Feature weighting with augmented visual phrase in visual product recognition

Zhang, Wen

2016

Zhang, W. (2016). Feature weighting with augmented visual phrase in visual product recognition. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/68832>

<https://doi.org/10.32657/10356/68832>

Feature Weighting with Augmented Visual Phrase in Visual Product Recognition

Submitted by

Zhang Wen

Supervisor

Assoc. Prof. Yap Kim Hui

School of Electrical & Electronic Engineering

A thesis

submitted to

Nanyang Technological University

in partial fulfillment of the

requirements for the

Degree of Master of Engineering

October 2015

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....July, 2015.....

.....Zhang Wen.....

Acknowledgements

First of all, I would like to express my deepest gratitude towards my supervisor, Associate Professor Yap Kim Hui, for his nurturing, encouragement and continuous support during my research.

Secondly, I wish to express my sincere thanks towards Miao Zhenwei and Zhang Dajiang for their help.

I am also grateful towards all the team members in the ROSE lab, for their help, collaborations, and friendship.

Finally, I am deeply indebted to my family for their love and endless support. Without them, I would never have chance and courage to pursue a degree in Master of Engineering.

Abstract

Significant progress towards image search has been made in the past decade through the development of local invariant features. Among existing local feature detectors, the Scale Invariant Feature Transform (SIFT) [1] is widely used since it is designed to be invariant to minimal illumination changes and certain geometric transformations. However, in practice, the recognition performance is still subject to actual condition and there still exist various cases that SIFT can't handle, such as non-linear illumination changes [2][3] and certain affine transformation [4]. It is noted that images taken by mobile phones often suffer from such variations. In order to address this issue, we propose a feature weighting algorithm to determine the stability of interest points under various photometric and geometric transformations. By assigning different scores to these interest points according to the repeatability they are in the augmented samples of different illumination or geometric transformations, the stable interest points can be obtained.

The Bag of Word (BoW) based representation is widely used in image visual search due to its good performance and high computational efficiency. In traditional BoW representation, the local descriptors in the whole image are treated with equal importance. This restricts its performance as in most scenarios the local descriptors on foreground objects should be assigned more weight than the descriptors on the background. In view of this, we incorporate saliency information with the proposed feature weighting algorithm to further improve the performance. The weighted interest points are then used in the weighted scalable vocabulary tree (WSVT) framework for the image recognition task. Geometric verification (GV) is used to re-rank the images. Experimental results on a commercial product database show the proposed feature weighting algorithm outperforms the current SVT recognition without feature weighting by 5%.

Moreover, with the popularity of image editing softwares and social media networking services such as Instagram and Facebook, a growing number of users tend to process digital images using various post-processing filters before uploading. Some SIFT features are not stable given the nonlinearity of these filters, which may degrade recognition performance. Therefore we further proposed a feature

weighting method based on post-processing filters and incorporate it into the previous recognition framework. Experimental results on a commercial product database show the proposed algorithm outperforms the current SVT recognition without feature weighting by more than 10% in average recognition performance for various post-processing effects.

Query images captured by mobile phones often suffer from illumination, scale, and viewpoint changes. This poses a great challenge in visual recognition. In view of this, we propose a framework that uses Augmented Visual Phrases (AVP) in the Bag-of-Phrase model. By checking the consistency between keypoints in the original image and augmented images, a pool of augmented features are constructed by merging all the discriminative keypoints and their corresponding descriptors from the original as well as the transformed images. AVPs are then selected from the pool of augmented features. These selected visual phrases are meaningful as they incorporate features carrying more diverse information. Database images are indexed with a two-dimensional inverted index of visual phrases. To eliminate spurious matches of visual phrases, we apply geometric verification (GV) to top-ranked retrieved images. Experimental results show that the proposed AVP method outperforms the current BoP method by 9% in recognition rate.

Contents

Acknowledgements	i
Abstract	ii
Table of Contents	iv
List of Figures	vi
List of Tables	viii
List of Acronyms	ix
1 Introduction	1
1.1 Background	1
1.2 Objectives	2
1.3 Contributions	4
1.4 Organization of Thesis	4
2 Literature Review	6
2.1 Interest Point Detectors and Descriptors	6
2.1.1 SIFT and the Relative Works	6
2.2 Bag-of-Words Model	9
2.3 Scalable Vocabulary Tree and Its Variants	11
2.4 Geometric Verification	14
2.5 Summary	15
3 Transform and Saliency Feature Weighting in Image Recognition	17
3.1 Photometric Feature Weighting (PFW)	17
3.2 Geometric Transformation Feature Weighting (GFW)	21
3.3 Transform and Saliency Feature Weighting (TSFW)	23
3.4 Post-processing Filters Feature Weighting (PFFW)	26

3.5	Experimental Results and Setting	29
3.6	Experimental Results on Photometric Feature Weighting (PFW) Image Recognition .	31
3.6.1	Recognition Rate for Different Gamma Values	31
3.6.2	Experimental Results on Different Number of Reference Images	31
3.7	Experimental Results on Geometric Feature Weighting (GFW)	33
3.7.1	Experimental Results for Individual Geometric Transformation	33
3.7.2	Experimental Results Using Combined Geometric Transformation	34
3.8	Experimental Results for Joint Photometric and Geometric Feature Weighting (JFW)	35
3.9	Experimental Results for Transform and Saliency Feature Weighting (TSFW)	36
3.10	Experimental results on post-processing filters feature weighting (PFFW)	38
3.11	Experimental results on combined feature weighting	39
3.12	Conclusion	42
4	Augmented Visual Phrase (AVP) in Product Recognition	43
4.1	Augmented Visual Phrases (AVP) Generation and Selection	43
4.2	AVP Indexing and Querying	47
4.3	Experimental Results on AVP Product Recognition	48
4.4	Conclusion	49
5	Summary and Future Work	51
5.1	Summary	51
5.2	Future Work	51
	Author's Publication List	53
	Bibliography	54

List of Figures

2.1	Overview of the SIFT detector scheme [5].	7
2.2	Keypoint descriptors calculated from image gradients [5].	9
2.3	Illustration of BoW histogram [6]	11
2.4	(a) Construction of a SVT by hierarchical k-means clustering (b) SVT and the associated inverted index. [7]	12
2.5	Process of SVT construction [8]	12
2.6	Original frontal-view images and four generated images with other perspectives	14
2.7	Matching pairs (a) before GV and (b) after GV	15
3.1	Gamma compression and expansion	18
3.2	Proposed image recognition framework	19
3.3	(a) original image; (b) vertical stretching; (c) horizontal stretching; (d)scaling; (e)shearing; (f)rotation.	22
3.4	(a) One image from database (b) The corresponding saliency map	25
3.5	Survey on Instagram filter by Tarek Riman [2013]	26
3.6	Survey on Instagram filter popularity by Brandon Gaille [2013]	27
3.7	Post-processing filters feature weighting (PFFW)	28
3.8	From left to right, (a) the original image and (b)-(d) images processed by Lomo-fi, Xpro II, Nashville filter effects	28
3.9	Sample of reference images from the commercial product database	30
3.10	Sample of query images from the commercial product database	30
3.11	Recognition rate for different gamma values	32
3.12	Five reference images with gamma value= 2.1^{-2} , 2.1^{-1} , 1, 2.1 , 2.1^2 respectively	32
3.13	Experimental results on different number of reference images	33
3.14	Experimental results for joint photometric and geometric feature weighting (JFW) . .	36
3.15	(a) One image from database (b) The corresponding saliency map	37
3.16	Experimental results for transform and saliency feature weighting (TSFW)	37

3.17	Comparison of case study for the SVT [8] method without transformation, SVT [8] + saliency [9] method and the proposed TSFW method	38
4.1	Proposed Augmented Bag-of-Phrase Framework	44
4.2	The extracted SIFT keypoints in the original image and in two images after photometric transformation. From left to right, the original image, the darkened image with gamma value 2.1^{-1} and brightened image with gamma value 2.1	45
4.3	A sample 1-D vocabulary tree with depth factor 2, branch factor 3 and the associated inverted index	46
4.4	An example of AVP formation	47
4.5	Inverted index of AVPs and online querying.	48
4.6	Case study of AVP method with SVT [8] and BoP [10]	50

List of Tables

3.1	Experimental results for individual geometric transformation	34
3.2	Experimental results for combining various geometric transformation types	35
3.3	Experimental results for feature weighting on instagram filters, the training filters are added from the 1 st popular filter to 7 th popular filter	40
3.4	Experimental results for feature weighting on Instagram filters, the training filters are added from the 7 th popular filter to 1 st popular filter	40
3.5	Experimental results for different feature weighting on Instagram filters, with part of filters including Hefe, Nashville and Lomo-fi replaced to Toaster, Gotham and Walden	41
3.6	Experimental results for combined feature weighting on instagram filters	41
4.1	Performance comparison of the SVT [8], BoP [10] and proposed AVP method	49

List of Acronyms

SIFT Scale-invariant Feature Transform

BoW Bag-of-Words

SVT Scalable Vocabulary Tree

WSVT Weighted Scalable Vocabulary Tree

GV Geometric Verification

RANSAC Random Sample Consensus

GBVS Graph-Based Visual Saliency

BoP Bag-of-Phrases

DVP Descriptive Visual Phrases

AVP Augmented Visual Phrases

PFW Photometric Feature Weighting

GFW Geometric Feature Weighting

JFW Joint Photometric and Geometric Feature Weighting

TSFW Transform and Saliency Feature Weighting

PFFW Post-processing Filter Feature Weighting

Chapter 1

Introduction

1.1 Background

The proliferation of devices with imaging functionality (e.g. digital cameras, mobile phones) and increasing popularity of social and media sharing platforms (e.g. Facebook, Instagram) has led to an explosive growth in the number of images being generated and shared in recent years. Hence there is a strong need to address the timely issue of visual image search and retrieval [11] [12] where the user and system can search and retrieve visually similar images from image databases that are being generated by social media users and Internet.

In recent years local feature detection has drawn great attentions and made a tremendous impact on content-based visual search. Local features detectors have many advantages over global features, such as sparse representation of the image, robust to occlusion and noisy background. Local features have been used in various computer-vision applications, including image retrieval, object recognition, object categorization, texture classification and video shot retrieval [13]. Based on the detected local image patterns, the local feature detectors can be roughly grouped into three categories: corner detectors, blob detectors and region detectors. Many works has been done to comprehensively evaluate these detectors using various criteria, including visual inspection, repeatability, consistency and matching score [14] [15].

The current standard pipeline for image recognition relies on extracting or learning a model that can make predictions for the query image. Models are often constructed via unsupervised classification methods, such as the well-known K-means clustering [16], scalable vocabulary tree construction (SVT) [8], which proposes an inverted-index strategy that provides highly efficient image retrieval.

