Florida International University
FIU Digital Commons

FIU Electronic Theses and Dissertations | University Graduate School

11-12-2015

Next Generation of Product Search and Discovery

Kaiman Zeng
kzeng001@fiu.edu

DOI: 10.25148/etd.FIDC000207

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

NEXT GENERATION OF PRODUCT SEARCH AND DISCOVERY: VISUAL SEARCH AND RECOMMENDATION

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Kaiman Zeng

2015

To: Interim Dean Ranu Jung
College of Engineering and Computing

This dissertation, written by Kaiman Zeng, and entitled Next Generation of Product Search and Discovery: Visual Search and Recommendation, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Jean H. Andrian

Hai Deng

Deng Pan

Bogdan Carbunar

Kang K. Yen, Major Professor

Date of Defense: November 12, 2015

The dissertation of Kaiman Zeng is approved.

Interim Dean Ranu Jung
College of Engineering and Computing

Dean Lakshmi N. Reddi
University Graduate School

Florida International University, 2015

DEDICATION

To my parents. Thank you for always being there for me.

To my husband, who supports and believes in me unconditionally.

To my son and daughter. You are the magic of my life.

ACKNOWLDGMENTS

First of all, I would like to express my most sincere and deepest gratitude to my major professor Dr. Kang K. Yen. Without his support, patience, and understanding throughout my Ph.D. study, I would not be able to have the opportunity building this research work I am passionate about. His expertise and guidance sharpen my critical thinking skills and scientific writing skills. To my committee, Dr. Jean H. Andrian, Dr. Hai Deng, Dr. Deng Pan, and Dr. Bogdan Carbunar, I am extremely grateful for their assistance and suggestions during my completion of the dissertation.

I would also like to thank my colleagues Dr. Nansong Wu, Dr. Arman Sargolzaei in the System Dynamic Lab, and Dr. Lei Li in Samsung Research for their inspiring and invaluable discussions with me, which help expanding my knowledge and discovering interesting research directions.

Further, I would like to thank the department of Electrical and Computer Engineering for the stimulating and thoughtful curriculum, as well as providing me with a teaching assistantship in 2012-2014. My acknowledgment is also extended to the Applied Research Center and Dr. Jeffrey Fan for supporting me with a graduate assistantship in 2010-2012, which helps me start my Ph.D. study.

ABSTRACT OF THE DISSERTATION

NEXT GENERATION OF PRODUCT SEARCH AND DISCOVERY: VISUAL SEARCH AND RECOMMENDATION

by

Kaiman Zeng

Florida International University, 2015

Miami, Florida

Professor Kang K. Yen, Major Professor

Online shopping has become an important part of people's daily life with the rapid development of e-commerce. In some domains such as books, electronics, and CD/DVDs, online shopping has surpassed or even replaced the traditional shopping method. Compared with traditional retailing, e-commerce is information intensive. One of the key factors to succeed in e-business is how to facilitate the consumers' approaches to discover a product. Conventionally a product search engine based on a keyword search or category browser is provided to help users find the product information they need. The general goal of a product search system is to enable users to quickly locate information of interest and to minimize users' efforts in search and navigation. In this process human factors play a significant role. Finding product information could be a tricky task and may require an intelligent use of search engines, and a non-trivial navigation of multilayer categories. Searching for useful product information can be frustrating for many users, especially those inexperienced users.

This dissertation focuses on developing a new visual product search system that effectively extracts the properties of unstructured products, and presents the possible

items of attraction to users so that the users can quickly locate the ones they would be most likely interested in. We designed and developed a feature extraction algorithm that retains product color and local pattern features, and the experimental evaluation on the benchmark dataset demonstrated that it is robust against common geometric and photometric visual distortions. Besides, instead of ignoring product text information, we investigated and developed a ranking model learned via a unified probabilistic hypergraph that is capable of capturing correlations among product visual content and textual content. Moreover, we proposed and designed a fuzzy hierarchical co-clustering algorithm for the collaborative filtering product recommendation. Via this method, users can be automatically grouped into different interest communities based on their behaviors. Then, a customized recommendation can be performed according to these implicitly detected relations. In summary, the developed search system performs much better in a visual unstructured product search when compared with state-of-art approaches. With the comprehensive ranking scheme and the collaborative filtering recommendation module, the user's overhead in locating the information of value is reduced, and the user's experience of seeking for useful product information is optimized.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Statement of Problem

In recent years online shopping has become an important part of people's daily life with the rapid development of e-commerce. As a new fashion of shopping, online shopping has surpassed or even replaced the traditional shopping method in some domains such as books, CD/DVDs, etc. Product search engine, as an imperative part of the online shopping system, greatly facilitates users retrieving products and doing shopping. The traditional product search is based on keyword or category browser. Since in e-commerce products are usually presented in the form of images, the visual search, which retrieves an image by its visual contents and features, brings a new insight into product search and attracts growing interests among researchers and practitioners. In particular, mobile visual search is one of the promising areas because of the increasing popularity and capability of mobile devices. And the number of consumer applications that are built on visual search technology has increased substantially recently. Some well-known examples of visual search applications in industry include Google Goggles, Bing Image Search, Amazon Flow, and Kooaba. Other examples in the fashion domain include Neiman Marcus's Snap Find Shop powered by Slyce, and Macy's visual search engine supported by Cortexica. Another important component of online shopping website is the product recommendation service. Recommendation becomes a mainstream feature in nowadays e-commerce because of its significant contributions in promoting revenue and customer satisfaction. In some scenarios, search is difficult for users, while exploration is easy. Recommendation provides users a tool to discover what they might like. It brings products to the user without any manual search effort from users. This increases the number of available options for the user's purchasing decisions, and potentially unlocks user's hidden purchases. According to Forrester report [1], 15% of consumers explicitly admitted that they purchased when they saw recommendations on a page. Amazon's conversion to sales of on-site recommendations could be as high as 60% in some cases based off the performance of other e-commerce sites. Besides, recommendation engine offers users a personalized shopping experience by analysis of users' on-site behavior to understand their preferences, which is able to improve customers' retention. The better a user is understood, the better the service can be provided. The more a user like the website, the higher he/she is likely to return and buy. As reported by Forrester [1], up to 20% of retailers revenue could be attributed to personalized product recommendations.

Most visual search applications involve a process of taking a photo of the real world, recognizing objects in the photo, and retrieving the corresponding information and metadata about these objects from an online database or a local database [2]. Analogous to typing a text query into a search engine to retrieve desired information, visual search uses a photo as a visual query to search an image database. These applications are powered by computer vision algorithms that perform image matching on visual data captured by camera. Typically, the matching query and database images will be separated by severe geometric and photometric distortions, so the search algorithm must be robust against these issues. A popular approach in computer vision is to describe an image in terms of scale and rotation invariant local image features. In this approach, interest points are detected at different locations and scales in the image by interest point detectors. These interest points are designed to be highly repeatable and detectable under various viewing conditions. An image descriptor is then extracted from a local image patch centered at each interest point. Usually represented as a high-dimensional vector, the image descriptor is designed to be discriminative, so that only descriptors from accordant image patches are matched. When image searching is performed on a large database, a set of local features from a query image can be compared with sets of local features from database images in order to reliably determine which database image best matches the query image. When the image database contains primarily non-rigid objects, the retrieval performance of the image feature based approach is typically poor. One reason for this poor performance is that the interest point detectors, which are commonly used in visual, are not designed to explicitly capture the salient characteristics of non-rigid objects. Research [3], [4] have concentrated developing local features on rigid objects of fixed or known a priori shape to simplify the task of object recognition. These approaches have difficulty in dealing with unstructured objects, and thus cannot be applied to more generic categories of objects.

Recommending products require understanding large-scale data of user logs and product records, such as user purchase patterns, product attributes, price ranges, and product categories. Given hundreds of millions of user activity logs and product items, accurate and efficient recommendation is a challenging computational task. Data mining provides possible tools to tackle this problem. These algorithms include clustering, classification techniques, the generation of association rules, and the construction of similarity graphs through techniques such as Horting [5]. There is a broad spectrum of data and activities that are logged by the server. Profile, purchases, reviews, queries, and click logs are some of the examples. These are typical representative resources to extract patterns of user behavior in recommendation systems. According to survey by Brand Reputation 84% of consumers said they were more likely to check online for reviews prior to making a purchase [6]. Also, according to a survey report by E-consultancy, high product ratings will increase the likelihood of purchases for more than 55% of consumers [7]. As a result, in this research we will focus on analyzing rating data from users and develop clustering techniques to automatically extract patterns undermined in users' behaviors.

In this dissertation, to establish an effective, efficient and comprehensive product search system especially for unstructured products such as clothes, three major challenges have been considered:

- **Image distortion.** Since the query images are usually taken from different positions and orientations, we require feature descriptors invariant with respect to projective transformations. First, there are large variations and deformations in appearance within the object classes. Second, there are value changes in the image color space, which are due to illumination variations, different gamma corrections, different cameras, etc.

- **Textual data association.** It is important to address the issue of how to leverage the rich contextual information in a visual computational model to build more robust visual search systems and to better satisfy the user's need and intention.
- **Rich user logs.** E-commerce generates tremendous amount of user access logs. It is not a trivial task to automatically extract useful information from the raw activities carried out by users. What is the relationship between users and their behaviors? How could interesting patterns previously unknown and potentially useful be discovered? With these discoveries, how could one customize and optimize the navigation of a visit to the website?

This dissertation focuses on developing a new visual search system that effectively exploits the properties of unstructured products, and addresses the three major technical challenges and problems discussed above. We design and develop a feature extraction algorithm that retains product color and local pattern features, and the experimental evaluation on the benchmark dataset demonstrated that it is robust against common geometric and photometric visual distortions. Besides, instead of ignoring product text information, we investigate and develop a ranking model learnt via unified probabilistic hypergraph that is capable of capturing correlations among product visual content and textual content. Moreover, we propose and design a fuzzy hierarchical co-clustering algorithm for collaborative filtering product recommendation. Via this method, users can be automatically grouped into different interest communities according to their rating activities, and customized recommendation can be performed based on these implicitly detected relations.

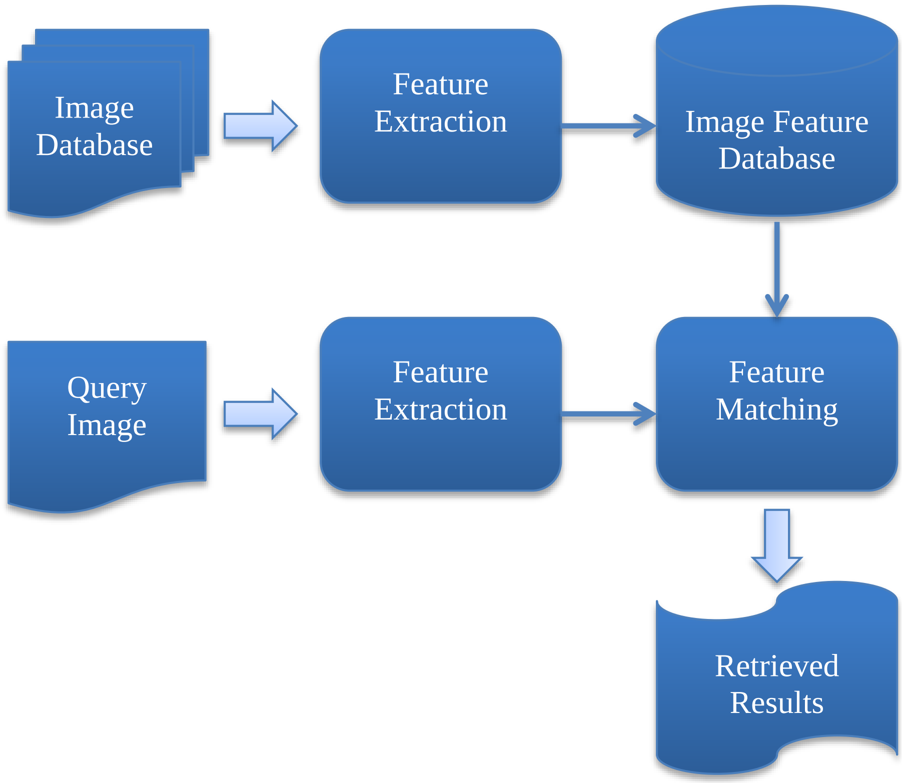


Figure 1.1 Pipeline of conventional content-based image retrieval

The developed visual search system has a hierarchy that is similar to the conventional visual feature based approach. The pipeline of conventional visual content-based image retrieval system is shown as in Figure 1.1. There are three main differences between the conventional system and our proposed system. First, the traditional grey scale feature extraction is replaced with a color-fused algorithm. Then, a new hypergraph ranking model is brought in over the traditional visual matching ranking. Third, the product search engine is reinforced by adding a novel collaborative filtering recommendation module to complement user’s product search experience. By explicitly exploring fashion products properties, the developed system can perform much better in visual product search when compared with other approaches. With the recommendation module, the customized product discovery is facilitated and the search time users spend on product exploration will be saved.

1.2 Contribution

The main contributions of this dissertation are described as follows:

- **A novel color boosted local pattern feature descriptor**. We invented a new color SIFT descriptor that captures salient local pattern patches around the color interest points. The designed color keypoint detector introduced the amended color histogram factor into the SIFT, which is resistant to color variations due to imaging condition changes. The developed descriptor not only retains sufficient color information but also is robust to photometrical variations.
- **An efficient ranking model to integrate visual feature and textual metadata.** We developed unified probabilistic hypergraph ranking algorithm, which, modeling the correlations among product visual features and textual features, extensively enriches the description of the image and extensively improve the retrieval performance over the visual distance based ranking. Once the hypergraph model is learnt, re-ranking the visual search results is efficient.
- **A new hierarchical clustering approach for collaborative filtering product recommendation.** We proposed to use hierarchical co-clustering to detect possible interest groups underlying users' behavior data. A soft hierarchical clustering is further devised to flexibly assign different types of resources into several groups, allowing one entity appearing in multiple co-clusters. In this way, we can easily capture the inner relationship among different types of data

resources simultaneously. A case study on recommendation is performed and shows that the soft hierarchical clustering algorithm can provide more meaningful recommendation results.

1.3 Organization

The remaining chapters of the dissertation are organized as follows:

- **Chapter 2** summarizes prior research relevant to the thesis. It focuses on the related work for image feature-based search, visual feature detection, ranking in image retrieval systems, and recommendation algorithms.
- **Chapter 3** introduces the color histogram detector that we have developed for product image matching and describes how the local pattern descriptor is formed from the SIFT descriptor. It presents the design of the detector and the pipeline that is used to perform local pattern matching with the SIFT. Breakdowns of the matching performance and retrieval performance compared to other state-of-the-art approaches are given.
- **Chapter 4** presents the image retrieval system that uses unified probabilistic hypergraph learning to rank the retrieved images from databases. It describes how the ranking model embraces visual features and textual features into a specific unified hypergraph, and how the high-order hyperedges are formed from various products' information including visual content, product categorization labels, and product descriptions. Evaluation results of retrieval experiments for unlabeled user captured product images are presented.

- **Chapter 5** introduces the product recommendation module that we have proposed to assistant user's product query and exploration. It describes how the soft hierarchical co-clustering algorithm is designed to organize two different data resources into one tree structure. It also discusses a hybrid similarity measurement to balance different types of data resources. The chapter then presents the evaluation of the recommendation performance of the proposed Fuzzy Hierarchical Co-Clustering method and compares it to other traditional collaborative filtering methods.
- **Chapter 6** concludes the dissertation with a discussion of its major findings and lessons learned, and proposes possible future research directions.

CHAPTER 2

BACKGROUND

This chapter describes previous research and approaches that are relevant to the topics of the thesis. The general problems of product search system are discussed in Section 2.1. In order to tackle these problems, we propose several possible research directions. Previous works in the proposed areas are reviewed in Section 2.2. Section 2.2.1 discusses the general image retrieval techniques based on textual metadata or visual content, especially the content-based image retrieval. Then, Sections 2.2.2 and 2.2.3 present related research on two essential visual features in building our visual product search system: (1) color feature extraction, and (2) local feature detection and description. We discuss the conventional ranking model and the learn-to-rank ranking model in Section 2.2.4. And finally Section 2.2.5 reviews partitional and hierarchical clustering algorithms, as well as their applications in recommendation.

2.1 Challenges in Product Search

Not too many years ago, most people shopped in local stores, and were faced with problems of parking, weather, long lines, wobbly shopping carts and so on. Even after the emergence of online shopping, people felt uncomfortable and insecure giving out their credit card details and personal information. But now things have all changed. All over the world, online buying has grown in an exponential manner. Online shopping has become part of people's daily life. Apparel, computers and consumer electronics are three dominant purchases in current online sales [8]. Compared with traditional retailing, e-commerce is information intensive. Thus, one of the key factors to succeed in business and maintain competitive advantage is how to facilitate the consumers' approaches to product search.

A product search system is such a system integration of hardware and software that help users find product information they need. One type of product search system is integrated in the websites of online retailer, and provides the functionality of searching product inventory of this specified online retailer. Another type of product search system helps users find and buy products from various online retailers, and allows users to compare prices among different retailers. The general goal of a product search system is to enable users to quickly locate information of interest and to minimize users' efforts in search and navigation. The time, which the system requires to search and locate the needed information, can be defined as the effort from a user's perspective. The duration starts when a user initiates a query, and ends when he/she terminates the interaction with the system. It could end up with some positive feedbacks that the user finds the items of interest, or negative feedbacks that the user fails to do so and feels disappointed with the returned results. In this process human factors play a significant role. Finding product information could be a tricky task and may require an intelligent use of search engines, and a non-trivial navigation of multilayer categories. Search for useful product information can be frustrating for many users, especially those inexperienced users. They may practically unable to obtain the information they are seeking for, even though such products are available in the e-commerce website. In Ward and Lee's research, it is reported that an easy to use search engine promotes the product visibility, and alleviates the need for advertisement [9].

