of the canonical form may take exponential time in the worst case scenario.

In many real-world recognition problem, exact matching methods are often inapplicable, due to the complexity of shapes, presence of noises, occlusions, distortions or errors in the stage of transformation of underlying data into the graphs representing them. Moreover, one major drawback of exact graph matching methods is their high computational complexity, which limits the applicability of these approaches in complex applications.

Table 3.1: Different Approaches of notable Graph Matching Algorithms, including Ullmann [Ullmann, 1976], SD [Schmidt and Druffel, 1976], Nauty [McKay et al., 1981], VF [Cordella et al., 1999], VF2 [Cordella et al., 2004], VF3 [Carletti et al., 2017]

Algorithm	Techniques	Matching Types
Ullmann	backtracking & look ahead function	graph and subgraph isomorphism
SD	distance matrix & backtracking	graph isomorphism
Nauty	group theory & canonical labeling	graph isomorphism
VF	DFS strategy & feasibility rules	graph and subgraph isomorphism
VF2	VF's strategy & advanced data structures	graph and subgraph isomorphism
VF3	VF2's strategy & enhanced for large and dense graphs	graph and subgraph isomorphism

Inexact Graph Matching

For graph-based CBIR systems, besides the intrinsic complexity of algorithms to analyze graphs, applying exact graph matching usually results in a severe drawback: *low repeatability*. As an exact matching algorithm will output a sharp yes or no answer, it will reject the match between two graphs in the case of even the slightest discrepancy between them, e.g. a mismatch of a single node, while the two graphs may be "almost similar".

From this perspective, it is essential that the exactness constraint has to be re-

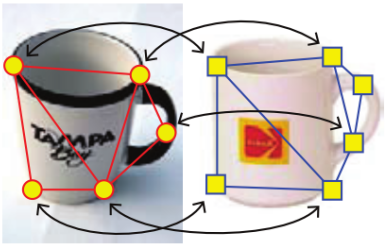


Figure 3.1: Example of matching graphs representing two mugs using inexact graph matching. The target graph and the query one have 6 and 5 nodes, respectively. Inexact matching scheme allows the mapping between corresponding nodes and edges, without requiring an exact similarity in structure.

laxed, as exact graph matching generally cannot return expected results in a CBIR system. Thus the second category in graph matching involves inexact matching methods, where a strict edge-preserving mapping between the two graphs being compared are not required. The term "inexact" means that in these cases no isomorphism is to be expected between both graphs, and the matching problem does not consist in finding the exact way to match nodes and edges of the two graphs being compared, but in finding the best matching possible between them instead.

One solution consists in taking advantage of the available exact graph matching algorithms and customizing the compatibility function for pairing nodes and edges. However, the main drawback of such approaches, is the need to define thresholds for these compatibilities. This leads to the need of a whole new approach known as inexact graph matching. Inexact matching algorithms are created to take errors and noise into account, thus inexact graph matching is also referred to as *error-tolerant* graph matching [Bunke, 1998]. Inexact matching aims at finding a non-bijective correspondence between a data graph and a model graph. They are generally needed when no significant identical part of the structure together with the corresponding node and edge attributes in graphs G_1 and G_2 can be found. Instead of imposing the edge-preservation constraint, the match is associated to a penalty cost [Tsai and Fu, 1979, Messmer and Bunke, 1998]. In other words, the aim of inexact methods is to determine a mapping from one graph to another such that the overall cost of the matching is minimized.

Moreover, these methods are often attributed, as the graph structure itself is generally not sufficient to perform pattern recognition. In real-world applications of pattern matching, node and edge attributes are needed. The attributes can be symbolic (name, function, etc) or numerical (position, size, descriptors) resulting from a feature extraction step often found in pattern analysis. For example, when node and edge attributes are numerical values (scalar or vectorial), the penalty cost for the mapping can then be defined as the sum of the distances between label values. Two nodes and/or edges can be matched (or substituted), even if their attributes are not equal. By that, the matching problem turns from a decision one to an optimization one, as an error-tolerant matching algorithm can overcome the low repeatability drawback.

In general, matching cost can also be based on some sort of edit operations (e.g., node insertion, node deletion, node substitution, etc.), which are called graph edit cost [Sanfeliu and Fu, 1983]. This is an extension of a well known method, called string-edit distance [Levenshtein, 1966]. The idea is to define the operation with smallest cost needed to transform one graph into the other. The $A*$ technique is usually employed to compute graph edit cost [Bunke, 2000]. As in exact matching, tree search with backtracking can also be used for inexact matching. In this case the cost of the current partial matching along with the estimated matching cost for the remaining nodes direct the search process. This estimated total cost is used either to prune search paths in a branch and bound traversal, or to determine the order of branches to be traversed. Tree search based inexact graph matching can be found in [Tsai and Fu, 1983, Wong et al., 1990]. Many other methods for inexact graph

matching based on continuous optimization have been proposed. Examples include the fuzzy graph matching [Perchant and Bloch, 2002], kernel methods [Neuhaus and Bunke, 2007, Shawe-Taylor and Cristianini, 2004], spectral methods etc [Leordeanu and Hebert, 2005].

A different approach to model the uncertainty of structural patterns was proposed in [Wong and You, 1985], where random graphs are defined as a particular type of graphs which convey a probabilistic description of the data. Also based on the construction of a state-space which is then searched with branch-and-bound technique, in [Seong et al., 1994], the authors developed a branch-and-bound algorithm to find the optimal isomorphism between two random graphs in terms of an entropy minimization formulation. These algorithms are regarded as *error-tolerant* algorithms.

Continuous Optimization Methods

Unlike exact or inexact graph matching algorithms, *approximate* (or continuous optimization) matching algorithms offer the advantage to reach a solution in polynomial time and to be applied in exact graph matching problem. Algorithms that fall in the group of approximation methods include [Huet and Hancock, 1999, Christmas et al., 1995, Gold and Rangarajan, 1996, Wilson and Hancock, 1996, Kittler and Hancock, 1989], where the continuous optimization involves some form of probabilistic relaxation labeling.

Probabilistic relaxation labeling is an iterative process, which assign labels to set of objects using contextual constraints. The idea is similar to the discrete relaxation methods [Kitchen, 1978, Wilson and Hancock, 1997], however, the node-to-node assignment is defined in terms of a probability function that is updated by the relaxation procedure, instead of a binary formulation. In these approaches, each node of the first graph is to be assigned to one label out of a discrete set of possible labels that are described by a vector, which holds the estimated probabilities of correspondence to each vertex of the other graph. These methods are iterative: in the initial labeling step, these probabilities are computed based on node attributes, node connectivity,or other information available. During the matching process, these probabilities are refined in an iterative procedure until either the labeling converges, or a maximum number of iterations is reached. However, since the similarity function which they try to minimize can converge in a local minimum, they may not find the globally optimal solution.

Continuous optimization approach in inexact matching can also be based on neural networks: the nodes of a neural network can represent node-to-node mappings and the connection weights between two network represent a measure of the compatibility between the corresponding mappings [Kuner and Ueberreiter, 1988, Suganthan et al., 1995]. However, one problem of neural networks is that the minimization procedure is strongly dependent on the initialization of the network.

Graph Edit Distance-based Methods

The matching cost to transform one graph to another can also be based on some sort of edit operations. Graph Edit Distance (GED) is a dissimilarity measure for graphs that represents the minimum-cost sequence of basic editing operations to transform a graph into another. The basic idea of GED is to evaluate the dissimilarity based on the number as well as the strength of the distortions that have to be applied to transform a source graph into a target graph. GED hence becomes particularly interesting as this dissimilarity measure can be readily turned into a similarity measure. A major application of graph edit distance is in inexact graph matching, such as error-tolerant pattern recognition in machine learning.

Given two graphs, the source graph $g_1 = (V_1, E_1)$ and the target graph $g_2 = (V_2, E_2)$, the basic idea of graph edit distance is to transform g_1 into g_2 using some edit operations: insertion, deletion and substitution of nodes and/or edges. The mathematical definition of graph edit distance is dependent upon the definitions of the graphs over which it is defined, i.e. whether and how the vertices and edges of the graph are labeled and whether the edges are directed, but for a general definition can be referred to Definition 2.9 in Chapter 2.

The graph edit distance between graphs is related to the string edit distance between strings. Originally, the concept of edit distance has been proposed for string representations [Wagner and Fischer, 1974]. If we consider strings as special type of graph (connected, directed acyclic graphs of maximum degree one), classical edit distance (such as Levenshtein distance, Hamming distance and Jaro-Winkler similarity [Deza and Deza, 2009]) may be interpreted as graph edit distances for a specific subclass of graphs. Then mathematically, the concept of graph edit distance was first reported and formalized in [Sanfeliu and Fu, 1983] and [Bunke and Allermann, 1983]. Eventually, the edit distance has been extended from strings to more general data structures such as trees and graphs (see [Gao et al., 2010] for a survey on the development of graph edit distance).

General approaches for graph edit distance-based pattern recognition are briefly reviewed. Note that the structures of the considered graphs do not have to be preserved in any case. Structure distortions are also subject to a cost which is usually dependent on the magnitude of the distortions. So the meta parameters of each of deletion, insertion and substitution affect the matching process. The discussion around the selection of cost functions and their parameters is beyond the topic of this chapter.

Let $\gamma(G_1, G_2)$ denote the set of edit paths that transform G_1 into G_2. To select the most promising edit path among all the edit paths of $\gamma(G_1, G_2)$, a cost, denoted by c_{ed}, is introduced. Thus, for each operation (edge/vertex substitutions, edge/vertex deletions and edge/vertex insertions) a penalty cost is added. GED tries to find the minimum overall cost $(d_{\lambda_{\min}}(G_1, G_2))$ among all generated costs. Formally saying, GED is based on a set of edit operations ed_i where $i = 1..k$ and k is the number of edit operations. This set is referred to as the edit path.

The traditional approach to graph edit distance-based pattern recognition is given by the $k-$nearest-neighbor classification ($k-$NN). In contrast with other classifiers such as artificial neural networks, Bayes classifiers, or decision trees, the underlying pattern space need not be rich in mathematical operations for nearest-neighbor classifiers to be applicable. More formally, in order to use the nearest-neighbor classifier, only a pattern dissimilarity measure must be available. Therefore, the $k-$NN classifier is perfectly suited for the graph domain, where several graph dissimilarity models, but only little mathematical structure, are available.

The $k-$NN classifier proceeds as follows. Assume that for a graph domain G, an appropriate definition of a graph edit distance $d : G \times G \to \mathbb{R}$, a set of labels Ω, and a labeled set of N training graphs $\{(g_i, \omega_i)\}_{1 \leqslant i \leqslant N} \subseteq G \times \Omega$ are given. The $1-$nearest-neighbor classifier ($1-$NN) is defined by assigning an input graph $g \in G$ to the class of its most similar training graph. So, the $1-$NN classifier $f : G \to \Omega$ is defined by $f(g) = \omega_j$, where $j = \underset{1 \leqslant i \leqslant N}{\arg\min}\, d(g, g_i)$.

If $k = 1$, the $k-$NN classifier's decision is based on just one graph from the training set, no matter if this graph is an outlier or a true class representative. So, the decision boundary is largely based on empirical arguments. To render nearest neighbor classification less prone to outlier graphs, it is common to consider not only the single most similar graph from the training set, but evaluate several of the most similar graphs. Formally, if $\left\{\left(g_{(1)}, \omega_{(1)}\right), \ldots, \left(g_{(k)}, \omega_{(k)}\right)\right\} \subseteq \{(g_i, \omega_i)\}_{1 \leqslant i \leqslant N}$ are those k graphs in the training set that have the smallest distance $d(g, g_i)$ to an input graph $g \in G$, the $k-$NN classifier $f(g) = \omega_j$ is defined by

$$f(g) = \underset{\omega \in \Omega}{\arg\max} \left|\left\{\left(g_{(i)}, \omega_{(i)}\right) : \omega_{(i)} = \omega\right\}\right|$$

Nearest-neighbor classifiers provide us with a natural way to classify graphs by means of graph edit distance. However, the major restriction of nearest-neighbor classifiers is that a sufficiently large number of training graphs covering a substantial part of the graph domain must be available.

Kernel Method Kernel methods have become one of the most rapidly emerging subfields in pattern recognition and related areas. A thorough introduction to kernel theory was given in [Shawe-Taylor and Cristianini, 2004]. It can be explained by the nature of kernel methods: kernel theory allows extending basic linear algorithms for pattern recognition (originally developed for vectorial data) to more complex, nonlinear, and structural data such as strings, trees, or graphs in a unified and elegant manner. So, kernel methods can be seen as a theory to bridge the gap between statistical and structural pattern recognition.

The mathematical definition of graph kernel was mentioned in Chapter 2, Section 2.4.1. The key idea of kernel methods is based on an essentially different way of how the underlying data is represented. In the kernel approach, an explicit data representation is of secondary interest. Rather than defining individual representations for each pattern, the data is represented by pairwise comparisons via kernel functions.

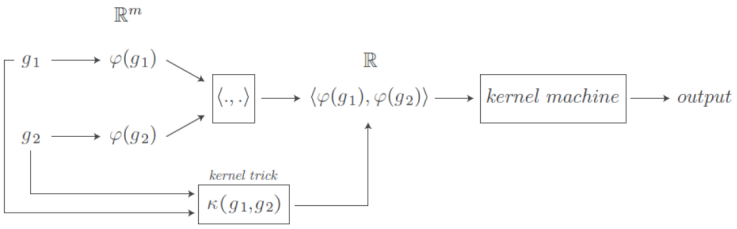


Figure 3.2: Illustration of the kernel trick applied to graphs. Thanks to kernel trick, all kernel machines that have been developed for feature vectors become applicable to graphs.

Standard graph kernels such as convolution kernels, random walk kernels, or diffusion kernels have been substantially extended by means of graph edit distance in [Neuhaus and Bunke, 2007].

A huge amount of important algorithms has been *kernelized*, i.e., entirely reformulated in terms of dot products. Kernelized algorithms are commonly referred to as kernel machines. These algorithms include SVM, nearest-neighbor classifier, perceptron algorithm, principal component analysis, Fisher discriminant analysis, $k-$means clustering, self-organizing map, partial least squares regression, and many others. Any kernel machine can be turned into an alternative algorithm by replacing the dot product $\langle .,. \rangle$ by a valid kernel $\kappa(.,.)$. This procedure is commonly referred to as kernel trick.

The kernel trick is especially interesting for graph-based pattern representation, since a graph kernel value (for instance the transformed GED defined above) can be fed into any kernel machine (e.g., a support vector machine). In other words, the graph kernel approach makes many powerful pattern recognition algorithms instantly applicable to graphs, as illustrated in Figure 3.2.

Spectral Methods

Spectral theory is the theory in which graphs are compared by means of eigenvalues of their adjacency matrix. The idea of spectral graph matching methods is based on the observation that the eigenvalues and eigenvectors of the adjacency matrix representing a graph are invariant to vertex permutations. Thus, if two graphs are isomorphic, their adjacency matrices have the same eigenvalues and eigenvectors, i.e. the same eigendecomposition (but the inverse is not necessarily true). Spectral methods for graph matching have received considerable research interest [Wang and Hancock, 2004, Leordeanu and Hebert, 2005, Cour et al., 2006, Carcassoni and Hancock, 2003], since computation of eigenvalues has polynomial time complexity.

As shown in [Wang and Hancock, 2004, Carcassoni and Hancock, 2003], spectral methods are not robust for matching patterns of very different sizes. Besides, and as pointed out in [Neuhaus and Bunke, 2007], the main problem of spectral methods

is that they are sensitive to structural errors, such as missing or spurious vertices. In an extensive survey, the authors in [Conte et al., 2004] reviewed, discussed and categorized more than 160 papers reporting graph matching algorithms in the context of the Pattern Recognition and Computer Vision during three decades of graph matching. In that survey, the links between the different application areas and the graph-based techniques employed have also been highlighted to provide useful hints when considering the use of graph matching in a particular domain. In that spirit, the authors in [Foggia et al., 2014] continued the survey in [Conte et al., 2004] in 2004 with another survey that covers the period of the following ten years (from 2004 to 2014) concerning graph matching and learning in Pattern Recognition.

Due to the large volume of bibliography sources reviewed, the authors in [Conte et al., 2004] organized their review by presenting two taxonomies. The first one is the taxonomy of matching algorithms, which was presented to compare and discuss different problems and strategies involved. The second one is the taxonomy of the most common applications of graph-based techniques in the pattern recognition field.

For the first taxonomy, the algorithms are divided into two broad categories: the first category contains exact matching methods that require a strict correspondence among the two objects being matched or at least among their subparts. The second category defines inexact matching methods, where a matching can occur even if the two graphs being compared are structurally different to some extent. This first taxonomy (based on algorithms of graph matching) is summarized as in Figure 3.3, and it may help the readers to have a structural view while following the papers reviewed in that paper [Conte et al., 2004], as well as in this section. For the second taxonomy, the authors mentioned the possibility to identify at least six application areas where graph matching techniques have been successfully used: 2D and 3D image analysis, document processing, biometric identification, image databases, video analysis, biological and biomedical applications.

3.1.2 Graph Embedding

In retrieval and recognition tasks, other than graph matching, graph embedding is also a commonly used tool. The motivation of graph embedding is similar to that of the kernel approach, i.e., making the algorithmic tools originally developed for vectorial data applicable to graphs. Yet, in contrast with kernel methods, which provide an implicit graph embedding only, graph embedding techniques result in an explicit vectorial description of the graphs.

The idea of a recent graph embedding [Kaspar and Horst, 2010] is based on the work of [Pkkalska and Duin, 2005]. The key idea of this approach is to use the distances of an input graph to a number of training graphs, termed *prototype graphs*, as a vectorial description of the graph. That is, one makes use of the dissimilarity representation for pattern recognition rather than the original graph-based representation.

The selection of the N prototypes $P = \{p_1, p_2, \ldots, p_N\}$ is a critical issue in the embedding framework. Not only the prototypes p_i themselves but also their number N

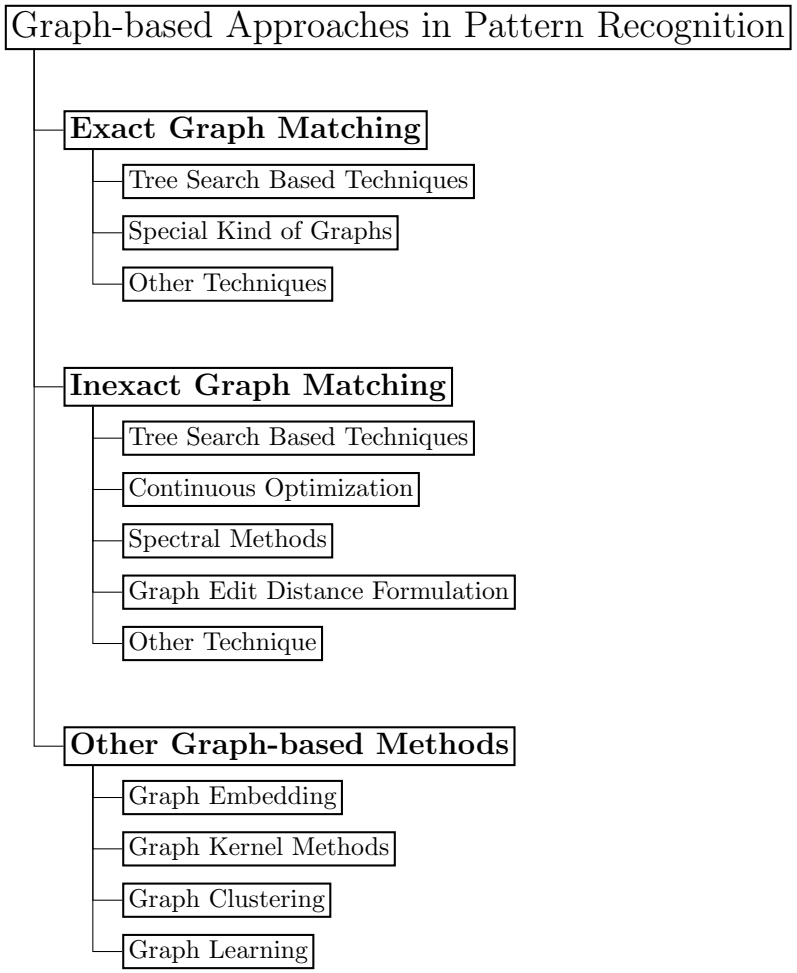


Figure 3.3: Taxonomy of Graph-based approaches in Pattern Recognition based on the taxonomies introduced in [Conte et al., 2004, Foggia et al., 2014]

affects the resulting graph embedding $\Phi_n^P(.)$, and thus the performance of the pattern recognition algorithm in the resulting embedding space. In [Kaspar and Horst, 2010] the selection of prototypes $P = \{p_1, p_2, \ldots, p_N\}$ is addressed by various procedures. A number of prototype selection methods have also been introduced in [Spillmann et al., 2006, Kaspar and Horst, 2010, Bunke and Riesen, 2008]. These prototype selection strategies use some heuristics based on the underlying dissimilarities in the original graph domain. Basically, these approaches select prototypes from T that best possibly reflect the distribution of the graph set T or that cover a predefined region of T. The rationale of this procedure is that capturing distances to significant prototypes from T lead to meaningful dissimilarity vectors.

In [Riesen et al., 2007a] the problem of prototype selection has been reduced to a feature subset selection problem. That is, for graph embedding, all available elements from the complete set T are used as prototypes, i.e., $P = T$. Next, various feature selection strategies [Jain and Zongker, 1997] are applied to the resulting large-scale vectors eliminating redundancies and noise, finding good features, and simultaneously reducing the dimensionality of the graph maps.

A severe shortcoming of prototype selection strategies is that the dimensionality of the embedding space has to be determined by the user. Thus, a prototype selection method that automatically infers the dimensionality of the resulting embedding space has been proposed in [Riesen and Bunke, 2009]. This scheme is adopted from well-known concepts of prototype reduction [Bezdek and Kuncheva, 2001] originally used for the task of condensing training sets in nearest-neighbor classification systems.

3.1.3 Graph Mining

This part provides an overview of graph pattern mining and graph mining methods, where we will focus more on Frequent Subgraph Mining as it serves as an important stage in our system of retrieving comic images represented by graphs.

Other than comparing two graphs as a whole, conventionally there are two ways for measuring similarity between graphs in terms of their components. One approach is to perform a pairwise comparison of the nodes and/or edges in two graphs, and calculate the overall similarity score. This approach takes quadratic time in the number of nodes and edges, thus makes it computationally feasible even for large graphs. However, the shortcoming of this strategy is that it ignores the structure embedded in the graphs by treating them as sets of nodes and edges instead of graphs with topological structure. An alternative, which still takes into consideration graph structure, would be to assert the similarity between two graphs by determining if they share many common substructures, or technically, if they share many common subgraphs. (The term structure should be understood as a general concept that covers many different kinds of structural forms such as directed graphs, undirected graphs, lattices, trees, sequences, sets, single items, or combinations of such structures).

In the latter approach, we however still have to deal with the problem of subgraph listing and subgraph isomorphism. Subgraph isomorphism is about finding subgraphs

in a target graph which are isomorphic to a given graph (query graph). These tasks are not trivial: for example, listing all the subgraphs is equivalent to the problem of listing all the subsets in a set, which increase exponentially, then we still have to perform isomorphism check on each listed subgraph. In general, all the problems mentioned above are $NP-$complete, i.e., the computational cost of this problem increases exponentially with problem size [Garey and Johnson, 1990].

To improve the situation, we can either impose further restrictions on the graphs, or find algorithms which are faster in some cases. The key is to efficiently list the subgraphs without omitting a case and without re-encountering a substructure. Those are the central interest of **Frequent Subgraph Mining** (FSM) or **Graph Pattern Mining** research area.

Graph Pattern Mining algorithms play an important role in further expanding the use of data mining techniques to graph-based datasets. The primary goal of Graph Pattern Mining is to extract statistically significant and useful knowledge from the data represented by the graph database [Han et al., 2011]. Indeed, pattern mining often helps in the discovery of inherent structures in the data, as patterns are considered to carry more information gain than a single attribute (feature) in general. Note that Graph Pattern Mining is a more general term than Frequent Pattern Mining or Frequent Subgraph Mining, since the former may indicate the discovery of rare, specific, or negative patterns as well. However, while the term "mining" may refer to mining frequent subgraph patterns, graph classification, graph clustering, and other analysis tasks, Frequent Subgraph Mining (FSM) is the essence of the "mining" process [Han et al., 2011, Witten et al., 2016], and thus when there is no ambiguity, these terms can be used interchangeably.

Frequent Subgraph Mining (FSM) is defined as finding all the subgraphs in a single graph or a set of graphs, that appear more times than a given value σ. Driven by the huge success of graph based business models, such as social graphs, web graphs etc., Frequent Subgraph Mining has become a major research theme in data mining to generate recurrent structures, and has many interesting applications. It can be used to generate intermediate output of other tasks such as graph classification, graph indexing, graph clustering, etc. [Cook and Holder, 2006]. In general, frequent graph patterns (and techniques to mine them) are of great interest not only for the sole purpose of comparing graphs, but also for the reason that they are useful for characterizing graph sets, clustering graphs, building graph indices, and facilitating similarity search in graph databases. Classification of graphs can also be performed effectively using frequent and discriminative subgraphs as features. In [Yan et al., 2004], the authors showed that a compact and effective graph index can be built based on the concept of frequent and discriminative graph patterns. For example there are numerous classification methods, but it has been shown that frequent patterns can be used as building blocks in the construction of high quality classification models, hence the term *pattern-based classification*.

Overall, the working of a FSM algorithm is as follows: the input to FSM algorithm is a graph dataset and user defined minimum support (*minsup*), the output is the set of frequent subgraphs. It broadly consists of two steps: (1) generating all

the candidate subgraphs from the graph dataset, and (2) calculating support of those candidates. The second step requires repeated (sub)graph isomorphism test between each candidate graphs with all the graphs in the graph dataset. Since graph isomorphism test is a procedure with high complexity where we typically cannot gain much in performance, most studies on FSM focus on the optimization of the candidate generating step [Han et al., 2011]. Obviously, the fewer the number of candidates, the fewer the support computations are required. To be effective in candidate listing, the generation of duplicate or impossible candidates should be avoided.

Graph datasets are available in two types: in the first type, the dataset comprises a number of small graphs called the transactional setting (e.g. biological and chemical datasets), in the second type, the dataset comprises of a single massive graph (e.g. social networks, computer networks, etc.). Single graphs are considered to be more general data model but more expensive because of the repetitions and redundancy among data. Therefore, support of a subgraph refers to its occurrence counting, which can be computed using either *transaction-based* counting or *occurrence-based* counting [Jiang et al., 2013, Han et al., 2011]. In the former, applicable only to the mining of a collection of graphs, the support is defined by the number of graphs (transactions) in the collection in which a frequent subgraph g occurs, no matter how many times it may appear in each graph.

We briefly introduce the current state-of-the-art of FSM, including various methods, their extensions, and applications of frequent subgraphs mining. The straightforward idea behind FSM candidates generating is to "grow" candidate subgraphs, in either a breadth-first-search or depth-first-search manner. If a graph is frequent, then all its subgraphs are frequent too. Although the algorithms can be categorized with regards to: candidate generation strategy, the mechanism for traversing the search space, or the occurrence counting process, but in general a common way to classify is to divide FSM into two basic approaches [Jiang et al., 2013]: (1) ***Apriori*-based approach**, and (2) **pattern growth-based approach**.

Graph pattern mining has been studied extensively, notable work include [Holder et al., 1994], [Inokuchi et al., 2000], [Kuramochi and Karypis, 2001], [Yan et al., 2004, Yan and Han, 2002], [Borgelt and Berthold, 2002], [Huan et al., 2004], and [Nijssen and Kok, 2005]. The algorithms differ in the type of input graphs, the search strategy they use,the method of representation of graphs, etc. Hence, there exist many algorithms based on different approaches. We present a survey and classify these algorithms to help understanding and analyzing various properties and limitations of few of these algorithms.

Apriori Approach

FSM in many aspects can be considered as an extension of Frequent Item set Mining, or Association Rule Mining. As a result, many of the proposed solutions are based on similar techniques found in the domain of Frequent Item set Mining.

Apriori approaches adopt the pruning strategy founded in the work of Agrawal

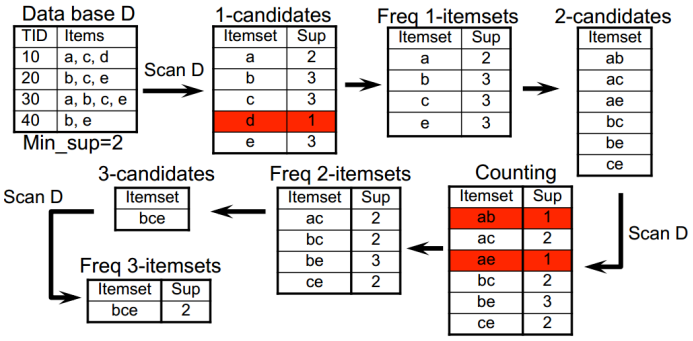


Figure 3.4: Illustration of *A-priori*-based approach in finding frequent pattern [Agrawal et al., 1993]

and Srikant about Association Rule Mining (ARM) [Agrawal et al., 1993]. The *Apriori property* (in the context of data cubes in ARM) states that: *"If a given cell does not satisfy minimum support, then no descendant (i.e., more specialized or detailed version) of the cell will satisfy minimum support either."* Many algorithms in association rule mining, as well as graph mining have adopted this property. The strategy can be used to substantially reduce the size of the search space, by pruning away the exploration of the descendants of a certain candidate. For example, if the count of a subgraph g is less than the minimum support threshold σ, then the count of any descendant of g in the search tree (i.e. the supergraphs of g) can never be greater than or equal to σ and thus can be pruned.

As shown in Algorithm 1, the *Apriori*-based approach proceeds in a generate-and-test manner using a Breadth First Search (BFS) strategy to explore the subgraph lattice of the given database. Therefore, before considering $(k+1)$ subgraphs, this approach has to first consider all k subgraphs. The pattern growth-based adopts a DFS strategy is depicted where, for each discovered subgraph g, the subgraph is extended recursively until all frequent supergraphs of g are discovered [Han et al., 2011]. Figure 3.4 give a simple example as an illustration of finding frequent itemsets by step-by-step expanding the mined, smaller frequent patterns.

Frequent Pattern Growth Approach

The *Apriori*-based approach has to use the BFS strategy because of its level-wise candidate generation. To determine whether a graph of size $(k+1)$ is frequent, it must check all of its corresponding subgraphs of size k to obtain an upper bound of its frequency. Thus, before mining any size $(k+1)$ subgraph, the *Apriori*-like approach usually has to complete the mining of size$k$ subgraphs. In contrast, the pattern-growth approach is more flexible regarding its search method. FSM algorithms following strategy stem from FP-growth algorithm. They can use breadth-first search as well as depth-first search (DFS). While BFS tends to be more efficient in that it allows for the pruning of infrequent subgraphs (at the cost of high memory usage) at an early

Algorithm 1 Apriori-based frequent substructure mining

Input:

D: a graph data set

minsup: the minimum support threshold

Output:

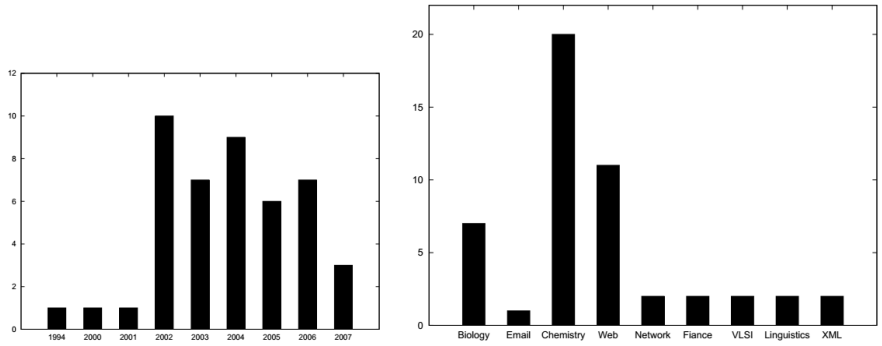
S: the frequent graph set

```
procedure AprioriGraph(D, minsup, S_k)
    S_{k+1} ← ∅                                   ▷ Initiate S to an empty set
    for each frequent g_i ∈ S_k do
        for each frequent g_j ∈ S_k do
            for each size (k+1) graph g formed by merging g_i and g_j do
                if g is frequent in D and g ∉ S_{k+1} then
                    insert g to S_{k+1}
    if S_{k+1} ≠ ∅ then
        AprioriGraph(D, minsup, S_{k+1})
    return
```

stage, DFS requires less memory usage (in exchange for less efficient pruning) [Han et al., 2000].

