Queen Mary
University of London

School of Electronic Engineering and Computer Science

Scalable Image Retrieval based on Hand Drawn Sketches and their Semantic Information

Thesis submitted to the University of London in partial fulfillment
of the requirements for the degree of Doctor of Philosophy.

Konstantinos Bozas

Supervisor: Prof. Ebroul Izquierdo

London, November 2014

Declaration

I hereby declare that this dissertation is entirely the result of my own work, it arises out of my own research, and I have made full acknowledgments of the work and ideas of any

Konstantinos Bozas

Acknowledgments

First, I would like to thank my supervisor Prof. Ebroul Izquierdo who has given me the opportunity and freedom to pursue my research ideas. I would also like to thank all my colleagues and friends from the Multimedia and Vision group that made our lab a warm and inspiring environment.

Special thanks go to my friends in London and Greece, Thanasis, Ioannis, Giannis, Nikos, Lemonia, Sander and Zoe for the great times we had together that made these last 4 years unique and enjoyable.

My gratitude goes to my parents, aunt and grandmother who haven't stop unconditionally supporting me all these years. To them I dedicate this work.

Abstract

The research presented in this thesis aims to extend the capabilities of traditional content-based image retrieval systems, towards more expressive and scalable interactions. The study focuses on machine sketch understanding and its applications. In particular, sketch based image retrieval (SBIR), a form of image search where the query is a user drawn picture (sketch), and free-hand sketch recognition. SBIR provides a platform for the user to express image search queries that otherwise would be difficult to describe with text. The research builds upon two main axes: extension of the state-of-the art and scalability. Three novel approaches for sketch recognition and retrieval are presented. Notably, a patch hashing algorithm for scalable SBIR is introduced, along with a manifold learning technique for sketch recognition and a horizontal flip-invariant sketch matching method to further enhance recognition accuracy.

The patch hashing algorithm extracts several overlapping patches of an image. Similarities between a hand drawn sketch and the images in a database are ranked through a voting process where patches with similar shape and structure configuration arbitrate for the result. Patch similarity is efficiently estimated with a hashing algorithm. A spatially aware index structure built on the hashing keys ensures the scalability of the scheme and allows for real time re-ranking upon query updates.

Sketch recognition is achieved through a discriminant manifold learning method named Discriminant Pairwise Local Embeddings (DPLE). DPLE is a supervised dimensionality reduction technique that generates structure preserving discriminant subspaces. This objective is achieved through a convex optimization formulation where Euclidean distances between data pairs that belong to the same class are minimized, while those of pairs belonging to different classes are maximized.

A scalable one-to-one sketch matching technique invariant to horizontal mirror reflections further improves recognition accuracy without high computational cost. The matching is based on structured feature correspondences and produces a dissimilarity score between two sketches.

Extensive experimental evaluation of our methods demonstrates the improvements over the state-of-the-art in SBIR and sketch recognition.

Contents

Glossary

2D	2-Dimensional
3D	3-Dimensional
AP	Average Precision
BoF	Bag-of-Features
BoW	Bag-of-Words
CBIR	Content Based Image Retrieval
CSS	Curvature Scale Space
DPLE	Discriminant Pairwise Local Embeddings
DSSD	Diffusion Distance Shape Descriptor
DT	Distance Transform
DWT	Discrete Wavelet Transform
Edgel	An edge pixel
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
IR	Image Retrieval
KL	Kullback-Leibler
LBP	Local Binary Patterns
MAP	Mean Average Precision
MDS	Multi-Dimensional Scaling
MST	Minimal Spanning Tree
NHI	Normalized Histogram Intersection
OCM	Oriented Chamfer Matching
PHOG	Pyramidal Histogram of Oriented Gradients
SBIR	Sketch Based Image Retrieval
SHOG	Structural HOG
SMO	Sequential Minimal Optimization
SSD	Self Similarity Descriptor
SVM	Support Vector Machines

Chapter 1

Introduction

Contents

1.1 Motivation

The exponential growth of publicly available digital media during the last two decades has highlighted the need for efficient and user-friendly techniques to index and retrieve images from large multimedia databases. Nowadays, the utility of scalable retrieval algorithms manifests bolder than ever. Despite the considerable progress of content-based image retrieval (CBIR) [35, 116], where the goal is to return images similar to a user-provided image query, most of the multimedia searches are traditionally text-based (e.g. Google Images, YouTube, Bing Images). Currently, out of all the big web image search engines only Google provides the possibility to search by image. Text-based image search requires user intervention to tag all the available data and has two main drawbacks: i) Image labeling is a laborious, time consuming task and most importantly subjective ii) Images cannot be succinctly communicated based on words; different people would probably use different words to describe a scene based on their cultural background and experience. On the contrary, CBIR techniques allow effortless tagging and export non-biased image summaries, but suffer from the so-called semantic gap. That is the discrepancy between human and computer representation of knowledge on a topic.

Endeavors to bridge the semantic gap led to extensive research on feature extraction and relevance feedback methods for CBIR, while application-oriented aspects such as interface, visualization, scalability, and evaluation have traditionally received lesser consideration [35]. To remedy this imbalance, sketch based image retrieval (SBIR) emerged. Frequently, users are looking for an image without having any related image available. Therefore, they need a natural way to express their query. Consider the example in Figure 1.1, the users have a particular image on their mind, which cannot be easily expressed with text. If they attempt to make a text-based image search using the keyword *mountains*, the results will be very generic and time needs to be spent browsing the collection for desired images. In this scenario, a sketch query consists a straightforward and intuitive way to describe the users' thoughts to the machine. Obviously, a detailed rendition of the query requires artistic skill. A more convenient way, which this work adopts, is to sketch the main feature lines of a shape or scene. The sketch can be drawn using the mouse of a personal computer or the touch screen of a modern mobile phone. Furthermore, recent studies have shown that lines are drawn along contours [29] and line drawings can encode certain shapes almost as well as shaded images [30].

The intuitive sketch generation process can be effortlessly ported to other platforms apart from personal computers, especially smart phones and tablets offer a fruitful market and their users embrace new technologies. Drawing on a touch screen device is admittedly easier and the produced sketches more accurate than those of the traditional mouse interfaces [46]. Despite the appealing advantages, SBIR research faces many open questions. Current approaches are mostly based on successful CBIR models and while promising results have been published, there is still room for vast improvements. The main challenges of SBIR are the discrepancy between color images, which contain rich information combined with noise from various sources, and terse sketch drawings, as well as recognition of common drawn sketch symbols human often use to represent common objects (i.e. a stick-man instead of a realistic human figure). The latter plays an important role to discard semantically irrelevant results

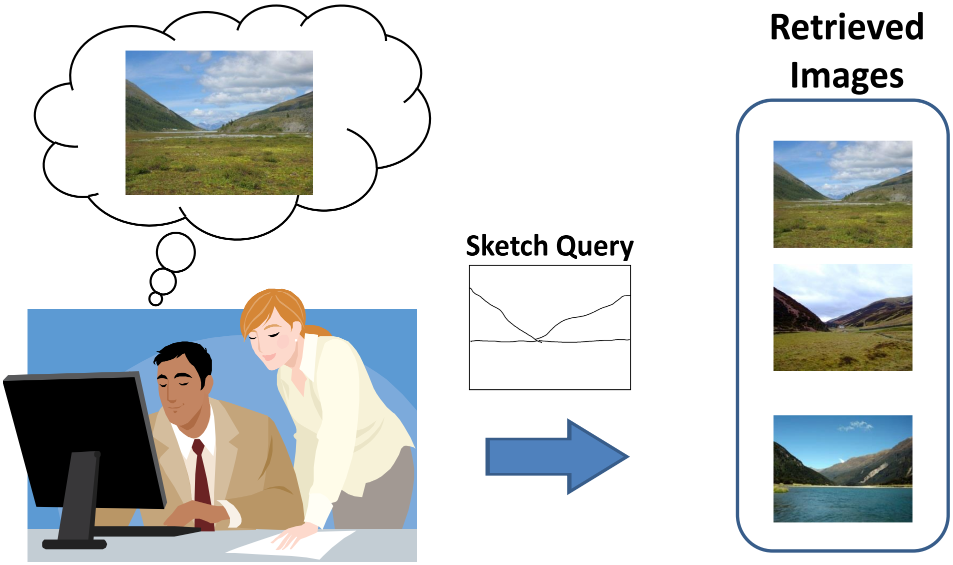


Figure 1.1: What will be the appropriate text query for the above search scenario? A sketch query is more expressive in this case.

that the retrieval algorithms conceive as visually similar, thus enhancing the overall quality of the top ranked retrieved images. Moreover, it bridges the modality gap between images and sketches, as many sketched symbols do not have one-to-one correspondence with photos. This thesis addresses both these problems by presenting novel scalable techniques achieving state-of-the-art results.

This chapter first introduces the concept of content based image retrieval in Section 1.2, which lays down the foundations of SBIR . Next, the objectives of the thesis are presented and a framework is developed upon them in Section 1.3. Section 1.4 summarizes the contributions of this work. Finally, Section 1.5 gives an overview of the organization of the thesis.

1.2 Content Based Image Retrieval

Content based image retrieval attempts to establish visual similarities between an image collection and an image query. The word content on the definition points out that the search is based on image features and excludes metadata,

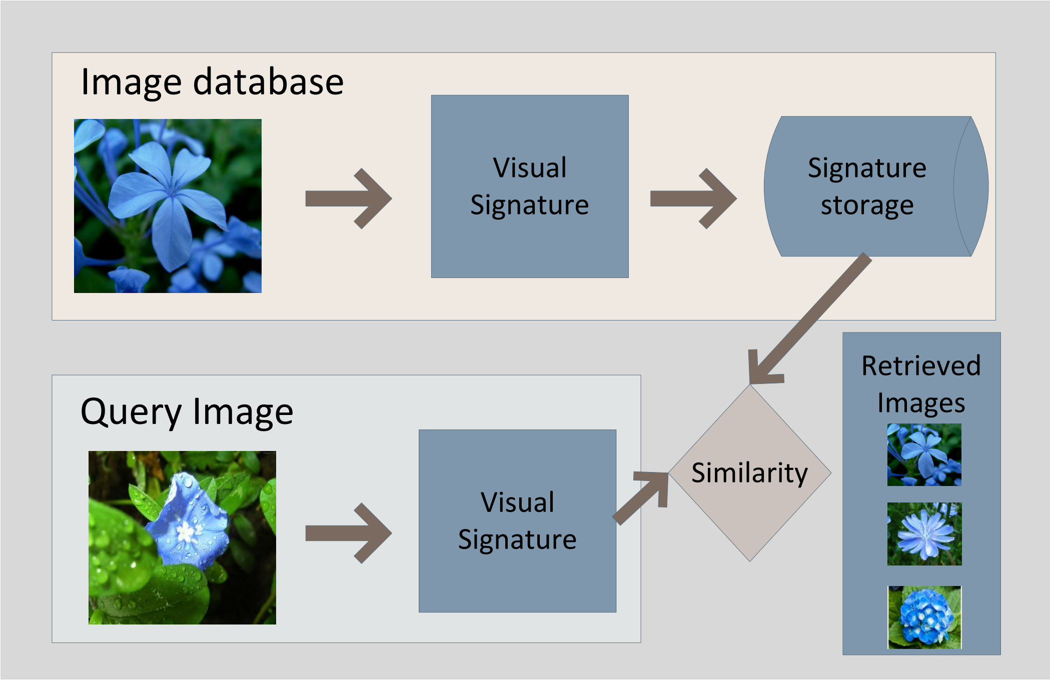


Figure 1.2: Generic schema of content based image retrieval

such as tags or labels. As image feature is considered any information we can extract from an image. Some of the most common modalities are *color*, *shape* and *texture* and have been extensively used in the literature [35, 116].

A generic schema of a content-based image retrieval system is outlined in Figure 1.2. There are two stages, first in the offline stage (top of Figure 1.2) a visual signature is extracted from all the images of the database and subsequently stored in an index structure. In the online or query stage, a new unknown image arrives, usually provided by the user. Again, its visual signature is extracted and compared against the database. A similarity function is in charge of generating a ranking of the database images. Both the visual signature and the similarity function can vary depending on the application requirements. A fundamental assumption of the feature extraction process is the expectation that images with similar content will produce similar signatures and vice-versa.

There is a vast range of fields where CBIR have been applied. Medical imaging [93], painting retrieval [20] and digital forensics [26] are a few examples. Google recently introduced a generic CBIR system for web image search

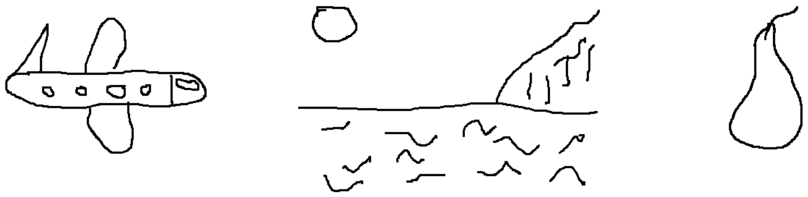


Figure 1.3: Sketch query examples.

[1], where users can upload an image query. Google Goggles is a commercial CBIR application that can retrieve landmarks and logos. It follows from the above that CBIR is a well-established research field, yet in the frequent case when a user seeks a particular image an expressibility barrier rises. Specifically, there is no specified input mechanism to describe the aforementioned image, hence the search is rendered void. Sketch based image retrieval provides a solution with an interactive query generation approach.

1.3 A Semantic SBIR Framework

Sketch is defined as a rough or unfinished drawing or painting, often made to assist in making a more finished picture [115]. In the case of SBIR, sketch is a hand-drawn rendition provided by the user with the purpose of finding a range of visually similar images. Researchers have been experimenting with various sketch types, color sketches [60, 61], shaded sketches or even combinations of drawing and text [78, 19]. A detailed discussion of these approaches can be found in the state-of-the-art chapter. In this work, we adopt sketches that are simple curve drawings and define the boundaries of an object or a scene. Some examples are illustrated in Figure 1.3. This design path is primarily taken to keep the query generation process simple and user friendly. However, our decision is also based on findings of [29, 30] which state that rough contour drawings can describe shape equally well to shaded sketches.

By slightly modifying the generic CBIR case, an appearance-based SBIR system can be developed. The modification takes place in the offline stage by adding an edge detection step prior to visual signature extraction. Each

[1] http://www.google.com/insidesearch/features/images/searchbyimage.html

image is subject to edge detection in an attempt to bridge the modality gap with the sketch queries. Besides that, the functionality remains the same. It becomes quickly evident that there are performance limitations on a SBIR system solely based on content similarities. Major contributors to that are a) the modality gap between images and sketches; b) the noise introduced by the edge detection process; c) the quality of the provided sketch query; d) the lack of semantic interpretation of the drawn sketches. The first two issues are handled by the feature extraction and dissimilarity evaluation techniques, the third is out of the application's control; unless a drawing helper tool is added to the loop [72]. Finally, the fourth can be tackled by infusing semantics priors into image retrieval.

Semantic retrieval attempts to improve image search accuracy by switching the focus from low-level image properties to higher-level concepts that can be deducted. Image descriptors capture observable properties of a photo, like color or shape, but they can't relate these properties with the search concept of each user. For example, a user may provide an image query of a tree with green leafs on a field of grass. An appearance-based system will most likely return similar images rich in green tones, but what if the user intended to look for different color variations of the tree? In that case, we need to perform object recognition to the query image to gain understanding of what is being depicted and return a greater variety of trees to the user. Liu *et al.* [79] identify five major approaches to reduce the semantic gap: (1) using object ontology to define high-level concepts; (2) using machine learning methods to associate low-level features with query concepts; (3) using relevance feedback to learn users' intention; (4) generating semantic templates to support high-level image retrieval; (5) fusing the evidences from HTML text and the visual content of images for WWW image retrieval. Most of the literature follows the second category. A successful CBIR framework should smoothly integrate high-level semantics into the retrieval process.

This work introduces a semantic SBIR framework. It consists of a sketch recognition component and a content retrieval component. An overview is presented in Figure 1.4. First, the sketch recognition module attempts to categorize the incoming sketch. Successful recognition lies in the understanding of human sketch drawing mechanics. Key properties of the recognition module

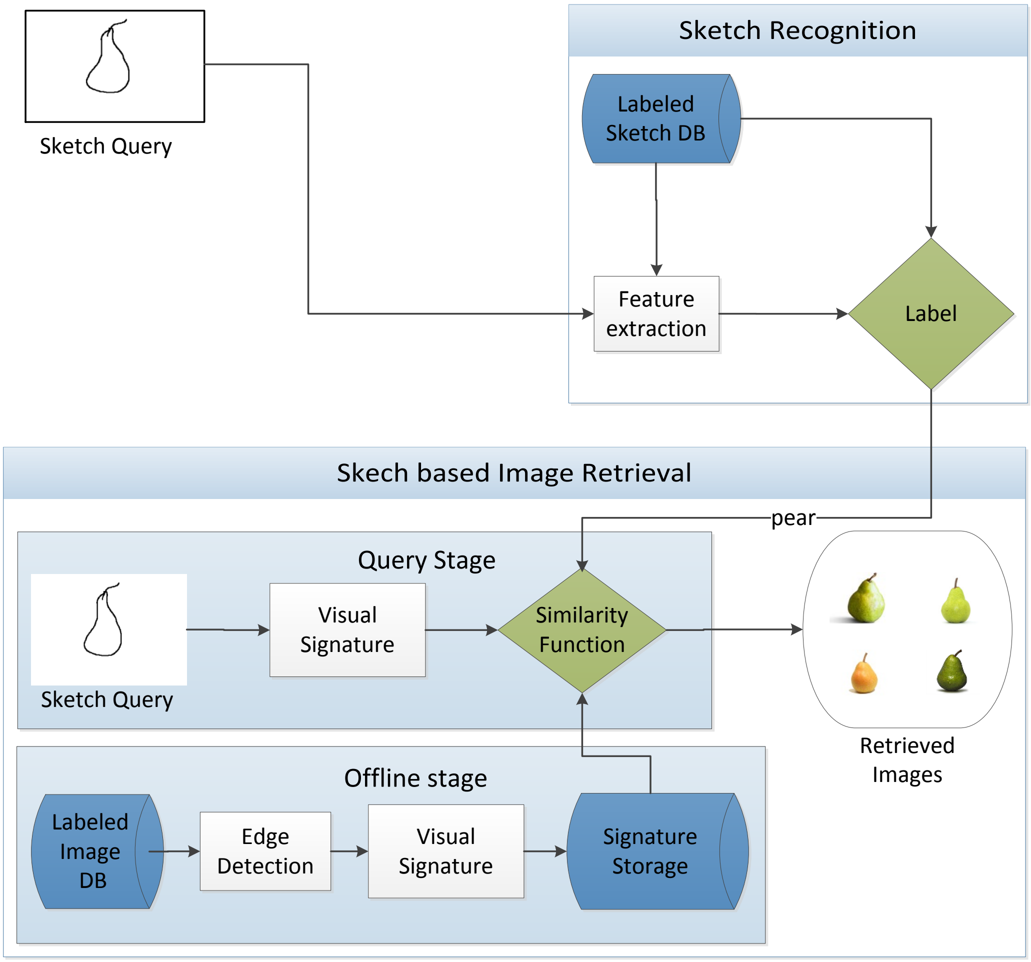


Figure 1.4: Schema of our proposed semantic sketch based image retrieval framework

are a large, diverse collection of human sketches and a robust machine learning algorithm. Chapters 4 and 5 investigate this problem. The recognition step produces a categorization of the input sketch. Using this knowledge, the returned results can be filtered accordingly to include only images that belong to the same category as the sketch. Obviously, a labeled image database is required. It can be readily obtained by crawling the web or using an off-the-shelf solution. The content retrieval component establishes visual similarities between database images and sketches and generates a ranking as in CBIR. Both components should be scalable, robust and able to be employed in an online SBIR environment. The next section overviews the contributions of this thesis towards the aforementioned goals.

1.4 Contributions

The primary aim of this work is to propose insights for a scalable, semantic sketch retrieval framework as described in the previous section. Within this framework, several contributions are identified.

- In Chapter 3, we present a scalable appearance-based approach to SBIR. Our method operates on local image regions and estimates shape similarity via a series of hash functions. Moreover, a location-aware reverse index enables look-ups in constant time during the query stage. The final rankings are generated through a voting process which enforces holistic structure constraints. We demonstrate the superiority of our technique over the state-of-the art in three benchmark evaluations. We also show that our approach can scale better in large volumes of images than other methods in literature.

- In Chapter 4, we introduce Discriminant Pairwise Local Embeddings (DPLE), a supervised manifold learning algorithm for sketch recognition. The main idea is to learn a discriminant subspace where the data will be better separated than in the original input space, without violating much its local neighborhood. The latter ensures that the data will maintain their manifold structure in the learned subspace, so classification algorithms can generalize better. We form these goals in a convex optimization problem that can be efficiently solved through eigendecomposition. A kernelized version is also explained to further enhance classification accuracy. Experiments in a large multi-class sketch database demonstrate the advantages of our technique over similar dimensionality reduction algorithms. We further show the generalization efficiency of DPLE in a face recognition problem.

- In Chapter 5, we present a horizontal flip-invariant sketch matching technique. A dissimilarity score between two sketches is generated based on feature and structure similarity of local patches. The flip-invariance property is shown to offer superior results over non flip-aware methods. Contrary to traditional slow matching techniques our approach is able to evaluate a large number of sketch pairs in real-time. The generated sketch rankings can be employed to facilitate sketch recognition with state-of-the-art accuracy. The projected labels are exploited in a semantic SBIR framework that drastically improves retrieval quality.

The contributions were published in two international conferences. One more conference submission is curretnly under review and ACM Transactions journal manuscript is currently under major revisions. See also the Publications section for the full listing.

1.5 Thesis Overview

The remainder of the thesis is organized as follows. Chapter 2 offers an overview of the theory and evaluation methodology in image retrieval and an thorough review of the state-of-the-art in SBIR. Chapter 3 describes in detail our scalable patch hashing approach for SBIR. Chapter 4 presents the theory and evaluation of a novel supervised manifold learning technique, namely Discriminant Pairwise Local Embeddings (DPLE), which is employed to recognize objects in sketches. Chapter 5 describes a horizontal flip-invariant sketch matching algorithm that further enhances recognition accuracy. In this chapter, our semantic SBIR framework is experimentally evaluated. Chapter 6 summarizes the thesis and offer conclusions and future work suggestions. The list of the author's publications is given at the end of the thesis.

Chapter 2

State of the art

Contents

2.1 Fundamentals of Image Retrieval

In this section, we give an overview of the fundamental components of an image retrieval system. By the nature of its task, CBIR technology boils down to two intrinsic problems: a) how to mathematically describe an image, and

b) how to assess the similarity between a pair of images based on their abstracted descriptions [35]. Roughly speaking, a visual signature is extracted from the query image and then compared against all the precomputed signatures of an image database. The original image representation of pixels with different intensities is complex and noisy and does not include image semantics. Therefore, the need for a mathematical algorithm to represent an image based on key visual properties like color, texture, shape, is imminent in image retrieval (IR). The produced feature vectors or visual signatures should fulfill a number of requirements:

- **Repeatability**: Related images should return similar visual signatures even after changes in illumination scale, viewpoint etc.
- **Encoding of desired visual properties:** Depending on the domain of the IR application, features should be able to capute key visual characteristics of an image such as shape, texture, color etc.
- **Small memory footprint**: Hundreds of thousands of candidate images must be loaded to the computer memory to be compared with the query, hence a visual signature should be kept as small as few kilobytes.

An image description algorithm could be inspired from the human vision mechanics and at the same time generate a semantic description of the image. The latter is analogous to the functionality of the optical center of the brain, which receives information from the optical nerve and semantically interprets it. Marr's study [88] have shown evidence supporting that the Laplacian of Gaussian (LoG) function models adequately the human visual system response and many feature extraction algorithms adopted this filter. Gabor function sets [43, 64] provide models of the human visual cortex area. Other methods, such as the Wavelet Decomposition [85], has been shown to match well the dyadic structure of receptive fields in the human primary visual cortex [5, 36]

The major properties of an image are color, shape, texture and saliency. These features are utilized either in a global extraction framework, where a single signature is extracted from the image, or in local extraction schemes where several feature vectors describe small individual pixel neighborhoods. Color features capture dominant color layout in an image, texture features are intended to capture granularity and repetitive patterns in a picture. For example, a plain white wall has uniform texture while the patterns on a brick wall result in high vertical and horizontal spatial frequencies. As a result, we can easily differentiate these two objects in terms of texture. Texture allows us to infer on the properties of objects in an image, thus contributes to the reduction of the semantic gap. Shape description algorithms aim to discover shape traits in an image. Recent shape descriptors perform well even under affine or non-rigid transformations [92, 34, 77]. Shape is the key feature for a sketch based image retrieval system due to the lack of other properties in binary drawings, so in the following sections we will further elaborate on shape description techniques tailored for the particular problem of SBIR. Features based on salient points or corner points are usually extracted on local pixel neighborhoods and can deal with significant affine transformations and illumination distortions.

Similarity measurement is also crucial to CBIR. Datta *et al.* [35] identify a large number of fundamentally different measurement frameworks that have been proposed over the years. The major prenciples behind the design of image similarity models are summarized below:

- robustness to noise.
- computational efficiency.
- agreement with image semantics.

Obviously, the outcome of the extraction algorithm constrains to some extend the applicability of specific similarity measurement techniques. For instance, if the outcome of the extraction algorithm is a vector descriptor one may apply the Euclidean dissimilarity measurement. In the case of probability distribution data a preferable approach could be an entropy-like measurement such as the Kullback-Leibler (KL) divergence. Table 2.1 illustrates some of the most popular distance metrics for similarity measurement as noted in [35]. Every measurement has its own advantages and disadvantages; simple methods are easily implemented and efficiently computed, but they cannot cope with challenging data. More complex techniques offer robustness but might be prohibitively slow for some applications. The choice of the optimal method is dictated by the nature of the feature extraction algorithm and the type of retrieval system.

2.1.1 Extraction of Visual Signature

2.1.1.1 Global Feature Extraction

During the early years of CBIR global feature extraction was the dominant technique to describe an image [116, 33, 61, 112]. In this approach, an algorithm describes a picture according to globally measured properties, like the total number of line segments or circles, the prevailing texture direction or the dominant colors. These simplistic methods offer a generic representation of the image and cannot distinguish between local variations. In many cases, global similarity is not enough and results in lots of false positives due to images sharing some generic characteristics but their semantic information is divergent. On the other hand, global features are computed efficiently and their memory footprint is adequately small, since we describe the whole image with just one feature vector. They can be used in a CBIR system to boost the performance or eliminate outliers. Let's suppose a query image depicts an animal in a forest; a hierarchical clustering algorithm (e.g. agglomerative clustering) based on a global color distribution will help us eliminate database images where green is not the dominant color (e.g. desert or water images), hence the candidate images will be significantly pruned. On the contrary, the global color distribution alone provides little help to identify the animal in the image.

Global feature extraction can be still useful in specific problems, hence research interest has not totally concentrated on local methods. Theoharatos *et al.* [122] proposed a graph-theoretic description of an image. Their representation consists of the minimal spanning tree (MST) of the graph created by sampling pixels in the RGB space. The MST structure is unchanged under transformations like translations, rotation and relative insensitive to small amounts of noise. They also suggest a novel statistical similarity measure, the Multivariate Wald-Wolfowitz test to assess whether two multidimensional point samples $\{X_i\}_{i=1,\dots,d}$ coming from the same multivariate distribution. Liu *et al.* [80] recently published the Diffusion Distance Shape Descriptor (DDSD) for comparing 3D shapes of molecules. The diffusion distance [28] is calculated for every pair of points in a sample set, offering robustness to non-rigid deformation and a histogram of the probability distribution of diffusion distances between sample point pairs is the output of the algorithm. The DDSD was successfully employed to molecule shape retrieval.

Global features have small computational costs and low memory requirements, but they lack of high descriptive capabilities. As a result, modern CBIR approaches adopt local features as their core visual signature extraction algorithm, while global features play a support role in the process.

2.1.1.2 Local Feature Extraction

In local feature extraction, a set of properties is computed at each image pixel using its adjacent pixels. A visual example of the local neighborhood of a pixel is given in Figure 2.1. Since CBIR similarity evaluation algorithms traditionally require a single visual signature to describe an image, the local vectors of each neighborhood are usually concatenated to form a global feature vector or combined together under a weighting scheme. Obviously, the computational cost to calculate a descriptor for every pixel is in most cases restrictive. As we will discuss further in this section there are techniques to reduce this computational burden.

The information extracted from each pixel's neighborhood enables us to obtain rich image representations. Consider again the previously mentioned example of an image depicting an animal in a forest. This time let's assume that a local feature extraction algorithm is applied to extract the mean color value of every pixel neighborhood. The returned visual signature will contain entries representing the green color which is dominant in the image, however there will be enough values describing the animal's color characteristics. This new information enables us to infer that an object is also depicted in the picture apart from a forest. Local features can be employed to learn any of the major image properties (i.e. color, shape, texture), plus many of the global extraction methods can be modified into a local framework. The idea behind this is that the descriptor calculation is performed iteratively for every neighborhood instead of doing it once for the whole image.

Computing a descriptor at every pixel puts great burden on the hardware

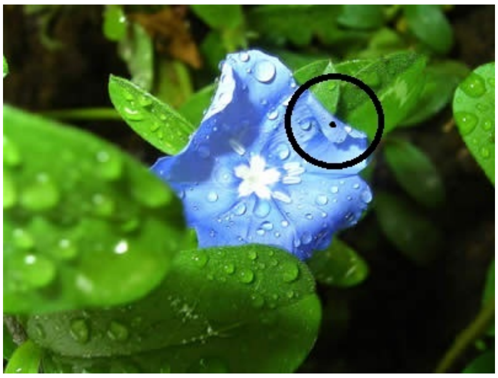


Figure 2.1: A local neighborhood (area inside the black circle) of a pixel (black point).

and propagates complexity to the vector formulation process. To reduce computational costs, an image may be divided into small blocks and features are computed individually for each block. The features are still local because of the small block size, but the amount of computation is only a fraction of that required for obtaining features around every pixel. To achieve a global description of an image, each individual block vector is concatenated to a single high-level visual signature vector or combined under more elaborate schemes [71, 142, 120]. Figure 2.2 illustrates the block based feature extraction and concatenation.