Conventional product search engine, origin from document search, performs searches on the words or phrases through a large index of the products repository. The query words or phrases are compared with each of the product records in the database. Since in the e-shopping environment a large collection of product information is available for the search engine to access, metadata are normally used to represent product information and an index of metadata is utilized to achieve retrieval efficiency. Typical metadata could include product identifier, bar code indicator, product form, measure, title, subjects, price, URL, related products and so on. Some search engines also use textual description of product and digital image of product. Then, an inverted index is created to accelerate the search on these product metadata records. It is important to make the basic indexable metadata distinctive, so that the original product information will be pre-processed to create the searchable index. The original product information is always kept for display purpose, and the terms of metadata can refer to the original product information.

In this thesis, we will consider two fundamental problems in product search. The first one is: what users enter into a product search system. The other one is: what users get from the product search system. In order to answer these two questions, we would like to discuss two primary challenges faced in product search system. The first challenge is the gap between the ways a user describes the product he/she is seeking for and the product is presented in the e-commerce system; namely, the difference between the expressions of the user and the database administrator. When an administrator creates a new record of a product item, they will describe the products in words how they want the product value to be distributed and spread out. Users tend to use vocabulary that they are accustomed to express their needs. Besides, users' knowledge, capability of expression, and the ambiguity of language will also affects the semantic mismatch. In some cases, even though both administrator and user know the same vocabulary, the query keywords varies in which term is more generally used to represent the same concept, e.g. notebook versus laptop, cellphone versus mobile phone, and trike versus tricycle. In recent years, there is an increasing attention on using an image as the input of a query, which limits the impact of human factors on search results. In most e-commerce websites, product image is an essential part of product information. Starting a search with a product image opens up new windows of helping shoppers find products, and adds a new dimension of complexity in search specification. Moreover, query by image provides a simple, intuitive and highly versatile means of search, especially for current generation of web searchers, who value process simplicity and efficiency higher than the quality of results [10]. As a result, we will focus our research on the visual content-based information retrieval for product search in this dissertation. And we will discuss the related technologies in Section 2.2.1.

Another major challenge we want to discuss in product search system is how to effectively present the possible items of attraction to users so that the users can quickly locate the ones they would be most likely interested in. Most product search systems provide the retrieval results in an order of the relevance to the user's query. However, users tend to have low threshold of frustrating and have little patience to browse long list of retrieval results in a linear sequential order. So if there are too many redundant information or false hits on the first page of returned results, the user will be unable to get what they are looking for even though the optimum set of items of interest has been returned. How to optimize the ranks of returned items is one of the possible solutions to this issue. Related technologies of ranking are discussed in Section 2.2.4. In addition, some e-retailers, e.g. Amazon.com, offer the shopper a recommendation tool to discover items of most likely interest. Human brains naturally have the ability to parallel process information. It is feasible and efficient to have the users focus and localize on the items of interest from two sources: retrieval and recommendation at the same time. The display of product search is optimized rather than simply showing a sequential retrieval results. We discuss the recommendation related technologies in Section 2.2.5.

2.2 Related Technologies in Product Search

2.2.1 Content-based Image Retrieval

Traditional image retrieval is mostly based on textual descriptions, e.g. metadata, of the images. Parts of the metadata are automatically generated when capturing the image, such as date, time, device information, and resolution. Currently the GPS enabled devices can record the geographic location to image files. However, these metadata are irrelevant to image contents and semantics. So people can also manually assign text tags to their image contents to facilitate the organization and later search. Flickr, Facebook, and Instogram have shown that the image annotation is helpful in managing image data. One problem of this traditional method is that the quality of the search result highly depends on the quality and accuracy of text metadata of images. Also un-annotated images are hardly retrieved. It retrieves images that are semantically related to the user's query from a database.

An alternate method is to search images according to their visual contents, a.k.a. Content-Based Image Retrieval (CBIR). The visual content of the image can refer to color, texture, shape or any other features that can be automatically extracted from the image itself. CBIR system retrieves images that are similar to the user's query in one or more forms of visual features, and then ranks the retrieved results according to the similarity measure. The similarity measure is calculated between the query image and images in the database and determines their relevance. Color, shape, texture, and local feature are some of the most common features in search. The search results greatly depend on the detected feature used and may vary tremendously. We will focus on image color feature and local feature in this thesis. Color is an important characteristic of an image, and has been applied in the field of CBIR for a long run. The local feature based methods extract local patterns around specific keypoints, and handle occlusion and clutter better compared to the global feature based methods [11]. This type of feature extraction method has proven to perform notably better in the area of object recognition [3, 4, 12]. In Section 2.2.2 and Section 2.2.3, we will discuss the related work in the field of color feature and local visual feature extraction, respectively.

2.2.2 Color Feature

In CBIR systems, color is the commonly adopted feature to represent the characteristics of an image. The color feature is relatively less dependent on image size, orientation and perspective compared with other visual features. The color feature extraction is simple to compute, and has therefore received extensive and intensive research in image processing and computer vision. Researchers have explored many techniques to categorize the color into different color spaces. Red, Green, and Blue (RGB) space is a well-known common color space for the public because it works similarly as the human visual system. Mixing three primary colors, e.g. red, green, and blue, can generate 16.7 billion colors. The RGB color space is a commonly used color language in electronic input and output devices, such as displays, scanners, and cameras. The RGB color space has many variants including ISO RGB, Extended ISO RGB, standard RGB (sRGB), Adobe RGB, Apple RGB, NTSC RGB, etc. HSV color space was developed in the 1970s for computer graphics applications. H, S, and V represent Hue, Saturation, and Value, respectively]. Here, Hue indicates the color type, Saturation indicates the color purity, and Value indicates the color brightness. The HSV color space closely models the natural human perception. Each point in the RGB color space can be mapped into a point in the HSV color space. The transform between the RGB color space and HSV color space is nonlinear and invertible. First we change the RGB color range from [0, 255] to [0, 1] through dividing R, G, and B values by 255:

$$\begin{aligned}
R' &= R/255 \\
G' &= G/255 \\
B' &= B/255 \\
C_{\max} &= \max(R', G', B') \\
C_{\min} &= \min(R', G', B') \\
\Delta &= C_{\max} - C_{\min}
\end{aligned}$$

Then, hue calculation is defined as:

$$H = \begin{cases} 60^\circ \times \left(\dfrac{G' - B'}{\Delta} \bmod 6 \right), & C_{max} = R' \\ 60^\circ \times \left(\dfrac{B' - R'}{\Delta} + 2 \right), & C_{max} = G' \\ 60^\circ \times \left(\dfrac{B' - G'}{\Delta} + 4 \right), & C_{max} = B' \end{cases}$$

Saturation is calculated by:

$$S = \begin{cases} 0, & C_{max} = 0 \\ \dfrac{\Delta}{C_{max}}, & C_{max} \neq 0 \end{cases}$$

And value is defined as:

$$V = C_{max}$$

Compared with the RGB color space, the HSV information in three channels is relatively independent to each other.

Color histogram is another widely used color features. It represents the color distributions in an image, and normally refers to the pixel counts at different intensity values found in the image. The color histogram can be constructed on any types of color spaces. More often it is used for three-dimensional spaces like RGB and HSV. The color histograms can be taken in a 3-D histogram, where there are three axes representing three-dimensional spaces, and a brightness at each point representing the number of pixel counts. The joint probabilities of the intensities of the three-color channels are captured by the 3-D color histogram. The color histograms can also be taken in either individual channel of color spaces. We can normalize the color histogram in the range of [0, 1] to generate the normalized color histogram by dividing the total number of pixels N in the image:

$$H_{X,Y,Z}(a,b,c) = \frac{1}{N} \times Counts(X = a, Y = b, Z = c)$$

where X, Y, and Z respectively represent the three different color channels, such as R, G, and B in the RGB color space, and H, S, and V in the HSV color space. N is the total number of pixels in the image.

2.2.3 Local Feature

Local feature, representing local patches of an image, has shown promise in many tasks of computer vision, such as image match, object recognition, image registration and so on. Feature detection is utilized as the initial step in local feature extraction algorithms. It is a classic research area in image processing and computer vision. And there are a variety of different types of features, e.g. edges, corners/keypoints, regions of interest and ridges. The corner/keypoint is treated as the same concept since a corner can be not only considered as an intersection of two lines, but also a point that has two different edge directions within a local window of the point. Likewise, a keypoint can be defined as a corner, line endings, a point of local intensity maximum or minimum, or a point on a curve where the curvature is local maximum. As a result, the corner/keypoint detection is mainly divided into edge-based method and gray density based method. Current research is focused on gray density based corner/keypoint detection, since a small degree variation of the target object lead to great difference in edge extraction, and the edge extraction is computationally expensive [13, 14]. Gray density based approach detects the corner/keypoint by calculating the curvature and gradient of points. Moravec operator, Forstner operator, Harris operator and SUSAN operator are some of the examples. Harris operator [15] is the most classic detector among them. Mikolajczyk takes the scale space theory into consideration and proposes Harris-Laplace detector, which applies Laplace-of-Gaussian (LoG) for automatic scale selection [16]. It obtains scale and shape information and can represent local structure of an image. Lowe applies Difference-of-Gaussian (DoG) filter, an approximate to LoG, in the SIFT algorithm to reduce computational complexity [12]. Also, in order to increase the algorithm efficiency, Hessian Affine, FAST, Hessian-blobs, and MSER are further proposed. In [17], Mikolajczyk et al. extract 10 different keypoint detectors within a common framework and compare them for various types of transformations. Van de Sande extracts 15 types of local color features, and examines their performance on transformation invariance for image classification. Many detection methods are studied seeking a balance between keypoint repeatability and computational complexity [18].

After the keypoint detection, we compute a descriptor on the local patch. Feature descriptors can be divided into gradient-based descriptors, spatial frequency based descriptors, differential invariants, moment invariants, and so on. Among them, the histogram of gradient-based method has been wildly used. The gradient histogram is used to represent different local texture and shape features. The Scale Invariant Feature Transform (SIFT) descriptor proposed by Lowe is a landmark in research of local feature descriptor. It is highly discriminative and robust to scaling, rotation, light condition change, view position change, as well as noise distortion [12]. Since then, it has drawn considerable interests and a larger number descriptors based on the idea of SIFT emerges. SURF [19] uses the Haar wavelet to approximate the gradient SIFT operation, and uses image integral for fast computation. DAISY [20, 21] applies the SIFT idea for dense feature extraction. The difference is that DAISY use Gaussian convolution to generate the gradient histogram. Affine SIFT [22] simulates different perspectives for feature matching, and obtains good performance on viewpoint changes, especially large viewpoint changes. Since SIFT works on the gray-scale model, many color-based SIFT descriptors are proposed to solve the color variations, such as CSIFT, RGB-SIFT, HSV-SIFT, rgSIFT, Hue-SIFT, Opponent SIFT, and Transformed-color SIFT [18, 23, 24]. Most of them are obtained by computing SIFT descriptors over channels of different color space independently; therefore they usually have higher dimension (e.g. 3×128 dimension for RGB-SIFT) descriptors than SIFT. Song et al. proposed compact local descriptors using an approximate affine transform between image space and color space [25]. Burghouts et al. performed an evaluation of local color invariants [26].

2.2.4 Image Ranking

Ranking and hypergraph learning are two research field related to our work. These two topics have received intensive attention in information retrieval and machine learning. The conventional image ranking is developed from textual retrieval. The ranking model is defined based on the bag-of words, e.g. BM25 [27], the Vector Space Model [28], and the Language Modeling for Information Retrieval [29]. Another type of ranking model is based on hyperlink analysis, such as HITS [30], PageRank [31], and its variations [32-34].

In the CBIR systems, the ranking is commonly obtained from the similarity measure of adopted visual features. One type of similarity measures is calculated from Minkowski distance, Cityblock distance, Infinity distance and Cosine distance. They are usually called Minkowski and standard measures. Statistical measure, e.g. Pearson coorelation coefficient, and Chi-square dissimilarity, is another type of similarity or dissimilarity measure methods. The third type of similarity measures is divergence measure, which includes Kullback-Liebler divergence, Jeffrey divergence, Kolmogorov-Smirnov divergence, Cramer-von Mises divergence and so forth. There are some other measures, such as Earth Mover's distance, diffusion distance and so on [35, 36].

The learning to rank model, has gained increasing attention in recent years, utilizing machine learning algorithms to optimize the ranking function by tuning some of the parameters and incorporating relevance features [37, 38]. Manifold ranking [39], a graph-based semi-supervised learning method ranks the data through exploiting their intrinsic manifold structure. Manifold ranking was firstly applied to CBIR in [40], and significantly improved image retrieval performance. Liu et al. [41] proposed a graph based approach for tag ranking, by which a tag graph was built to mine the correlations among tags, and the relevance scores were obtained through a random walk over the similarity graph. These researches demonstrated the effectiveness of graph-based semi-supervised learning techniques in solving different ranking problems. However, they are inadequate for the relations in images via pairwise graphs solely. It would be of great benefit to take into consideration of the relationship among 3 or more vertices. Such a model capturing higher order relations is called hypergraph. In a hypergraph, a non-empty set of vertices is defined as a weighted or un-weighted hyperedge; the magnitude of the weight represents the degree that the vertices in the hyperedge belong to the same cluster. Agarwal et al. [42] firstly introduced hypergraph to computer vision, and proposed a clique averaging graph approximation scheme to solve the clustering problems. Literature [43] formulated the probabilistic interpretation based image-matching problem as the hypergraph convex optimization, and achieved a global optimum of the matching criteria. However, they set up three restrictions that are the same degree of all hyperedges, the same number of vertices in two graphs, and a complete match. Sun et al. [44] employed the hypergraph to capture the correlations among different labels for multi-label classification. The proposed hypergraph learning formulation showed the effectiveness on large-scale benchmark data sets, and its approximate least squares formulation maintained efficiency, as well as competitive classification performance. One shortage of their work is that they limited the target applications to linear models, and thus did not have a general performance evaluation on other multi-label applications, such as the kernel-induced multi-labels. In [45] the spatio-temporal relationship among different patches are captured by the hypergraph structure, and the video object segmentation is modeled as hypergraph partition. Further, weights are added on important hyperedges. The experimental results have shown good segmentation performance on nature scenes. In the case that there are several different types of vertices or hyperedges, the hypergraph is called unified hypergraph. Li. et al. [46] proposed a unified hypergraph model for personalized news recommendation where users and multiple news entities are involved as different types of vertices, and their implicit correlations are captured. The recommendation is modeled as a hypergrph ranking problem. The hypergraph learning algorithms have demonstrated their capability of capturing complex high-order relations. Their applications in image retrieval are also promising. In [47], a probabilistic hypergraph is built for image retrieval ranking. The hyperedge is formed by a centroid image and its *k*-nearest neighbors based on their visual similarity. Gao et al. [48] proposes a hypergraph learning algorithm for social image search, where the weight of hyperedges, representing the impact of different tags and visual words, is automatically learned with a set of pseudo-positive images. They both use the visual content as hyperedges, while are lack of establishing correlations between visual content and text content. As a result, the search must start with a user assigned keyword.

2.2.5 Clustering in Recommendation

The spectrum of different search strategies of consumers may range from a purposeful specific item search to the aimless browsing for “something interesting”. A search could start with general browsing and turn out to be a very focused outcome. The initial intent of search can also be a very targeted object and then extend to many more resources. The directed search and general browsing could interconvert at any point of their progress. In literature [49], users on e-commerce websites are categorized into five types: directed product searchers, directed buyers (directed searchers with a buying intent), browsers, bargain hunters (a type of browsers), and entertainment seekers. A user’s initial directed search does not necessarily bring satisfactory results. Recommendation system, as a value added service to conventional search, provides an effective tool to extend users’ search and help discovering users’ potential interests.

One of the successful pioneers in recommender technologies is collaborative filtering (CF) [50-52]. CF starts by constructing a database based on individual consumers' preferences of products. Statistical data is used to find consumers, who have history similar to the target customer. For example, they rate different products similarly and buy similar set of products. These set of customers are called neighbors. Once a neighborhood of customers is formed, these systems use algorithms to produce recommendations, and then recommend products to target customers based on the opinions of others.

In the recommender systems, clustering is one of the data mining techniques that are usually used to form neighborhoods. Compared with other CF techniques, such as methods based on correlation criteria, non-negative matrix factorization, or singular value decomposition, clustering methods has lower computational cost, and thus have received extensive research from a number of different recommender domains. With the purpose of discovering natural or meaningful groups, the unsupervised learning assigns items to groups based on similarity. The similarity is determined by a distance measure, such as Euclidean distance, Minkowski distance, Mahalanobis distance, Cosine similarity, Pearson correlation, and so on. The clustering algorithm can minimize intra-cluster distances and also maximize inter-cluster distances.

Partitional clustering and hierarchical clustering are two major types of clustering methods. Partitional clustering algorithms divide data items into a certain number of disjoint or overlapping clusters. Hierarchical clustering algorithms consecutively cluster items in the clusters it detected, and produce a set of nested clusters that are organized into a hierarchical tree. A classic partitioning algorithm called *K*-means clustering partitions a dataset into predefined *k* clusters by minimizing the distance between each data point. Ungar and Foster [53] presented a repeated *k*-means and Gibb sampling clustering technique to cluster users with similar items. Xue et al. [54] proposed a later commonly used clustering method in the context of recommendation systems. This method introduces the *k*-means algorithm as a pre-processing step to help forming the cluster neighborhood. The distances between the user and the centroids of different cluster are used as the pre-selection criterions for neighbors. They also suggested a cluster-based smoothing technique. In this technique, the missing values for users in a cluster are replaced by cluster representatives. Their method is reported of performing slightly better than classic *k*NN-based collaborative filtering. Similar to Xue's method, Sarwar et al. [55] applied the bisecting *k*-means algorithm to partition the user space into clusters in order to form neighborhood in the next step. Their approach proved a significant improvement in efficiency over traditional *k*NN collaborative filtering, and provides comparable recommendation quality. In literature [56] O'Connor and Herlocker performed clustering on items instead of users. They compared four algorithms: average link hierarchical agglomerative clustering, robust clustering algorithm for categorical attributes, kMetis, and hMetis using the Pearson Correlation similarity measure.