For each discovered graph g, it performs extensions recursively until all the frequent graphs with g embedded are discovered. The recursive procedure stops once no frequent graph can be generated. This pattern growth strategy is simple, but not efficient, as the same graph can be discovered many times. Graphs that are discovered more than once are duplicate graphs. Repeated discovery of the same graph increase the redundant isomorphism check, and thus in order to reduce the generation of duplicate graphs, each frequent graph should be extended as conservatively as possible. This principle leads to the design of several new algorithms. A typical such example is the *gSpan* algorithm [Yan and Han, 2002], *gSpan* is arguably one of the most frequently cited FSM algorithms. The *gSpan* algorithm is designed to reduce the generation of duplicate graphs. It does not require searching previously discovered frequent graphs for duplicate detection, or extending any duplicate graph, yet still guarantees the discovery of the complete set of frequent graphs.

In an extensive survey which compared and discussed different algorithms for FSM [Jiang et al., 2013], the authors drew a note that the periods of high activity in FSM-related work are the early 1990s (coinciding with the introduction of the concept of data mining), followed by another period of activity from 2002 to 2007. After that period, there has been much work focused on variations of existing algorithms instead of introducing a new approach, indicating that the field has quite reached maturity. Figure 3.5 presents an overview of the domain of FSM regarding the number of significant FSM algorithms that have been proposed since 1994, and the major applications. Other than the research on FSM itself, the importance of FSM is also



(a) Year (b) Application

Figure 3.5: Distribution of the most significant FSM algorithms with respect to the year of introduction and application domain [Jiang et al., 2013]

reflected in its various areas of application, where notable examples can be found especially in chemistry, bioinformatics, drug discovery (mined frequent structures can be used as features to classify chemical compounds to study protein structures), traffic analysis in communication networks, and web data mining [Cook and Holder, 2006].

Exact FGM algorithms are much more common than inexact search based FGM algorithms. They can be applied in the context of graph transaction based mining or single graph based mining. A fundamental feature for exact search based algorithms is that the mining is complete, *i.e.* the mining algorithms are guaranteed to find all frequent subgraphs in the input data. As noted in [Kuramochi and Karypis, 2004], such complete mining algorithms perform efficiently only on sparse graphs with a large amount of labels for vertexes and edges. Due to this completeness restriction, these algorithms undertake extensive subgraph isomorphism comparison, either explicitly or implicitly, resulting in a significant computational overhead.

Inexact search based FGM algorithms use an approximate measure to compare the similarity of two graphs, *i.e.* any two subgraphs are not required to be entirely identical to contribute to the support count, instead a subgraph may contribute to the support count for a candidate subgraph if it is in some sense similar to the candidate. Inexact search is of course not guaranteed to find all frequent subgraphs, but the nature of the approximate graph comparison often leads to computational efficiency gains.

There are only a few examples of inexact frequent subgraph mining algorithms in the literature. However, one frequently quoted example is the SUBDUE algorithm [Ketkar et al., 2005]. SUBDUE uses the minimum description length principle to compress the graph data; and a heuristic beam search method, to narrow down the search space. Although the application of SUBDUE shows some promising results in domains such as image analysis and circuit analysis, the scalability of the algorithm is an issue, i.e. the run time does not increase linearly with the size of the input graph. Furthermore, SUBDUE tends to discover only a small number of patterns.

Two notable inexact search based FGM algorithms are *gApprox* [Chen et al., 2007] and RAM (Randomized Approximate Graph Mining) [Zhang and Yang, 2008]. The *gApprox* algorithm uses the notion of an upper-bound for support counting, and an approximation measure to discover frequent approximately connected subgraphs in very large networks. Empirical studies based on protein-protein interaction networks indicated that *gApprox* is efficient and that the discovered patterns were biological meaningful. RAM is founded on a formal definition of frequent approximate patterns in the context of biological data represented as graphs, where the edge information tended to be inaccurate. Reported experiments showed that RAM can discover some important patterns which can not be found by exact search based mining algorithms.

Another inexact search based FGM algorithm is GREW [Kuramochi and Karypis, 2001]. However, GREW is directed at finding connected subgraphs which have many vertex-disjoint embeddings, in single large graphs. GREW uses a heuristic based approach that is claimed to be scalable, because it employs ideas of edge contraction and graph rewriting. GREW deliberately underestimates the frequency of each discovered subgraph in an attempt to reduce the search space. Experiments on four benchmark data sets showed that GREW significantly outperformed SUBDUE with respect to: runtime, number of patterns found, and size of patterns found.

To summarize, Frequent Subgraph Mining is a prolific research area, where different types of mining strategies, performed on different types of graph, to output different kinds of patterns, can be identified in the literature. Frequent Pattern Mining, a more general term, is a mining task that discovers patterns that occur frequently and/or have some distinctive properties that distinguish them from others, often disclosing something inherent and valuable. The task also includes the discovery of rare patterns, or negative patterns revealing items that occur very rarely yet are of interest. Frequent pattern mining can help distinguish between noises and meaningful patterns. We may assume that the appearance of very infrequent item(s) can be caused by random noise while relatively frequent patterns often carry more information gain for constructing more reliable models and should not be filtered out.

3.1.4 Graph Indexing

As described in the previous section, frequent subgraphs mining expose the intrinsic characteristics of a graph database. An index is a list of keys and pointers useful to speed-up the access to some organized content: in the context where the images are represented by graphs, the image indexing and retrieval task is transformed into graph indexing and retrieval task.

It is inefficient to perform graph search by checking subgraph isomorphism between the query q and each of the graphs in the graph database $g_i \in \mathbf{D}$, because subgraph isomorphism checking can be computationally expensive ($NP-$complete in the worst case scenario) and it has to be executed for all entries in the database.

Given a query shape, the goal of indexing is to efficiently retrieve, from a large database, similar shapes that might account for the query, or some portion thereof

(in the case of an occluded query or a query representing a cluttered scene). These candidate models will then be compared directly with the query, *i.e.*, verified, to determine which candidate model best accounts for the query. Similarly, given a graph query, it is desirable to retrieve graphs quickly from a large database via graph indices. In general, during offline index construction, a set of indexing features are selected. For each feature f, we build an index which points to entries that contain f. In online query phase, for each query q is issued, a candidate graph g_i in the database can be quickly pruned without performing subgraph isomorphism checking, if feature f is contained in q but is not contained in g_i. There are a lot of work in literature that deal with graph based indexing [Giugno and Shasha, 2002, Yan et al., 2004, Cheng et al., 2007, Zhang et al., 2009, Williams et al., 2007, Han et al., 2010]. If a graph database is well-indexed based on well-selected features, it has been shown that indexing results in significant query processing speed-up.

The indexing mechanism works as follows: after creating the graph representation of each image the frequent subgraphs are discovered. The key issue is the appropriate choice of the minimum support threshold because it determined which subgraphs are frequent. If the threshold is set too high, only a few subgraphs will exceed the threshold in terms of appearance frequency, thus they will not represent the database correctly. Setting the minimum support threshold too low results in too many "frequent" subgraphs listed, which will reduce the uniqueness of the features and effectively introduce more "noise" into the indexing scheme.

The graph mining can be done using any of the known graph mining algorithms. After discovering the frequent subgraphs in the database the images are indexed using the frequent subgraphs as the indexing key. It is necessary to build graph indices in order to help processing graph queries. XML query is a kind of graph query, which is usually built around path expressions. Various indexing methods [Goldman and Widom, 1997, Milo and Suciu, 1999, Cooper et al., 2001, Kaushik et al., 2002, Chen et al., 2003, Shasha et al., 2002] have been developed. These methods are optimized for path expressions and semi-structured data. In order to answer arbitrary graph queries, systems like GraphGrep [Shasha et al., 2002] are proposed. All of these methods take path as the basic indexing unit. We categorize them as path-based indexing approach. Historically, people have been using frequent subgraphs for indexing features. The index built with this model can achieve better performance in processing graph queries. Since discriminative frequent structures capture the intrinsic characteristics of the data, they are relatively stable to database updates, thus facilitating sampling-based feature extraction and incremental index maintenance.

3.2 Content-Based Image Retrieval

In this section, we introduce our literature survey on Content-Based Image Retrieval (CBIR) which has been fundamental for our research works presented in the next chapters. Again, as the amount of the related work is too vast to be covered thoroughly here, we focus our literature research on the selected topics of CBIR including

image representation, feature selection, and feedback integration in CBIR.

As briefly introduced in Section 1.2, CBIR systems were introduced to overcome these disadvantages in text-based retrieval systems. The term "content-based" in CBIR signifies that these techniques make use of the inherent visual contents of an image to perform a query, rather than depending on manually assigned annotations as in other image retrieval methods. Pioneering work on this problem started as early as in the 1980s, for example in [Chang and Liu, 1984] the author presented a picture indexing and abstraction approach for pictorial database retrieval. However the term CBIR was first coined in [Kato, 1992] to describe the experiment of digital image retrieval by comparing image color and shape features of each database image with that of the query image. It has been widely used since then to refer to all similar techniques and processes of searching and retrieving images using the common representative features such as colors, shapes, textures, etc.

The following parts of this section provide background information, including an overview of CBIR and a survey of the current state-of-the-art. We introduce various techniques used to improve CBIR accuracy, and identify the gaps in existing techniques.

3.2.1 Features Selection and Description

What if one is asked to rank comic images in a comic book by their similarity or relevance to a query containing some comic character? An objective answer to this question could be obtained if *every* element of these images could be quantitatively measured and compared. However the old saying "a picture is worth a thousand words" holds true: the amount of information encoded by an image is so high that it is impractical to describe it all, let alone the huge amount of comics need to processed for the system to become practical. The major challenges for CBIR include the application-specific definition of similarity (based on users' criterion), extraction of image features that are relevant to this definition of similarity, and organizing these features into indices for fast retrieval from large-scale repositories. The choice of features becomes a critical task when designing a CBIR system, because it is closely related to the definition of similarity.

A typical CBIR system makes use of low-level features to describe images. Generally, CBIR systems follow a common sequence of processes which can be grouped into two stages – offline and online:

The offline stage: In this stage, also known as image representation stage, visual features or descriptors from the images are extracted and organized to create the signatures of the images, so that the images can be searched and compared later by their signatures. In general there are two types of features: local features and global features. Global features characterize the image as a whole, usually related to texture and color information, while local features encode the local description of a point or a region in the image.

The online stage: Once the query image is given, the same process as in the

offline stage is applied to the query image in order to find its signature. A measurement is defined to compute the similarity between the query image and all the images indexed in the offline stage. This measurement of similarity is then used to rank the images in the order of relevance, or can be used to classify the images in binary manner as "similar" or "not similar". The most similar images are then returned as the results of the retrieval process. The ranking of the relevance can also be displayed to the user, since the integration of user feedback (or relevance feedback) allows further refinement of the results to overcome the semantic gap.

Features extraction is the process by which an image is analyzed to obtain quantifiable and objective properties of its visual contents. The common features include, but are not limited to, color, texture, shape, and the spatial arrangement of ROIs (Regions of Image). In the offline stage, feature extraction approaches can be classified into two categories: (i) global approach by using global visual features, and (ii) local approach by considering images as the combination of multiple objects, keypoints or regions.

Global Approaches: In global techniques, images are characterized by visual/statistical features calculated from the entire image. Global features capture the overall characteristics of an image but fail to identify important visual characteristics if these characteristics occur in only a relatively small part of an image.

Local Approaches: The features to be extracted can also be local, e.g. calculated from specific Regions of Images – ROIs. Image segmentation techniques are used to define the ROIs for local feature calculation. Local features describe the characteristics of a region or a small set of pixels, so they are inherently better in describing the details. Notable examples of the most widely used local features include Scale Invariant Feature Transform (SIFT) [Lowe, 1999], Speed-Up Robust Features (SURF) [Bay et al., 2006], Binary Robust Independent Elementary Features (BRIEF) [Calonder et al., 2010], Oriented FAST and rotated BRIEF (ORB) [Rublee et al., 2011]. These features are local and based on the appearance of the object at particular interest points, and are designed to be invariant to location, scale and rotation. They are also robust to changes in illumination and noise. These algorithms allow the keypoints to be highly distinctive, which allows object identification with low probability of mismatch. They are also fast to extract, making it easy to perform the matching process against a large database of local features.

In recent years, there has been a shift towards the use of local features, largely driven by the belief that most images are too complex to be described in a general manner; however, the combination of local and global features remains an area of investigation for practical computer vision applications.

Nevertheless, besides the features that are sophisticatedly design (e.g., the keypoint's signature that captures gradient information around a detected keypoint), the most common features extracted for CBIR purpose are usually primitive features and fall into several categories:

- **Color:** Color features are the most effective features, and almost all systems

employ colors. They are relatively robust to background complications, and are independent to image size, viewpoint and orientation. Color histograms are used to compare images in many applications. Besides color histograms, color information can also be represented by color moments, and several other representations. Their advantage is their efficiency, as they are almost trivial to compute. However, color histograms lack spatial information, so images with different appearances can very well have similar histograms. Color is one of the most important image indexing features employed in CBIR. Some of the popular methods to characterize color information in images are color histograms [Swain and Ballard, 1991, Hafner et al., 1995], color moments [Stricker and Dimai, 1996], etc. The authors in [Del Bimbo, 1999] and [Schettini et al., 2001] provide a comprehensive survey of various methods employed for color image indexing and retrieval in image databases.

- **Texture:** Texture is an innate property of almost all surfaces, containing information about the structural arrangement of these surfaces and their relationships to other surfaces. Texture can be extracted from the coefficients of wavelet transforms and Gabor filters [Do and Vetterli, 2002, Manjunath and Ma, 1996]. The most popular texture features are the Haralick texture features [Haralick et al., 1973] extracted from a co-occurrence matrix. These features are most useful for regions or images with homogeneous texture pattern. Like color features, they are insensitive with respect to image rotations; and shifts or scale changes can be included into the feature space.

- **Shape:** Shape description is an important issue both in object recognition and classification as shape features capture the geometric details. There are many shape representation and description techniques in the literature. Among notable shape features, the seven Hu moments [Hu, 1962] are invariant for transformations and thus are ideal for situations where shapes may be rotated, translated, or have varying scales. The Zernike moments [Khotanzad and Hong, 1990] is also among the well-known shape descriptors. Methods that are based on boundary (contour) description can also be classified as methods based on shape feature, e.g. kAS ($k-$adjacent segments) [Ferrari et al., 2008] or boundary fragment models (BFM) [Opelt et al., 2006]. In BFM method, fragments are represented by disjointed boundary contours of an object, the idea is to learn a codebook of contour fragments with regards to an object's centroid, and use it for detection. In kAS method, contour segments are partitioned into lines, and several adjacent lines are combined to form a robust feature. In an extensive survey, the authors in [Zhang and Lu, 2004] discussed thoroughly on different shape descriptions representations. Shape features have been widely used in CBIR in conjunction with color and other features for indexing and retrieval.

The effectiveness of a CBIR approach highly depends on the quality (or suitability) of the chosen features. General-purpose features can be extracted from almost all images but are not necessarily appropriate for all applications. The features used in CBIR are generally dependent upon the specific domain and for a particular purpose. Besides, it is usually necessary to extract many features in order to describe the

objects sufficiently, but on the other hand, this might result in very high-dimensional feature vectors. These high-dimensional feature vectors cause a problem commonly known as "curse of dimensionality". This dimensionality problem can get even worse if we want to include the structural information contained in an image into the feature vector.

3.2.2 Measuring Image Similarity

To determine if a certain image is relevant to the query one, or to rank the retrieved images in terms of relevance, we have to be able to measure the similarity of the properties of their visual components. In CBIR, the purpose of similarity measurement is to compare the signature of the query image to that of all images in the collection, to find out the most similar images to the query one. For example, the signature of images can be represented in form of a histogram of visual words. Once we obtain the signatures of the query image and the images in the database, it is straightforward to calculate the histogram distances [Chen et al., 2009].

In the online stage, it is an essential process to assess the differences between the features extracted (the signature) of individual images and that of the query. If two images hold a small histogram distance, they contain similar visual words with similar distribution and therefore, they should be similar to each other. The images in the collection that have minimum histogram distances will be returned in a ranked list as the result of the retrieval process. In practice, the similarity functions and optimizations such as which features to take into calculating and their weights, usually depend on the domain for which the CBIR system is being applied. The result of similarity measurement can either be a ranking of returned images based on the degree of relevance with the query image, or a binary classification.

One of the commonly used methods is to calculate the Minkowski distance [Deza and Deza, 2009], also known as generalized (or weighted) Euclidean distance. It is defined as:

$$D = \left(\sum_{i=1}^{n} w\left(x_i\right) \left(x_i\left(I\right) - x_i\left(Q\right)\right)^p \right)^{1/p} \tag{3.1}$$

where x_i represents the $i-$th feature component, $w(x_i)$ represents the weight for the feature, n represents the number of features, $x_i\left(I\right)$ represents the database image and $x_i\left(Q\right)$ represents the query image.

Minkowski distance is typically used with p being 1 or 2, which correspond to the weighted Manhattan distance and the Euclidean distance, respectively. However this is only a possible way of defining distance, we will discuss several distances commonly used in comparing image signatures as well.

Given two histograms p, q: $p = \{p_1, p_2, \ldots, p_n\}$ and $q = \{q_1, q_2, \ldots, q_n\}$ of n dimensions, there are a lot of distance metrics to calculate the histogram distance between two images. The choice of a best histogram distance metric depends on the specific image collection, the length of visual vocabulary. Popular histogram distances

that are used in the literature include $l1$-norm, $l2$-norm, χ^2, Jaccard Distance [Deza and Deza, 2009], Bhattacharyya distance [Bhattacharyya, 1943], etc.:

$$d_{l1}(p,q) = \sum_{i=1}^{n} |p_i - q_i| \tag{3.2}$$

$$d_{l2}(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{3.3}$$

$$d_{\chi^2}(p,q) = \frac{1}{2} \sum_{i=1}^{n} \frac{(p_i - q_i)^2}{p_i + q_i} \tag{3.4}$$

$$d_{Battacharyya}(p,q) = 1 - \sum_{i=1}^{n} \sqrt{p_i q_i} \tag{3.5}$$

For illustration purpose, we assume that a simplistic way to compare images is to determine their similarity based on color distribution: a vector of most represented colors in the image are obtained, and fed to the distance calculation given by equation 3.1 to see if the two images have a "small" distance in terms of their colors. However, if similarity was based on colors alone, then two images of very different subjects could be considered similar, and besides, this choice is inappropriate for grayscale images.

An underlying assumption of most CBIR systems is that the chosen features used are sufficient to describe the image accurately. The choice of features must therefore be made to minimize the semantic gap, which is a major challenge. Semantic gaps occur when the expected result from the user differs to the images retrieved by the system. The reason is that CBIR systems are unable to interpret images, they do not understand the "meaning" in the images in the same way as human, i.e., CBIR is performed on the basis of image features not image interpretations.

Normalization A problem arising from using the aforementioned metrics directly, is its subjectivity to the magnitude of the feature values. For instance a feature ranging over a larger scale will contribute more to the dissimilarity measure than one with a smaller scale thereby corrupting the representative value. This can be solved through normalization steps. Normalization refers to the re-adjusting of values within a particular feature vector. This ensures that the values are more appropriately represented, based on the proportional magnitude relative to values in their sets. Normalizing all the feature measurements result in values lying in the same dynamic range $[0, 1]$, thus removing the previously bias similarity measure. A similarity measure can be trained to favor particular interpretations. Neural networks, Bayesian classifiers, support vector machines (SVMs), and Hidden Markov Models (HMM) can be used for this purpose.

3.2.3 Image Representation

Once the features are chosen, detected and described, image representation is also one of key components linked with different tasks. The purpose of image representation

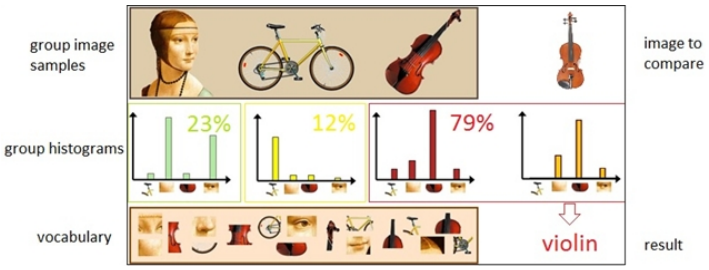


Figure 3.6: A bag of visual words for each of the objects including a woman, a bicycle, and a violin. Each bag represents a histogram of the "visual words" or key image patch occurrences from a given dictionary. The input image of a violin has its bag of visual words histogram that closely resembles the category of class "violin", and is assigned the label "violin" as this class has the smallest histogram distance with the bag representing the querying object.

is to organize the extracted features to create image's signature. Concerning this purpose, a variety of solutions have been proposed. We introduce here several of recent related work in this aspect.

Bag-of-Words Representation

Bag-of-Words (BoW) is a CBIR representation paradigm worth mentioning due to its huge popularity. The idea of Bag of Visual Words (BoVW) model was inspired by Bag of Words model in text retrieval and document classification. In BoW model, each document is represented by a "bag" of words. This "bag" is a sparse vector of frequency of words from a dictionary made up by important and prominent key-words that occur in the text documents. A document is then represented as a histogram over the vocabulary in the dictionary. The same idea is applied in BoVW model: the visual words, also called codewords, are an analogy to words in BoW for text documents. The codebook made up from visual words is an analogy to a word dictionary, so each image can be represented as an orderless collection of those visual words. Figure 3.6 shows an example of how different objects can be described by a different "bag" of visual words for each one of them, and how the matching is done by comparing the histogram of visual words [1].

In general, a CBIR system based on BoVW approach has the following steps: First, the local feature detection is used to detect the key-points in the image, (e.g. SIFT [Lowe, 1999] or SURF [Bay et al., 2006]), for a set of image patches. The image is then considered as a set of the features: the patches can be either at the key-point locations or densely sampled on a regular grid of the image. The second row in Figure 3.6 illustrates this: the images of the three different objects are represented by different "patches" (features). Images with different objects contain different sets

[1]Image excerpted and reproduced from the source in http://deepcore.io/2017/04/18/BoVW_Part_1.html

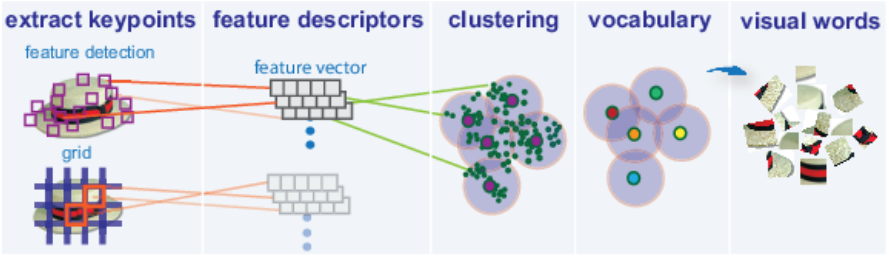


Figure 3.7: The construction of visual words. A clustering algorithm is used to divide the feature space, where each green dot presents a single local feature, into clusters. The centroids of those clusters are used to build the visual vocabulary.

of local features, vice versa, images with the same objects or scenes should contain similar ones. The next step of BoVW model is to determine a set of distinctive features to serve as visual vocabulary and uses this vocabulary to represent every features found in the image.

After that, a clustering technique is applied to quantize these feature descriptors into a fixed number of clusters. The centres of the generated clusters become the visual words or codewords. The whole set of cluster centres is the dictionary or codebook. Two main methods are usually used for visual vocabulary construction: $k-$means [Nister and Stewenius, 2006, Jain, 2010] and Gaussian Mixture Model (GMM) [Stauffer and Grimson, 1999]. GMM assumes that the local features are made from the combination of several Gaussian distributions of different means and variances. Unlike $k-$means method for building vocabulary, where each feature is assigned to a single cluster, GMM method allows a feature to belong to several clusters with a probability function. GMM implements Expectation Maximization technique to determine the cluster.

Once the visual vocabulary is obtained, each feature descriptor will be associated with the nearest visual word (or visual words) in the dictionary, and an image can be described as a feature vector, according to the presence and count of each visual word. As illustrated in Figure 3.7, the three leftmost panels show the process of extracting and clustering the descriptors of local features, and the two right most panels show the centroids of the clusters become "visual words". An assignment is called Hard Assignment (HA) if each descriptor is assigned to a single visual word, and Soft Assignment (SA) otherwise [Van Gemert et al., 2010]. Each bin of the histogram is the frequency of the appearances of one word in the image.

Finally, the classification or comparison stage turns into a histogram based classification: the feature vectors representing the images in the database are usually fed into a supervised learning platform object and scene classification later.

The basic Bag-of-Visual-Words model uses histogram as image signatures. Hence, typically BoVW models do not contain geometry information or spatial layout of local features in the image. The most basic BoVW also ignores how the image features are

distributed across images. Considering this information as an important attribute, different research work have tried to remedy the original drawback of BoVW. Here we briefly introduce several approaches that have been proposed in the literature with regards to this problem.

For example, approaches to embed spatial information of visual words can be found in [Khan et al., 2012b, Marszaek and Schmid, 2006, Chen et al., 2009, Tsai et al., 2014]. In [Khan et al., 2012b], the authors introduced and proposed the use of PIW (Pair of Identical visual Words), which is defined as the set of all the pairs of visual words of the same type. By the use of PIW (Pair of Identical visual Words), a spatial distribution of visual words could be represented as a histogram of orientations of the segments formed by PIW. Extended Bag of Features (EBOF) model [Tsai et al., 2014] divides the image into fan-shaped sub-images and a BoVW model is applied to each of them. A 2D Gaussian weighting mask is then applied to highlight the contribution of visual words located closer to the center, and the BoVW histograms of all sub-images are combined to build the image representation. Due to nature of the circular-correlation, the authors of EBOF claim that it is more robust to rotation, translation and scaling. In the same spirit, in [Marszaek and Schmid, 2006], the authors presented a spatial weighting extension to BoF model by increasing the weights of features that agree on the position and shape of the object, and suppressing the weights of background features.

However, one of the most well-known approach to embed spatial information to BoVW is Spatial Pyramid Matching (SPM). SPM is a simple yet effective extension of original BoVW representation, introduced by [Lazebnik et al., 2006]. The idea is first to recursively divide the image into multiple sub-blocks ($2^r \times 2^r$) at multiple resolutions, then compute the histogram of each sub-block at several spatial granularities. The final signature of the image is obtained via the concatenation of the histograms of each sub-block, with different weights according to their level in the spatial pyramid. The weight associated with level l of the pyramid is given by $w_l = \frac{1}{2^{L-l}}$. Figure 3.8 demonstrates this method. In such a way, not only the visual features of the image are taken into consideration, but their spatial distribution is encoded into the spatial pyramid image representation as well. SPM outperforms traditional BoVW by a large margin, and performs competitively against other elaborate methods. Directly based on the idea of traditional SPM, several other schemes have been proposed to improve the pyramid matching, notably sparse coding [Yang et al., 2009] to reduce the matching complexity; Locality-constrained Linear Coding (LLC) [Wang et al., 2010], Spatially Local Coding [McCann and Lowe, 2012], where the visual words are encoded with local information, instead of only coding by their visual appearance, leaving the spatial information to the pyramid grid.

In [Ren and Malik, 2003], the authors introduced the concept of grouping pixels into "superpixel". The purpose of grouping pixels into superpixels is to over-segment image into a large number of coherent, local regions, which retain most of the structures necessary for segmentation at the scale of interest. The compact superpixels can capture diverse spatially coherent information and multi-scale visual patterns of a natural image. Originally, superpixel algorithm used Normalized Cuts (NCuts) based

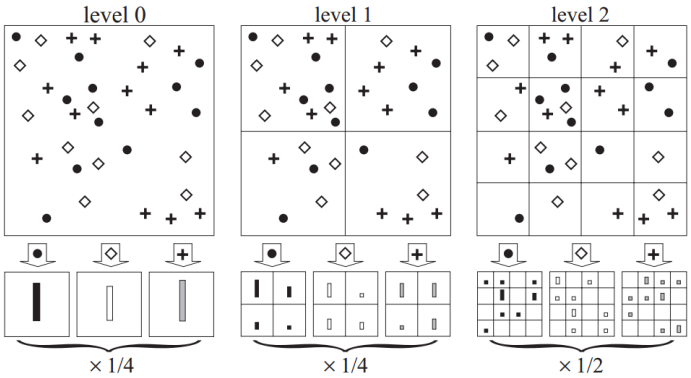


Figure 3.8: Example of constructing a three-level pyramid. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, SPM method subdivides the image at three different levels of resolution. Next, for each level of resolution and each grid cell, it counts the features that fall in each spatial bin. Finally, the spatial histograms are weighted and concatenated according to a formulation of pyramid match kernel. Figure excerpted from [Lazebnik et al., 2006].

on contour and texture cues, but later, different algorithm to construct superpixels have been developed, e.g. TurboPixels [Levinshtein et al., 2009] or SLIC (Simple Linear Iterative Clustering) superpixels [Achanta et al., 2012]. One drawback of superpixel approach is that while superpixels are usually expected to align with object boundaries for segmentation, but it may not be the case practice due to blur object boundaries and cluttered background [Schick et al., 2012].

Features encoding, i.e. transforming local image descriptors into histograms is a crucial part in BoVW model, so a large number of methods have been proposed to improve this process in feature space. For example, the use of Fisher vector image representation [Perronnin and Dance, 2007], in which the authors proposed to apply Fisher kernels to visual words represented by means of a GMM. Among the work that employ Fisher vector are [Krapac et al., 2011, Perronnin et al., 2010, Simonyan et al., 2013]. In comparison to the BoVW representation, this more sophisticated representation results in fewer visual words required. VLAD (Vector of Locally Aggregated Descriptors) [Jégou et al., 2010b] is considered a simplification of Fisher kernel.

Region-based Representation

Besides low level features such as color and texture, relative location of image regions is also a useful information in region classification and thus for image retrieval. To better support semantic-based image retrieval, a spatial context modeling algorithm is presented in [Ren et al., 2002] which considers six spatial relationships between region pairs: left, right, up, down, touch and front. For example, for retrieving the objects "sky" and "sea", if we only rely on color and texture feature, that two objects may

be really similar in characteristic (having blue shade and smooth texture). However, their spatial locations are different with "sky" usually appears at the top of an image, while "sea" should be found at the bottom. In general, the techniques that exploits spatial location usually define the location of the objects as one of the information for retrieval [Song et al., 2003, Mojsilovic et al., 2002].

A lot of region-based image representation (RBIR) methods can be found in the literature. In an early work [Stricker and Orengo, 1995], the authors proposed to divide an image into 5 fuzzy regions that partially overlap each other. For indexing, the first three moments of color distribution of each region is computed, and the image is represented by a feature vector composed of color moments of 5 fuzzy regions, i.e., a vector of length $5 \times 3 \times 3 = 45$ (number of regions $\times$ number of moments $\times$ number of color channels). Based on fuzzy regions, the feature vectors in the index can be relatively invariant to small translations and rotations. In [Ma and Manjunath, 1997], region centroid and its minimum bounding rectangle are used to provide spatial location information. In [Mezaris et al., 2003], spatial center of a region is used to represent its spatial location. In [Omhover and Detyniecki, 2004], spatial structure of the regions is formulated by fuzzy similarity features. The regions are described by a set of color and shape features.

Bag-of-Regions (BoR) model was introduced in [Vieux et al., 2012]. Inspired by BoVW model, in this approach low-level descriptors from segmented regions are extracted and represented as BoVW histograms. However, to address the semantic gap problem, the authors tried to make a translation between low-level image features and high-level user perceptions by building BoR signatures instead of image signatures. Co-occurrence criteria such as *min*, *max*, *median*, *sum* etc. are applied to build the BoR signatures. The authors argued that BoR signatures are a good counterpart that could improve the traditional BoVW approach, and could be more appropriate depending on the specific queries.

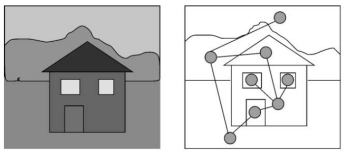
In several work on RBIR [Jing et al., 2004, Jing et al., 2002], the authors showed their approach of transforming local features of regions into a compact and sparse representation. Suppose after segmentation, an image I contains N regions, it is then represented by a vector of the form:

$$\hat{I} = \{(CI_{R_1}, w_{R_1}), \ldots, (CI_{R_N}, w_{R_N})\}$$

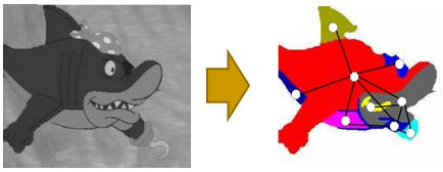
where CI_{R_i} and w_{R_i} are the codeword index and the important weight of region R_i, respectively. The sum of importance weights for an image should be equal to 1, i.e., $\sum_{i=1}^{N} w_{R_i} = 1$. The selected features are the first two color moments of each regions, and the codebook is generated by iteratively clustering these features. Finally each region in an image is labelled by a codeword index corresponding to the cluster it belongs to.

