Local Invariant Feature Based
Object Retrieval in a Supermarket

Yuhang Zhang

A thesis submitted for the degree of
Master of Philosophy at
The Australian National University

February 2009
Except where otherwise indicated, this thesis is my own original work. Part of the content in this thesis has been published in a conference paper:

Y. Zhang, L. Wang, R. Hartley, H. Li: Where’s the Weet-Bix? ACCV (1) 2007: 800-810

Yuhang Zhang
1 August 2008
This work is the result of three years of research work under the supervision of Dr. John Smith, Professor Richard Hunter and Dr. Maria Lee. First of all I would like to express my appreciation to those kind lecturers. Day by day guidance which has been essential in the field of research within materials. Without the help and the understanding of Richard and the support of Maria, this work would not have been possible. Another nice detail to be mentioned here is Dr. Angela White which I also only copied instead of writing down to the proper version. I would like to say, though, that my work would not have been possible without the help of my family. My parents are dedicated to people who can give me appreciation during the work.

To Shuge and my parents.
This work is the fruit of more than one year of research work under the supervision of Dr Lei Wang, Professor Richard Hartley and Dr Hongdong Li. First of all I would like to express my appreciation to them for their day by day guidance which led me from innocent to professional in the field of computer vision research. Particularly, without the help and the contribution of Richard and Lei, the image database creation would have been much more difficult. Another name needs to be mentioned here is Dr Krystian Mikolajczyk. He not only carried out excellent work in the development of local affine features, which enables the image retrieval in this thesis, but also helped me correctly utilize the local invariant features. More thanks are dedicated to all people whoever gave me inspiration during the course.

I also want to express my thanks to my parents, my sister and especially my girlfriend Shuge. Their understanding and encouragement help me through all the challenge and distraction.
Acknowledgements
Abstract

Experienced decades of rapid development on digital imaging sensors and storage devices, today’s world is flooded with digital images. The methods which retrieve images by the text information annotated to each image become less efficient. During the past decade, content-based image retrieval has been paid much attention to and has made significant progress. Inspired by the recent success of content-based image retrieval with local invariant features, this work builds up a retrieval system which can efficiently localize an object in a big supermarket by using its visual information. In contrast to those in the literature, the object retrieval in this work needs to retrieve the images containing extremely strong background clutters and multiple copies of identical objects. Furthermore, there is significant scale difference between a query image and the images in the database. To meet these challenges, this work discusses the deficiency of the existing retrieval methods on this object retrieval task and develops a new similarity measure and a multiple scales retrieval approach to better handle the above problems. In order to achieve high retrieval speed, the mechanisms of the visual word and the invert file borrowed from the text retrieval are applied to the retrieval in this work. In addition, a visual word based spatial check method is proposed to conduct the post-verification in a more efficient way. Through extensive experimental study, this work demonstrates the advantages of the proposed methods over the existing ones for this object retrieval task, as well as the excellent object retrieval performance achieved by this system.
Abstract
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>vii</td>
</tr>
<tr>
<td>Abstract</td>
<td>ix</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation and Objective</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Challenge Identification</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Contribution</td>
<td>6</td>
</tr>
<tr>
<td>1.4 Thesis Outline</td>
<td>6</td>
</tr>
<tr>
<td>2 Literature Review</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Local Invariant Feature Based Image Description</td>
<td>7</td>
</tr>
<tr>
<td>2.1.1 Image Description</td>
<td>7</td>
</tr>
<tr>
<td>2.1.2 Local Feature Detection</td>
<td>10</td>
</tr>
<tr>
<td>2.1.3 Local Feature Description</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Image Retrieval and Visual Vocabulary</td>
<td>14</td>
</tr>
<tr>
<td>2.3 Similarity Measure</td>
<td>17</td>
</tr>
<tr>
<td>2.3.1 Weighting Scheme</td>
<td>17</td>
</tr>
<tr>
<td>2.3.2 Stop List</td>
<td>18</td>
</tr>
<tr>
<td>2.3.3 Similarity Measure Scheme</td>
<td>18</td>
</tr>
<tr>
<td>2.4 Post-verification</td>
<td>19</td>
</tr>
<tr>
<td>2.5 Image Indexing</td>
<td>21</td>
</tr>
<tr>
<td>3 Description of the Images from a Supermarket</td>
<td>23</td>
</tr>
<tr>
<td>3.1 Image Database Creation</td>
<td>23</td>
</tr>
<tr>
<td>3.2 Feature Detection and Description</td>
<td>25</td>
</tr>
<tr>
<td>3.2.1 Feature Detection</td>
<td>25</td>
</tr>
<tr>
<td>3.2.2 Feature Description</td>
<td>26</td>
</tr>
<tr>
<td>3.3 Building Visual Vocabulary</td>
<td>28</td>
</tr>
<tr>
<td>3.3.1 Visual Vocabulary Generation</td>
<td>29</td>
</tr>
<tr>
<td>3.3.2 Image Description with Visual Vocabulary</td>
<td>31</td>
</tr>
<tr>
<td>4 Object Retrieval in a Supermarket</td>
<td>35</td>
</tr>
<tr>
<td>4.1 Similarity Measure with Strong Background Clutters</td>
<td>37</td>
</tr>
<tr>
<td>4.2 Weight of Visual Words</td>
<td>41</td>
</tr>
<tr>
<td>4.3 Image Indexing with an Invert File</td>
<td>42</td>
</tr>
<tr>
<td>4.4 Stop List</td>
<td>43</td>
</tr>
<tr>
<td>4.5 Image Retrieval under Multiple Scales</td>
<td>44</td>
</tr>
<tr>
<td>4.5.1 Image Description under Multiple Scales</td>
<td>44</td>
</tr>
<tr>
<td>4.5.2 Combination by Similarity Score</td>
<td>46</td>
</tr>
<tr>
<td>4.5.3 Combination by Rank</td>
<td>47</td>
</tr>
<tr>
<td>4.6 Visual Word Based Spatial Check</td>
<td>47</td>
</tr>
</tbody>
</table>
5 Experiment

5.1 Performance of Similarity Measure ........................................ 54
5.2 Usage of Weighting Scheme .................................................. 55
5.3 Usage of Partition .......................................................... 58
5.4 Effect of Scale Difference ................................................... 68
5.5 Combining the Results from Multiple Scales ....................... 76
   5.5.1 By Score .......................................................... 76
   5.5.2 By Rank .......................................................... 81
5.6 Retrieval Speed ........................................................... 81
5.7 Retrieval Examples ......................................................... 83

6 Conclusion ................................................................. 87

Bibliography ................................................................. 91
Chapter 1

Introduction

1.1 Motivation and Objective

Many people enjoy shopping, but shopping can be painful sometimes. Imagine one evening, after a whole day’s hard work you are exhausted physically and emotionally. Then you enter a big supermarket to purchase some salted peanuts and beer to reward yourself. However, before you can gain your favourites the multiple level shelves and the endless aisles are waiting for you to scan through (Let us take Figure 1.1 and Figure 1.2 as a trial). More than that, even during holidays when we are energetic, there have been moments when we find everything but what we are really looking for. Although the shops have put up many guiding signs to direct the customers, the number of the signs is just overmatched by that of the products. Shopping is only an example. Every day people are looking for things among clutters and we need help.

Although vision is one of the most familiar senses to human, it has been hard to enable a machine to ‘see’, for example, to make a computer recognize an object by its appearance. In recent years, some local feature based image retrieval systems have been built up and have shown promising retrieval performance. Examples can be found in [Nistér and Stewenius 2006] where the photos of CD covers are retrieved in real time. One interesting question would be whether we can help people find objects in the real world with computer vision technology.

This thesis aims at challenging such a problem by developing an object retrieval system in the environment of a supermarket. The main criterion for the images to be retrieved is that each of them should contain at least one copy of the object in the query image. Even if they appear to be so different from the query image due to the existence of background clutters like that in Figure 1.1, they should be retrieved. Therefore, this retrieval system is named as an object retrieval system in this thesis. To achieve this aim, the following subtasks need to be accomplished.
Figure 1.1: Can you find the Green Tea shown in the image on the left from the image on the right?

Figure 1.2: Can you find the Green Tea in this image? In a real supermarket, surely you will see things much more than those in this image.
1. Building up an image database that can model the complete view of a supermarket: this subtask requires thousands of photos to be collected from a supermarket. It is desirable that each photo only captures a single item on a shelf, because this manner can produce images with little background clutter. However, that is impractical since it requires too heavy human labour. Instead, each photo in this work covers three or four levels of a shelf, containing all the objects on them. As a result, this object retrieval problem is of the difficulty level as finding things in the real world.

2. Describing the images in the database: a proper description to the images should be invariant to the significant changes in scale and background, because the query images provided by users may differ in the above aspects drastically from the images in the database. Besides, viewpoint difference and illumination difference also need to be considered. At the same time, the image description should also be discriminative enough to distinguish thousands of different objects from each other. Thereamong the scale difference and the existence of background clutters are the major obstacles in the object retrieval in a supermarket.

3. Defining a similarity measure: a proper similarity measure for this retrieval task not only needs to identify truly matched images in the case of strong background clutters and large scale difference but also needs to be highly efficient in handling a large image database.

1.2 Challenge Identification

The work in this thesis takes a local feature based approach to tackle the object retrieval in a supermarket, because it can effectively handle the changes in viewpoint, illumination and scale, as well as the existence of background clutters. However, in the way to the above goals, there are still several major challenges to be addressed.

- **Significant scale difference:** scale difference has been one of the major obstacles in local feature based object recognition. To identify a true match, same or similar descriptions should be extracted from the same object in the query image and the database images. However, after a significant scale change, different descriptions are often extracted from the same object in different images. In such a case, the matches need to be identified based on the descriptions bearing significant difference, as shown in Figure 1.3.

- **Strong background clutters:** due to the limitation of the discriminative power of local features, different image regions may share same or similar description, leading to false
matches. Background clutter is one of the factors which contribute to the false matches. The false matches between local features are not fatal as long as they do not outnumber the true ones. However, in the images used in this thesis, the background clutters are extremely strong. In this case, the number of true matches between a query image and a database image can easily be surpassed by that of the false matches. Hence, methods are required to reduce the adverse effect of the background clutters.

- **Multiple copies**: in the literature, a loose spatial check is often employed to remove the false matches that do not have consistent spatial layout. However, in a supermarket several identical items are usually placed together on a shelf. They will lead to multiple sets of identical local features. This case will confuse the spatial check and may wrongly filter the true matches out. An example of such a case is shown in Figure 1.4. Moreover, the existence of multiple copies can also affect the similarity measure between two images, which will be discussed in Chapter 4.1.

- **Speed**: speed will be concerned in two steps of this retrieval task. During building a visual vocabulary, millions of local features need to be clustered into thousands of groups. How to achieve this within endurable time is the first challenge. After describing images with the obtained visual vocabulary, how to find the image having the query object from the database containing thousands of images in real time is another challenge.
Figure 1.4: Assume that the regions with the same colour are true matches. The graph on the left shows the spatial layout of three local features in a query image, including the regions labelled as R, G, B, the pair-wise distances among them, and the angles formed by the lines connecting them. The graph on the right shows the positions of all the matched features in a database image where four copies of the query object are placed side by side. The edges between these copies are shown as the dashed lines which will not be provided in a real problem. As we can see, R2, B1 and G4 are all true matches. However, when being treated as a whole, they could be filtered out because their spatial layout appears to be different from that of the query.
1.3 Contribution

In this thesis, a new object retrieval problem is proposed and solved by way of local invariant feature based approach. To handle the strong background clutters in the database images, this thesis proposes a new similarity measure and improves the retrieval performance. In this object retrieval problem, there is often significant scale difference between the query images and the database images. This work discusses the deficiency of the local invariant features in this case and develops a multiple scale mechanism to elevate the retrieval performance against the scale difference. To achieve a real time retrieval speed, the mechanisms of visual word and invert file are adopted in this work to represent and index the images. A spatial check often incurs considerable extra computational overhead. To improve its computational efficiency, a visual word based spatial check is also proposed. It conducts spatial checking simultaneously with the retrieval process, with little loss in the retrieval speed. In addition, a web based demonstration system is built up to exhibit the retrieval performance achieved by this work.

1.4 Thesis Outline

In Chapter 2, a literature review is carried out on the previous work of content-based image retrieval. The structure of a visual word based image retrieval system will be introduced. The function of each of its components is discussed in detail. Chapter 3 begins with the creation of the image database. After that, the local invariant feature extraction and the visual vocabulary generation are presented. As the core part of this thesis, Chapter 4 discusses how to handle the object retrieval with large scale difference, strong background clutters and the possible existence of multiple identical copies. Particularly, the new similarity measure will be proposed and different weighting schemes are discussed with this measure. Also, the multiple scale retrieval mechanism and the visual word based spatial check will be developed. In Chapter 5, thorough experimental studies will be conducted to support the analysis appearing in the previous chapters and demonstrate the effectiveness of the proposed methods. More discussions will be presented based on the experiment results. Chapter 6 summarizes the work in this thesis and points out the future work.
This thesis is inspired by the local invariant feature based image retrieval approach used in [Nistér and Stewenius 2006] and [Sivic and Zisserman 2003]. In their work, this approach has been taken to retrieve video frames from movies and the photos of CD covers from a large image database. Before diving into the detailed technical content, this chapter first presents a whole picture about this retrieval approach. Figure 2.1 illustrates the prototype of the systems in [Nistér and Stewenius 2006] and [Sivic and Zisserman 2003]. This review will go through each of the components shown in Figure 2.1, introducing what its function is and what has been studied in the literature. Section 2.1 discusses the local invariant features in image description, including both feature detectors and feature descriptors. Section 2.2 introduces the idea of visual words and visual vocabulary, as well as their significance in local feature based image retrieval. Image similarity measures are reviewed in Section 2.3, together with the weighting scheme and the stop list mechanism. Post verification is introduced in Section 2.4 and the methods for image indexing are discussed in Section 2.5.

2.1 Local Invariant Feature Based Image Description

2.1.1 Image Description

Image recognition consists of two major steps. The first step describes each image with a set of visual features, and in the second step the similarity between images is evaluated based on the above visual features. As well known, the visual appearance of an object varies with the condition of viewpoint, illumination, scale, background clutters and occlusions. Moreover, there are innumerable different objects to be distinguished from each other. Hence, how to achieve invariant and discriminative visual features in the first step is critical for successful image recognition.