Hierarchical clustering generates a set of nested clusters organized as a hierarchical tree or dendrogram. Since any desired number of clusters can be obtained by selecting the hierarchical tree at the proper level, assuming the number of clusters beforehand is unnecessary. Hierarchical clusters can sometimes correspond to meaningful taxonomies. Traditional hierarchical algorithms merge or split one cluster at a time according to a similarity or distance matrix. Agglomerative hierarchical clustering and divisive hierarchical clustering are the two major approaches. The agglomerative hierarchical clustering starts with the points as individual clusters. Then it consecutively merges the closest pair of clusters until only one cluster is left. The divisive hierarchical clustering starts with a single all-inclusive cluster. Then it successively splits the clusters until each cluster contains only one point. Merialdo [57] proposed a hierarchical clustering algorithm to cluster users and items into two independent cluster hierarchies. In the cluster hierarchies, the author used the nodes on the path from the roots to the leaves. The recommendation is made by the calculated weighted sum of the defined centers of theses nodes.

Co-clustering is the clustering of multiple types of data. The co-clustering technique performs better to handle the sparse and high-dimensional matrices than traditional clustering [58-61]. Dhillon [62] presented a method to co-cluster words and documents based on bipartite spectral graph partitioning. Long et al. [63] introduced a general principled model, called Relation Summary Network, for co-cluster heterogeneous data presented as a k-partite graph. Hierarchical clustering deals with only one type of data and co-clustering produces just one level of data organization. Hierarchical co-clustering aims at simultaneously construction of two or more hierarchies [64, 65]. These approaches are typically preferences in biological and medical sciences [66, 67]. Co-clustering appears under the term ‘bi-clustering’ in these disciplines. Vlachos et al. [68] proposed a co-clustering algorithm based on *k*-means and agglomerative hierarchical clustering approaches. Their approach analyzed and visualized the connections between users and items for ranked product recommendations.

CHAPTER 3

COLOR BOOSTED LOCAL FEATURE EXTRACTION

In this chapter, we focus on the problem of defining the discriminative features for our target non-rigid objects. To address this problem a new formation of local feature descriptor that combines color histograms and the SIFT features is proposed. This descriptor is a core module of building a mobile visual search system, and is published in my paper [69]. In the following, Section 3.1 first describes the problems we aims to solve to extract distinctive features from product images, as well as the contributions. In Section 3.2, classic and recent local feature detectors and descriptors are discussed. Section 3.3 introduces the system framework of our mobile visual search system and discusses the design of proposed color boosted local feature descriptors. In Section 3.4, several matching experiments of different combinations of feature detectors and descriptors are conducted on the benchmark datasets. Product query experiments are also performed on an apparel dataset. Finally, we make a conclusion and propose future working areas in Section 3.5.

3.1 Research Objective and Contributions

Feature extraction plays a decisive role in visual content-based image retrieval. A good feature should properly represent the image characteristics, be repeatedly detected in images that capture the same objects/scenes while under different imaging condition, and also be distinctive so that it could distinguish it from other similar images. Besides, an ideal feature should be robust to imaging variations, such as rotation, viewpoint changes, illumination changes and occlusions. There is no universal defined feature, since different problems and different types of applications often have different characteristics. When the application domain changes, it usually requires re-designing feature detector and descriptor to capture features and achieve high performance. A feature is referred to as an interesting point/region in an image. Interesting points/regions are visually salient. Design of feature extraction method is probably the single most important factor in achieving high performance of various computer vision tasks [70].

In this chapter, we break down the problem of designing the feature descriptor for our MVS system to two problems. First, given the large number of feature extraction methods researched in the literatures, which feature extraction method is the best for our given application? Second, what characteristics of the target application should be considered and utilized to customize and optimize the generic feature extraction method so as to achieve a better performance? These questions lead us to first characterize the available feature extraction methods, so that the most promising methods could be sorted out. Then we design a novel optimized feature extraction method in respect to the characteristics of product image especially the apparel product image. For apparel products the color, texture and styles are sometimes difficult or unclear to express in words, while the images provide a good and natural source to describe these features. Difference from generic image search, there must be an interested object aligned in the center of the image in product search. The clustered background noise thus is weighted less in our application. On the other hand, the imaging position would result light, scale, and affine variation. Hence, the objective of this research is to develop a local feature descriptor that not only maintains robustness and repeatability to certain imaging condition variation, but also retains the salient color and style features of the apparel products.

The main contribution of this work is introducing a new idea of feature extraction to address issues of existing local feature extraction methods, especially for apparel product search. Proposed approach combines product color feature and local pattern feature in a way that they complement each other. We detect the keypoints by extracting the salient keypoints within the quantized and amended RGB color histograms, rather than SIFT, in which the keypoints are detected only on the gray density channel, or most other color SIFT methods, which perform SIFT computation over different color space channels separately.

Besides, the effectiveness of several promising local features on 3D non-rigid objects are explored and investigated. We configure different visual feature detectors and descriptors, and evaluate each configuration in detail. To the best of our knowledge, existing research on the comparison of visual feature detectors and descriptors are conducted for other computer vision tasks, such as visual tracking, image registration, image matching, and image retrieval, rather than non-rigid 3D object recognition. The performance of different combination of visual feature detectors and descriptors on non-rigid 3D object has not been fully understood. Another contribution of our work is filling this knowledge gap.

3.2 Local Feature Extraction for Non-rigid Object

Image local feature extraction usually consists of two stages: feature detection and feature description. A local feature commonly refers to a local pattern in an image that changes from its direct neighborhood in property or multiple properties of intensity, color, and texture simultaneously. Feature detection is algorithms that compute abstractions of image information and make local decisions at every image pixels whether there is an image feature of a given property type. The resulting features are subsets of the image domain, often in the form of isolated points, continuous curves or connected regions. Once the feature is detected, the local image patch around the feature is extracted and generated as the feature descriptor.

In this section, we discuss the visual features considered in our work. The feature detectors include Harris, FAST, SIFT, SURF, and BRISK detectors. For the descriptions, the BRISK, SIFT, and SURF feature descriptors are considered. We choose these feature detectors and descriptors for the following reasons. First, Harris detector is the best known operator around. The SIFT is the most widely used and successful detector developed in recent decade for different computer vision. The FAST, SURF, and BRISK detectors achieve a good balance between the detection performance and computation complexity. Second, the selected feature descriptors have the potential to handle the task of object recognition based on previous studies of other researchers. For instance, Chandrasekhar et al. [4], compared several feature descriptors for visual search application, and reported the SIFT feature descriptor as one of the promising one. The SIFT and SURF are concluded in Lankinen’s work [71] as the top two reliable descriptors for visual object classification. The BRISK descriptor is considered in our work because of its big advantage in computation speed.

3.2.1 Harris detector

Harris detector, proposed by Harris and Stephens [15], is developed from the auto-correlation matrix, also called the second moment matrix. Given an image *I*, an approximation to the local auto-correlation matrix of *I* is computed at every pixel (*x*, *y*):

$$M(x,y)=\begin{bmatrix} \sum_{u,v} w_{u,v} \cdot I_x^2(x_r,x_y) & \sum_{u,v} w_{u,v} \cdot I_x(x_r,x_y)I_y(x_r,x_y) \\ \sum_{u,v} w_{u,v} \cdot I_x(x_r,x_y)I_y(x_r,x_y) & \sum_{u,v} w_{u,v} \cdot I_y^2(x_r,x_y) \end{bmatrix}$$

where I_x and I_y are the partial derivative of image $I(x,y)$ with respect to *x* and *y*. $(x_r,y_r)=(x+u,y+v)$ and *w*(*u*, *v*) is the weighting function. *w*(*u*, *v*) can be a constant or a Gaussian function $\exp(\frac{-(u-x)^2-(v-y)^2}{2\sigma^2})$.

M presents the gradient distribution in a local neighborhood of an image pixel (*x*, *y*). The image pixel can be classified into three regions according to the eigenvalues λ_1 and λ_2 of M. If both λ_1 and λ_2 are small, the image pixel belongs to flat region. If λ_1 is far larger than λ_2 or vise versa, the image pixel is located in edge region. If both λ_1 and λ_2 are large and $\lambda_1 \approx \lambda_2$, the pixel is the corner in the image. In order to reduce the computation cost, Harris proposed a cornerness measure that derived from two eigenvalues:

$$c(x,y)=\lambda_1\lambda_2-\alpha(\lambda_1+\lambda_2)^2=\det(M(x,y))-\alpha\,[\mathrm{trace}(M(x,y))]^2$$

where $c(x,y)$ denotes the cornerness measure, $\det(M(x,y))$ is the determinant of $M(x,y)$, and $\mathrm{trace}(M(x,y))$ is the trace of $M(x,y)$. α is the experience constant, typically ranging from 0.04 to 0.06.

Then, non-maximum suppression is performed in a 3×3 or 5×5 neighborhood, and the local maxima of the corneress function forms the corner features of the image.

3.2.2 Features from Accelerated Segment Test

FAST is a high-speed corner detector developed by Rosten and Drummond [72]. The detection is performed on a discrete Bresenham circle around a candidate image pixel *p*. If there is a set of contiguous pixels at least nine on the circle around *p*, and they are all brighter or darker than the intensity of *p* by a pre-defined threshold *t*, then *p* is considered as a corner candidate. Besides, the algorithm is accelerated with a decision tree to reduce the number of pixels that need to be processed. Subsequently, the following score is computed at each corner candidate to remove the false candidates:

$$s(p)=\max\left(\sum_{q\in S_+}\left|I_q-I_p\right|-t,\ \sum_{q\in S_-}\left|I_q-I_p\right|-t\right)$$

where S_+ is the subset of contiguous pixels that are brighter than *p* by *t* on the circle. S_- is the subset of contiguous pixels that are darker than *p* by *t* on the circle. The corner candidates, who have an adjacent corner with a higher score, will be removed. Then, non-maximum suppression is applied to locate corner features.

3.2.3 Binary Robust Invariant Scalable Key Points

BRISK, proposed by Leutenegger et al. [73], is a binary local feature detection and description method with very high computational efficiency. The first step is to create a scale space pyramid, generally consisting of 4-layer octave images and 4-layer intra-octave images. Each octave is half-sampled from previous octave, and each intra-octave is down-sampled so that it is located between two octaves. Next, the FAST detector score s is computed at each octave and intra-octave to generate the keypoint candidates. Non-maximum suppression is then performed at each octave and intra-octave so that score s is the maximum within a 3×3 neighborhood; and score s is the largest among the scales above and below. These maxima are then interpolated using a 1D quadratic function across scale spaces and the local maximum is chosen as the scale for the feature found.

Given a set of the detected keypoints, the BRISK descriptor is constructed as a binary descriptor by simple brightness comparison tests. The brightness comparison test is performed on the samples in a pattern. This pattern is defined as N equally spaced locations on circles concentric with the keypoint.

3.2.4 Scale-Invariant Feature Transform

SIFT, introduced by Lowe [12], is a scale invariant feature detector with highly distinctive feature descriptor. In order to achieve scale invariance, a scale space pyramid of images is first built through convolutions of image I with differences of Gaussians (DoG) at different scales σ:

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

Then, each sample is compared with its 3×3 neighbors at current layer I_n, as well as the 3×3 neighbors from layers above and below (I_{n-1} and I_{n+1}) at the same octave. These local extrema are considered as keypoints. Further, the keypoint location is refined by interpolating the sample points and its direct neighbors. Keypoints with low contrast and small ratio of principal curvatures are removed. Subsequently, the gradient magnitudes and orientations of the remaining keypoints are computed. The orientations are then weighted by a Gaussian window and the gradient magnitude, and the dominant orientations are sorted out from the histogram of the weighted orientations. If multiple dominant orientations exist at a keypoint, for every dominant orientation an additional keypoint are generated.

Now, the located keypoints have been assigned with orientations and scales. A local coordinate system can be defined to compute the SIFT descriptor. A new orientation histogram is computed within a 16×16 local window and then 4×4 sub windows. For each sub window, the orientation histogram is calculated with 8 bins and weighted again by a Gaussian window and corresponding gradient magnitude. This yields the SIFT descriptor of length 128 ($4 \times 4 \times 8$).

3.2.5 Speeded-Up Robust Features

SURF, designed by Bay et al. [19], is similar to SIFT with faster feature detection and description. SURF detector is developed from the determinant of the Hessian matrix:

$$H(x,y,\sigma)=\begin{bmatrix} \frac{\partial^2}{\partial x^2}G(\sigma)\ast I(x,y) & \frac{\partial}{\partial x}\frac{\partial}{\partial y}G(\sigma)\ast I(x,y) \\ \frac{\partial}{\partial x}\frac{\partial}{\partial y}G(\sigma)\ast I(x,y) & \frac{\partial^2}{\partial y^2}G(\sigma)\ast I(x,y) \end{bmatrix}$$

It then employs box filters to approximate the second order Gaussian partial derivative for scale space analysis. The score in SURF is defined as:

$$s(x,y,\sigma)=D_{xx}(\sigma)\cdot D_{yy}(\sigma)-[0.9D_{xy}(\sigma)]^2\approx\det(H(x,y,\sigma))$$

where D_{xx}, D_{yy} and D_{xy} are the convolution of the image using box filters. Constant factor 0.9 is chosen to make the approximate solution closer to $\det(H(x,y,\sigma))$. Then, a non-maximum suppression is performed in a $3\times3\times3$ neighborhood, and the resulted maxima are interpolated across scale spaces to localize the keypoints.

Once the SURF features are localized, the SURF description is addressed in two steps: first, extracting an orientation according to the information from a circular region around the keypoints; second, defining a square region oriented along the formed orientation, and computing the SURF descriptor from the square region. Specifically, the circular region in the first step is convoluted with Haar wavelet along *x* and *y* axes. The radius of the circular neighborhood is decided by the scale, at which the keypoint is detected. So do the sampling step and wavelet response. The wavelet response is then weighted with a Gaussian, and represented as a vector with response strength along *x* and *y* axis. The dominant orientation is determined by the sum of all responses within a rotating square window. Next, this orientation window is further split up to 4×4 sub square windows, and the descriptor vector is defined as:

$$\boldsymbol{v} = \begin{bmatrix} \sum d_x & \sum d_y & \sum |d_x| & \sum |d_y| \end{bmatrix}$$

d_x and d_y denote the Haar wavelet responses in *x* and *y* directions for each sub square region. The generated descriptor vector has a length of 64 ($4 \times 4 \times 4$).

3.3 Color Boosted Local Feature Extraction

In our research a color boosted local feature extraction method is proposed. For apparel items like dresses, the most important characteristics are their color, pattern, and shape features. We break down the problem of identifying an apparel item from a query image into three main stages: (i) extract the item features from a color image, (ii) match its visual features to a large dataset of images of apparel items, and (iii) return a set of matching images with its brand and style information. We match apparel items in images by combining color feature and local feature in a complementary way.

We propose a novel process to fuse these two visual features. The flowchart of proposed mobile product search system is shown in Figure 3.1. The first is to capture the relative size of and frequency information about groups of pixels with uniform color attributes. Second the salient keypoints within the extracted color histograms are detected. Then, a local image patch around the detected feature points is computed, known as local feature descriptor. The detail image processing procedures are discussed in the remainder of this section.

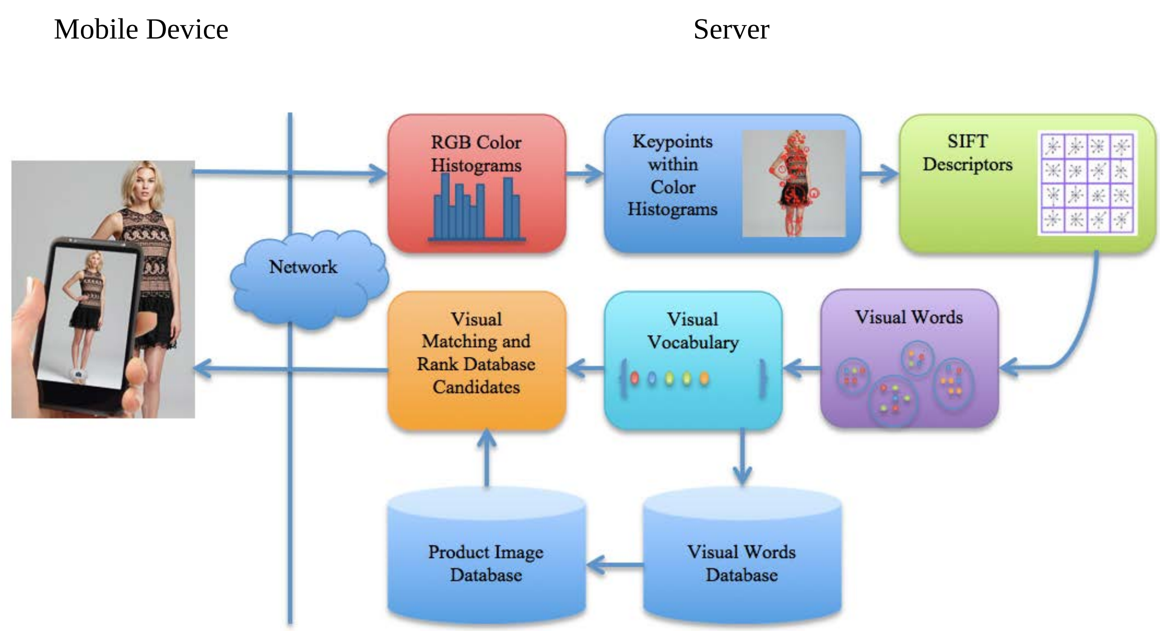


Figure 3.1. Product query flowchart of proposed mobile product system based on color boosted local features

3.3.1 Color Features

As an intuitive thought if two images have similar domain color or color distributions, they are regarded as matched in color, which is researched in papers [74] and [75]. Here we use RGB color space histograms to obtain the apparel items color information and evaluate the similarity between query image and database images. In the RGB color space, each pixel is represented by a combination of red, green, and blue intensities. To have the histogram not only retain enough color information but also robust to certain variations, the 256 RGB color scale is quantized into 21 bins. Besides, we adjust the histograms by weighted mean of the consecutive bins to diminish quantization problems.