Structural Model-based Representation

In retrieval by image content, to better assess the similarity between images and reduce the semantic gap, multiple objects and their relationships in space could be



(a) Structural representation of an image by Regional Adjacency Graph (RAG).



(b) RAG representation of a comic image.

Figure 3.9: Representation of image by Regional Adjacency Graph approach.

employed. Simple collections ("bags") of local features are sometimes inadequate: besides the requirement that the shape, color and texture properties of individual image regions must be similar, they should have a similar arrangement (spatial relationships) as well.

We have mentioned above that RBIR methods take into account the relative position of image regions, or directional relationships between regions. However, such primitive relationships alone are still not sufficient to represent the semantic content of images as they ignore the topological relationships. That led to the rise in popularity of graph-based object representation. The centerpiece of the technique in this category is the consideration of an image as a structurally connected group of objects or Regions of Interest (RoI). When objects are adjacent, the arrangement of the local neighborhood can provide relationship features, as shown in Figure 3.9(a).

The relationships between components in an image are often in the form of graphs, trees, or hierarchies. Ideally, the relationships are described with a graph as the Attributed Regional Graphs (ARGs) or Containment Trees [Petrakis et al., 2002,Fischer et al., 2004]. Other well-known methods for representing spatial relationships include the use of triangular spatial relationships (angles between groups of entities) [Hoàng et al., 2010], directional information [Berretti et al., 2003], matrices indicating the relative ordinal (compass) directions of objects [Jaworska et al., 2010], and complex strings for detailing the topological and geometric interactions between objects [Wang et al., 2012]. However, graph-based representations received the most attention, and as a consequence, methods to compare graphs which represent image content have become of great interest.

This approach normally requires segmentation process. When we take into consideration of spatial relation between different regions of the image, the primitive directional relationship representation become the representation as Attributed Regional Graphs (ARG), in which the nodes often represent local features (objects parts), and edges represent the proximity relationships between nodes (Figure 3.9). This representation however increases the complexity of matching algorithms and the feasibility of indexing schemes. For this reason, no definitive and comprehensive solution has been yet proposed to support the application of ARGs in CBIR. In fact, while the distance between two sets of independent vectors can be computed in polynomial time,

the distance between two graphs requires the identification of an optimal error correcting (sub)graph isomorphism, which is an NP-complete problem with exponential time.

Generally speaking, when a graph-based representation is adopted for image description, the problem of image content comparison turn into some form of graph matching problem (which consists of finding correspondences between the nodes of the two graphs), or measuring the distance between attributed graphs. Graph matching is a mapping in aim to find a correspondence between the nodes/edges of one (possibly larger) graph to the other that model the patterns of interest. Depending on how the matching problem is formulated, the graph matching step can be either exact matching, inexact matching, graph kernels, graph embedding, or other matching techniques which were introduced in the previous section (Section 3.1).

With such characteristic, it is interesting to develop CBIR systems where images are represented by graphs and the retrieval stage involves exploiting benefits of graph representation. Out of the specific context of CBIR, the problem of comparing an input graph against a large number of model graphs is addressed in [Messmer, 1996, Messmer and Bunke, 1998] using a decomposition approach, where model graphs are repeatedly decomposed in subgraphs, which are organized by size in a global hierarchical index. At runtime, matching is accomplished by comparing the input graph against the subgraphs. Since there are still the gap between structural and statistical representation of images, recent approaches tend to bridge that gap by graph embedding techniques, explicitly by prototype, such as in [Bunke and Riesen, 2008, Ferrer et al., 2011, Riesen et al., 2007b] or spectrally such as in [Ren et al., 2008, Robles-Kelly and Hancock, 2007]. However, by doing so, we lose the structural links attained by graph construction. In our work in this thesis, we will dig deeper into the methods which preserve the structural links in graphs (i.e. the links between the nodes extracted from the comic images).

3.2.4 Indexing, Displaying Results and Integrating Feedback

Indexing

The large volume of modern image repositories and high feature dimensionality of images has also contributed to challenges in efficient real-time retrieval. In many cases, it is no longer viable to compare a query to every element of the data set. Efficient indexing schemes are necessary to store and partition the data set so it can be accessed and traversed quickly, without needing to visit or process irrelevant data; alternatively, the search space can be pruned by using only a subset of the features or applying weights to features.

Displaying

The difference between content-based and text-based image retrieval systems is that the human interaction is still an indispensable part of the latter system. Humans tend to use high-level and abstract features (concepts), such as perception of the visual information, keywords, text descriptors, to interpret images and measure their similarity, while the features automatically extracted using computer vision techniques are mostly low-level features (color, texture, shape, spatial layout, etc.). The total removal of human interaction factor results in a number of issues, such as the ability to deal with semantic attributes of images.

In general, there is no direct link between the high-level concepts and the low-level features. This well-known issue called semantic gap is the driving force behind studies about CBIR systems. There are possibilities that have been investigated to remedy this situation. Since semantic gaps generally require user to be involved as part of the retrieval process, this entails the design and development of a convenient system which allows users to have their say in the retrieval process, i.e., the user must confirm if the results of the retrieval match his intent.

A retrieval system must therefore provide some way for the user to meaningfully view and interact with the displayed data. Typically a CBIR system will not return only a single image result, but a sorted list of potential matches instead. The resulting images are ranked according to how well they satisfy the search criterion. In $k-$nearest neighbor search, the k most similar images are returned, based on their distance from the query in the feature space. Generally, arranging retrieved images based upon their similarity is helpful in picture selection tasks, i.e.,the properties of elements, the relationships between them, and supplementary data should be displayed to enable user semantic interpretation. The ability to iteratively refine queries to narrow down the search space is also valuable. The user then browses through the retrieved images to locate the best matched images, according to their semantic interpretation.

One of a huge improvement to the usability of the system, is the ability to refine the query, either through filtering, sorting or iterative searching. Relevance Feedback (RF) [Rui et al., 1998] is a mechanism by which the user is able to iteratively improve the pertinence of the retrieved images by marking retrieved images as "relevant" or "not relevant". The two most common ways of using RF in CBIR are to: (1) modify the query image based on the true positives returned, and (2) modify the results by assigning weights showing their relevance. Weights can be calculated from a set of labeled samples with ground truth information and adjusted to reflect a user's specific needs. The initial calculation is solely based on computed figures and how their values are distributed.

By using fixed user-defined weights, we usually face a problem that, what a user might perceive to be prominent in the query, might not be the same as what the computer registered. Relevance Feedback remedies this problem by focusing on the high-level features i.e. through the actual image displays, instead of the low-level features. It takes into account the semantics that could not be defined with the computed low-level features, by incorporating the user's perception and judgment.

This enables the user to narrow the retrieval to those most relevant to their semantic interpretation of the query, a means to bridge the semantic gap. While relevance feedback can be implemented in a number of ways, there must be a focus on displaying the images and features to the user, i.e., showing the user which features have made an image similar.

To improve the performance of similarity matching, different approaches have been introduced in the literature such as re-ranking and query expansion technique.

Re-ranking Technique: the similarity matching returns the result in form of a list of images which are similar to the query one. This list is ranked based on the distance of the images from the query image. The idea of re-ranking technique is to use additional indicator or tests to refine the results obtained by similarity matching step. Some examples of using re-ranking technique to enhance the performance image retrieval: geometrical re-ranking [Jégou et al., 2010a] and spatial re-ranking [Shen et al., 2012]. Jegou et al. [Jégou et al., 2010a] first use similarity matching to find the short-list of images similar to the query image and then match each descriptor of query images with the 10 closest ones in all images of the shortlist images. Then, the affine 2D transformation estimation is used as additional indicator to refine the shortlist. The images that pass the geometrical estimation filter are moved to first positions of the list and ranked with a score based on the number of inliers. Spatial re-ranking method in [Shen et al., 2012] shares the same idea: to use the spatial constraints to refine the results of similarity matching. They estimate a transformation between the query region and each target image, based on how well its feature locations are predicted by the estimated transformation and then use the separability of the spatially verified visual words to re-rank the result list.

Query Expansion Technique: The idea of query expansion techniques is to use the highly ranked images in the results as the new query images. By doing this, we can find some new relevant images which were not returned by the normal similarity match. A draw back of this technique is that it may return incorrect results if the expanded query image is not relevant [Chum et al., 2007, Chum et al., 2011].

3.2.5 Image Retrieval and Real World Applications

CBIR has vast and diverse application possibilities that range from library management, artwork retrieval, to biomedical research and fabric design. Since the early days of CBIR, commercial products and experimental prototype systems have been developed, such as QBIC by IBM [Flickner et al., 1995, Faloutsos et al., 1994], VisualSEEK [Smith and Chang, 1997], Photobook [Pentland et al., 1996], Virage [Gupta and Jain, 1997], PicHunter [Cox et al., 2000], Blobworld [Carson et al., 1999], SIMPLIcity [Wang et al., 2001].

As a challenging area of research, CBIR techniques, tools and algorithms originate from various fields such as Pattern Recognition, Statistics and Computer Vision. It attracted a huge research interest and has been growing tremendously in the last two decades, in terms of both the people involved and the papers published. An overview

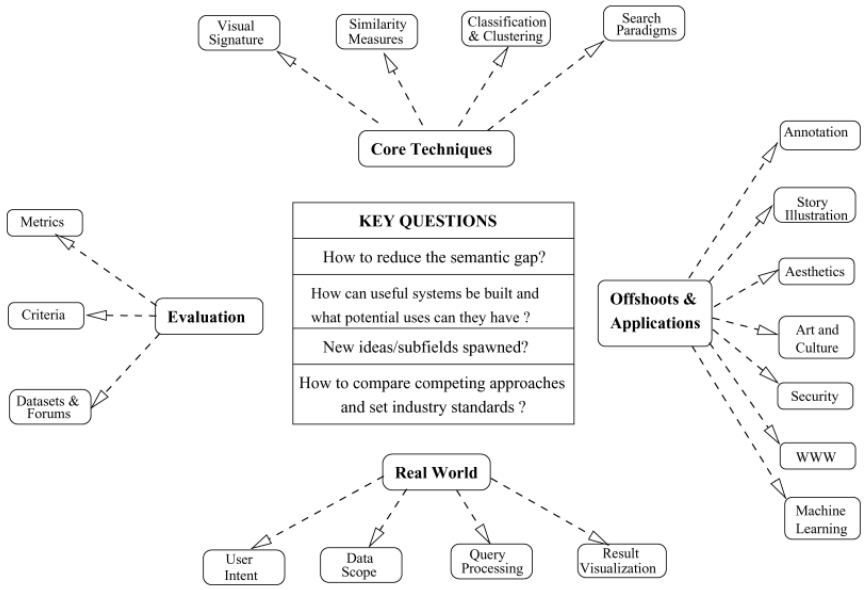


Figure 3.10: An overview of the many facets of different problems to pose for CBIR as a field of research. Image excerpted from [Datta et al., 2008].

of work in this area can be found in [Veltkamp and Tanase, 2001, Smeulders et al., 2000, Rui et al., 1999, Goodrum, 2000]. The diagram in Figure 3.10 illustrates an overview of the image retrieval problem. CBIR is a vast area of research, with many problems resolved at different levels or are staying unresolved.

Text information is a common complement to image features in general CBIR research. While CBIR implies the search is based on visual content, it is totally appropriate if the search can be enhanced with the combination of non-image data. In [Rahman et al., 2009] the authors presented a technique that used the correlation between text and visual components to expand the query. Their comparison of text, visual, and combined approaches revealed that the text retrieval had a higher mean average precision than the purely visual method, while the combined method outperformed both text and visual features alone. A similar approach can be found in [Chu et al., 1994], which returned a better retrieval results. However, the result can be explained by the nature of the used data set: the images in this medical data set were highly annotated and this made combining text-based retrieval inherently better than using only visual features.

Summary

To close this section about Content-Based Image Retrieval, it is worth remarking that CBIR systems are generally not created to replace the traditional annotation-based image retrieval systems, but to complement them instead. However, with the

ever-increasing amount of images generated every day, the ability to retrieve images without having to rely on labels has made CBIR the ideal, and sometimes the only solution for large repositories, where it is not feasible to manually assign keywords and annotations. As CBIR systems became sophisticated, in many cases, CBIR proves to be much more practical, since it allows retrieving the desired images without the ponderous, subjective and error-prone task of image description, and has therefore dramatically improved the usability of the image retrieval systems in general.

3.3 Comic Book Images Analysis

This section aims to provide a comprehensive survey and summarizes significant results in the area of comic images analysis and retrieval. Besides introducing related work in the literature concerning comic image analysis in general, we will put a greater focus on comic image retrieval, as this task is directly related to the work done in the thesis.

Specifically within computer vision area, the research about comics can be organized into different inter-dependent categories: comics element content analysis, user interaction, comics content generation, and comics content retrieval. While "content analysis" seems to be a broader term that encompass other categories, by using the term analysis here, we are referring to that category of work done in extracting constituent elements of a comic page, such as panel, balloon and text.

3.3.1 Comics Element Extraction

Comic book images are composed of text and graphics that can be decomposed as drawings and line drawings.The first challenge of comics book image analysis is to retrieve the layers that correspond to each step of the comic image design process (e.g. stroke, text, color). Processing each layer separately would greatly simplify layout and content retrieval. Due to the specific characteristics of comic books, the effort to extract the components of comic images may include: detecting the panels, locating the dialog balloons, extracting and recognizing dialog texts from the graphic background.

Layout analysis consists in segmenting the image into several geometrical blocks that contain the same type of information: text, graphics, table, drawing, etc. Then logical information can be retrieved using domain knowledge and the spatial position of the elements (e.g. "header and title are on the top", "page number is on the bottom-right corner", "reading order" can be considered some of the knowledge). Like advertisements or posters, comic book images can be classified as mixed content documents. Those documents generally contain non-standard text fonts, various text sizes and orientation mixed with graphics, images and logos, making layout analysis a non-trivial problem.

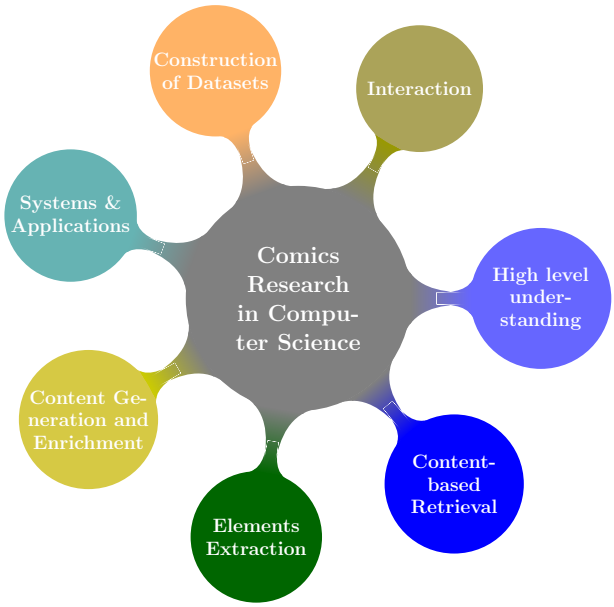


Figure 3.11: Comics Research in Computer Science

Layout, Panels and Balloons

Panels Extraction

The ability to localize the position of the panels in a comic page is an essential step due to several reasons. First, in terms of document content understanding, the reading order can be deduced from the layout, and since each panel contain a key moment of the story, panel localization can then be further analyzed to extract features used for content based indexing and retrieval [Le et al., 2016]. Second, in terms of improving reading experience, with the prevalence of reading comics on mobile devices (some of which may have a small screen), panel localization becomes extremely helpful to present the comic pages in panel-by-panel order to the reader, to avoid constantly zooming/panning and scrolling [In et al., 2011].

Several techniques have been proposed to extract panels for that purpose. Early approaches are usually based on white line cutting algorithm [Chan et al., 2007], line segmentation using Canny operator and polygon detection [Li et al., 2014], recursive X-Y cut [Han et al., 2007], corners detection [Stommel et al., 2012], watershed [Pon-sard et al., 2012], or gradient [Tanaka et al., 2007].

Those methods however, only take into consideration of "typical panels", which are (usually rectangular) boxes, with clear and closed border constructed by lines, while ignoring special cases such as joined panels, unclosed panels, or borderless panels (e.g. there are cases where panels are not closed, or are marked by a change in the

background). Connected-component approaches were introduced to address these shortcomings, such as in [Pang et al., 2014], where the method first closes the open panel and identifies a page background mask, then employs a recursive binary splitting with dynamic splitting line to disjoint the panel from a group of panel; or [Arai and Herman, 2010], where frame detection by blob extraction method and division line detection. Still, these methods are more error-prone at regions that connect several panels by some artistic effect.

Other approaches are based on region growing and mathematical morphology, such as a technique to detect panels and speech balloons in [Ho et al., 2012]. The authors indicated that the frame layout in comic pages may vary from page to page, but can be separated into three types: *simple* where panels are only in closed shape and totally separated by gutters, *complex* where the plate consists of regular panels and the object can be extended over the border of the panels, and the *hard* type where panels are in free style and there are complex overlaps of panels. The page is first gone through binarization using region growing algorithm, which results in panel in black blocks, then mathematical morphology is applied to break the link between the black solid blocks if existed. This method can remove such connecting elements but also remove information on the panel border.

Balloons Extraction

Balloons or bubbles are the visual unit that conveys spoken dialogs or thought. They are key elements in comics, as they offer the link between the textual content and the comic characters. There are many specialized forms of balloons, mostly to convey emotions or to mark the type of the text content inside. Traditionally, balloons are presented in oval shape, with a pointer or tail to indicate to which object they are associated with [Goggin, 2010, Robin Varnum, 2007]. Thus, apart from being crucial for document understanding, speech balloon detection can also support localization of the characters [Sun and Kise, 2011], or can be beneficial in applications such as translation assistance, or text-to-speech or reading order deduction [Guérin, 2012].

In [Arai and Tolle, 2011], the authors proposed a blob detection method based on connected-component detection with four filtering rules applied to manga analysis. The rules are based on blob minimum size, white pixel occurrence, inclusion of vertical straight lines and width to length ratio. Similarly, in [Ho et al., 2012], besides panel detection, the authors also proposed a scheme to detect speech balloons in each panel by connected-component approach. Based on the assumption that the speech balloons are always in light color, candidate areas are first selected by their color values in HSV color space. A blob is then considered as speech balloon if it contains text (ratio between the text area and the blob bounding box higher than a threshold), however text block detection was not addressed in details. According to the described result, this method can deal with complicated case where the author draw extended contents that overlap two panels or more.

The analysis of open balloons is quite different than the closed ones, and may require a different approach than blob and connected-component. In [Rigaud et al., 2013a, Rigaud et al., 2013b], an active contour model for speech balloon detection was proposed. The authors argue that in most cases, the location of text is generally

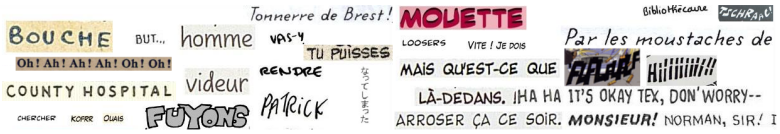


Figure 3.12: Various types of text in comic images. Image excerpted from [Rigaud, 2014]

a good hint for balloon detection, thus the problem of speech balloon detection can be considered as the fitting of a closed contour around text area. The active contour (also known as "snake") model is a deformable model that moves through the spatial domain of an image to minimize the energy function [Kass et al., 1988]. Based on this concept, the active contour model was adapted to detect balloons, by introducing new energy terms based on domain knowledge about the relationship between text and balloons (the text is considered to have been already detected in the image). Each energy term (including *external energy*, *internal energy* and *text energy*) is mathematically defined to construct the energy function of the contour model. Given that the text lines are already detected and grouped, the idea is to progressively push the initial snake (which tightly covers the text area) away from the text area and towards the balloon boundary, increasing the weight to both external energy and text energy terms.

Other than the principal function of balloons as the areas encompassing textual content, balloons may come in many different shapes, and the shapes are also an important indicator to understand the context as well as emotions and other non-verbal information. In [Rigaud et al., 2015a, Rigaud et al., 2015b], it was demonstrated that the detection of speech balloons' tails allows locating ROIs (regions of interest, which likely contain talking characters), and character-to-speech association (knowledge-driven approach).

Text Localization, Extraction and Recognition

Text localization and recognition problem in comic images is still a research problem of great interest as it opens up several interesting applications, such as OCR training, translation, speech synthesis and re-targeting to different mobile reading devices [Matsui et al., 2011]. A large number of studies have been done on text localization in natural images and document images. Yet text localization in comic images is quite different from both of them, or to be more precise, lies somewhere in between. Figure 3.12 gives a glance on the great diversity of texts that can be found in comic images. Besides, the content inside each panel comprises backgrounds of a graphical nature, and since a comic page can be considered free-style documents, there is no template or regular structure to facilitate the extraction of the layout and to predict the location of the text.

Scene text localization in images and videos is a prolific research topic [Jung et al., 2004, Gomez and Karatzas, 2013]. However, applying existing real scene text

detection methods to comics would require modification and improvement for several reasons. First, although both the tasks of text localization in comic and natural images include filtering out background noises from the actual text areas, natural images contain abundant variation in color, brightness and texture that can be used in the text localization process, while comic images are mainly composed of simple white-black line drawings. Second, there can be a great variation of fonts, case, orientation, scale, spelling and hyphenation in a single comic page image. Third, text lines are quite short compared to other types of documents, although text lines in comic images align horizontally and vertically within the text area (similar to that in the standard document images), but they only take up a small portion of the whole comic images, and are surrounded by the drawn content. As a result, document image processing techniques such as projection profile analysis, run-length smoothing algorithm and crossing count cannot be used.

Bottom-up approaches heavily rely on connected-components. The work of [Ponsard et al., 2012] partially addressed the problem by focusing on speech text of a single typewritten font and language for which an OCR system is trained for. Conversely, top-down approaches heavily rely on speech balloons, they start from balloons detection (white or bright colored blobs), followed by mathematical morphology operations, which have been proposed in [Arai and Tolle, 2011, Yamada et al., 2004, Sundaresan and Ranjini, 2012].

The authors in [Sundaresan and Ranjini, 2012] proposed method for text extraction using blob extraction method. This method refined the problem to only detect and extract text situated within a speech balloon (balloon detection has to be done before text extraction). First, each comic image is converted to a binary image by applying a threshold, then gone through some noise removal filter (median filter). Connected-component labeling (CCL) algorithm is then applied to determine sets of pixels which are not separated by boundary. CCL step produces blobs of pixel which can be text balloons or non-text blobs (false detection). The authors tried to reduce number of blobs detected by defining a criteria: if the area of the blob is greater than about 10% of the total image area, it is considered a text balloon. Note that this benchmark is set empirically based on the set of image in the experiment. After this balloon detection, text extraction steps are performed. Previously in [Arai and Tolle, 2011], the authors also proposed a method to detect and extract Japanese character within a manga page. The text extraction is based on their comic frame extraction using blob extraction. In many ways, there was a strong resemblance between the approaches of [Sundaresan and Ranjini, 2012] and [Arai and Tolle, 2011], although they are different group of authors, not to mention some verbatim quotes. In [Sundaresan and Ranjini, 2012], the authors described how the text can be retrieved using OCR. Not much details were discussed on how to perform that OCR. Testing results were provided, but it seems that the data set in the experiment contains too few image (five of them), and the text are in "decent", printed glyph (Comic Sans). It is likely that the method will not work in the case of variety of stylized, artistic text (which are common in comics).

In [Li et al., 2013], the authors proposed an unsupervised method for text localiza-

tion in the sense that it is free of training data. It works with multi-segment characters (such as Chinese and Japanese characters) of different font sizes, or text aligned either horizontally or vertically, and punctuations. The first stage is to generate some of the character strings based on the concurrence of characters, while the fonts and gaps of the adjacent characters within the character string are also obtained. In the second stage, the obtained fonts and gaps of adjacent characters are used to detect the rest of the character strings via Bayesian classifier. Similar to [Sundaresan and Ranjini, 2012], the first steps include converting the image into binary version, and applying connected-component algorithm. However, in [Li et al., 2013] the authors introduced a rule to define if each pair of connected component is vertically/horizontally adjacent, or not. This rule helps to significantly reduce the false alarm. Moreover, the authors provided a second round including some fine-tuning process for the result of the first round, such as font comparison, font propagation, text area merging, etc.

Aiming at a better compression rate for comic SVG file, in [Su et al., 2011] the authors proposed a method for recognizing text elements in raster comic images and use that text elements to provide better compression. The proposed method uses Sliding Concentric Windows (SCW) and Support Vector Machine (SVM) to identify text regions. Then, OCR is applied to recognize text elements in those regions. Instead of encoding the text regions as vectors, the text elements are embedded in the SVG file along with their coordinate values. Based on the observation that text regions in the comic image have significant properties, such as irregularities in texture and abrupt changes in local intensity, the authors adopted SCW to segment text from the comic images. Several features, including aspect ratio, orientation, edge density variation and coverage are calculated and then forwarded to SVM to classify real text regions.

In [Rigaud et al., 2012] the authors make use of the median value of the border page pixels to binarize the image, extract connected component and then classify them into "noise", "text" or "frame" (based on connected components' heights). This method assumes that the page always contains text and that the text background color is similar to the paper background. Note that, although there have been quite a lot of work concerning text localization in comic images, there was no work that deals with text recognition until recently. This is probably due to the lack of dataset and ground truth to evaluate the text recognition algorithms. In [Rigaud et al., 2017], the authors first classified the difficulty levels of text recognition task for Latin script in comic images, and evaluate the performance of different approaches to solve them, including pre-trained OCR and segmentation-free approach.

Characters Detection and Content Understanding

Detecting (which includes localizing and recognizing) the characters in comics is considered a difficult task, mainly because of the abstract expression and large variability and transformation of the drawing (a character can be represented by various expression with rotation, occlusions, different perspective, drawing effect, etc.). Human detection, posture detection and face extractions in real scene images have progressed

a lot during the last decades. However, several studies that are related to comics have shown that human detection techniques are readily applicable to comics character, or need to be trained for specific comic types.

Face Detection and Recognition Approach
Preliminary work about cartoon and comics faces recognition have been carried out in [Takayama et al., 2012], where the authors attempted to localize face regions by using some special features of face in a subset of colored manga, such as the skin color, the special V-shape of the jaw contour and edges of the hair area. In [Cheung et al., 2008], the authors demonstrate and compare face detection results for human-like for colored comic character between (1) low level analysis, here, skin color segmentation with some assumptions about skin color, and (2) using similar approaches as in human face detection, such as Adaboost-based algorithm [Viola and Jones, 2004] (boosted cascade of Haar-like features), or EBGM (Elastic Bunch Graph Matching) algorithm [Wiskott et al., 1997]. It was shown that despite of their imperfectness, the most workable techniques are the training-based methods like Adaboost, and their approaches are more reasonable. However, a big obstacle is that, unlike the availability of real human faces to train, the source of "nicely colored comic" faces is quite limited. With a classifier trained for human face as a complementary detector, the more realistic a comic face is (human face look-alike), the better detecting result for it will be.

In [Lu and Zhang, 2010], based on analysis of clustered group, the authors proposed their retrieval model named "Vi-Thesaurus". Clustered group model is considered due to the observation that, in *anime* (a term for a style of animation originating in, and commonly associated with Japan) or comics some, certain clusters appear at the same time for the same sort of objects (e.g. the clusters shaped by eyes, nose and mouth simultaneously appear in the workspace when retrieving the character's face). The proposed method divides objects into a set of several minority part, and looks upon the part as the letters of a visual semantic vocabulary. Note well that the author seems to mostly discuss the theoretical description of the model, no explicit experiment setups and comparison were addressed.

Local Feature Approach
When it comes to manga, one of the important characteristics is the total lack of color, trading in by much more drawing details and realistic style. Without an important information such as color, the methods performed on manga normally have to resort to different types of feature.

Notable examples that utilize the local features are the work of Sun *et al.* [Sun et al., 2013, Sun and Kise, 2009]. In [Sun et al., 2013], the authors proposed a framework to detect specific (selected, with samples given beforehand) characters, based on local feature matching (here SIFT keypoints are used). The framework includes applying feature extraction (SIFT 128-dimensional feature vector) on a training set of comic pages. The key process is to decompose every pages into panels, where SIFT keypoint vectors are extracted and local feature matching is performed between every two panels. For matching the keypoints, Euclidean distances between feature vectors are applied, if two matched keypoints from two panels are of the same region of the same character, they will be marked as positive keypoints, and if they are matched

but not from the same region, they will be marked as negative keypoints. In [Iwata et al., 2014], the authors discussed the local feature extraction and the approximate nearest neighbor (ANN) search . Also based on local feature vectors extraction and matching, the authors in [Sun and Kise, 2009] proposed a method to reduce the burden of calculation to detect line drawing using hash table.

To our knowledge, no more work that solely employ local features for character detection or retrieval in comics/manga has been proposed: neighborhood features were shown to be generally not suitable. However, dense feature extraction such as Histogram of Oriented Gradients (HoG), combined with common techniques such as sliding windows, proved to be an effective feature in representation and matching. Examples of work following this approach are the work of Sun *et al.* [Sun and Kise, 2010, Sun and Kise, 2013] and Matsui *et al.* [Matsui et al., 2014, Matsui et al., 2015]. In [Sun and Kise, 2010, Sun and Kise, 2013], the authors used Maximally Stable Extremal Regions (MSER) [Matas et al., 2004] or faces detection [Viola and Jones, 2004] for region detection and Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005] as region descriptor. However the main goal was to find the exact or similar copies of the query (for partial copyright protection). Colored comics may be compared to cartoon images sequence, for which a first work based on HOG, SVM and color attributes has been published in 2012 [Khan et al., 2012a].

In [Ho et al., 2013], the authors used graph representation techniques to detect recurring characters. The main idea is that the same character should be represented by similar sub-graphs in different panels where it appears, and inexact graph matching is applied to find redundant structures among the set of graphs. The graph nodes store the local feature, and their edges encode local structure. This approach can also be applied to find other redundant objects in the panels, rather than only comic characters.

Knowledge Driven Approach

In a comics page, if we know the panel positions, then we can make a better guess at the location of the comics characters, as most of the time they are inside the panels, or if we can locate the speech balloons, we have better chance of finding textual content inside those balloons. Similarly, spatial inferences can also be used to infer the comic books reading order, for panels in the page and balloons in the panel follow some certain conventional order [Guérin et al., 2017]. A semantic annotation tool [Hermann et al., 2012] makes use of previous knowledge and consistency information to suggest new knowledge to the user in an interactive way.

In computer science, ontology is a formal representation of the knowledge by a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain and may be used to describe the domain. In theory, ontology is a "formal, explicit specification of a shared conceptualization". It provides a shared vocabulary, which can be used to model a domain via the type of objects and concepts in that domain, and their properties and relations. The ontology has been applied in many different ways. An ontology of comics has been proposed from a philosophical approach in [Meskin and Cook, 2011].

In [Rigaud et al., 2015a], the authors proposed a knowledge-driven system that understands the content of comics by segmenting all the sub-parts. This knowledge-driven system's core is an inference engine that interacts with two sub-systems, the image processing level system, and the expert system. The image processing system first proposes to the expert system assumptions about a set of regions in the image, and assign them the region type (i.e. panel, text, balloon, etc.). The expert system then evaluates these assumptions, validates the correct ones and deletes the others. New information is inferred from valid assumptions, and put in perspective of the domain knowledge. The inferred information can be used for the next iteration to extract more complex elements, e.g. speaking characters, detected by the rule of having the tail tip of the speech balloon pointing to them, and this loop can be run as many times as new information is discovered.

Deep Learning Approach

We can observe a clear trend recently, that the use of deep learning, a sub-field of machine learning, has become extremely popular in image recognition and many other areas because of their superior performance. However, deep learning has just been in the first steps of being applied in comics-related research in computer science. Some of the first work that employ deep learning in comic character detecting are recent ones by Nguyen *et al.* [Nguyen et al., 2017, Nguyen et al., 2018]. In these work, the authors demonstrated the use of deep network to train the comic characters detector based on their annotated dataset. The goal was not to develop a new character detection method but to introduce a state-of-the-art baseline method thanks to the work being the first one to apply deep learning in character detection. It is also understandable that despite the recent popularity of deep learning, applications in comic understanding cannot fully benefit from deep learning yet, since the training require a huge amount of comic database with annotated information, which takes a lot of time and effort to create.