Generally, the visual features used for image description can be categorized into two
Figure 2.1: System prototype: each blue box represents a component of the system, while the yellow boxes within a blue one illustrate the involved methods. The circles represent the input and output data of each component. The solid and dashed arrows show the data flows in runtime retrieval and off-line training process, respectively.
classes, namely the geometric features and the luminance signature features [Schmid and Mohr 1997]:

- **Geometric features**: pure geometry information such as lines, angles and graphic layouts is extracted from image content to form the geometric features. The efficiency of this type of features strongly relies on the appearance of the object to be described. If the object has no solid edges, angles, or predictable graphic layouts such as clouds or bushes, the geometric features can hardly describe them effectively.

- **Luminance signature features**: the luminance signature features describe an image directly based on its intensity values, such as the colour and grey values. Since it does not require the visual appearance of an object to have a clear geometrical structure, the luminance signature features are often more robust in describing the objects in a wider domain. There are two subclasses in the luminance signature features:
  - **Global features**: are named as ‘accumulating features’ in some literature [Smeulders et al. 2000], because they aggregate the information from an entire image or a whole partition of the image. Possessing the advantage of being simple in form, they are often easy to be compared with each other. However, they are vulnerable to background clutters and occlusions. The successful usage of global features either relies on a pre-segmentation or bears a strong hypothesis that few clutters or occlusions exist in the image.
  - **Local features**: are extracted from different regions of an image and it represents this image by describing the regions separately. Therefore, they are often robust to partial occlusion and background clutters. However, the local features usually result in a more complex form since they appear as a set of features. Because the local features are not sensitive to the existence of background clutters, pre-segmentation is usually unnecessary for local feature based image description.

Due to their efficiency in handling background clutter and partial occlusion, the local features are used more extensively in object recognition than their global counterparts are. In [Schmid and Mohr 1997], they proposed to use a local greyvalue based feature for image retrieval. Their experiments showed that with such a kind of features, more than 1,000 images in their database can be efficiently distinguished from each other. Besides the invariance to image rotation, viewpoint change and partial occlusion, that local feature can also handle a scale change less than 2. Impressed by the efficiency of Schmid and Mohr’s work, much work [Lowe 1999; Schaffalitzky and Zisserman 2002; Tuytelaars and Gool 1999; Tuytelaars
and Gool 2000; Matas et al. 2002; Lowe 2004; Mikolajczyk and Schmid 2004; Kadir et al. 2004; Mikolajczyk and Schmid 2005] has been carried out using local invariant features for object recognition. The extraction of local features generally involves the following two steps:

1. **Feature Detection**: instead of computing local features from every pixel of an image, a feature detector is often used to identify the interest regions in the image. One important characteristic of the interest regions is that they are able to be detected repetitively when the viewpoint, the scale and the illumination of an image change.

2. **Feature Description**: each interest region is described by using its visual content such as the intensity gradient. The feature description needs to be invariant to image rotation and the changes of scale, illumination and viewpoint.

### 2.1.2 Local Feature Detection

A comprehensive review of the existing feature detectors has been carried out in [Mikolajczyk et al. 2005]. Detectors, including the Harris-Affine [Mikolajczyk and Schmid 2002; Mikolajczyk et al. 2005], Hessian-Affine [Mikolajczyk and Schmid 2002], MSER (maximally stable extremal regions) [Matas et al. 2002], EBR (Edge-based regions) [Tuytelaars and Gool 1999], IBR (intensity extrema based regions) [Tuytelaars and Gool 2000] and Salient Regions detector [Kadir et al. 2004] are studied and experimentally compared with each other. After testing against different conditions such as scale and viewpoint changes, the review concluded that no feature detector consistently outperforms all the others in all conditions. However, compared with the others, the Harris-Affine and the Hessian-Affine detectors often generate a larger number of interest regions, whose areas are often smaller. This characteristic makes the two detectors handle the recognition of the smaller-sized objects better than the others, especially in the case of the presence of strong background clutters. The development of the Harris-Affine detector has experienced the steps from the Moravec’s corner detector, through the Harris corner detector, the Harris-Laplacian detector, to the Harris-Affine detector. These steps equip the Harris-Affine detector with functionality in different aspects and finally build up this illumination, rotation, scale and affine transform invariant or semi-invariant feature detector. The Hessian-Affine detector is derived from the Harris-Affine detector with minor modification in the step of interest point detection.

Chris Harris and Mike Stephens proposed the Harris corner detector in [Harris and Stephens 1988]. The Harris corner detector is developed from Moravec’s corner detector which is the first intensity based local feature detector. The Harris corner detector is isotropic and can assess the intensity change in any directions whereas the Moravec’s corner detector can only
deal with the intensity change in every 45 degrees. The Harris detector employs a smooth Gaussian kernel to demarcate the assessed region being while the Moravec’s detector uses a binary rectangular window. Moreover, instead of following the Moravec’s detector to measure the intensity change directly, the Harris detector computes a second moment matrix from the image intensity and uses a response function to distinguish the local regions of an image to be the corners, the edges and the flat regions in a image. The region detected by the Harris corner detector is rotation invariant. Moreover, based on the derivatives of image grey value the Harris detector is robust against illumination change.

One problem that is not handled by the Harris corner detector is the scale difference between images. The scale of the Gaussian kernel used in the Harris detector is a constant. Hence a Harris corner detector always detects the regions of the same number of pixels in all images. However, under different scales, the same image content is described by different number of pixels in different images. To find a region detector which can handle the scale difference between images, a variety of methods can be found in [Mikolajczyk and Schmid 2004; Dufournaud et al. 2000; Lindeberg 1998]. In [Mikolajczyk and Schmid 2004], the detection of the corner points is implemented in the scale space. That is as the scale of an image is gradually decreases by a same factor, the Harris corner detection is implemented under each of the obtained scales. The points which cannot be detected over sequential scales are discarded since they are regarded as not stable against the scale variance. For each of the remaining points which can be detected over sequential scales, a Laplacian of Gaussian is then used to identify the characteristic scales. Under the so called characteristic scale, the Laplacian measure of the corner point achieves a local maximum in scales. In this way, the related local regions in different images can be always detected with the pixels corresponding to the same region in the characteristic scale, leading to the same region description. The experiment in [Mikolajczyk and Schmid 2001] shows that compared to the Square gradient, the Difference of Gaussian and the Harris response measure, the Laplacian measure gains the highest accuracy in the characteristic scale selection. However, as the scale difference increases, the repeatability of the local regions detected by the Harris-Laplacian detector will decrease [Mikolajczyk and Schmid 2001]. Firstly, this is because as the scale goes down, some points in the image disappears due to the pixel mergence and can no longer be detected by the Harris corner detector. Secondly, since the same set of detection scales is applied to different images, corresponding regions cannot be detected in some of the images as the local maximum of their Laplacian measure do not exist in the detection scales.

Both the integration Gaussian kernel and the local Gaussian kernel used in the Harris-Laplacian detector are uniform in all directions. That is correct only if the scale of a local
region varies uniformly in all directions. In other words, if the scale change of a local region is different in different directions, for example, due to the change in viewpoint angle, errors will be incurred. To improve the detector's robustness against the change in viewpoint angle, the work in [Mikolajczyk and Schmid 2002] proposed to use two non-uniform Gaussian kernels constructed from two $2 \times 2$ matrices $\Sigma_I$ and $\Sigma_D$ to replace the two uniform Gaussian kernels, respectively. Particularly, they set $\Sigma_I = s \Sigma_D$ where $s$ is a scalar. To figure out the $\Sigma_I$ or the $\Sigma_D$, a solution provided in [Lindeberg and Garding 1997] is used. In the solution, each anisotropic region indicated by a non-uniform Gaussian kernel is related to an isotropic region indicated by a uniform Gaussian kernel in the so called normalized frame. Through such a transformation, the corresponding local regions detected in images with different viewpoint angles are affine transformed from the same pattern in the normalized frame. Hence the invariance to the affine transform is achieved. Reader are referred to [Mikolajczyk and Schmid 2002] for more details about the detection algorithm.

As another local affine region detector, the Hessian-Affine detector is almost identical to the Harris-Affine detector except for the way of localizing the interest points in a 2-dimension image. In other words, the Hessian-Affine detector can be constructed from the Harris-Affine detector by replacing the Harris detector with the Hessian detector. To build a Hessian-Affine detector, the second moment matrix in the Harris-Affine detector is replaced with the Hessian matrix. An interest point will be detected if it forms a local extreme value in both the determinant and the trace of the Hessian matrix. By forcing the interest points to form an extreme value in the trace, the Hessian-Affine detector is able to localize the blob-like patterns in an image. In the work of [Mikolajczyk et al. 2005], the Hessian-Affine detector performs second best to the MSER (maximally stable extremal regions) [Matas et al. 2002] detectors among all the affine invariant region detectors. However, it suits this project better than the MSER because it can detect regions of smaller size and bigger quantity than MSER does.

2.1.3 Local Feature Description

With a local feature detector, corresponding regions can be detected between images with different illuminations, viewpoints, scales and etc. To identify the true matches between the images, the detected regions need to be represented with the descriptors which can be compared with each other. Moreover, these descriptors should be invariant to the above differences between images.

Lowe proposed the SIFT (Scale Invariant Feature Transform) descriptor in [Lowe 2004]. The comparison in [Mikolajczyk and Schmid 2005] shows that the SIFT descriptor outper-
forms the other descriptors in the literature, such as the shape context [Belongie et al. 2002],
the steerable filters [Freeman and Adelson 1991], the differential invariants [Koenderink and
van Doom 1987] and etc. The idea of the SIFT descriptor is inspired by the biological vision
model used in [Edelman et al. 1997] and computed from the distribution of the intensity gra-
dients in an interest region. The steps of computing this descriptor are as follows.

1. Sampling intensity gradients in an interest region: the magnitude and orientation of the
   intensity gradient are sampled within each sub-region of the interest region, as illus-
   trated by Figure 2.2.

2. Weighting the gradient magnitude: a Gaussian kernel is used to assign different weights
to different sub-regions. The size of this Gaussian kernel is one half of the sampled area,
so that the sub-regions farther from the interest point have smaller impact to the region
descriptor. This is used to reduce the affect caused by the detection error such as a small
shift of the interest point’s position or the variance in the size of the detected region.

3. Creating a histogram over 4 × 4 adjacent sub-regions: for every 4 × 4 adjacent sub-
regions, a histogram with 8 bins is created to summarize the gradient distribution in 8
directions, as shown in Figure 2.3. In this way, the position shift of the points within
the 4 × 4 adjacent sub-regions will not change the region description. Moreover, a
trilinear interpolation is used to assign each magnitude into the adjacent bins of the
histogram. In this manner, a smooth rotation of the gradient orientation will not signif-
icantly change the region descriptor.

Figure 2.2: This figure shows the sampled intensity gradients over 8 x 8 sub-regions around
an interest point.
Figure 2.3: The gradient magnitudes of the sub-regions in Figure 2.2 are accumulated into $2 \times 2$ sub-windows. Each sub-window summarizes the distribution of the gradients in a set of $4 \times 4$ adjacent sub-regions in Figure 2.2. The length of each arrow shows the total gradient magnitude in the corresponding direction.

4. Generating an SIFT descriptor: as shown in [Lowe 2004], the best recognition performance is achieved by dividing the whole interest region into $16 \times 16$ sub-regions and describing every $4 \times 4$ adjacent sub-regions with a histogram of 8 bins. That gives rise to a histogram of 128 ($4 \times 4 \times 8$) bins for the whole interest region. Thus the SIFT descriptor of each detected region is of 128 dimensions.

5. Handling illumination changes: linear contrast change is tackled by normalizing the SIFT descriptor to a unit vector. Linear brightness change does not change the intensity gradients and the SIFT descriptor. Compared with the gradient magnitude, the gradient orientation is less sensitive to non-linear illumination change. Hence, the SIFT descriptor emphasizes more on the orientation. After normalization, if the value in any bin is larger than 0.2, it will be replaced by 0.2 and the histogram is normalized again.

2.2 Image Retrieval and Visual Vocabulary

As early as 1970’s, people began to use computer systems to retrieve images from a large database. At the beginning, the image retrieval systems mainly relied on the annotations attached to each image. Methods of this type are named text-based image retrieval. Nevertheless, compared to text documents, images are much harder to sufficiently and accurately describe. What makes the situation worse is that a vast amount of labour is required to do the manual annotation. To improve such a situation, content-based image retrieval (CBIR) is pro-
posed, which integrates the techniques of computer vision, information retrieval and database management.

One of the major obstacles in content-based image retrieval is the lack of effective image description. An ideal description should be discriminative enough to distinguish objects and adequately robust against the changes in illumination, scale, background clutters and occlusions. Also, the description should facilitate fast retrieval in a large image database. The work in [Schmid and Mohr 1997] has demonstrated the robustness of the local invariant features in image retrieval. However, some mechanism is still required to boost the runtime speed of the local feature based image retrieval in order to handle a large image database.

The recent years have witnessed the successful application of a visual vocabulary in the local feature based image retrieval. A visual vocabulary is composed of visual words, which are the outcome of analogizing content-based image retrieval to text retrieval [Sivic and Zisserman 2003]. In text retrieval, documents are described with the appearance frequency of different words, which leads to a high retrieval efficiency. Since images do not contain words naturally, algorithms are required to generate 'words' for images. In [Sivic and Zisserman 2003], the generation of visual words is approached by clustering the local feature descriptors extracted from a set of video frames selected from a movie. After described with a SIFT descriptor, each local region is mapped to a point in a 128-dimensional space. Through clustering, the points corresponding to the local regions with similar visual appearances would be assigned to the same cluster. A visual word can be used to summarize the points in the same cluster, just like the case in text retrieval where the appearances of 'drink', 'drinks' and 'drinking' would all be summarized as 'drink'. Visual words are generated through clustering the region descriptors in a high dimensional space, and all the visual words compose a visual vocabulary.