Recently, a popular approach is to extract local features only at salient points. The motivation behind this scheme is that not all image regions are equally important. As in human vision, attention is given to strong edges and contours instead of uniform regions [88, 86, 87]. Therefore, we can compute features only on the interesting points discarding the rest non-informative regions. This method requires an interesting point detector. Currently, the most widely adopted detector is based on the extrema of the Difference of Gaussians (DoG) function proposed by Lowe [82]. Depending on the application other approaches might be more suitable. In the case of sketch images (binary drawings), instead of trying to detect points of interest, all the edge pixels (edgels) can be used as salient points. An average camera photo contains a few thousands interesting points so the concatenation of all these local features to a global vector produces a very high-dimensional visual signature. It is known that high-dimensional representations suffer from the so called *curse*

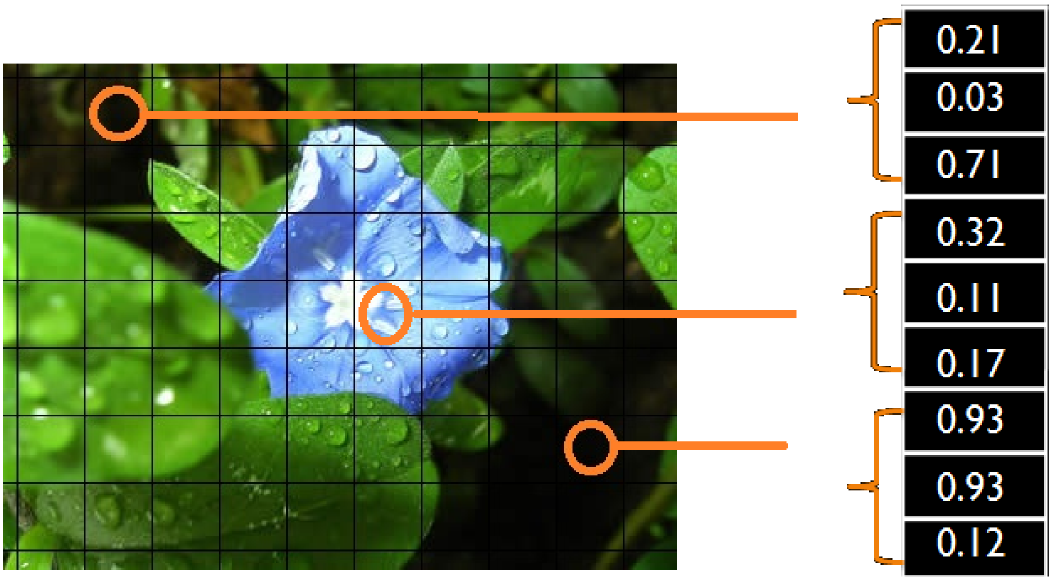


Figure 2.2: Block based local feature extraction.

of dimensionality [38]. The complexity can be reduced by the quantization of the ensemble of features vectors with the help of a predefined codebook which is usually obtained via clustering. More details are presented in the following section.

2.1.2 Bag-of-Words: An Approach Inspired by Text Retrieval

The most widely adopted approach for large-scale CBIR borrows successful concepts from text retrieval. An overview of the bag-of-words (BoW) scheme is given in Figure 2.3. First proposed by Sivic and Zisserman [114], the BoW method treats every image in a database as a collection of visual words. While in text retrieval a dictionary of words is already defined, in CBIR the visual words concept is harder to standardize. A dictionary of visual words should be created from the available images. A clustering algorithm is required to group the feature vectors of all the images in a database, so as to create a codebook of visual words. The generation of a reliable codebook is an open-ended problem due to the huge variation of visual information and the high-dimensional nature of the feature vectors. In [114] a simple k-means

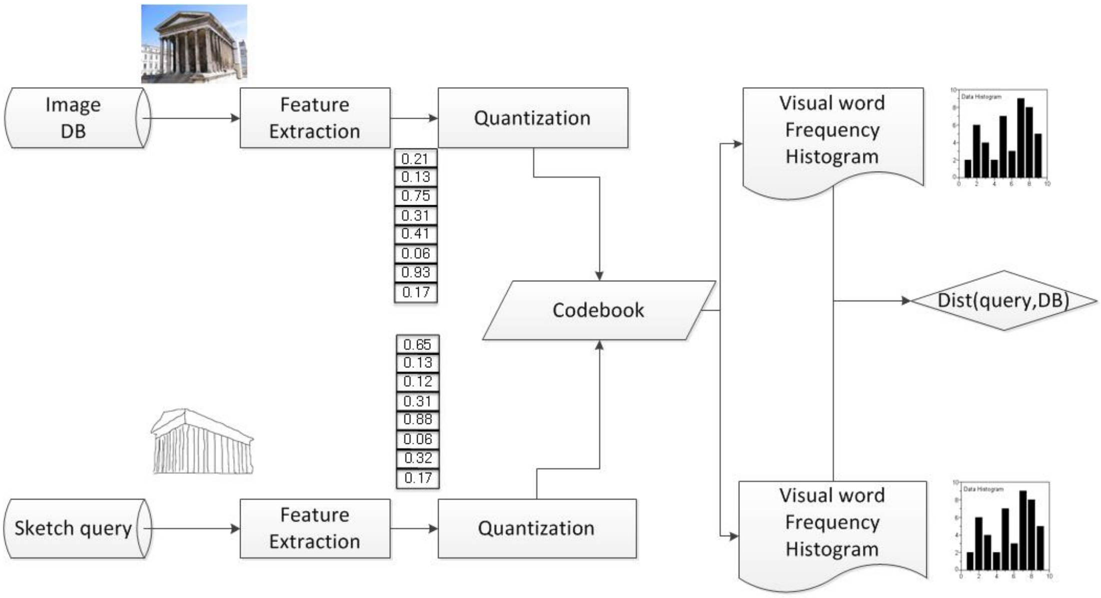


Figure 2.3: Overview of the bag-of-words scheme.

clustering algorithm was employed to form the codebook, which is efficient but difficult to scale to large image collections. On top of that, a larger database requires a richer dictionary to represent all the possible variations. To address this problem, Philbin *et al.* [100] compared different scalable methods for building a vocabulary and introduced a novel quantization method based on randomized trees. Instead of solving a large clustering problem they reduce complexity by solving many smaller problems. K-means is the most common approach, yet mean-shift [65] and hierarchical k-means [95] algorithms have also been studied in literature. Recently, research focused on finding more discriminative codebooks [131, 135].

To ease the discussion, let us suppose that a dictionary of visual words has been successfully generated. Every image is represented by an ensemble of local feature vectors. Subsequently, quantization of the vectors (i.e. assignment to the most similar codebook entry) will create an image representation based on visual words, hence the name bag-of-words (BoW). From the BoW representation a histogram of visual word frequency for each image will be generated. Note that the dimensionality of the word frequency vector equals the number of the words in the visual codebook, therefore this parameter plays an important role in the overall performance and it is often tuned empirically. The formation of the document-word vector is often built according to a weighting scheme, because not every word in the codebook is equally important. The standard weighting technique, frequently used in text retrieval, is the tf-idf scheme [104]. It is a statistical measure used to evaluate how important a word is to a document in a given collection or corpus. It is computed as follows: suppose there is a vocabulary of m words, then each image I_j is represented by a histogram vector $\mathbf{h} = [w_1, w_2, \ldots, w_m]^T$ of weighted word frequencies with component:

$$w_i = \frac{n_{ij}}{n_d} \log \frac{D}{D_i} \tag{2.1}$$

where n_{ij} is the number of occurrences of visual-word w_i in image I_j, n_d is the total number of words in image I_d. D_i is the number of documents containing the term w_i and D is the number of images in the whole database. This weighting is product of two terms: the *visual-word frequency* $\frac{n_{ij}}{n_d}$ and the *inverse document frequency* $\log \frac{D}{D_i}$. Tf-idf tends to filter out common terms. The weight value will always be greater than or equal to zero. Intuitively, this calculation determines how relevant a given word is in a particular document. Word frequency assigns higher weights to words occurring more often in a particular document and thus describes it well, while inverse document frequency lower the importance of words that appear often in the collection or corpus and therefore do not help in discriminating between different documents. Additionally, the resulting vector is sparse and this property can be exploited to speed up the retrieval process.

To measure the similarity between two image tf-idf weighted representations $h_j = [w_1^{(j)}, w_2^{(j)}, \ldots, w_m^{(j)}]^T$, $j = \{1, 2\}$, the cosine angle distance between their term frequency vectors w_i is ordinary used:

$$\cos(\mathbf{h}_1, \mathbf{h}_2) = \frac{\mathbf{h}_1 \cdot \mathbf{h}_2}{\|\mathbf{h}_1\| \, \|\mathbf{h}_2\|} \tag{2.2}$$

The cosine angle similarity can be seen as a method to normalize document length during evaluation. The resulting similarity ranges from 0 meaning exactly opposite, to 1 meaning exactly the same. In-between values indicate intermediate similarity or dissimilarity. Relevant results are ranked according to the cosine similarity between the query and all the images in the database.

Obviously, when the database size grows, the query time and memory requirements are increasing as well, so instead of linearly scanning through all the images we can perform speed-up optimizations like using an inverted index of visual words or organize hierarchically the term frequency vectors.

2.1.3 Similarity Measurement

The selection of the appropriate metric to compare visual signatures consists the second fundamental problem of CBIR. The type of signature created by the feature extraction algorithm determines the choice of similarity evaluation to be followed. In the previous chapter three types of visual signatures were explained:

- a global feature vector
- a set of block-based feature vectors
- a summary of feature vectors generated by quantized local features

For each type we will describe the popular distances in literature.

When we are dealing with vector data there is a vast range of similarity measures. The Minkowski distance family, which includes the Euclidean and the Manhattan distances, constitutes a well-defined choice. It can be computed fast and works well in most cases, but it suffers from the curse of dimensionality [38]. One approach to tackle this issue is to search for a non-linear manifold in which the feature vectors lie, and to replace the Euclidean distance with the geodesic distance [121]. The assumption here is that human visual perception is described better within a non-linear subspace than in the original linear space. Computation of similarity may then be more appropriate if performed non-linearly along the manifold. The geodesic distance between two points A and B is approximated as the shortest path from vertex V_A to vertex V_B in a graph G and offers invariance to non-rigid transformations such as bend or shear.

The technical emphasis on block-based signature similarity rests on the definition of a distance between sets of vectors, which differs from the traditional single-vector functions. The Hausdorff distance could be employed for this purpose and has been applied to CBIR by Ko and Byun [67]. First, consider

Table 2.1: Popular distance metrics for image similarity.

Distance measure	**Input**	**Computation**
Minkowski distance	$\mathbf{x}, \mathbf{y} \in R^n$ (vectors)	$(\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{y}_i)^p)^{\frac{1}{p}}$
Hausdorff distance	$\{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ $\{\mathbf{y}_1, \ldots, \mathbf{y}_1\}$ (vector sets)	$\max(\max_i \min_j d(\mathbf{x}_i, \mathbf{y}_j), \max_j \min_i d(\mathbf{x}_j, \mathbf{y}_i))$
KL divergence	$A, B \in R^n$ (histograms)	$\sum_i A(i) \frac{A(i)}{B(i)}$
Histogram intersection	$A, B \in R^n$ (histograms)	$\sum_{i=1}^{n} \min(A(i), B(i))$
χ^2 distance	$A, B \in R^n$ (histograms)	$\sum_{i=1}^{n} \frac{(A(i)-B(i))^2}{A(i)+B(i)}$

an image signature in the form of a set of feature vectors $\{x_1, x_2, \ldots, x_n\}$. Let us denote two signatures by $I_m = \{\mathbf{x}_1^{(m)}, \mathbf{x}_2^{(m)}, \ldots, \mathbf{x}_n^{(m)}\}, m = 1, 2$. In Hausdorff distance every $\mathbf{x}_i^{(1)}$ is matched to its closest vector in I_2, say $\mathbf{x}_j^{(2)}$ based on a $d(\cdot)$ metric, usually the Euclidean distance. The dissimilarity between I_1 and I_2 is then defined as the maximum among all the matched pairs. Hausdorff distance is made symmetric by additionally computing the distance with role of I_1 and I_2 reversed and choosing the larger of the two distances. The formal definition of the Hausdorff distance between two finite sets is:

$$D_H(I_1, I_2) = \max(\max_i \min_j d(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(2)}), \max_j \min_i d(\mathbf{x}_j^{(1)}, \mathbf{x}_i^{(2)})) \tag{2.3}$$

Similarity between quantized feature vectors ordinary follows text retrieval evaluation measures, for instance the cosine distance. Due to the signature vector being often a histogram representation of the feature codebook, the Histogram Intersection distance (HI) or χ^2 distance could also offer improved performance. Moreover, representations based on probabilistic models can be measured by the Kullback-Leibler (KL) divergence [70]. The KL divergence, also known as the relative entropy, is an asymmetric information-theoretic measure of the difference between two distributions f $(\cdot)$ and g$(\cdot)$, its mathematical formulation is available in Table 2.1.

The different distance measures discussed so far have their own advantages and disadvantages. While simple methods lead to very efficient computation, which in turn makes image ranking scalable, a quality that greatly benefits real-time applications, they have limited efficiency on noisy data. Depending on the specific application and the nature of image signature, a very important step in the design of an image retrieval system is the choice of a distance metric.

2.1.4 Evaluation Techniques

Performance evaluation of content based image retrieval systems is a cumbersome task, mostly due to the subjective interpretation of the quality of the retrieved images. A typical CBIR evaluation system consists of three components:

- A benchmark dataset that ideally contains images covering a large range of semantics and is large enough for the evaluation to be statistically important.
- A ground truth for the provided dataset. Ground truth is subjective because it is established by humans.
- A metric for evaluation. The metric should try to model user requirements.

The design of an evaluation system is a cumbersome procedure. Images should frequently be handpicked to ensure the coverage of a broad range of semantics topics. Moreover, ground truth should be gathered from a considerable large number of observers to reduce bias. Deciding on metric and evaluation criteria is another difficult problem. CBIR technology is expected to satisfy the needs of people who use it, hence a fair objective evaluation should comply with user expectations.

Evaluation metrics have been adopted from information retrieval theory, as intrinsically CBIR constitutes an information retrieval problem. The most popular metrics used are summarized below:

- *Precision*: The fraction of the images retrieved that are relevant to the query.

$$P = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} \tag{2.4}$$

- *Recall*: the fraction of the images that are relevant to the query that are successfully retrieved.

$$R = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} \tag{2.5}$$

Precision and recall are inversely related. Precision falls as the number of retrieved images is augmented, while recall increases. Typically, results are depicted as precision-recall curves. The *F-measure* is the harmonic mean of precision and recall can also be used to obtain a unified indication of the performance. The balanced *F-measure* is defined as:

$$F_m = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2.6}$$

The above are single-value metrics based on the whole list of documents returned by the system. For systems that return a ranked sequence of images, as in CBIR, it is desirable to also consider the order in which the returned documents are presented. Average Precision (AP) emphasizes ranking relevant documents higher. It is the average of precision measurements evaluated at the position of each of the relevant documents in the ranked sequence:

$$AP = \frac{\sum\limits_{r=i}^{N} P(r)rel(r)}{D_r} \tag{2.7}$$

where r is the rank of the current document in the returned image set. N is the number of retrieved documents, $rel()$ is a binary function on the relevance of a given rank. $P(r)$ and D_r are respectively the precision and the number of relevant documents at the given cut-off rank . The Mean Average Precision (MAP) for a set of queries is used as evaluation measurement for popular

CBIR benchmarks like TRECVID [97]

$$MAP = \frac{1}{Q}\sum_{q=i}^{Q} AP(q) \tag{2.8}$$

Q is the number of queries in the benchmark. The system that holds the higher MAP on a benchmark database is considered as the state-of-the-art in the specialization area defined by the benchmark (e.g. shape databases, portrait databases etc.).

2.2 Sketch Based Image Retrieval

2.2.1 Appearance Features

The particular nature of SBIR renders some of the most popular feature detectors and descriptors inappropriate for providing a robust visual signature. As previously discussed, shape and structure are the only meaningful appearance based modalities to capture dissimilarities between binary sketches and edge maps extracted from images. In this section, we will briefly overview shape descriptors that can been applied to SBIR.

Appearance based descriptors extract information from local image patches at points of high saliency. Interesting points detectors use a range of filters to locate such points in an image. The applicability of these filters to binary images is limited due to the inherited singularities. In sketches and edge maps, edge pixels designate the set of interesting points. They are a sparse subset of all the image pixels and are available without the need of extra computation. SBIR applications commonly extract image descriptors on edge points or on a densely sampled grid.

Histogram of Oriented Gradient (HOG) descriptors capture the orientation distribution of edges. From an image $\boldsymbol{I}$ the gradient $\nabla \boldsymbol{I} = \boldsymbol{G}$ is computed. From $\boldsymbol{G}$, the gradient magnitude $\boldsymbol{M}$ and orientation $\boldsymbol{O}$ can be derived. Note that in SBIR the orientation is represented in the range $[0, \pi)$ as the direction is not a discriminant property. At pixel position $\mathbf{p} = [x, y]$ the gradient magnitude is defined as m_p and the orientation as o_p. The orientation matrix is subsequently quantized into r channels. The r-th quantized orientation channel can be written as $\boldsymbol{O}^{(r)}$. In a local neighborhood $\mathcal{N} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k\}$ the orientation distribution at bin k is defined as:

$$\mathbf{h}^{\text{HOG}}(k) = \sum_{\mathbf{p}\in\mathcal{N}} \sum_{o_p\in\boldsymbol{O}^{(k)}} m_p \tag{2.9}$$

The description vector $\mathbf{h} = [h_1, h_2, \ldots, h_2]^T$ is formed from the r histogram bins. Usually the local neighborhood $\mathcal{N}$ is further divided into $n \times n$ blocks and a histogram is acquired for each block. The final descriptor is a concatenation of the individual block histograms. A subsequent normalization step assures that the number of edges in each image will not bias the description.

The structure tensor [41] has been especially designed for sketch description. It follows the foundations of HOG, yet instead of measuring the distribution of edge orientation in each block the structure tensor of the region is returned. Under this scheme the main edge orientations are captured. The structure tensor for a neighborhood $\mathcal{N}$ is:

$$T^{(\mathcal{N})} = \sum_{\mathbf{p}\in\mathcal{N}} \mathbf{g}_p \mathbf{g}_p^T \tag{2.10}$$

where g_p is the gradient vector at pixel p. Each tensor is normalized with its Frobenius norm.

Shape context [9] express the configuration of a shape relative to a reference point. For a point $\mathbf{p}_i$ a histogram of distances against the remaining points is calculated. More precisely:

$$\mathbf{h}_i^{\text{shape context}}(k) = \#\{\mathbf{q} \neq \mathbf{p}_i : (\mathbf{q} - \mathbf{p}_i) \in \text{bin}(k)\} \tag{2.11}$$

This descriptor offers translation, rotation and scale invariance and has been widely used in shape matching.

The motivation behind *self-similarity descriptor* (SSD) [110] is related to that of shape context. A local internal image similarity is computed at each pixel. A compact descriptor is formed with correlation scores between a small image patch centered at pixel p and a larger image region surrounding the patch. The ensemble of descriptors can be coded in a bag-of-features scheme for retrieval.

Other descriptors focus on analyzing statistical properties of the edgel configuration and avoid the use of orientation distributions. Popular approaches are Fourier analysis [140] and Zernike moments [73]. These transformations are applied directly to the pixels and provide invariant descriptions. The amount of detail of the description can be regulated by modifying function coefficients, but more complex representations suffer from numerical instability.

Contour based descriptions are also encountered in the literature. Ferrari *et al.* [44] presented a local scale invariant shape descriptor formed from groups of adjacent contour segments. In this work, image contour segments abiding to Gestalt principles are merged into groups. Each group is described by a set of distinctive geometrical properties. The group ensembles can be used for image matching or can be further coded into a bag-of-feature vector for retrieval

Local Binary Patterns (LBP) [96] implement a pixel comparison mechanism in local neighborhoods and have been successfully applied to texture recognition [53]. Recently, it has further been determined that when LBP is combined with the HOG it improves human figure detection performance considerably on some datasets [128]. The BRIEF descriptor [16] operates on the same pixel comparison principle. Pre-smoothing local patches reduces noise sensitivity. The produced descriptors are compact binary patch representations and can be used in conjunction with the Hamming distance for fast retrieval. Jegou *et al.* [62] proposed the VLAD compact descriptor for very large datasets. VLAD features are formed by the residuals between a local patch SIFT descriptor and a pre-computed codebook. The residuals undergo a normalization process [8] that produces a compact, discriminative vector representation of an image.

The list of appearance descriptors can grow longer, but the above summarize the most popular approaches in literature. Quantitative evaluation of the performance of these descriptors in SBIR has been attempted in several studies [42, 56, 55, 41]. Across all studies HOG-like descriptors consistently achieve superior performance, in many cases by large margins. Evidently, the histogram representation of edge orientations is tolerant to noise and can produce robust visual signatures tailored for sketch retrieval.

2.2.2 Gestalt Principles

Gestalt is a psychology term which means ‘unified whole’ [6, 68]. It refers to theories of visual perception developed by German psychologists in the 1920s. These theories attempt to describe how people tend to organize visual elements into groups or unified wholes when certain principles are applied. These principles are:

- *Figure-ground relationship*: We group elements as either figures (distinct elements of focus) or ground (the background or landscape on which the figures rest).
- *Proximity*: We perceive objects that are close to each other as forming a group
- *Similarity*: We group similar to each other elements together
- *Symmetry*: We perceive objects as being symmetrical and forming around a center point.
- *Continuity*: WE group elements together if they are aligned within an object.
- *Closure*: We perceive shapes that are not completely there. Specifically, when parts of a whole picture are missing, our perception fills in the visual gap.

Gestalt theory has been applied in several domains, especially in *user-interface* design [22]. Computer vision research has also been influenced by the perceptual organization principles [98, 132, 139]. Detecting Gestalt principles in images is a computational expensive task, as a result it has not been widely researched from an image description perspective. Recent work from Bileschi and Wolf [11] demonstrated recognition improvements over the state-of-the-art results, against the HOG descriptors, by encoding a selected subset of the Gestalt principles. Hand-drawn sketches display symmetries. In Chapter 5, we investigate how this observation can be exploited to boost sketch recognition.

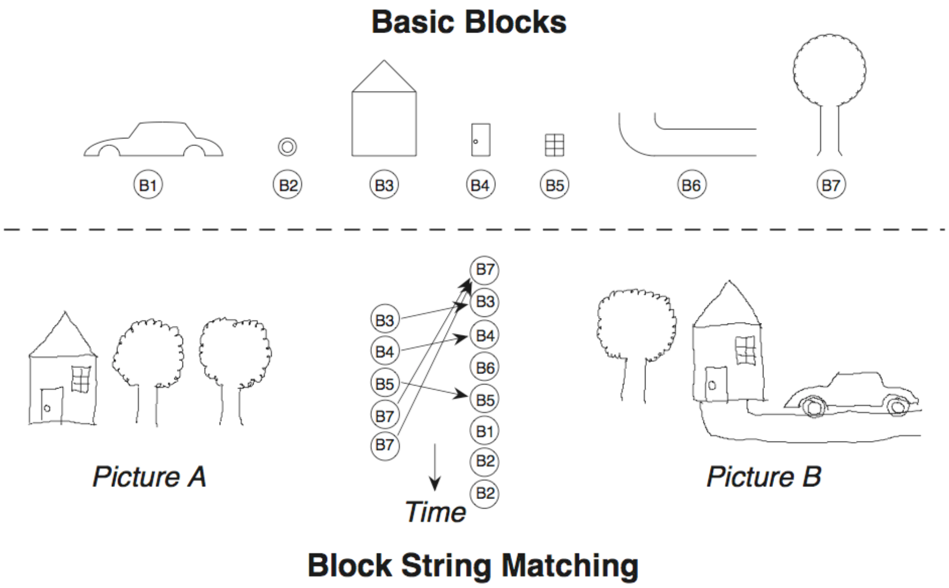


Figure 2.4: Stroke matching applied to hand-drawn pictorial queries. Figure from Lopresti *et al.* [81].

2.2.3 Early Approaches

Probably the proposal of a SBIR system was Chang and King-Sun [23] paper in 1979. They proposed a query language for image retrieval that can process both pictures and sketch inputs. Sketches formulated by indicating lines or regions of interest in an image and authors calculated simple features for each entry such as the length of a line, perimeter, area etc. Users could perform queries to the database to retrieve images based on a property, like two shapes that share the same center.

Hirata *et al.* [54] performed visual search in 205 digitized oil-paintings. Each database image was first rescaled to a predefined regular size and then edges were extracted for each painting. The query consisted of a rough sketch illustration of a database painting. Images were normalized in size and subdivided into 8×8 local blocks. For each local block, the best local correlation was computed by searching in a small window of local blocks. The global similarity was then defined as the sum of the local correlation values. This method benefits from translation and local deformation invariance, but is slow due to the matching process where each block should be compared with all the corresponding blocks in the database.

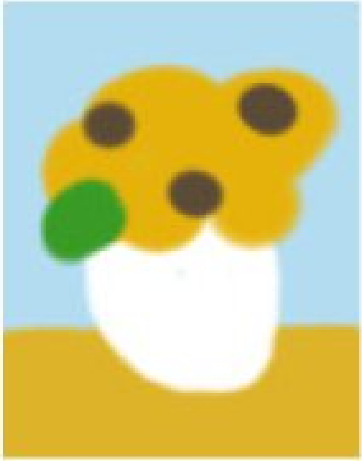


Figure 2.5: A colored sketch query as used in [60].

Lopresti and Tomkins [81] addressed the problem of recognition both for handwritten text and drawings. They took advantage of the inherently sequential nature of drawing in the temporal domain. They segmented a digital pen's input into strokes and a standard set of features was then extracted for each stroke, e.g. stroke length, total angle traversed. The generated vectors were clustered in a small dictionary of basic stroke types. Similarity between images was computed by matching segments that had been drawn approximately in the same order; see Figure 2.4 for an overview. This approach works well in handwritten text recognition where the stroke order is well defined, i.e. when users start writing the first letter and then proceed sequentially. It struggles with pictorial queries due to the arbitrary order of the strokes. For instance, consider the example displayed in Figure 2.4. In Picture B we cannot define whether the tree, the car or the house has been drawn first.

An early attempt for scalable SBIR appeared in the Jacobs *et al.* [60] paper in 1995. A database of 20,000 images have been employed and considering the computational power and digital media availability in 90's, it could be accredited as a large database for that time. The authors introduced a colored sketch as the input query of Figure 2.5, which can be painted by the user. In every image a Discrete Wavelet Transform (DWT) is applied which is fast to compute. DWT requires little storage space because just a small set of high magnitude coefficients contain most of the visual information, hence the majority of weak coefficients can be discarded. Haar wavelets had been chosen as the core filtering unit and offer simple implementation and efficient computation. In addition, user-painted queries tend to have large constant-colored regions, which are well represented by this basis. On top of the truncation of the coefficients, two-level quantization is employed. Positive coefficients are mapped to +1 and negative to -1. Finally, a fast metric was introduced to compare the coefficients between two images and return a similarity value. The method provides adequate retrieval results, but suffers from recall problems. The number of false positives raises as the volume of database images increases. Moreover, the use of colored sketches adds complexity to the query generation process making the system less user friendly.

Jain *et al.* [61] presented a fusion approach with color and shape characteristics combined together. Color was represented by three 16-bin histograms, one for each of the RGB channels, and shape modeled by a histogram of 36 main edge orientations. Similarity was measured as a linear combination of color and shape histogram distances. The authors managed to retrieve 99% of times the given query in the first two positions from a 400 trademark image database.

Del Bimbo *et al.* [37] presented a technique which is based on elastic matching of sketch templates over the shapes in images. The amount of strain and bend energy spent deforming the template was employed to derive a measure of similarity between the sketches and the images. This scheme provides translation invariance. The authors also provide solutions to handle scale variances and spatial relationships between objects in multi-object queries. The method was applied to a pool of 100 images and in spite of its success to that particular database it cannot be employed on larger scale with conventional hardware, due to the cost of computing the elastic deformation between the query and every picture. The matching is also sensitive to rotation transformations.

During the 90's the Curvature Scale Space (CSS) [91] theory was quite popular and had been successfully applied to many shape recognition tasks. Consequently, SBIR approaches based on CSS foundations were also studied [90]. The main idea of CSS is to represent curves at various scales, so that each

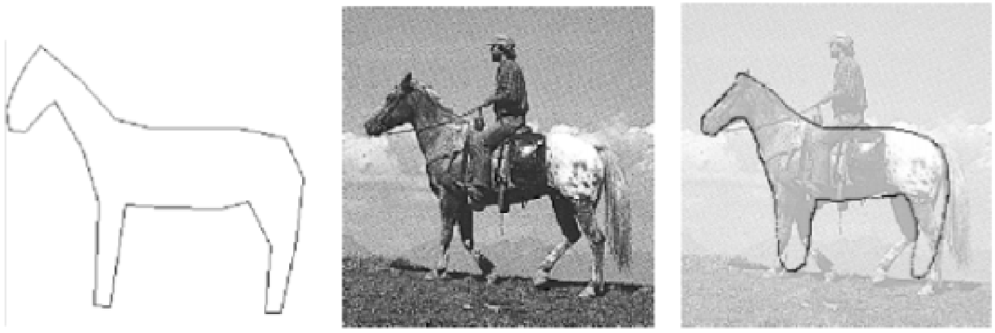


Figure 2.6: Elastic deformation of a sketch to match a shape. Figure from [37].

structure can be represented at its appropriate scale. A curve is parametrized by its arc length and successive convolutions with a Gaussian kernel approximate the scale space. In [90], Matusiak *et al.* evaluated a distance metric on the maxima of the CSS function of two images and use this metric to rank 800 database images. CSS is sensitive to shallow an deep concavities of a shape and therefore not well suited for hand-drawn sketch description.

2.2.4 Recent Approaches

As hardware technology progressed, computers have benefited from a great rise in computational power and memory capacity. Nowadays, personal computers possess processing power several magnitudes higher than machines in the 90's. Therefore, SBIR has entered a new era in which raw computational power is much more affordable and available. Along with the demand for robust sketch/image matching a new requirement emerged; fast query responses on large image databases.

In 2005 Chalechale *et al.* [21] performed angular partition in the spatial domain of images, as a means to extract compact and effective features. The first step of the process is to obtain the edge map of all the natural images, in order to transform them in a format more suitable for matching against binary sketches. Sketch queries were preprocessed by a morphological thinning filter to better match the edge maps extracted from the images. An angular partition of an image is employed and divides images in K angular regions.

K can be adjusted to achieve hierarchical coarse to fine representations. The number of edge points for each region R_i, $i = \{1, 2, \ldots, K\}$ is chosen to represent each slice feature. An 1-D Discrete Fourier Transform (DFT) is then computed for each region of the image and by keeping only the magnitude of the DFT, scale and rotation invariance is guaranteed. Authors also note that this scheme provides robustness against translation as well. Similarity between images and sketches is measured by the l_1 distance between the two feature vectors. This system was tested on a database of 3,600 images. At least 13% percent of the images had to be recovered from the database in order to retrieve the correct image, a fact that highlights the noise sensitivity of DFT.