In spite of the notable strides achieved within the last decades, Matching and learning visual objects

is still challenging on a number of aspects. First, images from the same object category can produce very different images in diverse lighting environment, perspective changes, partial occlusions, and various backgrounds. In addition, computational complexity must be taken into consideration as well. Hence it is necessary for an image recognition algorithm to efficiently and accurately extract local features from query image, to accommodate high-dimensional image representations and to quickly return an image from an image database.

1.2 Objectives

Among various local feature detectors, scale-invariant feature transform (SIFT) detector, proposed by Lowe [1] is a state-of-art algorithm providing one of the best performance in visual search. It extracts local features by employing the difference of Gaussian to approximate the normalized Laplacian of Gaussian of the image. Compared with other interest point detectors, this method can speeds up the process and improve the computational efficiency. However, in practice, there still exist various cases that SIFT can't handle, such as non-linear illumination changes [2][3] and certain affine transformation [4].

In order to address this issue, we proposes a new feature weighting algorithm. This algorithm selects interest points that are robust under different photometric and geometric transformations. Gamma correction is used to simulate the photometric changes, while vertical stretching, horizontal stretching, scaling, shearing and rotation are used for geometric transformation. Repeatability of the interest points under these variations is tested. Strong weights are assigned to the interest points which have a high repeatability. It's shown that the matching accuracy can be enhanced when the weights are assigned according to this feature weighting strategy.

The Bag of Word (BoW) [17] [18] [19] based representation is widely used in image visual search due to its good performance and high computational efficiency. In the standard BoW representation, the local descriptors in the whole image are treated with equal importance. This restricts its performance as in most scenarios the local descriptors on foreground objects should be given more weight than the descriptors on the background. The traditional BoW representation can be improved by incorporating saliency information. In view of this, we incorporates saliency information to further improve the performance. A saliency map for each image in the database is generated based on the Graph-Based Visual Saliency (GBVS) algorithm. Then each interest point is weighted by the repeatability of the point under various variations and the corresponding saliency value on the saliency map. The weighted interest points are then used in the weighted scalable vocabulary tree (WSVT) framework for the image recognition task. Geometric verification (GV) is used to re-rank the images. The

proposed algorithms are tested on the commercial product database, which contains 3881 reference images and 333 query images from 41 categories. The experimental results show that the proposed algorithm outperforms the current SVT recognition method [8] without feature weighting by about 5% to give a recognition rate of 91%.

Moreover, with the popularity of image editing softwares and social media networking services such as Instagram and Facebook, a growing number of users tend to process digital images using various post-processing filters (e.g., “lomo-fi” and “early bird” in Instagram) before uploading to express strong individual styles. The post-processed images can be quite different from the original ones in color tone, contrast, saturation etc. Some SIFT features are not stable given the nonlinearity of these filters, which may degrade recognition performance. Therefore we also proposed a feature weighting method based on post-processing filters and incorporate it into previous recognition framework to recognize images after these effects. Experimental results on a commercial product database show the proposed algorithm outperforms the current SVT recognition method [8] without feature weighting by above 10% in average recognition performance for various post-processing effects.

Indeed, BoW based method provide good recognition performance in many application domains. However, even a finely grained visual word in BoW method has limited descriptive and discriminative ability [20]. This is because this approach ignores the spatial correlation between visual words, hence restricting its performance. Especially, it is also more challenging to distinguish foreground product of interests from the background clutters without spatial information for product recognition. Combining BoW with geometric verification (GV) is one way to utilize the spatial information. However, GV is only feasible for limited number of top ranked candidates due to its high computational cost. In recent years, some works have been proposed to utilize the spatial and contextual information of local descriptors by grouping multiple neighboring visual words into visual phrases [21]. Particularly in BoP approaches, visual words co-occurring within certain spatial region are selected as discriminative visual phrases (DVP) in [20], [22] and [10]. However, these methods only consider the original database images for DVP candidate generation and selection. Since SIFT keypoints are not stable and the descriptors may change under significant photometric and geometric transformation, the original DVPs can’t match the query images which may suffer from these distortions, hence reducing the recognition performance. To handle these issues, we propose a framework that uses Augmented Visual Phrases (AVP) in BoP model. By checking the consistency of keypoints in the original image and augmented images, a pool of augmented features are constructed by merging all the original descriptors as well as the transformed descriptors. AVPs are then selected from the pool of augmented features. These AVPs selected are comprehensive and incorporates effective represen-

tative features of query images. Ideally, matches between higher-order visual phrases indicate higher confidence or reliability. But they also suffer from low repeatability. Therefore, we only consider the second-order visual phrases, i.e., visual phrases consisting of two visual words. The online image matching process using visual phrases is conducted efficiently with a two-dimensional inverted index. Geometric verification is then performed on the top-ranked images to further improve the recognition accuracy.

1.3 Contributions

First, we propose a new feature weighting framework: (1) By giving different weights to SIFT descriptors according to their stability in various transformation, we can address the problem that some SIFT descriptors are more stable towards illumination changes and certain geometric transformation. (2) The saliency information is incorporated into feature weighting, where more salient descriptors on the foreground are strengthened with larger weights. Experimental results on a commercial product database demonstrates that the recognition rate is significantly improved when compared with current SVT recognition without feature weighting.

Second, we also proposed a feature weighting method based on post-processing filters and incorporate it into the proposed recognition framework to recognise images after these effects. Experimental results on a commercial product database show the proposed algorithm outperforms the current SVT recognition without feature weighting by more than 10% in average recognition performance for various post-processing effects.

Besides, we proposed an AVP algorithm that can address the issue of photometric and geometric distortions in the query images. We generated augmented features using transformed images. The AVP algorithm integrate two properties that can enhance the recognition performance. (1) Spatial information to make features more discriminative; (2) Augmented features to handle potential distortion, including different illumination and perspective changes. Experimental results on a commercial product dataset shows the proposed AVP method achieves a recognition rate of 93.1%, which outperforms the current BoP method [10] by 9%.

1.4 Organization of Thesis

The rest of the thesis is organized as follows:

- In Chapter 2, a review on SIFT, SVT, inverted-index and Geometric Verification is presented.
- In Chapter 3, a feature weighting algorithm is proposed. Then some experiments are carried as follows: the experiments on photometric feature weighting (PFW), geometric feature weighting (GFW), joint photometric and geometric feature weighting (JFW), transform and saliency

feature weighting (TSFW), and post-processing filters feature weighting (PFFW). Then a conclusion is given summerizing the various proposed feature weighting techniques.

- In Chapter 4, the proposed AVP technique on product recognition and experimental results of the proposed AVP method are presented.
- In Chapter 5, the research works in this thesis are summarized and a few future research directions are presented.

Chapter 2

Literature Review

In this chapter, a detailed literature review of the framework for image recognition including interest point detectors and descriptors, Scalable Vocabulary Tree(SVT), inverted-index and Geometric Verification are presented.

2.1 Interest Point Detectors and Descriptors

Image features are categorized into two types: global and local features [23][1]. The global feature generalizes the entire image and describes the overall properties, such as colors, shapes and textures. Most of the global representation-based recognition approaches compare the entire images or the entire image windows. However, they cannot handle partial occlusions, or strong viewpoint variation.

The local features [24] [25] [26], on the other hand, represent the local characteristics based on the particular salient patches or interest points. Those features have a significant impact on the development of robust and efficient recognition algorithms that can work under a wide variety of viewing changes and partial occlusion. Based on the detected local image patterns, the local feature detectors can be roughly grouped into three categories: corner detectors [27] [28] [29], blob detectors [30] [31] [32] and region detectors [33] [34] [35]. Among various local feature detectors, scale-invariant feature transform (SIFT) detector is a state-of-art algorithm in visual search.

2.1.1 SIFT and the Relative Works

By employing the difference of Gaussian to approximate the normalized Laplacian of Gaussian of the image, SIFT detector speeds up the local feature detection process and improve the computational efficiency. SIFT feature extraction involves four major steps: scale-space extrema detection, keypoint localization, orientation assignment, and keypoint descriptor.

Figure 2.1: Overview of the SIFT detector scheme [5].

Scale-space extrema detection In the first stage, as shown in Fig. 2.1, images in different scales are generated by continuously smoothing with Gaussian kernel. This stage searches for invariant features among all possible scales using the scale space function $L(x, y, \sigma)$. This function is produced from the convolution of a variable-scale Gaussian $G(x, y, \sigma)$ with an image input $I(x, y)$, where (x, y) represents the coordinate of a pixel location and σ represents the scale:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.1)$$

where

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right) \quad (2.2)$$

To make the keypoint detection more efficient, David Lowe has suggested applying scale-space extrema in the difference-of-Gaussian (DoG) function convolved with the input image. This can be calculated from the difference of two nearby scales which are separated by a multiplicative constant k ,

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma). \quad (2.3)$$

Local extremum of the difference-of-Gaussian images in the spatial and scale dimensions are detected as the interest points candidates.

Keypoint localization In the second stage, a detailed fit is performed for each keypoint candidate to the nearby data for location, scale, and ratio of principal curvatures. The discrete difference-of-Gaussian image is interpolated based on the Taylor expansion, and the refined location of the keypoint $\hat{x}$ is the nearest extremum of the interpolated curve. This is achieved by taking the derivative of interpolation function, and setting it to zero, giving

$$\frac{\sqrt{X} - \sqrt{t}}{\sqrt{X_{max}} - \sqrt{t}} \quad (2.4)$$

where D and its derivatives are computed at the location (x, y) and $\mathbf{x} = (x, y, \sigma)^T$ is the offset from this point. The relative contrast is estimated as

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}}. \quad (2.5)$$

Low contrast points are considered as noise and rejected by a threshold of the contrast. The principal curvatures at each keypoint are computed from the its Hessian matrix, defined as

$$\mathbf{H} = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} \quad (2.6)$$

where $I_{mn}(m, n \in [x, y])$ represents the second order derivative of the image. High ratio of the principal curvatures indicates the point on the edges. A threshold of the ratio is used to remove this kind of keypoints. The ratio r is measured by $Tr(\mathbf{H})^2 / Det(\mathbf{H}) = (r + 1)^2 / r$ to avoid computing the two eigenvalues of $\mathbf{H}$.

Orientation assignment In this stage, the keypoints are assigned with one or more orientations based on the gradient directions of local image in order to achieve rotation invariance. The keypoints scale is used to select the Gaussian smoothed image L in order to achieve scale-invariant computation. Then the gradient magnitude, $m(x, y)$ and orientation $\theta(x, y)$ are pre-computed using pixel difference for each sample image $L(x, y)$:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.7)$$

$$\theta(x, y) = \arctan((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))) \quad (2.8)$$

Keypoint descriptors Through the previous steps, scale, location and orientation have already been assigned to each keypoint. In this step, keypoint descriptor is computed by measuring the local image gradient magnitude and orientation at the sample point near the keypoint . The descriptors are transformed into a representation which is invariant to obvious change in local shape and illumination condition.