3.3.2 Interaction and Reading Behaviors

Besides content analysis/generation/retrieval, the HCI (Human-Computer Interaction) in comics reading is also an interesting category in the literature. Studies on this topic usually fall into two different sets of problem. The works in the first group stem from a premise that analyzing the reading session of the reader might very well in turn become beneficial in analyzing the content he is reading. For example, as discussed in [Rigaud et al., 2016], while reading comic books, we readers constantly collect different kinds of signal from many sources, both verbal and non-verbal: from the position of speech balloons to the arrangement of panels to follow. Hence in comic book content understanding, it is necessary that we incorporate as much contextual information as possible. The works in the second group focus on creating interfaces between the readers and comics or enhancing the medium to provide better reading experience (may be used to bring the access for impaired people as well). They can be especially helpful as the trend of creating augmented contents is gaining popularity, e.g. to augment comics with new multimedia contents such as sounds, vibration, etc., it is important to trigger these effects at a good timing. In this case, it requires

recognizing when the user turns pages, or estimating the position he is looking at.

One of the important signal to get from the reader is where he keeps his eyes on, the time spent on specific part of a comic page, etc. In order to get information of where and when a reader is looking at some specific parts of a comic, eye tracking systems are employed [Duchowski, 2007]. To our knowledge, one of the earliest work on this topic is [Carroll et al., 1992] in the context of a Visual&Cognition research, where the authors show patterns of short distance between peripheral control of eye movements and deep processes that serve understanding, and that readers tend to look at the artwork to get a clue before reading text. In [Rigaud et al., 2016], we deemed that the extra information was gained by tracking eye-gaze of readers, and it can be used to locate "interesting" zones in a page. The experiment shows some interesting patterns of reading path, and pointed out that most of the time spent on a single page is for reading text, and looking at the face of the comic characters. In the same spirit, it was shown in [Iwata et al., 2014] that a few manually labeled data can make the detection more robust against various postures and facial expression.

The page layout influences the reader at choosing pathway [Cohn, 2013b],thus another way to analyze the interaction between the reader and the comics is to see if they can follow the correct reading order of the panels. While the general rule to follow the pages is to read from right to left, from top to bottom, line by line, following the "reverse-Z" shape (for manga) and Z-shape (for European/American comics), several studies however indicated that in practice, it is not always the case [Cohn, 2013b, Guérin, 2012]. The configuration of the page, the size and the shape of the panels can be so different from one page to the next, and some decisions are purely artistic choice, even if there are certain conventions in comic design.

Cohn presented many work related to comic analysis [Cohn, 2011, Cohn, 2013a, Cohn, 2013c], in terms of interaction between readers and the content, and in the approach of cognitive science, explored that some cognitive tricks can ensure that most of the readers will follow the intended reading path. The artists control the flow of reader's attention via placement of elements, to lead the reader through the pages [Cohn, 2013b, Cohn, 2018].

However, the analysis of interactions between users and comics has only been partially covered (e.g. eye-gaze analysis). The use of other types of sensors with regard to brain activity, muscle activity, etc., thanks to advancement in technology, could help in collecting interesting information to analyze content in comics images or to provide a more enhanced reading medium. For example, concerning interactive comic reading interface, a system was proposed for human-robot interaction while reading comics: hand gesture to rotate objects, flip pages; facial expression to recolor the page [Kang and Ju, 2009].

3.3.3 Comic Book Images Retrieval

In [Lucas, 1996], the authors proposed a rapid-content based retrieval from document image databases. Apply $n-$tuple recognition techniques to searching document image

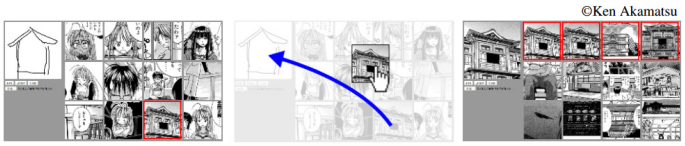


Figure 3.13: Query retouch is a powerful paradigm. User can reuse a retrieved result by dragging the result to the search canvas. With relevance feedback, we can modify either the initial sketch or a query taken from the results of a retrieval.Image excerpted from the authors' paper [Matsui et al., 2014]

is reported. The method involves scanning an $n-tuple$ classifier over a chain-coded of the image. In scanning $n-$tuple systems, the traditional advantage of $n-$tuple recognition i.e. training and recognition speed are retained. (Application searching DB of comic strip image)

In [Rigaud et al., 2014], the authors proposed matching between the query and local regions in comic page images using a descriptor composed of the most representative colors of the query. More recent, the authors in [Yanagisawa et al., 2014] tried to improve detection of faces in manga by features extraction and deformable part model.

Graphs are popular data structures used to model pairwise relations between elements representing parts of the comic image, and graph theory has been used to find redundant color structure in order to localize characters. In [Luqman et al., 2013], the problem of indexing and retrieval of comic images was modeled as a subgraph spotting task.

[Rigaud et al., 2014] a query by example approach that ask the user to select a part of the object he is looking for in one comics image and the system retrieves it automatically everywhere in all the pages of the comics album, assuming that they have been digitized under the same conditions.

For manga, where the color information is usually not available, the authors in [Matsui et al., 2014, Matsui et al., 2015] focused on the retrieval aspect and developed an interesting sketch-based retrieval system that allows query retouch and query expansion. Since a query can be presented as a form of sketch, the author proposed a special feature tailored for sketch queries called FMEOH (Fine Multi-scale Edge Orientation Histogram), which basically is a customized version of HoG, performed on sliding window of different scales over the manga pages. The set of FMEOH features representing a manga page is then converted to binary string to store as index, and to perform manga retrieval.

For now, few tools and dataset have been made available. This is one of the reasons that keeps comics analysis to progress as fast as other field of research of document analysis. The question of how researchers can make publicly available the data images (which are usually copyrighted) is still an obstacle, but indeed worth

considering as it would greatly contribute to the improvement of comics research.

With the analysis and processing of data comes the need of the output results evaluation. In the work by [Guerin et al., 2013], the authors introduced the first database consisted of different digitized comics and manga and the ground truth to evaluate the performance of algorithms on that database. Such a ground truth is made publicly available so anyone can challenge his own algorithm. Besides introducing the construction of the comic database (named *eDBtheque* [2] with the definition of *panels*, *text lines*, *balloons*, the authors also describe the semantic annotation of each segmented object (by a set of predefined metatada). An error evaluation section was also introduced to evaluate the quality of constructing the ground truth since the fact that many people working together in building the database make it almost impossible to have a perfectly homogenous segmentation. By this step, an acceptance threshold is defined for the segmentation performed by each person.

3.3.4 Summary

The effort in this section is to list the recent results that are related to comics image processing, and organize them and in different corresponding categories. In computer science research on comics, a lot of unexplored areas remain, for example, the content generation and augmentation, and there are plenty of potential for other areas to improve, for example content understanding and retrieval. Most of the work in the literature use different copyrighted images which can not be shared publicly. This is a key issue for researchers which can not share, reproduce and compare results on identical data in a collaborative way.

3.4 Conclusion

In this chapter, we presented and discussed the state-of-the-art related to graph, graph matching, graph retrieval, and applications of graph techniques in pattern recognition, as well as Content-Based Image Retrieval (CBIR). We have also surveyed the researches in the literature, especially the recent ones, that are related to comic images analysis, comics understanding and comic image retrieval. The state-of-the-art presented in this chapter provides the necessary elements for placing this thesis within the appropriate context.

By exploring the state-of-the-art work related to all the components that make a CBIR system, we want to review if the current CBIR techniques can accommodate the comic retrieval task. It is clear that the natural difficulties of processing queries that are comic images, are not sufficient to be done with the current approach, and graph representation and graph mining can help with this issue, as will be discussed in the following Chapters.

[2] Available at http://ebdtheque.univ-lr.fr

Some of the challenges of comics image analysis can be highlighted from the above state-of-the-art reviews. According to the results of the reviewed methods, we deem that although many effective approaches of CBIR have been reported, most of the existing methods share a common weakness that they have been developed to deal with rigid objects abstract models such as comic characters or comic object. Moreover, some features like SIFT do not carry enough information in a comic image. Based on the difficulties mentioned above, and on the analysis of state-of-the-art approaches, we can point out a different approach for comic content recognition and retrieval. In the scope of this thesis, we are especially interested in using graphs as a tool to model spatial relationships among image features. This idea is developed in the next Chapter.

Chapter 4

Comics Retrieval Based on Graph Representation

As previously discussed in Chapter 3, graphs are powerful tools to model spatial information in image representation. Following the structural approach, the objects in images (comic characters in this work) are identified by decomposing the image into parts and analyzing the structural relationships between them. As an example of a decomposition of an object into parts, consider the hierarchical relationships among the parts of the silhouette of a person's body: four relatively long extremities (arms and legs) and a blob-like shape (the head) connected to a trunk. By using graphs, we expect to capture such semantic relationship (the body together with the limbs and the head making a structured group). In this Chapter, we present a system for retrieving comic images in Query-by-Example (QBE) based model, using graph representation of the drawing content in the images.

4.1 Introduction

We provide a description of a Content-Based Image Retrieval system for comic image using the tools of graph representation and graph mining, and we show how this approach will address the challenges of retrieving comic images that were mentioned in the previous Chapters.

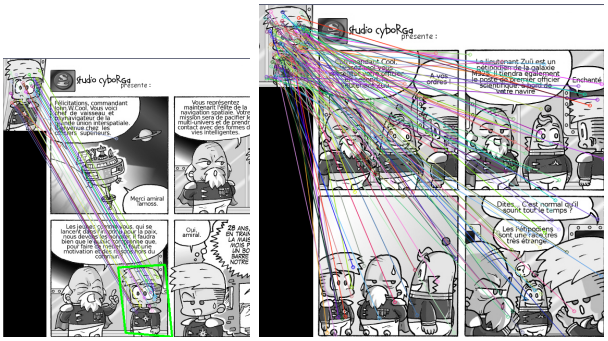
There are different factors that contribute to the difficulties in content-based comic image retrieval, such as rotation, scaling, occlusion, or different expressions of the same character. The visual characteristics of comic images are very different from naturalistic images. First, drawings are very poor in texture information or not textured at all, and it is therefore difficult to extract local features of the target object from the scene. Strictly speaking, the local features can be extracted, but since comic images most of the time do not have smooth and varying gradient intensities, traditional local feature description, e.g. SIFT-like, may not be suited to describing

Figure 4.1: Different expressions of a single character: different facial expressions, face cut in half, different scales, characters behind different objects, complex background, etc. "Cosmozone" ©: , Studio Borga.

comic images. As designed, SIFT allows matching between a patch of an image with itself, even with a slight stretch and perspective change (Figure 4.2(a)). However, even with the keypoints detected, the lack of gradient change makes the keypoint descriptors not reliable to any meaningful match (Figure 4.2(b)). It is therefore necessary to resort to methods taking into account larger patches of the object instead of relying on detected keypoints.

Figure 4.1, containing different frames of the same character, gives an overview illustration of the typical difficulties in this content-based comic image retrieval task. The character we want to retrieve may appear in different panels in different postures, with different facial expressions, at different scales, or only appears partially, due to occlusion or simply to the way it is drawn. Occlusion and overlapping between objects is an inevitable problem in recognition of comic content. Figure 4.3 gives another example of such occlusions in drawn images.

Inter-class and intra-class variations are also the central issues for this kind of pattern recognition problem. In CBIR, especially CBIR of comic images, these variations lead to a problem of retrieving incorrect results for a desired character. For example, there are cases where intra-class variations (i.e. the variations between the sample belonging to the same character due to deformation, rotation) can even be larger than inter-class variations (the difference between different characters) from computer vision point of view. As shown in Figure 4.4, the variations among the characters are quite low (both have yellow skin, orange costumes, perfectly circled and similar eyes, quite similar pattern of the head), and in the quite extreme case of *The Smurfs*, all the main characters are all alike (yet the readers still can recognize who is who based on the small details or by other source of signal such as dialog content or flow from one panel to another). On the other hand, the intra-class variation between postures of the same character can be exceptionally high for many comic titles, e.g. *Obelix* as



(a) (b)

Figure 4.2: (a) SIFT features in a case of almost-exact matching. (b) The keypoints however are unreliable to get other meaningful matches.

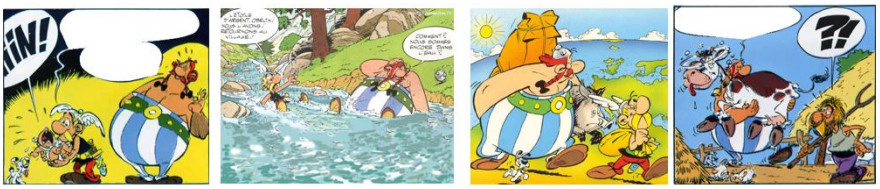


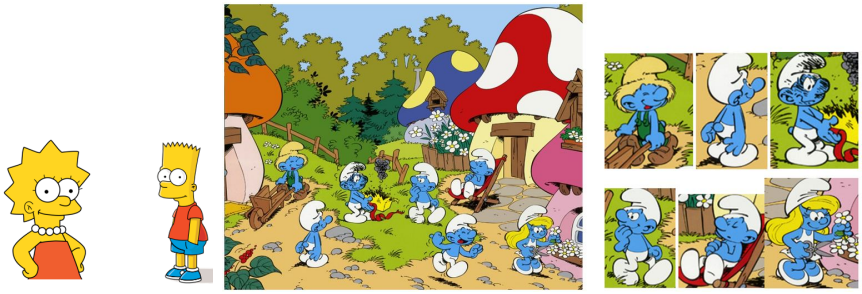
Figure 4.3: Different occlusion level of Obelix, from no occlusion at all to almost covered by different objects.

shown in Figure 4.3. A successful recognition approach should therefore be able to deal with these difficulties.

With scale variation challenge, again, in object recognition from natural images, the scale variation problem can be considered solved partially, with the power of the local features, e.g. SIFT-like features, since these features already take into account the scaling factor by scanning through scale spaces. However, directly applying that approach to comic images does not work, even though the extracted features have local scale information, due to the lack of stable keypoint signature. Thus, an efficient method is needed to cope with these problems.

We detail below our system to represent, index, and retrieve comic images by attributed RAGs. The input is the comic page images, so as the first step, the comic pages are preprocessed to locate the panels inside. Our system then converts each panel into a single attributed graph, as illustrated in Figure 4.5. From the graphs representing the panels, we show how to mine the frequent patterns (the subgraphs) and to build the index of graphs from the frequent patterns. Finally we describe the retrieval step: how to retrieve the corresponding graphs from the query subgraph representing the query image.

The system consists of the following main steps:



(a) (b)

Figure 4.4: (a) Bart and Marge Simpsons, two main characters from *The Simpsons*, (b) Characters from *The Smurfs* shown as an example of inter-class variation.

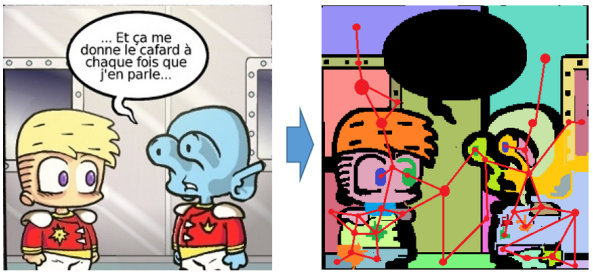


Figure 4.5: The original comic image (left) and the false colored image (right) to illustrate segmented regions and the corresponding RAG.

- It first extracts the graphical structures and local features of each panel.

- For each panel, an attributed Region Adjacency Graph (RAG) is constructed. The graphs representing the panels are then separated into different layers where the layers contain different types of features.

- After that, list of frequent subgraphs for each layer is obtained by using frequent subgraph mining (FSM) technique.

- For indexing and CBIR, the recognition and ranking are done by checking for isomorphism between the graphs representing the query versus the discovered frequent subgraphs.

Figure 4.6: Localized panels in a full page image.

4.2 Graph Construction

4.2.1 Panels Preprocessing and Segmentation

A comic-page image normally consists of several smaller panels (or frames), each one captures a moment of the story. First, the connected-component labeling method proposed by Rigaud *et al.* [Rigaud et al., 2013c] is employed to extract the panels and text balloons. We then review and adjust (if necessary) the locations of a small portion of the panels and balloons which cannot be accurately detected due to artifacts (e.g. if their boundaries are not fully closed). In Figure 4.6, we show how a panel is detected from a comic page image with multiple panels inside.

After localizing elements in a comic page, each panel is segmented into regions, as illustrated in Figure 4.5. Among many segmentation techniques that can be employed in this step, MSER (Maximally Stable Extremal Region) [Matas et al., 2004] was selected, as it has been shown to be superior in terms of stability and performance compared to other techniques which are also based on intensity extrema [Mikolajczyk et al., 2005].

Maximally Stable Extremal Region

SIFT (Scale-Invariant Feature Transform) [Lowe, 1999], works by first locating points of interest (keypoints), then it assigns a descriptor vector constructed as local histograms of image gradient orientations around the points. One of the main disadvantages of SIFT is that it is not affine-invariant. The descriptor itself is oriented by the dominant gradient direction, which makes it rotation-invariant. The keypoints are

searched in scale-space and appear at different resolutions of the image, which makes SIFT also scale-invariant.

An affine-invariant alternative to the SIFT widely used in computer vision applications is the MSER (Maximally Stable Extremal Region) [Matas et al., 2004]. This approach extracts stable regions from the image by considering the change in area with respect to the change in intensity of a connected component defined by thresholding the image at a given gray level. The change of area, normalized by the area of the connected component, is used as the stability criterion. The area ratio is invariant to affine transformations and so does the extracted region after appropriate canonization. Let $X \subset \mathbb{R}^2$ be a domain on which a grayscale image $I : X \to [0, 1]$ is defined. Every image can be represented as a collection of its level sets. A level set of I at some given $t \in [0, 1]$ is the set $\{x \in X : I(x) = t\}$. Topologically, a level set may contain zero or more connected components of dimension zero (points) or one (isolines). The algorithm returns ellipse regions where local binarization is stable over a certain change of intensities as local feature regions. The sizes of the extracted regions are then expanded multiple times, where the expanded regions of small size are discarded.

As each region is represented by a node, over-segmentation may introduce more irrelevant nodes as noise, while under-segmentation may skip descriptive details. Due to the great variety of styles or levels of details in different comic titles, no global setting for segmentation is sought. The parameters of MSER are chosen based on "style signature" of the author. By controlling the parameters of the segmentation, we can determine the upper limit of the number of nodes (and so the size) of the resulting graphs.

To discuss about style, it can be observed that an author tends to generate rather consistent content in terms of drawing elaboration. There are different studies addressing the problem of style classification, but they were proposed mainly for oil or watercolor paintings. Style classification in comics has not been explored much, from neither the aspect of feature design nor classification. Drawing style is an important characteristic for readers to retrieve or manage comic images. Diversity of drawing styles in comics can be perceived quite easily by humans but from the perspective of pattern recognition, rigorously defining "style" is indeed a difficult problem, as "style" is heavily subjective, and in general, not an immediately quantifiable concept. In an early work [Wayner, 1991], although not specifically targeted comics, the author described a simple algorithm for identifying the artist who created a picture. The algorithm relies upon computing and comparing the distribution of long and short lines in the image, and it was tested on seven samples of comic stripes of different authors. Although the algorithm proved to be right more than 95% of the time, the approach was rather primitive. In a much more recent work dedicated to the classification of artistic style in manga [Chu and Chao, 2014], the authors argued that with proper features and statistical analysis, drawing styles can be effectively characterized. As characters and objects in comics are mainly created by lines, the proposed features take into account arrangement, density and magnitude of lines, relationship between nearby lines, and other line properties that constitute an artist's drawing style. It

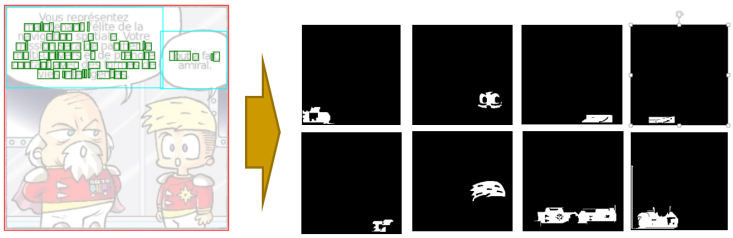


Figure 4.7: Examples of MSER regions extracted from a comic panel (not all regions are shown). Each small box with black background shows a single extracted MSER region, at its location to help visualize the result of the extraction.

was shown that using SVM classifiers, the proposed line-based features allow "style" classification using SVM, from recognizing the difference between boy magazines and girl magazines, to discriminating four different artists' works. It was also suggested by the authors that the ability to classify the drawing style would open up possibilities for novel applications, and facilitate comic browsing and retrieval in a content-based manner [Chu and Chao, 2014].

4.2.2 Local Feature Extraction

In almost all CBIR applications, general visual content including color, texture, shape and spatial relationship between parts of the image are important features to consider. In the following parts, the main features generally adopted in image retrieval are described. For those classical features, we will first give a brief introduction, and subsequently, describe how they are extracted and stored as graph attributes.

In this work, the low-level features to extract include color features, shape features, and spatial relationship features of the extracted regions. While texture is one of the fundamental and important features, it was already mentioned previously that drawings in comics are poor in texture information. This is why besides spatial relationship, we rely on color and shape information only. Node labeling is done by assigning the characteristics of each region to its corresponding node. Each node is characterized by the attributes detailed as below.

Colors

Color is one of the major features used in CBIR systems. This popularity is attributed to the ease in implementation (it is trivial to compute color), and the distinguishing differences between colors. It is a robust feature to changes such as the scene layout or viewing angle. In colored comics, the color information gives the identity of characters and plays a main role for character spotting with speech balloon positions.

Color can be defined using different reference spaces, such as RGB, L*a*b, L*u*v, HSI, YCrCb. Most of acquisition devices, such as digital cameras or scanners, process the digitized images in the RGB format, that is why RGB space is widely used in the applications of image processing. The RGB model can be visualized as a cube. One corner of the cube is the origin $L(0,0,0)$ and each of the three primary colors Red, Green and Blue are assigned an edge to represent the axis from the origin. Any other individual color obtained after combining the red, green and blue components then lie in this coordinate space. The RGB model is limited in representing the full human perception which includes details such as the brightness and purity of a color. However non-linear transformation from RGB to HSI can be used to capture these additional properties.

In the color cube of RGB space, the distance between *blue* $(0,0,255)$ and *magenta* $(255,0,255)$ equals the distance between magenta and *white* $(255,255,255)$. However, the human vision system considers the perceptual distance between *blue* and *magenta* less than the distance between *white* and *magenta*. Thus RGB space is not uniform in with regards to human visual perception, i.e. the relative distances between colors do not reflect the perceptual differences. That means RGB representation has several drawbacks that might decrease the performance of the systems which depend on the distance between the hues and shades.

HSI representation is another popular model. The HSI model can be visualized as the cone obtained when the RGB cube is viewed from the origin. The three channels of the HSI space are Hue, Saturation and Intensity. The Intensity axis denotes the brightness of a color from its vertical position in the cone. The Saturation is computed as the radius from the Intensity axis for any value and reflects the purity of the color. The Hue component is the value of the angle with respect to the Red axis for any point in the coordinate space and represents the tone of a color. The following equations are used for converting from RGB to HSI values:

$$I = \frac{1}{3}(R+G+B)$$

$$S = 1 - \frac{3}{R+G+B}[\min(R,G,B)]$$

$$H = \begin{cases} \theta & B \leqslant G \\ 360-\theta & B > G \\ where & \theta = \arccos \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}} \end{cases}$$

HSI space has been developed as a closer representation to the human visual perception, which can easily interpret the primaries of this space. In HSI space, the dominant wavelength of color is represented by the hue component. The purity of color is represented by the saturation component, and the brightness of color is determined by the intensity component. While the HSI space is already a considerable improvement from the RGB space, and is suitable for a lot of applications based on color images analysis, this color space presents some problems. For example, there are non avoidable singularities in the conversion from RGB to HSI. Besides, this representation is still not a true perceptual system.

The XYZ color space (CIE XYZ 1931), developed by the International Commission on Illumination (CIE) is based on direct measurements of the human eye, and serves as the basis from which many other derivative color spaces are defined. For example, the YUV color (used in the PAL system) of color encoding in analogical video, which is part of television standards. In YUV space, the colors are defined in terms of one luminance and two chrominance components. An alternative of YUV is the YIQ which is used in the NTSC TV standard [Wyszecki and Stiles, 1982].

Another attempt to derive a set of "effective" color features is by systematic analyzing of 100 different color features which have been used in region segmentation of color images. A feature is said to have large discriminant power if its variance is large. The author in [Ohta et al., 1980] has determined that the set of "effective" features are:

$$\begin{aligned} I_1 &= \frac{R+G+B}{3} \\ I_2 &= \frac{R-B}{2} \\ I_3 &= \frac{2G-R-B}{4} \end{aligned} \tag{4.1}$$

Those selected color feature are named as I1I2I3 model. However YUV, XYZ and I1I2I3 are non-uniform spaces, therefore the CIE has recommended L*a*b and L*u*v as uniform color spaces (and they are non-linear transformation of RGB space) [Wyszecki and Stiles, 1982]. There is no simple formulas for conversion between RGB and L*a*b. The conversion must be done via an intermediary color space such as absolute color space (sRGB) or CIE XYZ color space as follows [Reinhard et al., 2001]:

$$\begin{aligned} L^* &= 116 f\left(\frac{Y}{Y_n}\right) - 16 \\ a^* &= 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \\ b^* &= 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \end{aligned} \tag{4.2}$$

where:

$$\begin{aligned} f(t) &= \begin{cases} \sqrt[3]{t} & t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29} & otherwise \end{cases} \\ \delta &= \frac{6}{29} \end{aligned} \tag{4.3}$$

and X_n, Y_n, Z_n are the CIE XYZ tristimulus values of the reference white point: $X_n = 95.047, Y_n = 100.00, Z_n = 108.883$ [Wyszecki and Stiles, 1982].

L*a*b is a color-opponent space, which has been specially designed to be perceptually uniform based on human visual system. The L component closely matches human perception of lightness, and by having it as an independent quantity to control, it can be used to make accurate color corrections without affecting the a and b color twins. Euclidean distances in this space correspond more closely and uniformly

to the differences of the colors perceived by human eyes. This means incremental changes of the same amount in color value should produce changes of about the same visual importance. This Euclidean perceptual property is the main reason for us to choose using L*a*b (or L*u*v). In our scheme to select the feature of graph nodes, the color information of a region is the average value of the color values of all pixels within it, encoded in CIE L*a*b color space.

Moment Invariants

The shape of a region is also an important discriminatory characteristic, hence shape is a common feature used to describe the geometric characteristics of images. In general, shape descriptors are some set of numbers that are produced to describe (and compare) given shapes. Shape features may include aspect ratio, circularity, Fourier descriptors, moment invariants, consecutive contour boundary segments, etc. The shape may not be entirely reconstructible from the descriptors, but the descriptors for different shapes should be different enough that the shapes can be discriminated. Hu was the first to proposed a set of seven Moment Invariants for this purpose [Hu, 1962]. Those formulas have proved to recognize images after rotation, translation and scaling.

What qualifies as a good descriptor? In general, a better descriptor means the difference in the descriptors of significantly different shapes will be greater than that of similar shapes. What then qualifies similarity of shape? It is not easy to answer this one, since it depends on the shapes in real case scenario, and in fact, if we could perfectly quantify similarity of shape, we would already have the perfect descriptor. In practice, this is what descriptors are for: they are designed to quantify shapes in ways that agree with human intuition (or task-specific requirements). Regions can either describe boundary-based properties of an object or they can describe region-based properties.

Moment-based invariants are the most common region-based image invariants, which have been used in many computer vision applications [Flusser et al., 2009, Flusser, 2006,Reeves et al., 1988]. The set of geometric moment invariants introduced by Hu [Hu, 1962] were derived from the theory of algebraic invariant. They consist of groups of nonlinear centralized moment expressions, given by the following expression in integral form:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y)\, dxdy \tag{4.4}$$

where $p, q \in \mathbb{N}$ are order indices, (x, y) are Cartesian coordinates, f is a non-negative continuous function with bounded support so that integration within the available image plane is sufficient to gather all the signal information.

For digital a gray image of size $N \times M$, which is represented as a function $f(x, y)$,

the moment of order $(p+q)$ is defined in discrete form as:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x,y) \tag{4.5}$$

where $p, q = 0, 1, 2, 3, \ldots$
The result is a set of absolute orthogonal moment invariants, which can be used for scale, position, and rotation invariant pattern identification. They were used in a simple pattern recognition experiment to successfully identify various typed characters. The center moment μ_{pq}, which represents a normalized version of the previous one, is defined as:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x-\bar{x})^p (y-\bar{y})^q f(x,y) \tag{4.6}$$

where $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{y} = \frac{m_{01}}{m_{00}}$.

In order to obtain scale invariant moments, the central moments is once again normalized as:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \quad \gamma = 1 + \frac{p+q}{2} \tag{4.7}$$

Because the zero-order moment of a binary object gives the area of the object, the normalized moments are scaled by the area of the object. Second-order moments, for example, are scaled with the square of the area.

$$\begin{cases} M_1 = \eta_{20} + \eta_{02} \\ M_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ M_3 = (\eta_{30} - \eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ M_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ M_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\ M_6 = (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} - \eta_{03})\right] \\ M_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \end{cases}$$

The result is a set of absolute orthogonal moment invariants, which have shown to be invariant under translation, changes in scale, mirror symmetry and rotation and can be used for pattern identification. From the definition detailed above, we can observe how moments respond to transformations and deduce that they indeed have invariant characteristic:

- **Translation:** If we translate the object, we only change the mean, not the variance or higher-order moments. So, none of the central moments is affected by translation.

Figure 4.8: Images of five simple capital characters, their corresponding Hu moments are shown in Table 4.1.

	A	***I***	***O***	***M***	***F***
M_1	2.837e-1	4.578e-1	3.791e-1	2.465e-1	3.186e-1
M_2	1.961e-3	1.820e-1	2.623e-4	4.775e-4	2.941e-2
M_3	1.484e-2	0.000	4.501e-7	7.263e-5	9.397e-3
M_4	2.265e-4	0.000	5.858e-7	2.617e-6	8.221e-4
M_5	-4.152e-7	0.000	1.529e-13	-3.607e-11	3.872e-8
M_6	1.003e-5	0.000	7.775e-9	-5.718e-8	2.019e-5
M_7	-7.941e-9	0.000	-2.591e-13	-7.218e-24	2.285e-6

Table 4.1: Values of the Hu moments for the five simple characters of Figure 4.8, their moments suggest some intuitive assessment concerning the shapes of the characters.

- **Rotation:** If we rotate the shape we change the relative variances and higher-order moments, but certain quantities such as the eigenvalues of the covariance matrix are invariant to rotation.
- **Scaling:** Resizing the object by a factor of a shape S is the same as scaling the x and y coordinates by S. Hence, the $n-$th moments scale by the corresponding power of Sn. Ratios of same-order moments, such as the ratio of the eigenvalues of the covariance matrix, stay the same under scaling, as do area-normalized second-order moments.

The author demonstrated the discriminative power of these seven moments in the case of recognition of different printed capital characters in the paper that introduced the moment invariants (as shown in Figure 4.8). Table 4.1 suggests how these moments behave, for example, the capital letter "I", which is symmetric under 180 degree rotations and reflection, has a value of 0 for moment M_3 to M_7, while the letter "O", which has (somewhat) similar symmetric properties, but has all non-zero moments. Another observation is that the moments tend to be smaller in higher orders. This behavior can be expected as by definition, higher Hu moments have higher powers of various normalized factors, and since each of these normalized factors is lower than one, the products of them to yield higher moments will tend to get smaller fast.

Other than assessing the moments of printed capital letters, a direct generalization is using moments to describe arbitrary, general shapes in other pattern recognition applications. the idea here is that by combining moments, we can produce invari-

Figure 4.9: An extracted region with its scaled and rotated versions

Table 4.2: The first three moments of the shapes shown in Figure 4.9, S1 being the original shape (on the left). S2 to S12 are the scaled, rotated, translated versions of S1, they denote the shapes from left to right, on first and second row.

Shape	M1	M2	M3
S1	0.302983	0.033314	0.008469
S2	0.301319	0.032219	0.008898
S3	0.298361	0.030637	0.011182
S4	0.300351	0.032081	0.008779
S5	0.299193	0.030754	0.016486
S6	0.300596	0.032194	0.008638
S7	0.302571	0.032230	0.015247
S8	0.308926	0.035542	0.015569
S9	0.295788	0.029478	0.007937
S10	0.300463	0.031691	0.017546
S11	0.288031	0.028482	0.006141

ant functions representing different aspects of the image, ones that are invariant to rotation, translation, and scaling. In our scheme of creating graph node attribute to describe the comic page content, we employ these 7 Hu moments due to these invariance characteristics.