3.3.2 Local Features

In this research, we capture the local pattern features of an apparel item based on the SIFT features. It consists of four steps:

Step 1) Extrema detection: Incremental Gaussian convolution is performed on the input color histograms to create DoG space. Next, extrema are searched in three nearby scales, and the initial locations of keypoints are obtained. DoG is a convolution of a variable-scale Gaussian function $G(x,y,\sigma)$ and input image $I(x,y)$ with regard to *x* and *y*.

$$L(x,y,\sigma) = G(x,y,\sigma)I(x,y)$$

Here $L(x,y,\sigma)$ represents the scale-space of an image, and

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

We achieve scale invariance using DoG. SIFT suggests that for detection of keypoint in certain scale, DoG can be obtained by doing subtraction of two images of nearby Gaussian scale-space in response to image $D(x,y,\sigma)$. Similar to LoG, keypoint can be located in location-space and scale-space using non-maximum suppression, as shown in the equation below.

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) \square I(x,y) = L(x,y,k\sigma) - L(x,y,\sigma)$$

where k is a constant multiplicative factor in nearby scale-spaces. In fact, DoG and its response is an approximation to LoG and $\sigma^2\nabla^2 G$, as can be seen in the following equation.

$$G(x,y,k\sigma)-G(x,y,\sigma)=(k-1)\sigma^2\nabla^2 G$$

Step 2) Accurate keypoint localization: Taylor expansion (up to the quadratic terms) of scale-space function $D(x,y,\sigma)$ is used and interpolated to obtain the location of keypoints, scale sub-pixel values. Low contrast points and edge response of instability points are eliminated.

Step 3) Orientation assignment: For each keypoint is assigned one or more orientations to achieve invariance to rotation. An orientation histogram is formed from the gradient orientations. The highest peak correspond to dominant directions is detected, and any other local peak that is within 80% of the highest peak is used as auxiliary orientations to enhance the robustness of the keypoints.

Step 4) Generation of keypoint descriptor: Coordinates of each point in 16 × 16 region around the keypoint location are transformed into 4 × 4 array weighted by Gaussian window. Then multiply the weighted opposite orientation to give 8-orientation histogram, thereby obtaining 128-dimention feature descriptor.

3.3.3 Quantization

We quantize the SIFT descriptors to get the visual words using *k*-means. The SIFT descriptors are reduced in size after quantization. Every image is represented by a certain number of visual words. The computation of query is also reduced, since only the images that have common visual words with a query image will be examined, rather than compare the high dimensional SIFT descriptors of all images. However, we should notice that quantization decreases the discriminative power of SIFT descriptors, since different descriptors would be quantized into the same visual word, and hence be treated as match to each other. Then, we build the visual vocabulary using *kd*-tree.

3.3.4 Match and Similarity

The similarity between a query image and a database image is assessed via the extracted features. We use Euclidean distance to determine two feature descriptors match or not. The similarity between the query image and the image in the database is defined as

$$Similarity = \frac{Matched\ feature\ descriptors}{Total\ feature\ descriptors\ in\ query\ image}$$

3.3.5 Retrieval Performance Evaluation

In this paper, we use the normalized recall and precision [76] to evaluate the system performance. This method takes the ranking into considerations, and hence has a more comprehensive measurement of retrieval results. Recall and precision are defined as follows.

$$R_n = 1 - \frac{\sum_{i=1}^{n} R_i - \sum_{i=1}^{n} i}{(N-n)n}$$

and

$$P_n = 1 - \frac{\sum_{i=1}^{n} \log_{10} R_i - \sum_{i=1}^{n} \log_{10} i}{\log_{10} \frac{N!}{(N-n)!n!}}$$

where R_n is recall and P_n is precision; R_i is the ranking of *i*th relevant image in the retrieval results; *n* is the total number of relevant images in database; *N* is the total number of images in the database. The precision 1 indicates the best retrieval and 0 indicates the worst retrieval.

Since every query image will have at most 2 relevant images in our database, we consider another performance evaluation method called top-*N* retrieval rates, which evaluate whether the correct dress image (front or back) is among the top *N* returned images. We calculate the average retrieval rates at top-1, top-10, and top-20 returned images.

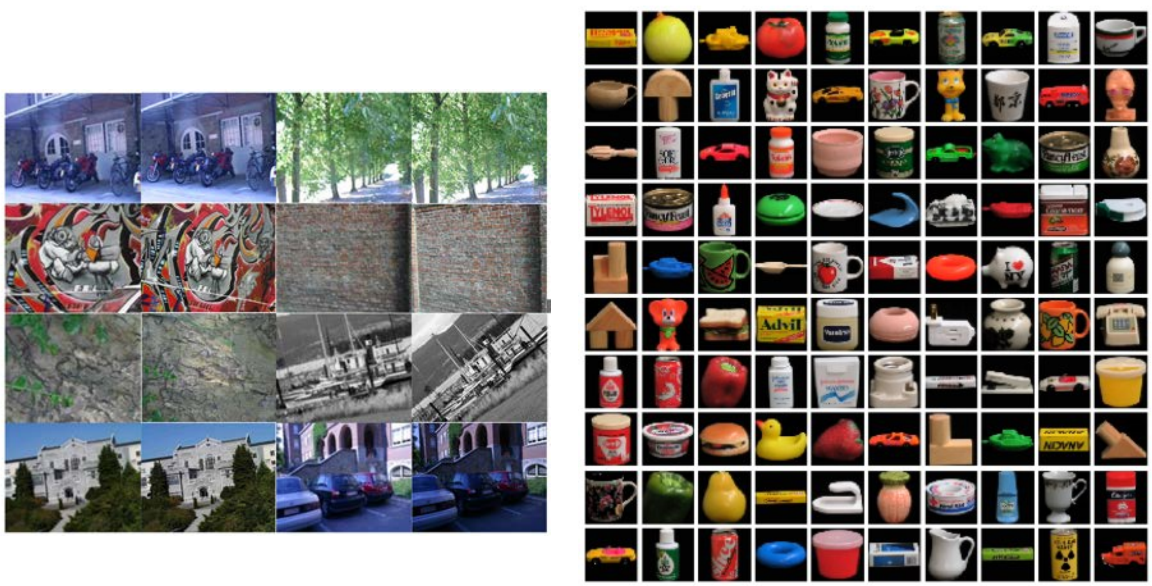
3.4 Experimental Results

In this section, we compare our method with conventional color histogram, state of the art SIFT, and color SIFT features. For a fair comparison, the features of different methods are all quantized to 65 visual words to build the visual vocabulary. All experiments are conducted in an apparel dataset crawled from an online shopping website.

3.4.1 Dataset

In order to evaluate the performance of different feature detectors and descriptors, we conducted several experiments of image matching on the benchmark dataset of Oxford Dataset [77]. We also perform experiments on the benchmark dataset of Columbia Object

Image Library - COIL 100 [78]. Figure 3.2 show typical images selected from these datasets. The Oxford dataset has been widely used for evaluating performance of local image descriptors. It contains image pairs under various image transformations, including scale, rotation, image blur, illumination, JPEG compression and viewpoint changes. The dataset also contains ground truth homographies corresponding to the image pairs. Figure 3.2 (a) shows some image pairs under different image transformations in this dataset. COIL 100 is a database of color images of objects. The objects are placed on a motorized turntable against a black background. The turntable is rotated through 360 degrees to vary object pose with respect to a fixed color camera. Images of the objects are taken at pose intervals of 5 degrees. This corresponds to 72 poses per object and the images are size normalized.



(a) (b)

Figure 3.2 Typical images selected from the datasets: (a) Oxford dataset (b) COIL 100 dataset

Current product image datasets, like Stanford mobile visual search dataset [2] only contain rigid objects like cards, paintings, books, and CD/DVDs. There is no existing benchmark dataset specifically for apparel items. Hence, we collect a list of prominent brands of women apparel from Bloomingdales.com. As of October 23, 2013 it contained 1 category, 58 brands, and 3684 images. The dataset provides the library of product images as well as product brands and styles. At least two images were acquired for each item. One is the front view on model; the other is the back view on model. Each image has a resolution of 356 × 446. The image has a gray background and a certain volume of shadow. Models in the images under similar but not exactly the same lighting conditions. The apparel item shown in the image would have occlusions and variations in viewpoint, color and shape. This dataset is practical and challenging due to the fact that it is extracted from the real online shopping department store.

3.4.2 Matching Experiment

In this experiment we implement 5 feature detectors (Harris, BRISK, FAST, SIFT and SURF) and 3 descriptors (BRISK, SIFT, and SURF) in MATLAB. All combinations are evaluated except for the SIFT-BRISK, since the SIFT detector is not compatible with BRISK descriptor. The average accuracy of image matching for every combination of feature detectors and descriptors are recorded in Table 3.1. The results show that the SIFT-SIFT provides the most accurate matching features, followed by FAST/SIFT. And with the same detector, SIFT descriptor and BRISK descriptor performs better than SURF descriptor in general, except for the case of SURF detector.

Table 3.1. Average Accuracy for Different Combinations of Feature Detectors and Descriptors

Detector	Descriptor		
	BRISK	**SIFT**	**SURF**
Harris	0.3351	0.3264	0.3018
BRISK	0.4288	0.4113	0.3907
FAST	0.4637	0.5021	0.4579
SIFT	N/A	0.5173	0.3725
SURF	0.411	0.4556	0.423

3.4.3 Ranking Experiments

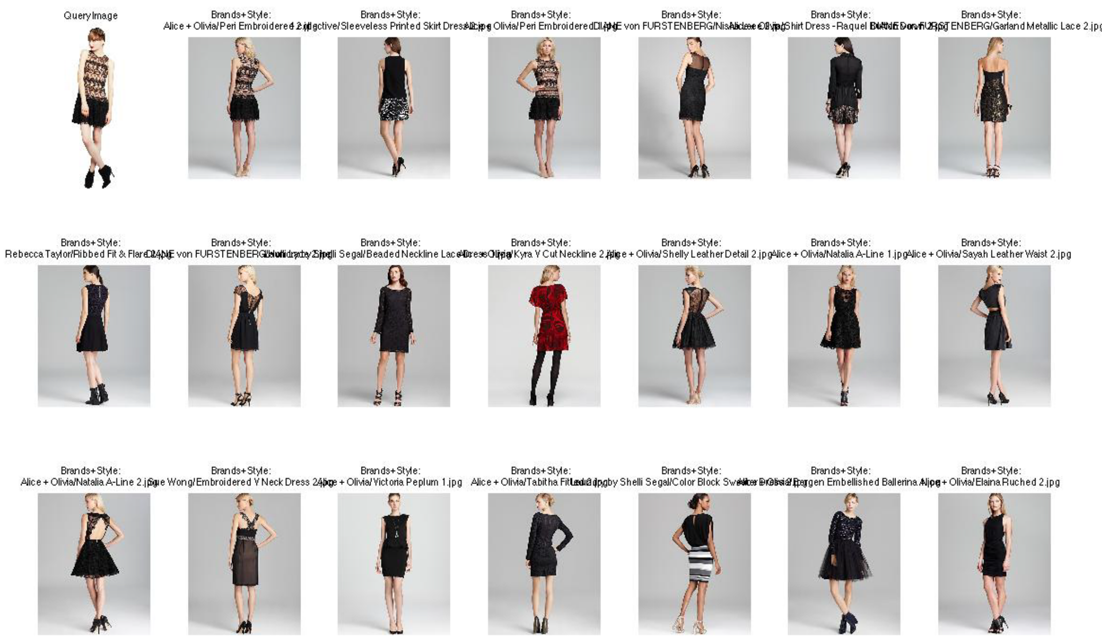


Figure 3.3. Retrieval results of the proposed method for different model image not in the database

First, we use three different query images of the same dress to test the system performance. One is the front model image in the database, another is a model image not in the database, and the third is the dress image in front view. Figure 3.3 shows an example of the retrieval results of a query model image not in the database.

Table 3.2. Image retrieval ranking in top-20 results

Query Image	**Rank of Target Product Image - Front** 				**Rank of Target Product Image - Back** 			
	Proposed Method	**SIFT**	**RGB Histogram**	**Hue-SIFT**	**Proposed Method**	**SIFT**	**RGB Histogram**	**Hue-SIFT**
Model Image in Database 	1	18	1	1	8	1	9	0
Different Model Image Not in Database 	3	14	0	0	1	10	0	0
Dress Image in front 	5	0	0	0	0	0	0	0

In the case of returning top-20 search results, the ranking of the retrieved dress images (front and back) are summarized in Table 3.2. For the model image in the database, two correct images are returns at top-10 search results with the proposed method. With RGB color histogram and original SIFT, these two images are returned within top-10 and top-20, respectively, while in top-20 Hue-SIFT can only return the front image. In practical situations a user seldom uses an exactly same image in the database as a query image. So we further test the three methods using two other typical types of image. One is a different model image not in database; the other is the dress image in front view. For the different model image not in the database, the proposed method ranks the two correct images at 1st and 3rd, and the SIFT ranks them at 10th and 14th. The RGB color histogram and Hue-SIFT cannot rank them in top-20. For the dress image in front view, the proposed method returns the correct result at the 5th image, while all other methods cannot return them in top-20 search results. Obviously, the proposed method gains better retrieval performance than that of SIFT, Hue-SIFT, and RGB color histogram. The proposed color boosted feature extraction makes the color local features complement each other, and the feature descriptors are robust to a certain variations of color, shape and perspective.

3.4.4 Recall and Precision Experiments

Table 3.3. Average precision and recall of image retrieval

Method	Proposed method	RGB color histogram	SIFT	Hue-SIFT
Recall	0.8709	0.5947	0.7500	0.7082
Precision	0.5966	0.2155	0.3065	0.3984

Table 3.4. Average retrieval accuracies of top-1 top-10 and top-20 results

Method	Proposed method	RGB color histogram	SIFT	Hue-SIFT
Top-1	0.1667	0.0000	0.0000	0.0000
Top-10	0.4583	0.0000	0.0000	0.2083
Top-20	0.5000	0.0000	0.0833	0.3750

Next, we assess our system with multiple query images. All query images are different model images that are not in the database. The recall and precision based on ranking are computed by (6) and (7), respectively. The average results are summarized in Table 3.3. The average precision of the proposed method is higher than that of other methods. The recall is also superior to other methods. For users, if a correct image is returned after 50 ranks, such result is usually non-attractive and useless. Then, we compare the retrieval accuracies of top-1, top-10, and top-20 results in Table 3.4. As we can see, the proposed method achieves 0.1667 of the correct retrieval at top-1 result, while other methods return none. At top-10, SIFT and RGB color histogram still cannot return any correct results. The proposed method performs 0.4583 of correct retrieval, better than 0.2083 of Hue-SIFT. At top-20, the proposed method gains 0.5000 outperforming Hue-SIFT's 0.3750 and SIFT's 0.0833. For all retrieval rates RGB color histogram can hardly have correct returns within top-20 results.

3.5 Concluding Remarks

In this chapter, we first evaluate the effectiveness of the possible combinations of different visual feature detectors and descriptors for 3D non-rigid object. Several common visual feature detectors (Harris, BRISK, FAST, SIFT, and SURF) and descriptors (BRISK, SIFT, and SURF) have been selected and evaluated. The primary difference between this work and the comparison studies of other researchers lies in the different targeted applications. Chandrasekhar et al. [4] compared the effectiveness of different visual feature detectors and descriptors for mobile visual search of rigid product like books and CDs. The comparison in [71] focused on the application of visual object categorization. Neither of these comparisons targeted the effectiveness of 3D object recognition, the focus of this paper. Considering that different applications pose different challenges and requirements for visual feature detectors and descriptors, it is unclear whether the conclusions and findings in [4] and [71] are accurate for 3D object recognition. The evaluation results indicated that the SIFT achieved the best overall performance in describing image local features. This finding helps when we design the feature descriptor for the MVS system of apparel items. It could also benefit reshaping existing or ongoing other visual feature based research work in visual search. Future work will focus on the use of these findings to tune and extract visual features so as to improve recognition accuracy and adapt to different applications.

In this chapter, we have also provided a scheme for mobile product search based on the color feature and local pattern features of apparel items. The main contribution of this work is introducing a new idea of feature extraction to address issues of existing local feature extraction methods, especially for apparel product search. We detect the keypoints by extracting the salient keypoints within the quantized and amended RGB color histograms, rather than SIFT, in which the keypoints are detected only on the gray density channel, or most other color SIFT methods, which perform SIFT computation over different color space channels separately. The experiment results indicate that our proposed method retains the salient color and local pattern of the apparel products while maintains its robustness and repeatability to certain imaging condition variation. It outperforms RGB color histogram, original SIFT, and Hue-SIFT.

Through observation there are several false retrievals in our system. This is mainly caused by large portion of occlusion, over complex background, great imaging condition changes like perspective and lighting, etc. Therefore, our future work includes: (i) exploring research in areas of cloth texture features, object global outline shape features and segmentation from clustering background, as well as feature indexing to further improve retrieval performance, (ii) expanding our dataset to cover more apparel categories such as tops, tees, shorts, skirts, pants, shoes, and handbags, and (iii) extending our method to mobile video search in future work.

CHAPTER 4

UNIFIED HYPERGRPAH LEARNING BASED RANKING SCHEME

In this chapter, we introduce the ranking model by understanding the complex relations within product visual and textual information in visual search systems. To understand their complex relations, we focus on using graph-based paradigms to model the relations among product images, product category labels, and product names and descriptions. The proposed unified probabilistic hypergraph ranking approach has been published in my paper [79]. In the following, Section 4.1 first describes the importance of ranking in a search engine, our research objective and major contributions. Section 4.2 discusses the design of the proposed unified probabilistic hypergraph ranking algorithm. In Section 4.3, several retrieval experiments are conducted on an apparel data set and compared with conventional CBIR ranking methods. Finally, we conclude with the proposed ranking scheme and discuss future works in Section 4.4. For the reader's convenience, the symbols used in this chapter are listed in Table 4.1.