As shown in Fig.2.2, the samples are weighted by the Gaussian window. Subsequently, these samples are accumulated into orientation histograms summarizing the contents over 4x4 sub-regions. The length of each arrow corresponds to the sum of the gradient magnitudes near that direction within the region (only 2x2 descriptor array computed from an 4x4 set of samples are shown in the figure).

Figure 2.2: Keypoint descriptors calculated from image gradients [5].

Since SIFT obtained good recognition performance, it is broadly used in many applications in different fields, such as image retrieval, image matching, and machine vision. After SIFT was proposed, many researchers have been trying to improve it in many aspects, including its invariance to geometric and illumination transformation. Among the improved algorithms, the two that draw most attention are Affine-SIFT (ASIFT) [36] and Color-SIFT (CSIFT) [37]. ASIFT aims to improve the performance in the situation when SIFT does not perform well with images with affine changes. ASIFT is mathematically proved to be invariant to affine transformation by adopting an image affine transformation model resulting from simulating all views based on the two camera orientation parameters. CSIFT addresses the problem that SIFT is designed mainly for gray images and lost valuable color information. Instead of using the gray space to represent the input image, the proposed approach builds the SIFT descriptors in a color invariant space. Experimental results show CSIFT is more robust than the base SIFT with respect to color and illumination changes.

In addition to the variances of SIFT, some works try to boost the SIFT descriptor against illumination variation through creating additional samples in different illumination conditions and extracting the desired features of these conditions [38] [39]. In order to extract more discriminative and representative features, [40] combined Harris corner feature with SIFT and apply feature weighting according to corner degrees to distinguish contribution among features. Experimental results show the method worked well for image classification.

2.2 Bag-of-Words Model

The local features extracted by the SIFT algorithms provides an efficient way for image recognition and retrieval. However, directly computing the distance between descriptors is very time consuming. Bag-of-Words (BoW) model [6] is proposed to address the issue and improve the image recognition efficiency.

The Bag of Words model is commonly used for document classification and information retrieval. The text in a document comprises a collection of words, and the occurrence of each word is treated as a feature for training. Recently, this model has also been widely used in computer vision and pattern recognition.

When using BoW model in image recognition, each image is regarded as a document and the image features are the counterpart of the “words” from the document. In this way, an image can be seen as a collection of image features, i.e. “a bag of word”.

After extracting the image features from the training database, we need to construct a “codebook” by applying clustering algorithms, such as K-means clustering. The codebook can be considered as a collection of so-called “visual word”.

The K-means clustering include the following steps:

1. Initialize the value of K as the number of partitions and the cluster centers, $\mu_1, \dots, \mu_k$.
2. Assign each keypoint to its nearest cluster center according to the following rule. $x_1, \dots, x_N$ are data points or vectors of observations:

$$\mathcal{C}_k = \{n : k = \arg \min_k ||\mathbf{x}_n - \mu_k||^2\} \quad (2.9)$$

where $\mathcal{C}_k$ represents the set of points belong to μ_k .

3. Recalculate the new cluster centers, $|\mathcal{C}_k|$ is the number of points in $\mathcal{C}_k$:

$$\mu_k = \frac{1}{|\mathcal{C}_k|} \sum_{n \in \mathcal{C}_k} \mathbf{x}_n \quad (2.10)$$

4. Repeat the steps until the algorithm has converged when positions of cluster centers no longer change.

After the codebook generation, the entire image is summerized based on its distribution (histogram) of word occurrences. Fig.2.3 shows three examples for BoW histogram. Analogous to bag of words representation commonly used for documents, column in the histogram indicates the frequency of occurrence for each codeword from the image.

Figure 2.3: Illustration of BoW histogram [6]

2.3 Scalable Vocabulary Tree and Its Variants

In order to provide more efficient training process and enables the use of a larger vocabulary, an approach called scalable vocabulary tree (SVT) [8] combined with inverted index was proposed to quickly compare similarities between database images and a query image.

The SVT for a particular image database is built by recursively applying hierarchical k-means clustering all the feature descriptors representing the training database. A tree of a specific database is determined by two parameters, K and L , which represent the branch factor (number of children of each node) and the maximum number of level respectively. SVT includes $C = K^L$ leaf nodes in total. Fig.2.4(a) shows an SVT with $K = 3, L = 2$.

Initially, K-means algorithm is applied to all the representative training data to generate C cluster centers, meanwhile the descriptors are divided into C groups. The same division process is recursively run on each groups of descriptors, until sufficient nodes are generated to obtain good recognition accuracy. The construction process is illustrated in Fig. 2.5.

After the SVT construction, the feature descriptors of the query image are passed through the vocabulary tree and by applying a hierarchical scoring strategy, the most similar database image can be found. The scoring strategy is explained as follows.

First, the i component of query image q and one database image d can be represented respectively

Figure 2.4: (a) Construction of a SVT by hierarchical k-means clustering (b) SVT and the associated inverted index. [7]

Figure 2.5: Process of SVT construction [8]

as follows:

$$q_i = n_i * w_i \quad (2.11)$$

$$d_i = m_i * w_i \quad (2.12)$$

in which n_i and m_i represent the number of descriptors of query image and database image, respectively, with a path through node i . w_i is an inverse document frequency weight used to penalize nodes that are visited with a high frequency. Typically w_i is based on entropy as follows,

$$w_i = \ln \frac{N}{N_i} \quad (2.13)$$

where N represents the number of database images, N_i is the number of images in the database with a minimum of one vector path through node i .

Then a score s for the database image can be calculated according to the normalized difference between query vector and a current database image:

$$s(q, d) = \left\| \frac{q}{\|q\|} - \frac{d}{\|d\|} \right\| \quad (2.14)$$

Normalization is implemented in order to make it possible to use inverted files. A standard normalization $L2$ norm is applied to compute the Euclidian distance:

$$\|q - d\|_2^2 = 2 - 2 \sum_{i|q_i \neq 0, d_i \neq 0} q_i * d_i. \quad (2.15)$$

From the equation we can see that only the nodes where $q_i \neq 0$ & $d_i \neq 0$ are needed to get the matching score. Therefore in order to reduce computation complexity, inverted index file is built for each database image to only traverse the nodes where $q_i \neq 0$ & $d_i \neq 0$. As illustrated in Fig. 2.4(b). Each node in the vocabulary tree has an inverted file attached which included two lists, id-numbers of the images ($I_{i1}, I_{i2}, \dots, I_{iN_i}$) indicating which N_i database images have visited the particular node i , and the corresponding array of visiting frequency ($m_{i1}, m_{i2}, \dots, m_{iN_i}$).

There are some recent works to improve the performance of baseline SVT. One main shortcoming of SVT is that the SVT built from only frontal-view database images is insufficient in categorising query images which suffer from perspective distortion. Hence proposed Multiview Scalable Vocabulary Trees [41][42] was proposed to extend the single-view SVT to a set of multi-view SVTs that are simultaneously used for image classification. In their method, a projective transformation is applied to the frontal-view images to generate additional 4 images with different views, as shown in Fig.2.6. Then they extract local features from the images and build SVT for each view. Several SVTs are simultaneously employed for each query image. The top database candidates to emerge from each SVT are further subjected to pairwise comparison using a ratio test and geometric verification. Across

all views, the database image associated with the maximum number of post-ratio-test and feature matches after geometric verification is deemed as the best match.

Another work proposed by Xiaoyu Wang [43] improves the performance of baseline SVT by incorporating a descriptor contextual weighting (DCW) and a spatial contextual weighting (SCW) of local features. The DCW utilizes the frequencies of descriptors quantization paths on a vocabulary tree in one image, which lower the weights for less informative features. The SCW leverages spatial statistics for each local feature, including the density of its adjacent features and their orientations and scales.

Figure 2.6: Original frontal-view images and four generated images with other perspectives

2.4 Geometric Verification

After establishing matched feature pairs between the query image and the candidate database images, the next step is geometric verification (GV) [44]. Geometric verification is a way to verify if the obtained feature correspondences are consistent with a geometric configuration. By this, we mean that the locations of corresponding keypoints in the two images should be related by a common geometric transformation.

The random sample consensus (RANSAC) [45] is a common method to perform geometric verification. By using RANSAC, it is possible to increase the accuracy of the feature matching by evaluating the geometric relations between the query image and database image.

The process of RANSAC can be summed up as follows:

1. Sample a minimal subset of m matching pairs.
2. Estimate a transformation function $\mathbf{G}$ from these m pairs.
3. Check the consistency of all remaining pairs with the transformation function, and calculate the number of inliers I .

4. If $\mathbf{I} > \mathbf{I}^*$, where $\mathbf{I}^*$ is the number of inliers based on previous transformation function, store the new model $\mathbf{G}^* \leftarrow \mathbf{G}$, together with its number of inliers $\mathbf{I}^* \leftarrow \mathbf{I}$.
5. Repeat the above steps until the termination condition is met.

Inliers are the keypoints whose transformation error based on the estimated transformation function is smaller than a certain threshold. Fig.2.7 shows an example of GV on the query image and one candidate image. Fig.2.7(a) shows the feature matches on the SVT, while Fig.2.7(b) shows the consistent matches with the selection of RANSAC. We can see from the figure that many outliers are eliminated through geometric verification.

Figure 2.7: Matching pairs (a) before GV and (b) after GV

2.5 Summary

Significant progress towards visual search has been made in the past two decades through the development of local invariant features. Among existing local feature detectors, scale-invariant feature transform (SIFT) detector, proposed by Lowe [1] is one of the state-of-art algorithm, providing an efficient way for image recognition and retrieval. Besides, in order to address the issue when SIFT is not invariant to affine transformation and color/illumination changes, some SIFT variances including ASIFT, CSIFT are proposed recently to improve the performance of baseline SIFT. Further, some works try to boost the SIFT descriptor against illumination variation through creating additional samples in different illumination conditions and extracting the desired features of these conditions [38] [39]. However, none of them can solve all the problems of SIFT. For example, ASIFT only works to images that suffer from affine transformation and is ineffective in images under various

illumination transformations. And [37] [36] [38] [39] are only effective in images under illumination changes and cannot handle geometric transformation. Besides, for [38] [39], although they generate additional images to enhance SIFT in various illumination variations, each SIFT feature is treated with same weight ignoring the fact that they have different stability under various illumination conditions, which will lower the recognition rate. Similarly, in [40], the features are weighted only according to the corner degrees. Hence it also does not address the problem regarding to illumination and geometric distortions.

In the recognition phase, for all the SVT-based image recognition mentioned above, the local descriptors in the whole image are treated with equal importance. This restricts its performance as in most scenarios the local descriptors on the foreground object should be given more weights than the descriptors in the background. In view of this, we propose a new feature weighting algorithm in the next chapter: (1) By studying the effect of SIFT with respect to illumination and geometric variations, different scores are assigned to interest points according to their stability in the augmented samples of various transformations. Hence the stable interest points can be obtained. (2) the saliency information is incorporated into feature weighting, where more salient descriptors on the foreground are strengthened with larger weights.