For example, the version of a segmented region and its scaled and rotated versions shown in Figure 4.9 will yield the sets of moments with a slight variation in their values. The first three moments of these shapes are shown in Table 4.2.

As mentioned in Chapter 3, one direct and obvious use of region invariant moments is shape matching and classification. Given two binary shapes S_1 and S_2 with their moments (i.e. "feature" vectors), one approach could be to simply calculate the difference between shapes by the Euclidean distance of these vectors, as illustrated in Figure 4.10, Table 4.3 and 4.4. Figure 4.10 shows five different shapes from the segmentation process. The shapes are denoted as S1 to S5, and their moments are shown in Table 4.3. Table 4.4 shows the Euclidean distances of moment values of shapes S1 to S5.

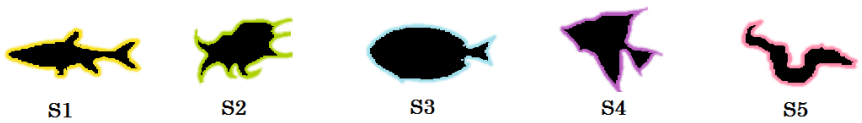


Figure 4.10: Five different shapes from the segmentation process, assigned label as S_1 to S_5.

Table 4.3: Moment Invariants of the shapes S_1 to S_5 in Figure 4.10

	S_1	S_2	S_3	S_4	S_5
M_1	0.3730017575	0.2545476083	0.2154034257	0.2154034257	0.2154034257
M_2	0.0012699373	0.0004247053	0.0002068089	0.0002068089	0.0002068089
M_3	0.0004041515	0.0000644829	0.0000274491	0.0000274491	-0.0000274491
M_4	0.0000097827	-0.0000076547	0.0000071688	-0.0000071688	-0.0000071688
M_5	0.0000012672	0.0000002327	0.0000000637	0.0000000637	-0.0000000637
M_6	0.0000001090	-0.0000000483	0.0000000041	0.0000000041	-0.0000000041
M_7	0.2687922057	0.1289708408	0.0814034374	0.0814034374	-0.0814034374

Table 4.4: Distances between normalized shape moment invariants vectors for the five reference shapes S_1 to S_5. Off-diagonal values should be consistently significant to allow good discrimination.

	S_1	S_2	S_3	S_4	S_5
S_1	**0.00**	0.183	0.245	0.255	0.037
S_2	0.183	**0.00**	0.062	0.071	0.149
S_3	0.245	0.062	**0.00**	0.011	0.210
S_4	0.255	0.071	0.011	**0.00**	0.220
S_5	0.037	0.149	0.210	0.220	**0.00**

However, Euclidean distance being applied directly to the moments may raise a problem. First, the magnitude of the individual moments varies over a large range, so the comparison (matching) of shapes is typically dominated by a few moments, or even a single moment of relatively large magnitude, while the small-valued moments play virtually no role in the distance calculation. Second, these moments are generally very small in absolute numerical value (for example, see Table 4.1). A solution could be adding the weight to the moments, or using another distance (e.g. Mahalanobis distance). In practice, a quite simpler solution is that the values to be used in calculation (i.e. which are the values stored as node attribute as well) are in logarithmic scale, and the distance between descriptors A and B in terms of moments h_i $(i = 1..7)$ is calculated as:

$$d(A, B) = \sum_{i=1}^{7} \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right| \tag{4.8}$$

where $m_i = sign(h_i) \log(|h_i|)$.

Roundness and Compactness

Compactness is an intrinsic property of objects: it can be understood as the relation between a region's area and its perimeter. We can use the fact that, if a 2D shape is scaled (with or without rotation), its perimeter P increases *linearly* with the zooming factor, while its area A increases *quadratically* to deduct that, for a particular shape, the ratio $\frac{A}{P^2}$ should stay the same at different scales. When applied to a closed disc of any diameter, this ratio has a value of $\frac{1}{4\pi}$, so by normalizing it against a filled circle, we create a feature that is sensitive to the roundness, or *circularity* of a shape.

A compactness measure (also known as circularity) is often associated with the old and classical ratio of the area of the shape to the area of a circle having the same perimeter:

$$circularity = 4\pi \frac{area}{perimeter^2} \tag{4.9}$$

This ratio measure can thus be used as a shape descriptor that is invariant under translation, rotation, and scaling. Shapes may have the circularity value in the range of $[0, 1]$. A circularity value of 1.0 indicates a perfect circle (the most compact shape), while smaller values indicate more elongated or spiky shapes.

In image processing, the perimeter of a shape S is defined as the length of its contour, where S must be connected region. As illustrated in Figure 4.11, the perimeter of a region can be calculated by the chain code (also referred to as Freeman code [Freeman, 1974]), which is obtained by following pixels of the region's outer contour. The type of neighborhood relation must be taken into account for this calculation. To compute a chain code, begin traversing the contour from a given starting point x_s in the counter-clockwise direction. Encode the relative position between adjacent contour points using the directional code for either $4-$connected or $8-$connected neighborhoods. The length of the resulting path, calculated as the sum of the individual segments, can be used to approximate the true length of the contour (Figure 4.11).

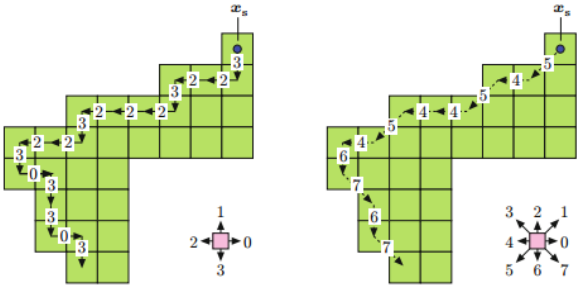


Figure 4.11: Chain codes with 4− (left) and 8− (right) connected neighborhoods. Left: 4−chain code 3223222322303303...111, which has length of 28, right: 8−chain code 5454456767...222, which has length of $16 + 6\sqrt{2} \approx 24.49$ (the length of each step equals 1 for direction codes $c = 0, 2, 4, 6$, and equals $\sqrt{2}$ for direction codes $c = 1, 3, 5, 7$).

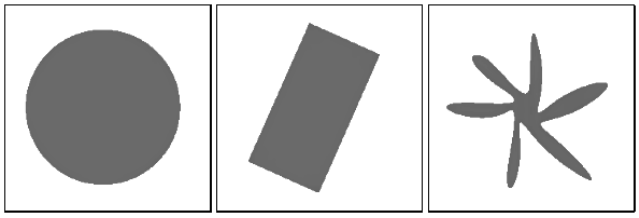


Figure 4.12: Circularity values, from higher to lower, for different shapes. Shown are the corresponding estimates for their circularity value, and circularity value with corrected perimeter factor (shown in parentheses). From left to right: $c = 0.904(1.001)$, $c = 0.607(0.672)$, $c = 0.078(0.086)$.

The real perimeter is systematically overestimated: when using a 4−neighborhood, the measured length of the contour (except when that length is 1) will almost always be larger than its actual length. So in practice, an empirical factor of 0.95 is applied as a simple correction: $P_{calc} = 0.95 P_{measured}$. Figure 4.12 shows the circularity values of different regions as computed with the Equation 4.9 and with correction factor as shown in parentheses.

Summary of Feature Selection

To summarize, features are extracted from regions of a comic panel, and are represented as attributes in graph nodes. A good selection of discriminating features is essential for an accurate CBIR system. However, this selection will vary depending on the feature relevance in a particular query image. This is why a typical system makes use of multiple features and then assigns corresponding weights to denote their respective importance for a retrieval session. While obtaining a much larger number of features from an image is possible and it may even sound desirable, in fact too

many features diminish the distinguishing power of each feature. Besides, including more features means extra computations, additional amount of data and increased complexity, which is not desirable.

It has been shown that arbitrarily increasing the number of features has a direct impact on the quality of retrieval, as it raises the issue of whether or not nearest neighbor searching is even meaningful in such a high dimensionality [Indyk and Motwani, 1998]. It therefore will be of little help to the system to just arbitrarily increase the number of features. In particular, in uniform distribution, the distances between the query and the nearest and farthest neighbors tend to converge as the dimension increases [Indyk and Motwani, 1998].

In this work of content-based comic image retrieval, information about a region in a comic panel, which is encapsulated in a graph node, includes color information in L*a*b space, moments of the region and compactness of the region. As some dimensions are more significant than others in retrieval, different weights are applied to these features in accordance with the amount of discriminating characteristics they carry.

4.2.3 Spatial Relation between Regions

After having encoded the features in each node, we construct the edge of the graphs that represent how the nodes are linked. The edges retain the local structure of the regions (i.e. the relationship between regions). When we compare the query image to the comic image to search against, we are not basing the retrieval result on the similarity in the features, but also the similarity in the local structure. For each pair of regions of the considered panel, we set an undirected edge linking the nodes representing them based on region proximity.

To determine the proximity between two regions a thresholding technique is used. There are several methods for determining the adjacency of regions, for example, but in this case the regions are blobs extracted from MSER, which means they do not necessarily share the same set of pixels as their common border. We therefore use a slightly modified algorithm as shown below. Let A and B denote the two regions obtained from the segmentation process; let C_A and C_B be the sets of the points on the contours of A and B respectively, then A and B are considered as proximate regions if C_A and C_B satisfy the proximity check presented in Algorithm 2. According to this algorithm, two regions not sharing a border can still be considered connected. It also ensures a certain level of "quality" in defining two closed regions: two regions of elongated shapes bordering each other by the tipping point have a lower chance of being marked as adjacent than two regions sharing a larger part of their contours.

The thresholds in this step regulate the sparseness of the resulting graphs, e.g. for very large values of τ_1, edges are placed between all pairs of nodes, while very small values result in almost no edges at all. Since the extra information about the node arrangement is based on the presence of an edge, edges between every pairs of nodes indicate nothing about distance, and thus are against the original

Algorithm 2 Procedure for Region Proximity Check

Input: 2 contours C_A and C_B
Input: threshold of closeness τ_1 and threshold τ_2

function ISCLOSE(C_A, C_B, τ_1, τ_2)
  $s, n \leftarrow 0$ ▷ Number of distances and close distances
  check $\leftarrow$ false
  for each point $\mathbf{i} = \left(x_i^A, y_i^A\right) \in C_A$ **do**
    for each point $\mathbf{j} = \left(x_j^A, y_j^A\right) \in C_B$ **do**
      $d(\mathbf{i}, \mathbf{j}) = \sqrt{(x^A - x^B)^2 + (y^A - y^B)^2}$
      $n \leftarrow n + 1$ ▷ total $|C_A| \times |C_B|$ distances
      if $d < \tau_1$ **then**
        $s \leftarrow s + 1$ ▷ count a close pair of points
  if $\frac{s}{n} > \tau_2$ **then**
    check $\leftarrow$ true ▷ A and B are considered as close regions
  return *check*

purpose. On the other hand, the thresholds have to be also in the range where we obtain connected graphs, and to keep the resulting graphs' density index within suitable range of recognition application. Different values of threshold, corresponding to different resulting set of edges are shown in Figure 4.13. We set the lower bound and the upper bound based on the criteria: the upper bound τ_1 is selected to obtain the connected graph, while the lower bound τ_2 is selected experimentally to be > 0.7 to set an edge between neighboring regions while ignoring anomaly cases, such as two very elongated shape happen to be close together only at their end points.

Edges Labeling

This step assigns labels to the edges to quantify the relationship between connected regions. Based on the assumption that the proportions between parts of a certain character will be preserved, regardless it is drawn from near or far perspective, we propose to use the ratio of surface area between adjacent regions.

Let R_1 and R_2 be two connected regions so that $S_{R_1} < S_{R_2}$, where S_{R_1} and S_{R_2} denote the surface areas of R_1 and R_2. An edge is placed between these two regions, pointing from the smaller region to the larger one (R_1 to R_2). As illustrated in the Figure 4.14, the edge between R_1 and R_2 is labeled by the ratio $\frac{S_{R_1}}{S_{R_2}}$. This value will be used to verify the match in the retrieval stage. For example, assuming the area of R_1 and R_2 in terms of pixel are 720 and 1850, respectively, the label of the edge connecting R_1 and R_2 would be 0.389.

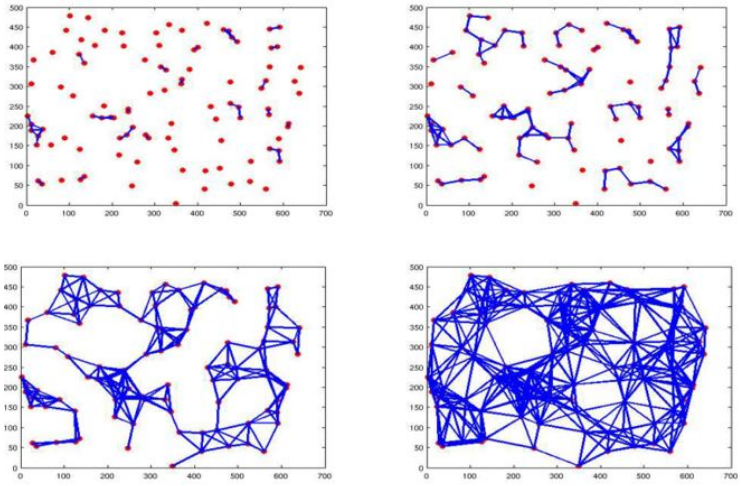


Figure 4.13: Illustration of edge formation with different spatial thresholds.

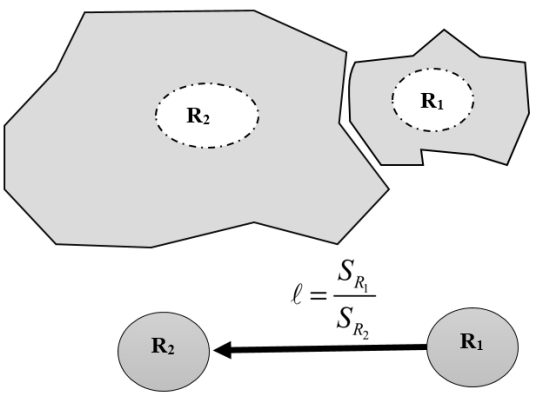


Figure 4.14: Edge labeling between two neighboring regions R_1 and R_2 (shown on top). The label ℓ of the edge connecting two nodes representing R_1 and R_2 is a numerical value determined by $\ell = \frac{S_{R_1}}{S_{R_2}}$ (shown in the lower part).

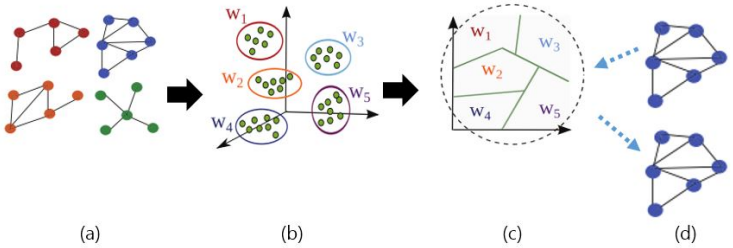


Figure 4.15: (a) A single graph layer, (b) Value space of all the nodes' attribute in that layer, (c) A codebook is built, and the observed values of nodes are separated, (d) Re-assigning the attribute in each node by its corresponding label to the codebook in (c).

Quantization of Node Labels and Multilayer Representation

In evaluating the similarity between nodes, the regions' different characteristics normally do not contribute the same weight in determination of how a node is different from another. For example, in the presence of color feature, retrieval based on color has been shown to outperform retrieval based on shape alone, both in terms of efficiency and robustness [Jain and Vailaya, 1996]. As illustrated in Figure 4.15, we propose the following steps to separate each type of node information into a single graph layer, to gain flexibility in calculating the matching scores from different types of node characteristic. The FSM process requires a single value (label) in each node. If we gather all the regions properties in a node and calculate as a single value, even after normalization, it will not give advantage as separating them and feed them to different FSM processes. For example: a substructure may not be a frequent subgraph with the shape descriptor as node properties, but is a frequent subgraph with color properties. With all the information merged into one single label value, we would not have that flexibility.

- **Step 1:** Each graph is triplicated but each copy only stores one of three information: color information, moment invariants, and compactness value. From the original dataset D of graphs representing the panels in the volume, we obtain the sets D_{color}, D_{moments}, and D_{compact}.
- **Step 2:** In each set, all the observed values of node attributed are gathered and clustered using $k-$means algorithm. An operational definition of clustering can be stated as follows: Given a representation of n objects, find K groups based on a measure of similarity such that the similarities between objects in the same group are high while the similarities between objects in different groups are low. The number of clusters is based on the characteristic of the observed node values (e.g. with color, a digital-born comic image is normally more colorful than a traditional one).

  Each clustering technique has its own advantage and disadvantage. In our work, the clustering step was not a crucial step, and was designed to convert data

points representing graph node's attributions to labels for the matching step later. We therefore did not focus on evaluating the best clustering technique, and our goal was to use one of the classic clustering algorithms for achieving the latter. Spectral clustering takes priority of connectivity instead of geometrical proximity, this approach works better for the case where data is not well geometrically distributed, while $k-$means works better when the distribution of the cluster has a spherical shape (like circle or sphere). We assume (and we visualize that) the features extracted from the comics image is evenly distributed (e.g. the colors) so $k-$means could be a decent choice.

k-means algorithm: Given a set of local features $\{x_1, x_2, \ldots, x_n\}$ extracted from the training set, each local features is a $d-$dimensional vector, the purpose of $k-$means algorithm is to partition those n features into k sets $\{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares. In other words, its objective is to find:

$$\min \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{4.10}$$

where μ_i is the mean of points in S_i, μ_i is also known as the gravity center of cluster S_i. First we initialize k clusters, each of them is represented by a center. An iterative process is then applied to adjust those clusters. After each step we assign all the features $\{x_1, x_2, \ldots, x_n\}$ to k clusters based on the minimum distance, and then recompute the mean of each cluster based on the feature belongs to it. At this step, a feature may move from a cluster to another one. This process is repeated until no more feature moves.

A different version of $k-$means to build the visual vocabulary named hierarchical $k-$means was introduced in [Nister and Stewenius, 2006]. Hierarchical $k-$means is a faster version of $k-$means and it allowed creating a visual vocabulary with over 1 million visual words using SIFT descriptors. The drawback of this method is that it produces deficient clusters compared to normal $k-$means [Nister and Stewenius, 2006]. Another version of $k-$means, which is approximate $k-$means, presented in [Philbin et al., 2007]. This method returns a clustering quality that is very close to normal $k-$means algorithm in a similar computational time to the hierarchical $k-$means technique.

- **Step 3:** The node's attribute is reassigned to its cluster label based on the built dictionary. After this step, we have three layers of graphs representing the comic volume. The graphs in each layer have the same edge structure but different type of attribute in nodes. The matching, and sorting as described in the following section, will use the single label value in each node. (FSM techniques require sorting node and edge attributes to a linear order)

4.3 Indexing

For having an isomorphism or sub-isomorphism between two given graphs, exact graph matching requires that the topology together with the corresponding vertices

are similar. These strict constraints make exact graph matching applicable to only a few applications. Especially in graph-based recognition and retrieval, these strict constraints need to be relaxed to deal with intrinsic variability of the object class, presence of noise, and occlusion problem. Besides, geometric information of the features (e.g. node attributes), which is very important in computer vision applications, is not taken into account for exact graph matching.

Our approach is based on graph mining algorithm to retrieve the resulting images. For large databases, indexing is one of the techniques with which the process of obtaining the result data can be enhanced significantly. One of the input parameters of the mining algorithm is the minimum support threshold. If a support of a subgraph exceeds the user given minimum support threshold, it is denoted as frequent subgraph. The indexing mechanism is based on setting frequent substructure as well by setting them as key in indexing table. Given a query graph G_q, if G_q is a frequent subgraph, then it is indexed and the images containing G_q can be retrieved quickly. If G_q is not a frequent subgraph, it may contain a subgraph that is small enough to be frequent. In this case the images constitute the result image set which contain the frequent subgraph of G_q. This is a candidate set to be processed to discover whether they contain the graph G_q. After discovering the frequent subgraphs in the database the images are indexed using the frequent subgraphs as the indexing key. The key issue is the appropriate choice of the minimum support threshold because it determined which subgraphs are frequent.

The index is constructed from the mining of frequent patterns. Frequent patterns are patterns (e.g., set of items in lists, sub-structures, or subsequences) that appear frequently in a data set. Finding frequent patterns plays an essential role in mining associations, correlations, and other interesting relationships among data. Moreover, it helps in data classification, clustering, and other data mining tasks. We apply *gSpan* algorithm (graph-based Substructure pattern) [Yan and Han, 2002] to mine the frequent subgraphs. It uses a canonical representation for graphs, called DFS (Depth First Search)-code. A DFS-traversal of a graph defines an order in which the edges are visited. The concatenation of edge representations in that order is the graph's DFS-code.

Steps of *gSpan* algorithm is detailed in pseudocode in Algorithm 3. Let D be the dataset of graphs $D = \{G_1, G_2, \ldots, G_n\}$ comprising n graphs, and $minSup$ be the *minimum support* threshold; the goal of the algorithm is to enumerate all subgraphs G_f that are frequent, i.e. $sup\,(G_f) \geqslant minSup$ (in absolute or percentage value).

For example, applying FSM with $minSup = 45\%$ on a graph dataset D containing 1000 graphs returns a set **FP** of k frequent subgraphs (or frequent patterns) $\mathbf{FP} = \{fp_1, fp_2, \ldots, fp_k\}$ where each fp_i occurs in *at least* 450 graphs of D. The size of frequent pattern set decreases with the increase of $minSup$; and with high values of $minSup$, **FP** may even reduce to an empty set. The algorithm starts by scanning through the whole graph database, it then recursively grows the DFS (Depth First Search) code from an empty one with valid, canonical extensions, and stops when the code tree can grow no more.

Algorithm 3 gSpan Algorithm [Han et al., 2011]

Input:

s: DFS code s,
D: a graph data set
minsup: the minimum support threshold

Output: S: the frequent graph set

$S \leftarrow \emptyset$
procedure PATTERNGROWTHGRAPHS($s, D, minsup, S$)
    $S \leftarrow \emptyset$ ▷ Initiate S to an empty set
    return S
    insert s into S

    $C \leftarrow \emptyset$
    scan D once, find all the edge e such that s can be right-most extended $s \oplus_r e$, insert $s \oplus_r e$ into C and count its frequency
    sort C in DFS lexicographic order
    for each frequent $s \oplus_r e$ in C **do do**
        call gSpan($s \oplus_r e$, D, *minsup*, S)

The choice of the frequent subgraph mining (FSM) is mainly based on several reasons. First, due to the vast amount of work related to graph mining problem, it is not feasible to re-evaluate them all (reimplementing different algorithms was indeed a difficult task, especially when apply to a specific problem, such as the one described in this thesis, since each algorithm was originally designed for different purpose, and they require different formats for input, intermediate file and final output). Thus a reliable source for quantitative comparison is to base on surveys of the literature. For example, a recent survey [Jiang et al., 2013] provided an extensive overview about FSM algorithms. It was noted that *gSpan* is arguably the most frequently cited FSM algorithm (and outperforms several algorithms, such as FSG, by an order of magnitude). In comparison with another mining algorithm, such as GASTON [Nijssen and Kok, 2005] experiments show that GASTON is at a competitive level with a wide range of its peers, however it was noted that GASTON performs best when the graphs are mainly paths or trees). Second, based on the comparison charts of the more recent algorithm (e.g. [Vo et al., 2015, Petermann et al., 2017]), it was shown that the improved algorithm had a marginal gain (e.g. in running time, or in memory consumption) compared with the algorithm they originally based in (*gSpan* and variations). Third, the surveys showed that the size of the graph databases in comparison is relatively large (ten thousand of graphs to a hundred thousand of graphs, compared to the size of our database of less than a thousand graph) so we deem that the gain between is significant only when we have a larger database to process.

We apply FSM process to all layers of graphs and obtain a list of frequent patterns for each layer, from smaller to larger structures, together with their frequency and occurrence (i.e. which graphs contain each of these frequent patterns).

Once frequent subgraphs are selected, we construct graph index table to store and retrieve them. It translates the subgraphs into sequences and holds them in a prefix list. Each pattern is associated with an id list: the ids of graphs containing this fragment. Given a query q, we can, based on the index table, enumerate all its fragments up to a maximum size and locates them in the index. Then it intersects the id lists associated with these fragments.

4.4 Querying and Retrieval

We propose a searching approach using a list of frequent structures obtained from Frequent Subgraph Mining (FSM) process, based on the assumption that the same character, or at least some parts of it, would be represented by similar (sub)graphs repeatedly in the panels where it appears. Characters of interest normally do not appear only once or twice, as they are the focus of the storyline. Thus they should have a degree of redundancy and occurrence frequency higher than other arbitrary objects. We limit the retrieval here to content-based retrieval, which means the input is an image query containing the character of interest, and the output is the possible panels where the corresponding character appears. This in turn can be used for auto-annotation (auto-tagging).

4.4.1 Querying by Examples Approach

Unlike the scenario of image searching via web search engine, we can gain some advantage of the contextual information of the text content nearby, such kind of information is generally not available in comics search. Currently the OCR quality in comics is far from perfect due to a lot of reasons (e.g. artistic effects of text, fluctuation in baselines' direction, etc). For a natural and intuitive interface, we suggest a visual-aid query scheme, such as sketch-based or example-based query. While sketching requires a certain level of drawing skill, query-by-example (QBE) offers interactive choosing, and especially suits the context of comics retrieval and browsing. An example in Figure 4.16 shows a list of queries to chose from. In this model, readers will be offered a set of characters of interest, which are cropped from the original comic-page images to serve as examples.

As an adaptation of Bag-of-Words model to graph domain, in this scheme, graphs are considered as combinations of their subgraphs mined from FSM process. So instead of pairwise comparing the query graph with all graphs in the dataset, we only compare the query against the frequent patterns. In other words, we determine if the query has some degrees of similarity with all the panels, and rank the panels by summing signals of matching.

Figure 4.16: Cropped areas from panels to be used as Queries by Example

Query Input. A minimum user interaction is necessary to tell the system what to search within the current comic title. We only asks the user to locate a bounding rectangle around the object of interest. Any pointing device should be easy enough to realize the selection. The query image I_q is undergone the same process as the original comic-page images. After this step we obtain the query graph G_q.

Given a query G_q, we enumerate all its subgraphs up to a maximum size and locates them in the index. Then it intersects the id lists associated with these fragments.

4.4.2 Comparing and Ranking Results

Similarity Measure We then check for occurrences of isomorphism or sub-isomorphism between G_q and each discovered frequent pattern P_{freq}. The (sub)isomorphism is checked using VF2 algorithm [Cordella et al., 2004] (the generations of VF algorithm are also mentioned in Chapter 3, Section 3.1.1. A mapping M between G_q and P_{freq} is subgraph isomorphism if and only if M is an isomorphism between P_{freq} and a subgraph of G_q.

Edge Verification In each edge match of a mapping M, we further verify the match by comparing the edges' attribute. As edge's attribute denotes the surface area ratio between regions, an edge match between edge E_1 and edge E_2 is only verified if the variation between their labels e_1 and e_2 is under a threshold: $\frac{|e_1 - e_2|}{\max(e_1, e_2)} \leqslant \omega$. Here ω is set to tolerate some degree of distortion, and in our experiments we empirically choose $\omega = 0.3$.

For each frequent pattern P_{freq} matched with the query graph G_q, we retrieve all the original graphs containing these frequent patterns. A list of candidate graphs is set by merging the instances of the retrieved graphs. For simplicity, the merging is done by sum pooling, i.e. each graph G_c in the candidate set is associated with the total number of the occurrence of frequent patterns contained in both G_c and G_q, forming a tuple (G_c, f_G):

$$f_{G_c} = \sum \varphi\left(P_{\text{freq}}, G_c\right)$$

where:
$\varphi\left(P_{\text{freq}}, G_c\right) = 1$ if P_{freq} is (sub)isomorphic with G_c
$\varphi\left(P_{\text{freq}}, G_c\right) = 0$ otherwise.

The problem of how to weight features for feature combining was discussed in [Hore and Ray, 2002]. As we consider color information gives more accurate and reliable matching (for the same character drawn in different poses, the shape distortion is generally much more severe), we assign each subisomorphism between P_{freq} and a graph G_c of layer color twice the weight of a subisomorphism between P_{freq} and a graph of other layers. Finally we sort the candidate set to f_{G_c}, and return the panels represented by candidate graphs G_c with the highest f_{G_c} values.

It is possible that some images retrieved by the scheme will fail to meet the expectations of the users. Precision and recall are two quality measures defined to calculate the accuracy of a retrieval paradigm. Precision refers to the proportion of retrieved images that are relevant, i.e., the proportion of all retrieved images that the user was expecting. Recall is the proportion of all relevant images that were retrieved, i.e., the proportion of similar images in the data set that were actually retrieved. The ideal case would be a retrieval system that achieves both 100% precision and recall. The reality is that most existing algorithms fail to find all similar images, and many of the retrieved images contain dissimilar images or false positives.

4.5 Integrating Contextual Information into Indexing and Retrieval Process

A comic book constructs a story with a sequence of panels that have a time-wise coherence. Thus, as in video frame, consecutive panels or panels within a certain proximity may contain the same characters and decorations. This is what we refer to context in this thesis, i.e. the previous and following panels of a given one. It is also common that we encounter groups of panels with similar setup at a "distance" of many pages away from the reference panels.

To assess the similarity between the two panels, we turn it into the problem of assessing the distance between the two collections. We define *graph similarity score* or *intersection score* of graph A and graph B as follows, suppose A is the reference graph:

$$\text{Intersec}\,(A,B) = \frac{\sum_{i=1}^{N} \ell_{\text{FP}_i \in A} * \delta(\text{FP}_i, B)}{\frac{1}{2}\left(\sum_{i=1}^{N} \ell_{\text{FP}_i \in A} + \sum_{i=1}^{M} \ell_{\text{FP}_i \in B}\right)}$$

where $\text{FP}_i \in A$ is the $i-$th pattern of collection A (collection of all the frequent patterns found in graph A, $i = 1, 2, .., n$). Similarly, $\text{FP}_i \in B$ is the $i-$th pattern of collection B.

The size factor $\ell_{\text{FP}_i \in A}$ is the length of $\text{FP}_i \in A$, in terms of number of nodes in $\text{FP}_i \in A$. This is to incorporate the size factor of the frequent patterns: the larger a frequent fragment is, the higher importance it has. Indeed, two graphs sharing a larger common subgraph have more significant matching score than when they share a smaller common subgraph.

The delta function $\delta(\text{FP}_i, B)$ checks if each $\text{FP}_i \in A$ can be found in graph B,

$\delta = 1$ if *Yes*, and $\delta = 0$ otherwise. Determining a certain frequent pattern FP to be in graph A or graph B is done by subgraph isomorphism check [Cordella et al., 2004]. With this metric, the intersection score is incremented by the number of common frequent patterns between the two graphs in comparison. The denominator acts as a normalization factor. This will generate a value between 0 and 1, where 0 is least similar, and 1 is most similar. When a graph is compared with itself, the score will be 1: $\text{Intersec}(A, A) = 1$. It has symmetric characteristic as well, as $\text{Intersec}(A, B) = \text{Intersec}(B, A)$.

In panels grouping or in extended retrieval application, a panel is taken as the reference one. The intersection scores between the graphs representing it and all other panels are computed. The plotting of the relationship between the intersection scores versus panel number illustrates the switching between "scenes". When the scores are quite consistent over a number of panels, we can expect the content of background and foreground belong to the same composition.

4.6 Conclusion

In this Chapter we have presented our proposed system that addressed several problems: how to represent comic images by attributed region adjacency graphs; mining and indexing these graphs to obtain frequent pattern, and can perform comic content retrieval in the domain of subgraph isomorphism. In this approach, the retrieved characters from comic images dataset is done by a Query-by-Example model, i.e. the sample representing them extracted from a given comic title. The separation regions' features into different graph layers to flexibly score matching signals, and the quantization of node values were done for faster matching, as well as to avoid the problem of repeatability. By mining and dealing with frequent subgraphs only, we avoided the heavy computation of direct graph matching. Using this scheme, we will show in the next chapter that the returned results include different poses, scales, orientations, and facial expressions of the same character. We will also verify that by integrating context information, the retrieval is not only a list of random results for a query, but provide more relevant results.

Chapter 5

Experimental Results and Discussion

In this chapter we present the experiment setup and the evaluation of the proposed scheme. In the first section, we present the proposed dataset and other available datasets used in the first part of the experiments. In the second part, we describe the data selection and the evaluation measure used for each step. Ideally, the evaluation is done by checking the output of the proposed algorithm against a ground truth that represents what an ideal output should be. Such a ground truth is usually made publicly available so anyone can test his own algorithm. Particularly, we attempt to highlight the effect and impact of graph-based techniques and their use in retrieval of comic images application. Besides, our work is compared to other work of the literature.