4.1 Research Objective and Contributions

Ranking plays an essential role in a product search system. Given a query, candidate products should be ranked according to their distance to the query. The effectiveness of the product search system is evaluated by its ranked search results, e.g. in the form of precision or recall. In addition, the efficiency of a system is evaluated by its running time of a query. The best scenario is that the system returns a series of relevant products at the top of retrieved results. However, in certain cases, even if a system finds the particular

Table 4.1 Notation and definition

Notation	Definition
$G=(V,E,w)$	G indicates a hypergraph.
$V=\{v_1,v_2,\cdots,v_n\}$	V indicates the set of vertices of hypergraph, v_i is the *i*-th vertex, and n is the number of vertices.
$E=\{e_1,e_2,\cdots,e_m\}$	E indicates the set of hyperedges, is the *i*-th hyperedge that connect a finite set of vertices, and m is the number of hyperedges with non-empty set of vertices.
$w=[w(e_1),w(e_2),\cdots,w(e_m)]$	w indicates the $m\times 1$ weight matrix of the hyperedges, and $w(e_i)$ is the weight of hyperedge e_i.
$d(v_i)$	Degree of a vertex v_i.
$d(e_i)$	Degree of a hyperedge e_i.
D_v	$n\times n$ diagonal matrix containing vertex degrees.
D_e	$m\times m$ diagonal matrix containing hyperedge degrees.
W	$m\times m$ diagonal matrix containing weights of hyperedges
H	$n\times m$ vertex-hyperedge incidence matrix.
$X=\{x_1,x_2,\cdots,x_n\}$	X indicates the product image pool, x_i denotes the *i*-th image, and n is the number of images in dataset.
F	Set of visual feature words of product images.
S	Set of product style labels.
N	Set of product name and description.
E^{FSN}	Set of feature-style-name hyperedges.
E^{FS}	Set of feature-style hyperedges.
E^{FN}	Set of feature-name hyperedges.
E^{F}	Set of visual feature hyperedges.
E^{S}	Set of style hyperedges.
E^{N}	Set of name hyperedges.
E^{kNN}	Set of *k*-nearest neighbor hyperedges.
y	$n\times 1$ query vector. The elements of pre-ranked relevant results are set to 1, and the others are set to 0.
f	$n\times 1$ ranking score vector.

relevant product, it is still considered as ineffective for the reason that the retrieved product does not present in the top list but is buried in a number of irrelevant results. In order to compensate for this rank inversion issue, the automated learning techniques and the skills of users are utilized to improve the representation of the query product.

A natural extension of such add-value process is to request users to label the returned results as relevant or irrelevant, which is called relevance feedback (RF). However, in reality users are not willing to initial a query by labeling retrieval metadata and samples, nor to give feedback of the retrieved results, since these methods makes the retrieval procedure inconvenient. Therefore, the insufficient user-labeled images undermine the prospect of supervised learning methods in the CBIR field. A promising and relatively unexplored research direction is to exploit transductive or semi-supervised learning, among which graph-based methods [48, 80-82] have demonstrated their effectiveness in image retrieval and therefore received increasing attention. In the graph-based methods, a graph is built on the image data set and each image is considered as a vertex in the graph. An edge and its weight are defined between two images according to a certain relationship definition. For example, the edge can be defined as the images visual similarity. The weight is formulated by the visual distance between any two image vertices. Then, the ranking can be formulated as a random walk on the graph [80, 83], or an optimization problem [81]. However, these graphs created in pairs, cannot sufficiently show the relations among images. Hypergraph is introduced to the CBIR field. Hypergraph is a generalization of a simple graph. In a hypergraph, an edge, called hyperedge, can connect any number of vertices; it is a non-empty set of vertices. Recent research [48, 82] proves the effectiveness of hypergraph learning in solving ranking problems. Motivated by their work and [46], we propose a novel hypergraph based transductive algorithm for product retrieval ranking. We use a unified probabilistic hypergraph to model multiple features of the products and explore the implicit relations among various visual and textual features.

We summarize the contributions in the following three aspects. First, hypergraph is used to represent a commercial product image dataset. We explore the relation between visual and text features of these images. Second, a new product retrieval framework for the product search is designed. Third, we create a novel strategy of starting a query. We establish relations between visual features and textual features; embrace them into a specific unified probabilistic hypergraph. For problems that lack user labeled query keywords, we solve them using transductive inference on the hypergraph.

4.2 Ranking on Unified Hypergraph

In this research, we employ a unified hypergraph to represent the relations of commercial product images, its textual descriptions, and its categorization labels. We propose a model for searching and ranking images based on hypergraph learning. Conventional visual search systems sort and search images based on the similarity of their visual content. The idea of this model is to learn the relevance of different product features: images visual feature, textual feature, and the hybrid visual-textual feature, and then combine them with the results of visual similarity based retrieval.

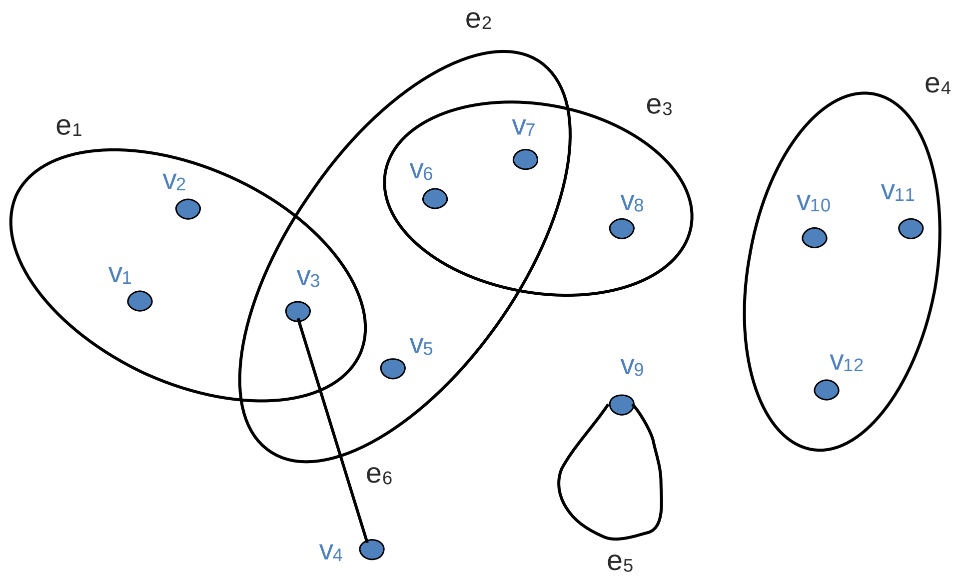


Figure 4.1 An example of a hypergraph

4.2.1 Problem Definition

Let *V* represents a finite set of vertices. *E* represents a family of hyperedges on *V*, and each hyperedge $e \in E$ contains a list of vertices that belong to *V*. The hypergraph can be denoted as $G = (V, E, w)$ with a weight function *w*. The degree of a hyperedge *e* is defined by $d(e) = |e|$, i.e., the number of vertices in *e*. The degree of a vertex *v* is defined by $d(v) = \sum_{e \in E} w(e)$, where $w(e)$ is the weight of the hyperedge *e*. The hypergraph can be formulated to a vertex-hyperedge incidence matrix $H \in R^{|V| \times |E|}$, where each entry *h*(*v*, *e*) is defined as:

$$h(v,e) = \begin{cases} 1 & \text{if } v \in e \\ 0 & \text{otherwise} \end{cases}$$

Then we have $d(v)=\sum_{e\in E} w(e)h(v,e)$, and $d(e)=\sum_{v\in V} h(v,e)$. Let D_v and D_e denote the diagonal matrices containing the vertex and hyperedge degrees respectively, and W be a $|E|\times|E|$ diagonal matrix containing the weights of hyperedges.

Consider a simple example of hypergraph $G = (V, E)$, built as shown in Figure 4.1. $V=\{v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8,v_9,v_{10},v_{11},v_{12}\}$ and $E=\{e_1,e_2,e_3,e_4,e_5,e_6\}$. The incidence matrix H is defined as:

$$H = \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_3 & 1 & 1 & 0 & 0 & 0 & 1 \\ v_4 & 0 & 0 & 0 & 0 & 0 & 1 \\ v_5 & 0 & 1 & 0 & 0 & 0 & 0 \\ v_6 & 0 & 1 & 1 & 0 & 0 & 0 \\ v_7 & 0 & 1 & 1 & 0 & 0 & 0 \\ v_8 & 0 & 0 & 1 & 0 & 0 & 0 \\ v_9 & 0 & 0 & 0 & 0 & 1 & 0 \\ v_{10} & 0 & 0 & 0 & 1 & 0 & 0 \\ v_{11} & 0 & 0 & 0 & 1 & 0 & 0 \\ v_{12} & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

The problem of ranking on the hypergraph is formulated as: given a query vector y, a subset of vertices in the hypergraph $G=(V,E,w)$, a ranking score vector f is produced according to the relevance among vertices in the hypergraph and the query. We define the cost function of f as follows [48]:

$$\Omega(f)=\frac{1}{2}\sum_{i=1,j=1}^{|V|}\sum_{e\in E}\frac{1}{d(e)}\sum_{\{v_i,v_j\}\subseteq e}w(e)\left\|\frac{f_i}{\sqrt{d(v_i)}}-\frac{f_j}{\sqrt{d(v_j)}}\right\|^2+\mu\sum_{i=1}^{|V|}\left\|f_i-y_i\right\|^2$$

where $\mu > 0$ is the regulation factor. The first term, known as the normalized hypergraph Laplacian, is a constraint that vertices sharing many incidental hyperedges are supposed to obtain similar ranking scores. The second term is a constraint of the variation between the final ranking score and the initial score.

In order to obtain the optimal solution of the ranking problem we seek to minimize the cost function:

$$f^* = \arg\min \Omega(f)$$

With the derivations in [48], we can rewrite the cost function as

$$\Omega(f) = f^T(I-\Theta)f + \mu(f-y)^T(f-y)$$

where $\Theta = \mathrm{D}_\mathrm{v}^{-\frac{1}{2}}\mathrm{HWD}_\mathrm{e}^{-1}\mathrm{H}^\mathrm{T}\mathrm{D}_\mathrm{v}^{-\frac{1}{2}}$. Then the optimal f^* can be obtained by differentiating $\Omega(f)$ with respect to *f*:

$$\left.\frac{\partial\Omega}{\partial f}\right|_{f=f^*} = (I-\Theta)f^* + \mu(f-y)^* = 0$$

$$f^* = (I-\frac{1}{1+\mu}\Theta)^{-1}y\,.$$

4.2.2 Unified Probabilistic Hypergraph Ranking Model

In the following we will explain our improved hypergraph formulation for the product retrieval and ranking. In a typical online shopping system there are three different types of information representing a product. They are product image, product name and description, and product labels, which are discussed in detail in Section 4.3. With these three types of information we design 6 types of hyperedges. Each image in the product image dataset is considered as a vertex in the unified hypergraph. Let X denote the product image pool, and $x_i \in X$ is a particular product image. Let F denote the visual feature description of the images, or say, visual words, S denote the set of product style, and N be the name and description of the product. The unified hypergraph G that contains 6 different types of hyperedge could represent the following implicit relations:

(1) E^{FSN} (the set of images feature-style-name hyperedges): the product, which share the same product name, product style, and visual feature word;

(2) E^{FS} (the set of images feature-style hyperedges): the product, which belongs to a certain product style, contains same visual feature word;

(3) E^{FN} (the set of images feature-name hyperedges): the product, containing the same visual feature word, share a common keyword in name;

(4) E^{F} (the set of images visual feature hyperedges): the product images might contain the same visual feature word;

(5) E^{S} (the set of images style hyperedges): the product belong to the same product style;

(6) E^N (the set of images name hyperedges): the product has similar keywords in its name and description.

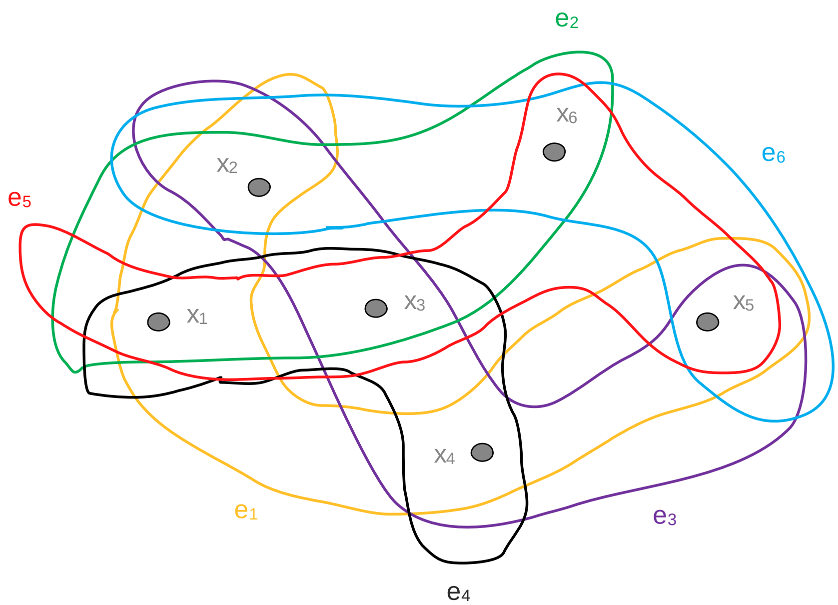


Figure 4.2. An example of probabilistic hypergraph

Typically we assign 1 to the weights of these hyperedges. Rather than traditional hypergraph structure, in which an image vertex x_i is assigned to a hyperedge e_j in a binary way, i.e., $h(x, e)$ is either 1 or 0, we propose a probabilistic hypergraph to describe the relation between vertex and hyperedge. For hyperedge E^F, each image vertex is treated as a centroid, and the hyperedge is formed by the centroid image and its k-nearest neighbors. The incidence matrix H of the probabilistic hypergraph is defined as follows:

$$h(x_i, e_j) = \begin{cases} Sim(x_j, x_i) & \text{if } x_i \in e_j \\ 0 & \text{otherwise} \end{cases}$$

where x_j is the centroid of e_j. In the proposed formulation, a vertex x_i is softly assigned to a hyperedge e_j based on the similarity between x_j and x_i, which overcomes the limitation of truncation loss with the binary assignment. Besides, we use a parameter to set desired similarity. Figure 4.2 demonstrates an example of constructing such hyperedges. Each vertex and its top 3 similar neighbors form a hyperedge. We constrain that only the vertices with a pair similarity larger than 0.4 would be connected into a hyperedge.

The incidence matrix H of the proposed probabilistic hypergraph is

$$H = \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline x_1 & 1 & 0.81 & 0 & 0.63 & 0.45 & 0 \\ x_2 & 0.81 & 1 & 0.43 & 0.52 & 0 & 0 \\ x_3 & 0 & 0.43 & 1 & 0.83 & 0.62 & 0 \\ x_4 & 0.63 & 0.52 & 0.83 & 1 & 0 & 0 \\ x_5 & 0.45 & 0 & 0.62 & 0 & 1 & 0.8 \\ x_6 & 0 & 0 & 0 & 0 & 0.8 & 1 \end{array}$$

With the hyperedges as designed above we can form the 6 types unified weight matrix W, and have the vertex-hyperedge incidence matrix H. The size of both matrices depends on the cardinality of product image data set involved, and they are all sparse matrices. As a result, the computation of the proposed hypergraph ranking algortihm is fast. It is implemented in two stages: offline training and online ranking. In the offline training stage, we construct the unified hypergraph with matrices H and W derived from above. Then based on the matrices, we calculate the vertex degree matrix D_v and the hyperedge degree matrix D_e. Finally $(I-\frac{1}{1+\mu}\Theta)^{-1}$ can be computed, where $\Theta = D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$. Note that $I-\frac{1}{1+\mu}\Theta$ is invertible, since the hyperedge E^{kNN} ensures that H is full rank. Then the online ranking procedure can be described as: firstly build the query vector y, and secondly compute the ranking score vector f^*. The elements of the pre-ranked relevant images are set to 1, and the others are 0.

4.3 Experimental Results

In the experiment, we build the unified images hypergraph using different combinations of hyperedges to test the effect of different factors on the ranking performance. We then investigate the performance of different hypergraps. The superiority of the transductive inference is demonstrated in handling the queries that lack user labels. We use the visual similarity based ranking as a baseline. We compare the different hypergraph based ranking models with the visual similarity ranking. Also we use the visual similarity ranking score to deduce the pre-ranked score in hypergraph ranking.

Product Image	Product Name and Short Description	Product Labels
	• BCBGMAXAZRIA Dress - Leyla Sleeveless Keyhole Lace Fit and Flare • Round neck, sleeveless, keyhole front, contrast lace bodice, contrast waistband, drop waist	Occasion: Cocktail Type: Fit and Flare Type: Lace Length: Short Sleeve Length: Sleeveless

Figure 4.3. A typical product representation in system

For an online shopping system, a product is represented by three types of information, as shown in Figure 4.3: (1) images, which demonstrate the product visually. This usually has several photos taken from different viewpoint; (2) name, which is the name of the product or give a brief description of the product; (3) labels, which is the textual tags that classify the product into different categories according to the sorting rules. For example, for apparel products, we could have different categories like style, length, sleeve length, occasions, etc.

The product image data set used in the experiment is obtained from a list of prominent brands of women apparel. It contains 3 product categories, 58 brands, and 4210 images. We use different dress categories such as type, length and sleeve length to form the set of product style, which contains 7 types, 3 lengths and 6 sleeve lengths. The product name is the product brand, its style name and a short description. Here we generate a bag of words to represent it. For visual features, we first extract a color boosted SIFT feature [69], which captures the product color feature and its local patterns, and then quantize the visual feature descriptors into 65 visual words. For parameter k and μ, we follow the setting in literature [48], where they are empirically set to 100 and 0.001. The Normalized Discounted Cumulative Gain (NDCG) [84] is employed to evaluate the ranking performance. NDCG at position k is defined as

$$NDCG@k = \frac{\sum_{i-1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}}{IDCG}$$

In our research, an experiment participant is asked to judge the relevance of each retrieval result to the query. Each returned image is to be judged on a scale of 0 - 3 with *rel* = 0 meaning irrelevant, *rel* = 3 meaning completely relevant, and *rel* = 1 and *rel* = 2 meaning "somewhere in between".