Chapter 3

Transform and Saliency Feature Weighting in Image Recognition

3.1 Photometric Feature Weighting (PFW)

Studies have shown that Scalable Vocabulary Tree is effective in image recognition. The current SVT framework include: constructing a SVT using all representative training data. During online recognition phase, the feature descriptors of the query image are passed through the vocabulary tree, then the database images can be quickly scored according to inverted-index scheme.

In both SVT construction or inverted index process, the local descriptors in the whole image are treated with equal importance. This restricts its performance as some SIFT keypoints are not robust to lighting change. In view of this, the descriptors that are more robust to photometric transformation should be given more importance.

In this section, in order to address the problem, we proposed an image recognition framework that incorporates the new photometric feature weighting (PFW) method. Before going into the details of the new scheme, some techniques which are used to adjust lighting environment of digital images are introduced.

There exists three main ways to adjust brightness and contrast of a digital image [46]. For simplicity, we define g as a pixel value of an image after normalization, where $0 \leq g \leq 1$.

Figure 3.1: Gamma compression and expansion

Brightness Adjustment This method is used to generate linear brightness increment or decrement.

$$G_1(g, \beta) = \begin{cases} 1, & \text{if } g + \beta \geq 0 \\ g + \beta, & \text{if } 0 \leq g + \beta \leq 0 \\ 0, & \text{if } g + \beta < 0 \end{cases} \quad (3.1)$$

where β is the amount of increment or decrement of a pixel value. If β is positive, the image brightness is increased and vice versa.

Gamma Correction Gamma correction is the name of a nonlinear operation commonly used to adjust the image contrast:

$$G_2(g, \gamma) = g^{\frac{1}{\gamma}} \quad (3.2)$$

where γ is the gamma parameter. The process of utilizing $\gamma < 1$ is called gamma compression, which tends to make the image darker, especially for the low pixel values. On the contrary, the image is brightened when $\gamma > 1$ and the application of the expansive power-law nonlinearity is called gamma expansion.

Fig. 3.1 shows “Gamma compression” and “Gamma expansion” with $\gamma = 1/2.2$ and $\gamma = 2.2$ respectively.

Figure 3.2: Proposed image recognition framework

Contrast of the middle-tone pixel values adjustment This method is used to increase or decrease the contrast of the middle-tone pixel values of an image.

$$G_3(g, \alpha) = \frac{1}{1 + \exp^{-\alpha(g-0.5)}} \quad (3.3)$$

where α controls the degree of transformation.

The PFW is used to determine which interest points are more stable to various photometric transformations. By assigning different scores to interest points based on their repeatability in some generated samples of different illumination conditions, the stable interest points are obtained. Then these interest points are used in weighted scalable vocabulary tree (WSVT) for recognition task. The PFW recognition framework can be summerized in Fig. 3.2 .

The offline phase include:

1. Image generation. For each image in the database, we artificially generate p photometric-transformed images whose gamma values form a geometric sequence.
2. Feature extraction. Extract SIFT keypoints for all the p reference images.
3. Feature selection and weight assignment. Compare the similarity between each keypoint in the image and the keypoints in the p generated images. If the similarity meets the following two requirements, the keypoint in the image is considered as a repeatedly detected ones and

assigned weight w according to the number of repeatability, $w \in \{1, 2, \dots, p+1\}$. For example, a keypoint can be repeatedly detected in four generated images, keeping the original keypoint and the corresponding four transformed versions in the generated images, the five keypoints will be assigned score '5'. In the end, all the weights of the keypoints will be normalized to 0-1.

We define the similarity between the keypoint in the image and the keypoint in generated images based on two requirements:

Neighborhood Checking The spatial distance between two points is smaller or equal to 2 pixels.

Descriptor Similarity $\text{sim}(A, B) = \mathbf{A} * \mathbf{B} / (|\mathbf{A}| * |\mathbf{B}|) \geq 0.94$, where A and B represent two descriptors.

4. WSVT construction. Suppose M descriptors are kept for all the reference images after feature selection and weighting. All the obtained descriptors, $D = \{d_i\}, i = 1, 2, \dots, M$, and their weights, $W = \{w_i\}, i = 1, 2, \dots, M$, are used in WSVT construction. The algorithm considers the weight information associated with each SIFT descriptor during clustering computation. The WSVT algorithm are summarized in Algorithm 1.

Algorithm 1 WSVT construction

Input: SIFT descriptors $D = \{d_1, d_2, \dots, d_M\}$ and their corresponding weights $W = \{w_1, w_2, \dots, w_M\}$;
Output: A vocabulary tree of L levels with branch factor K consisting of K^L leaf nodes, $n^{l,h} \in T$, where $l \in \{0, 1, \dots, L\}$ indicates the level, $h \in \{1, 2, \dots, K^{L-l}\}$ indicates the node's index at l^{th} level;
1: Start with $l = L$, applying K-means clustering to partition all samples into K^{L-l} clusters. Each cluster $n^{l,h}$ contains samples $D^{l,h}, h \in \{1, 2, \dots, K^{L-l}\}$;
2: Update $n^{l,h}, h \in \{1, 2, \dots, K^{L-l}\}$ using the weight information. $n^{l,h} \leftarrow \sum_{d_h \in D^{l,h}} w_i d_i / \sum_{d_i \in D^{l,h}} w_i$;
3: For each cluster $n^{l,h}$, generate K finer clusters for $(l-1)^{th}$ level: $N = \{n^{l-1, (h-1)K+a}\}, a \in \{1, 2, \dots, K\}$;
4: Repeat the pattern assignment and cluster center update until there's no change of the cluster centers for level l ;
5: $l \leftarrow l - 1$, repeat the weighted K-means clustering until $l = 0$;

5. Weighted BoW histogram representation. After the SVT construction, the feature descriptors of the an image are passed through the vocabulary tree to find the nearest node. Each descriptor can be represented by (2.11) for query images or (2.12) for database images. Therefore the image can finally be represented as a sparse high-dimensional Bag-of-Words histogram H .

The traditional histogram model treats each descriptor as equally important. However, in our proposed method, the descriptors that are more stable to photometric transformation should

be given more importance in image histogram construction. For an image consisting of m descriptors $D = \{d_i\}, i = 1, 2, \dots, m$ with corresponding transform weights $W = \{w_i\}, i = 1, 2, \dots, m$, each descriptor d_i is passed through the weighted vocabulary tree at each level to choose its closest clusters. As such d_i is quantized in $T(d_i) = \{n^{l, h_i}\}_{l=1}^L, h_i \in 1, 2, \dots, K^{L-l}$. In order to suppress the unstable descriptors and enhance the more stable ones, weight information is incorporated in weighted BoW histogram construction as follows:

$$H(c) = \frac{m}{\sum_{i=1}^m w_i} \sum_{i=1}^m w_i \cdot I(c = \arg \min(\|n - d_i\|_2))_{n \in T} \quad (3.4)$$

where I is the indicator function in [8].

During online phase, the same feature weighting method described in 1–3 is applied to each query image. Then the weighted BoW histogram for a query image is also built by traversing through the WSVT. The similarity between a database image $\mathbf{H}_d$ and a query image $\mathbf{H}_q$ can be computed as:

$$\text{sim}(\mathbf{H}_q, \mathbf{H}_d) = \left\| \frac{\mathbf{H}_q \cdot \mathbf{f}}{\|\mathbf{H}_q \cdot \mathbf{f}\|} - \frac{\mathbf{H}_d \cdot \mathbf{f}}{\|\mathbf{H}_d \cdot \mathbf{f}\|} \right\| \quad (3.5)$$

where $\mathbf{f}$ is the inverted index vector in [8]. The shortlisted database images which have the highest similarity based on histogram comparison in (3.5) will be returned. Finally, Geometric verification (GV) [7] is applied to ensure the consistency of feature matches between the query image and database images. Based on the geometric transformation model estimated by random sample consensus (RANSAC), the plausibility of the matches is improved by rejecting inaccurate and inconsistent matches.

3.2 Geometric Transformation Feature Weighting (GFW)

SIFT is originally designed to be invariant to certain geometric transformation. However, in practice, the recognition performance is still subject to the actual condition. Some keypoints are more stable under transformation while others are less stable and can't be repeatedly detected. Hence similar to the feature extraction method in the previous section, in this part we will study the stability of the keypoints under various geometric transformation. In order to do that, we will also develop a Geometric transformation feature weighting (GFW) strategy.

In order to check the stability of the keypoints under different geometric transformation, we first need to artificially generate some new images under various geometric transformations.

Suppose the coordinate of a pixel in the original image is $[x, y]$, we can use a 2 by 2 matrix to perform the transformation:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.6)$$

In our experiment, we use five types of affine transformations, as shown in Fig. 3.3.

vertical stretching ($a_1 > 1, a_2 = 0, a_3 = 0, a_4 = 1$);

horizontal stretching ($a_1 = 1, a_2 = 0, a_3 = 0, a_4 > 1$);

shearing ($a_1 = 1, a_2 = 0, a_3 = \tan(\theta), a_4 = 1, 2\pi > \theta > 0$);

scaling ($a_1 > 1, a_2 = 0, a_3 = 0, a_4 > 1$);

rotation ($a_1 = \cos(\theta), a_2 = -\sin(\theta), a_3 = \sin(\theta), a_4 = -\cos(\theta), 2\pi > \theta > 0$)

Figure 3.3: (a) original image; (b) vertical stretching; (c) horizontal stretching; (d)scaling; (e)shearing; (f)rotation.

GFW is used to determine interest points' stability under various geometric transformations. The image recognition framework using the proposed feature extraction algorithm is also summarized in Fig. 3.2. First, we test the five transformations individually.

The recognition framework based on GFW includes:

1. Image generation. For each image, we artificially generate four different geometric-transformed images belonging for each transformation type.
2. Feature extraction. Extract SIFT keypoints for the original image and the generated images.

3. Keypoint back-projection. For each SIFT keypoint in the four generated images, we map its coordinate back to the original image using:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}^{-1} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \quad (3.7)$$

4. Feature selection and weight assignment. Compare the similarity between each keypoint in the image and the keypoints in four generated images after back-projection. If a keypoint is repeatedly detected in t generated images by checking the similarity described in Section 3.1, keep this keypoint and the t transformed versions. Meanwhile assigning weight w equal to $t+1$ for all the $t+1$ keypoints, $w \in \{1, 2, \dots, 5\}$. Next, normalize the weight w to be between 0-1.
5. Image recognition. Suppose M descriptors are kept for all the reference images after feature selection and weighting. All the obtained descriptors, represented as $D = \{d_1, d_2, \dots, d_M\}$ and their corresponding transform weights $W = \{w_1, w_2, \dots, w_M\}$ are used to construct WSVT in Algorithm.1. After that, each image is represented as a weighted BoW histogram, introduced in Section 3.1, to perform image matching.
6. Repeat step (1)–(5) until we get recognition rate for each transformations. Transformations can be combined together to further improve the image recognition performance. Suppose finally g additional reference images are generated for each image.