Sivic and Zisserman managed to retrieve video frames containing particular objects by way of image retrieval in [Sivic and Zisserman 2003]. In their work, the Shape Adapted regions [Schaffalitzky and Zisserman 2002] and the MSER (maximally stable extremal regions) [Matas et al. 2002] were extracted from 10,000 frames selected from a movie. Since the content in the video frames of a movie is continuous, the stability of the detected regions can be checked over adjacent frames. The 10% most unstable regions were discarded. Then the remaining regions were described into SIFT descriptors and a subset of these descriptors were selected to generate the visual vocabulary. The descriptors of the two types of regions were clustered with the k-means algorithm respectively. The distance between two SIFT descriptors was calculated by the Mahalanobis distance and as declared in [Sivic and Zisserman 2003] the Euclidean distance can be an option. The k-means algorithm was run multiple times
and the best clustering result was selected. With the obtained visual vocabulary, each video frame was described with a histogram showing the frequencies of each visual word appearing in the frame. The similarity between two frames were then measured through comparing the associated histograms.

Nistér and Stewenius followed Sivic and Zisserman's work to retrieve the images of CD covers from a database containing 40,000 images [Nistér and Stewenius 2006]. The MSER and the SIFT were utilized as the region detector and the region descriptor respectively. Their work showed that a larger-sized visual vocabulary performs better in retrieval. The reason is that with a larger-sized vocabulary, the region descriptors in each cluster are closer to each other and thus the descriptors in the same cluster are more likely to form true matches. To achieve high retrieval performance, the work in [Nistér and Stewenius 2006] used as many as 35,000 images to generate a much larger-sized vocabulary (containing one million visual words) than that used in [Sivic and Zisserman 2003].

Using more images to generate a larger-sized vocabulary incurred heavier computational load. To handle such a challenge, the work in [Nistér and Stewenius 2006] proposed a hierarchical $k$-means algorithm. All the region descriptors were clustered into $k$ subsets at the first level, and the descriptors in each subset were further clustered into another $k$ subsets at the next level. Therefore, $k$ denotes the branch factor at each level instead of the total number of the resulting clusters. Even with a small $k$, this hierarchical algorithm can still cluster all the region descriptors into a large number of subsets through several levels. This is much more computationally efficient than finishing such a clustering in one shot, because each level handles a much smaller clustering problem. Another advantage brought by the hierarchical algorithm is that the tree structure formed during the vocabulary generation defines a set of search routines. These search routines can efficiently assign new descriptors to proper clusters.

One important issue is whether and how the usage of a visual vocabulary can improve the efficiency of image retrieval. The answer is positive. Before using a visual vocabulary, the local invariant feature based image retrieval [Schmid and Mohr 1997] and object recognition [Lowe 1999] employ a runtime nearest neighbour search to find the true matches between region descriptors. With a large number of high dimensional region descriptors, this process becomes quite time consuming, even with the aid of approximate search and indexing mechanisms. In [Schmid and Mohr 1997], retrieving an object from a database containing 1,020 objects averagely costs about 5 seconds based on a Sparc 10 Station. In [Lowe 1999], to recognize the objects in an image took about 2 seconds on a Sparc 10 Station. However, when using a visual vocabulary, the nearest neighbours of each of the extracted region descriptors have already been identified during the vocabulary generation, namely the descriptors in the same cluster.
Hence the retrieval in a database containing 40,000 images can be carried out within real time in [Nister and Stewenius 2006]. Other image retrieval systems that use the visual vocabulary mechanism can be found in [Chum et al. 2007; Philbin et al. 2007; Sivic et al. 2004] and etc.

### 2.3 Similarity Measure

After the creation of a visual vocabulary, each image can be described with a histogram showing the appearance frequency of each visual word in the vocabulary. The similarity between images is represented by the similarity between histograms.

In [Sivic and Zisserman 2003], two mechanisms were borrowed from the text retrieval to improve the similarity measure in image retrieval, namely the \textit{tf-idf} weighting scheme and the stop list. Before going into more details of the similarity measure between images, an introduction to the weighting scheme and the stop list is given in the following two sections respectively.

#### 2.3.1 Weighting Scheme

In text retrieval, different words have different power in distinguishing documents. Words like 'a', 'the', 'of' and 'and' are too common in all documents. People or machines can hardly distinguish a document from the others by merely knowing that it contains these words. On the other hand, words like 'music' and 'mathematics' are not likely to appear in every document, so they are much more effective in distinguishing documents. Weighting algorithms have been designed to measure the weight of each word in document description. The commonly used \textit{tf-idf} weighting scheme is given by Equation 2.1.

\[
t_{ij} = \frac{n_{ij}}{n_{j}} \log \frac{N}{n_{j}}
\]  

(2.1)

In Equation 2.1, \(t_{ij}\) is the weight of the \(j\)th word in the \(i\)th document, \(n_{ij}\) is the occurrence number of the \(j\)th word in the \(i\)th document, \(n_{i}\) is the number of all the words in the \(i\)th document, \(n_{j}\) is the occurrence number of the \(j\)th word in the whole database, and \(N\) is the number of the documents in the database. In this equation, the part \(\frac{n_{ij}}{n_{i}}\) measures a word's weight according to the information within the \(i\)th document, while the part \(\log \frac{N}{n_{j}}\) measures this weight according to the information from the whole database. As they have shown, the more frequently a word appears in a document, the higher its weight. The fewer documents in the database contain such a word, the higher this word's weight.
Another form of tf-idf weighting scheme [Nistér and Stewenius 2006] is given by Equation 2.2.

\[ t'_{ij} = n_{ij} \log \frac{N}{N_j} \]  

(2.2)

Compared to Equation 2.1, this equation removes the \( n_i \) on the denominator and replaces the \( n_j \) with \( N_j \) which is the number of documents where the \( j \)th word at least appears once. According to [Nistér and Stewenius 2006], the alternation of \( n_j \) and \( N_j \) makes little difference to the ultimate retrieval performance. Moreover, since the \( d \)th document is represented by the vector of \( V_d = (t_{i1}, ..., t_{ik})^T \), the removal of \( n_i \) makes no difference at all if \( V_d \) is normalized to a unit vector. Note that \( k \) denotes the number of words in a vocabulary. It is straightforward to use this scheme to compute the weights of the words in a visual vocabulary.

### 2.3.2 Stop List

In [Sivic and Zisserman 2003], after sorting all the visual words in a vocabulary by their appearance frequencies in the database, the top 5% and bottom 10% are removed from image description. During retrieval, the data of image description needs to be maintained in computer memory in order to achieve high retrieval speed. Those frequently appearing visual words not only contribute little to the similarity measure but also occupy large memory space and computation. Therefore, it is reasonable to get rid of them completely with the stop list. The reason for putting words of extremely low appearance frequencies into the stop list is that this type of visual words seldom make contribution to the similarity measure since they can hardly be found in any image. Removing them would also save system resource with minor retrieval performance degradation.

### 2.3.3 Similarity Measure Scheme

In [Sivic and Zisserman 2003], after weighting and utilizing the stop list, each image is described with a vector

\[ V_i = (t_{i1}, ..., t_{ik})^T \]  

(2.3)

where \( i \) is the image identity number, \( k \) is the total number of the visual words in the vocabulary after applying the stop list. The similarity between two images is determined by the Euclidean distance between the corresponding normalized vectors:

\[ S_{Eul} = \frac{V_q - V_d}{\| V_q \|_2 \| V_d \|_2} \]  

(2.4)
In [Nister and Stewenius 2006], it was observed that replacing the Euclidean distance with the $L_1$-norm distance (Equation 2.5) results in better retrieval performance.

$$S_{L_1} = \left\| \frac{V_q}{V_d} \right\|_1 - \left\| \frac{V_q}{V_d} \right\|_1$$

(2.5)

### 2.4 Post-verification

Even between two visually completely different images, it is possible to find several or even more matched local regions. These false matched regions are detected from different objects but have similar visual appearance. Currently no descriptor is robust enough to avoid such situation. Neither can this problem be solved by using a stricter match condition because in that case many true matches would also be filtered out by mistake. In [Lowe 2004; Sivic and Zisserman 2003; Schmid and Mohr 1997; Zhang et al. 1995], different post-verification methods have been adopted to eliminate the false matches. All these methods are based on the spatial information which has been proved to be an effective complement to local feature descriptors.

In [Zhang et al. 1995] after computing the correlation between the local regions of two images, a number of matches were identified. Some regions in the first image may be matched to multiple regions in the second image and vice versa. All these matches were taken as candidate matches. To identify the true matches from all the candidate matches a criterion was designed to calculate the strength of each candidate match:

$$S(m_{1i}, m_{2j}) = c_{ij} \sum_{n_{1k} \in N(m_{1i})} \left[ \max_{n_{2l} \in N(m_{2j})} \frac{c_{kl} \delta(m_{1i}, m_{2j}; n_{1k}, n_{2l})}{1 + dist(m_{1i}, m_{2j}; n_{1k}, n_{2l})} \right],$$

(2.6)

where

- $m_{1i}$ is the $i$th local region in the first image (please see Figure 2.4),
- $m_{2j}$ is the $j$th local region in the second image,
- $n_{1k}$ is the $k$th neighbouring region of $m_{1i}$ in the first image,
- $n_{2l}$ is the $l$th neighbouring region of $m_{2j}$ in the second image,
- $N(m_{1i})$ is the set of all the neighbouring regions of $m_{1i}$, in which all the regions lie in a circle centered at $m_{1i}$ with $R$ as the radius,
- $N(m_{2j})$ is the set of all the neighbouring regions of $m_{2j}$, in which all the regions lie in a circle centered at $m_{2j}$ with $R$ as the radius,
Figure 2.4: $m_{11}$ and $m_{21}$ compose a candidate match, $n_{11}, n_{12}, n_{21}, n_{22}, n_{23}$ are the neighbouring regions affecting the matching strength between $m_{11}$ and $m_{21}$.

$$\text{dist}(m_{11}, m_{21}; n_{1k}, n_{2l}) = \frac{[d(m_{11}, n_{1k}) + d(m_{21}, n_{2l})]}{2}$$

where $d(m, n)$ is the Euclidean distance between the positions of region $m$ and $n$,

$$\delta(m_{11}, m_{21}; n_{1k}, n_{2l}) = \begin{cases} e^{-r/\epsilon_r} & \text{if } (n_{1k}, n_{2l}) \text{ is a candidate match and } r < \epsilon_r \\ 0 & \text{otherwise} \end{cases}$$

and $\epsilon_r$ is a user-defined threshold.

The functionality of this criterion can be summarized as follows:

1. the more the candidate matches exist between the neighbouring regions of the checked candidate match, the stronger the match being checked is;

2. a candidate match of higher similarity within the two neighbouring regions would contribute more to the matching strength between the checked candidate match;

3. the contribution of a candidate match within the two neighbouring regions is weighted by the distances between the neighbouring region and the checked region, such as $d(m_{11}, n_{1k})$ and $d(m_{21}, n_{2l})$. A neighbouring region which is farther away from the checked region contributes less.

4. in the case of that the distance between the checked region and its neighbouring region in the first image (e.g. $d(m_{11}, n_{11})$) is different from that in the second image (e.g.
The contribution of this match between the neighbouring regions will be scaled down. However, this rule is more tolerant for farther neighbouring regions.

[Zhang et al. 1995] also pointed out that in the case of knowing the image plane rotation range, the angle between $m_{1k}n_{ik}$ and $m_{2j}n_{2j}$ can be a spatial check criterion as well. For example, given two images between which the rotation angle is less than $\Theta$ and a candidate match between two neighbouring regions $(n_{1k}, n_{2j})$, if the angle between $m_{1k}n_{ik}$ and $m_{2j}n_{2j}$ is larger than $\Theta$, the corresponding $\delta(m_{1k}, m_{2j}; n_{1k}, n_{2j})$ would take a value of zero, indicating at least one of the two matches is a false match and they cannot support each other.

The idea of Zhang et al. has been adopted in the later work with small modifications. In [Schmid and Mohr 1997], the $p$ closest neighbours of each local region are selected. During matching, only if no less than 50% of their $p$ closest neighbours are matched do the two local regions match. In [Lowe 2004], where the 15 nearest neighbours of each local feature are selected. Any candidate match without any other match identified among their 15 nearest neighbours will be rejected.

As we can see in the above researches the well known RANSAC based epipolar geometry method [Fischler and Bolles 1981; Hartley and Zisserman 2003] was not utilized to check the spatial consistency between local regions. The reason is that in images with strong clutters, false matches are so common and numerous that they may outnumber the true matches. Under such conditions, RANSAC would consume a lot of time to reject the wrong sampling and often gives a wrong estimation of the fundamental matrix ultimately [Tuytelaars and Gool 2000]. Instead, using a much looser geometry constraint such as requiring the existence of matches between neighbouring regions often achieves promising improvement to the recognition performance.

2.5 Image Indexing

The most challenging and exciting part of an information retrieval system is the capability to handle large databases. As the number of the images and the dimension of the image descriptor increase, storing the data linearly in the database and scanning them one by one during retrieval will cost considerable time. To improve the capability of handling a large database, indexing mechanism becomes an indispensable component of an information retrieval system. So it is for an image retrieval system. Generally image description generates a vector (global feature based image description) or a set of vectors (local feature based image description) for each image. Since each vector corresponds to a point in a multi-dimensional space,
the indexing of images is actually indexing the points in the space.

Before the usage of visual words, most indexing methods in content-based image retrieval involve tree structures like \textit{kd-tree}, \textit{R-tree} and etc. In those methods, the descriptors are split into subsets recursively as the tree branches. During retrieval, a search is carried out with the aid of the tree structure and the retrieval time is saved due to no need of checking all the branches [Smeulders et al. 2000; Samet 1984].