The work of Liang *et al.* [75] was published the same year, but follows a different approach. Each sketch query is decomposed into basic geometric primitives by using the pen's speed and curvature, properties frequently used in handwriting recognition [101]. These strokes are later organized into a topological graph according to their spatial relations. In total, eight spatial relations were selected to accommodate scale, translation and rotation invariance. A graph is formed as follows: every vertex is represented by a stroke and an edge can be established between two nodes if and only if there is a spatial relation among the strokes. To enable topological graph comparison, a vector representation is required, hence the graph spectrum is calculated. The graph spectrum is the set of eigenvalues of the adjacency matrix of the graph. An issue here is that for different sketches with different number of primitive shapes, the dimensionality of the spectrum descriptors will vary. Dimensionality reduction is performed to remedy this imbalance. The Euclidean distance between two feature vectors was adopted as a similarity measure in this work. Authors propose a relevance feedback module that in conjunction with the retrieval system will improve query results. The main drawback of this algorithm is that the authors included only a database of sketches without including natural images to evaluate the system. Hence it is not known whether the algorithm can be applied to natural images. Segmentation of free-hand sketches have been studied with some success [119], yet geometric primitive extraction from natural images is a much more complex problem.

Hu *et al.* [55, 56] proposed a descriptor specialized for SBIR, the Gradient Field HoG (GF-HOG), which encapsulates local spatial structures in the sketch and facilitates retrieval based on a dictionary of visual words. Instead of calculating a shape descriptor in the edge map domain, image structure is represented using a dense gradient field interpolated on the sparse set of edge pixels. The interpolation is performed by solving Poisson's equation with Dirichlet boundary conditions. The well-known HOG [34] descriptor is then employed for every edgel in the interpolated space, at several scales, to export the final descriptor set for an image. Finally, the BoW coding scheme is adopted to enable queries in a database. In [56], an extensive evaluation on several image descriptors is carried out indicating the superiority of HOG-like features in SBIR. The authors, also made publicly available two SBIR databases, namely Flickr160 and Flickr15k. Section 2.2.6 provides more details on them. Our work in SBIR, presented in Chapter 3, outperforms GF-HOG and to our best knowledge tops the state-of-the-art on both datasets.

Eitz *et al.* [40] were the first to propose a SBIR method that can scale to large datasets. Indeed, they demonstrated their approach with an image collection of 1.5 million images from Flickr. They employed a block-based tensor descriptor, named Structural HOG (SHOG). Each image was first transformed to an edge map and edgels with magnitude less than a threshold were removed in order to eliminate responses from image areas with uniform intensity or color transitions. A fixed grid was then applied to the image and for each cell a structure-tensor descriptor computed based on the gradients of the edgels in the current block. This descriptor is similar to the edge histogram descriptor (EHD) [113], but instead of quantizing orientations into bins, it gives information about the main orientation of gradients in a block. In order to detect similarly oriented image edges, independent of the magnitude of the edges, every structure-tensor is normalized with its Frobenius norm. The Frobenius norm is suitable to normalize matrices and that is the reason it was selected. The process is visualized in Figure 2.7. Each image cell was represented by an ellipse with axes being formed by the eigenvectors of the structure tensor. The dissimilarity between two descriptors was defined by summing the structure-tensor distances of the cells. Cells not represented in the query were excluded and did not contribute to the metric, in order to enforce image/sketch spatial matching. A linear search in the database took between 0.4 and 3.5 seconds.

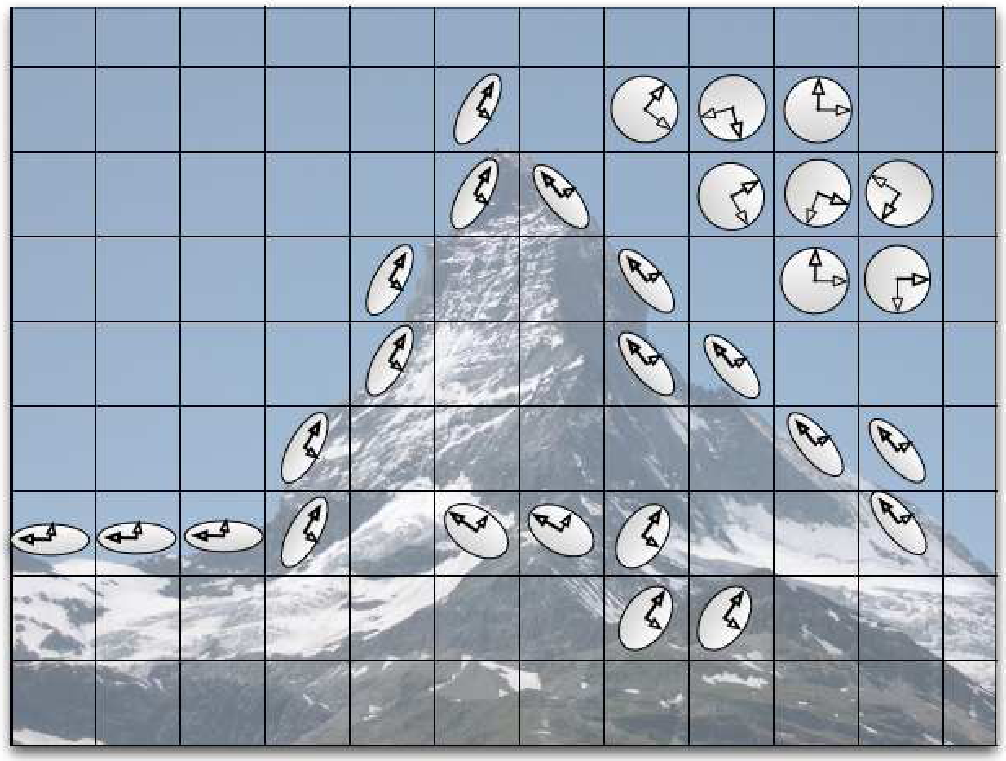


Figure 2.7: Illustration of the tensor descriptor for each image cell. Figure from [42].

A recent study [56] showed that the SHOG block-based approach does not provide robust results if compared against bag-of-features approaches. In a later review, the authors suggested a bag-of-features variant of [42], that overcomes some of its weaknesses. Our research presented in Chapter 3 demonstrates superior results over both SHOG methods.

Min Yoon [137] suggested a similar descriptor to SHOG. In this approach, rather than calculating a structure-tensor for every image cell, a tensor is computed for the local neighborhood of each edgel and the eigenvalues of each tensor are stored. The similarity metric is evaluated on the ensembles of two images. It is not known how efficiently this approach scales to large databases, since there is no mention of it in the paper. The method was applied to a pool of 600 images, a size inadequate to draw conclusions on scalability.

A structure similarity approach has been studied in [103]. Structure information is extracted from sketches and image edges in the form of key-shapes.

Key-shapes are collections of stroke primitives, like arcs, lines or ellipses. Each key-shape can be described by the distribution of the stroke primitives it encompasses in its neighborhood as well as by the orientation distribution in the same area. An ensemble matching approach based on the Hungarian Method [69] is employed to generate image rankings. This approach, only if combined with a bag-of-features descriptor, presents some promising results but the cubic complexity of the matching renders it inappropriate for large size databases. To give a feeling for scalability, authors state that each image/sketch pair is matched in 8ms. For a database of 100,000 images a sketch query will require a little more than 13 minutes to be completed.

Recently, the Oriented Chamfer Matching (OCM) distance has been applied to sketch/image matching [18, 118]. In its original form, OCM requires high computational and storage costs. In [18], an approximation of the OCM was introduced based on the Distance Transform (DT). Each edge image, post edge extraction, is divided in k orientation parts. For each part an inverted index structure is generated from its DT and stored into the memory. At the query stage, a look-up on the inverted index is performed for each sketch edgel. Under this scheme 2.1 million images are indexed in 6.5GB of memory. In [118], further optimizations were suggested to enable search in a 2-billion-image database. Here, the OCM is approximated by the dot product between the sketches edge map and the images DT. To achieve that, the original DT and edge map matrices are vectorized and subsequently projected in a lower dimensional space via PCA. A hashing framework further reduces the computational and storage requirements. These methods yield a highly scalable scheme, yet the quality of the retrieval results is questionable as evaluation on popular SBIR databases is not provided.

2.2.5 Hybrid Systems

This section covers methods that do not strictly use only appearance based cues to retrieve images. These systems retrieve a big volume of images using a text query, in an attempt to bridge the semantic gap, and then try to prune irrelevant results based on visual constraints. Optionally, some implementations provide the user the possibility to seamlessly collage several retrieved objects together to construct an image that better matches their needs.

Sketch2Photo [25] composes a realistic picture from a simple free-hand sketch annotated with text labels. A web search is performed based on the text labels attached to each of the drawings and an initial volume of images is returned. Since text image search generates lots of inappropriate results, the authors suggest a filtering scheme which gives a small set of images that match the depicted sketch. The goal of this system is to combine several retrieved images to create a new picture that meets the user's needs. This is achieved by providing the user with a set of background and scene images to select from. Subsequently, the user chooses a background image and objects from the scene images and stitch them together. Therefore, the filtering process consists of two separate functions: a) background image selection; b) object image selection. The background image is selected to be consistent with the query label and to be uncluttered for the composition to be performed. The consistency criteria are met by assuming that the majority of the returned images will be consistent to the text label, so a clustering in the LUV color domain is applied and few images closer to the largest cluster centroid are picked. Images with uniform regions are chosen by counting the number of segments they include, because pictures with few segments are more likely to be uniform. The process continues with the scene item image classification. Regions containing objects of interest are retrieved with a saliency measurement algorithm, so images with low saliency scores are pruned. The remaining candidates, after extracting their contours, are matched based on their shape with the corresponding sketched silhouettes. The images that have not been eliminated during this process are presented to the user. Obviously, Sketch2Photo cannot be applied to large-scale retrieval due to the computational demands of its filtering system, and was not implemented for this purpose. Still, it produced interesting results and provided an interactive query component.

Liu *et al.* [78] conceived an interface to incorporate episodic memory for specific image retrieval. Specific images refer to images one has certain episodic memory about, e.g. a picture one has seen before. Episodic memory is the memory of autobiographical events that can be explicitly stated. They propose an interactive query-generating process that allows users to specify the semantic category and rough area/color of the objects. Let us assume the user imagines a sea landscape with a boat in the foreground, as in Figure 2.8.

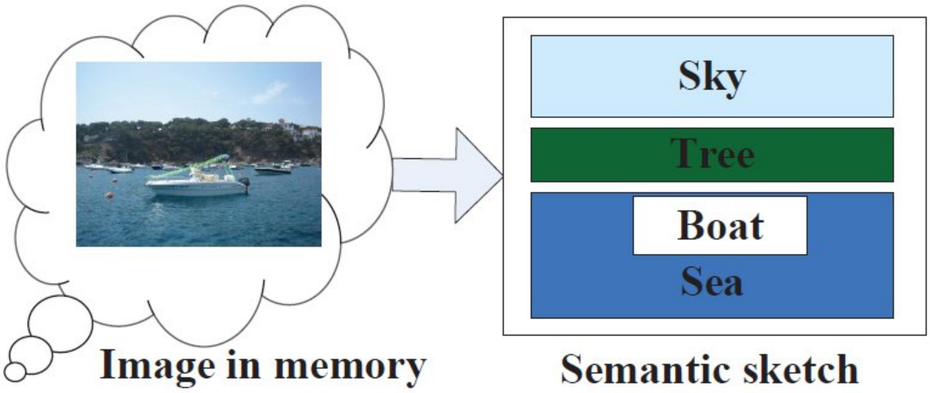


Figure 2.8: Semantic sketch query formulation as defined in [78].

The user draws the main objects in the canvas in the appropriate location and color and gives each one a text label. This method requires a pre-annotated dataset that additionally includes object in bounding boxes. Obviously, this poses limitations on the diversity of the queries that can be represented and includes the additional effort to manually annotate and detect objects in the pictures. When a new query occurs the system samples the reference database and retrieves object exemplars for each entity in the query, e.g. sky, tree. The authors propose a robust sampling method to deal with label noise. The method selects multiple object exemplars from the reference dataset to represent one missing sketched target object. Images are then matched against possible object exemplar combinations with a parallel local matching algorithm. For every exemplar a color SIFT descriptor is calculated. Following the bag-of-features approach, a histogram of words is generated based on the local descriptor for every image. The histogram similarity between the image and each exemplar is computed and the maximum matching score over all exemplar combinations is used to rank an image. Meanwhile, a spatial re-ranking algorithm is derived to offset inaccuracies from user sketches. The spatial re-ranking algorithm selects the best response in a set of local bounding boxes by a branch-and-bound procedure on multiple object exemplars simultaneously. Fianlly, ranked object retrieval results are combined to produce the final ranking. This work highlights that episodic memory is very helpful in retrieving the target image, given an interactive SBIR system that will help users express their thoughts.

MindFinder [19] encourages users to attach text labels to a hand drawn sketch and define the dominant color cues of the desired images. First, a pool of similar images based on the query is collected from the database using the work in [18]. Consequently, these images are filtered to match the user provided labels and color traits.

Hybrid SBIR systems incorporate semantics by requiring the user to label their drawn sketch. This contradicts the purpose of SBIR as defined in Chapter 1, which is to provide an alternative retrieval platform for search cases that cannot be expressed easily with words. It also prevents the query generation process to be simple and quick. Therefore, we suggest that the appropriate way to bridge the semantic gap in SBIR, is to divert the sketch recognition problem to the machine without burdening the user.

2.2.6 Datasets

Evaluating a sketch based image retrieval system is not a trivial task. On top of the challenges inherited from CBIR, like search for appropriate metrics and diverse semantics interpretation between users, the abstraction of the sketch query is added. It is even more difficult to match images and sketches due to the vague nature of the latter. A sketch can depict shapes or symbols or an imaginary scene, thus semantic convergence with photographic images is not always the case. A successful benchmark targeted to large scale SBIR should fulfill the following requirements.

- *Large and diverse image database.* Many images are required in order to measure the response time of a SBIR system; plus, a large database provides statistical significance and ensures that many semantic concepts will be represented.

- *Objective semantic links between pairs of images and sketches.* Images indicated as similar to a query must be objectively perceived as similar. On the other hand, images indicated as dissimilar should be objectively perceived as dissimilar.

- *Unambiguous sketch query concepts.* Benchmark queries concepts should not be too vague and some rules ought to be specified to users, like recommending to draw objects that will exist in the database.

- *Appropriate metric.* Depending on the nature of the benchmark generation process, a metric that evaluates a SBIR according to how similar it is to human performance should be conceived.

So far, three publicly available benchmark datasets have been published in the literature: EitzSBIR dataset [42], Flickr160 [55] and Flickr15k [56]. The work presented later in this thesis achieves state-of-the-art results in all three datasets.

2.2.6.1 EitzSBIR

This benchmark was published by Eitz *et al.* [42] and is based on a controlled user study of 28 subjects. It consists of 31 hand-drawn sketches, 1,240 images related to these sketches and 100,000 distractor images. It is also available online [1]. The authors establish sketch/image ratings based on user ratings in a controlled environment. Generation of input sketches was designed with focus on shape based retrieval. Users were prompted to avoid too much abstraction and symbols in their drawings and encouraged to generate sketches depicting objects or scenes in a way that they would expect to perform well for an image retrieval system. This process generated 164 sketches. Authors selected the 31 more precise and coherent sketches that reasonably matched a sufficient number of images in the database. Some of the input sketches are depicted in Figure 2.9. Each sketch was associated with 40 visually similar images according to the user rating, for a total of 1,240 images. 100,000 distractor images were also provided as noise and mixed with the 1,240 images.

The author's aim was to create a benchmark to quantitatively compare a machine's result with respect to the human performance. That is, how correlated is the ranking produced by a human, in this case the mean score of all the participants, to that of a computer. Kendall's tau is a measure of rank correlation, allowing assessment of the degree of correspondence between two rankings and defining the significance of this correspondence. Kendall's

[1] http://cybertron.cg.tu-berlin.de/eitz/tvcg_benchmark/index.html

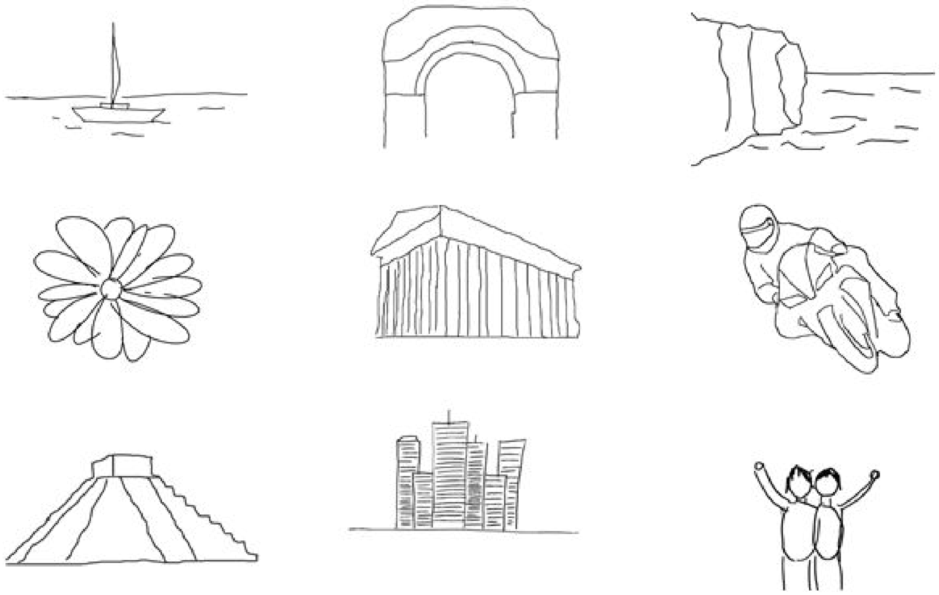


Figure 2.9: A subset of the sketch queries of EitzSBIR dataset [42].

rank correlation coefficient is computed as the difference between the number of concordant and discordant pairs in two rankings. Normalization by the total number of pairs is applied to gain independence of the test size. Mathematically, Kendall's tau is defined as: The author's aim was to create a benchmark to quantitatively compare a machine's result with respect to the human performance. That is, how correlated is the ranking produced by a human, in this case the mean score of all the participants, to that of a computer. Kendall's tau is a measure of rank correlation, allowing assessment of the degree of correspondence between two rankings and defining the significance of this correspondence. Kendall's rank correlation coefficient is computed as the difference between the number of concordant and discordant pairs in two rankings. Normalization by the total number of pairs is applied to gain independence of the test size. Mathematically, Kendall's tau is defined as:

$$\tau = \frac{n_c - n_d}{n(n-1)/2} \tag{2.12}$$

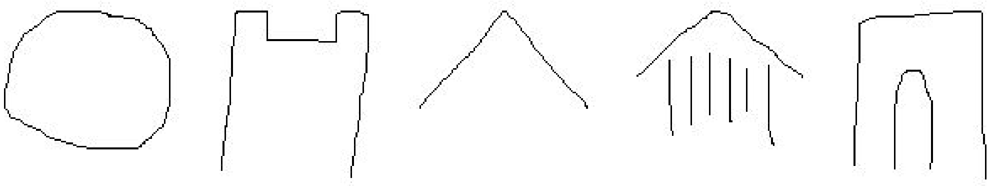


Figure 2.10: Five query categories of Flickr160 [55].

where n_c denotes the number of concordant pairs and n_d the number of discordant pairs and n is the size of the sample set. The correlation coefficient τ can take values in the range $[-1, 1]$ with -1 indicating a reversed ranking, 0 indicating that the two rankings are independent and 1 indicating that two rankings are the same.

During evaluation, researchers should compare the rankings generated from the 31 sketches to the ground truth, which is set as the average rating over the 28 participants. Each of those averaged rankings can be seen as a consensus between the participants. The mean of the 31 correlation values constitutes the generalized performance estimation of the evaluated method.

2.2.6.2 Flickr160

Flickr160[2] [55] is a small SBIR dataset. It consists of the 5 query categories depicted in Figure 2.10. For each category 5 hand-drawn sketches are provided, totaling to 25 sketch queries. Additionally, 32 images similar to each category are obtained from Flickr, forming a dataset of 160 images. The evaluation metric suggested in this benchmark is the MAP, defined in Section 2.1.4.

Comparing Flickr160 with EitzSBIR, one can observe that the latter is more diverse, contains better-quality sketches that cover a greater range of concepts. Moreover, Flickr160 offers a limited photo collection making scalability measurements unreliable. Flickr160 is useful to acquire an initial picture of the performance of a SBIR system or perform parameter tuning before evaluating it on a more challenging dataset.

[2] http://personal.ee.surrey.ac.uk/Personal/R.Hu/ICIP.html

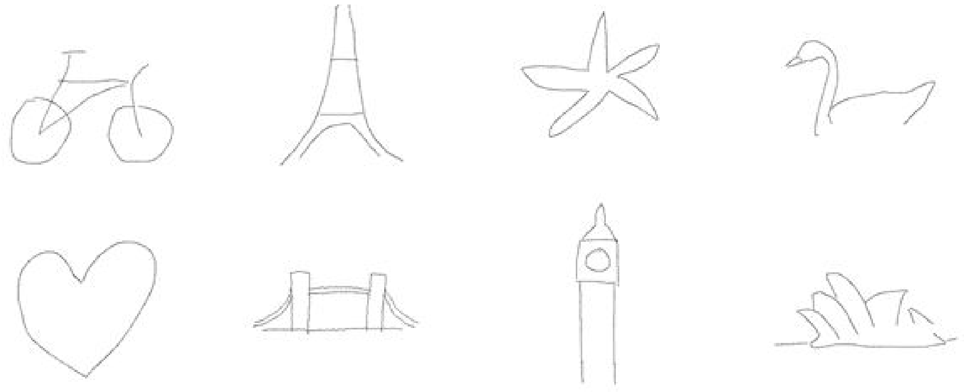


Figure 2.11: Some sketch queries from Flickr15k dataset [56].

2.2.6.3 Flickr15k

Flickr15k[3] [56] can be seen as an expansion of the Flickr160 dataset. In this study, 10 subjects generated a sketch query collection for 33 topics, describing shape, landmark buildings, objects and scenes. The total number of sketches is 330, some of them illustrated in Figure 2.11. For each topic a range of photographs was collected from Flickr. The dataset includes 14,660 images including affine transformation and background clutter. Most of the images are associated with a sketch category. A part of the database does not belong to any category and serves as noise. The evaluation metric recommended is the MAP over the set of queries.

Flickr15k is challenging both for query diversity and image variability. It is adequately large to evaluate scalability and response time. It can be employed in conjunction with the EitzSBIR dataset for a reliable estimate of a SBIR system's performance.

2.3 Sketch Recognition

2.3.1 Methods

Machine understanding of hand-drawn sketches is an open issue and has been studied since the early days of computer revolution [94]. The advantages of

[3] http://personal.ee.surrey.ac.uk/Personal/R.Hu/SBIR.html

accurate sketch recognition are obvious: digitization, indexing and fast search of handwritten notes and diagrams, user drawing assistance, semantics and much more. Great attention to diagram [109, 49, 7] and face sketch recognition [129, 66, 141] has been given in the literature. Recently, interest was emerged on recognition of rough depictions of everyday objects [39, 83, 74].

Diagram recognition techniques typically consist of a segmentation step, where primitive curves are extracted from an input sketch, and a recognition step, where the group of segmented curves is categorized to a particular sketch type or component. In [49, 99] a series of rule-based tests are applied to contour segments to classify them to basic diagram symbols like arrows or rectangles. Sezgin and Davis [109] apply a Hidden Markov Model (HMM) to encode the relationships between strokes and classify them to a set of predefined classes. SketchREAD [7] employs a hierarchical Bayesian network on low-level shapes to identify higher-level components of a diagram.

Face sketches are often used in forensics to describe the characteristics of a suspect. These representations are typically made from an expert drawer and depict finer information than binary sketches. In [129] Wang *et al.* suggest a patch matching technique based on a multi-scale HMM that can synthesize photos from sketches and sketches from images. To recognize a face mugshot, all database images are transformed into sketches and the recognition is carried out in the sketch domain. In [141], a projection tree maximizes the mutual information between photo and sketch patches achieving a joint quantized space. Based on the computed tree, features that allow discriminant comparison between photos and sketches are derived. Klare *et al.* [66] adopt a feature learning approach; combined local features from photos and sketches are used to learn a discriminant projection where sketch recognition accuracy is achieved.

The above methods provide solutions for specific domains. A SBIR system requires a more abstract recognition module that will be able to extract semantics from a hand-drawn sketch and recognize familiar drawn objects and concepts. For this purpose, Eitz *et al.* [39] collected 20,000 hand-drawn sketches. In their work, recognition is performed via a bag-of-features approach based on histograms of oriented gradients. For each image, local features are computed over several overlapping image patches. A codebook is built based on the extracted descriptors and each image is represented by a histogram of the visual words frequency. Classification is achieved with the k-Nearest-Neighbor (KNN) and Support Vector Machines (SVM) classifiers. Similarly, Ma *et al.* [83] calculate a HOG variant on a densely sampled grid of the sketch. Feature vectors are coded by a hierarchical vocabulary tree and the χ^2 distance is used for evaluation. Li *et al.* [74] follow an ensemble matching approach. A score is calculated between pairs of sketches based on correlations of patch similarity and location. This scheme offer robust recognition results but the slow matching process prevents scalability.

2.3.2 Datasets

Sketch recognition can be cast as an object recognition problem. The goal is to correctly predict to which class a given sketch belongs to. To ensure accurate prediction, a large labeled database of hand drawn sketches is required, where each sketch category is adequately represented by samples that capture a vast spectrum of deviations. Furthermore, it is essential that the database includes many sketch categories, to accommodate a large variety of queries.

2.3.2.1 EitzSKETCH

Currently, there is one available sketch recognition dataset that meets the above criteria published on 2012 [39]. We name this dataset EitzSKETCH. It consists of 250 categories with each category represented by 80 sketches. The categories are based on a taxonomy of a mixed set of the most frequent labels of LabelME dataset [102], the Princeton Shape Benchmark [111] and the Caltech 256 dataset [48]. The sketches were collected using crowd sourcing, specifically the Amazon Mechanical Turk. After the initial sketch collection, a manual data verification process was applied to obtain a dataset of 20,000 sketches. Along with the fully drawn sketches the individual strokes as they were drawn by each user are provided, thus allowing the exploitation of temporal information. Crowd sourcing was used once more to perform human classification on the dataset. Participants were presented with a sketch and they were asked to choose one of the predefined categories they thought it belong to. Humans recognized on average 73.1% of all sketches correctly, but great variance over categories was observed.

The authors provide a baseline method using SVMs that can recognize 56% of all the sketches correctly. Evaluating an algorithm on the dataset is fairly simple. Following the protocol defined in [39], 3-fold cross-validation is performed. Data are split in 3 partitions; 2 are used for training and the remaining for testing, repeating the process 3 times with different test set each time. Stratified sampling should be used so all the classes are equally represented in the training set. A sample is classified correctly, if the predicted label is the same as the ground truth. The mean accuracy over the test set defines the performance of the evaluated algorithm.

2.4 Conclusions

This chapter reviewed the core aspects of feature extraction and evaluation for a content based retrieval system. Additionally, a literature review on SBIR and sketch recognition was presented along with the available datasets for each task.

Appearance based SBIR techniques follow the simple strategy of local feature extraction and coding with bag-of-features. While the BoF scheme has been successful in generic image retrieval and in SBIR, it does not encode spatial information. That means the position of a local patch in the image is not taken into consideration. Research towards a spatially-aware SBIR system has been limited [40, 18], perhaps due to the positive results of the BoF approach. Our intuition is that spatial consistency will improve image/sketch matching, and we study it in Chapter 3.

Another under-explored aspect in SBIR literature is scalability. Most methods perform experiments on small databases of few hundred images and omit complexity analysis, or it is clear that they cannot cope with large amounts of data. Large-scale evaluations have been carried out in [40, 18]. Sub-linear times and parallelizable queries is a key property for a contemporary retrieval system.

As highlighted before, the research field of SBIR is characterized by inconsistencies in evaluation. Most of the published work uses arbitrary image datasets and metrics for evaluation, without making them available to the research community. Hence, a fair comparison cannot be always conducted.

Our work in SBIR is evaluated, and achieves state-of-the-art performance, in the online public datasets of Section 2.2.6. Therefore, the validity of our results is ensured and future reference and comparison against other methods is made transparent.

Approaches to sketch recognition vary depending on the nature of drawings. The sketching style of each domain dictates the appropriate feature representation. In a typical SBIR paradigm, an average user possesses moderate to low drawing skill. As a result, drawings of the same object encounter great variance. From this point of view, the EitzSKETCH dataset provides an appropriate platform to test the recognition accuracy of an algorithm with hand-drawn sketches. Current research indicates that a learning approach combined with a re-ranking step based on sketch matching could offer promising results. Additional attention should be given to visual properties more appropriate to sketch description and how they can benefit recognition accuracy.

Chapter 3

Sketch Based Image Retrieval via Patch Hashing

Contents

3.1 Introduction

In this chapter we describe our research in sketch based image retrieval focusing on image content. The desired outcome is to retrieve all the database images that are visually similar to a given hand-drawn sketch query. To expedite the drawing process, input sketches are restricted to simple black and white contour drawings. A successfully SBIR should handle well image/sketch matching and allow for scalability.

We tackle these challenges by decomposing a sketch query into several overlapping patches and retrieving image patches near-duplicate in terms of shape using a hashing technique. Contrary to the bag-of-feature approach, which abstracts out spatial information, our method assumes that users are looking for images spatially consistent with their query. For instance, if they draw a sunset scene and the sun has been placed at the top right of the canvas, images displaying the sun in approximately the same location will be preferred. Therefore our system is designed to retrieve visually similar images to the hand-drawn sketches up to small translations. At the core of our retrieval scheme lies min-hash, a set similarity estimation technique originally applied to identify duplicate web pages [14] and later modified for near-duplicate image search [27]. Our method differs from [27] where an image is described by a single set of visual words. Similarly to [72], we extract an ensemble of local image patches, yet our approach employs different patch description process and relies on a novel patch voting system to infer a ranking on the database images. Each patch is represented with a binary version of the HOG [34] descriptor which allows the utilization of the min-hash algorithm. This provides a more compact and spatially aware image description. In contrast with most SBIR techniques that employ the bag-of-features scheme and do not encode spatial information [42, 55, 56], we incorporate structural information in sketch/image matching and demonstrate that it significantly improves matching quality. Our scheme offers flexibility at query time, since we can omit patches that have not been filled during drawing. An index structure facilitates fast queries and online result updating. Instead of indexing each pixel's location as in [18], we suggest a more efficient patch based look-up mechanism. We evaluate the retrieval accuracy and scalability of our framework with three available datasets and demonstrate state-of-the-art results.