3.3 Transform and Saliency Feature Weighting (TSFW)

In the previous two sections we introduce two transform feature weighting algorithms based on photometric and geometric transformation respectively. In fact, the two algorithms can be combined to further improve the recognition rate. The feature weighting method is performed so that the obtained weighted descriptors are suitable for both types of transformation.

The feature weighting includes:

1. Feature extraction. Extract SIFT keypoints for the original image, the p photometric-transformed images and the g geometric-transformed images, obtained in the last two sections.
2. Keypoint back-projection. For each SIFT keypoint in the g generated images, we map its coordinate back to the original image using (3.7).

3. Feature selection and weight assignment. Compare the similarity between each keypoint in the image and the keypoints in the $p + g$ generated images. If a keypoint is repeatedly detected in t generated images by checking the similarity described in Section 3.1, keep this keypoint and the t transformed versions. Meanwhile assigning weight w equal to $t + 1$ for all the $t + 1$ keypoints, $w \in \{1, 2, \dots, p + q + 1\}$. Next, normalize the weight w to be between 0-1.
4. Image recognition. Suppose M descriptors are kept for all the reference images after feature selection and weighting. All the obtained descriptors, represented as $D = \{d_1, d_2, \dots, d_M\}$ and their corresponding transform weights $W = \{w_1, w_2, \dots, w_M\}$ are used to construct WSVT in Algorithm.1. After that, each image is represented as a weighted BoW histogram, introduced in Section 3.1, to perform image matching.

,

The traditional scalable vocabulary tree (SVT) is widely used in image recognition due to its good performance and high computational efficiency. However, in this representation, the local descriptors in the whole image are treated with equal importance. This restricts its performance as in most scenarios more importance should be assigned to the local descriptors on foreground objects than the descriptors on the background. Hence apart from assigning transform weights to local descriptors according to their stability in various transformations, we also incorporate saliency information with the proposed feature weighting to further improve the performance.

According to saliency map, the more salient descriptors on the foreground are given more weight than the descriptors on the background. A new bottom-up visual saliency model, Graph-Based Visual Saliency (GBVS) [9] was proposed to calculate saliency for an image. This approach includes two steps to generate a saliency map.

Firstly, an activation map A is produced with the given the feature map F . In GBVS, an importance factor is assigned to the directed edge from location (i, j) to location (p, q) :

$$e((i, j), (p, q)) = \left| \log \frac{F(i, j)}{F(p, q)} \right| * \exp \left(- \frac{(i - p)^2 + (j - q)^2}{2\alpha^2} \right) \quad (3.8)$$

where α is a free parameter. Then we can get the activation map A according to eigenvector computation algorithm based on graph theory.

Secondly, normalization followed by maps combination are performed to obtain the final saliency map S of the given image.

Fig. 3.4 shows a database image and its corresponding saliency map built by GBVS. It illustrates that the areas where the foreground object lies have higher saliency value, while the foreground is given lower saliency value.

Figure 3.4: (a) One image from database (b) The corresponding saliency map

The proposed transform and saliency feature weighting (TSFW) includes the following steps:

1. Image generation. For each image, we artificially generate the p new samples of different lighting conditions using gamma correction, and the g samples using various geometric transformation in (3.6) .
2. Feature extraction. Extract SIFT keypoints for the original image and all the generated images.
3. Keypoint back-projection. For each SIFT keypoint in the g geometric-transformed images, we map its coordinate back to the original image using (3.7).
4. Feature selection and weight assignment. Compare the similarity between each keypoint in the image and the keypoints in the $p + g$ generated images. If a keypoint is repeatedly detected in t generated images by checking the similarity described in Section 3.1, keep this keypoint and the t transformed versions. Meanwhile assigning weight w equal to $t + 1$ for all the $t + 1$ keypoints, $w \in \{1, 2, \dots, p + g + 1\}$. In the end, normalize the weight w to be between 0-1. Suppose M descriptors are kept for all the reference images after feature selection and weighting.

5. Saliency map generation. Produce a saliency map for each image according to GBVS in (3.8). The saliency values are normalized to between 0-1. Each local descriptor has its own saliency value in the saliency map, $S = \{s_i\}, i = 1, 2, \dots, M$.
6. Obtaining final weights. Multiplying the saliency values $S = \{s_1, s_2, \dots, s_M\}$ with the transform weights $W = \{w_1, w_2, \dots, w_M\}$ to obtain the final weights $W_f = \{s_1w_1, s_2w_2, \dots, s_Mw_M\}$.
7. Image recognition. All the obtained descriptors, represented as $D = \{d_1, d_2, \dots, d_M\}$ and their corresponding transform weights $W_f = \{w_{f_1}, w_{f_2}, \dots, w_{f_M}\}$ are used to construct WSVT in Algorithm.1. After that, each image is represented as a weighted BoW histogram, as introduced in Section 3.1, to perform image matching.

The whole image recognition framework based on TSFW is summerized in Fig. 3.2.

3.4 Post-processing Filters Feature Weighting (PFFW)

Figure 3.5: Survey on Instagram filter by Tarek Riman [2013]

With the popularity of image editing softwares and social media networking services such as Instagram and Facebook, a growing number of users tend to process digital images using various post-processing filters (e.g., “lomo-fi” and “early bird” in Instagram) before uploading them to express individual styles. Figure 3.5 and 3.6 show two representative surveys on Instagram filter popularity conducted by Tarek Riman, who is a professional Social Media Specialist, and a business internet

Figure 3.6: Survey on Instagram filter popularity by Brandon Gaille [2013]

marketing expert Brandon Gaille, respectively. The post-processed images can be quite different from the original ones in terms of color tone, contrast, saturation and etc. In view of the nonlinearity of these filters, applying SVT based method to recognize the processed images directly can't get satisfactory result. SIFT keypoints do not have equal stability under these filters. Hence the more stable keypoints should be given more weights. In this section, we study the effect of SIFT with respect to various post-processing filters and develop a feature weighting algorithm to incorporate the stability of SIFT and saliency information into weighted scalable vocabulary tree (WSVT) based recognition. The image recognition framework is illustrated in Fig. 3.7.

The post-processing filters includes the filters on Instagram, Photoshop and etc to change color tone, contrast and other factors of image style. The feature weighting method is to study the effect of SIFT with respect to various filters in Instagram.

The offline phase includes:

1. Image filtering and feature extraction. We process each image in the database with f filter effects in Instagram to get f additional samples. Adobe Photoshop action files created by Daniel Box is used to simulate the effects in Instagram. We show an example image and its variants processed by 3 popular filter effects in Fig. 3.8. Then we extract SIFT keypoints for

Figure 3.7: Post-processing filters feature weighting (PFFW)

Figure 3.8: From left to right, (a) the original image and (b)-(d) images processed by Lomo-fi, Xpro II, Nashville filter effects

all the $f + 1$ reference images.

2. Saliency map generation. Produce a saliency map for each image according to GBVS in (3.8). The saliency values are normalized to between 0-1. Each local descriptor has its own saliency value in the saliency map, $S = \{s_i\}, i = 1, 2, \dots, M$.
3. Feature selection and weight assignment. We compare the similarity between each keypoint in the image and the keypoints in the f generated images. The keypoint in the image is considered as repeatedly detected and assigned weight w according to the number of repeatability, $w \in \{1, 2, \dots, f + 1\}$ if the similarity meets the two requirements in Section 3.1. Suppose M descriptors are kept for all the reference images after feature selection and weighting.
4. Obtaining final weights. Multiplying the saliency values $S = \{s_1, s_2, \dots, s_M\}$ with the transform weights $W = \{w_1, w_2, \dots, w_M\}$ to obtain the weights based on post-processing filters feature weighting $W_p = \{s_1w_1, s_2w_2, \dots, s_Mw_M\}$.
5. Image representation. All the obtained descriptors, represented as $D = \{d_1, d_2, \dots, d_M\}$ and their corresponding weights $W_p = \{s_1w_1, s_2w_2, \dots, s_Mw_M\}$ are used to construct WSVT in Algorithm.1. After that, each reference image is represented as a weighted BoW histogram, as shown in (3.4), to perform image matching.

During online phase, the query images are processed by filters. A weighted BoW histogram for a query image is built with the saliency values using (3.4). The similarity between a database image $\mathbf{H}_d$ and a query image $\mathbf{H}_q$ can be computed by (3.5). Finally, geometric verification (GV) [7] is applied on the top-ranked images and the best-matched result is obtained.

In fact, since these three types of imaging conditions (photometric, geometric transformation and filter effect) can coexist in images, the integrated feature weighting method, i.e, the post-processing filters feature weighting (PFFW), photometric feature weighting (PFW) and geometric feature weighting (GFW) can be combined together to further improve the recognition performance for processed images. The same feature weighting method is performed on the f images after filter effects, p photometric-transformed reference images and g geometric-transformed reference images. All the weights $w \in \{1, 2, \dots, f + p + g + 1\}$ are normalized to be between 0-1.

3.5 Experimental Results and Setting

We perform experiments on a commercial product database, which include:

- 3882 reference images
- 333 test images
- 41 categories of commercial products.

Figure 3.9 and Figure 3.10 shows some reference images and query images from the database, respectively.

Figure 3.9: Sample of reference images from the commercial product database

Figure 3.10: Sample of query images from the commercial product database

The commercial products cover a wide range of common products, such as snacks and other daily supplies. The reference images are high quality images taken under good condition using HD camera. On the other hand, the test images are manually captured using mobile phone in different conditions, including illumination variation, cluttered backgrounds, partial occlusion, viewpoints and scale. For example, the first and second images in Figure 3.10 are captured with flash. The third and fourth

images are captured in dark lighting condition. Besides, the query images often contain complex backgrounds, which makes the image recognition more challenging. All images are resized to a height or width equals to 640 pixels.

In our experiments, a query image is considered as correctly recognized if its groundtruth category is the same as the category of the best matched image. The recognition performance is evaluated by recognition rate, which is defined as the percentage of query images recognized correctly. All experiments are performed on an Intel 2.0 GHz Quad Core computer with matlab implementation.

3.6 Experimental Results on Photometric Feature Weighting (PFW) Image Recognition

In this section, we test the recognition rate on the commercial product database based on the image recognition scheme in Section.3.1. Each experiment is run ten times to get the average.

In order to generate additional reference images after gamma correction to improve the recognition rate, we conducted some experiments to select appropriate gamma values for these generated images.

3.6.1 Recognition Rate for Different Gamma Values

First we start with three reference images(including the original one) whose gamma values form a geometric sequence. The three gamma values are $[r^{-1}, r^0, r^1]$, where r is the ratio of the geometric sequence. By changing the value of r , different sets of gamma values are tested and given in Figure 3.11. It's found that increasing the ratio value r leads to an improvement in the recognition performance. The recognition rate achieves the best at $r = 2.1$, and then becomes relatively stable afterwards.