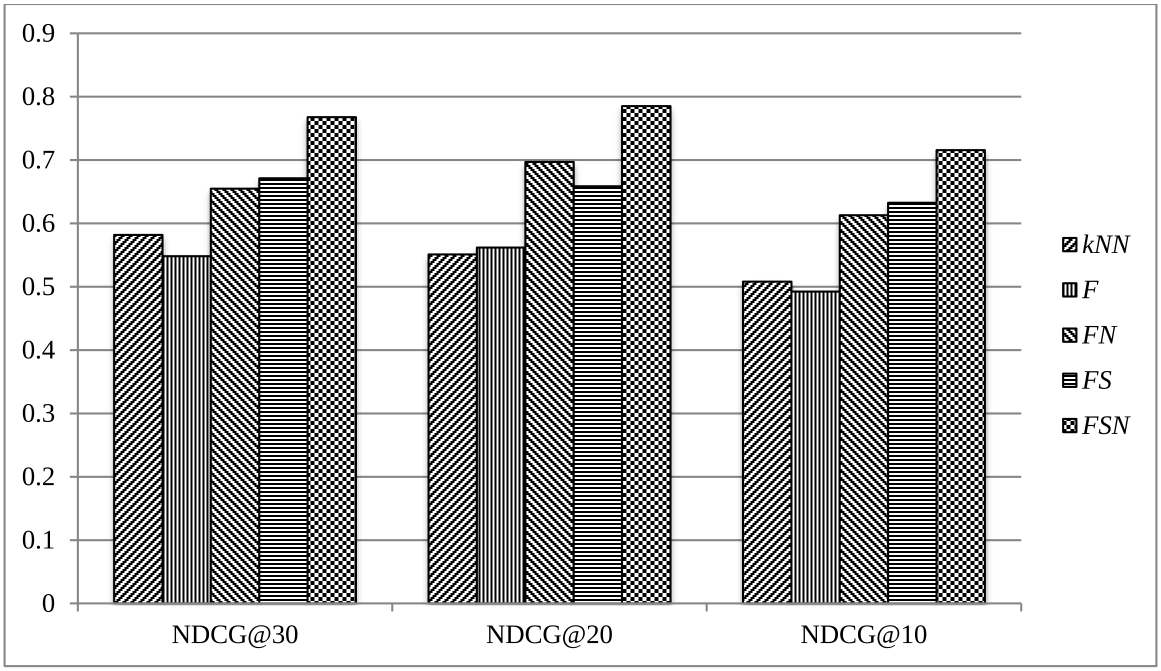


Figure 4.4. Performance comparison of different hypergraph constructions

In our proposed method, we integrate 7 different relations and hyperedges into constructing of product hypergraph so that it effectively represent of the product image data set. The hypergraph also encloses multiple correlations among different visual words and text features. To evaluate the effectiveness of such a representation in product search, we consider different hypergraph constructiosn with different hyperedge integration. Figure 4.4 illustrates the ranking performance in terms of average NDCG at different depths of 10, 20 and 30. It is evident that the hybrid hypergraph (*FSN*, *FN*, and *FS*) outperforms the simple construction of hypergraph (*F*) and the visual similarity based ranking (*kNN*). And the proposed unified hybrid hypergraph *FSN* achieves the best performance. The reason for this is quite straightforward: high-order correlations among product visual features and its textual labels are well captured in our unified hypergraph model. The representation and description of a product is extensively enhanced in database.

Figure 4.5. Top 10 retrieval results with different ranking methods. (a) Visual similarity based ranking. (b) Proposed unified probabilistic hypergraph learning ranking

In Figure 4.5 an example of query is demonstrated, in which the system cannot find the best match at the top 10. With the similarity ranking, a black tuxedo jumpsuit is recognized as dress, pants, and coats. While with the proposed unified probabilistic hypergraph learning ranking, the system provides a series of products with similar styles, which is meaningful for the online shoppers. The reason is that we not only capture the visual feature and textual feature separately, but also model the correlations between them. In this way, an improved search results are produced.

4.4 Concluding Remarks

In this chapter, we address the problem of ranking in product search by image. We focus on integration of various types of product textual information and visual image. We introduce a hypergraph learning approach to the visual product search and propose a more comprehensive and robust ranking model. In this way the supervised classification and unsupervised visual search are well balanced. Specifically, we construct the hypergraph by combining three types of product information that embed the relevance among textual features and visual images. Experimental results show that the proposed hypergraph learning framework is a promising ranking scheme for product search. In future work we will consider exploring the adaptive feature weight and other hypergraph learning operators.

CHAPTER 5

FUZZY HIERARCHICAL CLUSTERING FOR RECOMMENDATION

Recommending similar products is an important part of an online shopping website. A good product recommendation can save search time and delight the users. Its impact on promoting sales is also considerable and ever rising. This chapter introduces a novel soft hierarchical clustering algorithm, and applies this algorithm for collaborative filtering recommendation. In the following, Section 5.1 first describes how a recommendation would benefit in e-commerce, our objective and contributions. Then, Section 5.2 discusses the design of the proposed Fuzzy Hierarchical Co-Clustering (FHCC) algorithm. Section 5.3 presents recommendation performance of the proposed algorithm on a benchmark rating dataset. Finally, we conclude the chapter and discuss future works in Section 5.4.

5.1 Research Objective and Contributions

Recommendation service is gaining increasing attention in the big data era and has brought great benefit in e-commerce. Amazon, as the largest and most influential e-commerce company in the world, has successfully applied the recommendation in their online retail business. The recommendation system helps lower transaction costs and improves revenues in different ways, such as promoting cross-selling and upselling, and reducing labor cost of customer assistance by providing this self-service tools. Besides, it is capable of analyzing the ongoing consumer panel data, support marketing research and support product management. On the other hand, the recommendation system also benefits the consumers in a way that it facilitates the process of product search and discovery. It mitigates, if not overcomes, the problem of information overload by aiding customers in search and exploring new, relevant and interesting items (e.g., media, product, or service) and helps them identify which items are worth viewing in detail. One of the major features for a recommendation system is its serendipity, i.e., to help the users make fortunate discoveries that they were not explicitly looking for, as well as make personalized recommendations.

Recommender systems identify recommendations for a user based on recommending services' content, users' previous purchases, searches and other behaviors. Most recommendation systems take either of two basic approaches: content-based filtering or collaborative filtering. Content-based filtering is derived from information retrieval with a specific focus on long-term information filtering [85]. It recommends products which are the most similar to that the user interests in, with regard to the inner attributes (e.g. audio feature [86], image feature [87] and textual information [88]) of the products. Collaborative filtering is a sort of word-of-mouth advertisement. It assumes that if user A and user B have similar tastes on a specific item, such as similar behaviors of purchasing, rating, or watching the same item, they may act similarly on other items. Then, the user-item ratings data will be used to make predictions and recommendations. The ratings by users are explicit indications on a certain scale, normally 1 to 5, while purchases and click-throughs are implicit indications [89, 90]. Collaborative filtering highly relies on user historical usage data, and hence suffers from the new user problem and the new item problem. The new user problem represents the lacking of accuracy of the recommendations received by a new user when a significant number of votes are no yet fed to the recommendation system's collaborative filtering core. The new item problem is that the first user who rates the new item cannot benefit from the rating. The problem causes new items to be ignored until a substantial number of ratings are received for the same item [91]. Content-based filtering does not incorporate the similarity of preference across individuals. Additionally, hybrid approaches, such as the content-boosted content filtering algorithm and personality diagnosis, combine content-based filtering and collaborative filtering. Therefore the limitations of either approach are avoided, and the performance of recommendation is also improved [92].

In this paper, a novel method called Fuzzy Hierarchical Co-Clustering (FHCC) is developed to build hierarchical co-clusters of users and products, and then perform recommendation based on the similarity between the information of a new user and the properties of the clustered user-product groups. Inspired by approaches in [93], we design FHCC as follows: first the algorithm begins from singleton clusters. In the next steps, every two nearest clusters are repeatedly merged into one until there is only one cluster remaining. In our case, FHCC can merge a subset of the users with a subset of the products based on their internal similarity measurement in each step of the merging process. In addition, since the users and products could be related to different clusters in practice, we extend the hard HCC with fuzzy set theory to support soft clustering, in which each object can belong to one or more clusters based on the similarity measurement. We formalize each user-product instance as a three-dimensional virtual vector. To calculate the similarities between different instances, we evaluate the similarity score for each component in the virtual vector, and finally combine them into a hybrid similarity measure.

5.2 Recommendation via Soft Hierarchical Clustering

5.2.1 Problem Formulation

Generally speaking, the rating data from users contains three types of resources: users, products, and products' ratings reviewed by users. An example of rating data is depicted in Table 5.1.

Table 5.1. An example of rating data

User \ Product	p_1	p_2	p_3	p_4	p_5
u_1	5		5		4
u_2		4		3	
u_3			4		4
u_4	3				1
u_5		4	5	4	

With such data structure, we can formalize the recommendation problem as follows: assume we are given a set of $\boldsymbol{m}$ users $U = \{u_1, u_2, \cdots, u_m\}$, and a set of $\boldsymbol{n}$ products $P = \{p_1, p_2, \cdots, p_n\}$. We are also given $\boldsymbol{m} \times \boldsymbol{n}$ rating matrix $R = (r_{ij}) \in \mathbb{S}^{m \times n}$, where $\boldsymbol{r_{ij}}$ is the rating score that the *i*th user in U assigned to the *j*th product in *P*. Our objective is to design a recommendation model that could perform the following functions:

1) Simultaneously we generate a hierarchical clustering of U and of P based on matrix R. Each cluster contains a subset of these two data sources, and can be regarded as a potential community group underlying the users rating data.
2) When a new user joins, along with his/her profile $\boldsymbol{u'}$ and his/her interested product $\boldsymbol{p'}$, the system should be capable of recommending a series of products. The recommended products are based on the similarities between $\boldsymbol{u'}$ and U, $\boldsymbol{p'}$ and P.

We design and propose the following method to solve the problem and achieve the goals mentioned above:

1) We propose a soft agglomerative clustering algorithm – Fuzzy Hierarchical Co-clustering to recover the hidden co-clusters by recognizing the bipartite user-product rating matrix;
2) Based on the recovered co-clusters, hybrid similarity measurement is used to obtain the average similarities between the new user, his/her interested product and the co-clusters. Then we extract the potential recommendations from the most similar co-cluster.
3) We rank these recommendation candidates from high to low according to the product rating predications.

5.2.2 Fuzzy Hierarchical Co-Clustering

Traditional hierarchical co-clustering [68, 93] is a hard clustering technique, in which each item can only be assigned into one co-cluster. In the real world, people would be interested in various different categories of products, which can be revealed via their behaviors like search, click-on, purchase and rate. Therefore, users should be grouped to more than one interest community. It is a non-trivial task to decide which group a user belongs to, and it is not reasonable to assign a user showing interests in different products to only one group. To address this issue, a possible solution is to utilize soft clustering techniques on users rating data so that after clustering, a user might belong to multiple groups. In this study, we propose a novel hierarchical co-clustering algorithm with fuzzy set theory to support the soft clustering, in which each entity belongs to multiple groups, for our recommendation purpose. Membership levels that indicate the strength of the association between the data element and the cluster are assigned to each entity. Specifically, we first represent the users rating data as three-dimension virtual vectors, and then perform hierarchical clustering on these vectors by virtue of a hybrid similarity measurement. The basic steps of the FHCC are designed as follows:

Step 1: The initial co-cluster is formed by assigning each user to his/her rated product. If there are *N* rating scores, then there are *N* co-clusters, each containing just one user and one product. Let the similarities between the co-clusters defined as a hybrid similarity of three components that a co-cluster entity contains.

Step 2: The pair of co-clusters with highest similarity is discovered and merged into a single co-cluster. Therefore the number of co-clusters is reduced by one.

Step 3: The similarities between the new co-cluster and each of the old co-clusters are computed. In our study, we use single-link clustering, also known as connectedness method, to represent the new similarities. The new similarity between two clusters is equal to the greatest similarity of any two individual members from each of the two clusters.

Step 4: Step 2 and Step 3 are repeated until initial co-clusters are entirely clustered into a single co-cluster. It requires *N−1* rounds of iterations.

The algorithm is described in Algorithm 1.

Algorithm 1 The FHCC Algorithm Description.

Input: m users set *U*, n products set *P*, and rating matrix *R*.

Output: Dendrogram of clustering results.

Initialization:

Create an empty hierarchy *H*

for $i = 1$ to m **do**

  for $j = 1$ to n **do**

    if $r_{ij} > 0$**, then**

      Create node $e = \{u_i, p_j\}$

    end if

  end for

end for

$List \leftarrow \{e \in E\}$

$N \leftarrow$ sizeof $[E]$

Add *List* to *H* as the bottom layer

Hybrid similarity:

for $i = 1$ to N **do**

  for $j = 1$ to N **do**

$$Sim(e_i, e_j) = w_1 \cdot Sim^U + w_2 \cdot Sim^P + w_3 \cdot Sim^R$$

---

end for

end for

Sort E

Clustering:

for $i = 1$ to N **do**

Choose the pair of nodes that of highest similarity

Merge them into a new node $e = Merge(u, p)$

Remove $\{u, p\}$ from *List* and add *e* to *List*

Add *List* to *H* as the next layer

end for

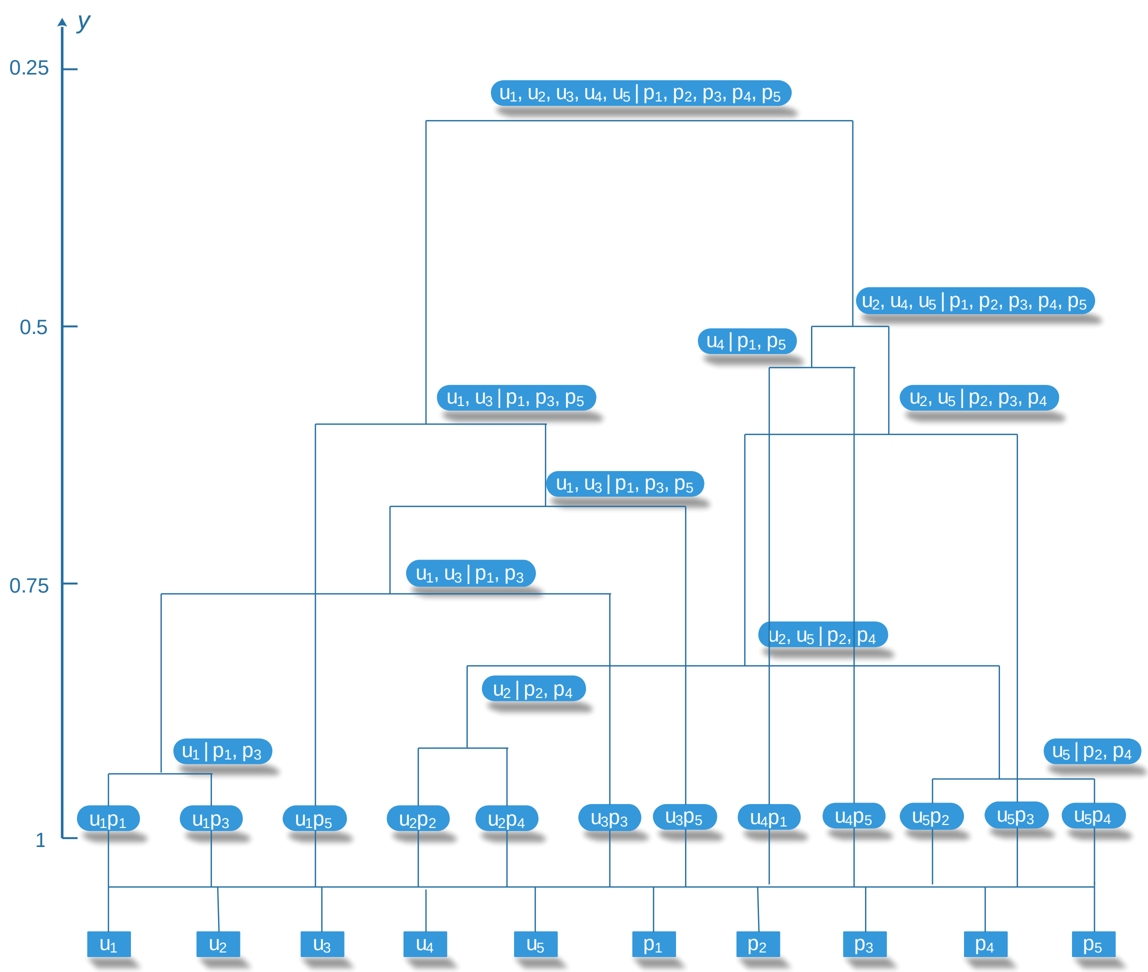


Figure 5.1. A sample of FHCC clustering dendrogram

A sample of FHCC clustering is visualized as a dendrogram in Figure 5.1. The horizontal lines represent the merging of clusters. The *y*-coordinate represents the similarity of the two co-clusters that are merged. In general, there are various types of measures to evaluate the similarity of two clusters, such as Euclidean Distance, Cosine Distance, Jaccard Similarity and so on. However, none of them can evaluate the pairwise similarity of multi-source entities. To address this issue, we propose a hybrid similarity measurement, consisting of three different similarity computations for the corresponding components. Given two three-dimensional vectors $e_i(u,p,r)$ and $e_j(u',p',r')$, the similarity between them can be calculated as

$$Sim(e_i,e_j) = w_1 \cdot Sim^U + w_2 \cdot Sim^P + w_3 \cdot Sim^R$$

Here, the weights w_1, w_2, and w_3 are the three control parameters, indicating how much we trust the corresponding components. In general they can be tuned appropriately to different applications. Sim^U represents the similarity between two user sets, Sim^P denotes the similarity between two product sets, and Sim^R evaluates how similar the two user sets rates the corresponding product sets.