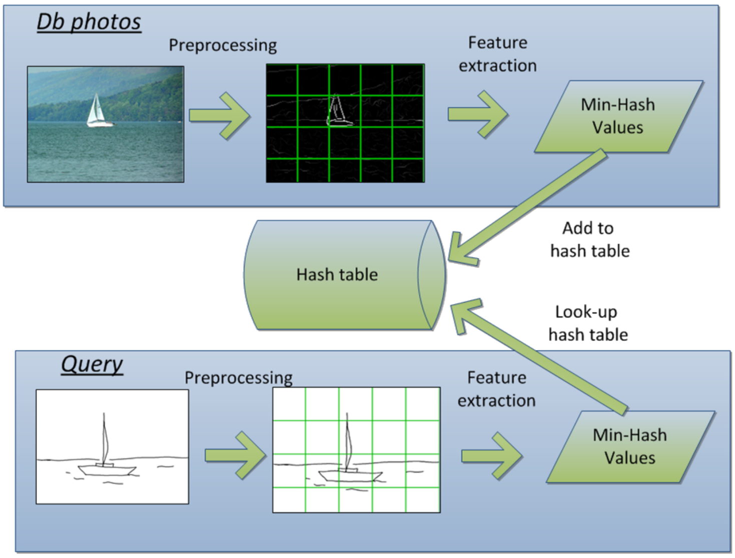


Figure 3.1: General pipeline of the patch hashing framework

Moreover, we examine the effect of affine variations on our framework and offer complexity analysis.

The rest of this chapter is structured as follows: in Section 3.2 we experimentally verify the benefits of spatial information in SBIR; Section 3.3 details our patch-hashing SBIR framework. Extensive experimental evaluation is given in Section 3.4 along with complexity analysis of our method in Section 3.5. Conclusions are presented in Section 3.6.

3.2 Benefits of a Spatially Aware Approach

In this section we verify experimentally our intuition that a strong image/sketch match goes beyond the bag-of-features (BoF) approach and also involves structural similarity. A well-defined image/sketch pair needs to correlate in appearance and in spatial configuration. For this purpose, we employ a hierarchical spatial grid on each image and demonstrate improvement in the retrieval accuracy over the BoF approach in the EitzSBIR dataset.

To represent images and sketches we use the Pyramidal HOG (PHOG) descriptor [13]. The descriptor is not based on local visual words, but on a spatial pyramid [15]. The image shape is represented in the form of edge orientations and magnitude histograms, as in HOG, for several spatial pyramid

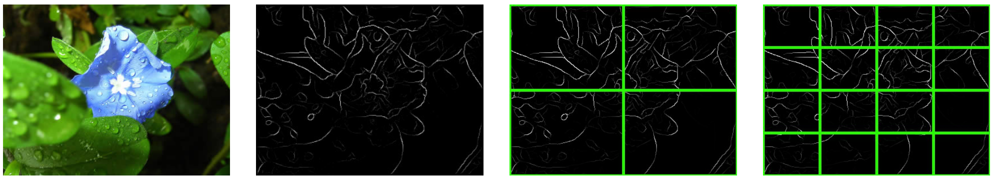


Figure 3.2: PHOG spatial pyramid

levels. The pyramid's spatial analysis gets finer as the level rises. An example is illustrated in Figure 3.2. In the first level ($l = 0$) a histogram of edge orientations is computed for the whole image. Then in the second level ($l = 1$) the input image is tiled in each dimension. The procedure continues by squaring the number of tiles on each axis, thus creating finer spatial grids, until a top level is reached,. This is a pyramid representation because the number of points in a cell at one level is simply the sum of those contained in the four cells it is divided into at the next level. Additionally, the spatial pyramid is applied to the high resolution image, so there is no scale-space Gaussian smoothing performed beforehand. Therefore, matching of objects in different scales is not possible under this scheme, which allows us to focus on the impact of spatial information in image/sketch matching. Moreover, scale invariance plays a secondary role in SBIR as look-alike images to drawn sketches are preferred. The final descriptor is a weighted hierarchical concatenation of the histograms in each level.

First, the edges are extracted for each database image, thus information that is not useful in sketch matching is reduced and shape characteristics are highlighted. Next, the spatial pyramid is applied to each image for L levels. Typically, L does not exceed 3 because beyond that point the descriptor becomes too local oriented and cannot cope with spatial variations of different images. Moreover, as the level of the pyramid grows, the dimensionality and the time of computation for the descriptor increases in parallel, imposing further performance barriers.

Since the spatial grid is doubled for each axis from one level to the next, we can calculate the grid density in each level as $\sum_{l=1}^{L} 4^l$ where l is the current level and L the total number of levels. For each tile at each level a histogram H_l of edge orientations is calculated with dimensionality K, which is the number of

Table 3.1: Evaluation of PHOG in EitzSBIR dataset for several distances.

	$K = 20$	$K = 40$	$K = 60$	$K = 80$
NHI	0.294	0.293	0.295	**0.302**
x^2	0.290	0.292	0.291	**0.299**
cosine	**0.280**	0.2653	0.252	0.250

orientation bins and is empirically tuned. In total, the dimensionality of H_l at level l is $K \sum_{l \in L} 4^l$. The histograms for each level are concatenated to form the final feature vector. The PHOG vector is normalized to sum to unity. Normalization ensures that images with more edges, for example those with high texture content, are not weighted more than others. An implementation with $K = 20$ and $L = 3$ will result in an 1,700-dimensional vector.

Images I_1 and I_2 represented by a PHOG descriptor can be matched by summing the distances between each level histogram $\mathbf{h}_l$:

$$D\left(I_1, I_2\right) = w_l d_l \left(\mathbf{h}_l^{(1)}, \mathbf{h}_l^{(2)}\right) \tag{3.1}$$

Where w_l is the weight at level l and $d_l(\cdot, \cdot)$ is the distance between I_1 and I_2 at pyramid level l. The PHOG descriptor is essentially a histogram, so recommended distances are the Normalized Histogram Intersection (NHI), the χ^2 distance and the cosine distance. The Euclidean distance is not suitable to compare histogram dissimilarities. Moreover, instead of weighting equally each pyramid level, it is better to value higher finer resolution levels than those at coarser resolution. A weighting scheme where $w_l = 1/2^{(L-l)}$ is proposed, yet depending on the application an empirical weight assignment may provide improvements. PHOG offers insensitivity to small rotations and a compact vector representation, due to the statistical description of edge orientations. Additionally, it benefits from spatial flexibility and is able to capture both coarse and fine shape similarities.

We evaluate the PHOG performance on the EitzSBIR benchmark, see Section 2.2.6.1, using a selection of dissimilarity metrics. Table 3.1 summarizes the results. For all database images, we first perform an edge detection step and we apply the PHOG descriptors in the edge maps. The PHOG descriptor is applied directly to the sketch queries.

By introducing the spatial pyramid, we observe a 9% increase from the

BoF approach of [42], in the EitzSBIR benchmark. PHOG achieves a correlation value of 0.302 when the BoF-SHOG approach scores 0.277. The dataset encompasses a large range of sketches and images, hence the observed increase is not negligible. We attribute the performance boost to the encoding of spatial information via the spatial pyramid, as the two schemes use a similar edge orientation description of patches and are otherwise the same. Correlation values are consistently higher than BOF-SHOG under a large range of orientation representations (K parameter). The NHI and χ^2 distances present the best performance with the cosine distance following closely. It is evident that increasing the number of orientations bins to more than 20 does not improve retrieval performance. In some cases, it might even introduce noise as hand-drawn sketches consist of few major orientations, although we haven't observed it in our experiments. A configuration of $K = 20$ and $l = 3$ results to a descriptor of 6.6KBs, if we assume each value is represented by a 4-byte float. A linear search in the dataset of 100,000 images takes approximate 3.2 seconds. Obviously, indexing and parallelization of the code could improve this figure, still the NHI and χ^2 distances are expensive to calculate.

Having demonstrated the advantages of the structural aware representations in SBIR, we proceed in the following section to describe a scalable and robust SBIR framework that will further enhance the state-of-the-art.

3.3 Retrieval Based on Patch Hashing

In this section, we describe how we incorporate the unification of patch location and description in a scalable framework for efficient image retrieval. We retrieve similar images to a sketch query based on accumulated similarities between local patches. Every image in the database is divided into overlapping squared blocks and for each region an edge orientation distribution is extracted. A reverse index is built on the unique min-hash values/location pairs pointing to the patches containing these values. A sketch query undergoes the same process and for each sketch patch, we look into the index to retrieve similar patches at nearby locations. Every index hit contributes a vote to the corresponding image and the final ranking is generated by summing the votes for each image. Min-hash is employed to estimate the similarity between two patches. Chum et al. [27] proposed to represent an image by an unordered set of visual words acquired by clustering on the feature space. Under this scheme, an image is represented by a single sequence of 0 and 1. Our approach adopts a sequence description for every local patch, but instead of utilizing a visual codebook to derive the sequences, we use the non-zero indexes of the binarized patch descriptor. An overview of the core modules of our method is presented in Figure 3.3

3.3.1 Min-Hash Overview

Min-hash is a probabilistic technique that can be used in conjunction with Local Sensitive Hashing [58] to estimate similarity between sets. Assume a set S of tokens x of size $|S|$. The set $S_i \subseteq S$ can be represented by a sequence of size $|S_i|$ where the presence of a token $x \in S_i$ is indicated by 1 and the absence by 0. The set overlap similarity, or *Jaccard similarity*, between two sets S_1 and S_2 is defined as the ratio of their intersection and union and is a number between 0 and 1; it is 0 when the two sets are disjoint, 1 when they are equal, and between 0 and 1 otherwise.

$$\text{Jaccard Similarity}(S_1, S_2) = sim(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \tag{3.2}$$

Eq. (3.2) values equally every member of the set. In [27], an extension is proposed to assign to each set token x a weight according to its importance. Min-hash approximates the set similarity by creating a random hash function $h : S \to \mathbf{R}$ sampled from a uniform distribution, mapping each element of S to a real number. Each hash function h defines an ordering on the members of S. Min-hash is defined as the smallest element of S under ordering induced by h.

$$\text{min-hash}\,(S, h) = \min_{x \in S} \{h\,(x)\} \tag{3.3}$$

The outcome of Eq. (3.3) is a real number for each input set. The probability of two sets having the same min-hash value is equal to their Jaccard similarity.

$$P\,(\text{min-hash}\,(S_1, h) = \text{min-hash}\,(S_2, h)) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = sim(S_1, S_2) \tag{3.4}$$

The above holds because h is a function sampled randomly from a uniform distribution. Therefore each x has the same probability of being the minimum element. If x is drawn randomly from $S_1 \cup S_2$ and min-hash $(S_1, h) =$ min-hash (S_2, h), then S_1 and S_2 have x in common. If the min-hash values are different then x is not a shared element. An unbiased similarity estimation is obtained by computing for each set S_i, k independent min-hash functions h_k and counting the occurrences of identical min-hash values for the two sets. To reduce the possibility of false positive retrievals the min-hash values of s independent hash functions are grouped together to form s-length tuples (often called sketches). Two sets are characterized similar, if they share at least one tuple.

The probability of two sets sharing j min-hash tuples out of k is given by the binomial distribution:

$$P(\text{min-hash}\,(S_1) \overset{j}{=} \text{min-hash}\,(S_2)) = \binom{k}{j} p^{js}(1-p^s)^{k-j} \tag{3.5}$$

where $p = sim(S_1, S_2)$, k is the number of min-hash tuples and s is the number of independent hash functions used to form an s-tuple.

The min-hash tuples can be computed fairly fast (linear in the size of S_i) and given two tuples the resemblance of the corresponding sets can be computed in linear time in the size of the tuples. The probability of collision under this scheme is:

$$P\{h(S_1) = h(S_2)\} = 1 - (1 - sim(S_1, S_2)^s)^k \tag{3.6}$$

Min-hash has been successfully applied to text [14] and image [27] domains to detect near-duplicate instances of a given set.

3.3.2 Patch Description

A well-defined patch description will set solid foundations for robust matching. A patch summary is obtained via a three-step process. First, all images are subjected to a preprocessing operation. Then, a visual signature is extracted for each patch. Finally, the descriptors are binarized in order to be hashed.

Preprocessing. We apply an edge detection filter to each image in the database. As the feature extraction will take place on the detected edge map, a human-like contour detection is of paramount importance. We found that if we keep the image dimensions reasonably small we can benefit from the excellent performance of the Berkeley BG detector [89] without burdening much the computational unit. Hence, every image is scaled by keeping the aspect ratio fixed and setting the largest side to 200 pixels. The rescaling also speeds up the feature extraction process at the expense of possible slight loss in retrieval accuracy. Edge detection is carried out via the BG detector in less than a second. Weak detected contours are further reduced by thresholding the returned edge map. Depending on the application, we can set the threshold high to keep only very strong basic lines or lower if more details are required. In the case of a hand-drawn sketch input, the edge detection step is substituted with a morphological thinning operation, which reduces thick drawn lines to single-pixel width. This removes drawing artifacts, such as double edges on each side of the trace of a line.

Feature Extraction. An overlapping spatial grid is applied to describe the generated edge map finely and feature vectors are extracted for every patch of the grid. The grid size and patch size are parameters that need to be tuned for each dataset. We found that an implementation with grid size equal to 17×17 and patch size of 40×40 performs well in the general case. Sketches contain sparse visual information, therefore local patches usually cover a large region of the image. These two parameters regulate how densely an image is sampled, allowing to choose the detail depth of the representation. The patch extraction process is visualized in the top part of Figure 3.3. Two patches are considered similar if they share shape characteristics, i.e. their edges have similar orientation histogram and spatial arrangement. We propose to quantify this similarity with the HOG descriptor, known to perform well in general object detection problems. Moreover, descriptions relying on histograms of oriented gradients achieve superior performance in SBIR, according to the literature [42, 56, 55, 41].

Binarization. Descriptions extracted from the previous process return real valued histograms. In order to make the descriptor vector compatible for use

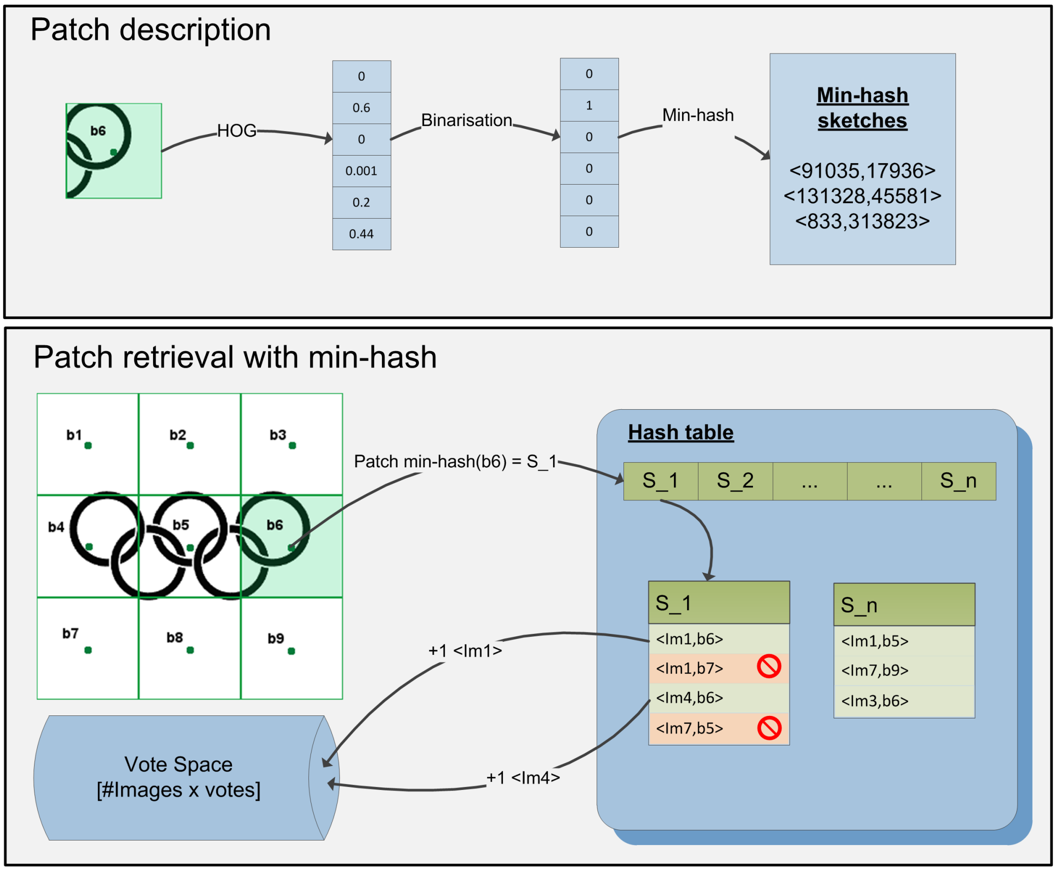


Figure 3.3: Patch retrieval framework overview. (Top) Feature extraction for a single patch. The HOG descriptor is computed and then a binarization process is applied to the feature vector. A list of min-hash tuples calculated from the non-zero descriptor indexes is the final patch representation. (Bottom) The patch retrieval mechanism. Every image patch is described separately, for each computed min-hash tuple a look-up in the hash table is performed. We only allow votes originating from neighboring patches.

with the min-hash algorithm, a binary representation is required. Min-hash expects a set S as an input. In [27], each image entry is represented by a set of visual words. This can be formed as a binary vector where a word presence/absence is indicated by 1/0. We choose to hash individually each local patch descriptor instead of a global BoF vector. We modify the HOG vector to abide to this scheme. Without loss of crucial structure orientation information, we can binarize the descriptor by setting the $b\%$ highest orientation values to 1 and the rest to 0. Parameter b is a scalar in the range $[0, 1]$. The binarization process is tailored to sketch/image matching as it highlights

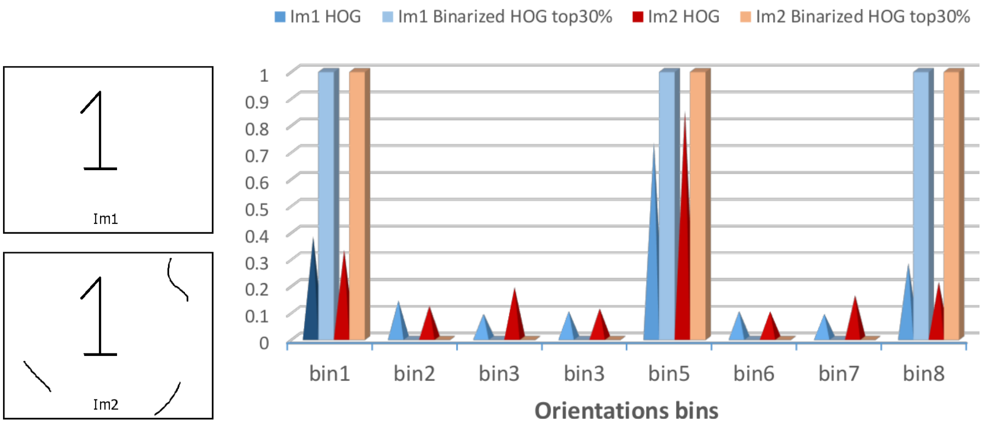


Figure 3.4: Example of the HOG binarization process. The outcome binary vector if we set the top 30% of the bin values to 1 and the rest to 0 is identical for the noise-free (Im1) and the noisy image (Im2). Best viewed in color.

the strongest patch orientations corresponding to solid continuous contours, while eliminating weak responses from noisy edges. A visual example is given in Figure 3.4. A HOG descriptor with 8 orientation bins is extracted from a noise-free and a noisy sketch representation of the digit 'one'. The noisy image has been chosen to emulate artifacts that are frequently present in sketches, thus few random strokes were added to the original image. As expected, the noise-free histogram has three clear peaks corresponding to the three main line orientations of the digit. On the other hand, the stroke artifacts introduced in the second image modify the orientation distribution. We observe, after binarization of both histograms by setting the top 30% of their values to 1 and the rest to 0, that their vectors are identical and accurately maintain the three dominant orientations of the image, without being affected by the injected artifacts. As we assess similarity between many local patch pairs there is no need for elaborate representations. This scheme captures the local structure of the images and by combining several local patch matches offers rich higher-level correspondences. Finally, for each binarized descriptor we calculate k s-tuples of min-hash values which will be used to efficiently retrieve similar patches.

3.3.3 Location-Aware Reverse Index

To estimate patch similarities, we employ the min-hash algorithm. The feature extraction step provides a set of min-hash tuples of length s for each patch. To assess similarity between images one should count how many common min-hash values exist between the two patch collections. An appropriate data structure for this purpose, that allows constant-time look-ups, is a *reverse index hash table*.

We would like to encode spatial information into our framework, hence we introduce spatial constraints in the matching scheme. The idea is to discard matches between distant image regions. In other words, a successful match is defined between two patches that are visually similar and approximately located at nearby image regions. We suggest a *location aware reverse index* built on the collection of min-hash tuples extracted from the dataset. A hash key is defined for each unique min-hash tuples. For each key, we store the identifier of the image that the current patch originates from, along with the location of the patch. The entries of the *key*-index structure are in the form <image-id,location>. The location information can be capitalized during the query process by rejecting non adjacent patches.

The probability of min-hash collisions depends on the patch similarity (Eq. (3.6)). In practice, as the database size grows patch duplicates are more likely to occur, leading to high probability of hash collisions. This results to large buckets for each index key and propagates complexity at the query stage, where the entries of each hash key need to be linearly accessed. As a consequence, query efficiency can deteriorate drastically. We tackle this issue by encoding the location information in each key instead of each bucket entry. An illustration can be found in Figure 3.5. We create an equivalent index structure, named *key-location*-index, where the hash key is defined as the pair <key,location>. As value, we only store the image id. This setup requires slightly more space to handle the increased hash key pointers but pays off in time efficiency. In *key-location*-index, each query accesses far less entries and also omits the expensive computation of the locality constraint as will see in the next section.

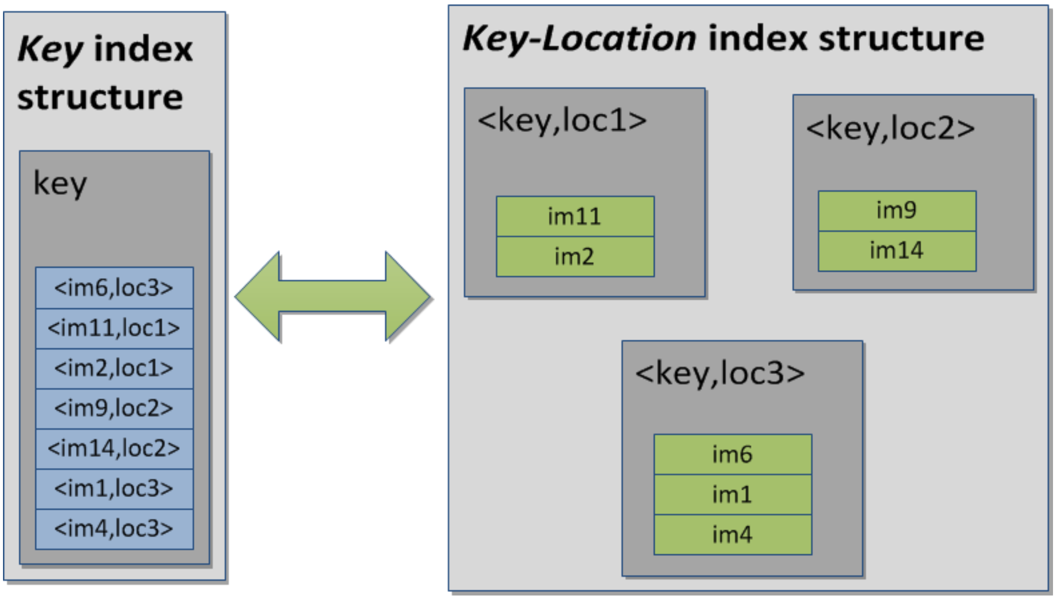


Figure 3.5: Two equivalent location-aware indexing structures. The *key*-index structure requires less space and more query time. The *key-location*-index is more time efficient but requires slightly more space.

3.3.4 Spatial Voting

Images similar to a sketch query are returned based on a voting process (bottom of Figure 3.3). The pipeline of the query step is as follows: given a binary drawing, features are extracted according to the process illustrated in the feature extraction paragraph; obviously the edge detection step is omitted. For every non-empty patch, k min-hash tuples are computed and for each tuple a look-up in the *key*-index is performed. If the key is found in the reverse index, we iterate through the entries and add a vote to the corresponding images. The locality constraint is enforced by discarding patches located further than a predefined distance threshold from the current examined patch. This requires an additional distance evaluation for each examined entry. Evidently, the proposed *key-location*-index expedites the voting process. The location information is embedded in the hash key and a successful look-up in the table returns, in constant time, all the images that contain visually similar patches at the same location as the examined patch. Additionally, the buckets contain fewer entries and can be quickly traversed. If there is a need to expand the spatial search radius, we can simply generate *key-location* queries for each patch by fixing the key and inserting nearby location coordinates to check.

An indexed image T is represented by a collection $\mathcal{T}$ of *key-location* values. A given *key-location* value v scores a hit on $\mathcal{T}$ if $v \in \mathcal{T}$.

$$hit(v, \mathcal{T}) = \begin{cases} 1, & \text{if}\, v \in \mathcal{T} \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

Based on Eq. (3.7), the matching score between a query $\mathcal{Q}$ and a database exemplar $\mathcal{T}$ is defined as:

$$score(\mathcal{Q}, \mathcal{T}) = \sum_{v \in \mathcal{Q}} hit(v, \mathcal{T}) \tag{3.8}$$

where v is a *key-location* hash value and $\mathcal{Q}$, $\mathcal{T}$ collections of *key-location* values. The final ranking is generated by assigning a score to each image, which corresponds to the number of votes it received. The higher the score, the better the given image matches the query.

The suggested patch based retrieval scheme enhances flexibility since lookups take place only for patches that have been drawn by the user, efficiently reducing query time and facilitating real-time result updating when a new stroke is drawn. The query routine can be easily parallelized to enhance scalability even further. Patch queries can be executed independently on different machines and return a partial vote count for every image. An integration process will then merge all the votes to generate the final ranking. The flexibility of spatial constraints can be controlled by the r parameter. Low values enforce strict structure correspondences, while higher values allow broader matches.

3.3.5 Handling Bias

Bias in patch hashing originates from two sources: a) images with rich contours and b) frequent min-hash tuples. Bias can significantly deteriorate the performance of a sketch based image retrieval system. Images with many edges will receive more votes than images with sparse contours and will dominate the rankings. One way to deal with bias is to assign a weight to each vote. Votes cast to dense images will weight less than votes cast to sparse images. This scheme adds additional time and space complexity, as there is need to keep a record of each image's density factor and perform a weight

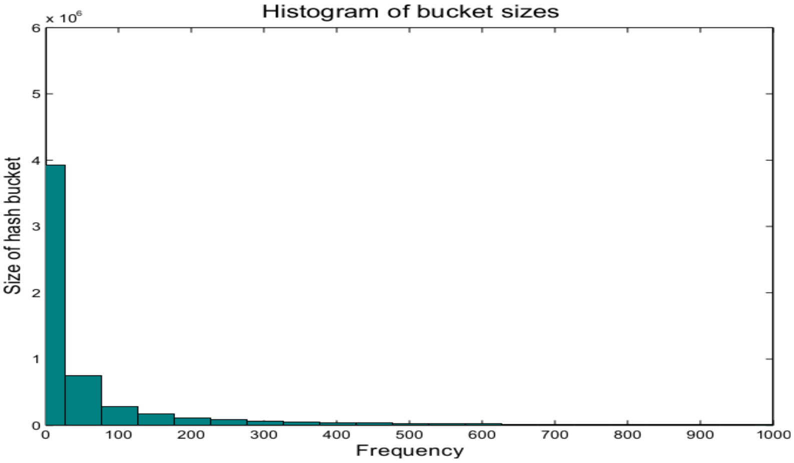


Figure 3.6: Histogram of bucket sizes of the *key-location*-index built on 100,000 images.

calculation for each vote.

We suggest an alternative way to resolve the bias effect that will further alleviate the index structure from redundant information. During index construction, we impose each image to be tied uniquely with a min-hash value. This means that when an image contains two or more patches with the same min-hash value only one will get indexed. This process efficiently prunes a large volume of duplicate entries and at the same time reduces the bias effect. By extracting several hash values for each patch, we ensure that no patch is under-represented in the index even if some of its min-hash entries are discarded.

Another source of bias is frequently occurring min-hash tuples. This is the equivalent of very common words in text retrieval, such as articles or conjunctions. By discarding these values we reduce the noise, thus producing a more discriminant index. Figure 3.6 illustrates the histogram of the frequency of hash keys over a database of 100,000 images. We observe a long-tail distribution. Most of the hash buckets contain a small number of entries. There are only few buckets with key count, corresponding to frequent patches, and we can efficiently prune them by applying a predefined cut-off threshold f.

3.4 Experiments

We evaluate our patch hashing framework on three publicly available SBIR datasets. We compare our results against the state-of-the-art for each benchmark. Table 3.2 summarizes the best results across all databases. We also examine how various parameters affect retrieval performance. Namely, the density of the grid size for feature extraction, the binarization threshold b corresponding to the top b% histogram values and the robustness of patch-hashing to affine variations. We found that the min-hash parameters k, s corresponding to k s-tuples for each patch have little effect on the results, hence we fix them to $k = 50$ and $s = 2$ for all the experiments.

3.4.1 Datasets

Flickr160. Flickr160 consists of 160 images and 25 hand-drawn sketches assigned to 5 categories. For each sketch category there are 32 associated images in the database. The evaluation metric is the Mean Average Precision (MAP) over all the queries as computed by the VLFeat library [125]. Note that we use the MAP score over all queries instead of the interpolated score used in TRECVID benchmarks [97]. The best performing method in the dataset is a bag-of-features approach based on the GF-HOG [55] descriptor. We follow the preprocessing steps of Section 3.3.2. Each image is downscaled and the BG edge detector is applied with threshold 175. During the *key-location* index construction we discard hash keys that occur more than 7,000 times in the database. At voting stage, for each patch we also look-up neighbor patches with Manhattan distance less than $r = 2$ from the original location.

Flickr15k. Flickr15k [56] can be seen as an expansion of the Flickr160 dataset. In this study, 10 non-expert subjects generated a sketch query collection for 33 topics, describing shape, building landmarks, objects and scenes. The total number of sketches is 330. For each topic a range of photographs collected from Flickr. The dataset includes 14,660 images of ranging affine variations and background clutter. Most of the images are associated with a sketch category. A part of the database does not belong to any category and serves as noise. The evaluation metric used in this dataset is the MAP

Table 3.2: Results comparison on three SBIR datasets. The metric in Flickr160 and Flickr15k is the MAP, while in EitzSBIR is Kendall's τ.

Method	**Flickr160**	**Flickr15k**	**EitzSBIR**
Tensor [41]	0.270	0.07	0.223
BoF SHOG [42]	0.420	0.109	0.277
BoF GF-HOG [55, 56]	0.540	0.122	N/A
Keyshapes [103]	N/A	N/A	0.289
Keyshapes + SHOG [103]	N/A	N/A	0.337
PH-HOG	**0.590**	**0.200**	**0.341**

over the set of queries. All sketch queries are translated so their centroid is at the center of canvas. For the experiments the BG detector threshold is set to 100. We also define the index bias threshold f to 200,000 and the Manhattan distance threshold r for nearby patch look-up to 4.