3.6.2 Experimental Results on Different Number of Reference Images

From the experimental results in the Section 3.6.1, we can see that the value 2.1 is a good value for the geometric sequence of gamma values. Next we will conduct some experiments to decide how many reference images should be used. Starting with three reference images, represented as $[2.1^{-1}, 2.1^0, 2.1^1]$. Each time two additional reference images are added based on the selected ratio value 2.1. For example, when we have five reference images, their gamma values will be $[2.1^{-2}, 2.1^{-1}, 2.1^0, 2.1^1, 2.1^2]$, as shown in Figure 3.12.

The experimental results are shown in Figure 3.13. From Figure 3.13, we observe that the recognition rate keeps increasing to 0.892 when five reference images are used. However, as we include more reference images according to the geometric sequence, the performance stops to increase and becomes stable afterwards. Therefore the number five is chosen in the subsequent experiments.

Figure 3.11: Recognition rate for different gamma values

Figure 3.12: Five reference images with gamma value= 2.1^{-2} , 2.1^{-1} , 1, 2.1 , 2.1^2 respectively

Figure 3.13: Experimental results on different number of reference images

3.7 Experimental Results on Geometric Feature Weighting (GFW)

In this section, we conducted some experiments to select transformation types to include and the appropriate transformation parameters. We test the recognition rate on the commercial product database based on the image recognition method in Section 3.2.

3.7.1 Experimental Results for Individual Geometric Transformation

We first test five types of geometric transformation individually, with each type comprising of five reference images generated by changing the parameters of the geometric transform matrix $[a_1, a_2; a_3, a_4]$, as shown in (3.6).

- For vertical stretching, $a_1 = [1, 1.1, 1.2, 1.3, 1.4]$;
- For horizontal stretching, $a_4 = [1, 1.1, 1.2, 1.3, 1.4]$;
- For scaling, $a_1 = [1, 1.2, 1.4, 1.6, 1.8]$, $a_4 = [1, 1.2, 1.4, 1.6, 1.8]$
- For shearing, $\theta = [0^\circ, 2^\circ, 4^\circ, 6^\circ, 8^\circ]$;
- For rotation, $\theta = [0^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ]$;

The experimental results are shown in Table 3.1. The table shows that individual geometric feature weighting can improve the recognition performance when compared to the current SVT method [8] without feature weighting. Among the geometric transformations, scaling alone can offer the most improvement, followed by rotation, shearing, horizontal stretching and vertical stretching.

Geometric transformation	Recognition rate
SVT [8] with no feature weighting	0.850
SVT + Scaling	0.883
SVT + Rotation	0.881
SVT + Shearing	0.869
SVT + Horizontal stretching	0.859
SVT + Vertical stretching	0.846

Table 3.1: Experimental results for individual geometric transformation

3.7.2 Experimental Results Using Combined Geometric Transformation

Individual geometric feature weighting is effective for query images only suffering from one type of geometric distortion. However, various geometric distortions can exist in the query images. Hence in this section we performed some experiments on combined geometric feature weighting to further improve the recognition performance. The five transformations in Table 3.1 are gradually combined from the one that with the best improvement to the one with the lowest. The experimental results in Table 3.2 show that the combination of scaling and rotation can give the best improvement. The performance does not show clear improvement as more transformation types are included. That is probably because the most common conditions exist in the dataset consist of scaling and rotation. Compared with the SVT in [8], the proposed geometric feature weighting improves the recognition rate by 3.7%.

Geometric Transformation	Recognition rate
Scaling only	0.883
Scaling + Rotation	0.887
Scaling + Rotation +Shearing	0.882
Scaling + Rotation +Shearing + Vertical stretching	0.884
Scaling + Rotation +Shearing + Vertical stretching + Horizontal stretching	0.879

Table 3.2: Experimental results for combining various geometric transformation types

3.8 Experimental Results for Joint Photometric and Geometric Feature Weighting (JFW)

In the previous two sections experiments are performed on two transform feature weighting algorithms based on photometric and geometric transformation, respectively. In fact, these two algorithms can be combined to further improve the recognition rate. Figure 3.14 shows the experimental results based on the joint photometric and geometric transformation (JFW). It illustrates that the joint photometric and geometric feature weighting (JFW) improves the recognition rate to 0.897, better than performing GFW and PFW individually.

Figure 3.14: Experimental results for joint photometric and geometric feature weighting (JFW)

3.9 Experimental Results for Transform and Saliency Feature Weighting (TSFW)

Fig. 3.15 shows a database image and its corresponding saliency map built by GBVS [9]. Figure 3.16 shows the experimental results of incorporating saliency information with the feature weighting algorithm. It illustrates that the areas where the foreground object lies have higher saliency value, while the foreground is given lower saliency value. From the figure we can see incorporating saliency information can improve the recognition accuracy to 0.906.

A case study of the proposed method and other comparative methods is given in Figure 3.17. It include the SVT method [8] without feature weighting, the SVT plus saliency method and the proposed method. The recognition result for the SVT method [8] without feature weighting is incorrect in this case. Even though the SVT plus saliency [9] method enhance the descriptors with high saliency values and decrease background distractions, it fails to recognize the product correctly because the stability of descriptors under various transformations is ignored. Using the proposed feature weighting, the descriptors of high stability and saliency value are assigned higher importance, hence resulting in correct match in the proposed method.

Figure 3.15: (a) One image from database (b) The corresponding saliency map

Figure 3.16: Experimental results for transform and saliency feature weighting (TSFW)

Query image	Method	Best matched image	Result
SVT [8]		×
	SVT [8]+ Saliency [38]		×
	The Proposed method		√

Figure 3.17: Comparison of case study for the SVT [8] method without transformation, SVT [8] + saliency [9] method and the proposed TSFW method

3.10 Experimental results on post-processing filters feature weighting (PFFW)

In Section 3.4, we discussed due to the nonlinearity of the filters on image processing softwares, applying current SVT method to recognize the images processed by filter effects directly doesn't give satisfactory result. Hence we proposed the post-processing filters feature weighting (PFFW) in Section 3.4. In the proposed method we study the stability of SIFT keypoints under various filter effects and incorporate the obtained weights and saliency information into weighted scalable vocabulary tree (WSVT) to improve the recognition rate. In order to check the stability of SIFT keypoints, we need to generate additional images under different filters for each reference image in the training process.

In this section we conducted some experiments to determine the combination of filter effects used for training which can provide good recognition performance. In our experiments different combination of filter effects are evaluated to recognize query images processed by top 7 popular Instagram filters in Figure 3.5. These include Early bird, Xpro II, Brannan, Hefe, Nashville, Lomo-fi and Toaster. Table 3.3 shows the individual and average recognition rates of the seven filter effects under different training combinations. Each row refers to a combination of training filters and each column refers to query images with specific filter effect. For example, in row 3, we use only the Original, Early bird, and Xpro II effects for training. From top to bottom, one filter type is added progressively based on

the popularity survey in Figure 3.5. From the figure it is observed that we can obtain almost the best individual and average recognition rate when four types of filter effects, namely the Original, Early bird, Xpro II and Brannan are used for training. The recognition performance remain stable and does not change much even if more filter effects are incorporated in the training. Further, it is observed that the proposed method also works for test filter effect that is not included in training, eg. “Toaster” effect in the last column. When we only use original images for training without PFFW (i.e. row 1), the recognition rate for the query images with “Toaster” effect is 0.841. This improves to 0.884 when the original, Early bird, Xpro II and Brannan effects are used for training.

In order to show that the observed pattern is not restricted to a specific order and types of training filters, we reversed the order of adding progressive training filters and replace part of the filters in Table 3.4 and Table 3.5, respectively. In Table 3.4, contrary to Table 3.3, the training filter effects are added from the 7th popular filter to the 1st filter. In Table 3.5, some training filters and test filters including Hefe, Nashville and Lomo-fi are replaced with Toaster, Gotham and Walden. In the two tables we can see similar pattern: (1) The best individual and average recognition rate are obtained when four types of filter effects including the original image are used for training. (2) The proposed PFFW method not only can improve the recognition rate of query images processed with the same training filters, but it can also work for filter types that are not included in training, like “Toaster” in Table 3.4 and “Hefe” in Table 3.5. In summery, it is observed that once sufficient diverse filter types are incorporated in the training, the recognition results will become relatively stable and it does not depend greatly on the order of the training filter types so long as sufficient diversities are introduced. Overall, the proposed post-processing feature weighting method outperformed the current SVT [8] method without feature weighting by 10% in average recognition rate.

3.11 Experimental results on combined feature weighting

The images shared online not only experienced post-processing filters, but may also experience geometric and photometric transformation. Therefore we conducted some experiments to combine the three kinds of feature weighting (i.e. photometric feature weighting (PFW), geometric feature weighting (GFW) and post-processing filters feature weighting(PFFW)) to study the recognition performance. The experimental results of the combined feature weighting are shown in Table 3.6. From the table we can see that the average recognition rate of 0.884 is obtained using the PFFW with four types of filter effects in training, including Original, Early bird, Xpro II and Brannan. By combining PFFW with GFW and PFFW, both the individual and average recognition rate are improved. The average recognition performance can be further improved to 88.9% by combining PFW with PFFW and GFW.

Table 3.3: Experimental results for feature weighting on instagram filters, the training filters are added from the 1st popular filter to 7th popular filter

Training filter	Test filter							
	Early bird	Xpro II	Brannan	Hefe	Nashville	Lomo-fi	Toaster	Average
Original(baseline)	0.762	0.824	0.787	0.842	0.733	0.714	0.841	0.786
Original,Early bird	0.868	0.898	0.869	0.882	0.894	0.837	0.883	0.876
Original,Early bird,Xpro II	0.869	0.898	0.873	0.871	0.893	0.847	0.879	0.876
Original,Early bird,Xpro II, Brannan	0.875	0.902	0.876	0.884	0.897	0.838	0.884	0.884
Original,Early bird,Xpro II, Brannan,Hefe	0.875	0.897	0.875	0.883	0.894	0.841	0.884	0.878
Original,Early bird,Xpro II, Brannan,Hefe,Nashville	0.871	0.897	0.874	0.884	0.893	0.839	0.879	0.877
Original,Early bird,Xpro II, Brannan,Hefe,Nashville, Lomo-fi	0.872	0.894	0.872	0.878	0.890	0.840	0.879	0.875

Table 3.4: Experimental results for feature weighting on Instagram filters, the training filters are added from the 7th popular filter to 1st popular filter