Sim^U: To evaluate the similarity between two user sets, we need to analyze what types of information in the users' profiles we can utilize. In general, the user profile may provide a couple of optional contents, such as self-definition, interests, job titles and so on. With such information, we treat the similarity computation as a multi-attribute comparison problem. For each attribute, we analyze the overlapping between two values and get a similarity score restricted in [0, 1]. The similarity between two different user profiles can be obtained by assigning different weights on different attribute comparisons, and sum them up. If the user sets contain multiple users, Sim^U is finally calculated as the average similarity among all the user pairs.

Sim^P : The similarity between two product sets also relies on what product information is applicable. In general, a product could contain information of product name, brand, category, description and so on. Similar to the Sim^U, similarity of product is computed as a multi-attribute overlapping measure. Also, if there are multiple products in the product sets, the average similarity score is regarded as the value of Sim^P.

Sim^R: To evaluate how similar rate sets are, we simply use the normalized difference of two rating scores.

Each similarity mentioned above is restricted in the range [0, 1], and the weights w_1, w_2, and w_3 are also normalized in the range [0, 1].

5.2.3 Personalized Recommendation

Assume we have detected several user-product groups by adopting the approaches mentioned above, say $C = \{C_1, C_2, \cdots, C_n\}$. It is obtained by cutting the clustering dendrogram at a pre-specified level of similarity. For example, we cut the dendrogram at 0.4 if we want clusters with a minimum similarity of 0.4. The higher the similarity is chose, the less the cluster groups achieves. Given a new user with a new interested product <$\boldsymbol{u'}$, $\boldsymbol{p'}$>, we need to compare <$\boldsymbol{u'}$, $\boldsymbol{p'}$> with user profile sets and product sets in each co-cluster group. To do so, we adopt the hybrid similarity measurement only with component U and P proposed in Section 3.2. We can obtain the closest group for the new user. Then we rank the products within the closest group by the predicted rating scores and finally select the top k products as the recommendation result.

5.3 Experimental Results

In this section, we empirically analyze the proposed methods using a benchmark dataset to assess the performance of them. We compare the recommendation performance of the proposed soft hierarchical co-clustering algorithm FHCC with the recommendations derived via traditional techniques based on k-means CF and association rules, which illustrates the ability of the proposed algorithm to reveal patterns hidden in the data. MovieLens 100K [94] - a benchmark dataset for recommendation systems is employed to evaluate the performance of our algorithm. This data set contains 943 users who have rated at least 20 movies on the scale of 1 to 5, a total of 1682 movies and 100,000 ratings.

5.3.1 Metric

In this research, the recommendation system is designed to recommend the most likely high-rated products to users. In the experiment, we mask 20% of the actual scores in the rating matrix to evaluate our extracted data model, and use the remaining 80% for training. For each user, we examined top-5 and top-10 recommendations. We define the performance evaluation as a binary class problem, and make some assumption on the experiment dataset. The movies that are actually rated in the recommendation list will be considered as true positive results. Otherwise, if the user does not rate a movie, we assume that he/she is not showing interests in it. The averaged F1-score and averaged NDCG is calculated for the testing users as comparisons. F1-socre is defined as

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

And *NDCG* at position *k* is defined as

$$NDCG@k = \frac{\sum_{i-1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}}{IDCG}$$

In our scenario, $rel_i = 1$ if the user has rated the recommended movie and 0 otherwise.

5.3.2 Behavior of the Recommendation System

Table 5.2. Performance comparison of different collaborative filtering methods

Method	User based CF	Item based CF	FHCC CF
prec@5	0.35136	0.31157	0.43853
prec@10	0.31948	0.28493	0.35011
recall@5	0.14074	0.10936	0.23909
recall@10	0.21633	0.16243	0.21272
F-1 score	0.22947	0.18440	0.28705
NDCG	0.59215	0.55374	0.63379

Table 5.2 presents the recommendation performance of proposed method and the traditional user-based and item-based CF methods. We compare the recommendation precision and recall at the top-5 and top-10 results, denoted by prec@5, prec@10,

recall@5, and recall@10, respectively. Precision reflects the ability of recommendation system to eliminate irrelevant items, and recall measures the ability of recommendation system to return all items that user may find interesting. As we can see, at the top-5 results, proposed FHCC CF achieves 0.43853 of precision, better than 0.35136 of user based CF, and 0.31157 of item based CF. The recall of proposed FHCC CF is also superior to the other two methods at top-5. At the top-10 results, FHCC CF still outperforms the user based and item based CF in terms of precision. With respect to recall, user based CF gains the best result of 0.21633, while the proposed FHCC CF performs 0.21272, which is comparative to the best. Then, we use F1-score to evaluate the balance of precision and recall.

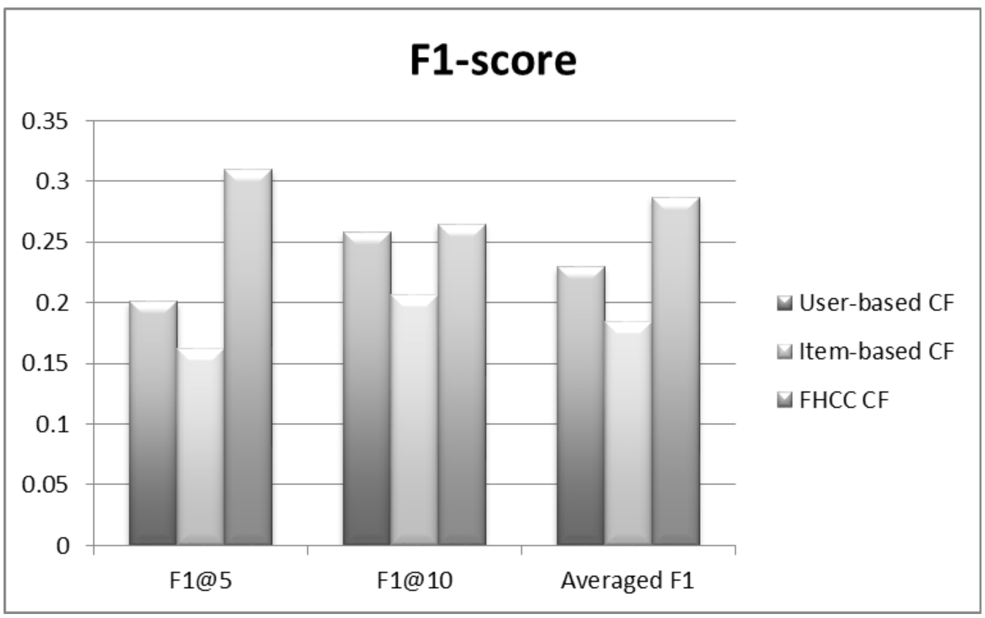


Figure 5.2. F1-score comparison

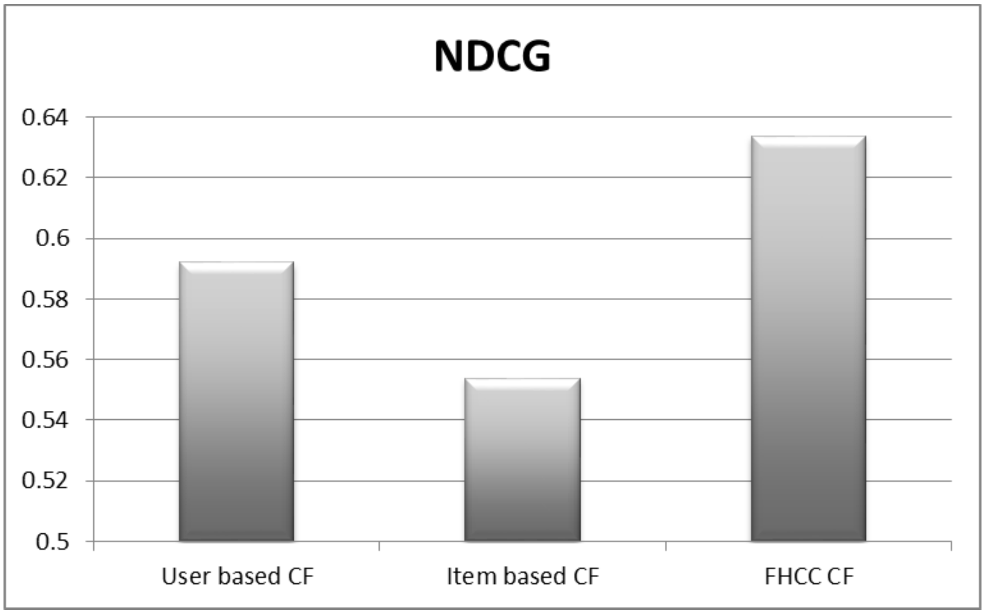


Figure 5.3. Averaged NDCG comparison

Figure 5.2 shows the comparison performance of F1- score at top-5, top-10, and the averaged. It is obvious that proposed FHCC CF is superior to other two CF methods, especially at the most top of the recommending results. Figure 5.3 shows the comparison performance of NDCG, which is used to measure the ranking quality of the recommended list based on a user's actual rating record. We expect that the higher a recommended item ranking is, the more likely the user likes it. If not, NDCG of the corresponding system would be undermined. We can tell from Figure 5.3 that proposed FHCC CF acquires the highest NDCG at 0.63379 which outperforms user based CF at 0.59215 and item based CF at 0.55374.

To sum up, it is apparently that the FHCC CF outperforms other two traditional CF methods from both accuracy and ranking perspectives. This is because in the proposed model, two different data sources, users and product items, are hierarchically clustered simultaneously, which discovers the relationship between the users and products within the same cluster, leads to a more meaningful group detection than other approaches, and hence makes the recommended results more accurate.

Besides, the proposed recommendation framework is more flexible than the traditional recommenders. In our method, a set of control parameters is provided to adjust the weights of user and product components according to different datasets. For example, under the scenario that users provide very limited personal information, we could degrade the weight of user profile similarity computation in the total hybrid similarity computations. It depends on what kind of business dataset we are working on, and how much information is accessible and available to be analyzed for the recommendation purpose, which makes our method more flexible to different applications.

5.4 Concluding Remarks

In this chapter, we have introduced a soft hierarchical co-clustering technique for binary matrices. We address the issue of detecting user-product groups by taking advantage of clustering techniques. Inspired by agglomerative hierarchical clustering approaches, our method is extended with fuzzy set theory to achieve higher flexibility. We have explicitly shown how our method can be coupled with a recommendation system that merges derived co-clusters and individual customer information, as well as the product properties for ranked recommendations. Therefore the method can serve as an interactive tool for examining hypotheses on product offerings. In addition, our approach can assist in the visual identification of market segments to which specific focus should be given, e.g., co-clusters with high propensity for buying emerging products, or products with high profit margin. Empirical experiments conducted on MovieLens 100K dataset demonstrate the effectiveness of our proposed recommendation framework.

Future work will focus on enriching the content of different data sources in order to understand their inner correlations more comprehensively. For instance, we could take into consideration of document similarity measure for product descriptions and user comments. Besides, with the increase of the amount of user rating data, especially documental data, the pairwise similarity comparisons become computational expensive. We will also work on techniques to accelerate the similarity computation and to handle the larges-scale issue in the future.

CHAPTER 6

CONCLUSIONS

6.1 Lessons Learned

Product search is a key component of almost any e-commerce website. When e-commerce search works, it's fast, convenient and efficient. It's no wonder that users prefer searching to clicking categories. Conventionally a user starts a query with textual keywords. The search performance highly relies on how accurate the keywords matches the product attributes defined in database, and it would be undermined by dismal support of unstructured or non-uniformed product information, phonetic misspelling, symbols or abbreviations search, and so on. Product visual context can be useful in product search to identify product items via computer vision techniques. Visual search allows users upload an image of the product they are looking for and find visually similar products on site. In this work, we studied how visual information detected from product images can be used to perform visual search and how visual information and textual information of the products can be leveraged to improve the retrieval performance. Recommendation is another core service in modern e-commerce website. Recommendation systems provide personalized recommendation of products or services to users, facilitate users to discover what they would be interested in, and bring remarkable economic benefit. In this dissertation, we explored how to understand complex relations among users and products from user behaviors for collaborative filtering recommendation.

First, we investigated what type of visual features should be extracted from images for the purpose of performing product search. Our work focused on local feature descriptors because these types of methods extract only local patterns around specific keypoints, and handle the imaging problems of scale, rotation, occlusion and clutter. Aiming at the unstructured characteristics of apparel products, several combinations of classic and promising local feature detectors and descriptors (Harris corner detector, FAST, BRISK, SIFT, and SURF) were assessed. It has been shown that the SIFT descriptor is effective for image matching and unstructured object recognition. All these descriptors are extracted from grey-scale space of image. Considering color is an indispensable characteristic to describe a product, we also studied color features of the image. A RGB color histogram based detector was developed to extract salient keypoints of images, and then customized SIFT descriptors were computed around these color boosted keypoint patches. We developed an image retrieval framework for mobile product search, which represents each database image as a bag of color boosted local features. The color boosted SIFT descriptors extracted from the query image are compared to database of color boosted SIFT descriptors to retrieve the product images in database. Experimental evaluation showed that the developed descriptor proved its capability of performing image search at a higher accuracy than original SIFT, famous color SIFT – Hue-SIFT, and the RGB color histogram alone.

Next, we studied how the textual metadata can be used in visual search system to optimize its retrieval performance. Prior visual search systems often rank the results using their visual distance. However, products in e-commerce database are described with rich, detailed and structured textual content. In this research, we developed a unified probabilistic hypergraph algorithm to model the complex relations within high-dimensional product visual features and textual content. Then, the problems of ranking is formulated as an optimization problem on the constructed hypergraph model. The proposed unified probabilistic hypergraph was compared to visual distance method, and other hypergraph methods for ranking images on a database of apparel products from their visual features. The proposed ranking model achieved comparably better performance.

Finally, we researched complex correlations among high-dimensional user profiles and product characteristics. We focused on clustering techniques for recommendation and proposed a new soft hierarchical co-clustering approach. Through proposed clustering approach, complex relations among different data sources can be comprehensively understood. And the approach is able to adapt to different types of applications according to the accessibility of data sources by carefully adjust the weights of different data sources. Experimental evaluation was performed on a rating dataset to extract user-product co-clusters, and the recommendations can be generated from the user-product preference communities. The generated recommendation results were superior over existing item-based and user-based collaborative filtering recommendations in terms of accuracy and ranked position.

6.2 Implementation Analysis

When it comes to application, the first challenge we face is to handle the scalability issue. Today, the e-commerce sites have millions of products for sale, and serve millions of customers. The data of product information and user logs are generated at unprecedented scale. In this section, we will discuss how to scale our solutions to massive data in implementation. We have an overview on the types of current computational environments that can tackle the big data problems, and provide guidance according to the nature of the proposed algorithms and processed data. These computational solutions include distributed computing, heterogeneous computational environments, and cloud-based computing.

6.2.1 Distributed Computing

In order to address the big data and computational challenges, one of the most important technologies to consider is advanced distributed computing infrastructures. Allocating tasks and workload over more computing and memory resources is a natural solution to the computational and data intensive problems. MapReduce framework is a distributed computing paradigm proposed by Google. It breaks a task down into multiple homogeneous sub-tasks in a map step. In the following reduce step the outputs of sub-tasks are combined in parallel. [1] Through MapReduce framework, scalability and fault tolerance are achieved for massively computing procedures that involve large number of simultaneous processes. MapReduce libraries have been written in many programming languages, which greatly facilitates the development of parallel computing applications. Although the Google implemented MapReduce is proprietary, an open source implementation of MapReduce concept is widely available through Apache Hadoop project. Apache Hadoop is written in Java, and popular in industry and academia for large data sets by virtue of its simplicity and scalability.

Especially, several frameworks have been designed to support scalable processing of graph-structured data, in which the proposed ranking scheme models the relations among products. Apache Giraph and GraphLab are two open source frameworks of them. Giraph is a distributed and fault-tolerant framework built upon Hadoop. Giraph adopts the Bulk Synchronous Parallel programming model to run parallel algorithms for processing large-scale graph data. In Giraph, computations of graph algorithms are executed as a sequence of iterations called superstep. In each superstep, the user-defined function is invoked for each vertex, conceptually in parallel. [2] GraphLab, written in C++, is another graph-based, high-performance, distributed computation framework for processing large graph-structured data. GraphLab performs similar operations to MapReduce, while is specifically designed for sparse iterative graph algorithms. It consists of three major parts: the data graph, the update function, and the sync operation. The user provided data graph represents the program state with arbitrary blocks of memory associated with each vertex and edges. The update function and sync operation are analogous to map and reduce functions. The update function is defined by the user and executed on graph data with small neighbors. The sync operation aggregates data in the whole graph. [3] GraphLab is a parallel-programming abstraction and each GraphLab process is multithreaded to fully utilize the resources of computer clusters.

6.2.2 Heterogeneous Computing

General purpose high-speed, low-cost heterogeneous computing has become available in the recent few years, and has achieved notable successes. Several important algorithms in computer vision have been successfully deployed on GPU, and achieved impressive speedup factors range from 5 to 10 compared with multi-thread CPU implementation. [4] Proposed visual feature extraction is a highly computational complex process. And the GPUs are optimal solutions to problems with tightly coupled fine-grained parallelism.

6.2.3 Cloud Computing

In recent years, with the development of virtualization technology, supercomputing has been made more and more affordable and accessible. The adoption of these on-demand virtual computers, known as cloud computing, could also be an option of choices to solve scalability due to its flexibility and conveniences, although there might be data transfer issues and privacy concerns. Petabyte scales of data can be manipulated and processed with Current cloud computing services in the market can deliver processing capability of petabyte scales of data with these flexible computer architectures.