With the introduction of visual words, successful indexing methods of text retrieval are transplanted into content-based image retrieval. Invert file is one of the most efficient indexing mechanisms in text retrieval. As its name illustrates, an invert file stores the content of an ordinary file in an inverted way. Commonly, the identity number of each file in a database is used by a system to localize this file, and the words in a text file can only be accessed after the file is localized. In contrast, in an invert file, it is the words in a documents that are used by the system to localize the file. After that, the identity numbers of all the documents containing a particular word can be immediately obtained. An invert file that is familiar to most people is the ‘index’ at the end of a book. For example, if you want to find out in which page an encyclopedia mentions computer vision, instead of reading the book from the first page you would probably search for the key word ‘computer vision’ in the index and go to the page directly. Successful applications of the invert file mechanism to image retrieval can be found in [Sivic and Zisserman 2003; Nister and Stewenius 2006].
Description of the Images from a Supermarket

3.1 Image Database Creation

To build an image database which can model the environment in a supermarket, one simple manner is to take the photo of each object in a supermarket and assemble all the photos to form an image database. However such a manner is impractical due to the requirement of a large amount of human labour. Instead, each image in this work contains all the objects on three or four levels of a shelf. As shown in Figure 3.1, some of the objects within one image may be identical. Totally 3,153 images are collected from a local supermarket named Coleus in Canberra. The format of the images is JPEG and their resolution is either $2,272 \times 1,704$ or $2,592 \times 1,944$. These images cover 18 rows of shelves each of which is about 30 meters long. Two adjacent images have some overlapping to ensure that no part of a shelf is missed and that each individual object appears completely at least once in all the collected images. When these photos were being captured, no special illuminance, viewpoint or scale constraints were used. Blur may exist in some of the images. All the 3,153 images form the image database used by this work to conduct object retrieval in a supermarket.

A query set is also created for the object retrieval. Each query image contains a single object in a plain background. The resolution of each query image is either $2,272 \times 1,704$ or $2,592 \times 1,944$. To test the system’s capability in handling image rotation and viewpoint difference, two or three query images are captured from each object with different rotation angles and different viewpoints, as shown in Figure 3.2. Based on their shapes, all the query objects can be roughly categorized into three classes, namely the boxes, the bottles and the bags. Among all the three, the bags are the most difficult to retrieve because they are non-rigid. The bottles are rigid hence easier than the bags, but their surfaces are not planar. The
boxes are the easiest to retrieve due to their rigid and planar surfaces. Besides the shape, the size of a query object is another significant factor for the retrieval performance. Generally the objects of bigger sizes are easier to retrieve than those of smaller sizes because more local invariant features can often be extracted, leading to a more comprehensive description.

With the above image database and the query set, the proposed retrieval system will function like this: given a query image, the system will return at least one database image containing the object in the query image in real time. Figure 3.1 and Figure 3.2 demonstrate that this object retrieval problem has the following characteristics:

1. the illumination and the viewpoint of the same object may be quite different between the query images and the database images;

2. the scales of identical objects in the query images and the database images may be drastically different. Generally the scale of the object in a database image is much smaller than that in a query image;

3. there often exist multiple copies of an object within a single database image;

4. the background clutters in the database images are extremely strong. The object to be retrieved only occupies a small region of a database image and all the rest part is full of various objects, which become the background clutters in retrieval.

Figure 3.1: Sample images from the database
3.2 Feature Detection and Description

Based on the characteristics of the proposed retrieval problem, an effective description to these images should be capable to handle strong background clutters. Global features are not reliable because they summarize the image content altogether and lose the detailed information for each individual object. On the other hand, local features characterize an image by describing different local regions separately. Therefore they are a better option. To describe an image with local features, feature detection and feature description need to be implemented successively.

3.2.1 Feature Detection

Several types of local affine region detectors have been proposed in the literature. The Hessian-Affine detector produces local regions of smaller sizes and larger quantity than others under the same situation. These characteristics suit the retrieval problem in this thesis better, and the reasons are as below.

1. There are dozens of objects in each database image. Therefore a large number of features are required to sufficiently represent each object.

2. Each object in an database image occupies only a small region thus a large-sized local region would bear a high risk to describe more than one objects together. In other words, local regions of smaller sizes are more effective.
3. Small-sized local regions would be less likely affected by non-rigid and non-planar transforms, hence they are more robust in describing bottles and bags.

During the implementation, the Hessian-Affine regions are detected with the binaries provided by Mikolajczyk\(^1\). Figure 3.3 and Figure 3.4 show the Hessian-affine regions detected from a database image. Note that the sizes of the detected regions vary drastically. Whereas some of them are only composed of about 250 pixels, some others may contain as many as 100,000 pixels. For the interest of the retrieval goal, those \textit{too big} regions are abandoned because they are always describing more than one objects. Regions in reasonable sizes are shown in Figure 3.4. We can see that some of them are corner based and some of them are blob based. In addition, identical patterns have been detected repetitively as expected.

3.2.2 Feature Description

The detected regions are described with the SIFT descriptors proposed in [Lowe 2004]. During description, each local region is partitioned into 16 × 16 subregions, and the gradient of grey value is calculated within each subregion. These subregions are then grouped into 4 × 4 non-overlapping windows. Each window covers 4 × 4 subregions. The distribution of the gradient in the same window is represented by a histogram composed of 8 bins recording the total gradient magnitude in 8 different directions. Since there are 16 windows, altogether they form a histogram of 16 × 8 = 128 dimensions. Each dimension denotes the gradient magnitude along a particular direction in one of the 16 windows of a local region.

Linear illumination variance would change pixel value by adding or multiplying a constant. The gradient of the grey value would only be affected by the latter change. Lowe proposed to normalize each 128 dimensional SIFT descriptor to remove such affect in [Lowe 2004]. On the other hand non-linear illumination variance would cause unpredictable change of the pixel values and the gradients. However, as mentioned in [Lowe 2004], such variance can hardly change the gradient direction. Lowe proposed to cap the value of each dimension of the histogram with 0.2. In case of violation, the values larger than 0.2 would be set to 0.2 and then the whole histogram is normalized again. After such a process, the nonzero dimensions of the histogram are still nonzero, but their values are more uniform. In this way, the feature description emphasizes more on the direction of the gradient but less on the magnitude of the gradient. Note that the direction is indicated by the existence of nonzero dimensions. However, in extreme situation some histograms may have less than 25 dimensions nonzero. As a result, some dimension(s) must be larger than 0.2 in order to maintain a unit vector. I

\(^1\)downloaded from http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries
Figure 3.3: Hessian-affine region detection: the top is the original image and the bottom shows the region detected based on the grey value image. Each yellow ellipse indicates a Hessian-Affine region.
Figure 3.4: Some Hessian-affine regions with ideal size: similar patches can be found among them and they are extracted from the different copies of identical objects in the image.

simply abandon these histograms.

To realize the rotation invariance, the dominating direction of each local region is defined as the direction along which the grey value change is strongest. 36 directions over 360 degrees are checked to identify such a direction. For each local region multiple dominating directions might be identified in case the change of the grey value on other directions is larger than 80% of the maximum one. Thus more than one SIFT descriptors might be generated from such a local region.

On average, about 20,000 SIFT descriptors are extracted from every database image, which finally results in 51,537,169 SIFT features describing all the database images.

3.3 Building Visual Vocabulary

Having realized the efficiency of local invariant features in object recognition, some research work started describing an image with a bag of local invariant features. Such a description divides image matching into two sub-tasks:

1. identifying the true matches between local regions based on their descriptors;

2. identifying the true matches between images based on their bags of descriptors.
For the first sub-task, namely the region level matching, one simplest solution would be to compute the distances between these descriptors. Only when two region descriptors bear a distance less than the preset threshold, can they be identified as a true match. However, computing the distances among a large number of high dimensional descriptors is very time consuming. Moreover, a heuristically set threshold may accept many false matches and wrongly filter out a lot of true matches.

For the second sub-task, namely the image level matching, a voting algorithm has been proposed in [Schmid and Mohr 1997]. In this algorithm, the number of matched descriptors between a query and each database image is counted to rank the database images. The Earth Mover's Distance (EMD) has also been adopted as an image similarity measure in [Rubner et al. 2000]. However both methods are quite time consuming. During voting, for each descriptor extracted from the query image, a nearest neighbour search has to be carried out to find the best match from all the descriptors in the database. The computation of an EMD is virtually an optimization procedure, which cannot efficiently deal with the images having a large set of descriptors.

As mentioned in Chapter 2.2, visual word and visual vocabulary are the outcome of analogizing content-based image retrieval to text retrieval. They are adopted as the major retrieval mechanism in this thesis. As we will see in the following section, this mechanism gives a more efficient solution to both the region level and image level matching.

3.3.1 Visual Vocabulary Generation

A visual vocabulary is composed of visual words, which are generated through clustering the region descriptors extracted from the images in a database. One simple but efficient clustering method is the k-means algorithm:

1. initialize the $k$ cluster centers by randomly sampling $k$ points from the data set being clustered;

2. with respect to a predefined distance, assign each point in the data set to its nearest cluster center;

3. recompute each of the $k$ cluster centers by calculating the mean of all points that have been assigned to this cluster;

4. if the new set of $k$ cluster centers are significantly different (with respect to a predefined threshold) from the current ones, replace the latter with the former and go to step 2; otherwise the clustering algorithm ends.
As we can see, the procedure of clustering discovers the areas in a feature space where the data densely distribute. The data points in each of these areas are identified as being sufficiently similar to each other.

Applying the $k$-means clustering algorithm to all the region descriptors from an image database computes the distances among the descriptors and assigns different descriptors into different clusters. Since all the descriptors falling in the same cluster are highly similar to each other, they can be generalized by one 'visual word'. In this way, $k$ visual words are obtained in total, which give a visual vocabulary. During retrieval, the two descriptors related to the same visual word would be identified as a match with no need of further computation. This solves the problem of region level matching and avoids performing the runtime nearest neighbour search during retrieval. Implementing the above clustering process also brings an advantage on data storage. That is, rather than a 128-dimensional histogram, it is the identity number of the associated visual word that needs to be stored for each descriptor.

One major obstacle that lies in the visual word generation is that running the $k$-means algorithm with a large data set is extremely time consuming, especially when the value of $k$ is also large. Therefore, in [Sivic and Zisserman 2003] only a subset of all the region descriptors are used to generate the visual words. A more efficient way is used in [Nister and Stewenius 2006], where a hierarchical $k$-means clustering is employed to cluster a large number of descriptors to one million clusters.

Another issue that worths concern is that the dimension of the SIFT descriptor is as high as 128, and this results in heavy computational load in calculating the Euclidean distance between them. Let $p_i$ be the $i$th descriptor and $c_j$ be the $j$th cluster center, respectively. The Euclidean distance between $p_i$ and $c_j$ is expressed as

$$Eud(p_i, c_j) = \sqrt{\|p_i - c_j\|^2} \quad (3.1)$$

Since each SIFT descriptor has been normalized to a unit vector (in order to deal with the illumination change), the Euclidean distance can be calculated as below:

$$Eud(p_i, c_j) = \sqrt{\|p_i\|^2 + \|c_j\|^2 - 2\langle p_i, c_j \rangle} = \sqrt{1 + \|c_j\|^2 - 2\langle p_i, c_j \rangle} .$$

Since the Euclidean distance is used to rank the $k$ cluster centers, $c_1, \ldots, c_k$, with respect to their distances from $p_i$, it will not affect the ranking by removing the square root operator and the
constant '1'. Thus, the following can be computed instead:

\[ \text{Eud}'(p_i, c_j) = \|c_j\|^2 - 2\langle p_i, c_j \rangle. \]  (3.2)

Moreover, the \( k \) cluster centers, \( c_1, ..., c_k \), are constant at each iteration. This means that their norms only need to be calculated once and stored in memory for use. Thus, the original Euclidean distance computation is transformed into an inner product which is more computationally efficient.

In stead of generating a vocabulary from a subset of all region descriptors, the visual vocabulary used in the work of this thesis is built from all the local region descriptors extracted from all the database images. With a branch factor 100, all the 51,537,169 SIFT descriptors, each of which is 128 dimensional, are finally clustered into 1,000,000 clusters over three levels.

### 3.3.2 Image Description with Visual Vocabulary

Recall that each image has been described with a bag of region descriptors. After the hierarchical clustering, each descriptor will be labelled by a visual word. Hence, for an image in a database, the ‘bag of descriptors’ based description can be easily transformed to a histogram.

---

**Figure 3.5:** Hierarchically kmeans: in this graph, the \( k \) is 100. After clustering in three levels, the total number of the final clusters is 1,000,000.
Each bin of this histogram records the number of occurrences of each visual word in this image. For a query image, its region descriptors do not participate the hierarchical clustering. In this case, each of its descriptors will also be assigned to the nearest visual word but in runtime, and a histogram is obtained in a similar way. Such a transform from ‘bag of descriptors’ to a histogram significantly simplifies the image description, as discussed below.

Describing an image with a bag of region descriptors is virtually describing the image with a set of high dimensional vectors. They can be viewed as a $m \times n$ matrix, in which $m$ is the number of descriptors extracted from the image and $n$ is the dimension of the descriptor which is 128 for the SIFT. Such a description is not convenient because i) comparing two images now involves comparing two high dimensional matrices, which is computationally expensive and ii) the value of $m$ varies in different images which leads to the comparison of matrices of different sizes. Nevertheless, after associating a descriptor with a visual word, this high dimensional descriptor will be replaced with an integer, namely the identity number of this visual word. Hence the image description is transformed from a set of vectors into a histogram. In this way, comparing two matrices of different sizes becomes the comparison of two histograms with the same length.

The usage of a visual vocabulary not only gives a more compact description to images but also leads to a more efficient implementation of image retrieval. It is known that evaluating the similarity between two images involves the region level and image level matching. The previous work, for example, the voting algorithm in [Schmid and Mohr 1997], carries out...
both the region level matching and the image level matching during runtime. However, by using the visual vocabulary mechanism, the region level matching is accomplished during off-line. The runtime retrieval only needs to handle the image level matching. Such a division significantly boosts the retrieval speed.
Object Retrieval in a Supermarket

As mentioned in previous chapters, the image database used in this object retrieval problem has the following characteristics:

1. **large scale difference**: the scales of the objects appearing in the query images and the database images bear significant difference;

2. **strong background clutters**: the object to be retrieved only occupies a minor area in each database image with many more irrelevant objects as the background clutters;

3. **multiple copies of identical objects**: the objects in every database image, no matter those to be retrieved or those being the background clutters, usually have multiple identical copies.