Training filter	Test filter							
	Early bird	Xpro II	Brannan	Hefe	Nashville	Lomo-fi	Toaster	Ave
Original(baseline)	0.762	0.824	0.787	0.842	0.733	0.714	0.841	0.786
Original,Lomo-fi	0.863	0.893	0.864	0.877	0.889	0.841	0.878	0.872
Original,Lomo-fi,Nashville	0.876	0.904	0.882	0.889	0.903	0.847	0.892	0.885
Original,Lomo-fi,Nashville, Hefe	0.882	0.909	0.881	0.891	0.906	0.847	0.892	0.887
Original,Lomo-fi,Nashville, Hefe,Brannan	0.872	0.898	0.873	0.882	0.894	0.843	0.884	0.878
Original,Lomo-fi,Nashville, Hefe,Brannan,Xpro II	0.871	0.898	0.876	0.886	0.896	0.842	0.887	0.879
Original,Lomo-fi,Nashville, Hefe,Brannan,Xpro II, Early bird	0.869	0.892	0.873	0.884	0.889	0.842	0.883	0.876

Table 3.5: Experimental results for different feature weighting on Instagram filters, with part of filters including Hefe, Nashville and Lomo-fi replaced to Toaster, Gotham and Walden

Training filter	Test filter							
	Toaster	Gotham	Walden	Early bird	Xpro II	Brannan	Hefe	Ave
Original(baseline)	0.841	0.620	0.754	0.762	0.824	0.787	0.842	0.776
Original,Toaster	0.882	0.773	0.852	0.865	0.900	0.872	0.880	0.861
Original,Toaster,Gotham	0.887	0.776	0.863	0.874	0.902	0.872	0.888	0.866
Original,Toaster,Gotham, Walden	0.887	0.781	0.867	0.875	0.907	0.876	0.888	0.869
Original,Toaster,Gotham, Walden,Early bird	0.887	0.786	0.864	0.874	0.906	0.876	0.888	0.865
Original,Toaster,Gotham, Walden,Early bird,Xpro II	0.880	0.778	0.867	0.874	0.899	0.872	0.882	0.865
Original,Toaster,Gotham, Walden,Early bird,Xpro II, Brannan	0.874	0.776	0.853	0.869	0.889	0.842	0.883	0.855

Table 3.6: Experimental results for combined feature weighting on instagram filters

Feature weighting method	Test filter							
	Early bird	Xpro II	Brannan	Hefe	Nashville	Lomo-fi	Toaster	Average
PFFW	0.875	0.902	0.876	0.887	0.897	0.838	0.884	0.884
PFFW + GFW	0.881	0.902	0.881	0.894	0.901	0.849	0.893	0.886
PFFW + GFW + PFW	0.885	0.905	0.883	0.895	0.905	0.854	0.897	0.889

3.12 Conclusion

In this section, a new transform and saliency feature weighting algorithm is proposed. By giving different weights to SIFT descriptors according to their stability in various transformation, we can address the problem that some SIFT descriptors are more stable towards illumination changes and certain geometric transformation. Besides, we also incorporate saliency information with the transform feature weighting algorithm to further improve the recognition rate. Experimental results demonstrated that the recognition rate improves by 5% when compared with the current SVT recognition without feature weighting in [8].

Besides, the current SVT method can't address the issue of post-processing filter distortions in the query images. We proposed a feature weighting method based on post-processing filters and incorporate it into the proposed recognition framework. Experimental results on a commercial product database show the proposed algorithm outperforms the SVT recognition in [8] by more than 10% in average recognition performance for various post-processing effects.

Chapter 4

Augmented Visual Phrase (AVP) in Product Recognition

In order to handle photometric and geometric variations in query images, we proposed a feature weighting method in Chapter 3. In the proposed feature weighting method, augmented features are generated and incorporated into WSVT-based recognition. However, visual word in SVT-based method has limited descriptive and discriminative ability [20] since the spatial information is ignored. To utilize the augmented features to handle diverse imaging conditions, while integrating spatial information to make features more discriminative, we proposed a framework that uses Augmented Visual Phrases (AVP) in product recognition. We select AVPs from a pool of augmented features constructed by merging all the discriminative keypoints and their corresponding descriptors. These selected visual phrase are meaningful as they incorporate features carrying more diverse information.

4.1 Augmented Visual Phrases (AVP) Generation and Selection

The framework is summerized in Fig. 4.1. Broadly, there are two main phases: 1) Offline Phase; 2) Online Phase. In Offline phase, augmented features extracted from database trains vocabulary tree. Then AVPs are selected and indexed by 2D inverted index. In online phase, an unknown query image is recognized using AVP technique. A best matched image is retrieved and product is recognized. In order to train the system that can handle variations caused by illumination, geometric changes etc., we need to add features from such distorted images. To do so, we first produced the augmented pool of features for visual phrase selection. The augmented visual phrase selection include:

1. Image generation. For each image in the database, we artificially generate p augmented photometric-transformed images using gamma correction and g samples using various geometric transformation in (3.6).

Figure 4.1: Proposed Augmented Bag-of-Phrase Framework

2. Augmented features generation. We extract SIFT keypoints for all the $p + g + 1$ reference images. Figure 4.2 shows the extracted keypoints in the original image and two photometric-transformed images. We can see from these three images that there are many additional new keypoints detected (shown in red) in the photometric-transformed images (b) and (c). These new keypoints are not available in the original image (a), and they can impact the recognition results. We compare the similarity between each keypoint in the original image and the keypoints in the $p + g$ augmented images. If the similarity meets the following two requirements, the keypoint is considered as a repeatedly detected ones. Finally a pool of augmented features is produced.

Neighborhood Checking The spatial distance between two points is smaller or equal to 2 pixels.

Descriptor Similarity $\text{sim}(A, B) = \mathbf{A} * \mathbf{B} / (|\mathbf{A}| * |\mathbf{B}|) \geq 0.94$, where A and B represent two descriptors.

Figure 4.2: The extracted SIFT keypoints in the original image and in two images after photometric transformation. From left to right, the original image, the darkened image with gamma value 2.1^{-1} and brightened image with gamma value 2.1

3. SVT construction and quantization. A visual vocabulary tree (VT) [1] with depth factor 5 and branch factor 10 is constructed by applying hierarchical K-means clustering to the SIFT descriptors extracted from the all the $p + g + 1$ augmented images. The constructed vocabulary tree is denoted as $\mathbf{W} = \mathbf{w}_1, \dots, \mathbf{w}_N$ where $N = 100000$. The inverted index associated with the SVT is shown in Figure 4.3. The pool of augmented features for each image is quantized using

SVT to form augmented visual words which are used for augmented visual phrases (AVPs) generation.

Figure 4.3: A sample 1-D vocabulary tree with depth factor 2, branch factor 3 and the associated inverted index

4. AVP generation. There are a few recent works on visual phrase recognition. [47] use k nearest neighbors to form visual phrases. In [48], a grid-based local feature bundling method was used to locate visual phrases. However, these methods ignored the fact that visual phrases are sensitive to scale change and can be quite different in different scales. In our method, we define a circular neighboring region around each interest keypoint. A visual phrase can be formed by combining the central visual word and the visual words within the region. The radius r of the neighboring region is controlled by $r = scale * \lambda$, where scale parameter is the scale of the central keypoint, and λ is a parameter that determines the range of the neighborhood. Figure 4.4 gives an example of AVP formation, w_i and w_j are two visual words quantized by SVT. If w_j is within the neighboring circular region of w_i with a radius of $r_i = scale_i * \lambda$, an AVP, $\Omega = (w_i, w_j) | 1 \leq i, j \leq N$ is selected. Although increasing λ is helpful to keep more meaningful visual phrases, larger λ will also introduce spurious visual phrases and increase the

computational cost. In our experiments we set λ as 5.

Figure 4.4: An example of AVP formation

5. AVP selection. AVPs which have a consistently high occurrence across all image categories are not discriminative to perform image recognition. Hence we adopt a two stages procedure to re-rank the AVPs and eliminate the non discriminative ones using the criterion in (4.1), where $freq(i, c)$ is the frequency of visual phrase candidate i in category c and $freq(i, D)$ is the frequency of AVP i through all categories in the database. Finally, AVPs with high $Score(i, c)$ are obtained. In our experiments we set the threshold of $Score(i, c)$ as 0.8.

$$Score(i, c) = freq(i, c) / \ln(freq(i, D)) \quad (4.1)$$

4.2 AVP Indexing and Querying

In the original BoW-based recognition framework, we build a one dimensional inverted index for the visual words. In the proposed BoP-based recognition framework, since a AVP is comprised of two visual words, a two-dimensional inverted index is built for all AVPs. The whole process of the inverted index construction and online querying is illustrated in Figure. 4.5. If the index of visual word in SVT is from 1 to N , an index of $N * N$ dimension in which every element represents a possible AVP index will be built. The time complexity of adding a AVP into the matrix is only $O(1)$. However, the two dimensional index tends to be very sparse and it will lead to high memory storage. Therefore we first construct a one-dimension inverted index same as the index in BoW method. These entries

correspond to the first visual word in a AVP. For the second dimension, only non-empty entries that belong to one or more reference images are incorporated. Two visual words of each AVP are sorted in ascending order to prevent duplicate entry. The online time complexity of locating a visual phrase entry is $O(\log N)$.

In the proposed framework, BoP histogram intersection is used as the criterion to determine the

Figure 4.5: Inverted index of AVPs and online querying.

similarity between two images. The histogram intersection is given as:

$$f(D, Q) = \sum_{n=1}^M \min\{d^n, q^n\} \quad (4.2)$$

where D and Q represent two histograms, d_n and q_n are the n th components of the histogram of the database image D and the query image Q , respectively. Although the performance of BoP method is better than BoW-based recognition method, spurious visual phrases extracted from cluttered background will have a negative impact on the final recognition performance. Therefore we apply RANSAC-based geometric verification (GV) on the top ranked image candidates after BoP recognition to further improve the recognition accuracy.

4.3 Experimental Results on AVP Product Recognition

The experiments are performed on the same commercial product dataset introduced in Section. 3.5. In the experiments, SIFT is used to detect the keypoints. Augmented features are formed by selecting

discriminative features from the reference and transformed images. The pool of augmented features are used to construct a SVT with a branch factor of 10 and depth of 5, giving rise to $N=100000$ leaf nodes. Subsequently, an image is represented as a histogram of selected visual phrase using AVP method, and indexed using the method described in Section. 4.2. To handle spurious matches and obtain good match, we apply GV to top 50 ranked images. The performance comparison of the proposed AVP method with respect to the developed BoP method in [10] and the implemented SVT method in [8] is shown in Table. 4.1.

The proposed method achieves a recognition rate of 93.1% which outperform the SVT method by 10.1%. This improvement benefits from the advantage of the AVP method that incorporates the spatial information. Further, the proposed method outperforms the efficient BoP by 7.8%. Even though the efficient BoP method in [10] uses spatial information, it cannot handle query images with high photometric and strong geometric distortions. Overall, the proposed method can address these issues and achieves a good performance.

A case study for comparison of our proposed AVP technique against SVT [8] and BoP [10] is shown in Figure. 4.6. From the figure, it can be seen that although the query image is taken under different perspective and illumination condition as compared to the reference database image, the proposed method is able to recognize the product correctly, whereas the other methods fail.