5.1 Datasets

We first describe the datasets used to evaluate comic content retrieval in our experiments.

5.1.1 eBDtheque dataset

eBDtheque datatset is the first annotated dataset and ground truth dedicated to comic analysis in the literature [Guerin et al., 2013][1]. The comic titles included in this dataset have been published between 1905 and 2012. 29 pages were published before 1953 and 71 after 2000. Quality paper, color saturation and textures related to printing technique changes can vary a lot from one image to another. The artworks are

[1] Available at http://ebdtheque.univ-lr.fr.

mainly from France (81%), United States (13%) and Japan (6%). Their styles vary from classical Franco-Belgium, Japanese manga to webcomic and American comics.

The pages themselves have very diverse characteristics. Among all, 72 are printed in colors and according to the authors and periods, there are a majority of the tint areas, watercolors and hand-colored areas. Among the remaining 28, 16 are gray scale and 12 are simply black and white. One album has two versions of each page, one in color and the other one in black and white. We have integrated examples of each of them in order to allow performance comparison of algorithms on the same graphic style by using color information or not. The comic characters or protagonists are specific to each album. Depending on the interpretation, most of them are not humanoid.

The concept of "character" may have different interpretations when used for comics and should be specified. Note that characters in a comic are not necessarily human. However there are trivial "characters" (e.g. characters appearing only once or twice, having no conversation, or having almost no impact on the flow of the story), so we deem that it would be inappropriate to annotate every character instance. We thus chose to limit the annotation to the comic characters that at least has one speech balloon or has significant appearance.

In each title several characters were randomly cropped and chosen as the target objects. For example, the title *Cosmozone* dataset consists of 94 pages, each pages has 3-5 panels, resulting in the total of 371 panels. For each target in the examples offered to readers, we manually identify its occurrence in all the panels. For example, we identified 159 panels containing character *John Cool* as positive samples, and the remainder as negative ones.

5.1.2 SSGCI Competition Dataset

The lack of freely available ground-truthed datasets makes it difficult to test and compare the methods of subgraph spotting. Motivated by that, and in the context of using subgraph spotting for comic retrieval, we initiated a competition for the 23rd International Conference on Pattern Recognition (ICPR 2016[2]). The competition, which focuses on the research problem of subgraph spotting in a database of attributed graphs, was named SSGCI (*Subgraph Spotting in Graph-representation of Comic book Images*). It was a joint effort from the members of the IAPR Technical Committees[3] on "Graphics Recognition (TC10)" and on "Graph Based Representation (TC15)".

The goal of the SSGCI competition is to spot a query attributed graph in a database of attributed graphs, i.e., for a given query attributed graph, the goal is to retrieve every graph in the database which contains this query graph and to provide node correspondences between the query and each of the result graphs. This main challenge of the competition represents an open research problem in graph-based structural pattern recognition. The problems of matching, indexing and retrieval of

[2] http://www.icpr2016.org/site/at-glance/
[3] http://www.iapr.org/committees/committees.php?id=6

graph-based representations of underlying data are actively researched into by the community employing exact as well as in-exact methods.

For this contest, we have selected the comic book images as a challenging data source for extracting and retrieving information from graph representations. The various building blocks of the comics exhibits challenging diversity in their shape and attributes while maintaining enough discriminatory information that permits their identification and recognition. The use of comic book images as the source for extracting the graph representations for the competition not only ensures that there is enough variability in the graph dataset, but it also ensures that there is not too much variability so that the graph matching becomes impossible.

We use a dataset of 500 panels from 4 different comic titles (*Kid Padle, Yoko Tsuno, Asterix*, and *Les 4 AS* – all Franco-Belgian comics), with different drawing style, each title has 125 panels. Figure 5.1 presents some comic book images that have been used for constructing the SSGCI dataset. The SSGCI dataset consists of a database of graphs (each graph represents a panel from a comic book image) and query graphs (each graph represents a comic character in a panel of a comic book image).

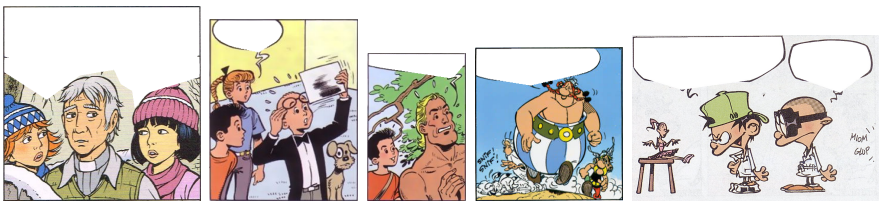


Figure 5.1: Some comic book panel images of one of the four albums that were used in SSGCI dataset. Note that the text were removed as a pre-processing step.

The comic book images have been pre-processed (and checked) to remove the speech text and to segment the panels. After preprocessing step, each panel in the comic book image is represented by an attributed region adjacency graph (RAG) [Le et al., 2015b]. The nodes of the graph represent the MSER regions in the panel and the edges of the graph represent the spatial relations (based on the proximity) between these MSER regions. The steps follow the proposed graph construction scheme, and were detailed in Chapter 4, Section 4.2.

The attributes on the nodes of the graph encode the properties of its segmented regions and the attributes on an arc encode the properties of the relations between its corresponding underlying MSER regions. The presence of a list of attributes on the nodes and arcs of the graphs in the SSGCI dataset ensures that the graphs not only represent the structural information in the comic book images but also the properties of these constituent structural units. The list of attributes on the nodes of the graph includes `region_ID`, `compactness`, `area_in_pixels`, `color(R,G,B)`, `color (L,a,b)`, `bounding_box (height, width,` x, y`)`, Hu-moment values, and `area_percentage`, i.e. the ratio between the area represented by the node to the

area of the panel containing the whole graph (this value expresses how significant a node is in terms of pixel area according to the panel containing it).

The dataset however only contains the graphs, actual images were not provided to the competitors except a small subset as an example for the competitors to have a glance of what kind of images that will be retrieved. That is because in this edition of the competition, we focus only on the performance of the spotting capability of the methods on the graphs constructed from the comic images, while trying to not involved some other training process.

The query characters are extracted from the different occurrences of the comic characters in the comic book images; representing a variety of deformations. The query graphs representing comic characters of SSGCI dataset are then constructed in a similar fashion as detailed above. To be exact, they represent the content inside a bounding box that tightly wrap the interested comic character, a sub-scene inside a larger scene (i.e. the whole panel). Originally we considered having the character cut out precisely, however the "exact" selection of a character were usually heavily subjective (for example, if a character usually holds some object, should that object be cut out; or what if the character is partially covered by some other character) That practice is actually not necessary as well, as in actual scenario, user would not precisely select a character to query as a QBE. That is why we stick with creating the query graph from the bounding box touching the comic character. Figure 5.2 shows some images of queries in SSGCI dataset.

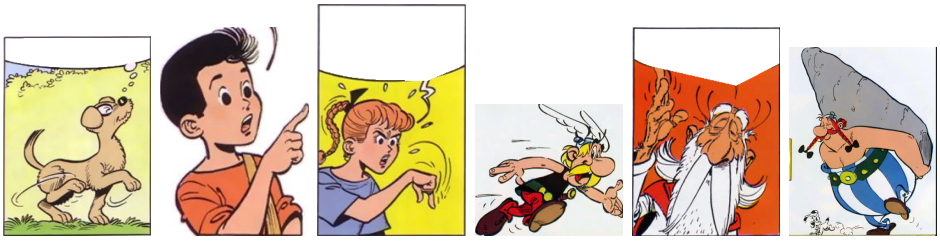


Figure 5.2: Samples of queries provided in SSGCI dataset.

5.1.3 Ground-truth of SSGCI dataset

Each database graph represents a panel from a comic book image and each query graph represents a comic character. It is important to note here that the marking node-to-node correspondences of all the occurrences of a query graph in the database graphs is not possible as we faced several difficulties in generation of the mappings/-correspondences between graphs representing the comic book images at node/edge level (i.e. between the query graph and its result graphs). The ground-truth thus contains the node-to-node correspondence of each query graph for only one database graph. The graphs are generated in a way that there is only one exact occurrence of each query graph in the graph database. The node-to-node correspondence between the query and its only one exact occurrence in database are saved in ground-truth

(for both sample and test sets). However as comic characters frequently undergo deformations/occlusions/change-in-pose etc., the ground-truth doesn't have the correspondences between query graph and all of its occurrences in database.

The graphs in the SSGCI dataset are saved using GraphML[4] format. It is an XML-based format offering an adaptable and flexible means to support interoperability between graph-based tools. The ground-truth information is saved in XML format. The SSGCI dataset is divided into "sample" and "test" sets. The sample set is used for adapting a subgraph spotting method to the graphs in the SSGCI dataset. The test set is used to evaluate the performance of a subgraph spotting method. Table 5.1 presents the details about the contents of the ground-truth of SSGCI dataset. The dataset is freely available for academic research[5].

Table 5.2 shows the occurrence frequency of some selected characters in the four titles which have been used in SSGCI competition. For a single character, different poses were used to query and we calculate the average result to evaluate the retrieval result. Depends on the comic title, the characters may have relatively consistent size and orientation in a few consecutive panels (the same scene), however the poses may still vary a lot from scene to another.

5.2 Experiment on Graph Construction

5.2.1 Graph Mining

Fig. 5.3 shows the result of the FSM process on color layer graph dataset. The lower the minSup, the more subgraph combinations the algorithm has to check. In this figure, the color graph layer has 10 values of node labels. Note that the number of frequent patterns increases significantly as the number of label decreases, as it is easier to create a repeated pattern with fewer node values (for example we assign 5 node labels to moment layer and compactness layer). In practice, we do not need to find all possible frequent substructures, since the most frequent ones will be discovered first.

5.2.2 Retrieval

We evaluate the performance of our method in the context of object retrieval by query by example. A retrieved panel is considered a correct match if it contains the same character/object as the query. Only single query is considered (i.e. there are no cases where several character existing together in a query is). Table 5.2 shows an overview of the dataset: in total we test to retrieve characters in comic panels from 5 different titles. In each title, we annotate the characters appeared (only significant enough characters are marked, i.e. characters which appear only once are not considered). We then generate the statistics of the appearance frequency of the character and

[4]http://graphml.graphdrawing.org
[5]http://icpr2016-ssgci.univ-lr.fr

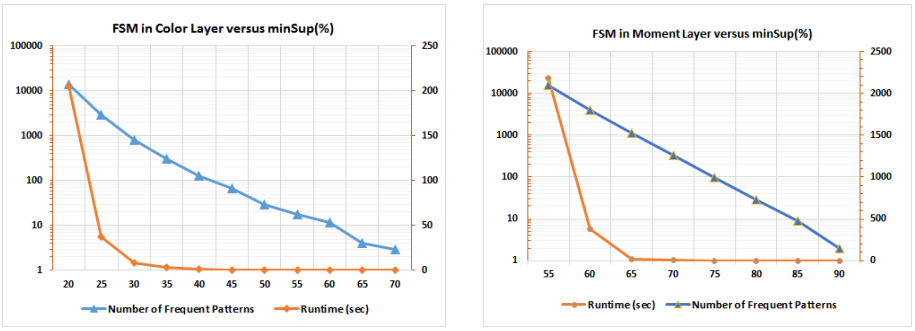
Table 5.1: A summary of contents of the ground-truthed SSGCI dataset.

	Description
Sample set	• 50 attributed graphs in graph database • 10 query attributed graphs • The ground-truth is comprised of 10 XML files (one for each query). The ground-truth XML file for a query attributed graph contains the following information: 1. The ID of the ONE result graph in the graph database for this query graph 2. Node-correspondences between the nodes of query graph and the ONE result graph
Test set	• 500 attributed graphs in graph database • 50 query attributed graphs • The ground-truth in case of exact match is comprised of 50 XML files (one for each query graph) which contain the following information: 1. The ID of the ONE result graph in the graph database for this query graph 2. Node-correspondences between the nodes of query graph and the ONE result graph • The ground-truth in case of inexact match is comprised of 50 XML files (one for each query graph) which contains the following information: 1. The IDs of the ALL result graphs in the graph database for this query graph 2. A list of nodes of each result graph for ALL the inexact matches of the query graph with result graphs from graph database
Evaluation scripts	• Python scripts for computing the graph retrieval results and subgraph spotting results.

sort them from the highest to lowest frequency. Each title has two rows showing the frequencies of its character. For title where the character names can be noted easily, we keep them as they are. For other titles where the character names are not explicit,

Table 5.2: Occurrence of Characters in Different Titles, Ordered by Frequency, and the Retrieval Precision of the 4 Most Frequent Characters in Each Title

	Characters								
Rank	1	2	3	4	5	6	7	8	9
Character	Cool	Zuu	Dokk	Goadec	Apachai	Admiral	Sylvie		
Times Appear	159	79	54	49	38	22	3		
Appear Freq.	42.8%	21.3%	14.5%	13.2%	10.2%	5.9%	0.8%		
Precision	**0.85**	**0.7**	**0.35**	**0.40**					
Character	R1	R2	R3	R4	R5	R7	R6		
Times Appear	46	42	39	32	23	21	17		
Appear Freq.	36.8%	33.6%	31.2%	25.6%	18.4%	16.8%	13.6%		
Precision	**0.46**	**0.36**	**0.4**	**0.44**					
Character	M2	M1	M6	M3	M4	M14	M13	M8	M5
Times appear	65	55	23	14	12	9	8	5	3
Appear Freq.	52%	44%	18.4%	11.2%	9.6%	7.2%	6.4%	4%	2.4%
Precision	**0.23**	**0.28**	**0.47**	**0.31**					
Character	G2	G1	G7	G3	G8	G9	G6	G5	G4
Times appear	91	81	14	9	9	8	6	6	3
Appear Freq.	72.8%	64.8%	11.2%	7.2%	7.2%	6.4%	4.8%	4.8%	2.4%
Precision	**0.78**	**0.71**	**0.28**	**0.23**					
Character	N2	N1	N3	N6	N18	N13	N10	N15	N5
Times appear	43	41	27	14	12	7	7	7	5
Appear Freq.	34.4%	32.8%	21.6%	11.2%	9.6%	5.6%	5.6%	5.6%	4%
Precision	**0.51**	**0.36**	**0.31**	**0.15**					



(a) (b)

Figure 5.3: Number of frequent patterns (left vertical axis), and run time (right vertical axis) for FSM process versus minSup (%) (horizontal axis). The axis for number of patterns is in logarithmic scale. (a) FSM process on color layer, (b) FSM process on moment layer

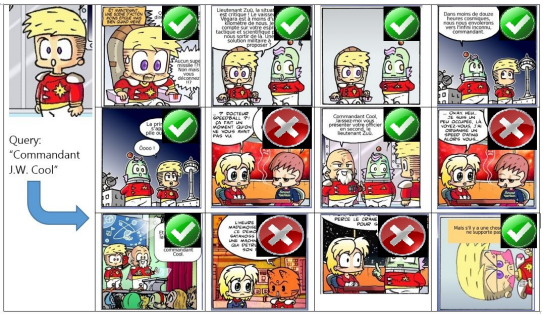
we assign an alias to each single character during the construction of the ground truth. For comparative study, the last line of each group shows the precision rate in retrieval of the top four characters that appear the most frequent in each title. The precision is calculated by counting the correct samples within the top 20 results returned for each query. For each selected character, three different poses of it are chosen, and the precision value is the average value.

Since the time needed to mine frequent patterns grows exponentially as we accept smaller subgraphs (i.e. more frequent, less distinctive), we set a cut-off threshold to yield the number of frequent patterns to be several times the population of the graph dataset (approximately 200 frequent patterns), so the system can run in acceptable time to be used as a retrieval application. The system yields the results by matching each of the layer of the graph representing the cropped area used as queries, versus the list of frequent subgraphs mined with minSup value of 55% to 65%.

Table 5.3 gives the result of precision and recall rate of the retrieval of 4 characters in *Cosmozone* title. Besides Recall rate, we show average Precision rate and Precision @10 and @20 measure how correct the top 10 and 20 returned results are. If, e.g. precision @10 of sample *Cool* is 0.6, a reader would receive 6 panels containing this character in the top 10 results. In Figure 5.4(a) and 5.4(b), we show an instance of the retrieval result for 2 different queries with different retrieval qualities.

As this system is a kind of *comics browsing*, a user is unlikely to browse through tens of returned pages. Even in the case most of them are the "correct" ones, the user would naturally want a more advanced or refined search. Thus the top results are the most interesting and important ones. As shown in Table 5.3, while the first two queries have rather good retrieval result: 8 out first 10 results are correct, the other two queries have the retrieval and precision rate dropped down significantly, it can be explained by their occurrence frequencies (quite lower than that of the first two),

(a) 5 panels retrieved with the QBE shown in the leftmost frame, the 3rd and the 5rd results are the accurately retrieved ones



(b) 12 out of top 20 retrieval results for the character in the query at upper-left corner. Results with green checkmark are the correctly retrieved ones

Figure 5.4: Example of retrieval results using QBE using graph mining method

Table 5.3: Recall and Precision Results of Several QBE

Query	*Cool*	*Zuu*	*Apachai*	*Akira*
Recall	54.86%	73.15%	50.31%	38.7%
Precision	74.1%	50.08%	34.24%	44.1%
Precision @10	0.8	0.8	0.4	0.3
Precision @20	0.85	0.7	0.25	0.35

thus the matching with the frequent subgraphs is actually the match with frequent subgraphs scattered over the whole dataset, and even the matching score from the top candidate graphs cannot boost them to be prominent, resulting in more irrelevant results. However this situation can be improved as we involve the relevant context information for a second round retrieval, as discussed below in Section 5.3.4.

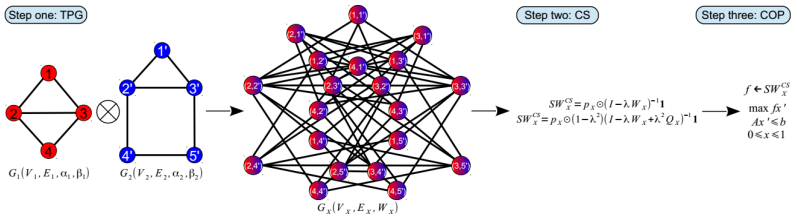


Figure 5.5: Outline of the Tensor Product Graph-based method. Step one: computation of the tensor product graph (TPG) G_X of two operand graphs G_1 and G_2, here $\otimes$ denotes the tensor product operation of two graphs. Step two: algebraic procedure to obtain contextual similarities (CS). Step three: constrained optimization problem (COP) for subgraph matching.

5.3 Retrieval with Different Graph Spotting Methods

In a previous section introducing the datasets (Section 5.1), we introduced the SSGCI datasets and competition. Besides the objective of creating more available datasets, we organized competition to invite other authors to test their methods on the graph representation of the comic data set as well. Our target audience are these researchers working on graph matching using exact as well as inexact approaches, who would like to participate in this competition to get their methods benchmarked with respect to the other state of the art methods. The comic panel selection process also ensures that while there is enough variability between different instance of a character, there is not too much variability so that the graph matching becomes too difficult.

In this section we will introduce the outline of the methods used by the participants, and compare with the results of retrieval from the proposed method in this thesis.

5.3.1 Method proposed by Participant 1: Tensor Product Graph for Inexact Subgraph Matching

This method attempts to take advantage of contextual information of nodes (i.e. neighboring structures) to make subgraph matching more robust and efficient. A second key component of this method is the formulation of subgraph matching with approximate algorithms. The authors therefore proposed an inexact subgraph matching based on *tensor product graph* (TPG): given two attributed graphs, it is quite straightforward to get the pairwise similarities and assign them as weights on the edges of TPG (Step one in Figure 5.5). Then one can think of having a random walk from node to node considering the weights on the edges as the probabilities to proceed to the next node. Finally, the probabilities of finishing a walk at each of the vertices are accumulated, which are referred to as *contextual similarities*, where the context of each node is the set of its neighboring nodes. This procedure essentially diffuse the

pairwise similarities in the context of neighboring nodes.

Since the edges of TPG contain the pairwise similarities between nodes and edges, the consideration of longer random walks on TPG re-evaluates the pairwise similarities between the nodes of the operand graphs in the context of other nodes. This random walk procedure essentially takes into account higher order similarity information, which can be obtained by simple algebraic operation (discounted exponentiation and summation) of the adjacency (or weight) matrix of the product graph (step two in Figure 5.5). A similar phenomenon is termed *diffusion on tensor product graph*, which is well known to capture the geometry of data manifold and higher order contextual information between two objects [Coifman and Lafon, 2006,Yang et al., 2013]. These works have shown that considering the manifold structure or context together with pairwise comparisons significantly improves ranking/retrieval performance [Yang et al., 2013], which acts as their further incentive towards this direction.

Therefore, to tackle the problem introduced in the contest, the authors proposed to formulate subgraph matching as a node and edge selection problem in Tensor Product Graph. To do that, they use the aforementioned contextual similarities, and formulate a constrained optimization problem to optimize a function constructed from the higher order similarity values (step three in Figure 5.5). The optimization problem is solved with linear programming (which is solvable in polynomial time). The higher order contextual similarities allows the relaxation the constrained optimization problem in real world scenarios. The detail description of this method can be found in [Dutta et al., 2018].

5.3.2 Method proposed by Participant 2: Minimum Cost Subgraph Matching

This approach relies on the Minimum Cost Subgraph Matching (MCSM) described in [Lerouge et al., 2016]. It is based on an integer linear program, in order to find the lowest cost association between a query graph and one of the subgraph of the target graph.

The global matching cost is the sum of costs of individual edit operations. The permitted edit operations are the matching of a query node (or edge) to a target node (or edge) or the deletion of this node (or edge). A cost that depends on the labels of the matched or deleted node or edge is associated with each edit operation. The integer linear program aims at minimizing this cost while respecting some constraints. These constraints imply that every node or edge of the query graph is either matched once or deleted. They also guarantee that each node or edge in the target graph is matched at most once. The solving of this integer linear program is performed by a mathematical solver. This simultaneously solving computes the matching between the query graph and the subgraph of the target graph and its associated cost. The solution of the Minimum Cost Subgraph Matching problem described above depends on the definition of costs for elementary edit operations.

The solution of the MCSM problem described above depends on the definition of

costs for elementary edit operations. In the context of the SSGCI contest, the cost associated with a matching operation has been defined as the weighted $L2-$norm between the node or edge labels, and the cost associated with a deletion operation has been defined as the weighted $L2-$norm of the deleted node or edge. As there was no training provided, it was not possible to determine the optimal weighting scheme for node or edge matching or deletion. So, four different settings for the weighting scheme associated with edit operations have been empirically defined. These weighting schemes were involving the labels given in Table 5.4.

Table 5.4: Labels involved in the computation of cost of edit operations

vertex labels	edge labels
R, G, B L*a*b height, width compact	dx, dy

Given a query graph, one of the objectives of the contest was to determine which of the target graphs were containing an actual distorted occurrence of the query. For each of the weighting scheme, the MCSM cost has been computed between the query and every target graph. The target graphs are more likely to be relevant if the MCSM cost is low. It has been decided to keep a graph as candidate if the MSCM cost was below $m - 3\sigma$, where m and σ are respectively the mean and standard deviation of MCSM costs between the query and all the target graphs for the considered weighting scheme. Each of the four weighting schemes is then associated with a list of target graphs that are candidate to be relevant. A target graph has finally been predicted as relevant if it is considered as a candidate for at least two weighting schemes. The detail original method described here can be found in [Lerouge et al., 2016].

5.3.3 Evaluation Protocol

The evaluation protocol of subgraph spotting result on the dataset of graphs constructed from comic images is two fold. On one hand it evaluates the capability of a subgraph spotting method for graph retrieval and on the other hand it evaluates the quality of the proposed node correspondences (between query graph and a result graph).

Evaluating graph retrieval capabilities

A first evaluation of the methods in consideration (the participants' methods and the proposed method) is performed by employing the classical precision and recall measures for evaluating its graph retrieval capabilities.

The precision and recall as defined for the SSGCI competition are given measures

below in Equation 5.1 and Equation 5.2, for a query graph:

$$\text{Precision} = \frac{\text{Num of relevant graphs} \cap \text{Num of retrieved graphs}}{\text{Num of retrieved graphs}} \tag{5.1}$$

$$\text{Recall} = \frac{\text{Num of relevant graphs} \cap \text{Num of retrieved graphs}}{\text{Num of relevant graphs}} \tag{5.2}$$

After computing the precision and recall values for each of the query, the average precision and average recall values for the given set of query graphs in test set, are used for reporting the retrieval capability of a subgraph spotting method.

Evaluating the subgraph-spotting capabilities

The subgraph-spotting capabilities of the three methods will be evaluated by calculating the quality of the exact and/or in-exact matching.

The subgraph spotting capabilities of a method is evaluated by calculating the quality of the exact and/or in-exact matching that it provides:

- If a subgraph spotting method provides the node-to-node correspondences between the query and the result graph (exact matching), a score is calculated based on the node-to-node correspondences provided by the method and the ground-truth by using Equation 5.3.

$$\text{Score} = \frac{\text{Num of correct node correspondences}}{\text{Total num of nodes in query graph}} - \text{Penalty} \tag{5.3}$$

  where the "Penalty" is computed from the incorrect and/or missing node correspondences, by using Equation 5.4:

$$\text{Penalty} = \frac{\text{Num of incorrect node correspondences}}{\text{Total num of nodes in query graph}} \tag{5.4}$$

- If a subgraph spotting method provides only a list of nodes of the result graph where the query graph is spotted (in-exact matching), the score is calculated based on the quality of overlap between the list of nodes provided by the subgraph spotting method and the ground-truth by using Equation 5.5 and Equation 5.6.

$$\text{ScoreP} = \frac{\text{nGT} \cap \text{nR}}{\text{nR}} \tag{5.5}$$

$$\text{ScoreR} = \frac{\text{nGT} \cap \text{nR}}{\text{nGT}} \tag{5.6}$$

  where, nQ is the set of nodes in query, nR is the set of nodes in the result graph provided by method and nGT is the set of nodes in the Ground Truth for the result graph.

Table 5.5: Average Precision, average Recall, average ScoreP and average ScoreR results of subgraph spotting on SSGCI dataset (500 graphs).

	Method 1	Method 2	Proposed Method
Avg. Precision (%)	24.73	75.40	45.60
Avg. Recall (%)	4.70	9.80	20.11
Avg. ScoreP (%)	74.43	82.18	80.71
Avg. ScoreR (%)	73.68	80.71	91.24

Table 5.5 presents the average Precision, average Recall, average *ScoreP* and average *ScoreR* results of the three subgraph spotting methods (two from the participants and the proposed method) over the 50 query graphs. The methods performed relatively good in terms of precision of graph retrieval (as shown in Figure 5.6) as compared to the recall of graph retrieval (Figure 5.7). These scores can be explained by the fact that most of queries have multiple instances in the graph database and the methods did not retrieve all of them, which is understandable as most of times top$-n$ results are sufficient in a retrieval system. According to the result, different approaches of subgraph spotting and subgraph mining can be used on the graph dataset proposed for the content-based retrieval task. The proposed method based on merging the frequent patterns has a better retrieval rate compared to the other two methods, however Method 2, as a trade-off to the low recall rate, (based on [Lerouge et al., 2016]) has a significantly better average precision record.

The quality of subgraph spotting (or focused retrieval) results is positively encouraging for all the methods (Figure 5.8 and Figure 5.9). The results in Figure 5.8 and Figure 5.9 as well as the two bottom lines in Table 5.5 show that these methods keep into consideration the structural properties of graphs (i.e. underlying comic content) while computing the similarity between a query and result graph. Here, the average scores are calculated as follows: for each query, each method returns different set of graphs from the graph database representing the comic images. For each correctly retrieved graph, we compare the node list representing the expected character, versus the nodes reported by the method. The *scoreP* and *scoreR* for each query is the average result of the node-level precision and recall of each correctly returned graph. This means while the recall rate indicate that the methods ignored many instance, but the returned results are highly relevant, based on the graph structure and the attributes in the nodes.

5.3.4 Retrieval with the Context Involved

We base this step on the following assumption: by representing each graph as a collection of smaller frequent subgraphs, the content it represents has some invariance to translation and rotation, and varies slowly under minor change. To assess the similarity between the two panels, we turn it into the problem of assessing the distance between the two collections.

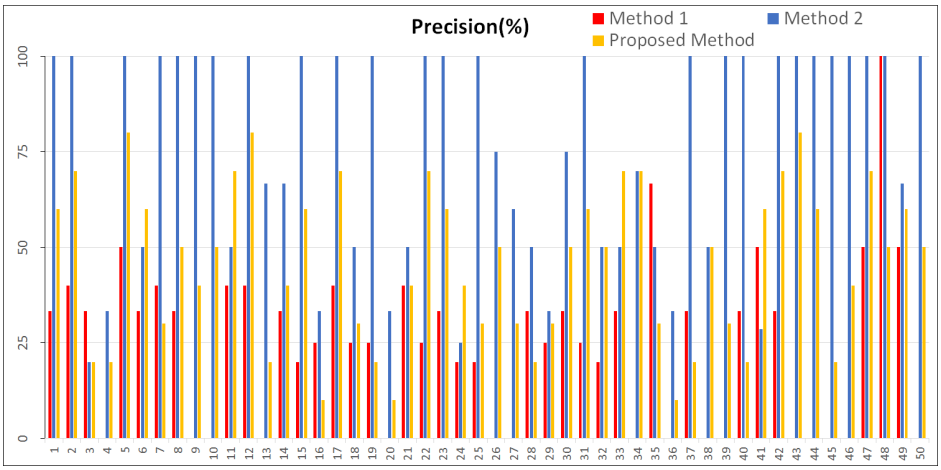


Figure 5.6: Precision results of graph retrieval on SSGCI dataset, the vertical axis shows the precision score in percentage value, the horizontal axis shows the corresponding results for each query ID from 0 to 49 of the three methods.

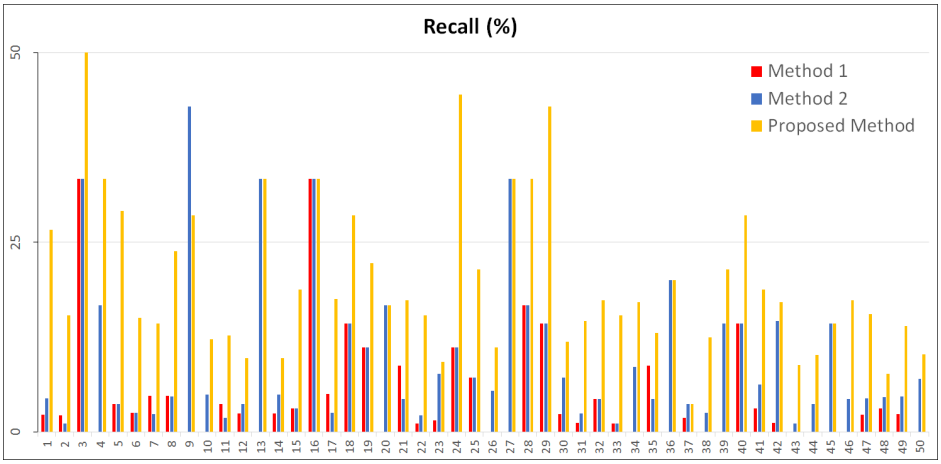


Figure 5.7: Recall results of graph retrieval on SSGCI dataset. The vertical axis shows the recall score in percentage value, the horizontal axis shows the corresponding results for each query ID from 0 to 49 of the three methods.

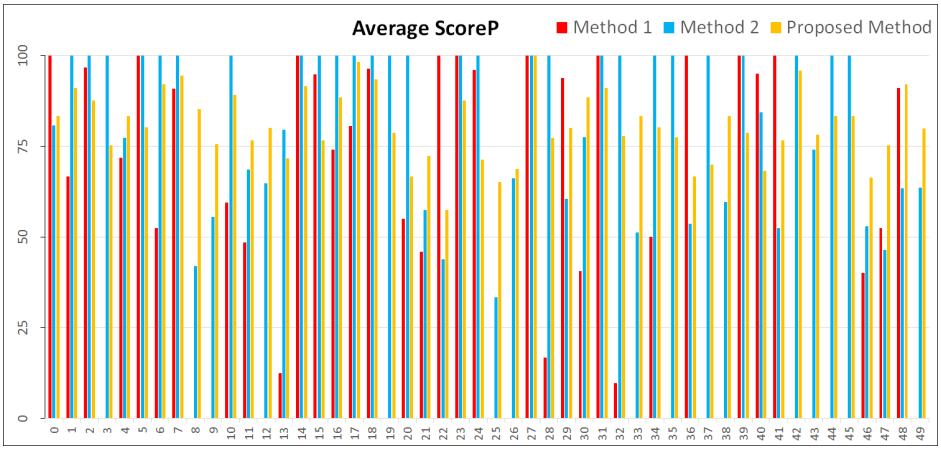


Figure 5.8: Average of "ScoreP"s of retrieved graphs for each query. The vertical axis shows the score in percentage value, the horizontal axis shows the corresponding results for each query ID from 0 to 49 of the three methods. For each query, each method returns a set of graphs, for each correctly retrieved graph, the node-level precision based on the reported nodes and the nodes representing the expected character is calculated to get an average score.