These characteristics make the image retrieval methods that work well in exist retrieval systems not completely suitable. To achieve accurate object retrieval in a supermarket, the system prototype presented in Figure 2.1 is refined as that in Figure 4.1. Two more components have been added, namely the Rescale of the query image and the Result combination of the retrieval outputs under different image scales. The two components are created to handle the large scale difference between the query images and the database images. Also, a new similarity measure is proposed to deal with the strong background clutters and the existence of the multiple copies. Based on this similarity measure, different weighting schemes are discussed. This chapter gives a thorough analysis to the new added components and the new similarity measure, explaining how they handle the above situations and why they outperform the previous methods. To make a clear statement, the object retrieval is first studied by assuming that the query images and the database images are under the same scale. After that, the adverse affect caused by the large scale difference will be discussed. A multi-scale retrieval mechanism would then be covered and the two new added components will be introduced. At last, a new visual word based spatial check approach is discussed.
Figure 4.1: The prototype of the system modified for the object retrieval in a supermarket
4.1 Similarity Measure with Strong Background Clutters

Assuming that the query images and the database images share the same scale, the performance of the object retrieval in a supermarket is mainly determined by the image similarity measure. This measure must be able to handle the strong background clutters encountered in this retrieval problem.

In the literature, the similarity measure widely used in visual word based image retrieval relies on the distances between the histograms associated to images [Sivic and Zisserman 2003; Nistér and Stewenius 2006]. These histograms describe the number of occurrences of the visual words in images. In [Sivic and Zisserman 2003], the two histograms to be compared are normalized to unit vectors, and the angle between them is calculated. Essentially, it is the Euclidean distance between the two normalized histograms that is used to reflect the similarity between two images. In [Nistér and Stewenius 2006], the $L_p$-norm distance is calculated between the two normalized histograms as the similarity measure. Their experimental result shows that the best retrieval performance is achieved when $p = 1$.

One interesting point of the above two similarity measures is that both of them really calculates the difference instead of the commonness shared by two images. The computation of distance accumulates the differences on each dimension of the two vectors. Under certain conditions, measuring the difference is equivalent to measuring the commonness. In [Nistér and Stewenius 2006], the images to be retrieved are the photos each of which exactly contains a single CD cover. Therefore, two photos with the smallest difference are surely those with the most commonness. This also applies to the retrieval problem in [Sivic and Zisserman 2003] in case that the video frames capturing similar scenes are compared to each other. However, this project namely the object retrieval in a supermarket is different in nature. As demonstrated by Figure 4.2, finding the image having smaller difference from the query image can hardly identify the true match. Such a result is caused by the existence of strong background clutters. The following analyzes this problem in a more principled way:

- Let us assume the case in Figure 4.2 as follows. The images are described with a visual vocabulary containing 10 visual words. The image on the left in Figure 4.2 is described with $h_q = [1, 1, 1, 1, 0, 0, 0, 0, 0, 0]$ and the one on the right is described by a vector $h_1 = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$. The six extra local descriptors are extracted from the background clutters. There is another image described by $h_2 = [0, 1, 1, 1, 0, 1, 0, 0, 0, 0]$. This image does not contain the query object but shares 3 identical region descriptors with the query image $h_q$. Meanwhile, it has fewer descriptors from the background clutters than the image $h_1$ does.
Figure 4.2: The demonstration of *difference* and *commonness* in the object retrieval in a supermarket: the image on the left is the query. The image on the right is one of the database images that should be retrieved, because it contains the 'Spree' detergent which is the *commonness*. Apparently, there is much more *difference* between these two images. If judging based on the difference, the similarity between images will be dominated by the background clutters, resulting in poor retrieval performance.

- with the similarity measure in [Sivic and Zisserman 2003], the cosine values of the angles among $h_q$, $h_1$ and $h_2$ are:

  \[
  \cos(h_q, h_1) = 0.6325 \\
  \cos(h_q, h_2) = 0.6708 .
  \]

Since the larger cosine value denotes the smaller angle, the angle between $h_q$ and $h_2$ is smaller. Thus they are more similar according to the similarity measure in [Sivic and Zisserman 2003]. Apparently, this is not correct.

- with the similarity measure in [Nistér and Stewenius 2006], the $L_1$-norm distance between among $h_q$, $h_1$ and $h_2$ are:

  \[
  \frac{||h_q - h_1||_1}{||h_q||_1} = 1.2 \\
  \frac{||h_q - h_2||_1}{||h_q||_1} = 0.8 ,
  \]

and the $L_2$-norm distance between the vectors are:

\[
\frac{||h_q - h_1||_2}{||h_q||_2} = 0.8574
\]
§4.1 Similarity Measure with Strong Background Clutters

\[ \frac{\| \mathbf{h}_q \|_2}{\| \mathbf{h}_q \|_2} - \frac{\| \mathbf{h}_2 \|_2}{\| \mathbf{h}_2 \|_2} = 0.8114 , \]

In both ways, the distances between \( \mathbf{h}_q \) and \( \mathbf{h}_2 \) are smaller than those between \( \mathbf{h}_q \) and \( \mathbf{h}_1 \). This result suggests that \( \mathbf{h}_q \) and \( \mathbf{h}_2 \) are more similar to each other. Again, this is not correct.

The reason for the above results is that although the image \( \mathbf{h}_q \) can be completely matched to a part of the image \( \mathbf{h}_1 \), there is still much more content in \( \mathbf{h}_1 \) which cannot find any match in \( \mathbf{h}_q \). This creates a large difference between the two images, and the distance based similarity measure penalizes such a difference. To measure the similarity between images in case of strong background clutters, this thesis proposes to use a measure which emphasizes the commonness between them and can be less affected by the difference caused by the descriptors extracted from the background clutters. This measure, denoted by \( S \), computes the inner product between the vectors without being normalized:

\[ S = \langle \mathbf{h}_q, \mathbf{h}_1 \rangle = \sum_j h_{qj} h_{ij} \quad (4.1) \]

Intuitively the calculation of an inner product is the procedure of accumulating the nonzero items shared by two vectors, namely the commonness. The difference that on some dimensions only one of the vectors has nonzero items is ignored. Applying this measure to the case that is assumed for the images in Figure 4.2 obtains

\[ S_{(\mathbf{h}_q, \mathbf{h}_1)} = \langle \mathbf{h}_q, \mathbf{h}_1 \rangle = 4 , \]
\[ S_{(\mathbf{h}_q, \mathbf{h}_2)} = \langle \mathbf{h}_q, \mathbf{h}_2 \rangle = 3 . \]

As they show, the similarity between \( \mathbf{h}_q \) and \( \mathbf{h}_2 \) is now smaller than that between \( \mathbf{h}_q \) and \( \mathbf{h}_1 \), which gives a correct retrieval result.

There is still a room to further improve the measure \( S \). As shown in Equation 4.1, beside the number of shared nonzero items, the value of each dimension in \( \mathbf{h}_q \) and \( \mathbf{h}_i \) also contributes to the magnitude of \( S \). As a result, the case of having a large value along a single dimension can easily give a larger inner product than that of having small values along multiple dimensions does. The following provides an example:

\[ S_{(\mathbf{h}_q, \mathbf{h}_1)} = \langle (1,1,1,0,0,0,0,0,0,0)^T, (3,5,0,0,0,0,0,0,0,0)^T \rangle = 8 \]
\[ S_{(\mathbf{h}_q, \mathbf{h}_2)} = \langle (1,1,1,0,0,0,0,0,0,0)^T, (1,1,1,0,0,0,0,0,0,0)^T \rangle = 4 \]
Although $h_q$ and $h_2$ are identical, $S(h_q, h_2)$ wrongly gives a lower score. The case of having a large value along a single dimension is resulted by the existence of multiple copies of an identical object, a characteristic of the object retrieval in a supermarket. An object with multiple copies can be either the object to be retrieved or an irrelevant object which is actually a part of the background clutters. Hence, the quantity information of each dimension in $h_q$ and $h_i$ is not reliable to be used for the object retrieval. In this work, such information is discarded by converting each dimension in $h_q$ and $h_i$ into a binary form, as indicated in Equation 4.2.

$$S_1 = (h'_q, h'_i) = \sum_j h'_{qj} h'_{ij}$$  \hspace{1cm} (4.2)

where

$$h'_{qj} = \begin{cases} 0, & \text{if } h_{qj} = 0 \\ 1, & \text{if } h_{qj} > 0 \end{cases} \hspace{1cm} h'_{ij} = \begin{cases} 0, & \text{if } h_{ij} = 0 \\ 1, & \text{if } h_{ij} > 0 \end{cases}$$

The advantage of the above $S_1$ over the $S$ in Equation 4.1 will be experimentally demonstrated in Chapter 5.1.

Besides the measure proposed above, another effective way to reduce the affect of background clutters is pre-segmentation. After an ideal pre-segmentation an image is partitioned into regions each of which contains only one object and the background clutters are removed. However, since segmentation itself is a difficult problem, till now no algorithm in the literature could ever implement segmentations at an ideal level or even near the ideal level to general images. Nevertheless, a loose pre-segmentation in which multiple objects may coexist in a single region has been able to be help. The simplest implementation of a loose segmentation is partition. After partition, an image is divided into a set of regions regardless of the image content. When the size of the objects in an image can be roughly estimated, a partition can easily shield many background clutters without over partitioning the objects therein.

In this project, each image contains dozens of objects lying on about three levels of a shelve. Generally the size of each single object is no larger than one ninth of the size of the image and most objects are considerably smaller than such a size. Hence if we evenly partition the whole image into 9 ($3 \times 3$) sub-images, each of them will be big enough to contain an individual object with much less background clutters. A concern would be that some objects are divided into multiple sub-images after this partition. A possible solution is to partition an image into more sub-images which partially overlap each other. However, such a solution is not necessary because local features are quite robust to identify partial matching images. Moreover, adding overlapping sub-images would create a large amount of redundant data.
since an identical feature would be extracted multiple times from the overlapping sub-images. The affect caused by the above partition to a database image is that its region descriptors are now divided into 9 subsets which are independent from each other. In this project, the rank of this database image is determined by the highest rank of its sub-images.

4.2 Weight of Visual Words

In text retrieval, there are words like *if, but and and* which appear in most if not all files and tell little about the content. There also exist words like *sugar, coffee and football* that only appear in certain files so that more information about the file can be derived from them. As we can see the words enumerated above contribute differently in distinguishing documents. To quantize such a difference during retrieval, weighting schemes are utilized to assign different weights to different words according to its importance in distinguishing documents. Term frequency-inverse document frequency (tf-idf) weighting scheme is used in [Sivic and Zisserman 2003]:

\[
    t_{ij} = \frac{n_{ij}}{n_i} \log \frac{N}{n_j}.
\]

(4.3)

This scheme quantizes the importance of each visual word according to its appearance frequency in both the whole database and the current image. Readers are referred to Chapter 2.3.1 for more detailed discussion about this weighting scheme. Slight modification was made to this weighting method in [Nistér and Stewenius 2006]:

\[
    t'_{ij} = n_{ij} \log \frac{N}{N_j}.
\]

(4.4)

As discussed in the Chapter 2.3.1, the only difference between Equation 4.3 and Equation 4.4 is caused by the alternate between \(n_j\) and \(N_j\) which causes minor disparity in the retrieval performance in [Nistér and Stewenius 2006]. However, the work in this thesis discovered that the disparity on retrieval performance increases due to the existence of multiple identical objects in a single database image. The reason is that in an image that does not contain multiple identical objects, a visual words bears a quite low possibility of appearing more than once.

Therefore, \(n_j\), the number of appearances of the \(j\)th visual word in the whole database, and \(N_j\), the number of database images that contain at least one copy of the \(j\)th visual word, are almost the same. However, in the images where an identical object appears more than once, the relationship between \(n_j\) and \(N_j\) becomes unpredictable and usually the value of \(n_j\) is much larger than that of \(N_j\). It is reasonable to assign a low weight to a visual word if it
can be extracted from many different objects. That is because such a visual word cannot help distinguish those objects. However, the weight of a visual word should not decrease if it can be extracted from multiple copies of an identical object. That is because such a condition does not suggests a low discriminative power of this visual word. As their definition indicates, the value of \( N_j \) is invariant whereas the value of \( n_j \) increases if there are multiple copies of an identical object in an image. Hence instead of using \( n_j \) which will wrongly assign a lower weight to the visual word, \( N_j \) is used in the this thesis. Experimental study on the difference between \( n_j \) and \( N_j \) will be presented in Chapter 5.2. Moreover, since the quantity information of each dimension in the histogram used to represent an image has been discarded in this object retrieval problem, the weighting scheme in Equation 4.4 is rewritten as

\[
t'_j = \log \frac{N}{N_j}.
\] (4.5)

By using this weighting scheme, Equation 4.2 is converted to

\[
S_1 = \sum_j (h_{ij}' t_j')(h_{ij}' t_j') .
\] (4.6)

where \( h_{ij}' \) and \( h_{ij}' \) take the binary value of 0 or 1.

### 4.3 Image Indexing with an Invert File

Compared to a bag of local feature descriptors, a bag of visual words is much more computationally efficient in image similarity comparison. However, to handle image retrieval in a large image database an indexing mechanism is still necessary for the purpose of retrieval speed. The usage of visual word makes it possible to utilize the well studied indexing methods in text retrieval, for example, the invert file. Successful applications of an invert file in image retrieval can be seen in [Nister and Stewenius 2006; Sivic and Zisserman 2003]. Especially in [Nistér and Stewenius 2006] image retrieval is implemented in realtime in a database containing 40,000 CD cover images.

For each visual word, an invert file records all the images that contain it. In this way, the database images sharing a same word are collected into a same set. It is called invert file because the relationship between words and documents is presented in a word-to-document way instead of the document-to-word in an ordinary file. In other words, the ordinary file describes a document with its words, whereas the invert file describes each word with all documents containing it.
After using the invert file, the whole process of image retrieval actually involves the collaboration of the ordinary file and the invert file. As the query image is described with an ordinary file, each query image is described with a bag of visual words. On the other hand, each visual word is described with a bag of database images in the invert file. Hence the query image can be further described with a bag of database images. The similarity between the query image and different database images can be found in such a description. Such a manner leads to a high retrieval speed because only those database images which appear in the above description need to be ranked during retrieval.