Table 4.1: Performance comparison of the SVT [8], BoP [10] and proposed AVP method

Method	Recognition rate
SVT [8]	83.0%
BoP [10]	85.3%
Proposed AVP	93.1%

4.4 Conclusion

In this section, a recognition framework based on augmented visual phrase (AVP) is proposed. We created a pool of augmented features using generated transformed images that handle query images that suffer from various distortions and imaging conditions. Using spatial features to make feature more discriminative and using augmented features to handle variability in query images are crucial components of the AVP technique. AVPs are able to characterize query images and alleviate recognition problem due to the different imaging conditions. Experimental results shows the proposed AVP technique can achieve good recognition rate of 93.1% for the commercial product dataset.

Query image	Method	Best matched image	Result
SVT [8]		×
	BoP [9]		×
	Proposed method		√

Figure 4.6: Case study of AVP method with SVT [8] and BoP [10]

Chapter 5

Summary and Future Work

5.1 Summary

In this thesis, we first propose a new feature weighting algorithm: (1) By giving different weights to SIFT descriptors according to their stability in various transformation, we can address the problem that some SIFT descriptors are unstable due to illumination changes and certain geometric transformation. (2) We also incorporate the saliency information with the transform feature weighting algorithm, where more salient descriptors on the foreground are strengthened with larger weights. Experimental results have demonstrated that the recognition rate is improved by 5% when compared with the current SVT recognition [8].

Second, we proposed a post-processing filter feature weighting recognition framework to address the issue of post-processing filter distortions in the query images. Experimental results on a commercial product database show the proposed recognition algorithm outperforms the SVT recognition [8] by more than 10% in average recognition performance for various post-processing effects.

Besides, we proposed an AVP algorithm that can address the issue of photometric and geometric distortions in the query images. We generated augmented features using transformed images. Using spatial relationship to make the feature more discriminative and using augmented features to handle variability in query images, the proposed method is able to perform recognition of commercial product with different illumination and perspective changes. Experimental results shows the proposed AVP method achieves a recognition rate of 93.1%, and outperform other current methods.

5.2 Future Work

Effective feature representations and similarity measures are crucial to the content-based image recognition. Over the past decades, some image representation methods based on local feature descriptors (e.g. SIFT [1], and SURF [32], etc.) have been widely adopted in image recognition, such as the bag-

of-words (BoW) model [17] and scalable vocabulary tree (SVT). Recently deep learning, especially the deep convolutional neural networks (CNNs) has been reported to achieve state-of-the art performance for a variety of tasks in computer vision and pattern recognition. Deep learning refers to a family of machine learning techniques, where many layers of information in hierarchical architectures are exploited for feature or representation learning. We proposed to incorporate deep learning techniques for learning feature representations and distance measures in our proposed recognition framework. By exploring deep architectures to learn features at multiple level from images, an improvement in recognition rate can be expected.

Author's Publications

1. W. Zhang , K. Yap, D. Zhang and Z. Miao “Feature Weighting in Visual Product Recognition”, *IEEE International Symposium on Circuits and Systems, Lisbon, Portugal, pages 734-737, 2015.*
2. W. Zhang , A. Skaria, D. Manandhar, K. Yap and Z. Miao “Augmented Visual Phrase in Mobile Product Recognition”, *International Conference on Information, Communications and Signal Processing, Singapore, 2015 (ICICS 2015).*

Bibliography

- [1] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, Springer, 2004.
- [2] Reza Javanmard Alitappeh and Fariborz Mahmoudi. Mgs-sift: A new illumination invariant feature based on sift descriptor. In *International Journal of Computer Theory and Engineering*, volume 5, page 99. IACSIT Press, 2013.
- [3] Jae-Han Park, Kyung-Wook Park, MH Baeg, and Moon-Hong Baeg. π -sift: A photometric and scale invariant feature transform. In *19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.
- [4] Jean-Michel Morel and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. In *SIAM Journal on Imaging Sciences*, volume 2, pages 438–469. Citeseer, 2009.
- [5] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, Now Publishers Inc., 2008.
- [6] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, Springer, 2010.
- [7] Bernd Girod, Vijay Chandrasekhar, David M Chen, Ngai-Man Cheung, Radek Grzeszczuk, Yuriy Reznik, Gabriel Takacs, Sam S Tsai, and Ramakrishna Vedantham. Mobile visual search. *Signal Processing Magazine*, 28(4):61–76, IEEE, 2011.
- [8] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168, IEEE, 2006.
- [9] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. *Advances in neural information processing systems*, pages 545–552, Citeseer, 2007.

- [10] Dajiang Zhang, Kim-Hui Yap, and Sinduja Subbhuraam. Mobile product recognition with efficient bag-of-phrase visual search. In *2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pages 65–68. IEEE, 2014.
- [11] Yunpeng Li, David J Crandall, and Daniel P Huttenlocher. Landmark classification in large-scale image collections. In *12th international conference on Computer vision*, pages 1957–1964, IEEE, 2009.
- [12] David Chen, Sam S Tsai, Vijay Chandrasekhar, Gabriel Takacs, Jatinder Singh, and Bernd Girod. Robust image retrieval using multiview scalable vocabulary trees. In *IS&T/SPIE Electronic Imaging*, pages 72570V–72570V. International Society for Optics and Photonics, 2009.
- [13] Jing Li and Nigel M Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10):1771–1787, Elsevier, 2008.
- [14] IS Kweon, S Kim, C Park, Y Choi, and S Kwon. Feature point detection by combining advantages of intensity-based approach and edge-based approach. World Academy of Science, Engineering and Technology (WASET), 2012.
- [15] Zhenwei Miao and Xudong Jiang. Interest point detection using rank order log filter. *Pattern Recognition*, 46(11):2890–2901, Elsevier, 2013.
- [16] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584. Morgan Kaufmann Publishers Inc., 2001.
- [17] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, IEEE, 2007.
- [18] Kim-Hui Yap, Tao Chen, Zhen Li, and Kui Wu. A comparative study of mobile-based landmark recognition techniques. *Intelligent Systems*, 25(1):48–57, 2010.
- [19] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Ninth International Conference on Computer Vision*, pages 1470–1477, IEEE, 2003.
- [20] Shiliang Zhang, Qi Tian, Gang Hua, Qingming Huang, and Shipeng Li. Descriptive visual words and visual phrases for image applications. In *17th ACM international conference on Multimedia*, pages 75–84. ACM, 2009.

- [21] Yimeng Zhang, Zhaoyin Jia, and Tsuhan Chen. Image retrieval with geometry-preserving visual phrases. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 809–816. IEEE, 2011.
- [22] Tao Chen, Kim-Hui Yap, and Dajiang Zhang. Discriminative soft bag-of-visual phrase for mobile landmark recognition. *IEEE Transactions on Multimedia*, 16(3):612–622, IEEE, 2014.
- [23] Kim-Hui Yap, Tao Chen, Zhen Li, and Kui Wu. A comparative study of mobile-based landmark recognition techniques. *Intelligent Systems*, 25(1):48–57, IEEE, 2010.
- [24] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Fredrik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, Springer, 2005.
- [25] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2):151–172, Springer, 2000.
- [26] Jing Li and Nigel M Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10):1771–1787, Elsevier, 2008.
- [27] H. P. Moravec. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. PhD thesis, Stanford University, 1980.
- [28] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.
- [29] S. M. Smith and J. M. Brady. SUSAN - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, Springer, 1997.
- [30] P. R. Beaudet. Rotationally invariant image operators. In *International Conference for Pattern Recognition*, pages 579–583, IEEE, 1978.
- [31] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, Springer, 2004.
- [32] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Computer Vision - European Conference on Computer Vision*, volume 3951, pages 404–417. Springer, 2006.
- [33] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, volume 1, pages 384–393. Elsevier, 2002.

- [34] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. In *Image and Vision Computing*, volume 22, pages 761–767, Elsevier, 2004.
- [35] H. Deng, W. Zhang, E. Mortensen, T. Dietterich, and L. Shapiro. Principal curvature-based region detector for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [36] Jian Wu, Zhiming Cui, Victor S Sheng, Pengpeng Zhao, Dongliang Su, and Shengrong Gong. A comparative study of sift and its variants. *Measurement science review*, 13(3), 2013.
- [37] Qing Li, Junzhou Chen, Qiang Peng, and Xiao Wu. Application of localized soft-assignment coding and csift in image classification. In *International Conference on Internet Multimedia Computing and Service*, page 246. ACM, 2014.
- [38] Reza Javanmard Alitappeh and Fariborz Mahmoudi. Mgs-sift: A new illumination invariant feature based on sift descriptor. *International Journal of Computer Theory and Engineering*, 5(1):99, IACSIT Press, 2013.
- [39] Zhenyu Guo and Z Wang. An adaptive descriptor design for object recognition in the wild. In *IEEE International Conference on Computer Vision*, pages 2568–2575, IEEE, 2013.
- [40] Haidi Sheng, Huichuan Duan, and Chao Kong. Bag-of-words model for image classification based on harris corner features weighting. In *IEEE International Conference on Computational and information sciences*, pages 1284–1289, IEEE, 2014.
- [41] David Chen, Sam S Tsai, Vijay Chandrasekhar, Gabriel Takacs, Jatinder Singh, and Bernd Girod. Robust image retrieval using multiview scalable vocabulary trees. In *IS&T/SPIE Electronic Imaging*, pages 72570V–72570V. International Society for Optics and Photonics, 2009.
- [42] David Ebri Mars, Hanwei Wu, Haopeng Li, and Markus Flierl. Geometry-based ranking for mobile 3d visual search using hierarchically structured multi-view features. In *IEEE International Conference on Image Processing (ICIP)*, pages 3077–3081. IEEE, 2015.
- [43] Xiaoyu Wang, Ming Yang, Timothee Cour, Shenghuo Zhu, and Kai Yu. Contextual weighting and efficient re-ranking for vocabulary tree based image retrieval, November 18 2014. US Patent 8,892,542.
- [44] Bernd Girod, Vijay Chandrasekhar, David M Chen, Ngai-Man Cheung, Radek Grzeszczuk, Yuriy Reznik, Gabriel Takacs, Sam S Tsai, and Ramakrishna Vedantham. Mobile visual search. *Signal Processing Magazine, IEEE*, 28(4):61–76, 2011.

- [45] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [46] Jiayuan Fan, Hong Cao, and Alex C Kot. Estimating exif parameters based on noise features for image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 8(4):608–618, IEEE, 2013.
- [47] Junsong Yuan, Ying Wu, and Ming Yang. Discovery of collocation patterns: from visual words to visual phrases. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, IEEE, 2007.
- [48] Yuning Jiang, Jingjing Meng, and Junsong Yuan. Grid-based local feature bundling for efficient object search and localization. In *18th IEEE International Conference on Image Processing (ICIP)*, pages 113–116, IEEE, 2011.