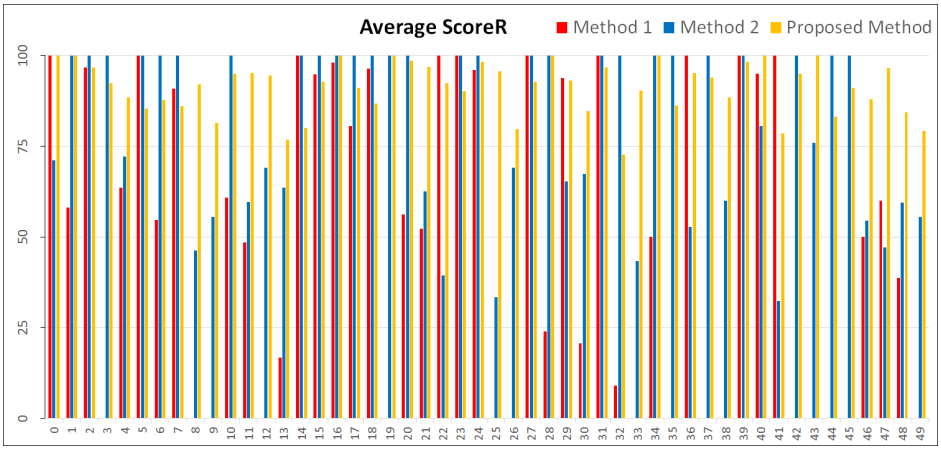
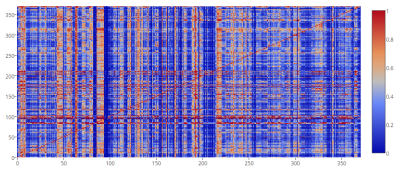
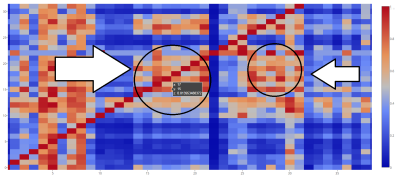


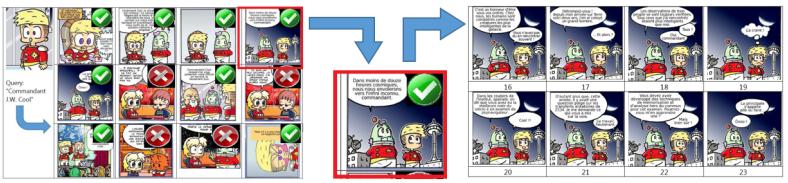
Figure 5.9: Average of "ScoreR"s of retrieved graphs for each query. The vertical axis shows the score in percentage value, the horizontal axis shows the corresponding results for each query ID from 0 to 49 of the three methods. For each query, each method returns a set of graphs, for each correctly retrieved graph, the node-level recall based on the reported nodes and the nodes representing the expected character is calculated to get an average score.



(a) 2D map of correlation scores between all pairs of panels in the dataset. Each axis marks the IDs (position) of the panels in the natural reading flow of the title. The color bar shows the correlation score, with the red end represents higher score, and the blue end represents lower score.



(b) A zoomed-in section of the map showing the similarity score of between panels of the first 35 panels. Each axis marks the IDs (position) of the panels in the natural reading flow of the title.



(c) Retrieved results by QBE (left), one correctly retrieved panel (middle), and its most visually similar panels using context information (similarity score).

Figure 5.10: Similarity map of the panels in a title (top row) and retrieved result of the second round using similarity information (bottom row).

We calculate the correlation score between all pairs of graphs as detailed in Section 4.5. A heat map is generated (Figure 5.10) to visualize the location of the panels which has higher similarity score. The diagonals on Figure 5.10(a) and 5.10(b) represent the similarity scores of each panel with itself, and contains only the scores of 1 (highest score possible). The warmer ranges of locations are the ones that mark the possible similarity in content between two panels.

Figure 5.10 shows an example of how the relevance between panels – represented by the warm regions – is distributed over the heat map. A zoomed-in section of the map shows the similarity score of between panels of the first 35 panels (Figure 5.10(b)). Other than the diagonal, we consider another warm regions: the regions in the 2 circles. The area inside the circle on the left corresponds to the panel with ID from 16 to 23: their scores show that the panels in that strip are quite similar to each other, and this can be verified visually. The circle on the right shows that we can also retrieve similar panels, not in consecutive order. After leaving the high similarity area of panels 16 to 23, the similarity scores drop, but they increase again for the range of panels from 28 to 31. Visually, the content and arrangement of the comic-page

images generate the higher similarity scores between a group of panels that are close to each other in a natural reading flow, which correspond with the score map (for example, as a zoomed-in area of heat map (Figure 5.10(b) and Figure 5.10(c)).

5.4 Conclusion

In this Chapter, we presented the datasets, the experiments, as well as some discussions on the retrieval results of the proposed graph-based method, which is the representation of comic-page images by attributed region adjacency graphs, and how to retrieve characters from comic-page images dataset using query-by-example model. By mining and dealing with frequent subgraphs only, we avoided the heavy computation of direct graph matching, and the low repeatability problem in exact matching. We also provided the comparison of the retrieval performance on the graphs constructed from the comic image, between the proposed method and the two other methods. The work towards this competition presents our recent work on the problem of subgraph spotting in graph-representation of comic book images. We presented a new dataset, which is freely available for academic research so that the researchers can benchmark their methods with respect to the other methods. Second, it presents new unpublished results of two state-of-the-art methods on this new SSGCI dataset. The experimental results show that our approach presented in this work is indeed practical for content-based comic search, and thus can be a great potential in improving comic-reading experience. Based on our proposed graph representation and the proposed matching scheme between the query and the mined frequent subgraphs, the system can retrieve different instances of comic characters with high relevance and reasonable recall rate.

Chapter 6

General Conclusions

This chapter summarizes each chapter by revisiting their main content and contributions. An overview of the future research possibilities in the area of comics analysis and understanding is also provided.

6.1 Summary and Contributions

In this thesis we have presented the graph-based approaches for comic book image retrieval. Chapter 1 gave a brief overview on (1) Content-based image retrieval problem, (2) structural representation approach in pattern recognition, and (3) the evolution of comics, from its creation to the 21st century with the impact of the Internet, its market place and the growing interest of the investigation of this research field. In this Chapter, the interesting and challenging objective of creating a system to provide an enhanced comic reading experience is addressed, of which the problem of content-based comic image retrieval is one of the core problem.

In Chapter 2, we introduced important definitions, concepts, terms and notations, that were used in this thesis. These include common concepts of graph, subgraph, and attributed graph, as well as important features of graphs and a section introducing important concepts on the graph mining.

In Chapter 3, an overview of the state-of-the-art has been presented, which we separated into three categories accordingly to what has been introduced in Chapter 1. Here we provided a literature review on graph matching, graph mining methods along with the advantages, disadvantages in different scenarios. CBIR is a vast research area and has many open questions and challenges, that is why literature review on CBIR was given a significant amount in this chapter too. Designing a CBIR system involves choosing particular feature representation techniques, optimal dimensionality and reliable similarity functions in order to achieve best results. The ultimate aim is to reduce the gap between semantic information in the image and the extracted

low-level features. We then highlighted the challenges of comic book analysis due to their intrinsic complexity of this specific content. Studies related to comic book image analysis have been reviewed.

Then, in Chapter 4, we have introduced our approach of CBIR for comic book image content using our proposed graph representation and mining technique. In this Chapter we presented our proposed system that addressed several problems: how to represents comic images by attributed region adjacency graphs; mining and indexing these graphs to obtain frequent patterns, and can perform comic content retrieval in the domain of subgraph isomorphism. In this approach, the retrieved characters from comic images dataset is by query-by-example model.

Finally in Chapter 5, we have provided an experimental evaluation of the method proposed in this thesis. We also introduced the dataset used in the SSGCI competition, to evaluate methods based on graphs to retrieve characters in comic images. While one participant's method (Method 2, based on [Lerouge et al., 2016]) may have performed with a better precision scores on certain queries, that score is a trade-off to the significantly low recall rate. The proposed method, on the other hand, while not reaching a very high precision score, has a significantly better average recall record. We deem that a better recall is more favored, as it can be equipped with other techniques such as query retouching or query expansion.

In terms of scientific contributions, by constructing and experimenting on a CBIR system using graph representation which targets a challenging type of document image such as comic image, we could reaffirm the power of the structural methods in pattern recognition, even for the task of retrieving abstract content. While the use of graph representation in computer vision has always been with extensive computation, in this thesis we showed that the problem of computation can be tackled by resorting to frequent subgraph mining algorithms. Overall, such approach presented in the thesis, from the graph representation to the matching against the frequent subgraphs is promising when applied to not only comic images but also another types of document images.

6.2 Direction for Future Research

There are several possibilities to take this work forward. As future work, we consider improving and extending our retrieval approach mainly in the following directions:

- Extend the current comic image databases (with groundtruth). In addition to our research on investigating into new subgraph spotting techniques, we are working on developing methods for automatically ground-truthing the node mappings and also to extend the size of currently available datasets; so that the learning based methods could benefit from these training set. In general larger and more standardized datasets will provide higher accuracy values thus facilitating the investigation of results. A larger database can also be used to increase confidence in the results obtained. Furthermore, experiments can be

run on a different data set for more rigorous proof of concept.

- Fully integrated relevance feedback function and query retouch (which allows user to draw on a returned result to make it another query, or to start the querying process with a sketch). This extension will allow smoother searching and browsing experience, as well as semi-supervised learning from reused results or re-ranked feedback from readers.
- Learning and deep learning to support the retrieval. For example, the ability to narrow down the areas in the panel which contain face or body of the interested characters is extremely helpful in the retrieval and recognition task later. However comic face detection and character detection are almost impossible without learning, and are still very challenging even with learning on even labeled data because it is hard to find features and heuristic rules which can well generalize faces/characters. Among machine learning methods, deep learning currently has the best detection model, and it has been shown that deep learning models are capable of detecting face and character in comic images. We believe that further studies in this direction of combining different tools could improve the detection and recognition accuracy.

Appendix A

List of Publications

The work on the topic of this thesis has led to the following publications:

Journal Papers

- Thanh-Nam Le, Muhammad Muzzamil Luqman, Anjan Dutta, Pierre Héroux, Christophe Rigaud, Clément Guérin, Pasquale Foggia, Jean-Christophe Burie, Jean-Marc Ogier, Josep Lladós, Sébastien Adam, *Subgraph Spotting in Graph Representations of Comic Book Images*, Pattern Recognition Letters, Vol. 112, 2018, pp. 118-124, ISSN 0167-8655. [Le et al., 2018]

- **[Under Review]** Thanh-Nam Le, Muhammad Muzzamil Luqman, Jean-Christophe Burie, Jean-Marc Ogier, *Unsupervised Comics Retrieval System Using Multilayer Graph Representation and Graph Mining*, Minor Revision, 2nd round, Pattern Recognition.

Conferences and Workshops

- Thanh-Nam Le, Muhammad Muzzamil Luqman, Jean-Christophe Burie, and Jean-Marc Ogier. 2016. *Retrieval of comic book images using context relevance information.* In Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding (MANPU '16). ACM, New York, NY, USA. [Le et al., 2016]

- Le, Thanh-Nam, Muhammad Muzzamil Luqman, Jean-Christophe Burie, and Jean-Marc Ogier. *Content-based comic retrieval using multilayer graph representation and frequent graph mining.* In Document Analysis and Recognition

(ICDAR), 2015 13th International Conference on, pp. 761-765. IEEE, 2015. [Le et al., 2015b]

- Le, Thanh-Nam, Muhammad Muzzamil Luqman, Jean-Christophe Burie, and Jean-Marc Ogier. *A comic retrieval system based on multilayer graph representation and graph mining.* In International Workshop on Graph-Based Representations in Pattern Recognition, pp. 355-364. Springer, Cham, 2015. [Le et al., 2015a]

- Rigaud, Christophe, Thanh-Nam Le, J-C. Burie, Jean-Marc Ogier, Shoya Ishimaru, Motoi Iwata, and Koichi Kise. *Semi-automatic text and graphics extraction of manga using eye tracking information.* In Document Analysis Systems (DAS), 2016 12th IAPR Workshop on, pp. 120-125. IEEE, 2016. [Rigaud et al., 2016]

- Rigaud, Christophe, Nam Le Thanh, J-C. Burie, J-M. Ogier, Motoi Iwata, Eiki Imazu, and Koichi Kise. *Speech balloon and speaker association for comics and manga understanding.* In Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, pp. 351-355. IEEE, 2015. [Rigaud et al., 2015b]

Bibliography

[Com, 2016] (2016). Comichorone, a resource for comic research. Website. 2

[Achanta et al., 2012] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282. 55

[Agrawal et al., 1993] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *ACM Sigmod Record*, volume 22, pages 207–216. ACM. xiv, 42

[Alexanderson, 2006] Alexanderson, G. (2006). About the cover: Euler and königsberg's bridges: A historical view. *Bulletin of the American Mathematical Society*, 43(4):567–573. 7

[Andreopoulos and Tsotsos, 2013] Andreopoulos, A. and Tsotsos, J. K. (2013). 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117(8):827–891. 6

[Arai and Herman, 2010] Arai, K. and Herman, T. (2010). Method for automatic e-comic scene frame extraction for reading comic on mobile devices. In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, pages 370–375. 64

[Arai and Tolle, 2011] Arai, K. and Tolle, H. (2011). Method for real time text extraction of digital manga comic. *International Journal of Image Processing*, 4(6):669–676. 64, 66

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer. 48, 52

[Berretti et al., 2003] Berretti, S., Del Bimbo, A., and Vicario, E. (2003). Weighted walkthroughs between extended entities for retrieval by spatial arrangement. *IEEE Transactions on Multimedia*, 5(1):52–70. 57

[Bezdek and Kuncheva, 2001] Bezdek, J. C. and Kuncheva, L. I. (2001). Nearest prototype classifier designs: An experimental study. *International journal of Intelligent systems*, 16(12):1445–1473. 39

[Bhattacharyya, 1943] Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distribution. *Bull. Calcutta Math. Soc.* 51

[Borgelt and Berthold, 2002] Borgelt, C. and Berthold, M. R. (2002). Mining molecular fragments: Finding relevant substructures of molecules. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 51–58. IEEE. 41

[Borodo, 2014] Borodo, M. (2014). Multimodality, translation and comics. *Perspectives*, pages 1–20. 3

[Bouissou, 2006] Bouissou, J.-M. (2006). Japan's growing cultural power. the example of manga in france. 3

[Brandon, 2014] Brandon, D. (2014). Graphic novels and comics for the visually impaired explored in award-winning paper. 3

[Brenner, 2007] Brenner, R. E. (2007). *Understanding Manga and Anime*. Greenwood Publishing Group. 2, 3

[Bron and Kerbosch, 1973] Bron, C. and Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577. 30

[Bunke, 1998] Bunke, H. (1998). Error-tolerant graph matching: a formal framework and algorithms. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 1–14. Springer. 32

[Bunke, 2000] Bunke, H. (2000). Graph matching: Theoretical foundations, algorithms, and applications. In *Proc. Vision Interface*, volume 2000, pages 82–88. 9, 32

[Bunke and Allermann, 1983] Bunke, H. and Allermann, G. (1983). Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245–253. 34

[Bunke and Messmer, 1997] Bunke, H. and Messmer, B. T. (1997). Recent advances in graph matching. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(01):169–203. 30

[Bunke and Riesen, 2008] Bunke, H. and Riesen, K. (2008). Graph classification based on dissimilarity space embedding. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 996–1007. 39, 58

[Calonder et al., 2010] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer. 48

[Carcassoni and Hancock, 2003] Carcassoni, M. and Hancock, E. R. (2003). Spectral correspondence for point pattern matching. *Pattern Recognition*, 36(1):193–204. 36

[Carletti et al., 2017] Carletti, V., Foggia, P., Saggese, A., and Vento, M. (2017). Introducing vf3: A new algorithm for subgraph isomorphism. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 128–139. Springer. xi, 30, 31

[Carroll et al., 1992] Carroll, P. J., Young, J. R., and Guertin, M. S. (1992). Visual analysis of cartoons: A view from the far side. In *Eye movements and visual cognition*, pages 444–461. Springer. 71

[Carson et al., 1999] Carson, C., Thomas, M., Belongie, S., Hellerstein, J. M., and Malik, J. (1999). Blobworld: A system for region-based image indexing and retrieval. In *International Conference on Advances in Visual Information Systems*, pages 509–517. Springer. 60

[Chan et al., 2007] Chan, C., Leung, H., and Komura, T. (2007). Automatic panel extraction of color comic images. In Ip, H.-S., Au, O., Leung, H., Sun, M.-T., Ma, W.-Y., and Hu, S.-M., editors, *Advances in Multimedia Information Processing - PCM 2007*, volume 4810 of *Lecture Notes in Computer Science*, pages 775–784. Springer Berlin Heidelberg. 63

[Chang and Liu, 1984] Chang, S.-K. and Liu, S.-H. (1984). Picture indexing and abstraction techniques for pictorial databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4):475–484. 47

[Chen et al., 2007] Chen, C., Yan, X., Zhu, F., and Han, J. (2007). gapprox: Mining frequent approximate patterns from a massive network. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 445–450. IEEE. 45

[Chen et al., 2003] Chen, Q., Lim, A., and Ong, K. W. (2003). D (k)-index: An adaptive structural summary for graph-structured data. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 134–144. ACM. 46

[Chen et al., 2009] Chen, X., Hu, X., and Shen, X. (2009). Spatial weighting for bag-of-visual-words and its application in content-based image retrieval. *Advances in Knowledge Discovery and Data Mining*, pages 867–874. 50, 54

[Cheng et al., 2007] Cheng, J., Ke, Y., Ng, W., and Lu, A. (2007). Fg-index: towards verification-free query processing on graph databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 857–872. ACM. 46

[Cheung et al., 2008] Cheung, S., of Hong Kong. Run Run Shaw Library, C. U., and of Hong Kong. Department of Computer Science, C. U. (2008). *Face Detection and Face Recognition of Human-like Characters in Comics*. Outstanding academic papers by students. Run Run Shaw Library, City University of Hong Kong. 68

[Christiansen, 2000] Christiansen, H.-C. (2000). *Comics & Culture: Analytical and Theoretical Approaches to Comics*. Museum Tusculanum Press. 1

[Christmas et al., 1995] Christmas, W. J., Kittler, J., and Petrou, M. (1995). Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on pattern analysis and machine intelligence*, 17(8):749–764. 33

[Chu and Chao, 2014] Chu, W.-T. and Chao, Y.-C. (2014). Line-based drawing style description for manga classification. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 781–784. ACM. 80, 81

[Chu et al., 1994] Chu, W. W., Ieong, I. T., and Taira, R. K. (1994). A semantic modeling approach for image retrieval by content. *The VLDB Journal - The International Journal on Very Large Data Bases*, 3(4):445–477. 61

[Chum et al., 2011] Chum, O., Mikulik, A., Perdoch, M., and Matas, J. (2011). Total recall ii: Query expansion revisited. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 889–896. IEEE. 11, 60

[Chum et al., 2007] Chum, O., Philbin, J., Sivic, J., Isard, M., and Zisserman, A. (2007). Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE. 60

[Cohn, 2011] Cohn, N. (2011). A different kind of cultural frame: An analysis of panels in american comics and japanese manga. *Image & Narrative*, 12(1):120–134. 71

[Cohn, 2013a] Cohn, N. (2013a). Beyond speech balloons and thought bubbles: The integration of text and image. *Semiotica*, 2013(197):35–63. 71

[Cohn, 2013b] Cohn, N. (2013b). Navigating comics: An empirical and theoretical approach to strategies of reading comic page layouts. *Frontiers in psychology*, 4(April):186. 71

[Cohn, 2013c] Cohn, N. (2013c). *The Visual Language of Comics: Introduction to the Structure and Cognition of Sequential Images.* A&C Black. 71

[Cohn, 2018] Cohn, N. (2018). In defense of a "grammar" in the visual language of comics. *Journal of Pragmatics*, 127:1–19. 71

[Coifman and Lafon, 2006] Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30. 113

[Conte et al., 2004] Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298. xiv, 28, 29, 37, 38

[Cook and Holder, 2006] Cook, D. J. and Holder, L. B. (2006). *Mining Graph Data.* John Wiley & Sons. 40, 44

[Cooper et al., 2001] Cooper, B. F., Sample, N., Franklin, M. J., Hjaltason, G. R., and Shadmon, M. (2001). A fast index for semistructured data. In *VLDB*, volume 1, pages 341–350. 46

[Cordella et al., 1999] Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. (1999). Performance evaluation of the vf graph matching algorithm. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 1172–1177. IEEE. xi, 30, 31

[Cordella et al., 2004] Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. (2004). A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence*, 26(10):1367–1372. xi, 30, 31, 99, 101

[Cour et al., 2006] Cour, T., Srinivasan, P., and Shi, J. (2006). Balanced graph matching. In *NIPS*, volume 2, page 6. 36

[Cox et al., 2000] Cox, I. J., Miller, M. L., Minka, T. P., Papathomas, T. V., and Yianilos, P. N. (2000). The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *IEEE transactions on image processing*, 9(1):20–37. 60

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In Schmid, C., Soatto, S., and Tomasi, C., editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334. 69

[Datta et al., 2008] Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5. xiv, 61

[Del Bimbo, 1999] Del Bimbo, A. (1999). *Visual information retrieval*. Morgan and Kaufmann. 49

[Deza and Deza, 2009] Deza, M. M. and Deza, E. (2009). Encyclopedia of distances. In *Encyclopedia of Distances*, pages 1–583. Springer. 34, 50, 51

[Do and Vetterli, 2002] Do, M. N. and Vetterli, M. (2002). Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. *IEEE transactions on image processing*, 11(2):146–158. 49

[Dolle-Weinkauff, 2006] Dolle-Weinkauff, B. (2006). The attractions of intercultural exchange: Manga market and manga reception in germany. In *Communication a la Conference International, Asia Culture Forum*. 3

[Duchowski, 2007] Duchowski, A. T. (2007). Eye tracking methodology. *Theory and practice*, 328. 71

[Dutta et al., 2018] Dutta, A., Lladós, J., Bunke, H., and Pal, U. (2018). Product graph-based higher order contextual similarities for inexact subgraph matching. *Pattern Recognition*, 76:596–611. 113

[Eisner, 1985] Eisner, W. (1985). *Comics and Sequential Art*. W. W. Norton and Company. 1

[Faloutsos et al., 1994] Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., and Equitz, W. (1994). Efficient and effective querying by image content. *Journal of intelligent information systems*, 3(3-4):231–262. 60

[Ferrari et al., 2008] Ferrari, V., Fevrier, L., Schmid, C., and Jurie, F. (2008). Groups of adjacent contour segments for object detection. 49

[Ferrer et al., 2011] Ferrer, M., Karatzas, D., Valveny, E., Bardají, I., and Bunke, H. (2011). A generic framework for median graph computation based on a recursive embedding approach. *Computer Vision and Image Understanding*, 115(7):919–928. 58

[Figueiredo, 2018] Figueiredo, B. (2018). Imagining the global: transnational media and popular culture beyond east and west. *Consumption Markets & Culture*, 21(2):187–190. 3

[Fischer et al., 2004] Fischer, B., Thies, C. J., Guld, M. O., and Lehmann, T. M. (2004). Content-based image retrieval by matching hierarchical attributed region adjacency graphs. In *Proceedings of SPIE*, volume 5370, pages 598–606. 57

[Flickner et al., 1995] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., et al. (1995). Query by image and video content: The qbic system. *computer*, 28(9):23–32. 60

[Flusser, 2006] Flusser, J. (2006). Moment invariants in image analysis. In *proceedings of world academy of science, engineering and technology*, volume 11, pages 196–201. 84

[Flusser et al., 2009] Flusser, J., Zitova, B., and Suk, T. (2009). *Moments And Moment Invariants In Pattern Recognition*. John Wiley & Sons. 84

[Foggia et al., 2014] Foggia, P., Percannella, G., and Vento, M. (2014). Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(01):1450001. xiv, 37, 38

[Foggia et al., 2001] Foggia, P., Sansone, C., and Vento, M. (2001). A performance comparison of five algorithms for graph isomorphism. In *Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*, pages 188–199. 30

[Freeman, 1974] Freeman, H. (1974). Computer processing of line-drawing images. *ACM Computing Surveys (CSUR)*, 6(1):57–97. 89

[Gao et al., 2010] Gao, X., Xiao, B., Tao, D., and Li, X. (2010). A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129. 34

[Garey and Johnson, 1990] Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA. 40

[Giugno and Shasha, 2002] Giugno, R. and Shasha, D. (2002). Graphgrep: A fast and universal method for querying graphs. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 112–115. IEEE. 46

[Goggin, 2010] Goggin, J. (2010). *The Rise and Reason of Comics and Graphic Literature: Critical Essays on the Form*, volume 1, chapter 4, pages 56–74. McFarland, Book News, Inc., Portland, OR, 1 edition. 64

[Gold and Rangarajan, 1996] Gold, S. and Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, 18(4):377–388. 33

[Goldman and Widom, 1997] Goldman, R. and Widom, J. (1997). Dataguides: Enabling query formulation and optimization in semistructured databases. Technical report, Stanford. 46

[Gomez and Karatzas, 2013] Gomez, L. and Karatzas, D. (2013). Multi-script text extraction from natural scenes. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 467–471. IEEE. 65

[Goodrum, 2000] Goodrum, A. A. (2000). Image information retrieval: An overview of current research. *Informing Science*, 3(2):63–66. 61

[Guérin, 2012] Guérin, C. (2012). Ontologies and spatial relations applied to comic books reading. In *PhD Symposium of Knowledge Engineering and Knowledge Management (EKAW)*, Galway, Ireland. 64, 71

[Guérin et al., 2017] Guérin, C., Rigaud, C., Bertet, K., and Revel, A. (2017). An ontology-based framework for the automated analysis and interpretation of comic books images. *Information sciences*, 378:109–130. 69

[Guerin et al., 2013] Guerin, C., Rigaud, C., Mercier, A., Ammar-Boudjelal, F., Bertet, K., Bouju, A., Burie, J.-C., Louis, G., Ogier, J.-M., and Revel, A. (2013). ebdtheque: A representative database of comics. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1145–1149. 73, 103

[Gupta and Jain, 1997] Gupta, A. and Jain, R. (1997). Visual information retrieval. *Communications of the ACM*, 40(5):70–79. 60

[Hafner et al., 1995] Hafner, J., Sawhney, H. S., Equitz, W., Flickner, M., and Niblack, W. (1995). Efficient color histogram indexing for quadratic form distance functions. *IEEE transactions on pattern analysis and machine intelligence*, 17(7):729–736. 49

[Han et al., 2007] Han, E., Kim, K., Yang, H., and Jung, K. (2007). Frame segmentation used mlp-based x-y recursive for mobile cartoon content. In *Proceedings of the 12th international conference on Human-computer interaction: intelligent multimodal interaction environments*, HCI'07, pages 872–881, Berlin, Heidelberg. Springer-Verlag. 63

[Han et al., 2011] Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier. 40, 41, 42, 97

[Han et al., 2000] Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. In *ACM Sigmod Record*, volume 29, pages 1–12. ACM. 43

[Han et al., 2010] Han, W.-S., Lee, J., Pham, M.-D., and Yu, J. X. (2010). igraph: a framework for comparisons of disk-based graph indexing techniques. *Proceedings of the VLDB Endowment*, 3(1-2):449–459. 46

[Haralick et al., 1973] Haralick, R. M., Shanmugam, K., et al. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621. 49

[Hermann et al., 2012] Hermann, A., Ferré, S., and Ducassé, M. (2012). Guided semantic annotation of comic panels with sewelis. In *EKAW*, volume 7603 of *Lecture Notes in Computer Science*, pages 430–433. Springer. 69

[Ho et al., 2012] Ho, A. K. N., Burie, J. C., and Ogier, J. (2012). Panel and speech balloon extraction from comic books. In *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, pages 424–428. Ieee. 64

[Ho et al., 2013] Ho, H. N., RIGAUD, C., BURIE, J.-C., and OGIER, J.-M. (2013). Redundant structure detection in attributed adjacency graphs for character detection in comics books. In *10th IAPR International Workshop on Graphics Recognition*. 69

[Hoàng et al., 2010] Hoàng, N. V., Gouet-Brunet, V., Rukoz, M., and Manouvrier, M. (2010). Embedding spatial information into image content description for scene retrieval. *Pattern Recognition*, 43(9):3013–3024. 57

[Holder et al., 1994] Holder, L. B., Cook, D. J., Djoko, S., et al. (1994). Substucture discovery in the subdue system. In *KDD workshop*, pages 169–180. 41

[Hore and Ray, 2002] Hore, E. and Ray, S. (2002). A sum-result indexing algorithm for feature combining in content-based image retrieval. 100

[Hu, 1962] Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187. 49, 84

[Huan et al., 2004] Huan, J., Wang, W., Bandyopadhyay, D., Snoeyink, J., Prins, J., and Tropsha, A. (2004). Mining protein family specific residue packing patterns from protein structure graphs. In *Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*, pages 308–315. ACM. 41

[Huet et al., 1999] Huet, B., Cross, A. D., and Hancock, E. R. (1999). Shape retrieval by inexact graph matching. In *Multimedia Computing and Systems, 1999. IEEE International Conference on*, volume 1, pages 772–776. IEEE. 28

[Huet and Hancock, 1999] Huet, B. and Hancock, E. R. (1999). Shape recognition from large image libraries by inexact graph matching. *Pattern Recognition Letters*, 20(11):1259–1269. 33

[In et al., 2011] In, Y., Oie, T., Higuchi, M., Kawasaki, S., Koike, A., and Murakami, H. (2011). Fast frame decomposition and sorting by contour tracing for mobile phone comic images. *Internatinal journal of systems applications, engineering and development*, 5(2):216–223. 63

[Indyk and Motwani, 1998] Indyk, P. and Motwani, R. (1998). 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. 91

[Ingulsrud and Allen, 2009] Ingulsrud, J. E. and Allen, K. (2009). *Reading Japan cool: patterns of manga literacy and discourse*. Lexington Books. 3

[Inokuchi et al., 2000] Inokuchi, A., Washio, T., and Motoda, H. (2000). An apriori-based algorithm for mining frequent substructures from graph data. *Principles of Data Mining and Knowledge Discovery*, pages 13–23. 41

[Iwata et al., 2014] Iwata, M., Ito, A., and Kise, K. (2014). A study to achieve manga character retrieval method for manga images. In *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*, pages 309–313. 69, 71

[Jain and Zongker, 1997] Jain, A. and Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE transactions on pattern analysis and machine intelligence*, 19(2):153–158. 39

[Jain, 2010] Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666. 53

[Jain and Vailaya, 1996] Jain, A. K. and Vailaya, A. (1996). Image retrieval using color and shape. *Pattern Recognition*, 29:1233–1244. 94

[Jaworska et al., 2010] Jaworska, T., Kacprzyk, J., Marín, N., and Zadrozny, S. (2010). On dealing with imprecise information in a content based image retrieval system. In *IPMU*, pages 149–158. Springer. 57

[Jégou et al., 2010a] Jégou, H., Douze, M., and Schmid, C. (2010a). Improving bag-of-features for large scale image search. *International journal of computer vision*, 87(3):316–336. 60

[Jégou et al., 2010b] Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010b). Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE. 55

[Jiang et al., 2013] Jiang, C., Coenen, F., and Zito, M. (2013). A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28(01):75–105. xiv, 41, 43, 44, 97

[Jing et al., 2002] Jing, F., Li, M., Zhang, H.-J., and Zhang, B. (2002). An effective region-based image retrieval framework. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 456–465. ACM. 56

[Jing et al., 2004] Jing, F., Li, M., Zhang, H.-J., and Zhang, B. (2004). An efficient and effective region-based image retrieval framework. *IEEE Transactions on Image Processing*, 13(5):699–709. 56

[Jung et al., 2004] Jung, K., Kim, K. I., and Jain, A. K. (2004). Text information extraction in images and video: a survey. *Pattern Recognition*, 37(5):977 – 997. 65

[Kang and Ju, 2009] Kang, H.-B. and Ju, M.-H. (2009). Viewing comics on robots. In *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*, pages 480–485. 71