4.4 Stop List

In text documents, words can be roughly categorized into three classes according to their appearance frequency in all documents, namely appearing frequently, appearing rarely and the medium. The reason of such a categorization is that words having different appearing frequencies contribute differently during text retrieval:

Appearing frequently: since they are included in so many documents, the presence of a word of this type can hardly indicate any association between documents. Hence they are not helpful during retrieval. Examples include the, a and and.

Appearing rarely: words of this type can help distinguishing a document from the others, however, since so few documents contain them they are rarely involved in retrieval. Hence they are not helpful in retrieval either.

The medium: words of this type help most during retrieval among the three categories. Examples include music, mathematic and history. They are not rare but not all files contain them.

Since the words appearing frequently and those appearing rarely do not contribute much to retrieval, ignoring them would not decline the retrieval performance much. By using a stop list, these words would be removed from the vocabulary. The advantage of using a stop list lies in two aspects:

1. reduce the false matches: visual words appearing too frequently represent those patterns that are easy to be duplicated and thus bear a high risk of mismatch. By putting them into the stop list, the number of false matches can be effectively reduced.

2. decrease the size of the invert file: the size of an invert file is determined by the total number of visual words detected from all images. Removing the less useful visual
words can effectively reduce the size of the invert file. This in turn saves the memory storage and will increase the retrieval speed.

4.5 Image Retrieval under Multiple Scales

The Harris-Affine detector is developed from the scale invariant Harris-Laplacian detector and the Hessian-Affine detector shares the same scale invariant mechanism with the Harris-Affine detector. Hence the regions detected by either of them can keep stable as the scale of an image varies. However, as introduced in Chapter 2.1.2, the region’s repeatability decreases as the scale difference increases. More than that, more descriptors can be extracted from the images of larger scales than from those of smaller scales. For the object retrieval in this work, the scale of an object is significantly different between a query image and the database images, and its scale in a query image is larger. During image retrieval, an object in a query image often produces many local descriptors that cannot be extracted from an identical object in a database image. This will result in many false matches and declines the retrieval performance.

This section proposes to solve the problem of significant scale difference by way of image retrieval under multiple scales. To compare a query image and the database images under similar scales, the query image has to be downsized. However, to which size cannot be known in advance. Therefore, this work downsizes the query image to multiple scales and compares it with the database images at each scale level. The similarity measures obtained in the multiple levels are then combined as a final score to rank the database images.

4.5.1 Image Description under Multiple Scales

It is not necessary to describe both the query image and the database image but one of them under multiple scales, as long as the scale of the other is covered by the multiple scales. A straightforward way may be to describe each database image under multiple scales during offline. However, using a multiple scale description for all the database images would enlarge the size of the invert file of the database by multiple times. If the invert file is too big to be contained in physical memory, the retrieval speed would decline significantly. Hence the query image is described under multiple scales instead. Since there is only one object in each query image, the local feature extraction in a query image is relatively fast. Especially, the speed of the local feature extraction can be faster as the query image being downsized.

Observing that each object appearing in the query image is of a much larger scale than that in the database images, each original query image is sequentially down scaled to 10 different
version (including the original version) using a step factor $\sigma = 1.25$. Labelled as 0th to 9th, the 9th scale is $9.3^{-1}$ times of the 0th (the original scale) so that a sufficiently large scale range has been covered. The scale step is small enough so that the local region detectors can efficiently handle the small scale difference between two adjacent scales.

Two steps constitute the multi-scale retrieval:

1. create the 10 versions of the query image as 10 individual query images and run retrieval separately with each of the 10 query images;

2. generate the final rank of each database image based on the retrieval results obtained with each of the 10 query images.

The second step brings us a multi-expert decision problem in which a decision is not dictated by a single expert but reached by cooperation between multiple experts. In our case, the retrieval results gained by the query image under 10 different scales are the individual decisions of 10 different experts. To generate the final decision, two major principles could be followed [Duda et al. 2000]:

1. **Synthetical principle**: the final class of each input pattern is determined by combining all the results gained from all experts. Following such a principle the final decision can be made according to the average score or average rank the input pattern gained from different experts.

2. **Winner-take-all principle**: the final class of each input pattern is determined solely by the expert with the highest confidence. Following such a principle the final decision can be made according to the expert who assigns the highest score with his conclusion among all experts.

However, the experiments in this thesis show that simply applying either principle does
not give satisfying retrieval results. This thesis proposes two mechanisms each of which can produce a better combination result than that achieved by simply using above principles.

4.5.2 Combination by Similarity Score

After retrieval under multiple scales, each database image would be assigned a similarity score by the similarity measure, such as that shown in Equation 4.6. This score shows the similarity between the database image and the query image. Intuitively, the final retrieval result can be produced by ranking the database images according to their highest score over different scales which is a practice of the winner-take-all principle. However, since the number of local features extracted from each image is positively correlated to the scale of the image, it is easier for a query image which is under a larger scale to assign a high similarity score to database images. Hence the final retrieval result is usually dominated by the similarity score between a database image and the query image under the largest scale. On the other hand, the synthetical principle does not work well either. That is because the similarity score under the largest scale is relatively so high that the largest scale becomes the dominating scale. As long as an irrelevant database image gets an extremely high score at a dominating scale, the relevant images cannot beat it by accumulating the scores from all scales. To avoid the occurrence of a dominating scale, the score of each database image has to be down scaled according to the number of the visual words contained in the query image. Hence Equation 4.6 is modified to Equation 4.7:

$$S_l = \frac{\sum_j (h_{q_i}^t f_j^t)(h_{i}^t f_j^t)}{\sum_j (h_{q_i}^t f_j^t)(h_{q_i}^t f_j^t)}$$.

Let $l$ denotes the one of the scales involved in the multiple scales retrieval, the final similarity score of each database image is determined by the sum of the similarity scores gained with Equation 4.7 under all scales:

$$S_1 = \sum l s_l$$

As we can see, in Equation 4.7 the original similarity score between the query image and the database image is divided by the similarity score between the query image and itself. The denominator shows the similarity score obtained by an image which is identical to the query image. Equation 4.7 virtually computes the percentage of a query image that is matched to a database image. This manner makes the similarity scores between a database image and the query images of different scales comparable to each other and a combination result of higher accuracy can be achieved. Relevant experiment can be seen in Chapter 5.5.1.
4.5.3 Combination by Rank

Another way to combine the results gained from different scales is to combine the ranks a database image gets under different scales. One advantage possessed by such a manner is that the dominating scale confronted in the similarity score based combination is naturally avoided. The winner-take-all principle is not practicable because we cannot tell which 'Top1' is the best among all the scales. The synthetical principle does not perform well either. That is because when combining ranks, the low rank which appears as a big number will dominate the final rank. Hence straightforwardly combining the ranks from all scales would result in a poor retrieval performance. Since a relevant database image would gain high ranks at sequential scales, it is reasonable to determine a database image's final rank by combining its rank from a number (denoted as $r$) of sequential scales under which it achieves higher rank than in the other scales. The choice of $r$ needs to be careful. A too large $r$ will bring too many scales into final decision and some of the scales give dominating low ranks. On the other hand, a too small $r$ considers too few scales which are not sufficient for a relevant database image to demonstrate its advantage over sequential scales. To pick the proper $r$, a thorough experimental study will be presented in Chapter 5.5.2. The dependency of the retrieval performance on the selection of $r$ and $\sigma$ (the step factor of scale) will also be discussed.

4.6 Visual Word Based Spatial Check

The spatial information within the image content is a distinctive feature in image description. The spatial information has been utilized to implement object recognition in some early work [Forsyth and Fleck 1999; Smith and Chang 1999]. However, it is quite difficult to design a spatial information extraction algorithm that can handle objects of wide domain with unpredictable visual appearances. Thus the spatial information based object recognition is limited to objects of narrow domain. For example in [Forsyth and Fleck 1999], the spatial information was utilized in human body detection. In [Smith and Chang 1999], the extraction of spatial information is accomplished by user's sketching but such a sketch cannot describe complex spatial information and is strongly user dependent.

The proposition of local invariant features gives a good jump-off point to spatial information extraction. The spatial information within an image can be measured from the layout of local features. However, a precise description of the spatial information is not necessary since the attributes like lengths and angles can all be changed by a simple scale or viewpoint variance. The estimation of the affine transform with a large point set including many outliers
is too heavy a computation for image retrieval and sometimes even results in errors. Since local features can achieve promising image retrieval performance solely, spatial information is usually assessed loosely as a complement in the post-verification step. In [Lowe 2004; Sivic and Zisserman 2003; Schmid and Mohr 1997; Zhang et al. 1995], local feature based spatial information check has been successfully applied to images which do not include repetitive patterns. Under the hypothesis of no repetitive patterns, a constraint can be imposed as that the layout of a set of corresponding regions in the second image should loosely accord with that of the first image, such as the angles, the distances and the p-nearest neighbours among local features. However, the repetition of patterns is quite common in some images, as shown in Figure 4.4. The one-to-multiple matches caused by these repetitive patterns make it difficult to figure out the natural layout of local features, especially when there is a large scale difference simultaneously. Unfortunately, both the repetitive patterns and the large scale difference exist in the object retrieval in a supermarket. Besides, since visual words do not contain any spatial information in its description, checking the spatial information among local features after retrieving the images with visual words requires describing each image not only with visual words but also with an additional file recording the spatial information of each local feature within the image.

One ideal scenario would be combining the spatial information of the image with the vi-
§4.6 Visual Word Based Spatial Check

It contains. One simple way would be to extend the description of each visual word in an image from a single word identity number into a 3-dimension vector \((id, x, y)\), where \(x\) and \(y\) are the position coordinates of this visual word in this image. With these position coordinates, each visual word can find its \(p\)-nearest neighbours (the \(p\) nearest visual words according to the position on image). The validity of the match between a pair of visual words can then be verified by the matches between their \(p\)-nearest neighbours [Sivic and Zisserman 2003; Schmid and Mohr 1997]. However, finding \(p\)-nearest neighbours during online is a time consuming process especially for an image containing a large number of visual words. On the other hand, finding the \(p\)-nearest neighbours for each visual word during off-line causes a huge consumption in storage. Assuming the 15-nearest neighbours for each visual word are recorded, then the description of each database image would be 16 times of the original in size. Besides, the \(p\)-nearest neighbours constraint does not get along well with a large scale difference between the query image and the database image. That is because as the scale of image increases, more local features will be generated. Some of the \(p\)-nearest neighbours of a local feature under a larger scale might fail to be extracted under a smaller scale. Due to the change in the \(p\)-nearest neighbours, some true matches may be filtered out during spatial check, as illustrated in Figure 4.5.

In this work, a visual word based spatial check method is proposed. This method assesses the spatial information which is stable with the existence of multiple identical copies of an object. Moreover, such spatial information is combined with each visual word and is easy to be compared between different visual words. The orientation information and the area information of local regions are assessed in this visual word based spatial check. The following demonstrates its effectiveness by using an example from the object retrieval in a supermarket.

Figure 4.6 shows the effect of the visual word based spatial check in eliminating false matches. Among the three images, the top one shows the matches between the query and the database image without spatial check. Each ellipse crops out a local region detected by the Hessian-Affine detector and each straight line connects a pair of matched regions. Without spatial check, out of the 93 pairs of matched regions that have been detected, only 12 of them are correct matches. The middle image shows the matches that survive the orientation check. The orientation of each local region is defined as the direction of the major axis of the ellipse. According to the change in region’s orientation between the query image and the database image, the matches between the two images have been classified into three groups which include: i) rotating more then 30° clockwise; ii) rotating more than 30° anticlockwise; iii) rotating no more than 30° in both direction. Among the three, the group possesses the largest number of matches will be identified as true matches. After orientation check, 38 pairs of
Figure 4.5: The image on the top left shows the 609 pairs of matched regions between the query of a large scale and a database image. After the $p$-nearest neighbour check, only 14 pairs of matched regions are left, as shown in the image on the top right. Here the $p$ is set as 15, and as long as there is another matched pair existing within the 15 nearest neighbours of the checked matching, the latter will be considered as a true match. On the other hand, the image on the bottom left and bottom right shows the matches before and after the spatial check between a down scaled query and the database image. 499 matches exist in the bottom left and 48 matches remain in the bottom left. The results of 14/609 and 48/499 show that the $p$-nearest neighbour constraint kills many true matches by mistake when a large scale difference exists.
matching regions have been left and among them 11 pairs are true matches. The bottom image shows the matches that pass both the orientation check and the area check. According to the ratio between the areas of the two matched regions, all matches are classified into 4 groups, namely the scale decreasing less than 4 times, between 4 and 16 times, between 16 and 64 times and more than 64 times. Combining them with the 3 groups classified during the orientation check, all matches are now distributed into $3 \times 4 = 12$ groups. The group composed of the most matches is regarded as true matches. As shown in the bottom image, 10 matches remain and all of them are correct.

The visual word based spatial check shows its value in Figure 4.6. Besides removing false matches, it adds few workload onto the computation. However, as we may have noticed, the number of the true matches between two images does not obviously surpass that of the false matches. That is because of the strong background clutters in the database images. Sometimes, the number of false matches can even fairly exceeds that of true matches, and the true matches are at risk to be filtered out in the spatial check. For that reason, till now the final object retrieval performance has not benefited much from the visual word based spatial check. In view of its immaturity, the detailed experiment data of the visual word based spatial check is not included in this thesis. However, it is believed that with more sophisticated criteria this mechanism can profit more to image retrieval in the future.
Figure 4.6: Visual word based spatial check: please note that on the shelf only the left two boxes are true matches to the query, and the two boxes next to them are similar but not identical objects.
Chapter 5

Experiment

To compare the efficiency of different retrieval methods, two criteria are used in this thesis. They are ALOT curve and Precision-Recall curve:

1. **ALOT**: for the top \( n \) retrieved images, their ALOT shows the percentage of the query images that can find at least one true match (ALOT). This criterion is defined as:

\[
ALOT = \frac{\text{the number of queries that find at least one true match}}{\text{the total number of the queries}}.
\]  
(5.1)

The value of ALOT grows as \( n \) increases, which generates an ALOT curve.