EitzSBIR The EitzSBIR benchmark [42] consists of 31 user-drawn sketch queries outlining objects and scenery. Each sketch query is associated with 40 photos assigned with a value between 1 (similar) and 7 (dissimilar). These 1,240 photos are mixed with 100,000 distractor images. A SBIR algorithm must generate a ranking of the database images for each query and retrieve the order of the 40 query-related photos. The Kendall's correlation is then calculated between the algorithm's ranking and the ground truth for a given query defined in (2.12) The final benchmark score is the average correlation value across the 31 queries. In this dataset, we set the thresholds for BG detector, index bias and voting distance to 10, $f = 300,000$ and $r = 2$ respectively.

3.4.2 Descriptors

Along with the HOG descriptor, we study the impact of two other patch descriptors. Our scheme requires binary vectors as input to the min-hash algorithm, therefore we choose the BRIEF descriptor [16], which is inherently binary, and the LBP descriptor [96] with a binarization step. In the experiments, we use our own implementation of the BRIEF descriptor and a public implementation of the LBP [1]. In the BRIEF descriptor, image patches are smoothed with a 9×9 Gaussian kernel with 0.5 variance and 512 normally distributed intensity tests form the final description vector. LBP is employed with the default implementation parameters resulting in a 256-dimensional vector. We compute the HOG features as described in the original paper [34] using an online available implementation [2]. We apply a 8×8 cell grid for each patch and in each cell we compute 8 orientation bins in the range $[0, \pi)$. The final HOG descriptor for each patch is a 512-dimensional vector. In the rest of this section, we denote as PH-HOG, PH-BRIEF, PH-LBP the patch hashing methods with the corresponding descriptors. All descriptors are calculated in a 40×40 pixels image region.

3.4.3 Vector Binarization

We test the impact of vector binarization in the Flickr160 dataset. We fix the spatial grid to 12×12 and apply a range of binarization thresholds. Figure 3.7 presents the binarization effect on PH-HOG and PH-LBP. PH-BRIEF is excluded from the evaluation as it is inherently binary. PH-HOG achieves a clear peak when the top 20% of the values are set to 1. For greater binarization thresholds the discriminant property of the binary descriptors drops rapidly. PH-LBP is not affected by the binarization process. Due to the sparse edge maps of the patches, the resulting vectors contain few non-zero entries. Therefore thresholding the top values does not affect the final description. Based on the above, we fix the binarization threshold to 20% for all the following experimental setups.

3.4.4 Impact of the Spatially-Aware Index

To verify the effectiveness of the location aware index in matching performance, we compare it against a non-spatial version. Additionally, we explore the role of the search radius parameter r during voting. The experiments were conduced in the Flickr15k dataset. Figure 3.9 illustrates our findings. The degradation of performance with an non-spatially-aware index is evident. The

[1] http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab

[2] http://www.mathworks.co.uk/matlabcentral/fileexchange/33863-histograms-of-oriented-gradients

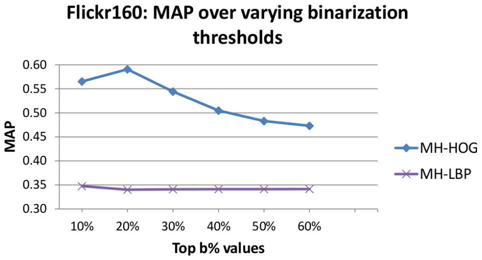


Figure 3.7: Flickr160: Impact of binarization threshold in retrieval. We set the top b% values to 1 and the rest to 0.

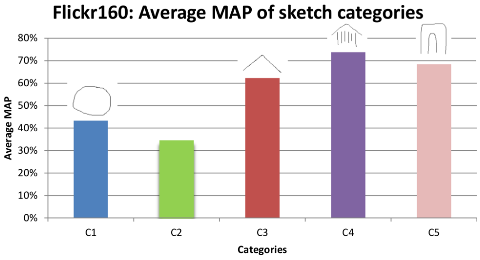


Figure 3.8: Flickr160: Average MAP score for the five sketch categories.

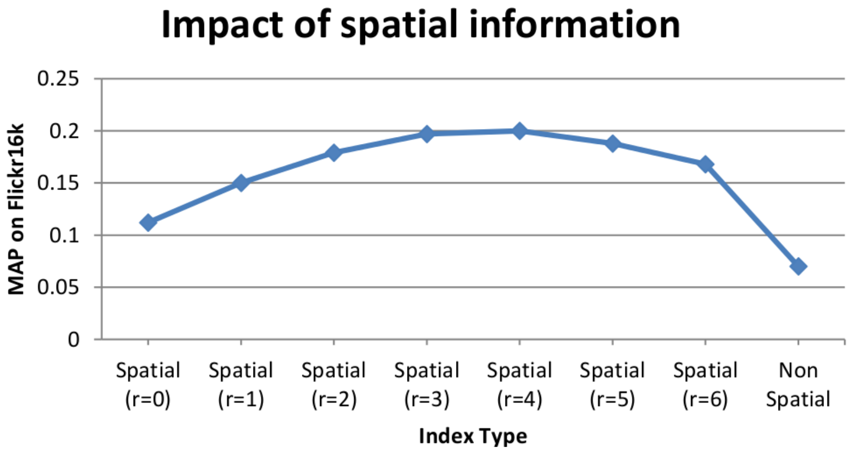


Figure 3.9: Effect of spatial information encoding in retrieval performance. Our spatially-aware index structure improves the MAP score in Flickr15k.

MAP score drops from 0.20 to 0.07. The performance decline occurs because two patches located in arbitrary positions are allowed to be matched. We observe a MAP increase, when spatial constraints are introduced. The gain is small with strict constraints, i.e. low r values, due to the inflexible narrow search window. Values greater than $r = 4$ render the search scope broad and the performance drops. A moderate radius size, between 2 and 4, constitutes a balanced choice between narrow and generic results.

3.4.5 Grid Resolution

We study the effect of spatial resolution of our approach. Local patches are extracted from a uniform sampled grid over the images. Figure 3.10 summarizes the MAP scores over varying spatial grid configurations in the Flickr160

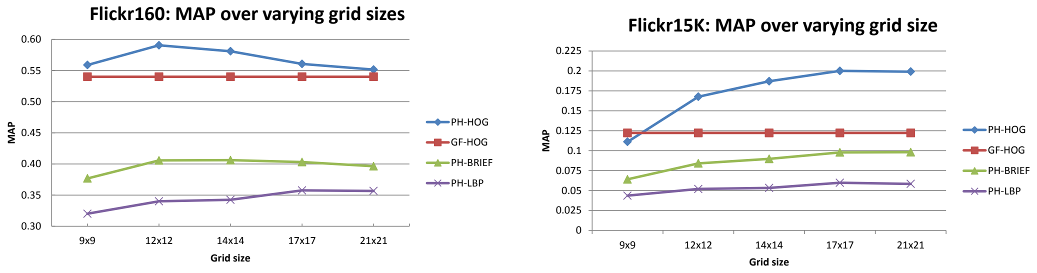


Figure 3.10: MAP scores over varying spatial grids. GF-HOG method is displayed for comparison purposes. Left: Results on Flickr160. Right: Results on Flickr15k.

and Flickr15k databases. Obviously, the denser the grid the more detailed the produced image description. PH-HOG outperforms the rest methods and is consistently better than GH-HOG. The highest MAP in Flickr160 is observed under a 12×12 grid. Performance gradually deteriorates as sampling becomes denser. Similar behavior under grid variations is presented for PH-BRIEF and PH-LBP. Due to the small size of the database a moderate spatial resolution is adequate for efficient image/sketch matching. We also notice the large performance variation between HOG and the other descriptors. This finding agrees with previous studies [42, 56, 55, 41] suggesting HOG as the most appropriate descriptor in SBIR. We attribute the low performance of PH-BRIEF and PH-LBP to the noise-sensitive intensity check mechanism these method implement. Sketch patches contain sparse information biasing the produced description vectors towards many zero values. BRIEF performs better than LBP due to the patch smoothing filtering which reduces to some extent the noise.

In Flickr15k, the findings are slightly different (right part of Figure 3.10). Retrieval accuracy rises as the spatial resolution increases and stabilizes at more fine grid configurations. We attribute this change to the larger database size which introduces image variations . As a result, more local details are required to accurately represent them. The descriptor performance follows the same trend as in Flickr160. HOG is the leading descriptor while BRIEF and LBP following by large margin.

3.4.6 Retrieval Quality

The best MAP score in Flickr160 is 0.590, achieved on a 12×12 spatial grid by the PH-HOG. This represents a 9% increase in performance comprated to the previous state-of-art best score. The spatially-aware patch matching enables more robust high-level level similarity estimation than the BoF approach. The *key-location*-index requires 16.2MB of RAM. The total number of unique *key-location* hashes is 404,969. The median number of entries in each bucket is 1. Empirically, the average bucket has approximately $\log N$ entries, where N is the number of images. The average query time excluding feature extraction is approximately 9ms.

In Flickr15k, we notice a vast increase in retrieval performance by our patch hashing scheme (see Table 3.2. Specifically, PH-HOG with a 17×17 grid configuration increases the state-of-the-art by 64% reaching a MAP score 0.200. In this setup, the hash index occupies 711MB of memory and the average query time is 0.2 seconds. The median hash bucket contains 4 entries. Again, we observe that the bucket size scales logarithmically to the size of the database.

The EitzSBIR dataset is the largest database of the evaluation in terms of image volume. Eitz *et al.* report a correlation score of 0.277 with their tensor descriptor [42] in conjunction with the BoF model. Recently, a key-shape approach achieved a slightly better performance at 0.289. The same study also reports a correlation value of 0.337 by combining keyshapes with the SHOG descriptor. The keyshape algorithm cannot scale well as it incorporates a cubic-complexity matching step. Our patch hashing framework outperforms the previous results. We report an average correlation value of 0.341. The 100,000 database images are indexed in 2.9GB of memory. The average query time is 0.2 seconds. We note that the query time does not increase as the database size grows by an order of magnitude. This is attributed to the logarithmic scaling of the number of buckets entries. For this dataset the median bucket size is 9. Appendix A illustrates several retrieval results over all the evaluated datasets.

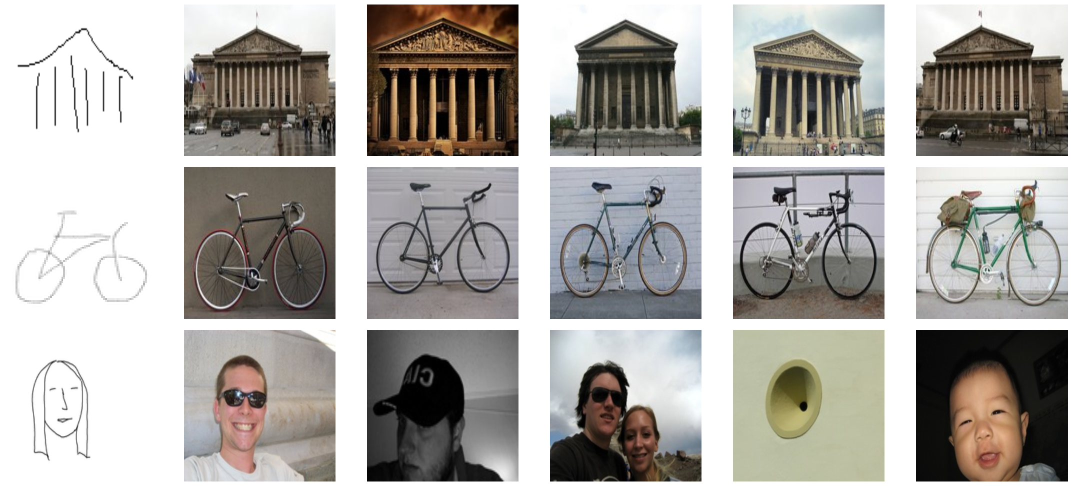


Figure 3.11: Top 5 retrieved images for the best performing queries in the three datasets. From top to bottom Flickr160, Flickr15k, EitzSBIR.

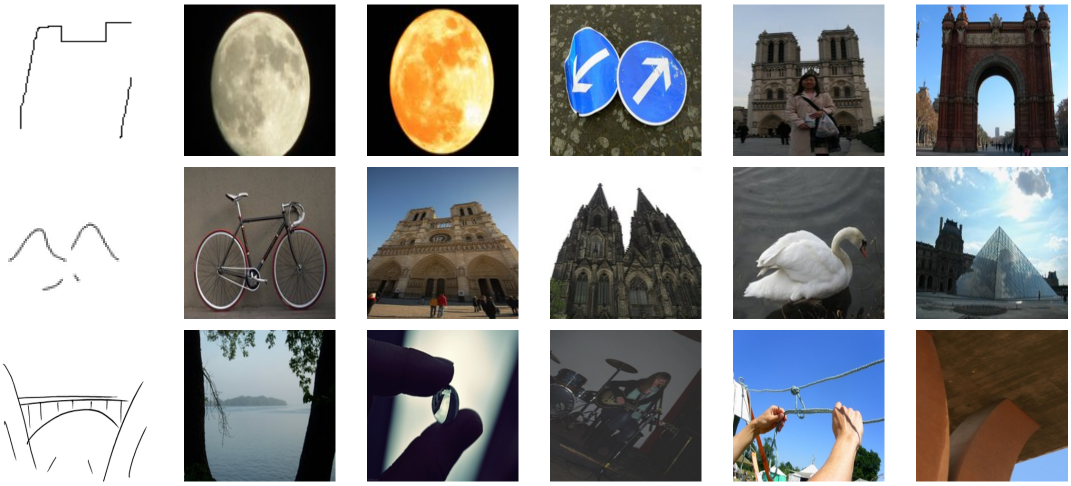


Figure 3.12: Top 5 retrieved images for the worst performing queries in the three datasets. From top to bottom Flickr160, Flickr15k, EitzSBIR.

3.4.7 Evaluation Under Affine Transformations

We examine how our patch hashing framework copes under affine transformations. Following the study in [56], we apply a series of transformations to the query sets of Flickr15k and EitzSBIR. Specifically, we apply translation in a random direction between $[-40, 40]$ pixels, rotation between $[-20, 20]$ degrees and uniform scaling with factors in the range $[0.6, 1.4]$. Figure 3.13 summa-

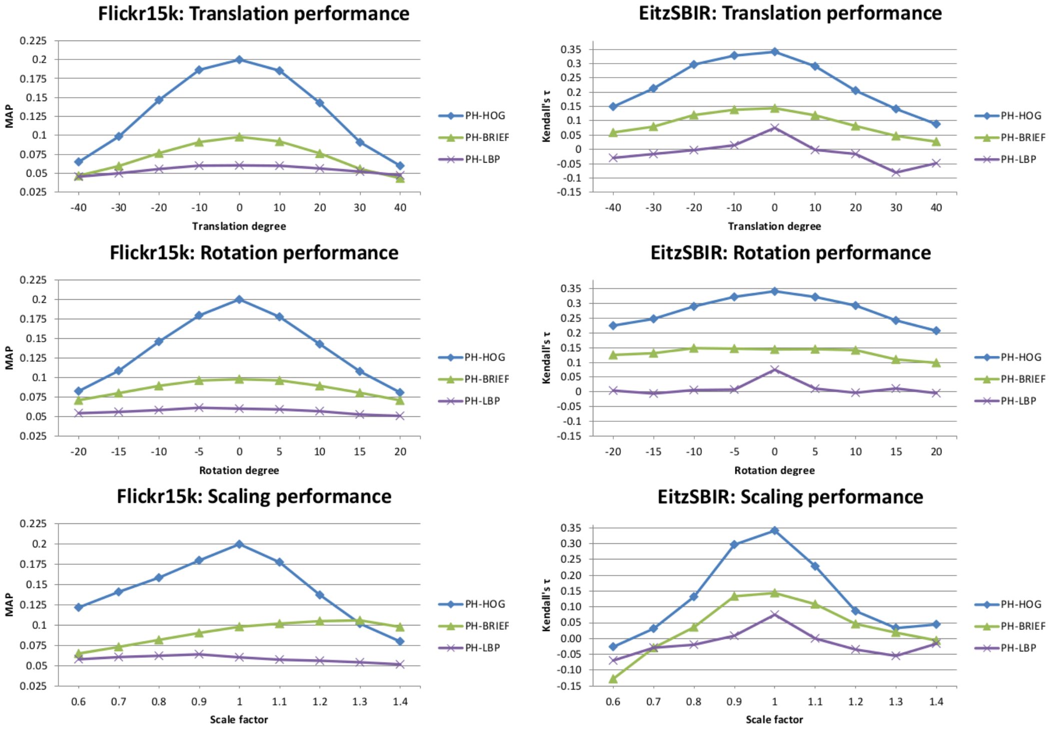


Figure 3.13: Performance variation of patch-hashing under affine transformations in Flickr15k and EitzSBIR datasets.

rizes the results. As expected by our design choices, retrieval performance deteriorates when the magnitude of transformation increases. Scaling appears to be the most challenging transformation for the patch hashing framework. The descriptors display similar behavior under the affine variations with the PH-HOG maintaining the performance edge. Affine invariance is not a fundamental feature in SBIR, still it can assist in few cases. For instance, manually cropped or scaled images may not be processed well under our current scheme. Insights in making patch-hashing invariant to affine transformation are given in section 3.6

3.4.8 Limitations

Figure 3.8 presents the average MAP per sketch category in Flickr160. We note that category C2 performs considerable lower than the rest. Figure 3.14 clarifies the reason. Categories C2 (Notre Dame) and C5 (Arc de Triomphe) contain visually similar queries and images. Sketches of category C2 outline

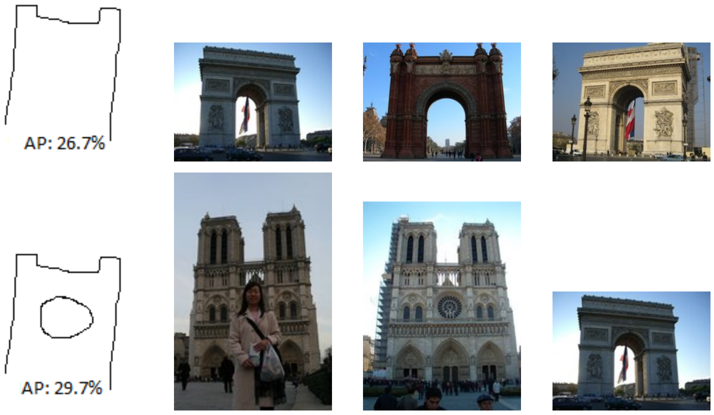


Figure 3.14: Upper row: Top 3 retrieved images for a query of the worst performing category in Flickr160. The original sketch query does not contain many details to precisely describe the correct category -Notre Dame-. Bottom row: Top 3 retrieved images for a more detailed version of the same query. This query depicts better the ground truth category, resulting to increased AP.

only the contour of Notre Dame which matches well with Arc de Triomphe images from category C5. To reduce the false positives for this category a more detailed query is recommended. Indeed, we can enhance the performance in C2 category by painting the characteristic circle of the Notre Dame inside the building contour. The new query achieves better AP in the database and the top two images are relevant to the ground truth.

More examples with low performing queries are illustrated in Figure 3.12. In most cases the retrieved images contain edges in the same position and direction as the sketches. The queries display high level of abstraction allowing for several arbitrary matches in the database. Adding semantic information in the query generation process can effectively reduce this ambiguity.

3.5 Complexity

Scalability is a critical attribute for a retrieval system. Here, we study how the complexity of our framework is affected in space and time as the database volume rises

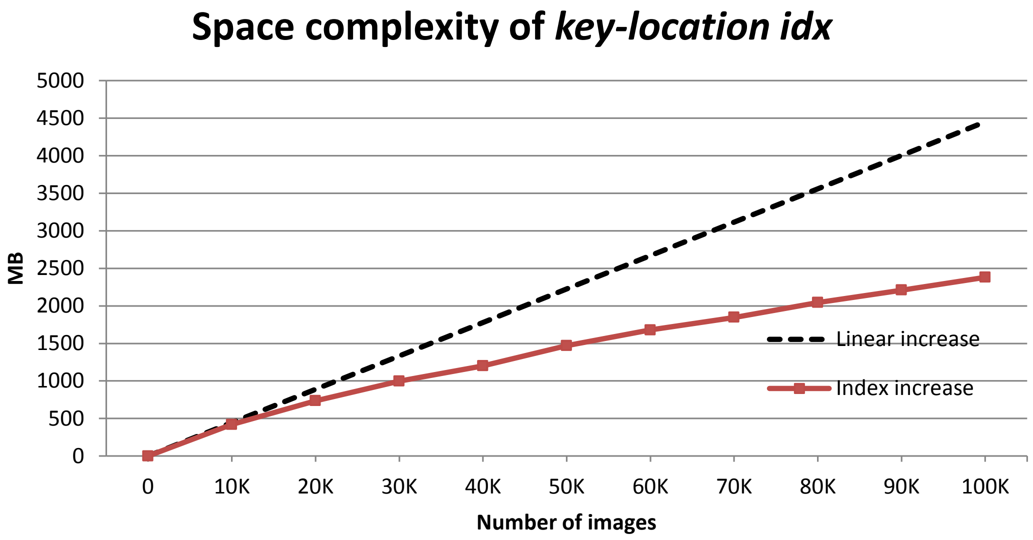


Figure 3.15: Space growth of the *key-location* index under increasing number of images. Size increases sub-linearly to the number of indexed images.

Space Complexity. The *key-location* index structure should be kept in memory for quick accesses. Therefore, maintaining the index size small as the database size increases is important for efficient look-ups. Our scheme requires $s \times 8$ bytes to represent the min-hash key plus 4 bytes to represent the location. s is the number of hash functions that are grouped together to form an s-tuple. Each hash bucket stores image identifiers, so 4 bytes can accommodate an adequate number of unique identifiers. An index that scales linearly to the database size can gradually drain the available resources. Our bias-handling filters guarantee sub-linear space growth of the index, as hash keys with excessive entries are pruned and no duplicate image entries are allowed in each hash bucket. Figure 3.15 demonstrates the space expansion of the index as the database size grows from 10,000 to 100,000 images. We illustrate the linear case for comparison purposes. It is clear that the index size increases sub-linearly with the size of indexed images.

Time Complexity. Our query system consists of three core steps that determine the performance: a) feature extraction, b) min-hash calculation, c) voting. Among these three steps only voting depends on the number of the database images. During feature extraction we calculate a descriptor for each grid location. The computation time depends on the nature of the description and the size of the patch. Typically, the HOG descriptor computation for an image is carried out in less than 10ms per image [56]. The binarization step can be performed in linear time on the number of vector elements via a selection algorithm [31].

To compute k s-tuples for each binary vector, a linear scan of the vector is required and $k \times s$ hash function evaluations for each non-zero entry. We can speed-up this step by pre-computing the hash function values. Few milliseconds are adequate to extract min-hash values from each image. At voting stage each patch needs to access linearly the bucket entries for all its k min-hash tuples. An additional scan of the nearby patch buckets is required as well. The total voting cost is therefore:

$$\begin{aligned}
&\text{Voting complexity} = \\
&= \mathcal{O}\left(\#patch \times \#neighbors \times k \times \#bucketSize\right) \\
&= \mathcal{O}\left(a \times c \times k \times \log N\right) \\
&= \mathcal{O}\left(\log N\right)
\end{aligned} \tag{3.9}$$

The only value that depends on the image number N is the bucket size. We have empirically shown that the average bucket size contains $logN$ entries. Therefore, if we ignore the constants, the complexity of the voting process is $\mathcal{O}\left(\log N\right)$. Each of the above steps can be readily parallelized as the operations on each patch are independent to each other leading to even lower response times. The impact of voting complexity can be noticed in the mean query times of Table 3.3. The query times remain constant in Flickr15k and EitzSBIR, even though the database size is increased 7 times.

Table 3.3: Time and space statistics of patch hashing over different size datasets. Flickr15k and EitzSBIR have similar mean query times

Dataset	Size	Mean Query Time (s)	Space (MB)
Flickr160	160	0.009	16
Flickr15k	15K	0.2	711
EitzSBIR	100K	0.2	2500

3.6 Conclusions

In this chapter, we have proposed a robust patch based retrieval technique which can scale to large image collections. Shape information in the form of contour orientations is extracted from a patch. The binarization process further enhances strong continuous contours while facilitating the application of min-hash algorithm. Alternative shape representations can be applied in this step, yet further exploration of more appropriate techniques is left as future work. A spatially-aware reverse index created on the unique min-hash values and locations allows for efficient search times and parallelization. State-of-the-art results were demonstrated in three SBIR benchmarks indicating the retrieval quality of our method and its benefits in sketch/image matching against the bag-of-feature approach.

Section 3.2 and the experimental evaluation of patch hashing revealed the discriminant value of spatial information in sketch description. Robust matching goes beyond local appearance similarity and should incorporate holistic structure correspondences as well. Our algorithm identifies these correspondences via the spatial voting process.

Patch hashing is not invariant to affine transformations. In many current sketch matching applications the lack of invariance to affine transformations is not critical. In future though, in order to have a more generic method, e.b. in dealing with cropped or scale images, this limitation should be addressed.

A drawback of the proposed algorithm is the lack of semantics. Several fail cases include visually similar images with irrelevant semantic content. The following chapters will focus on infusing semantics into the search process.

Chapter 4

Discriminant Pairwise Local Embeddings for Sketch Recognition

Contents

4.1 Introduction

Machine understanding of everyday human activities and actions consists a fundamental challenge for computer vision. Sketching to express feelings or elaborate on a topic is a task dating back to prehistoric times, yet it is still contemporary and used in daily activities. Sketch understanding requires little effort from humans. Furthermore, neuroscience studies [59, 127, 107] have shown that humans can decode complex natural scenes from simple line drawings. Evidently, sketching is an efficient and intuitive communication tool between humans. Human-computer interaction could therefore benefit from this expression channel given successful machine interpretation of human sketches. Towards this direction, a large database of 20,000 free-hand drawn sketches [39] initiated a computational study of how humans draw sketches. Computational recognition of line drawings is a challenging task due to the abstract nature of sketching and the inter and intra-class variations between drawings. Moreover, traditional object recognition techniques cannot be directly applied to the sketch domain given its lack of color or texture.

Sketch drawings on several occasions deviate from being realistic depictions of objects or scenes. As a result, algorithms based solely on visual features lack the advantages of semantic information. In the case of SBIR, where sketch symbols are frequently drawn, (e.g.. stick man instead of a human figure), sketch recognition brings a solid contribution towards more accurate and relevant results to a given query. This section proposes a novel supervised dimensionality reduction algorithm to accurately classify freely drawn sketches. We face sketch recognition from a subspace learning classification perspective, hence here we overview briefly previous work on dimensionality reduction.

Dimensionality reduction or *subspace learning* is the transformation that maps data from a high-dimensional space into a meaningful low dimensional space. It has been widely used in recognition tasks to mitigate the inherent drawbacks of high-dimensional spaces. Real-world data like images, videos and speech signals are by nature high-dimensional modalities. In order to efficiently process this data, its dimensionality needs to be reduced. Furthermore, such real world data are accompanied by noise which affects the accuracy of classification algorithms. By exposing the intrinsic dimensionality of the input data, we can generate projection bases that are immune to noise. The benefits of dimensionality reduction include classification, visualization and compression of high-dimensional data [124].

One of the first and classic approaches to dimensionality reduction is the PCA algorithm [63] that generates a subspace where data variance is maximized. PCA is a linear unsupervised technique, therefore does not produce discriminant subspaces. LDA [47] exploits the data labels via the within and

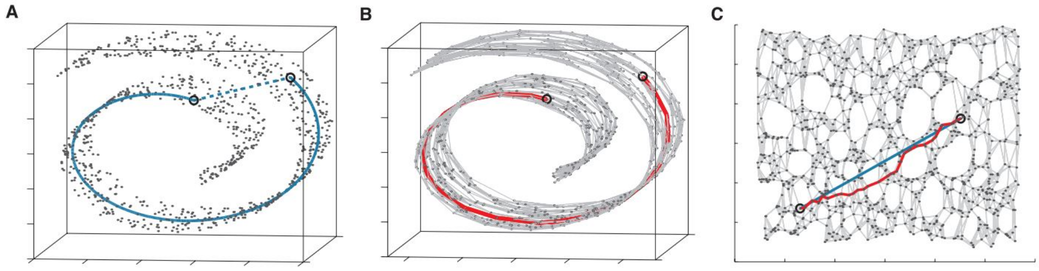


Figure 4.1: Benefits of manifold learning. A) The original Swiss roll structure in three dimensions. The Euclidean distance between two points (dashed lined) differs for the distance in the manifold (solid line). B) Approximation of the manifold distance (red line) via the pairwise distances graph. C) The unfolded Swiss roll. The picture is reproduced from [121].

between-scatter matrices and performs better in classification scenarios. PCA and LDA rely on the assumption that data follow a Gaussian distribution which often do not hold for real world applications. LFDA [117] takes local structure of the data into account so multi-modal data can be embedded appropriately. ICA [57] and probabilistic PCA [123] were also early extensions of PCA.

Manifold learning is the branch of dimensionality reduction that investigates the underlying manifold of data. Originated from ISOMAP [121], manifold learning techniques attempt to discover a low-dimensional manifold where the data lie on. A famous example is the Swiss roll which is originally embedded in a three dimensional space, yet if an 'unfolding' transformation is applied on it, a two dimensional manifold is unveiled. An illustration is presented Figure 4.1. Manifold learning reveals the intrinsic dimensionality of data which in turn improves accuracy on distance-based classifiers.

In the same spirit, Local Preserving Projections (LPP) [52] and its variants [50, 24, 76, 138, 51] generate low dimensional spaces that preserve the local neighborhood of the data, hence the restricting assumptions of PCA and LDA are avoided. LPP is an unsupervised technique, yet extensions have been published that make use of data labels. DLPP [138] incorporates in the optimization process the within and between scatter matrices to achieve class separability. ILPP [50], ARE [76] and max-margin MMP[51] are semi-supervised approaches obtaining label information from user feedback. ILPP updates its learned projection matrix according to user guidelines. MMP solves an eigenvalue problem that maximizes the margin between different labeled samples.

We introduce *Discriminant Pairwise Local Embeddings* (DPLE), a supervised manifold learning algorithm inspired by LPP [52] to facilitate sketch recognition. Nonetheless, our method is generic and can be applied to any other classification problem. The main idea is to learn a discriminant subspace where the data will be better separated than in the original input space, without violating much its local neighborhood. The latter ensures that the data will maintain their manifold structure in the learned subspace, so classification algorithms can generalize better. We form these goals in a convex optimization problem that can be efficiently solved through eigendecomposition. A kernelized version is also introduced to further enhance classification accuracy. Experiments on a large multi-class sketch database demonstrate the advantages of our technique.

DPLE's objective is similar to that of LDE[24]/ARE, yet our formulation is different and the superiority of our technique is attributed to the following factors: a) LDE does not exploit the importance of influential samples, i.e. samples with many close neighbors guaranteed not to be outliers. DPLE utilizes this information in its objective function. b) ARE employs a non-flexible encoding scheme for the relationships between data pairs. It weights equally every pair and does not take into account the distances of samples in the original space. This approach fails to alleviate the influence of noisy data pairs that belong to the same class but they are far away in the feature space. DPLE handles this problem by weighting these relationships with the affinity matrix.