6.3 Future Work

In this dissertation, we have investigated the problem of product search and discovery in the respects of description, ranking, and recommendation. The ultimate goal of this research is to create an intuitive thoughtful online shopping experience that is universally accessible to everybody, regardless of whether they use text or visuals, as well as to reduce the search and browsing time, making the overall product discovery and shopping experience simple. Despite a further research in proposed three aspects, which are discussed in the Concluding Remarks sections of Chapters 3, 4, and 5, there are several other directions interesting and worth to look into. For one thing, with the emerging wave of camera-enabled smart glasses and wearable devices, it would be another ideal platform for visual search and provide fast product discovery capabilities anywhere the user travels. Thus, adapting the system for wearable devices is a problem that needs to be investigated. For another, a significant research question is: in what form is the product visual data stored in the database? Advanced indexing techniques require to be explored to achieve effective retrieval performance, as well as filter away redundancy. Also, are different retrieval methods required to effectively index different types of visual features? Moreover, large scale is always a great challenge. It is possible that when the scale of the dataset goes tens of millions or even billions, the performance of algorithms discussed in this dissertation would be deteriorated. Thus, extended research is required to learn how the system would work for even larger datasets and to seek better methods to further improve the proposed algorithms.

REFERENCE

1. Brian K. Walker, C.J., Benjamin Zeidler, *The Forrester Wave™: B2C eCommerce Platforms, Q4 2010*. 2010: Forrester.

2. Girod, B., Vijay Chandrasekhar, David M. Chen, Ngai-Man Cheung, Radek Grzeszczuk, Yuriy Reznik, Gabriel Takacs, Sam S. Tsai, and Ramakrishna Vedantham, *Mobile visual search.* Signal Processing Magazine, 2011. **IEEE 28**(no. 4): p. 61-76.

3. Knopp, J., et al. *Hough transform and 3D SURF for robust three dimensional classification*. in *Computer Vision–ECCV 2010*. 2010. Springer.

4. Chandrasekhar, V., et al. *Comparison of local feature descriptors for mobile visual search*. in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. 2010. IEEE.

5. Schafer, J., *The Application of Data-Mining to Recommender Systems.* Encyclopedia of data warehousing and mining, 2009. **1**: p. 44-48.

6. Bulletin, T.R. *Consumers relying on web reviews soars by 84%*. 2009; Available from: http://www.theretailbulletin.com/news/consumers_relying_on_web_reviews_soars_by_84_26-10-09/.

7. E-consultancy. *How We Shop in 2010: Habits and Motivations of Consumers*. 2010; Available from: https://econsultancy.com/reports/habits-and-motivations-of-consumers/.

8. WWWMetrics. *The Growth of Online Shopping*. 2014 [cited 2015; Available from: http://www.wwwmetrics.com/shopping.htm.

9. Ward, M.R. and M.J. Lee, *Internet shopping, consumer search and product branding.* Journal of product & brand management, 2000. **9**(1): p. 6-20.

10. Labelle, P.R., *Initiating the learning process: A model for federated searching and information literacy*. Internet Reference Services Quarterly, 2007. **12**(3-4): p. 237-252.

11. Bayramoglu, N. *Shape index SIFT: Range image recognition using local features*. in *Pattern Recognition (ICPR), 2010 20th International Conference on*. 2010. IEEE.

12. Lowe, D.G., *Distinctive image features from scale-invariant keypoints.* International Journal of Computer Vision, 2004. **60**(2): p. 91-110.

13. He, X.-C. and N.H.C. Yung. *Curvature scale space corner detector with adaptive threshold and dynamic region of support*. 2004. IEEE.

14. Mokhtarian, F. and R. Suomela, *Robust image corner detection through curvature scale space*. Ieee Transactions on Pattern Analysis and Machine Intelligence, 1998. **20**(12): p. 1376-1381.

15. Harris, C. and M. Stephens. *A combined corner and edge detector*. 1988. Citeseer.

16. Mikolajczyk, K. and C. Schmid. *Indexing based on scale invariant interest points*. in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. 2001. IEEE.

17. Mikolajczyk, K. and C. Schmid, *A performance evaluation of local descriptors.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2005. **27**(10): p. 1615-1630.

18. Van De Sande, K.E.A., T. Gevers, and C.G.M. Snoek, *Evaluating color descriptors for object and scene recognition.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2010. **32**(9): p. 1582-1596.

19. Bay, H., T. Tuytelaars, and L. Van Gool, *Surf: Speeded up robust features*, in *Computer vision–ECCV 2006*. 2006, Springer. p. 404-417 %@ 3540338322.

20. Tola, E., V. Lepetit, and P. Fua, *Daisy: An efficient dense descriptor applied to wide-baseline stereo.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2010. **32**(5): p. 815-830

21. Winder, S., G. Hua, and M. Brown. *Picking the best daisy*. 2009. IEEE.

22. Morel, J.-M. and G. Yu, *ASIFT: A new framework for fully affine invariant image comparison.* SIAM Journal on Imaging Sciences, 2009. **2**(2): p. 438-469.

23. Behmo, R., et al., *Towards optimal naive bayes nearest neighbor*, in *Computer Vision–ECCV 2010*. 2010, Springer. p. 171-184.

24. Van De Weijer, J. and C. Schmid. *Applying color names to image description*. 2007. IEEE.

25. Song, X., D. Muselet, and A. Trémeau, *Affine transforms between image space and color space for invariant local descriptors.* Pattern Recognition, 2013. **46**(8): p. 2376-2389.

26. Burghouts, G.J. and J.-M. Geusebroek, *Performance evaluation of local colour invariants.* Computer Vision and Image Understanding, 2009. **113**(1): p. 48-62.

27. Robertson, S.E. and D.A. Hull. *The TREC-9 Filtering Track Final Report*. 2000.

28. Salton, G., A. Wong, and C.-S. Yang, *A vector space model for automatic indexing.* Communications of the ACM, 1975. **18**(11): p. 613-620.

29. Ponte, J.M. and W.B. Croft. *A language modeling approach to information retrieval*. in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 1998. ACM.

30. Kleinberg, J.M., *Authoritative sources in a hyperlinked environment.* Journal of the ACM (JACM), 1999. **46**(5): p. 604-632.

31. Brin, S. and L. Page, *Reprint of: The anatomy of a large-scale hypertextual web search engine.* Computer networks, 2012. **56**(18): p. 3825-3833.

32. Henzinger, M.R., *Hyperlink analysis for the web.* Internet Computing, IEEE, 2001. **5**(1): p. 45-50.

33. Baeza-Yates, R. and E. Davis. *Web page ranking using link attributes*. 2004. ACM.

34. Lempel, R. and S. Moran, *The stochastic approach for link-structure analysis (SALSA) and the TKC effect.* Computer Networks, 2000. **33**(1): p. 387-401.

35. Arevalillo-Herráez, M., J. Domingo, and F.J. Ferri, *Combining similarity measures in content-based image retrieval.* Pattern Recognition Letters, 2008. **29**(16): p. 2174-2181.

36. Collins, J. and K. Okada. *A Comparative Study of Similarity Measures for Content-Based Medical Image Retrieval*. 2012.

37. Cao, Z., et al. *Learning to rank: from pairwise approach to listwise approach*. 2007. ACM.

38. Liu, T.-Y., *Learning to rank for information retrieval.* Foundations and Trends in Information Retrieval, 2009. **3**(3): p. 225-331.

39. Zhou, D., et al., *Learning with local and global consistency.* Advances in neural information processing systems, 2004. **16**(16): p. 321-328.

40. He, J., et al., *Generalized manifold-ranking-based image retrieval.* Image Processing, IEEE Transactions on, 2006. **15**(10): p. 3170-3177.

41. Liu, D., et al. *Tag ranking*. in *Proceedings of the 18th international conference on World wide web*. 2009. ACM.

42. Agarwal, S., et al. *Beyond pairwise clustering*. 2005. IEEE.

43. Zass, R. and A. Shashua. *Probabilistic graph and hypergraph matching*. 2008. IEEE.

44. Sun, L., S. Ji, and J. Ye. *Hypergraph spectral learning for multi-label classification*. 2008. ACM.

45. Huang, Y., Q. Liu, and D. Metaxas. *Video object segmentation by hypergraph cut*. 2009. IEEE.

46. Li, L. and T. Li. *News recommendation via hypergraph learning: encapsulation of user behavior and news content*. 2013. ACM.

47. Huang, Y., et al. *Image retrieval via probabilistic hypergraph ranking*. 2010. IEEE.

48. Gao, Y., et al., *Visual-textual joint relevance learning for tag-based social image search.* Image Processing, IEEE Transactions on, 2013. **22**(1): p. 363-376.

49. McGregor, S.L.T., *Towards adopting a global perspective in the field of consumer studies.* Journal of Consumer Studies & Home Economics, 1998. **22**(2): p. 111-119 %@ 1470-6431.

50. Herlocker, J.L., et al. *An algorithmic framework for performing collaborative filtering*. in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 1999. ACM.

51. Konstan, J.A., et al., *GroupLens: applying collaborative filtering to Usenet news.* Communications of the ACM, 1997. **40**(3): p. 77-87.

52. Shardanand, U. and P. Maes. *Social information filtering: algorithms for automating "word of mouth"*. in *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1995. ACM Press/Addison-Wesley Publishing Co.

53. Ungar, L.H. and D.P. Foster. *Clustering methods for collaborative filtering*. in *AAAI workshop on recommendation systems*. 1998.

54. Xue, G.-R., et al. *Scalable collaborative filtering using cluster-based smoothing*. in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 2005. ACM.

55. Sarwar, B.M., et al. *Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering*. in *Proceedings of the fifth international conference on computer and information technology*. 2002. Citeseer.

56. O'Connor, M. and J. Herlocker. *Clustering items for collaborative filtering*. in *Proceedings of the ACM SIGIR workshop on recommender systems*. 1999. Citeseer.

57. Merialdo, A.K.-B., *Clustering for collaborative filtering applications.* Intelligent Image Processing, Data Analysis & Information Retrieval, 1999. **3**: p. 199.

58. Zhang, D., et al., *Cold-start recommendation using bi-clustering and fusion for large-scale social recommender systems.* Emerging Topics in Computing, IEEE Transactions on, 2014. **2**(2): p. 239-250.

59. Leung, K.W.-T., D.L. Lee, and W.-C. Lee. *CLR: a collaborative location recommendation framework based on co-clustering*. in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 2011. ACM.

60. Chen, G., F. Wang, and C. Zhang, *Collaborative filtering using orthogonal nonnegative matrix tri-factorization.* Information Processing & Management, 2009. **45**(3): p. 368-379.

61. Deodhar, M. and J. Ghosh. *A framework for simultaneous co-clustering and learning from complex data*. in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2007. ACM.

62. Dhillon, I.S. *Co-clustering documents and words using bipartite spectral graph partitioning*. in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. 2001. ACM.

63. Long, B., et al. *Unsupervised learning on k-partite graphs*. in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006. ACM.

64. Hosseini, M. and H. Abolhassani, *Hierarchical co-clustering for web queries and selected urls*, in *Web Information Systems Engineering–WISE 2007*. 2007, Springer. p. 653-662.

65. Ienco, D., R.G. Pensa, and R. Meo, *Parameter-free hierarchical co-clustering by n-ary splits*, in *Machine Learning and Knowledge Discovery in Databases*. 2009, Springer. p. 580-595.

66. Madeira, S.C. and A.L. Oliveira, *Biclustering algorithms for biological data analysis: a survey*. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 2004. **1**(1): p. 24-45.

67. Tanay, A., R. Sharan, and R. Shamir, *Biclustering algorithms: A survey*. Handbook of computational molecular biology, 2005. **9**(1-20): p. 122-124.

68. Vlachos, M., et al. *Improving Co-Cluster Quality with Application to Product Recommendations*. in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 2014. ACM.

69. Zeng, K., N. Wu, and K.K. Yen, *A Color Boosted Local Feature Extraction Method for Mobile Product Search.* Int. J. on Recent Trends in Engineering and Technology, 2014. **10**(2): p. 78-84.

70. Trier, Ø.D., A.K. Jain, and T. Taxt, *Feature extraction methods for character recognition-a survey*. Pattern recognition, 1996. **29**(4): p. 641-662 %@ 0031-3203.

71. Lankinen, J., V. Kangas, and J.-K. Kamarainen. *A comparison of local feature detectors and descriptors for visual object categorization by intra-class repeatability and matching*. in *Pattern Recognition (ICPR), 2012 21st International Conference on*. 2012. IEEE.

72. Rosten, E. and T. Drummond, *Machine learning for high-speed corner detection*, in *Computer Vision–ECCV 2006*. 2006, Springer. p. 430-443.

73. Leutenegger, S., M. Chli, and R.Y. Siegwart. *BRISK: Binary robust invariant scalable keypoints*. in *Computer Vision (ICCV), 2011 IEEE International Conference on*. 2011. IEEE.

74. Tico, M., T. Haverinen, and P. Kuosmanen. *A method of color histogram creation for image retrieval*. in *Proc. Nordic Signal Processing Symp.(NORSIG'2000)*. 2000. Kolmarden, Sweden.

75. Zarit, B.D., B.J. Super, and F.K.H. Quek. *Comparison of five color models in skin pixel classification*. in *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings. International Workshop on*. 1999. IEEE.

76. Eakins, J.P., K. Shields, and J. Boardman. *Artisan: a shape retrieval system based on boundary family indexing*. in *Electronic Imaging: Science & Technology*. 1996. International Society for Optics and Photonics.

77. *Oxford Affine Covariant Features dataset*. Available from: http://www.robots.ox.ac.uk/~vgg/research/affine/.

78. *Columbia University Image Library*. Available from: http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php.

79. Zeng, K., et al., *Ranking via Hypergraph Learning: Integration of Textual Content and Visual Content.* Information and Communication Technologies of WIT Transactions, 2015.

80. Hsu, W.H., L.S. Kennedy, and S.-F. Chang. *Video search reranking through random walk over document-level context graph*. in *Proceedings of the 15th international conference on Multimedia*. 2007. ACM.

81. Jing, Y. and S. Baluja, *Visualrank: Applying pagerank to large-scale image search.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2008. **30**(11): p. 1877-1890.

82. Huang, Y., et al. *Image retrieval via probabilistic hypergraph ranking*. in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. 2010. IEEE.

83. Cao, Z., et al. *Learning to rank: from pairwise approach to listwise approach*. in *Proceedings of the 24th international conference on Machine learning*. 2007. ACM.

84. Karypis, G. and V. Kumar, *Multilevel k-way hypergraph partitioning.* VLSI design, 2000. **11**(3): p. 285-300.

85. Pazzani, M.J. and D. Billsus, *Content-based recommendation systems*, in *The adaptive web*. 2007, Springer. p. 325-341.

86. Li, T. and L. Li, *Music data mining: An introduction.* Music data mining, 2011: p. 1.

87. Datta, R., J. Li, and J.Z. Wang. *Content-based image retrieval: approaches and trends of the new age*. 2005. ACM.

88. Mooney, R.J. and L. Roy. *Content-based book recommending using learning for text categorization*. in *Proceedings of the fifth ACM conference on Digital libraries*. 2000. ACM.

89. Miller, B.N., J.A. Konstan, and J. Riedl, *PocketLens: Toward a personal recommender system.* ACM Transactions on Information Systems (TOIS), 2004. **22**(3): p. 437-476.

90. Ansari, A., S. Essegaier, and R. Kohli, *Internet recommendation systems.* Journal of Marketing research, 2000. **37**(3): p. 363-375.

91. Adomavicius, G. and A. Tuzhilin, *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.* Knowledge and Data Engineering, IEEE Transactions on, 2005. **17**(6): p. 734-749.

92. Burke, R., *Hybrid web recommender systems*, in *The adaptive web*. 2007, Springer. p. 377-408.

93. Li, J., et al., *Hierarchical co-clustering: a new way to organize the music data.* Multimedia, IEEE Transactions on, 2012. **14**(2): p. 471-481.

94. *MovieLens Dataset*. Available from: http://grouplens.org/datasets/movielens/.

95. Dean J, Ghemawat S., MapReduce: simplified data processing on large clusters, 6th Symp. on Operating Systems Design and Implementation, 2004.

96. *Apache Giraph*. Available from: http://giraph.apache.org/.

97. Low, Y., Gonzalez, J. E., Kyrola, A., Bickson, D., Guestrin, C. E., & Hellerstein, J. (2014). *Graphlab: A new framework for parallel machine learning*. arXiv preprint arXiv:1408.2041.

98. Coates, A., Baumstarck, P., Le, Q., and Ng, A. Y., *Scalable learning for object detection with GPU hardware*, in Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 4287–4293, 2009.

VITA

KAIMAN ZENG

2010 – 2015	Ph.D. Candidate, Electrical Engineering Florida International University Miami, Florida
2006 – 2009	M.E., Software Engineering Beihang University Beijing, China
2001 – 2005	B.S., Chemical Engineering China University of Petroleum Beijing, China

PUBLICATIONS

Kaiman Zeng, Nansong Wu, and Kang K. Yen, FHCC: A Soft Hierarchical Clustering Approach for Collaborative Filtering Recommendation. International Journal of Intelligent Information and Database Systems. (submitted)

Kaiman Zeng, Nansong Wu, Arman Sargolzaei, and Kang K. Yen, Ranking via Hypergraph Learning: Integration of Textual Content and Visual Content, Information and Communication Technologies of WIT Transactions, 2015.

Kaiman Zeng, Nansong Wu, and Kang K. Yen, A Color Boosted Local Feature Extraction Method for Mobile Product Search, International Journal on Recent Trends in Engineering and Technology, Vol. 10, No. 2, pp. 78-84, Jan 2014.

Arman Sargolzaei, Kang K. Yen, Kaiman Zeng, Armin Motahari, and Shrin Noei, Impulse Image Noise Reduction Using Fuzzy-Cellular Automata Method, International Journal of Computer and Electrical Engineering, Vol. 6, No. 2, pp. 191-195, Apr 2014.