2. **Precision-Recall**: such a curve is generated from two sets of data, namely the Precision and the Recall. For the top \( n \) retrieved images, their Precision shows the percentage of the retrieved images that are true matches. It is defined as:

\[
Precision = \frac{\text{the number of retrieved true matchings}}{\text{the number of retrieved images}}.
\]  
(5.2)

The Recall shows the percentage of all the true match images that have been retrieved, among the top \( n \) retrieved images.

\[
Recall = \frac{\text{the number of retrieved true matchings}}{\text{the number of all the truth matchings}}
\]  
(5.3)

As \( n \) varies, a set of Precision and Recall would be generated. Using the Recall as the \( x \) coordinates and the Precision as the \( y \) coordinates, a Precision-Recall curve can show the retrieval performance by displaying the Precision rates as the Recall increases.

The aim of this project is to find at least one shelf that contains the query object. Hence ALOT is chosen to be the main criterion that evaluates a retrieval method’s performance in this thesis.
In the following sections, three retrieval methods will be compared. They are the method proposed in this thesis, the method proposed in [Sivic and Zisserman 2003] and the method proposed in [Nistér and Stewenius 2006].

As discussed in Section 4.5, a larger scale difference between the query and the database images will result in a poorer retrieval performance. Thorough experiments and discussion related to the scale difference between images will be presented in Section 5.4. In the other sections of this chapter, all image retrieval methods will mainly compared under scale-6. That is because under scale-6 all the retrieval methods gain their own best performance. In other words, the affect of the scale difference is minimized, so that the affect of other factors like the weighting scheme and the partition mechanism could be clearly presented.

In the experiments, 102 different query objects appearing in 300 query images with random viewpoint, illumination and scale are used. These query objects include boxes, bottles and bags as illustrated in Section 3.1. 3,153 database images constitute the database. Each of the 300 query images will be retrieved against the database individually and the average retrieval performance is accessed. The groundtruth is created by manually checking each query against each database image. A database image is labelled as a true match to a query image, if and only if the former contains exactly the same object appearing in the latter. A database image containing Coca-Cola Zero is not a true match to the query image containing Coca-Cola Classic, and a bottle coke is not a true match to a can coke either. The query object may appear partly in some database images. These database images are labelled as true matches if they contain more than a half of the query object.

5.1 Performance of Similarity Measure

This section compares the retrieval performance achieved by different similarity measures, including 1) the numerical vector based inner-product proposed in this thesis and labelled as $S^0$; 2) the binary vector based inner-product proposed in this thesis and labelled as $S^1$; 3) the normalized vector based Euclidean distance proposed in [Sivic and Zisserman 2003] and labelled as $S^2$; 4) the normalized vector based $L_1$-norm distance proposed in [Nistér and Stewenius 2006] and labelled as $S^3$. In experiments of this section, images are not partitioned, and no weighting scheme is adopted either.

Figure 5.1 shows the ALOT curves gained by different similarity measures. From the ALOT curves it can be seen that, within the Top5 retrieved images, the binary vector based inner-product method $S^1$ achieves the best performance. Especially at the Top1 position, its rate is more than 10 points higher than those of the other three similarity measures. After the
Top5, the Euclidean distance method $S_2$ becomes better than the binary vector based inner-product method $S_1$. The $L_1$-norm distance method $S_3$ remains to be the worst among the Top20. Since people are often more interested in the Top5, especially the Top1 retrieved images, the binary vector based inner-product method $S_1$ is the best in this comparison. From the Precision-Recall curve in Figure 5.2 we can reach the same conclusion.

The experiment results in the two figures demonstrated that:

1. the binary vector based inner-product based method works better than distance based methods in dealing with images of strong background clutters. As stated in Chapter 4.1, the database images have much more complex visual contents than the query images do, which results in much more visual words detected from a database image than from a query image. As a result, even the two compared images share identical parts, their similarity will not be high due to the existence of the different parts which are penalized by the distance $S_2$ and $S_3$. On the other hand, the inner-product method $S_1$ is not sensitive to the different parts between the two compared images and hence performs better.

2. the binary vector based inner-product outperforms numerical vector based inner-product in dealing with images containing multiple copies of identical objects. Since each database image contains plenty of visual words, the existence of false matches is common. Furthermore, the existence of multiple copies of identical objects would increase the number of both true matches and false matches, but in an unpredictable manner. Often, they might cause the number of false matches finally exceeds that of true matches. That is why the numerical vector based inner-product method $S_0$ is inferior to the binary vector based methods $S_1$.

From the next section, the numerical vector based inner-product will no longer be considered, since it never outperforms the binary vector based inner-product in the experiments.

## 5.2 Usage of Weighting Scheme

This section investigates the different affects caused by different weighting schemes on the similarity measures $S_1$, $S_2$ and $S_3$. The similarity measure $S_2$ and $S_3$ proposed in [Sivic and Zisserman 2003] and [Nistér and Stewenius 2006] respectively will use the tf-idf weighting scheme. The similarity measure $S_1$ which is proposed in this thesis will use the weighting scheme in Equation 4.5 which is specially designed to handle the existence to strong back-
Figure 5.1: The ALOT curves of different similarity measures under scale-6, no weighting, no partition, no spatial check. $S_0$: numerical vector based inner-product; $S_1$: binary vector based inner-product; $S_2$: Euclidean distance between normalized vector; $S_3$: $L_1$-norm distance.
Figure 5.2: The Precision-Recall curves of different similarity measure under scale-6, no weighting, no partition, no spatial check. $S_0$: numerical vector based inner-product; $S_1$: binary vector based inner-product; $S_2$: Euclidean distance between normalized vector; $S_3$: $L_1$-norm distance.
ground clutters and multiple copies. Particularly, the retrieval performance will be compared between \( n_j \) and \( N_j \) as discussed in Chapter 4.2.

Figure 5.3 shows the ALOT curves. The affect caused by using the \( n_j \) based weighting schemes to the three similarity measures can be clearly seen. Compared to Figure 5.1, \( S_2 \) benefits much more from this weighting scheme than the other two. At the Top1 position, \( S_2 \) goes up to about 55% from less than 45%. The proposed similarity measure \( S_1 \) is improved a bit at the Top1 position but declined at the Top5 position and \( S_3 \) almost remains the same as it was in Figure 5.1. Similar conclusion can be drawn from Figure 5.4 which plots the Precision-Recall curves.

The situation for the three similarity methods in Figure 5.5 and Figure 5.6 where \( N_j \) is used is quite different from that in Figure 5.3. Although \( S_2 \) is still the one who benefits most, its increase is slightly reduced. On the other hand, the increase of the proposed similarity measure \( S_1 \) is larger than that in Figure 5.3. At the Top1 position, \( S_1 \) goes up to 60% from the 55% in Figure 5.1. Again, there is no improvement on \( S_3 \).

The 60% at the Top1 in the ALOT curve achieved by \( S_1 \) with \( N_j \) which is proposed in Equation 4.2 is the best result so far. The reason why different similarity measures response differently to \( n_j \) and \( N_j \) is explained as follows. For \( S_1 \), the appearances of each type of visual words in an image is recorded in a binary way. That is consistent with the term \( N_j \) to which the multiple appearances of one type of visual word in one image will also contribute one; whereas in \( S_2 \), the exact appearance numbers of each type of visual words in an image is recorded, which is consistent with the term \( n_j \). For term \( n_j \), the exact appearance number of each type of visual word in an image will be recorded. Nevertheless, in either way, the increase of \( S_3 \) is almost zero. Later we will see that, the major barrier for \( S_3 \) is the background clutters.

The alternation between \( n_j \) and \( N_j \) makes minor difference on the similarity measure \( S_2 \) and \( S_3 \), which is consistent with the observation in [Nistér and Stewenius 2006]. However, \( S_1 \) works better with \( N_j \) obviously and gives the best retrieval performance among all the similarity measures after weighting.

### 5.3 Usage of Partition

To mitigate the adverse affect of strong background clutters, the partition mechanism is adopted in this section. The weighting schemes applied to the three similarity measures are all \( N_j \) based. Each database image is partitioned and each sub-image will participate in the retrieval independently. The final similarity score of each database image is determined by the high-
Figure 5.3: The ALOT curves of different similarity measure under scale-6, weighted with \( n_j \), no partition, no spatial check. \( S_1 \): binary vector based inner-product; \( S_2 \): Euclidean distance between normalized vector; \( S_3 \): \( L_1 \)-norm distance.
Figure 5.4: The Precision-Recall curves of different similarity measure under scale-6, weighted with $n_j$, no partition, no spatial check. $S_1$: binary vector based inner-product; $S_2$: Euclidean distance between normalized vector; $S_3$: $L_1$-norm distance.
§5.3 Usage of Partition

The number of retrieved images

Figure 5.5: The ALOT curves of different similarity measure under scale-6, weighted with \(N_i\), no partition, no spatial check. \(S_1\): binary vector based inner-product; \(S_2\): Euclidean distance between normalized vector; \(S_3\): \(L_1\)-norm distance.
Figure 5.6: The Precision-Recall curves of different similarity measure under scale-6, weighted with $N_j$, no partition, no spatial check. $S_1$: binary vector based inner-product; $S_2$: Euclidean distance between normalized vector; $S_3$: $L_1$-norm distance.
est similarity score achieved by its sub-images. To test local invariant feature's capability in handling partial occlusion, image retrieval are carried out under two conditions:

1. **partition with overlapping**: each database image is firstly partitioned evenly into 9 (3 x 3) sub-images and another 16 sub-images of the same size are then sampled to cover the boundaries or corners between the 9 initial sub-images. In such a manner, each database image is virtually increased to \( \frac{25}{9} \) times in size. So is the number of local features extracted from the database images. The advantage of partition a database in this manner is that each object appears completely at least once in one of the 25 sub-images.

2. **partition without overlapping**: each database image is simply partitioned evenly into 3 x 3 sub-images. In this way, the number of features in the database keeps unchanged, but some objects are never displayed completely in any sub-images.

The retrieval performance in the case of partition with overlapping is shown in Figure 5.7 and Figure 5.8. The retrieval performances of all the three similarity measures have been improved compared to that in Figure 5.5 and Figure 5.6. Among them, S2 and S3 benefit from the partition mechanism more than S1 does. From the author's perspective, this actually shows that as the background clutters become severe the performance of S2 and S3 deteriorate more quickly than S1 does. In other words, S1 is more robust to background clutters than the other two. S3 gains the most improvement and achieves almost the same performance as S2 does at the Top1 position. This suggests that S3 is the most sensitive to background clutters.

The results for partition without overlapping are plotted in Figure 5.9 and Figure 5.10. The performance of all the three similarity measures decreases slightly but remains better than that in the retrieval without partition. This shows that the partial occlusion can adversely affect the retrieval performance but the local invariant features can effectively handle it. One interesting phenomenon happens at the Top1 position on the ALOT curves. The rate of S1 is even lower than that in Figure 5.5. That is because S1 benefits so little from the partition that the gain from less background clutters is even lower than the loss due to the partial occlusion. Moreover, in the case of partition without overlapping, S3 goes down to 49% from 54%, which indicates that S3 is not only the most sensitive to background clutters but also the most sensitive to the partial occlusion. The curves also show that S2 is the most robust to the partial occlusion among the three.

After the usage of partition, S1 still achieves the best performance among the three similarity measures. With overlapping its ALOT curve reaches as high as 62% at the Top1 position
The number of retrieved images

Figure 5.7: The ALOT curves of different similarity measure under scale-6, weighted with $N_j$, partitioned with overlapping, no spatial check. $S_1$: binary vector based inner-product; $S_2$: Euclidean distance between normalized vector; $S_3$: $L_1$-norm distance.
Figure 5.8: The Precision-Recall curves of different similarity measure under scale-6, weighted with $N_j$, partitioned with overlapping, no spatial check. $S_1$: binary vector based inner-product; $S_2$: Euclidean distance between normalized vector; $S_3$: $L_1$-norm distance.
Experiment

Figure 5.9: The ALOT curves of different similarity measure under scale-6, weighted with $N_{ij}$, partitioned without overlapping, no spatial check. $S_1$: binary vector based inner-product; $S_2$: Euclidean distance between normalized vector; $S_3$: $L_1$-norm distance.
Figure 5.10: The Precision-Recall curves of different similarity measure under scale-6, weighted with $N_j$, partitioned without overlapping, no spatial check. $S_1$: binary vector based inner-product; $S_2$: Euclidean distance between normalized vector; $S_3$: $L_1$-norm distance.
and at the Top5 position it goes up to 80%. Moreover, with partition $S_1$ gets better performance than the other two within all top 20 positions.

5.4 Effect of Scale Difference

To implement the retrieval under multiple scales, all the query images are downsized to 10 different scales (including the original scales) with a step factor of $\sigma = 1.25$. Figure 5.11 shows the ALOT and Precision-Recall curves gained by the three similarity measures under all the ten different scales without applying the weighting scheme or the partition mechanism. Transparently, the performance of all the three similarity measures differ significantly as the scale of the query images varies. The performance of the three similarity measures all reaches the maximum at scale-6 and decrease as the scale moves away from scale-6. That is because at scale-6, the objects in most query images are of approximately the same size as those in the database images. In this case, the repeatability of the local invariant features reaches its maximum. For the scale levels different from scale-6, the repeatability of the local features becomes lower and this in turn affects the similarity between the query and the truly relevant database images. Thus the retrieval performance declines at these scale levels. Similar results can be observed from Figure 5.12 and Figure 5.13 as well. For the three similarity measures, the ten different scales are sorted in the order of decreasing retrieval performance as follows:

- $S_1$: 6, 5, 7, 4, 3, 8, 2, 1, 9, 0;
- $S_2$: 5, 6, 7, 4, 3, 8, 2, 1, 9, 0;
- $S_3$: 5, 6, 4, 7, 3, 2, 8, 1, 0, 9;

Comparing Figure 5.11, Figure 5.12 and Figure 5.13, the following phenomena can also be observed:

1. under scale-0 and scale-1, the Euclidean distance method $S_2$ and the $L_1$-norm distance method $S_3$ outperforms the inner-product method $S_1$; under scale-2 the three methods achieve almost the same retrieval performance; under scale-3 and the larger scales, the inner-product method $S_1$ outperforms the other two;

2. with the inner-product method $S_1$, the retrieval performance achieved under scale-9 is better than that under scale-0, but the situation is opposite with the $L_1$-norm distance methods. With the Euclidean distance method, although at the Top1 position, scale-9 is slightly better than scale-0, but scale-0 overtakes scale-9 from the Top2.
§5.4 Effect of Scale Difference

The number of retrieved images $\leq 0.3$

Figure 5.11: The ALOT and Precision-Recall curves of the binary vector based inner-product method $S_j$ under different scales

Figure 5.11: The ALOT and Precision-Recall curves of the binary vector based inner-product method $S_j$ under different scales
The number of retrieved images

Figure 5.12: The ALOT and Precision-Recall curves of the normalized vector based Euclidean distance method $S_2$ under different scales
§5.4 Effect of Scale Difference

Figure 5.13: The ALOT and Precision-Recall curves of the normalized vector based $L_1$-norm distance method $S_3$ under different scales
The above phenomena show that the inner-product based method works better than the distance based methods when the scale of the query images is smaller than or similar to that of the database images. When the scale of the query images is larger than that of the database images but the disparity is within a certain range, the inner-product based method can still outperform the distance based methods (e.g., under scale-4 and scale-3). Only in some extreme condition such as the scale-0, the performance of the inner-product method goes lower than that of the distance based methods. The reason is that when the scale of the query image is larger than that of the database image, more local features will be extracted from the object in the query image than from the matching object in a database image. Since these extra local features cannot find true matches in the database images, they actually play the role of background clutters in the query images. For the inner-product based similarity measure, the similarity between the query image and the database image will not be reduced by the existence of background clutters, but the existence of background clutters increases the number of false matches. Thus some database images that do not contain the query object may gain a high score. In some extreme cases, the score accumulated from false matches can exceed that from the true matches. That is what happens under scale-0 and scale-1.