4.2 Locality Preserving Projections Overview

In this section, we overview the definition of *Locality Preserving Projections* (LPP) [52]. LPP is an unsupervised dimensionality reduction technique which generates projection bases that preserve the neighbourhood structure of the data. LPP attempts to discover a low-dimensional manifold where the high dimensional data lie on.

Let n pairs of data samples and associated labels be denoted by$(\mathbf{x}_i, y_i)$, $i = \{1, 2, \ldots, n\}$, where $\mathbf{x}_i \in \mathbf{R}^d$ represents a data sample and $y_i \in \{1, 2, \ldots, |C|\}$ is the label of the i-th sample. $|C|$ is the total number of classes. Let $\boldsymbol{X} \in \mathbf{R}^{d \times n}$ be the matrix of all samples. The i-th column of $\boldsymbol{X}$ is $\mathbf{x}_i$. Let $\mathbf{z}_i \in \mathbf{R}^p(1 \leq p \leq d)$ be an embedded sample and p the dimension of the embedding space. Since we investigate dimensionality reduction scenarios, we usually require $p \ll d$.

Linear dimensionality reduction is performed via the transformation matrix $\boldsymbol{W} \in \mathbf{R}^{d \times p}$:

$$\mathbf{z}_i = \boldsymbol{W}^\top \mathbf{x}_i \tag{4.1}$$

In Section 4.3.2, we discuss non-linear dimensionality reduction scenarios, but for now we will focus on the linear case.

The structure information of the data set is represented in the *affinity matrix* $\boldsymbol{A}$. The matrix $\boldsymbol{A}$ captures similarities between data pairs and is defined as:

$$\boldsymbol{A}_{i,j} = \begin{cases} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2}, & \text{if } \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \\ & \text{or } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

where $\mathcal{N}_k(\mathbf{x})$ represents the set of k-nearest neighbors of $\mathbf{x}$. A simpler alternative to (4.2) is to set $\boldsymbol{A}_{i,j} = 1$ if $\mathbf{x}_i$ is a nearest neighbor of $\mathbf{x}_j$ or vice-versa; otherwise $\boldsymbol{A}_{i,j} = 0$. In both cases, a high value of $\boldsymbol{A}_{i,j}$ indicates that $\mathbf{x}_i$ and $\mathbf{x}_j$ lie close in the defined metric space and a low value that they lie apart.

Using $\boldsymbol{A}$ the desired projection matrix $\boldsymbol{W}_{LPP} \in \mathbf{R}^{d \times p}$ is acquired by the following optimization problem:

$$\begin{aligned} &\boldsymbol{W}_{LPP} = \arg\min_{\boldsymbol{W}} \frac{1}{2} \sum_{i,j}^{n} \|\boldsymbol{W}^\top \mathbf{x}_i - \boldsymbol{W}^\top \mathbf{x}_j\|^2 \boldsymbol{A}_{i,j} \\ &\text{subject to: } \boldsymbol{W}^\top \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^\top \boldsymbol{W} = \boldsymbol{I} \end{aligned} \tag{4.3}$$

where $\boldsymbol{D}_{i,i} = \sum_{j=1}^{n} \boldsymbol{A}_{i,j}$ is a diagonal matrix and $\boldsymbol{I}$ the identity matrix.

The above optimization problem attempts to map data pairs close in the embedding space if they lie close in the original feature space. The constrain $\boldsymbol{W}^\top \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^\top \boldsymbol{W} = \boldsymbol{I}$ is to avoid the trivial solution $\boldsymbol{W} = \mathbf{0}$.

Using linear algebra the minimization function (4.3) can be rewritten as:

$$\begin{aligned}
&\sum_{i,j}^{n} \|\boldsymbol{W}^\top \mathbf{x}_i - \boldsymbol{W}^\top \mathbf{x}_j\|^2 \boldsymbol{A}_{i,j} \\
&= \sum_{i}^{n} \boldsymbol{W}^\top \mathbf{x}_i \boldsymbol{D}_{ii} \mathbf{x}_i^\top \boldsymbol{W} - \sum_{i,j}^{n} \boldsymbol{W}^\top \mathbf{x}_i \boldsymbol{A}_{i,j} \mathbf{x}_j^\top \boldsymbol{W} \\
&= \boldsymbol{W}^\top \boldsymbol{X} \left(\boldsymbol{D} - \boldsymbol{A}\right) \boldsymbol{X}^\top \boldsymbol{W} \\
&= \boldsymbol{W}^\top \boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^\top \boldsymbol{W}
\end{aligned} \tag{4.4}$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ is the Laplacian matrix.

Therefore, LPP can be formulated as:

$$\begin{aligned}
&\boldsymbol{W}_{LPP} = \underset{\boldsymbol{W}}{\arg\min}\, \boldsymbol{W}^\top \boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^\top \boldsymbol{W} \\
&\text{subject to:}\, \boldsymbol{W}^\top \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^\top \boldsymbol{W} = \boldsymbol{I}
\end{aligned} \tag{4.5}$$

If we apply the Lagrange multipliers to (4.5) and set the derivative with respect to $\boldsymbol{W}$ to zero we end up with the following formula:

$$\boldsymbol{X} \boldsymbol{L} \boldsymbol{X}^\top \mathbf{w} = \lambda \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^\top \mathbf{w} \tag{4.6}$$

It follows from (4.6) that the optimal $\boldsymbol{W}_{LPP}$ is composed of the generalized eigenvectors corresponding to the p smallest eigenvalues of the generalized eigenvalue problem of (4.6).

LPP does not make use of sample labels, as it is an unsupervised algorithm. In the following section, we introduce a novel technique that handles efficiently labeled data and preserves the local structure.

4.3 Discriminant Pairwise Local Embeddings

This section describes *Discriminant Pairwise Local Embeddings* (DPLE), a novel dimensionality reduction technique and its kernelized variant. As shown in Section 4.2, LPP is an unsupervised manifold learning technique. In many recognition tasks though, unsupervised algorithms cannot perform comparably to their supervised counterparts due to the lack of semantic information provided by the training samples. We show that LPP can be modified to take advantage of the labeled samples while preserving the local data neighborhoods. Moreover, our algorithm can be extended to non-linear spaces utilizing the *kernel trick* [108].

4.3.1 Linear DPLE

Following the notation of Section 4.2, we assume each data sample $\mathbf{x}_i$ is associated with a label y_i. Based on the label information, we define two pairwise relation matrices. The *same-class matrix* $\boldsymbol{A}^{(s)}$ representing all the sample pairs that share the same label and the *different-class matrix* $\boldsymbol{A}^{(d)}$ representing all the sample pairs with different labels:

$$\boldsymbol{A}^{(s)}_{i,j} = \begin{cases} \boldsymbol{A}_{i,j}, & \text{if } y_i = y_j \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

$$\boldsymbol{A}^{(d)}_{i,j} = \begin{cases} \boldsymbol{A}_{i,j}, & \text{if } y_i \neq y_j \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

We observe from (4.7) and (4.8) that matrices $\boldsymbol{A}^{(s)}$ and $\boldsymbol{A}^{(d)}$ are weighted with the affinity matrix $\boldsymbol{A}$. If we assign a constant value to similar and dissimilar pairs as in [76]; for instance if $\boldsymbol{A}^{(s)}_{i,j} = 1$ when $y_i = y_j$ and $\boldsymbol{A}^{(d)}_{i,j} = 1$ when $y_i \neq y_j$, then all the sample pairs will have equal weights and as a result structure information will be lost. Instead, by employing the affinity matrix we assign an 'importance' value to each pair. Samples that lie close in the original input space are more significant and are forced to lie close in the embedding space. On the other hand, pairs that are apart in the original space are either ignored or contribute only slightly to the optimal solution. This idea is similar to the local variant of LDA [117], yet employed in a different learning framework.

We suggest the following optimization problem:

$$\begin{aligned} &\arg\max_{\boldsymbol{W}} J(\boldsymbol{W}) = \frac{1}{2}\sum_{i,j}^{n} \|\boldsymbol{W}^\top \mathbf{x}_i - \boldsymbol{W}^\top \mathbf{x}_j\|^2 \left(\boldsymbol{A}^{(d)}_{i,j} - \gamma \boldsymbol{A}^{(s)}_{i,j}\right) \\ &\text{subject to: } \boldsymbol{W}^\top \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^\top \boldsymbol{W} = \boldsymbol{I} \end{aligned} \tag{4.9}$$

where $\boldsymbol{D}_{i,i} = \sum_{j=1}^{n} \boldsymbol{A}_{i,j}$ is a diagonal matrix consisted of the row sums of

$\boldsymbol{A}$ and γ is a scalar to compensate for any imbalances occurred by different number of pair samples between $\boldsymbol{A}^{(d)}$ and $\boldsymbol{A}^{(s)}$. A recommended value to assign to γ is the ratio between negative and positive pairs.

The above formulation minimizes the Euclidean distances between all sample pairs that belong to the same category through matrix $\boldsymbol{A}^{(s)}$ and at the same time maximizes those between pairs belonging to different classes through matrix $\boldsymbol{A}^{(d)}$. We have previously seen that each pair relationship is weighted by the affinity matrix $\boldsymbol{A}$, therefore the intrinsic structure of data is maintained. The constrain $\boldsymbol{W}^\top \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^\top \boldsymbol{W} = \boldsymbol{I}$ is imposed to avoid the trivial solution $\boldsymbol{W} = \boldsymbol{0}$ and each entry $\boldsymbol{D}_{i,i}$ provides a measure of importance to the embedded sample $\mathbf{z}_i = \boldsymbol{W}^\top \mathbf{x}_i$.

The objective function in (4.9) can be rewritten as follows using linear algebra properties:

$$\begin{aligned}
J(\boldsymbol{W}) &= \frac{1}{2}\sum_{i,j}^{n} \|\boldsymbol{W}^\top \mathbf{x}_i - \boldsymbol{W}^\top \mathbf{x}_j\|^2 \left(\boldsymbol{A}^{(d)}_{i,j} - \gamma \boldsymbol{A}^{(s)}_{i,j}\right) \\
&= \sum_{i}^{n} \boldsymbol{W}^\top \mathbf{x}_i \boldsymbol{D}^{(d)}_{ii} \mathbf{x}_i^\top \boldsymbol{W} - \sum_{i,j}^{n} \boldsymbol{W}^\top \mathbf{x}_i \boldsymbol{A}^{(d)}_{i,j} \mathbf{x}_j^\top \boldsymbol{W} \\
&\quad - \gamma \left(\sum_{i}^{n} \boldsymbol{W}^\top \mathbf{x}_i \boldsymbol{D}^{(s)}_{ii} \mathbf{x}_i^\top \boldsymbol{W} - \sum_{i,j}^{n} \boldsymbol{W}^\top \mathbf{x}_i \boldsymbol{A}^{(s)}_{i,j} \mathbf{x}_j^\top \boldsymbol{W} \right) \\
&= \boldsymbol{W}^\top \boldsymbol{X} \left(\boldsymbol{D}^{(d)} - \boldsymbol{A}^{(d)}\right) \boldsymbol{X}^\top \boldsymbol{W} - \gamma \left(\boldsymbol{W}^\top \boldsymbol{X} \left(\boldsymbol{D}^{(s)} - \boldsymbol{A}^{(s)}\right) \boldsymbol{X}^\top \boldsymbol{W} \right) \\
&= \boldsymbol{W}^\top \boldsymbol{X} \left(\boldsymbol{L}^{(d)} - \gamma \boldsymbol{L}^{(s)}\right) \boldsymbol{X}^\top \boldsymbol{W}
\end{aligned} \tag{4.10}$$

where $\boldsymbol{L}^{(s)} = \boldsymbol{D}^{(s)} - \boldsymbol{A}^{(s)}$ and $\boldsymbol{L}^{(d)} = \boldsymbol{D}^{(d)} - \boldsymbol{A}^{(d)}$ are the Laplacian matrices of $\boldsymbol{A}^{(s)}$ and $\boldsymbol{A}^{(d)}$ respectively.

We can now reformulate our optimization problem:

$$\begin{aligned}
&\arg\max_{\boldsymbol{W}} J(\boldsymbol{W}) = \boldsymbol{W}^\top \boldsymbol{X} \left(\boldsymbol{L}^{(d)} - \gamma \boldsymbol{L}^{(s)}\right) \boldsymbol{X}^\top \boldsymbol{W} \\
&\text{subject to: } \boldsymbol{W}^\top \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^\top \boldsymbol{W} = \boldsymbol{I}
\end{aligned} \tag{4.11}$$

Similar to (4.5), we apply the Lagrange multipliers to the above problem and the set the derivative with respect to $\boldsymbol{W}$ to zero.

$$\boldsymbol{X}\left[\boldsymbol{L}^{(d)} - \gamma \boldsymbol{L}^{(s)}\right] \boldsymbol{X}^\top \bar{\mathbf{w}} = \bar{\lambda} \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^\top \bar{\mathbf{w}} \tag{4.12}$$

The result is a generalized eigenvalue problem and since $\boldsymbol{L}^{(s)}$, $\boldsymbol{L}^{(d)}$ and $\boldsymbol{D}$ are symmetric semi-definite matrices all the eigenvalues are real positive numbers.

The optimal projection matrix $\boldsymbol{W}_{DPLE}$ is given by:

$$\boldsymbol{W}_{DPLE} = \left(\sqrt{\bar{\lambda}_1}\bar{\mathbf{w}}_1 \mid \sqrt{\bar{\lambda}_2}\bar{\mathbf{w}}_2 \mid \cdots \mid \sqrt{\bar{\lambda}_p}\bar{\mathbf{w}}_p \right) \tag{4.13}$$

where $\{\bar{\mathbf{w}}\}_{i=1}^{p}$ are the generalized eigenvectors associated with the p largest eigenvalues $\bar{\lambda}_1 \geq \bar{\lambda}_2 \geq \cdots \geq \bar{\lambda}_p$ of (4.12).

The steps of DPLE are summarized in the following algorithm:

Algorithm 1: DPLE embedding

```
Data: $(\mathbf{x}_i, y_i)$ $i \in \{1, 2, \ldots, n\}, \gamma, p$
Result: Projection matrix: $\boldsymbol{W}_{DPLE}$
Compute affinity matrix $\boldsymbol{A}$ according to (4.2).
Compute matrices $\boldsymbol{A}^{(s)}$ and $\boldsymbol{A}^{(d)}$ from (4.7) and (4.8).
Solve the generalized eigenproblem of (4.12).
Form the columns of $\boldsymbol{W}_{DPLE}$ from the eigenvectors of (4.12)
corresponding to the largest eigenvalues.
```

DPLE exploits the labeled information encoded in the matrices $\boldsymbol{A}^{(s)}$ and $\boldsymbol{A}^{(d)}$ to generate discriminate projection bases without violating the intrinsic structure of the data. The latter is ensured by the leverage of the affinity matrix $\boldsymbol{A}$ which weights accordingly each sample pair. The embedded data lie on a discriminative semantic manifold which preserves local geometric relations. As a result classes become better separated in the learned subspace.

4.3.2 Kernelized DPLE

In most real world applications, data in the original input space cannot be linearly separated, because it is generated by non-linear processes. In such cases, linear algorithms like DPLE fail to produce efficient embedding spaces. We show that by using the *kernel trick* [108], we can generate a non-linear map from the original high-dimensional feature space to a lower-dimensional manifold where non-linear data can be efficiently represented.

Let $\phi : \mathbf{R}^d \rightarrow \mathcal{H}$ be a non-linear map function, mapping the Euclidean space $\mathbf{R}^d$ to Hilbert space $\mathcal{H}$. The Hilbert space is a vector space $\boldsymbol{H}$ with

Table 4.1: Popular kernel functions

Kernel $k(\mathbf{x}, \mathbf{y})$	**Formula**
Linear	$\mathbf{x}^T \mathbf{y}$
Polynomial	$(\mathbf{x}^T \mathbf{y} + c)^d$
Gaussian	$\exp\left(-\|\mathbf{x} - \mathbf{y}\|^2 / 2\sigma^2\right)$

an inner product $\langle f, g \rangle$ such that the norm defined by $|f| = \sqrt{\langle f, f \rangle}$ turns $\boldsymbol{H}$ into a complete metric space. In Hilbert space the eigenvector problem of (4.12) becomes:

$$\phi\left(\boldsymbol{X}\right)\left[\boldsymbol{L}^{(d)} - \gamma \boldsymbol{L}^{(s)}\right]\phi\left(\boldsymbol{X}\right)^{\top}\bar{\mathbf{w}} = \bar{\lambda}\phi\left(\boldsymbol{X}\right)\boldsymbol{D}\phi\left(\boldsymbol{X}\right)^{\top}\bar{\mathbf{w}} \tag{4.14}$$

There is no easy way to directly compute the mapping $\phi\left(\boldsymbol{X}\right)$, yet we can employ inner products of mapped data to solve the problem. We define the inner products of the mapped data as:

$$\mathbf{K}_{ij}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \phi\left(\mathbf{x}_i\right)^{\top}\phi\left(\mathbf{x}_j\right) \tag{4.15}$$

$\mathbf{K}$ is a kernel matrix. Some popular kernel functions can be found in Table 4.1.

The eigenvectors of (4.14) are linear combinations of $\phi\left(\mathbf{x}_1\right), \phi\left(\mathbf{x}_2\right), \ldots, \phi\left(\mathbf{x}_n\right)$, hence we can write:

$$\bar{\mathbf{w}} = \sum_{i=1}^{n} \alpha_i \phi\left(\mathbf{x}_i\right) = \phi\left(\boldsymbol{X}\right)\alpha \tag{4.16}$$

where $\alpha = \left[\alpha_1, \alpha_1, \ldots, \alpha_n\right]^{\top} \in \mathbf{R}^n$. Using (4.16) it is easy to obtain the *kernelized* eigenvalue problem:

$$\boldsymbol{K}\left[\boldsymbol{L}^{(d)} - \gamma \boldsymbol{L}^{(s)}\right]\boldsymbol{K}\alpha = \bar{\lambda}\boldsymbol{K}\boldsymbol{D}\boldsymbol{K}\alpha \tag{4.17}$$

As before, the optimal embedding is obtained from the p eigenvectors corresponding to largest eigenvalues. The embedding of a new sample onto the eigenvector $\mathbf{w}^k$ is achieved by:

$$\mathbf{z}^k = \left(\mathbf{w}^k\right)^{\top}\mathbf{x} = \sum_{i=1}^{n} \alpha_i^k \boldsymbol{K}\left(\mathbf{x}, \mathbf{x}_i\right) \tag{4.18}$$

The KDPLE algorithm is slightly modified from its linear version:

Algorithm 2: KDPLE embedding

```
Data: $(\mathbf{x}_i, y_i)\ i \in \{1, 2, \ldots, n\}, \gamma, p$, kernel function $k(\cdot, \cdot)$
Result: Projection matrix: $\boldsymbol{W}_{KDPLE}$
Compute affinity matrix $\boldsymbol{A}$ according to (4.2).
Compute the kernel matrix $\boldsymbol{K}$ for all samples as in (4.15).
Compute matrices $\boldsymbol{A}^{(s)}$ and $\boldsymbol{A}^{(d)}$ from (4.7) and (4.8).
Solve the generalized eigenproblem of (4.17).
Form the columns of $\boldsymbol{W}_{KDPLE}$ from the eigenvectors of (4.17)
  corresponding to the largest eigenvalues.
```

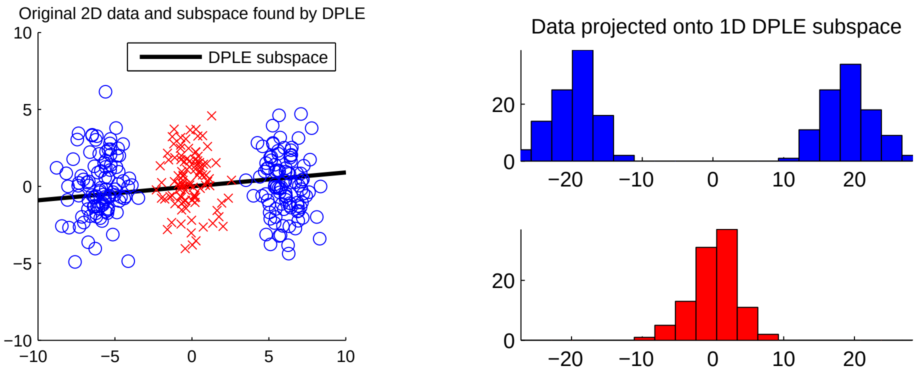
4.4 Experiments

We evaluate the learning generalization capabilities of our algorithm. First, some toy data are employed to demonstrate the discriminant features of DPLE. Subsequently, we perform extensive evaluations against competitive manifold learning algorithms in face and sketch recognition domains. DPLE and its kernelized version is superior than its alternatives and performs comparable to SVM in sketch recognition.

4.4.1 Toy Examples

Artificial two-dimensional data from three independent Gaussian distributions are generated to test the discriminant property of DPLE's projections. Figures 4.2, 4.3 , 4.4 visualize the results. The data consists of two classes indicated by red crosses and blue circles that are not linearly separated. We apply the DPLE embedding and generate a discriminant subspace of 1D. Figure 4.2a shows the computed subspace along with the projections of the original data. DPLE optimization outputs a meaningful linear projection that results to maximization of data variance and structure preservation.

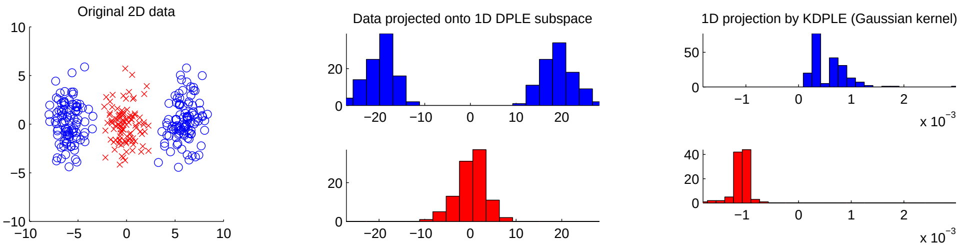
Subsequently, we test KDPLE's generalization ability on the same dataset. We again generate an 1D subspace for the linear and Gaussian kernels. As expected, the linear kernel fails to dichotomize the non-linear data, as opposed



(a) Original data in 2 dimensions.

(b) DPLE 1D projection.

Figure 4.2: DPLE example from 2D to 1D. There are two data classes indicated by red and blue.

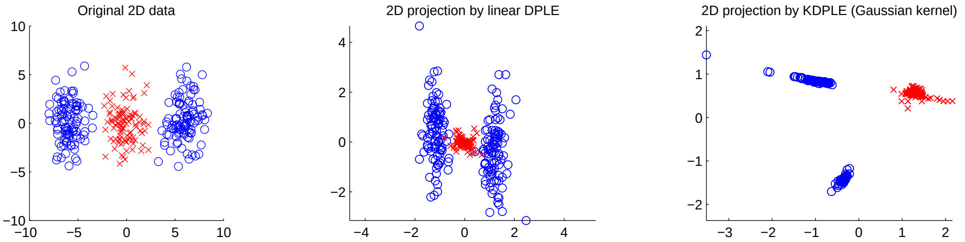


(a) Original data.

(b) DPLE 1D.

(c) KDPLE 1D.

Figure 4.3: DPLE example from 2D to 1D. There are two data classes indicated by red and blue. Data can be linearly separated after the KDPLE projection. The same does not hold for the linear DPLE subspace.



(a) Original data.

(b) DPLE 2D.

(c) KDPLE 2D.

Figure 4.4: Demonstration of DPLE properties in 2D. DPLE cannot linearly separate data but it reduces the variance within the classes. KDPLE generates a discriminant projection where data are linearly separated and within variance is greatly reduced.

to the Gaussian kernel which discovers a subspace where data are optimally separated. See the illustrations of Figure 4.3.

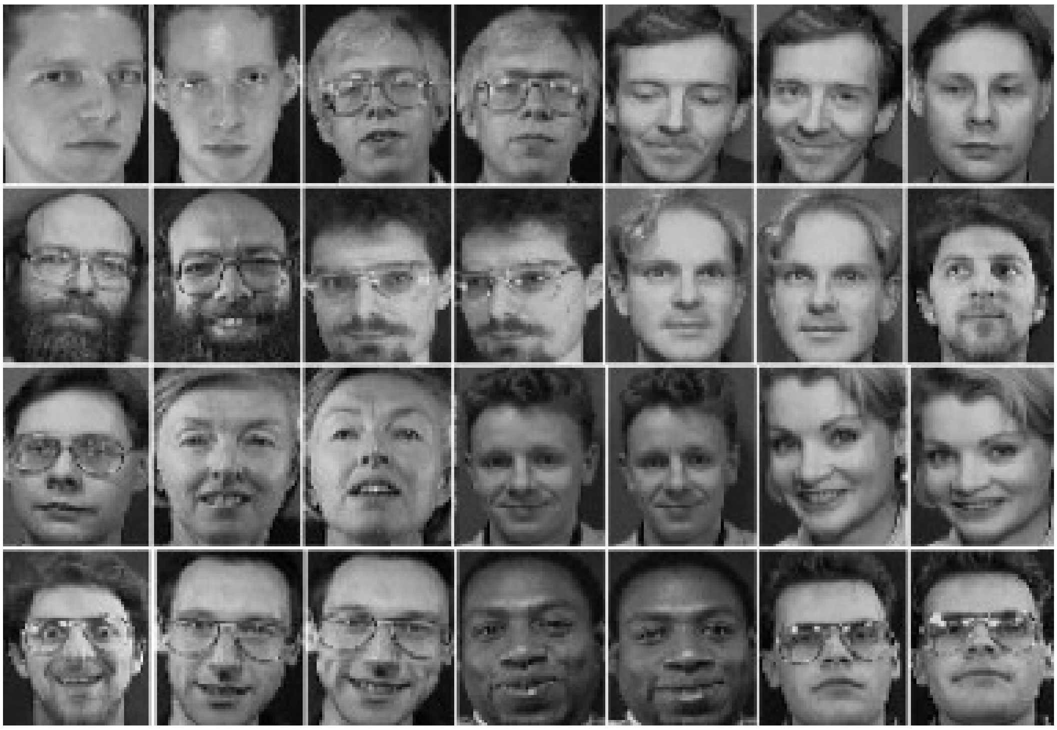


Figure 4.5: Preview of the ORL faces.

Furthermore, we generate 2D subspaces for DPLE and KDPLE respectively and compare them against the original input space. The results are plotted in Figure 4.4. We notice that the DPLE projections maintain the data structure but drastically reduce the variance within the classes, creating compact clusters. A possible application of such feature could be in data clustering where DPLE can act as a preprocessing operator. The learned KDPLE subspace, Figure 4.4c, demonstrates the optimal performance. Data are linearly separated and, as expected by the objective function's property, within-class variance is minimized while variance between different classes is maximized. The outcome is three highly compact clusters. We also note that neighbor relations are propagated in the projection. There are two separate blue clusters as in the original space.

4.4.2 Face Recognition

Next, we evaluate DPLE with face recognition. We employ the well-known ORL collection of faces [105] as the comparison dataset. ORL includes 40

Method	**ORL**
Linear	
NN	97.5% ($p = 136$)
PCA	98% ($p = 32$)
LDA	98% ($p = 24$)
LPP	96.25% ($p = 20$)
LFDA	98.5% ($p = 12$)
LDE	98.5% ($p = 21$)
DPLE	**99**% ($p = 23$)
Kernelized	
KLFDA	99% ($p = 24$)
KLDE	99.25% ($p = 21$)
KDPLE	**99.25**% ($p = 23$)

Table 4.2: Best recognition rates in the ORL face dataset. Parameter p represents the number of subspace dimensions.

subjects with 10 grayscale images per subject. Samples in a class exhibit slight variation in face expression and angle. The size of each image is 92x112 pixels, with 256 gray levels per pixel. Following the preprocessing of [24], we resize each image to 23×28 pixels and vectorize the outcome. The original dimensionality of the data is 644. We apply PCA to the image vectors and keep 98% of the information. The final features lie in an 136-dimensional space.

We compare our method with the KNN classifier in the original feature space denoted as (NN), the classic PCA and LDA algorithms and a collection of more sophisticated manifold learning techniques, namely LPP [52], LFDA [117] and LDE [24] along with kernelized versions for the last two. The KNN rule is used in all methods for classification and the best recognition accuracy over K values $\{1, 3, 5, 7, 9\}$ is reported. The optimal parameters of each algorithm are tuned empirically. In the kernelized version of the algorithms, we employ the RBF kernel with $\sigma = 1$. We perform 5-fold cross validation on all reported results.

We deduct by the performance of the KNN in the original data that the ORL classes are easily separated by the selected features, see Table 4.2. KNN

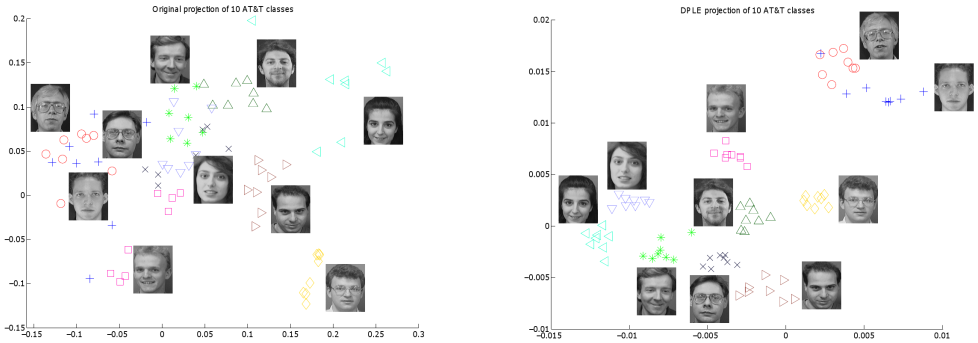


Figure 4.6: Left) 2D projection of the original feature vectors from ORL dataset. Only 10 of the 40 classes are displayed for visualization purposes. Right) 2D DPLE projection of the same data. Classes are more compact and clearly partitioned.

recognition rate is 97.5%. The discriminant manifold learning algorithms perform better than PCA, LDA and the unsupervised LPP. KDPLE achieves the highest recognition rates in this dataset.

The benefits of a DPLE embedding against the original space are illustrated in Figure 4.6. We perform a Multi-Dimensional Scaling (MDS) [12] embedding of the original and DPLE projected data in a 2D space for visualization purposes. MDS is designed to preserve the data distances in the new subspace. In the original feature space, several classes overlap and as a result KNN classification rate drops. Differently, in the DPLE subspace each class is distinctly separated from the rest and forms a solid cluster. Under this conditions KNN accuracy is boosted.

The ORL dataset is saturated and the resulting deductions, despite offering a general picture of each algorithm's performance, are marginal. In the next section, a more challenging domain will be tested where the pros and cons of each algorithm will clearly emerge.