[Kaspar and Horst, 2010] Kaspar, R. and Horst, B. (2010). *Graph classification and clustering based on vector space embedding*, volume 77. World Scientific. 37, 39

[Kass et al., 1988] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331. 65

[Kato, 1992] Kato, T. (1992). Database architecture for content-based image retrieval. In *SPIE/IS&T 1992 symposium on electronic imaging: science and technology*, pages 112–123. International Society for Optics and Photonics. 47

[Kaushik et al., 2002] Kaushik, R., Shenoy, P., Bohannon, P., and Gudes, E. (2002). Exploiting local similarity for indexing paths in graph-structured data. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 129–140. IEEE. 46

[Ketkar et al., 2005] Ketkar, N. S., Holder, L. B., and Cook, D. J. (2005). Subdue: Compression-based frequent pattern discovery in graph data. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pages 71–76. ACM. 44

[Khan et al., 2012a] Khan, F. S., Rao, M. A., van de Weijer, J., Bagdanov, A. D., Vanrell, M., and Lopez, A. (2012a). Color attributes for object detection. In *Twenty-Fifth IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*. 69

[Khan et al., 2012b] Khan, R., Barat, C., Muselet, D., and Ducottet, C. (2012b). Spatial orientations of visual word pairs to improve bag-of-visual-words model. In *Proceedings of the British Machine Vision Conference*, pages 89–1. BMVA Press. 54

[Khotanzad and Hong, 1990] Khotanzad, A. and Hong, Y. H. (1990). Invariant image recognition by zernike moments. *IEEE Transactions on pattern analysis and machine intelligence*, 12(5):489–497. 49

[Kitchen, 1978] Kitchen, L. (1978). Discrete relaxation for matching relational structures. Technical report, MARYLAND UNIV COLLEGE PARK COMPUTER SCIENCE CENTER. 33

[Kittler and Hancock, 1989] Kittler, J. and Hancock, E. R. (1989). Combining evidence in probabilistic relaxation. *International Journal of Pattern Recognition and Artificial Intelligence*, 3(01):29–51. 33

[Krapac et al., 2011] Krapac, J., Verbeek, J., and Jurie, F. (2011). Modeling spatial layout with fisher vectors for image categorization. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1487–1494. IEEE. 55

[Kubicka et al., 1990] Kubicka, E., Kubicki, G., and Vakalis, I. (1990). Using graph distance in object recognition. In *Proceedings of the 1990 ACM annual conference on Cooperation*, pages 43–48. ACM. 28

[Kuner and Ueberreiter, 1988] Kuner, P. and Ueberreiter, B. (1988). Pattern recognition by graph matching - combinatorial versus continuous optimization. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(03):527–542. 33

[Kuramochi and Karypis, 2001] Kuramochi, M. and Karypis, G. (2001). Frequent subgraph discovery. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 313–320. IEEE. 41, 45

[Kuramochi and Karypis, 2004] Kuramochi, M. and Karypis, G. (2004). An efficient algorithm for discovering frequent subgraphs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1038–1051. 44

[Lazebnik et al., 2006] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2169–2178. IEEE. xiv, 54, 55

[Le et al., 2015a] Le, T.-N., Luqman, M. M., Burie, J.-C., and Ogier, J.-M. (2015a). A comic retrieval system based on multilayer graph representation and graph mining. In Liu, C.-L., Luo, B., Kropatsch, W. G., and Cheng, J., editors, *Graph-Based Representations in Pattern Recognition*, pages 355–364, Cham. Springer International Publishing. 126

[Le et al., 2015b] Le, T. N., Luqman, M. M., Burie, J. C., and Ogier, J. M. (2015b). Content-based comic retrieval using multilayer graph representation and frequent graph mining. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 761–765. 105, 126

[Le et al., 2016] Le, T.-N., Luqman, M. M., Burie, J.-C., and Ogier, J.-M. (2016). Retrieval of comic book images using context relevance information. In *Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding*, MANPU '16, pages 12:1–12:6, New York, NY, USA. ACM. 63, 125

[Le et al., 2018] Le, T. N., Luqman, M. M., Dutta, A., Héroux, P., Rigaud, C., Guérin, C., Foggia, P., Burie, J.-C., Ogier, J.-M., Lladós, J., and Adam, S. (2018). Subgraph spotting in graph representations of comic book images. *Pattern Recognition Letters*, 112:118 – 124. 14, 125

[Leordeanu and Hebert, 2005] Leordeanu, M. and Hebert, M. (2005). A spectral technique for correspondence problems using pairwise constraints. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1482–1489. IEEE. 33, 36

[Lerouge et al., 2016] Lerouge, J., Hammami, M., Héroux, P., and Adam, S. (2016). Minimum cost subgraph matching using a binary linear program. *Pattern Recognition Letters*, 71:45–51. 113, 114, 116, 122

[Levenshtein, 1966] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. 32

[Levinshtein et al., 2009] Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., and Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2290–2297. 55

[Li et al., 2014] Li, L., Wang, Y., Tang, Z., and Gao, L. (2014). Automatic comic page segmentation based on polygon detection. *Multimedia Tools Applications*, 69(1):171–197. 63

[Li et al., 2013] Li, L., Wang, Y., Tang, Z., Lu, X., and Gao, L. (2013). Unsupervised speech text localization in comic images. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1190–1194. 66, 67

[Lopes, 2009] Lopes, P. (2009). *Demanding Respect: The Evolution of the American comic book*. Temple University Press. 1

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee. 6, 48, 52, 79

[Lu and Zhang, 2010] Lu, X. and Zhang, M.-Q. (2010). The animation and comics content retrieval model based on analysis of clustered group. In *Biomedical Engineering and Computer Science (ICBECS), 2010 International Conference on*, pages 1–4. 68

[Lucas, 1996] Lucas, S. (1996). Rapid content-based retrieval from document image databases. In *Intelligent Image Databases, IEE Colloquium on*, pages 10/1–10/6. 71

[Luqman et al., 2013] Luqman, M. M., Ho, H. N., Burie, J.-C., and Ogier, J.-M. (2013). Automatic indexing of comic page images for query by example based focused content retrieval. In *10th IAPR International Workshop on Graphics Recognition*, United States. 72

[Ma and Manjunath, 1997] Ma, W.-Y. and Manjunath, B. S. (1997). Netra: A toolbox for navigating large image databases. In *Image Processing, 1997. Proceedings., International Conference on*, volume 1, pages 568–571. IEEE. 56

[Manjunath and Ma, 1996] Manjunath, B. S. and Ma, W.-Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, 18(8):837–842. 49

[Marszaek and Schmid, 2006] Marszaek, M. and Schmid, C. (2006). Spatial weighting for bag-of-features. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2118–2125. IEEE. 54

[Matas et al., 2004] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767. 69, 79, 80

[Matsui et al., 2014] Matsui, Y., Aizawa, K., and Jing, Y. (2014). Sketch2manga: Sketch-based manga retrieval. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 3097–3101. xiv, 69, 72

[Matsui et al., 2015] Matsui, Y., Ito, K., Aramaki, Y., Yamasaki, T., and Aizawa, K. (2015). Sketch-based manga retrieval using manga109 dataset. *CoRR*, abs/1510.04389. 5, 11, 69, 72

[Matsui et al., 2011] Matsui, Y., Yamasaki, T., and Aizawa, K. (2011). Interactive Manga retargeting. In *ACM SIGGRAPH 2011 Posters on - SIGGRAPH '11*, page 1, New York, New York, USA. ACM Press. 65

[McCann and Lowe, 2012] McCann, S. and Lowe, D. G. (2012). Spatially local coding for object recognition. In *Asian Conference on Computer Vision*, pages 204–217. Springer. 54

[McCloud, 1993] McCloud, S. (1993). *Understanding comics: The Invisible Art*. Kitchen Sink Press. 1

[McKay et al., 1981] McKay, B. D. et al. (1981). Practical graph isomorphism. xi, 30, 31

[Meskin and Cook, 2011] Meskin, A. and Cook, R. T. (2011). *The Art of Comics: A Philosophical Approach - Ch 2 The Ontology of Comics*, volume 1, pages 218–285. John Wiley and Sons, New York, USA. 69

[Messmer, 1996] Messmer, B. (1996). Efficient graph matching algorithms for preprocessed model graphs [ph. d. thesis]. *University of Bern*. 58

[Messmer and Bunke, 1998] Messmer, B. T. and Bunke, H. (1998). A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):493–504. 32, 58

[Messmer and Bunke, 1999] Messmer, B. T. and Bunke, H. (1999). A decision tree approach to graph and subgraph isomorphism detection. *Pattern recognition*, 32(12):1979–1998. 30

[Mezaris et al., 2003] Mezaris, V., Kompatsiaris, I., and Strintzis, M. G. (2003). An ontology approach to object-based image retrieval. In *In Proc. IEEE Int. Conf. on Image Processing (ICIP03*, volume 2, pages 511–514. IEEE. 56

[Mikolajczyk et al., 2005] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72. 79

[Milo and Suciu, 1999] Milo, T. and Suciu, D. (1999). Index structures for path expressions. In *International Conference on Database Theory*, pages 277–295. Springer. 46

[Mojsilovic et al., 2002] Mojsilovic, A., Gomes, J., and Rogowitz, B. E. (2002). Isee: Perceptual features for image library navigation. In *Electronic Imaging 2002*, pages 266–277. International Society for Optics and Photonics. 56

[Neuhaus and Bunke, 2007] Neuhaus, M. and Bunke, H. (2007). *Bridging the gap between graph edit distance and kernel machines*, volume 68. World Scientific. 33, 36

[Nguyen et al., 2017] Nguyen, N.-V., Rigaud, C., and Burie, J.-C. (2017). Comic characters detection using deep learning. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 3, pages 41–46. IEEE. 70

[Nguyen et al., 2018] Nguyen, N.-V., Rigaud, C., and Burie, J.-C. (2018). Digital comics image indexing based on deep learning. *Journal of Imaging*, 4(7). 70

[Nijssen and Kok, 2005] Nijssen, S. and Kok, J. N. (2005). The gaston tool for frequent subgraph mining. *Electronic Notes in Theoretical Computer Science*, 127(1):77–87. 41, 97

[Nister and Stewenius, 2006] Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2161–2168. Ieee. 53, 95

[Ohta et al., 1980] Ohta, Y.-I., Kanade, T., and Sakai, T. (1980). Color information for region segmentation. *Computer graphics and image processing*, 13(3):222–241. 83

[Omhover and Detyniecki, 2004] Omhover, J.-F. and Detyniecki, M. (2004). Strict: an image retrieval platform for queries based on regional content. In *International Conference on Image and Video Retrieval*, pages 473–482. Springer. 56

[Opelt et al., 2006] Opelt, A., Pinz, A., and Zisserman, A. (2006). A boundary-fragment-model for object detection. In *European conference on computer vision*, pages 575–588. Springer. 49

[Pang et al., 2014] Pang, X., Cao, Y., Lau, R. W., and Chan, A. B. (2014). A robust panel extraction method for manga. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 1125–1128, New York, NY, USA. ACM. 64

[Pentland et al., 1996] Pentland, A., Picard, R. W., and Sclaroff, S. (1996). Photobook: Content-based manipulation of image databases. *International journal of computer vision*, 18(3):233–254. 60

[Perchant and Bloch, 2002] Perchant, A. and Bloch, I. (2002). Fuzzy morphisms between graphs. *Fuzzy Sets and Systems*, 128(2):149–168. 33

[Perronnin and Dance, 2007] Perronnin, F. and Dance, C. (2007). Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE. 55

[Perronnin et al., 2010] Perronnin, F., Liu, Y., Sánchez, J., and Poirier, H. (2010). Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384–3391. IEEE. 55

[Petermann et al., 2017] Petermann, A., Junghanns, M., and Rahm, E. (2017). Dimspan-transactional frequent subgraph mining with distributed in-memory dataflow systems. *arXiv preprint arXiv:1703.01910*. 97

[Petrakis et al., 2002] Petrakis, E. G. M., Faloutsos, C., and Lin, K.-I. (2002). Imagemap: An image indexing method based on spatial similarity. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):979–987. 57

[Philbin et al., 2007] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE. 95

[Pkkalska and Duin, 2005] Pkkalska, E. and Duin, R. (2005). The dissimilarity representation for pattern recognition. *World Scientific*. 37

[Ponsard et al., 2012] Ponsard, C., Ramdoyal, R., and Dziamski, D. (2012). An ocr-enabled digital comic books viewer. In *Computers Helping People with Special Needs*, pages 471–478. Springer. 63, 66

[Rahman et al., 2009] Rahman, M. M., Antani, S. K., Long, L. R., Demner-Fushman, D., and Thoma, G. R. (2009). Multi-modal query expansion based on local analysis for medical image retrieval. In *MCBR-CDS*, pages 110–119. Springer. 61

[Ralaivola et al., 2005] Ralaivola, L., Swamidass, S. J., Saigo, H., and Baldi, P. (2005). Graph kernels for chemical informatics. *Neural networks*, 18(8):1093–1110. 28

[Reeves et al., 1988] Reeves, A. P., Prokop, R. J., Andrews, S. E., and Kuhl, F. P. (1988). Three-dimensional shape analysis using moments and fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):937–943. 84

[Reinhard et al., 2001] Reinhard, E., Adhikhmin, M., Gooch, B., and Shirley, P. (2001). Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41. 83

[Ren et al., 2008] Ren, P., Wilson, R., and Hancock, E. (2008). Spectral embedding of feature hypergraphs. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 308–317. 58

[Ren et al., 2002] Ren, W., Singh, M., and Singh, S. (2002). Image retrieval using spatial context. In *Proceedings of the 9th international workshop on systems, signals and image processing*, pages 44–49. 55

[Ren and Malik, 2003] Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 10–17 vol.1. 54

[Report, 2014] Report, F. B. F. (2014). Information on the french book market. 2

[Riesen and Bunke, 2009] Riesen, K. and Bunke, H. (2009). Dissimilarity based vector space embedding of graphs using prototype reduction schemes. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 617–631. Springer. 39

[Riesen et al., 2007a] Riesen, K., Kilchherr, V., and Bunke, H. (2007a). Reducing the dimensionality of vector space embeddings of graphs. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 563–573. Springer. 39

[Riesen et al., 2007b] Riesen, K., Neuhaus, M., and Bunke, H. (2007b). Graph embedding in vector spaces by means of prototype selection. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 383–393. Springer. 58

[Rigaud, 2014] Rigaud, C. (2014). *Segmentation et indexation d'objets complexes dans les images de bandes déssinées*. PhD thesis, Université de La Rochelle. xiv, 65

[Rigaud et al., 2017] Rigaud, C., Burie, J.-C., and Ogier, J.-M. (2017). Segmentation-free speech text recognition for comic books. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 3, pages 29–34. IEEE. 67

[Rigaud et al., 2014] Rigaud, C., Burie, J.-C., Ogier, J.-M., and Karatzas, D. (2014). Color descriptor for content-based drawing retrieval. In *Document Analysis Systems (DAS), 11th IAPR International Workshop*, pages 267–271. 72

[Rigaud et al., 2013a] Rigaud, C., Burie, J.-C., Ogier, J.-M., Karatzas, D., and van de Weijer, J. (2013a). An active contour model for speech balloon detection in comics. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1240–1244. 64

[Rigaud et al., 2015a] Rigaud, C., Guérin, C., Karatzas, D., Burie, J.-C., and Ogier, J.-M. (2015a). Knowledge-driven understanding of images in comic books. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(3):199–221. 65, 70

[Rigaud et al., 2013b] Rigaud, C., Karatzas, D., deWeijer, J. V., Burie, J.-C., and Ogier, J.-M. (2013b). Automatic text localisation in scanned comic books. In *9th International Conference on Computer Vision Theory and Applications*, Barcelona. 64

[Rigaud et al., 2016] Rigaud, C., Le, T.-N., Burie, J.-C., Ogier, J.-M., Ishimaru, S., Iwata, M., , and Kise, K. (2016). Semi-automatic text and graphics extraction of manga using eye tracking information. In *DAS 2016*. 70, 71, 126

[Rigaud et al., 2015b] Rigaud, C., Le Thanh, N., Burie, J.-C., Ogier, J.-M., Iwata, M., Imazu, E., and Kise, K. (2015b). Speech balloon and speaker association for comics and manga understanding. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 351–355. IEEE. 65, 126

[Rigaud et al., 2012] Rigaud, C., Tsopze, N., Burie, J.-C., and Ogier, J.-M. (2012). Extraction robuste des cases et du texte de bandes dessinées. In *Proceedings of Colloque International Francophone sur l'Ecrit et le Document (CIFED)*, pages 349–360. 67

[Rigaud et al., 2013c] Rigaud, C., Tsopze, N., Burie, J.-C., and Ogier, J.-M. (2013c). Robust frame and text extraction from comic books. In Kwon, Y.-B. and Ogier, J.-M., editors, *Graphics Recognition. New Trends and Challenges*, volume 7423 of *Lecture Notes in Computer Science*, pages 129–138. Springer Berlin Heidelberg. 79

[Robin Varnum, 2007] Robin Varnum, Christina T., G. (2007). *The Language of Comics: Word and Image*. Studies in Popular Culture. University Press of Mississippi. 64

[Robles-Kelly and Hancock, 2007] Robles-Kelly, A. and Hancock, E. R. (2007). A riemannian approach to graph embedding. *Pattern Recognition*, 40(3):1042–1056. 58

[Rublee et al., 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE. 48

[Rui et al., 1999] Rui, Y., Huang, T. S., and Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10(1):39–62. 61

[Rui et al., 1998] Rui, Y., Huang, T. S., Ortega, M., and Mehrotra, S. (1998). Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, 8(5):644–655. 59

[Sanfeliu and Fu, 1983] Sanfeliu, A. and Fu, K.-S. (1983). A distance measure between attributed relational graphs for pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (3):353–362. 32, 34

[Schenker et al., 2004] Schenker, A., Last, M., Bunke, H., and Kandel, A. (2004). Classification of web documents using graph matching. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):475–496. 28

[Schettini et al., 2001] Schettini, R., Ciocca, G., and Zuffi, S. (2001). Color imaging science: exploiting digital media. *R. Luo, L. MacDonald, Wiley, New York*. 49

[Schick et al., 2012] Schick, A., Fischer, M., and Stiefelhagen, R. (2012). Measuring and evaluating the compactness of superpixels. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 930–934. IEEE. 55

[Schmidt and Druffel, 1976] Schmidt, D. C. and Druffel, L. E. (1976). A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices. *Journal of the ACM (JACM)*, 23(3):433–445. xi, 31

[Screech, 2005] Screech, M. (2005). *Masters of the Ninth Art: Bandes Dessinees and Franco-Belgian Identity*, volume 3. Liverpool University Press. 1

[Seong et al., 1994] Seong, D. S., Choi, Y. K., Kim, H. S., and Park, K. H. (1994). An algorithm for optimal isomorphism between two random graphs. *Pattern recognition letters*, 15(4):321–327. 33

[Sethi et al., 2001] Sethi, I. K., Coman, I. L., and Stan, D. (2001). Mining association rules between low-level image features and high-level concepts. In *Data Mining and Knowledge Discovery: Theory, Tools, and Technology III*, volume 4384, pages 279–291. International Society for Optics and Photonics. 11

[Shasha et al., 2002] Shasha, D., Wang, J. T., and Giugno, R. (2002). Algorithmics and applications of tree and graph searching. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 39–52. ACM. 46

[Shawe-Taylor and Cristianini, 2004] Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press. 24, 33, 35

[Shen et al., 2012] Shen, X., Lin, Z., Brandt, J., Avidan, S., and Wu, Y. (2012). Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3013–3020. IEEE. 60

[Simonyan et al., 2013] Simonyan, K., Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2013). Fisher vector faces in the wild. In *BMVC*, volume 2, page 4. 55

[Singh et al., 2004] Singh, S., Cheok, A. D., Ng, G. L., and Farbiz, F. (2004). 3d augmented reality comic book and notes for children using mobile phones. In *Proceedings of the 2004 Conference on Interaction Design and Children: Building a Community*, IDC '04, pages 149–150, New York, NY, USA. ACM. 3

[Smeulders et al., 2000] Smeulders, A. W., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12):1349–1380. 61

[Smith and Chang, 1997] Smith, J. R. and Chang, S.-F. (1997). Visualseek: a fully automated content-based image query system. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 87–98. ACM. 60

[Song et al., 2003] Song, Y., Wang, W., and Zhang, A. (2003). Automatic annotation and retrieval of images. *World Wide Web*, 6(2):209–231. 56

[Spillmann et al., 2006] Spillmann, B., Neuhaus, M., Bunke, H., Pekalska, E., and Duin, R. P. (2006). Transforming strings to vector spaces using prototype selection. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 287–296. Springer. 39

[Stauffer and Grimson, 1999] Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 246–252. IEEE. 53

[Stommel et al., 2012] Stommel, M., Merhej, L. I., and Müller, M. G. (2012). Segmentation-free detection of comic panels. In *Computer Vision and Graphics*, pages 633–640. Springer. 63

[Stricker and Dimai, 1996] Stricker, M. A. and Dimai, A. (1996). Color indexing with weak spatial constraints. In *Electronic Imaging: Science & Technology*, pages 29–40. International Society for Optics and Photonics. 49

[Stricker and Orengo, 1995] Stricker, M. A. and Orengo, M. (1995). Similarity of color images. In *Storage and Retrieval for Image and Video Databases III*, volume 2420, pages 381–393. International Society for Optics and Photonics. 56

[Su et al., 2011] Su, C.-Y., Chang, R.-I., and Liu, J.-C. (2011). Recognizing text elements for svg comic compression and its novel applications. In *Proceedings of International Conference on Document Analysis and Recognition (ICDAR)*, pages 1329–1333, Washington, DC, USA. IEEE Computer Society. 67

[Suganthan et al., 1995] Suganthan, P. N., Teoh, E. K., and Mital, D. P. (1995). Pattern recognition by homomorphic graph matching using hopfield neural networks. *Image and Vision Computing*, 13(1):45–60. 33

[Sun et al., 2013] Sun, W., Burie, J.-C., Ogier, J.-M., and Kise, K. (2013). Specific comic character detection using local feature matching. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 275–279. 68

[Sun and Kise, 2009] Sun, W. and Kise, K. (2009). Speeding up the detection of line drawings using a hash table. In *Pattern Recognition, 2009. CCPR 2009. Chinese Conference on*, pages 1–5. 68, 69

[Sun and Kise, 2010] Sun, W. and Kise, K. (2010). Similar partial copy detection of line drawings using a cascade classifier and feature matching. In Sako, H., Franke, K., and Saitoh, S., editors, *ICWF*, volume 6540 of *Lecture Notes in Computer Science*, pages 126–137. Springer. 69

[Sun and Kise, 2011] Sun, W. and Kise, K. (2011). Similar Manga Retrieval Using Visual Vocabulary Based on Regions of Interest. In *2011 International Conference on Document Analysis and Recognition*, pages 1075–1079. Ieee. 64

[Sun and Kise, 2013] Sun, W. and Kise, K. (2013). Detection of exact and similar partial copies for copyright protection of manga. *International Journal on Document Analysis and Recognition (IJDAR)*, 16(4):331–349. 69

[Sundaresan and Ranjini, 2012] Sundaresan, M. and Ranjini, S. (2012). Text extraction from digital english comic image using two blobs extraction method. In *Pattern Recognition, Informatics and Medical Engineering (PRIME), 2012 International Conference on*, pages 449–452. 66, 67

[Swain and Ballard, 1991] Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International journal of computer vision*, 7(1):11–32. 49

[Takayama et al., 2012] Takayama, K., Johan, H., and Nishita, T. (2012). Face detection and face recognition of cartoon characters using feature extraction. In *Image Electronics and Visual Computing Workshop (IEVC'12)*, Kuching, Malaysia. 68

[Tanaka et al., 2007] Tanaka, T., Shoji, K., Toyama, F., and Miyamichi, J. (2007). Layout analysis of tree-structured scene frames in comic images. In *IJCAI'07*, pages 2885–2890. 63

[Thompson, 2007] Thompson, J. (2007). How manga conquered the us, a graphic guide to japan's coolest export. *Wired Magazine*. 3

[Torresani et al., 2008] Torresani, L., Kolmogorov, V., and Rother, C. (2008). Feature correspondence via graph matching: Models and global optimization. *Computer Vision–ECCV 2008*, pages 596–609. 28

[Tsai et al., 2014] Tsai, C.-Y., Lin, T.-C., Wei, C.-P., and Wang, Y.-C. F. (2014). Extended-bag-of-features for translation, rotation, and scale-invariant image retrieval. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6874–6878. IEEE. 54

[Tsai and Fu, 1979] Tsai, W.-H. and Fu, K.-S. (1979). Error-correcting isomorphisms of attributed relational graphs for pattern analysis. *IEEE Transactions on systems, man, and cybernetics*, 9(12):757–768. 32

[Tsai and Fu, 1983] Tsai, W.-H. and Fu, K.-S. (1983). Subgraph error-correcting isomorphisms for syntactic pattern recognition. *IEEE Transactions on Systems, man, and cybernetics*, (1):48–62. 32

[Ullmann, 1976] Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42. xi, 29, 31

[Van Gemert et al., 2010] Van Gemert, J. C., Veenman, C. J., Smeulders, A. W., and Geusebroek, J.-M. (2010). Visual word ambiguity. *IEEE transactions on pattern analysis and machine intelligence*, 32(7):1271–1283. 53

[Veltkamp and Tanase, 2001] Veltkamp, R. C. and Tanase, M. (2001). Content-based image retrieval systems: A survey. Technical report, Department of Information and Computing Sciences, Utrecht University. 61

[Vieux et al., 2012] Vieux, R., Benois-Pineau, J., and Domenger, J.-P. (2012). Content based image retrieval using bag-of-regions. In *International Conference on Multimedia Modeling*, pages 507–517. Springer. 56

[Viola and Jones, 2004] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154. 68, 69

[Vo et al., 2015] Vo, B., Nguyen, D., and Nguyen, T.-L. (2015). A parallel algorithm for frequent subgraph mining. In *Advanced Computational Methods for Knowledge Engineering*, pages 163–173. Springer. 97

[Wagner and Fischer, 1974] Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173. 34

[Wang and Hancock, 2004] Wang, H. and Hancock, E. R. (2004). A kernel view of spectral point pattern matching. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 361–369. Springer. 36

[Wang et al., 2010] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE. 54

[Wang et al., 2001] Wang, J. Z., Li, J., and Wiederhold, G. (2001). Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on pattern analysis and machine intelligence*, 23(9):947–963. 60

[Wang et al., 2012] Wang, S., Liu, D., Gu, F., and Feng Yang, H. (2012). Similar matching for images with complex spatial relations. *J. Comput. Inform. Syst*, 8:8727–8734. 57

[Washio and Motoda, 2003] Washio, T. and Motoda, H. (2003). State of the art of graph-based data mining. *Acm Sigkdd Explorations Newsletter*, 5(1):59–68. 21, 29

[Wayner, 1991] Wayner, P. (1991). Identification of artistic styles using a local statistical metric. In *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, volume i, pages 110–113. 80

[Williams et al., 2007] Williams, D. W., Huan, J., and Wang, W. (2007). Graph database indexing using structured graph decomposition. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 976–985. IEEE. 46

[Wilson and Hancock, 1996] Wilson, R. C. and Hancock, E. R. (1996). A bayesian compatibility model for graph matching. *Pattern Recognition Letters*, 17(3):263–276. 33

[Wilson and Hancock, 1997] Wilson, R. C. and Hancock, E. R. (1997). Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):634–648. 33

[Wiskott et al., 1997] Wiskott, L., Krüger, N., Kuiger, N., and Von Der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):775–779. 28, 68

[Witten et al., 2016] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann. 40

[Wong and You, 1985] Wong, A. K. and You, M. (1985). Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):599–609. 33

[Wong et al., 1990] Wong, A. K., You, M., and Chan, S. (1990). An algorithm for graph optimal monomorphism. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(3):628–638. 32

[Wyszecki and Stiles, 1982] Wyszecki, G. and Stiles, W. S. (1982). *Color Science*, volume 8. Wiley New York. 83

[Yamada et al., 2004] Yamada, M., Budiarto, R., Endo, M., and Miyazaki, S. (2004). Comic image decomposition for reading comics on cellular phones. *IEICE Transactions*, 87-D(6):1370–1376. 66

[Yan and Han, 2002] Yan, X. and Han, J. (2002). gspan: Graph-based substructure pattern mining. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 721–724. IEEE. 41, 43, 96

[Yan et al., 2004] Yan, X., Yu, P. S., and Han, J. (2004). Graph indexing: a frequent structure-based approach. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 335–346. ACM. 40, 41, 46

[Yanagisawa et al., 2014] Yanagisawa, H., Ishii, D., and Watanabe, H. (2014). Face detection for comic images with deformable part model. In *4th IIEEJ International Workshop on Image Electronics and Visual Computing*. 72

[Yang et al., 2009] Yang, J., Yu, K., Gong, Y., and Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE. 54

[Yang et al., 2013] Yang, X., Prasad, L., and Latecki, L. J. (2013). Affinity learning with diffusion on tensor product graph. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):28–38. 113

[Zhang and Lu, 2004] Zhang, D. and Lu, G. (2004). Review of shape representation and description techniques. *PR*, 37(1). 49

[Zhang et al., 2009] Zhang, S., Li, S., and Yang, J. (2009). Gaddi: distance index based subgraph matching in biological networks. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pages 192–203. ACM. 46

[Zhang and Yang, 2008] Zhang, S. and Yang, J. (2008). Ram: Randomized approximate graph mining. *Lecture Notes in Computer Science*, 5069:187–203. 45

Représentation par graphes et Fouille de graphes : Application à la Recherche d'Images de Bandes Dessinées par le Contenu

Résumé :
Nous abordons d'abord le problème de la représentation sous forme d'un graphe de l'image et de ses applications en reconnaissance des formes, en mettant l'accent sur les applications de recherche d'images basées sur le contenu (CBIR). Les images utilisées dans cette thèse sont des images de bandes dessinées, qui possèdent des spécificités qui sont des freins pour les méthodes de recherche d'information par le contenu utilisées dans la littérature. Le contenu, des bandes dessinées, tel que les objets et les personnages, est complexe. La représentation des personnages, par exemple, peut, d'une case à l'autre, varier énormément en taille et avec différents effets de perspective, selon la situation que l'auteur souhaite retranscrire. Nous proposons ainsi une représentation qui permet d'obtenir des graphes stables et qui conserve des informations structurelles de haut niveau pour les objets d'intérêt dans les images de bandes dessinées. Ensuite, nous étendons le problème d'indexation et d'appariement aux structures de graphes représentant les images d'une bande dessinée et nous l'appliquons au problème de la recherche d'information. Un album de bandes dessinées est ainsi transformé en une base de graphes, chaque graphe correspondant à la description d'une seule case. La stratégie utilisée pour retrouver un objet ou un personnage donné, consiste donc à rechercher des motifs fréquents (ou des sous-structures fréquentes) dans cette base de graphes. Cette étape nécessite de surmonter le problème de non-répétabilité provoqué par les erreurs introduites dans la structure du graphe pendant la phase de construction dues notamment à la variabilité des dessins. Il apparait donc un écart sémantique entre le graphe et le contenu de l'image de bande dessinée.

Mots clés : mise en correspondance de graphes, modélisation par les graphes, fouille de graphes, fouille de motifs fréquents, reconnaissance des formes, reconnaissance d'objets déformables, recherche d'images par le contenu, recherche d'images de bandes dessinées.

Graph Representation and Mining Applied in Comic Images Retrieval

Summary:
In information retrieval tasks from image databases where content representation is based on graphs, the evaluation of similarity is based both on the appearance of spatial entities and on their mutual relationships. In this thesis we present a novel scheme of Attributed Relational Adjacency Graphs representation and mining, which has been applied in content-based retrieval of comic images. The images used in this thesis are comics images, which have their inherent difficulties in applying content-based retrieval, such as their abstractness, partial occlusion, scale change and shape deformation due to viewpoint changes. We propose a graph representation that yields stable graphs and allow to retain high-level and structural information of objects of interest in comic images. Next, we extend the indexing and matching problem to graph structures representing the comic image, and apply it to the problem of retrieval. The graphs in the graph database representing the whole comic volume are then mined for frequent patterns (or frequent substructures). This step is to overcome the non-repeatability problem caused by the unavoidable errors introduced into the graph structure during the graph construction stage, which ultimately create a semantic gap between the graph and the content of the comic image. Finally, we demonstrate the effectiveness of the system with a database of annotated comic images. Experiments of performance measures is addressed to evaluate the performance of this CBIR system.

Keywords: image processing, graphics recognition, document analysis, comics understanding.

Laboratoire L3i - Informatique, Image, Interaction

Pôle Sciences et Technologies, Université de La Rochelle,
avenue Michel Crépeau

17042 La Rochelle - CEDEX 01 - France