Figure 5.14, Figure 5.15 and Figure 5.16 show the ALOT curves and the Precision-Recall curves of the three similarity measures under all scales, where the $N_j$ based weighting scheme is used. As the figures show, applying a weighting scheme can improve the retrieval performance under all the scales and the rankings of different scales according to the retrieval performance may change after weighting. The following lists the rankings of different scales after weighting:

- $S_1$: 6, 5, 7, 4, 8, 3, 2, 9, 1, 0;
- $S_2$: 6, 5, 7, 4, 8, 3, 2, 1, 9, 0;
- $S_3$: 5, 6, 4, 3, 7, 2, 1, 8, 0, 9;

Comparing the ranks of different similarity measures under different scales, again we can see that the inner-product method works better with the queries of smaller scales whereas the $L_1$-norm distance method works better with the queries of larger scales. Such tendency of the two methods becomes stronger after weighting.

The retrieval performance of different similarity measures under different scales has also been tested with the $n_j$ based weighting schemes. The results are not presented in this thesis, because the two weighting schemes make no difference to the object retrieval under different scales.
Figure 5.14: The ALOT and Precision-Recall curves of the binary vector based inner-product method $S_1$ under different scales. The $N_j$ based weighting scheme is used.
Figure 5.15: The ALOT and Precision-Recall curves of the Euclidean distance method $S_2$ under different scales. The $N_j$ based weighting scheme is used.
Figure 5.16: The ALOT and Precision-Recall curves of $L_1$-norm distance method $S_3$ under different scales. The $N_j$ based weighting scheme is used.
Figure 5.17, Figure 5.18 and Figure 5.19 show the ALOT and Precision-Recall curves of the three similarity measures under different scales. The visual words are weighted with the $N_j$ based weighting schemes and the database images are now partitioned with overlapping. After using partition, the performances of all the three similarity measures have been improved under all scales. Now the ranks of different scales become:

- $S_1$: 6, 7, 5, 4, 8, 3, 2, 1, 9, 0;
- $S_2$: 5, 6, 7, 4, 3, 8, 2, 1, 0, 9;
- $S_3$: 5, 6, 4, 2, 7, 3, 1, 0, 8, 9;

Compared to the ranks obtained when no partition is implemented, we can see that all similarity measures now work better with the queries of larger scales. That is because after partition, the background clutters in the obtained sub-images are significantly reduced. Having a big number of visual words in the query image no longer incurs many false matches but increases the true matching numbers. On the other hand, the query images of smaller scales no longer take the advantage of having fewer false matches. Instead, it is affected by the problem of sharing less true matches with the database images.

### 5.5 Combining the Results from Multiple Scales

#### 5.5.1 By Score

Figure 5.20 shows the ALOT curves and the Precision-Recall curves of the final retrieval results obtained by combining from the similarity scores under multiple scales with different similarity measures that are explained in the caption of Figure 5.20. As we can see, when generating the final result by accumulating the similarity scores over different scales,

considering more scales can give rise to better retrieval performance. The measure of $s_{10}$ which performs best among all the similarity measures reaches 50% at the Top1 and rises to 70% at the Top5. However, by considering the matching percentage as discussed in Chapter 4.5.2, better performance is achieved as shown by $p_1$, $p_5$ and $p_{10}$. Moreover, the retrieval performance does not vary much as the number of involved scales changes. That is because the matching percentage gained by the true matching images at the proper scales becomes dominating to the final results. Its advantage persists as the number of involved scales varies.
§5.5 Combining the Results from Multiple Scales

Figure 5.17: The ALOT and Precision-Recall curves of the binary vector based inner-product method $S_i$ under different scales. The $N_i$ based weighting scheme is used and partition with overlapping is adopted.
Figure 5.18: The ALOT and Precision-Recall curve of Euclidean distance method $S_2$ under different scales. The $N_j$ based weighting scheme is used and partition with overlapping is adopted.
§5.5 Combining the Results from Multiple Scales

The number of retrieved images

Figure 5.19: The ALOT and Precision-Recall curve of $L_1$-norm distance method $S_3$ under different scales. The $N_j$ based weighting scheme is used and partition with overlapping is adopted.
Figure 5.20: $s_1$: the final result produced by selecting the similarity scores of the best scales; $s_5$: the final result produced by combining the similarity scores of the best 5 sequential scales; $s_{10}$: the final result produced by combining the similarity scores of all scales; $p_1$: the final result by selecting the matching percentage of the best scales; $p_5$: the final result produced by combining the matching percentage of the best 5 sequential scales; $p_{10}$: the final result produced by combining the matching percentage of all scales;
5.5.2 By Rank

As Figure 5.21 shows, compared to the final retrieval results obtained by combining the similarity scores, combining the ranks appears to be more accurate. When the step factor $\sigma$ is set as 1.25, the best retrieval result is gained by combining the ranks from the best 4 sequential scales. However, after changing the $\sigma$ to be $1.25^2$, the best retrieval result is achieved by combining the ranks from the best 3 sequential scales which is denoted by $R$. After combining the results from multiple scales with the method $R$, the final retrieval performance is almost as good as that of scale-6 under which the best retrieval performance is achieved when retrieval under a single scale. Of course the method $R$ is more useful than retrieving under scale-6 only because the former bears no assumption on the scale difference between the query and the database images.

5.6 Retrieval Speed

To test the system’s runtime speed, the experiment is conducted with 300 different query images under 10 different scales, hence totally 3,000 query images are tested. The database includes 3,153 different images, each of which are partitioned into 25 sub-images with overlapping. Hence it is equivalent that about 78,825 database images, each of which contains about 2,000 local features, are involved during each time of retrieval.

The runtime speed of the proposed object retrieval system mainly depends on three components, namely the speed of local feature extraction from a query image (extraction speed), the speed of visual word assignment for each local features extracted from the query image (assignment speed), and the speed of retrieving in the database with the visual word based description of the query (retrieving speed). The third component is the most important among the three because it determines the system capability in handling large image databases. Thanks to the visual word and the invert file, the retrieving speed of the proposed system can reach as high as 40 image/second on a PC with 4G physical memory. In other word, it costs the system only 0.25s to retrieve the query image under 10 different scales in the image database. Tests have also been carried out on computers with different hardware configurations. It is observed that the computational workload for the CPU is minor and the retrieval speed is mainly dependent on the size of the physical memory.

Currently, the major bottleneck for the system’s runtime speed is the extraction speed. On a PC with 4G physical memory, it takes about 60 seconds to extract local features from a query image under 10 different scales. If the retrieval is implemented under 5 different scales with a
Experiment

The number of retrieved images

Figure 5.21: $r_2$: the final result produced by combining the ranks of the best 2 sequential scales with $\sigma = 1.25$; $r_3$: the final result produced by combining the ranks of the best 3 sequential scales with $\sigma = 1.25$; $r_4$: the final result produced by combining the ranks of the best 4 sequential scales with $\sigma = 1.25$; $r_5$: the final result produced by combining the ranks of the best 5 sequential scales with $\sigma = 1.25$; $R$: the final result produced by combining the ranks of the best 3 sequential scales with $\sigma = 1.25^2$. $\sigma$ denotes the step factor which is used to downsized the query images.
step factor $\sigma = 1.25^2$ which achieves the best performance and is labelled as $R$ in Figure 5.21, the feature extraction can be finished in 30 seconds. With the tree structure created during visual words generation, the assignment of each local feature to its corresponding visual word is highly time efficient. Finding the nearest neighbour for each local feature among 100 candidates which is the branch factor of the hierarchical $k$-means at 4 different levels can be finished in realtime.

### 5.7 Retrieval Examples

Figure 5.22 and Figure 5.23 show some successful retrieval examples by the proposed system. Only the Top3 returned database images are displayed for each query image. Figure 5.24 gives some examples of the queries to which the proposed system fails to return any true matching database images among the Top5 or even the Top10. The reasons why the system fails on the queries in Figure 5.24 can be explained as follows:

1. **lack of local features**: some of the queries are of too simple patterns or the patterns are not sharp enough. Some others are too small in size. In either case, few local features can be extracted from them, which leads to the failure of the local feature based object retrieval.

2. **unstable visual appearance**: some of the queries are in the form of soft bags whereas some others have glistening surfaces. Either condition results in unstable visual appearance. This type of variances cannot be effectively handled by local invariant features.

The failure of the examples in the first row in Figure 5.24 is mainly caused by the first reason and the failure of the second row are mainly caused by the second reason. However, some queries hold two reasons simultaneously.
Figure 5.22: Example of good retrieval result: The first column shows the queries and the images on the other columns in order are the top three retrieved images.
Figure 5.23: More examples of good retrieval result
Figure 5.24: Retrieval Examples of low quality: only queries are shown
Conclusion

This work builds up a system to retrieve objects from a big supermarket. The new system is based on the framework of local invariant feature based image retrieval. However, the nature of the object retrieval problem studied in this work, namely the large scale difference between the query images and the database images, the strong background clutters and the existence of multiple copies of identical objects in an database image, prevents the existing local invariant feature based image retrieval methods from being applied straightforwardly to the retrieval problem in this work. To achieve good retrieval performance, this work designs a new similarity measure to handle the strong background clutters and the existence of multiple copies. Besides, a multiple scales based retrieval approach is developed to tackle the presence of large scale difference between images. A visual word based spatial check is also proposed to filter out the false matches with little computational overhead.

The presence of the strong background clutters in the database images is one of the most significant characteristics that distinguish this work from those in the literature. While a query image usually provides about 2,000 local features, around 20,000 local features can often be extracted from every database image. The number of the local features that appear in both the query image and the database images containing the query object may be no more than 20. The existing retrieval approaches often describe images locally but compare them globally. With those approaches, a database image containing the query object will be wrongly assigned a low similarity score due to the strong background clutters. To solve this problem the new similarity measure proposed in this work evaluates the similarity between two images based on only their commonness. By doing so, the strong background clutters cannot considerably affect the image similarity evaluation anymore. With this new measure, significant improvement to the retrieval performance has been achieved. Describing images with a histogram showing the number of occurrence of each visual word is widely used in the previous retrieval approaches. However, since the new similarity measure compares two unnormalized histograms, a high similarity score may be wrongly assigned to the database image where the
false matches are generated by the multiple copies of an identical object. After converting each histogram into a binary vector, the above problem is avoided and the similarity between images is actually measured by counting the number of types of visual words shared by the images. Compared to the use of the original histogram, one risk of using the binary vector is that the proportion information among different visual words detected from a single object is lost. However, different objects can usually be distinguished from each other by comparing the contained visual word types only. Therefore, omitting the proportion information does not really become a problem for the object retrieval in this work. Another advantage of using the binary representation is the significant reduction of the size of the feature database, since the exact number of occurrence of each visual word does not need to be stored anymore.

If the local invariant features can be extracted from a same region under different scales, the images presented under different scales can hold a solid similarity. However the repeatability of the state-of-the-art local invariant features over large scale difference is not high enough. Moreover, more features can usually be extracted from the images under larger scales. As a result, many features extracted from an image under a larger scale cannot find true match in an relevant image under a smaller scale but false matches in irrelevant images, which further elevates the matching difficulty between the images having large scale difference. To solve this problem a multiple scales retrieval mechanism is developed in this work and the final retrieval result is generated by combining the retrieval results obtained under multiple scales. The combination can be made based on the similarity scores or the rankings of the database images obtained under different scales. Experimental study shows that for the object retrieval problem in this work, the best retrieval performance is attained by considering only the higher rankings obtained by each database image under some of the scales, disregarding the scales under which their rankings are low.

By employing the visual word and the invert file, the proposed system achieves promising runtime retrieving speed. The generation of visual words accomplishes the computation required by nearest neighbour search during off-line. The usage of the invert file narrows the searching range from the whole database to the images which share at least one visual word with the query image. This speed is relatively independently from the size of the image database.

The spatial check methods adopted by the previous work bear relatively high computational load. Furthermore, additional information besides the visual words is required to implement the spatial check. This thesis improves this situation by integrating the spatial information into the visual words and implementing the spatial check together with the image similarity evaluation. The orientation and area information of local regions is used. Loosely
checking the spatial consistence based on the two types of information can remove a large number of false matches with minor computation sacrifice. However, due to the nature of the object retrieval problem in this work, the visual word based spatial check method does not bring forth significant improvement to the retrieval performance. Nevertheless, experimental result suggests that this spatial check method can well improve the matching credibility. Its potential to general visual word based image retrieval is worth exploring in the future work.
Bibliography


KADIR, T., ZISSERMAN, A., AND BRADY, M. 2004. An affine invariant salient region detector. (pp. 9, 10)