4.4.3 Sketch Recognition

We employ the EitzSKETCH [39] dataset for the sketch recognition evaluation. The dataset encompasses 20,000 unique sketches evenly distributed over 250 object categories. Each image depicts a binary sketch of a single object. All sketches are rescaled to a fixed size and centered in the image canvas to accommodate scale and translation invariance. The human recognition accuracy on the above database is 73.1% which highlights the challenge for machine classification. We observe *high inter-class* and *intra-class variability*. Some classes are easily recognized while others regularly misclassified to categories with similar visual appearance. Moreover, an object can be sketched quite differently by various individuals a fact that contributes to the aforementioned intra-class variations.

We follow the feature representation of [39]. Each sketch is represented by an ensemble of local features on a 27 × 27 overlapping grid that capture the main gradient orientations of a sketch region. Each feature vector has 64 dimensions. The local descriptors are then utilized to generate a bag-of-features representation of the sketch. The resulting descriptor is a 500-dimensional vector. The features are publicly available in the following link[1] and in our experiments we use them as provided with no modification.

We compare DPLE against competitive subspace learning methods as defined in Section 4.4.2. The comparison methods are KNN (NN), PCA, LDA, LFDA and LDE along with the kernelized versions of the last two. The KNN classification rule is employed over all evaluations to compute the recognition accuracy, with $K = \{1, 2, 3, 4, 5\}$ and Euclidean (l_2) and Manhattan (l_1) distances. The reported results are obtained with optimal parameters for each method over 3-fold cross-validation.

Table 4.3 summarizes our comparative evaluation in sketch recognition. The recognition rate of KNN classification in the original feature space is 45%, achieved with $K = 4$ and l_1 distance. We observe that PCA and LDA output much lower rates than NN, highlighting the limitation induced by the Gaussian distribution assumption of these methods. LPP is also failing to meet NN accuracy, because of its unsupervised nature. LPP solely focuses on data structure preservation, hence generates non-discriminant projection bases.

The linear version of the discriminant manifold algorithms, namely LFDA, LDE and DPLE, outperform the NN accuracy. All three methods accomplish similar recognition rates. That occurs because the data cannot be efficiently separated by linear projections; therefore the recognition accuracy of linear

[1] http://cybertron.cg.tu-berlin.de/eitz/projects/classifysketch/

Table 4.3: Evaluation of dimensionality reduction techniques in EitzSKETCH dataset. KNN classification with 3-fold cross-validation is used and the best accuracy over the parameter space is reported. Unless noted otherwise, $K = 5$ and distance is l_2.

Method	$p = 100$	$p = 150$	$p = 200$	$p = 250$	$p = 300$
Linear					
NN	45% (l_1)				
PCA	39.75% (l_1)	41.10% (l_1)	41.58% (l_1)	41.97% (l_1)	41.79% (l_1)
LDA	41.20%	40.06%	38.49%	36.89%	N/A
LPP	39.66%	41.02%	41.51%	41.67%	41.74%
LFDA	47.11%	46.72%	46.02%	44.95%	43.80%
LDE	46.48%	46.12%	44.89%	43.50%	42.22%
DPLE	46.43%	46.55%	46.41%	46.36%	46.39%
Kernelized					
KLFDA	47.86%	47.70%	47.29%	46.55%	46.28%
KLDE	49.46%	50.78%	51.38%	50.89%	50.23%
KDPLE	**49.52%**	**51.14%**	**52.03%**	**52.33%**	**51.19%**

methods has an upper limit which all three methods reach. We will show next that the kernelized versions can overcome this barrier. Another interesting observation is that DPLE achieves stable recognition rates across the dimension spectrum which never drops below 46%, as opposed to LFDA and LDE.

KDPLE's recognition rates are higher than those of the rest evaluation methods. An illustration of the recognition rate of the kernelized methods is presented in Figure 4.7. The noticeable increase in accuracy emphasizes the classification boost induced by the kernelized variations. Data are mapped in a higher dimensional space via the kernel trick where the learned embedding can effectively separate them. That does not happen for KLFDA though, we observe just a slight increase in the accuracy between its linear and non-linear variation. Furthermore, Figure 4.7 shows that KLDFA cannot cope with a challenging dataset as well as KLDE or KDPLE. The latter two methods set similar criteria to generate discriminant projections, but the optimization formulation of DPLE leads to enhanced performance. Specifically, the constrain of (4.9) implicitly weights the significance of each train sample.

We further study the impact of parameters K and p in the sketch recog-

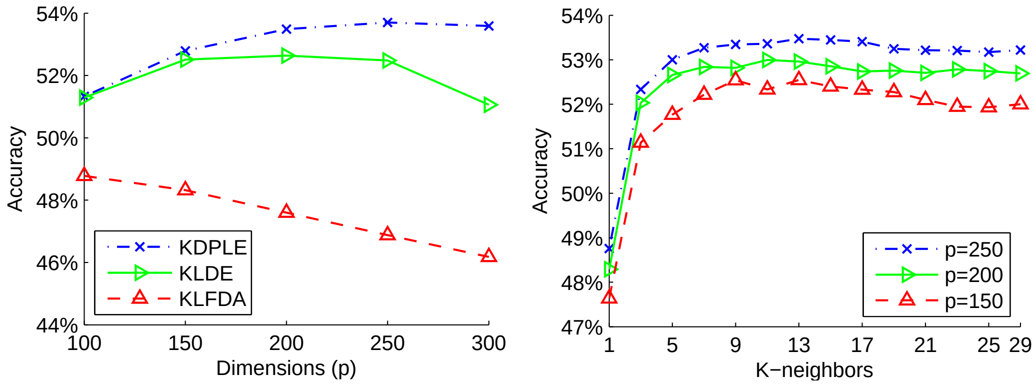


Figure 4.7: Left) Sketch recognition accuracy of kernelized algorithms across varying dimensionality using KNN classification. KDPLE outperforms the rest methods. Right) Sketch recognition accuracy of KDPLE under varying dimensionality p and number of nearest neighbors K using KNN classification.

nition accuracy of KDPLE. Figure 4.7 depicts the results. Classification accuracy sharply rises up to $K = 10$. Additional increase on the number of neighbors does not affect further the recognition rate. The best sketch recognition rate reached by KDPLE is 53.70%, achieved in a learned subspace of $p = 260$ by a KNN classifier with $K = 12$.

The SVM classification accuracy on this dataset is 56% [39]. DPLE achieves comparable recognition rates to SVMs by using the simple KNN classifier in the learned subspace. KNN is by design a multi-class classifier and offers much faster classification times than SVMs, which are binary classifiers. Furthermore, the dimensionality of data has been reduced to half. That leads to significant economy on storage space and allows for scalability.

4.5 Conclusions

In this chapter, we introduced DPLE a supervised manifold learning algorithm that generates discriminant embeddings with a convex optimization process based on pairwise relations between the data. A non-linear variant of the algorithm based on the kernel trick is also proposed. DPLE generates subspaces where the neighbor relations are preserved and classes are more compact. The convex optimization can be reliably and quickly solved via eigendecomposition. Evaluations in face and hand drawn sketch datasets against several competitive manifold learning algorithms proved the superiority of DPLE. In the sketch recognition domain, DPLE achieves comparable performance to the well known SVM classifier and compresses by half the space dimensionality. As a generic dimensionality reduction algorithm DPLE have broad applicability in several domains.

DPLE cannot modify the learned subspace upon new sample arrival. Instead a new optimization problem should be solved with all the available data. Online updating of the projection matrix with a limited amount of new samples is an interesting path to explore. Especially, in sketch recognition after a new query has been drawn, it could be used to expand the learned space with more information. Moreover, there are cases where the limited amount of training samples does not allow for reliable approximation of the underlying data manifold. In the literature, there are suggestions on how to overcome this limitation especially designed for the image domain [134, 136]. These methods work with two-dimensional matrices, instead of vectors, and define tensor products between them. Such an approach suit better the two-dimensional nature of the image domain and take into account the locality of pixels. Experimental evidence in face recognition have shown improvements over the traditional vector-based algorithms. DPLE might benefit by similar modifications. We leave the investigation of these issues as future work.

Chapter 5

Horizontal Flip-Invariant Sketch Matching via Local Structured Similarities

Contents

5.1 Introduction

In this chapter, we develop a methodology to efficiently match a line drawing to a database of sketches invariant to horizontal flip, i.e. mirror reflection across the vertical axis. The benefits of sketch recognition have been analyzed in previous chapters, therefore here we will discuss the advantages of a scalable

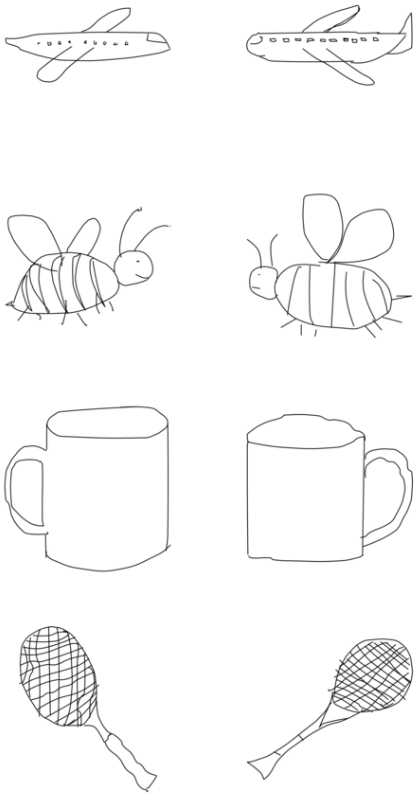


Figure 5.1: Sketch pairs from [39] exhibiting reflection symmetry across the vertical axis.

flip-invariant matching approach. Symmetry is a salient visual property. The Gestalt principle of symmetry, introduced in Chapter 2, states that the human mind perceives objects as being symmetrical and forming around a center point. Symmetry can be defined across any axis orientation. Evidence from several studies [45, 126, 130] support the claim that horizontal reflection symmetry is the most prominent. Visual inspection on the available sketches of EitzSKETCH dataset verifies that the same trend is present in sketch drawing. Figure 5.1 displays sketch pairs from the same category that exhibit horizontal reflection symmetry. All the above contributed to the decision to focus on horizontal symmetry and discard other possible orientations.

DPLE produced promising recognition results. Still, there is space for improvements. Towards this direction, a sketch matching algorithm can act as a post recognition filter to re-rank results and provide insight on ambiguous cases. A training stage is no longer required. Recent research [39, 74, 83] identified shape and structure as key properties for robust sketch matching. Our study in SBIR produced an efficient methodology to exploit these particular properties. In this chapter, we extend our patch hashing framework to facilitate sketch matching and recognition while accounting for flip-invariance. The modified algorithm generates a matching score between two sketches by counting their local region correspondences. We establish region correspondences based on similar patches in terms of shape, that are located in nearby positions. The feature description and index mechanism of Chapter 3 is adopted. Obviously, the same limitation of affine invariance is propagated as well. We employ the min-hash algorithm to estimate local shape similarities. Each new input sketch generates a ranking on the indexed sketches. The generated ranking can be exploited for robust sketch recognition. Furthermore, we propose a customized version of the patch hashing algorithm invariant to reflection symmetry across the vertical axis and we show that it can drastically improve recognition performance. We perform extensive experiments with two challenging sketch datasets with various appearance features and demonstrate state-of-the-art results in low computational time. Matching algorithms generally require tedious computations to produce robust results and suffer from scalability issues, as exhaustive one-to-one comparisons with all available exemplars must be performed. Our method can be seen as a sketch-to-sketch retrieval framework. As opposed to traditionally slow matching techniques, our algorithm can scale well and can generate a ranking on 20,000 sketches in a fraction of a second.

5.2 Related Work

Sketch-to-sketch matching is an open issue for researchers and has been studied since the early days of computers evolution [94]. Early approaches focused on sketch domains of structured nature, like diagram recognition [109, 49, 7]. These approaches extract simplistic stroke features and cannot cope with the complexity of freely drawn sketches. Deformable template matching approaches [37, 84] have also been studied in literature but face scalability issues due to their computational complexity.

Freely drawn sketches are often collections of rough strokes without any well defined form. For this reason, shape descriptors defined for closed-curve sketches, like the inner-distance [77], fail to capture the characteristics of such drawings. Advances in sketch based image retrieval [42, 56] identified histograms of oriented gradients as a pertinent feature for the sketch domain. Moreover, the sparse nature of line drawings dictates the adoption of large patches for more elaborate descriptions. Supervised learning methods like DPLE used a bag-of-features (BoF) [42] representation of these features for free hand sketch classification and showed promising results. BoF have been successful in generic object recognition [114, 9]. One of its drawbacks is the lack of spatial information in vector encoding. Solutions to this have been proposed in the form a spatial pyramid [71] or a spatial codebook [17].

Lately, attention has been given to structured features approaches. In [74], a star graph model is employed to establish appearance and structure similarities between features. Local HOG features are calculated for each sketch and a weighted combination of appearance and location cues arbitrates on matching quality. The star-graph model with a supervised category filtering achieves state-of-the-art accuracy in the EitzSKETCH dataset with 61.5% accuracy. This approach is computationally expensive as several distance evaluations are carried out for each matched pair. In [83], a sketch-to-sketch retrieval algorithm is suggested using dense feature sampling and hierarchical codebook to encode structure information.

5.3 Horizontal Flip-Invariant Matching

5.3.1 Matching and Recognition

The matching algorithm follows the same strategy as in Chapter 3. A brief summary of the process is as follows: each sketch is represented by several overlapping patches. A spatially-aware index is built upon the local extracted features and the score between two sketches is computed with (3.8). For presentation purposes we redefine (3.8) here. The reader can refer to Chapter 3 for a detailed explanation of how to extract features and construct the hash table. Our redefinition of (3.8) is:

$$score(\mathcal{Q},\mathcal{T}) = \sum_{v \in \mathcal{Q}} hit(v,\mathcal{T}) \tag{5.1}$$

where v is a *key-location* hash value and $\mathcal{Q}$, $\mathcal{T}$ collections of *key-location* values of a query and an indexed sketch respectively.

We observe that objects are often drawn with strong mirror symmetries across the vertical axis. An airplane sketch for instance can be drawn facing the left or the right side of the canvas depending on the drawing style of each individual. A airplane sketch facing to the left is therefore likely to match well to sketches of the same category facing to the right and vice-versa. We make our patching framework flip invariant across the vertical axis by generating a new horizontal flipped sketch for each new query and match both versions against the database. We keep the highest score among the two versions for each indexed exemplar.

$$score_{(flip)}(\mathcal{Q},\mathcal{T}) = \max\{score(\mathcal{Q},\mathcal{T}), score(\mathcal{Q}_{flipped},\mathcal{T})\} \tag{5.2}$$

The new flipped version of the sketch, $\mathcal{Q}_{flipped}$, is obtained by flipping its columns in the left-right direction.

The modified voting function expands the matching scope of (5.1) and is able retrieve flipped variants of a query in a database. The rank k retrieval induced by (5.2) and query $\mathcal{Q}$ is defined as:

$$rank(k,\mathcal{Q}) = \arg\min_{1,\ldots,k}\{score_{(flip)}(\mathcal{Q},\mathcal{T}_1),\ldots,score_{(flip)}(\mathcal{Q},\mathcal{T}_n)\} \tag{5.3}$$

where T_j is the j-th indexed sketch in the database and n the total number of indexed exemplars.

If there are $|C|$ available classes $\Omega = \{\omega_1,\omega_2,\ldots,\omega_{|C|}\}$ and each exemplar T_j is attached to a known class ω, we can predict the unknown class $\acute{\omega}\omega$ of a new query using the K-nearest neighbor classification rule on the ordered ranking of (5.3).

$$\acute{\omega}(\mathcal{Q},K) = \text{KNN}\left(rank(1,\mathcal{Q}), rank(2,\mathcal{Q}), ..., rank(K,\mathcal{Q})\right) \tag{5.4}$$

Equation (5.4) discovers the K nearest samples of $\mathcal{Q}$. The most frequent occurring class among the K samples decides on the class prediction $\acute{\omega}$. When multiple classes occurring equally frequently the prediction is set to the class of top ranked sample.

5.4 Category Filtering

The computational cost of matching can be reduced by generating a sort list of categories. A general purpose learning algorithm can be employed to select the m most probable categories of a query Q, where $m \ll |C|$. Subsequently, the matching and recognition can be executed only for samples belonging to the m pre-selected categories. Apart from scaling down the search domain, category filtering offers accuracy boost via sample elimination. The similarity criteria can vary between category filtering and matching. For instance, the former can select categories with local appearance similarity while matching can discover holistic structure and flip correspondences.

In this work, we choose the SVM classifier to implement category filtering over DPLE for two reasons. SVMs shown slightly better sketch recognition accuracy than DPLE and their classification output is category based instead of sampled based. That is convenient when the goal is to rank all the available categories given a sample. SVMs is a supervised binary classifier that attempts to locate a hyperplane that maximizes the separation margin between two classes. The objective function of the SVMs can be formed as a solution of a quadratic programming problem that can be solved via various methods, a popular one being the sequential minimal optimization (SMO) algorithm. More details and theory on SVMs can be found in [32]. The maximum-margin hyperplane can be described from the samples that lie on the margins. These samples are named support vectors and denoted as $\mathbf{s}_i$. Classification on more than two classes can be achieved by the *1-vs-all* rule. If there are $|C|$ classes in total, we train $|C|$ separate SVM classifiers with each paradigm receiving the train samples of one class as positives and the rest as negative. The decision function of a new sample $\mathbf{x}$ on the binary classifier of category c is defined as:

$$f^c(\mathbf{x}) = \sum_i a_i^c K(\mathbf{s}_i^c, \mathbf{x}) + b^c \tag{5.5}$$

where a_i^c is a weight assigned to each support vector during training and $b^{(c)}$ the bias threshold which is again obtained during training. K is a kernel function. The sample $\mathbf{x}$ is assigned to the class with the maximum function response, i.e. the class that represents the most confident decision. To facilitate category filtering, the m predominant categories with highest decision values are selected.

A BoF representation is adopted for sketches during the category filtering. We uniformly sample a 35×35 grid on each sketch and calculate a feature vector on a 40×40 image region. Each region is divided in 4×4 cells and coarse quantized into 4 orientation bins between $[0, \pi)$. The resulting local features vector has 64 dimensions. We generate a codebook of 500 words based on clustering of the local descriptors. The final sample descriptor $\mathbf{x}$ is a histogram of visual words. The SVMs are trained with cross-validation and optimized parametrization in each dataset.

5.5 Experiments

5.5.1 Datasets

The evaluation is carried out on two challenging sketch datasets. As in previous chapters, we use the EitzSKETCH dataset [39], which incorporates 250 object categories with each category being represented by 80 sketches. As the sketches are freely drawn by humans the dataset exhibits high variance over the categories. The participants of the study recognized on average 73.1% of all sketches correctly. A more detailed description of the dataset can be found in Section 2.3.2.

We also use the query set of the Flickr15k benchmark [56], as a second evaluation dataset. 10 subjects of average artistic skill participated in this study. In total, there are 33 sketch categories describing shape, building landmarks, objects and scenes and 10 sketches per category, one for each subject. Some categories display high visual overlap. An overview of the available classes is available in Figure 5.2.

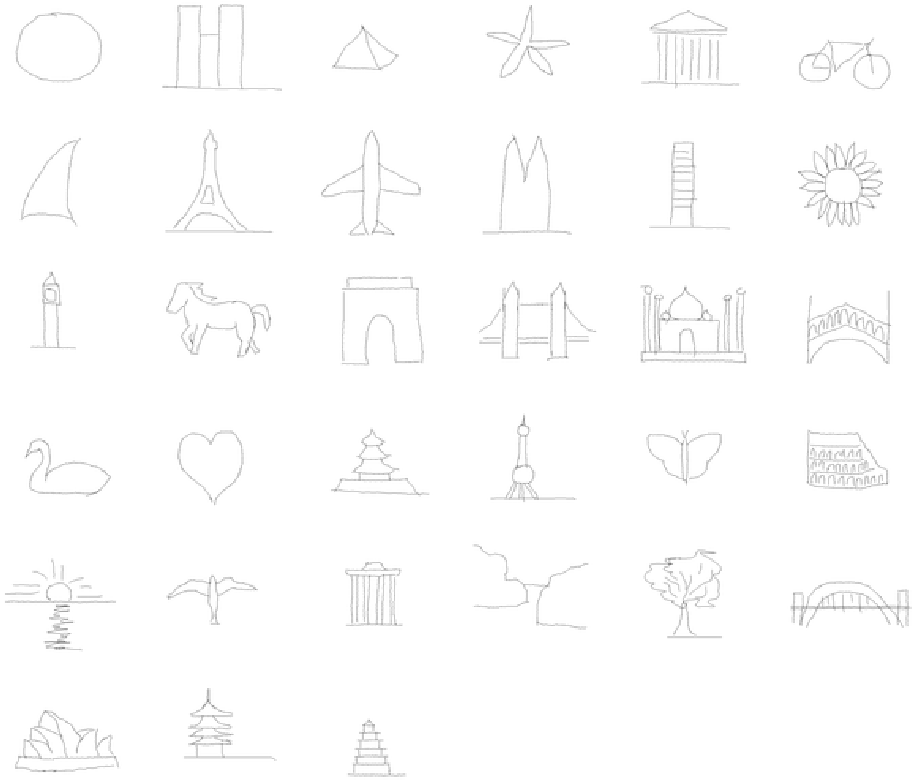


Figure 5.2: Preview of the 33 sketch categories of Flickr15k.

5.5.2 Experimental Setup

Features. In all experiments a 35 × 35 grid is applied to sketches and local features are computed in square patches of 20 pixels radius. We explore the performance of three type of features in sketch matching, namely HOG, BRIEF [16] and LBP [96]. BRIEF and LBP descriptors are implemented with the settings described in Section 2.2.1. The HOG features are computed according to [39]. A 4 × 4 cell grid is applied to each patch and in each cell a 4-bin orientation histogram in the range $[0, \pi)$ is computed. The final HOG descriptor for each patch is a 64-dimensional vector.

Parameter settings. We empirically found that the min-hash parameters k and s have little effect on performance, hence we fix them to $k = 50$ and $s = 2$. We also globally fix the binarization threshold to top 20% of the vector values for the HOG and LBP descriptors. During the *key-location-*

Table 5.1: Sketch recognition accuracy comparison in EitzSKETCH and Flickr15k datasets. In the case of patch hashing the supervised reported results are achieved with category filtering.

	TU-Berlin		**Flickr15k**	
Method	**Unsupervised**	**Supervised**	**Unsupervised**	**Supervised**
KNN	45% [39]	N/A	57.2% ± 3.7	N/A
SVM	N/A	56% [39]	N/A	76.9% ± 3.6
DPLE	N/A	53.7%	N/A	64.5% ± 2.9
Yi *et al.* [74]	53.3%	61.5%	N/A	N/A
PH-LBP	29.6 ± 0.2	39.1% ± 0.2	60.6% ± 2.8	66.0% ± 5.5
PH-LBP-flip	31% ± 0.1	39.7% ± 0.1	61.5% ± 5.8	67.2% ± 5.3
PH-BRIEF	44.4% ± 0.1	49.2% ± 0.2	67.2% ± 4.9	69.0% ± 3.8
PH-BRIEF-flip	45.5% ± 0.3	51.6% ± 0.1	67.9% ± 5.1	69.1% ± 2.7
PH-HOG	56.2% ± 0.2	61.4% ± 0.3	74.2% ± 1.8	77.4% ± 3.6
PH-HOG-flip	**58.5**% ± 0.2	**62.8**% ± 0.2	**75.7**% ± 2.8	**77.8**% ± 4.7

index construction we discard min-hash values that occur more than 100,000 times in EitzSKETCH and 7000 in Flickr15k dataset. At the voting stage, we use the Manhattan distance to enforce locality constraints and set the corresponding threshold to 4. For the KNN classification of the rankings, we use K values between $\{1, 3, 5, 7, 9\}$ and report the best score. Finally, category filtering is performed with SVM, as described in Section 5.4. We keep the top 5 categories in EitzSKETCH and top 2 in Flickr15k.

Notations. We denote as PH-HOG, PH-BRIEF, PH-LBP the patch hashing methods with the corresponding descriptors. We prefix the category filtered results with the *SVM-* keyword and suffix the flip invariant methods with the *-flip* keyword. For example, SVM-PH-HOG-flip denotes the patch hashing algorithm trained with HOG features, category filtering and flip invariance filters on.

Alternative methods. We compare our algorithm against recently proposed structure based techniques that demonstrated competitive sketch recognition results, namely, the star-graph model of Yi *et al.* [74] and the sketch retrieval algorithm of Ma *et al.* [83]. Additionally, we include comparisons against the baseline KNN and SVM methods in both datasets, built with the

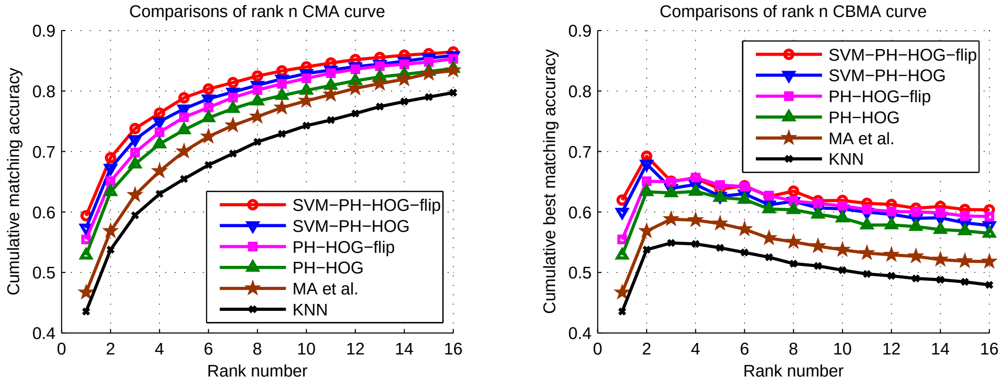


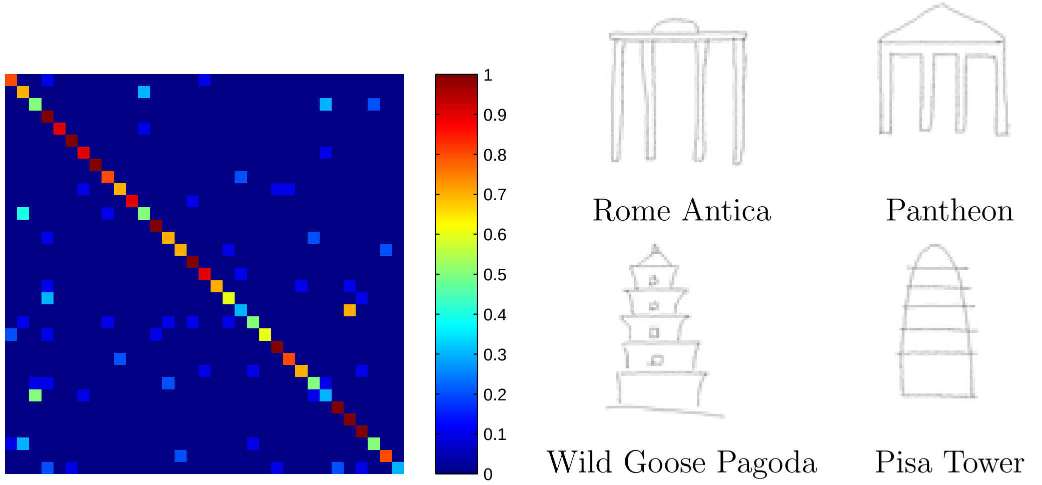
Figure 5.3: Rank n CMA and CBMA curves in the EitzSKETCH dataset. Best viewed in color.

HOG features.

Metrics. Following [74], we perform 4-fold cross validation in the EitzSKETCH dataset and 5-fold in Flickr15k. We measure the recognition accuracy on both datasets and additionally report the Cumulative Matching Accuracy (CMA) and the Cumulative Best Matching Accuracy (CBMA) in the EitzSKETCH dataset for a fair comparison with [83]. CMA shows how often the correct category appears in top n retrieved sketches, while CBMA measures the correctly retrieved sketches that account for the most of the top n retrieved sketches.

5.5.3 Discussion

Table 5.1 summarizes recognition accuracy over the two datasets. The category filtered (supervised) SVM-PH-HOG-flip algorithm achieves a new state-of-the-art score of 62.8% in the challenging EitzSKETCH dataset. We also note that the unsupervised PH-HOG-flip outperforms the SVM and the star-graph model by a large margin. Both our method and [74] are based on structured features. We attribute the superiority of patch hashing to the robust matching between local patches via the spatial voting and the binarization process that highlights the major patch orientations. Moreover, we verify that horizontal flip invariance improves the recognition performance from 56.2% to 58.5% and is a well-suited property for sketch matching. Figure 5.5 illustrates queries that benefit for flip-invariance. We observe that in several non flip-



(a) Confusion matrix of Flickr15k.

(b) Most confused class pairs.

Figure 5.4: Confusion matrix of Flickr15k for PH-HOG-flip. Red tones correspond to higher accuracy. The two most confused category pairs are illustrated on the right part of figure. Best viewed in color.

invariant cases the best matched exemplar belongs to different category than the query, although there is considerable visual similarity among them. The flip-invariant method re-evaluates the matching scores and ranks higher same class horizontal reflected sketches. We have also experimented with vertical symmetry invariance but found it has negative effect on the performance.

The alternative features fail to be as discriminant as their HOG counterparts. Congruent conclusions have been drawn in the previous sketch/image matching experiments in Chapter 3. BRIEF and LBP are sensitive to noise. Sketch patches contain sparse information and binary values, thus amplify this drawback. BRIEF performs better than LBP due to the patch smoothing filtering which reduces to some extent the noise.

Results on the Flickr15k dataset are coherent with the findings on EitzSKETCH. The impact of flip invariance is slighter due to the low number of samples per category that leads to limited reflection variations within each class. Figure 5.4 presents the confusion matrix for the PH-HOG-flip method along with the two most confused category pairs. Visual inspection of samples belonging to these categories reveals high appearance overlap between them. Indeed, even a human cannot distinguish between sketches belonging to categories *Rome Antica* and *Pantheon* without any disambiguation hint. This

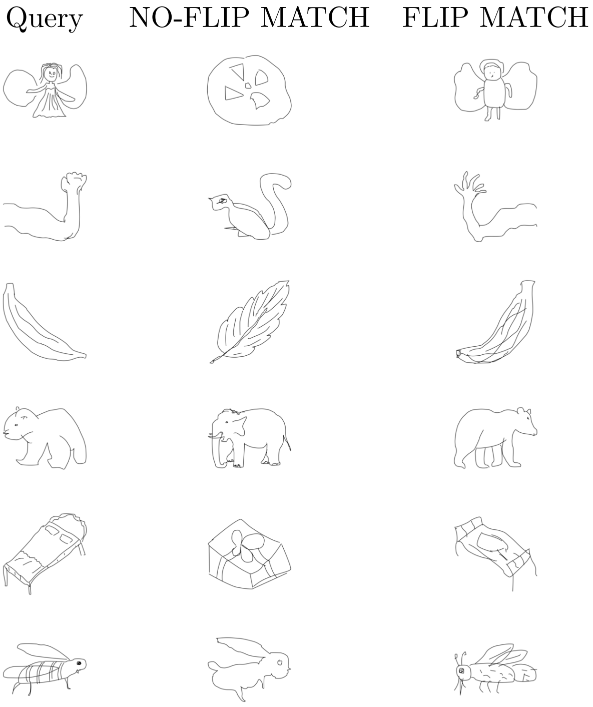


Figure 5.5: Examples that benefit from flip invariance. (Left) Original query. (Middle) False rank 1 classification by non flip-invariant matching. (Right) Correct rank 1 classification by flip-invariant matching.

highlights the limits induced in sketch recognition by the abstract oriented artistic skill of the average user. A text label accompanying each sketch can shed light on analogous situations.

We further evaluate PH-HOG in the EitzSKETCH dataset using the CMA and CBMA curves. The last 20 sketches of each category are used as queries. We compare patch hashing against Ma *et al.* [83] which has been especially developed for sketch retrieval. KNN classification [39] is also included in the evaluation as baseline. Figure 5.3 displays the curves. SVM-PH-HOG-flip achieves superior performance in both metrics and maintains the edge over all ranks. Once more, flip invariance contributes to more robust results. Our scheme is equally scalable to [83]. We implemented patch hashing in a 16-core machine. The average query time in EitzSKETCH dataset is 0.2s for the PH-HOG and 0.3s for the flip invariant version. The *key-location*-index occupies 514MB. Accordingly, in Flickr15k the index needs 48MB and average query times are 0.07s and 0.08s for the PH-HOG and PH-HOG-flip.

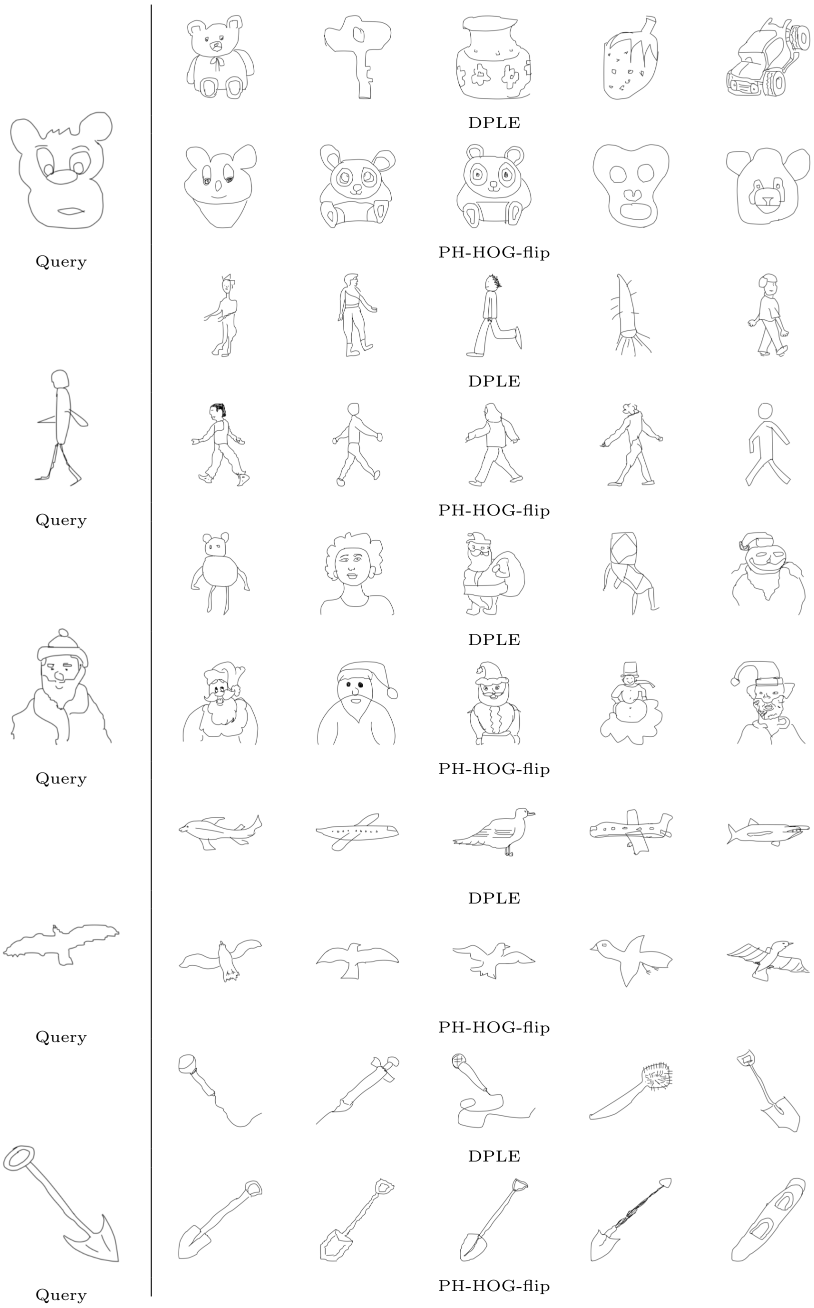


Figure 5.6: Top5 retrieved sketches for DPLE and PH-HOG-flip.

5.5.4 Improvements over DPLE

As noted in the introduction, sketch matching can refine the recognition results of any general learning algorithm. We design an experiment to demonstrate the performance edge gained by post recognition re-ranking. We divide the EitzSKETCH dataset in three partitions with stratified sampling and use the

Table 5.2: Recognition accuracy of KDPLE and re-ranked KDPLE with sketch matching in EitzSKETCH dataset.

Method	K=1	K=3	K=5	K=7	K=9
KDPLE	49.4%	51.1%	53.3%	53.7%	54.3%
KDPLE+PH-HOG	54.5%	58.6%	57.9%	57.0%	56.2%
KDPLE+PH-HOG-flip	**55.9%**	**60.1%**	**59.4%**	**58.2%**	**56.8%**

first two for training and the last for testing. The DPLE algorithm with optimal parameters is trained and a ranking of the training samples is generated for each query based on Euclidean distances in the learned subspace. The KNN accuracy over K values between $\{1, 3, 5, 7, 9\}$ for the DPLE rankings is reported. Subsequently, the first 12 retrieved samples of each query are re-ranked based on their matching score with the test sample. KNN classification accuracy is re-evaluated in the new rankings. We use the PH-HOG and PH-HOG-flip algorithms to perform the matching. Table 5.2 summarizes our findings. A noticable accuracy boost is observed over all K values reaching a peak of 18% increase for $K = 3$. The gain is greater for small K values as the first ranked samples represent the best matched sketches of the the train set and the confidence of belonging to the same category as the query is high. The positive impact of flip-invariance is verified once more in this experiment. Moreover, Figure 5.6 displays a comparison between the top 5 retrieved sketches of DPLE and PH-HOG-flip. As expected, the latter are more consistent and semantically similar to the query.

5.6 Semantic Sketch Based Image Retrieval

The motivation behind sketch recognition in our study is to infuse semantic information to the image retrieval module without user intervention. Semantics are essential in SBIR to eliminate outliers and irrelevant images from the search results. One way to acquire human knowledge is to prompt the user to provide a text label along with the drawn sketch. This renders the query process cumbersome and counter-intuitive to the purpose of SBIR which is drawing. Here, we show that a fully automatic semantic retrieval framework, that requires only a sketch as a user input, can produce robust retrieval results.

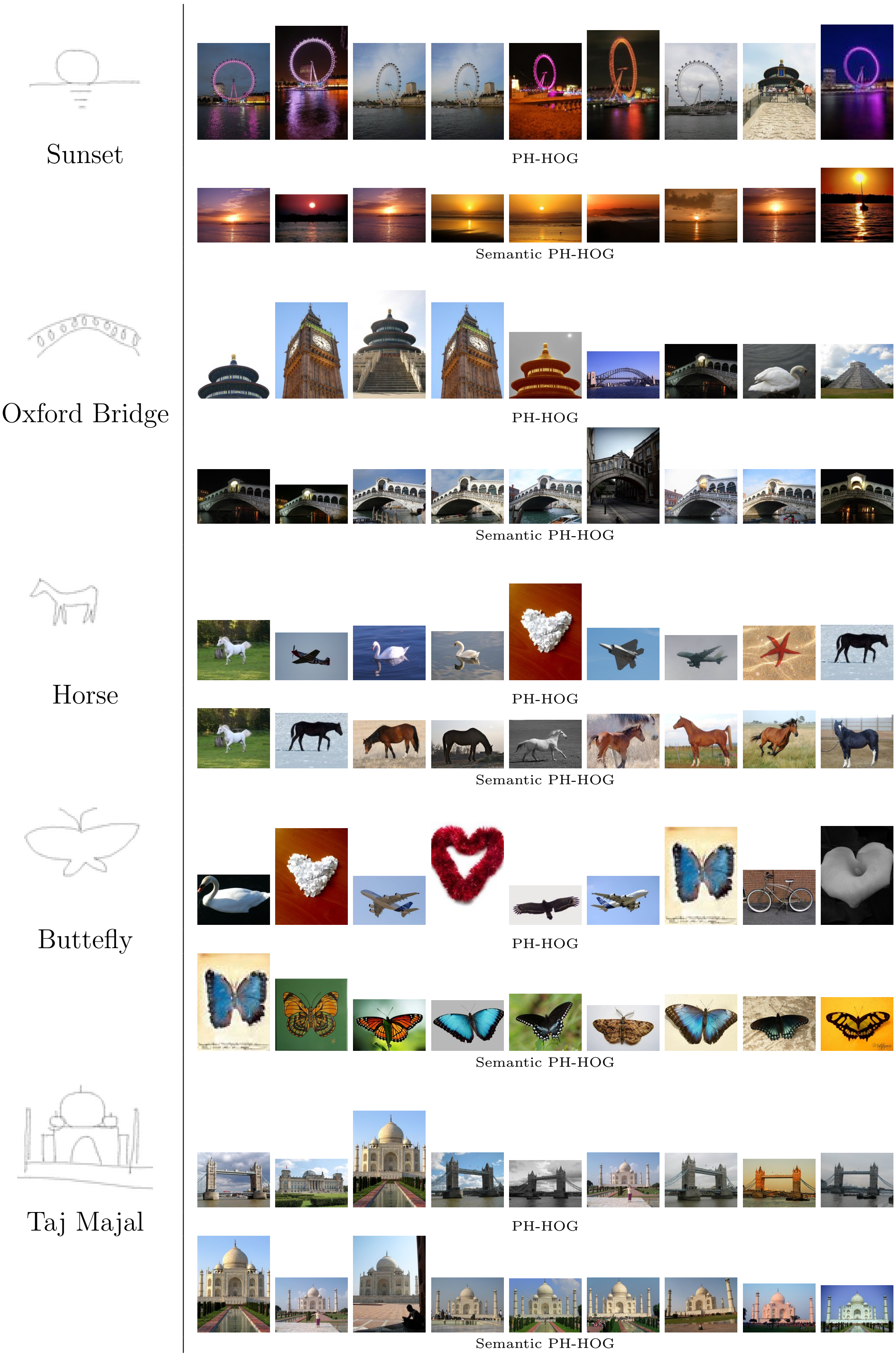


Figure 5.7: Top 9 retrieved images in Flickr15k dataset for few sample queries for PH-HOG and semantic PH-HOG. Semantics are automatically infused to PH-HOG via sketch recognition.

Table 5.3: Impact of sketch recognition in SBIR. Semantics are infused in the PH-HOG rankings via sketch matching. Retrieval quality measured with the MAP score in Flickr15k dataset.

Method	**MAP on Flickr15k**
PH-HOG	0.200
Semantic PH-HOG	0.762

An overview of our semantic framework is depicted in Figure 1.4.

We demonstrate the performance of our approach in the Flickr15k dataset. This image collection is labeled and each image is either related to one of the available 33 sketch categories or denoted as noise. When a new sketch query is provided, the recognition module categorize it to one of the available classes. Succeeding recognition, retrieval is performed to the subset of images that are labeled with the recognized category. The unsupervised flip-invariant matching algorithm, introduced in this chapter, implements the recognition module. Retrieval is carried out through the PH-HOG method presented in Chapter 3. The MAP metric evaluates the quality of the retrieval results, penalizing high ranked erroneous retrievals. Table 5.3 reports our findings. The MAP score of PH-HOG is 0.200. When semantics are injected in the process, the MAP score elevates to 0.762. A comparison between the top ranked images of the two models is illustrated in Figure 5.7. The quality of the semantic enhanced retrievals is evident. All the outlier images have been discarded from the top ranks. The diversity of the results has also been considerably improved. The whole process is fully unsupervised and does not require any user action, other than sketching.

The semantic MAP score directly correlates with the sketch recognition accuracy. Experiments in Section 5.5.3 have shown that our sketch recognition module reaches an accuracy rate of 75.7% in the Flickr15k dataset. We notice the semantic MAP score and the sketch recognition accuracy are approximately equal. Therefore, robust recognition is a fundamental requirement in our semantic framework. Still, there will be cases that recognition will fail. A possible remedy to this situation is to present the user with a set of the most probable categories in descending order. Even if the primary projection is false, there is high probability the correct answer is included in the set. For instance, in Flickr15k the correct sketch category is present 95.5% of the times in the top 15 retrieved sketches.

5.7 Conclusions

In this chapter, we presented a flip-invariant sketch matching and recognition algorithm. State-of-the-art results achieved in the challenging EitzSKETCH dataset over competitive alternative methods. Our category filtered algorithm recognizes correctly a sketch 62.8% of the times when the human rate is 73.1%. Best results are also demonstrated in the Flickr15k sketch set.

We identify the key components of a robust sketch-to-sketch match as the following: a) histograms of oriented gradient for patch representations b) structure c) horizontal flip-invariance. The latter boosted the recognition rates in all experimental setups and motivates further research in features that encode Gestalt principles, such as continuity and grouping. Moreover, there are recent results in the psychology literature [106] justifying the use of various types of symmetries as a set of general constraints to help in vision problems. The patch-hashing mechanism can not handle affine transformations. An approach that will combine affine invariance and structure preservation could be worth investigating.

Sketch recognition can act as a preprocessing step in a sketch based image retrieval framework to infuse semantics. We have experimentally evaluated the performance of this approach and found it induces great impact in the retrieval quality.

Chapter 6

Conclusions and Future Developments

6.1 Concluding Summary

The scope of this thesis is to review and extend the knowledge in machine sketch understanding and its applications. Our research focuses on the following major questions:

- How can a machine match a free hand-drawn sketch to a database of photos in real time?
- How possible is for a machine to recognize a sketch with human accuracy?

Our attempt to answer these questions lead to the development of contributions that extended the state-of-the-art in sketch based image retrieval (SBIR) and sketch recognition. Still, despite the considerable progress reported here, human-like accuracy has yet to be reached. In particular, sketch-to-image matching can be visually accurate but the lack of semantic information propagates inconsistencies to the results. Our semantic SBIR framework bridges to some extend this knowledge gap, yet is strongly dependent on the accuracy of sketch recognition. In this domain the findings are promising. We have shown recognition accuracy that reaches 62.8% in a database of 250 categories when the human accuracy is 73.1%. Furthermore, machine performance overcomes humans when top 3 category predictions are allowed.

A fundamental step towards machine sketch understanding includes the detection of discriminant properties that can uniquely identify a sketch. From our work, the following deductions emerged. Sketch-to-image and sketch-to-sketch matching share identical description criteria. Essentially, they form the same domain, if edge detection is applied to an image. Since edge detection research has reached a mature level and natural lines can be extracted from an image [89], it does not become a barrier in sketch-to-image matching. A combination of local histograms of orientated gradients and structural information offers robustness and should be a starting point for future improvements in sketch description. Our research showed that coarse patch representations focusing on dominant lines and discarding finer details suffice to efficiently describe a local region of a sketch. However, local visual appearance on its own can not adequately capture the global gist of a sketch. Structure constraints need to be enforced. This is equivalent to the pieces of a jigsaw puzzle which individually might look quite similar, but only if they are placed in an appropriate position reveal the whole picture.

The role of horizontal reflection symmetry in sketch-to-sketch matching consists another significant finding of our research. Its effect and application have not been earlier studied in the sketch domain. Horizontal symmetry is frequently exhibited in human drawings, hence a flip-invariant matching algorithm obtains an advantage over alternative non flip-aware methods. Our experiments revealed that vertical and diagonal symmetries do not grant performance benefits in sketch recognition.

The high volume of images on the web renders scalability a crucial aspect of any image retrieval algorithm. The need for parallel solutions is constantly increasing. Our spatial-aware hash index manages to encode the earlier mentioned appearance and structural features and enables queries in sub-linear time. Additionally, the index can be sharded into smaller chunks and be distributed across several machines for parallel computing.

Furthermore, this document constitutes an important reference for other researchers entering the area, as a useful starting point to grasp the current trends in the field, as well as look for promising routes for further research advancement. In the remainder of this chapter, we present a summary of the contributions of our work and insights for future research on the field.

6.2 Summary of Contributions

This section offers a brief overview of the contributions in each chapter.

- In Chapter 3, we proposed a method for scalable image retrieval given a sketch query. Our patch hashing algorithm combines appearance and structure features and employs the min-hash algorithm to efficiently index these properties. To validate our approach, we conducted experiments in three SBIR benchmarks of varying size from 160 to 100,000 images. Two kinds of metrics were employed in the evaluation. The Mean Average Precision (MAP) score which rewards correct top ranked images and the Kendall's coefficient which measures correlation between machine and human rankings. In both cases, patch hashing outperformed competitive techniques and set a new state-of-the-art score. The role of feature description was also investigated by conducting experiments with three different kinds of features. The HOG descriptor proved to be far superior than the alternatives. Space and time analysis of our algorithm showed that it can cope well with large datasets. A series of tests highlighted the lack of affine invariance in patch hashing which consists a possible drawback of our method. Furthermore, several cases of noise in the top ranked results motivated a research shift towards sketch recognition in an attempt to infuse semantics in the retrieval process.

- In Chapter 4, Discriminant Pairwise Local Embeddings (DPLE) a novel supervised manifold learning algorithm was introduced to facilitate, among others, sketch recognition. DPLE generates discriminant subspaces where data can be better separated, while their structure is maintained. Moreover, its optimization function corresponds to a generalized eigendecomposition problem and can be efficiently computed. As a general learning technique, DPLE was evaluated in two domains, namely face and sketch recognition. In both domains, a collection of competitive unsupervised and supervised dimensionality reduction algorithms were tested as well. DPLE produced consistently superior results. Specifically, in the sketch domain KNN classification in the DPLE subspace achieved comparable performance to the well known SVM classifier. On top of that, data dimensionality was reduced to half. A known issue of supervised subspace learning methods is that they can not update a learned embedding upon new sample arrival and need to be re-trained. The future work section offers directions on how to possibly overcome this problem.

- Finally, as described in Chapter 5, we investigate how a matching algorithm can improve the state-of-the-art in sketch recognition. We extend the earlier developed patch hashing model to incorporate horizontal flip-invariance. A category filter is also implemented to further boost recognition accuracy. Our matching method possesses dual functionality as it can be employed both for sketch-to-sketch retrieval and sketch recognition via the KNN classifier. We evaluated the retrieval quality and recognition accuracy our approach in the challenging EitzSKETCH and Flickr15k datasets. Horizontal flip-invariant matching achieved superior results over all experimental setups. Interestingly, the unsupervised performance of our method is higher than the supervised SVMs. Subsequently, the recognition results were exploited to perform semantic filtering on the SBIR rankings. Our unsupervised semantic SBIR framework demonstrated a significant increase in retrieval quality, measured with MAP, from 0.200 to 0.768.

6.3 Directions for Future Research

Research in machine sketch understanding is ongoing and rapidly evolving. The contributions and inquiries of this thesis lead to a series of paths that entail further exploration.

- We showed that the combination of appearance and structural features constitutes a description tailored to the sketch domain. Out patch hashing framework can encode these properties, yet is sensitive to affine transformations. Although, affine invariance is not a highly desirable feature in SBIR, it might be useful in sketch matching. Therefore, an extension of patch hashing robust to transformations such as scaling, rotation and translation while maintaining the structure setup of a sketch would be worth investigating. A possible way to handle this is to monitor relative distances between local patches. Therefore, if an affine transformation occurs we can refer to the original distance configuration and infer the changes. This approach requires a non-fixed grid and appropriate patch sampling should be performed.

- Exploration of alternative visual features better suited for sketch description. In this work, we focused on horizontal symmetry, still there are several other options worth investigating. For instance in [11], descriptors that can capture Gestalt principles such as repetition are suggested. Moreover, sketching is a time evolving process. The role of temporal information in free hand sketch description is currently not clear.

- Compact visual descriptors to accommodate extremely high volumes of images. In patch hashing each sketch is represented by multiple small collections of min-hash values. A more efficient solution would be a compact single sketch descriptor that can combine local properties and transfer them to a similarity function. Evaluations under this scheme can be expressed as a vector product and can be computed very efficiently. The traits of a recently introduced work [118], can be followed to achieve this goal.

- Out-of-sample extension of the DPLE method. As mentioned earlier, a new unknown sample can not expand the already learned subspace of DPLE without re-training with the full sample set. The out-of-sample extension allows a new sample to be injected in the subspace individually. Approaches similar to [10, 133], that treat images as matrices instead of vectors, could be followed to improve DPLE.

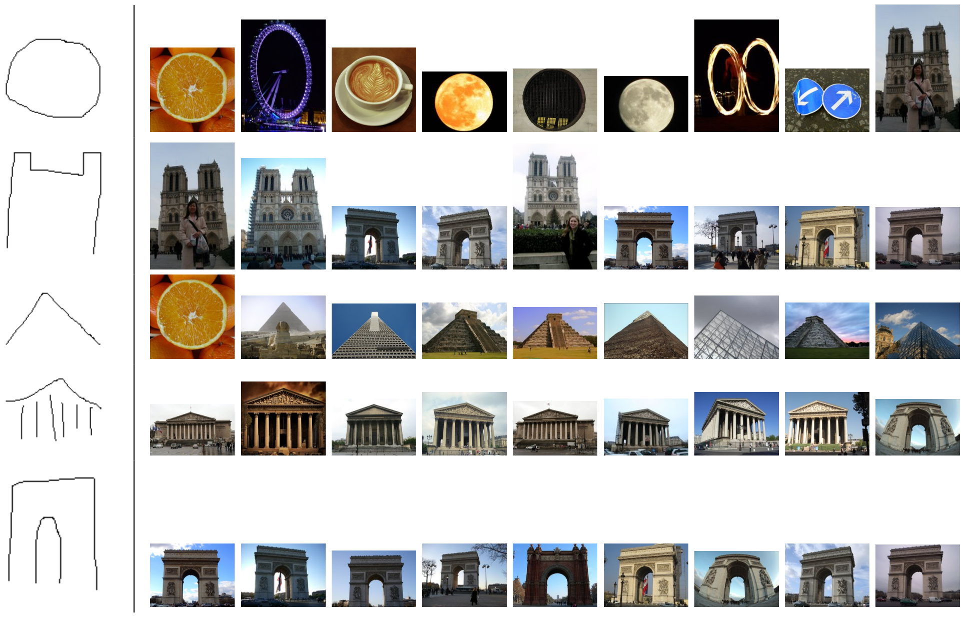
Appendix A

Sketch Based Image Retrieval Results

In this appendix, we display top ranked image retrieved images given queries from the three evaluations datasets, used in Chapter 3.

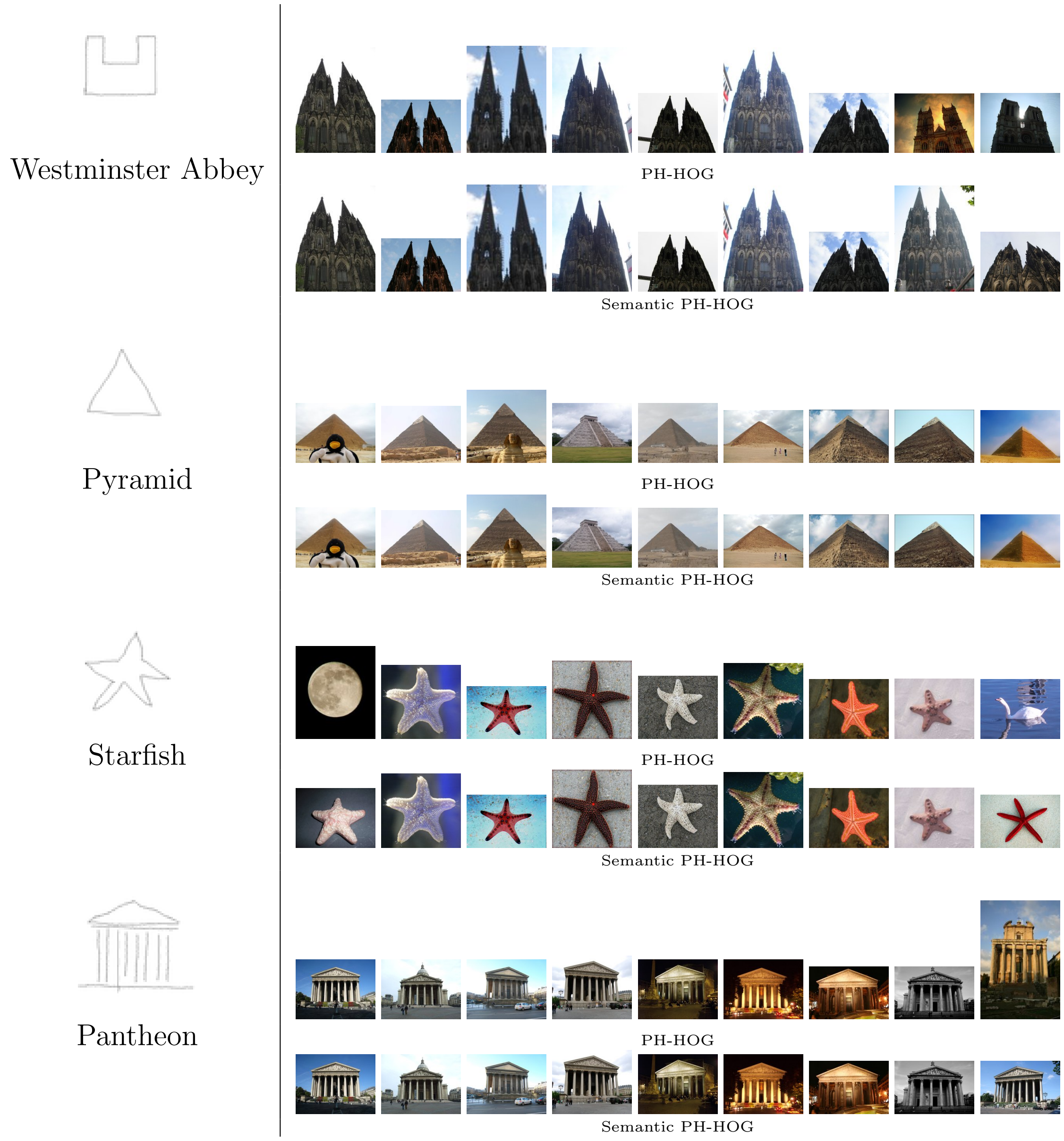
A.1 Flickr160

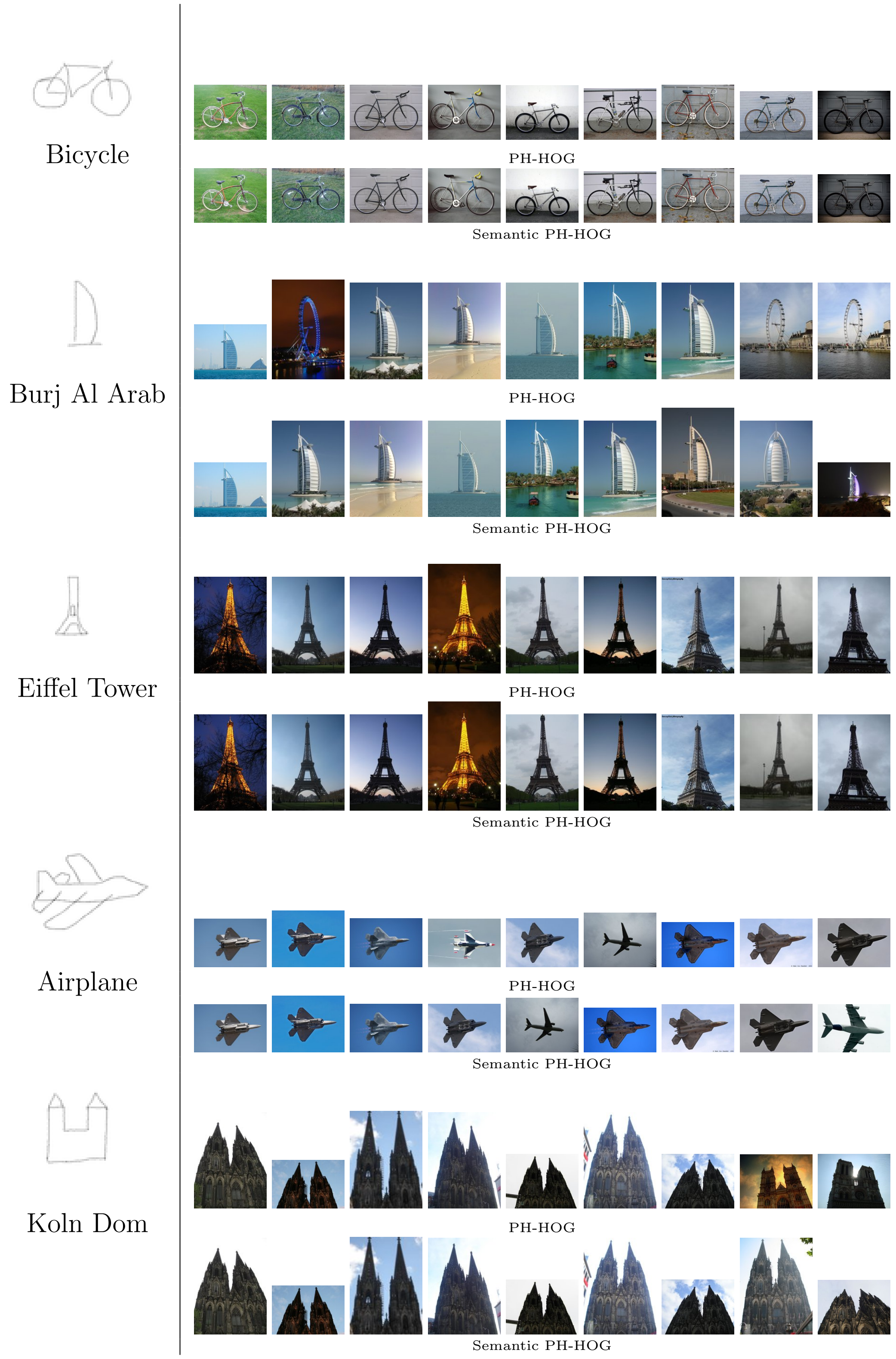
Figure A.1: Top 9 retrieved images for all query categories in Flickr160 dataset. The PH-HOG method of Chapter 3 is used for retrieval.

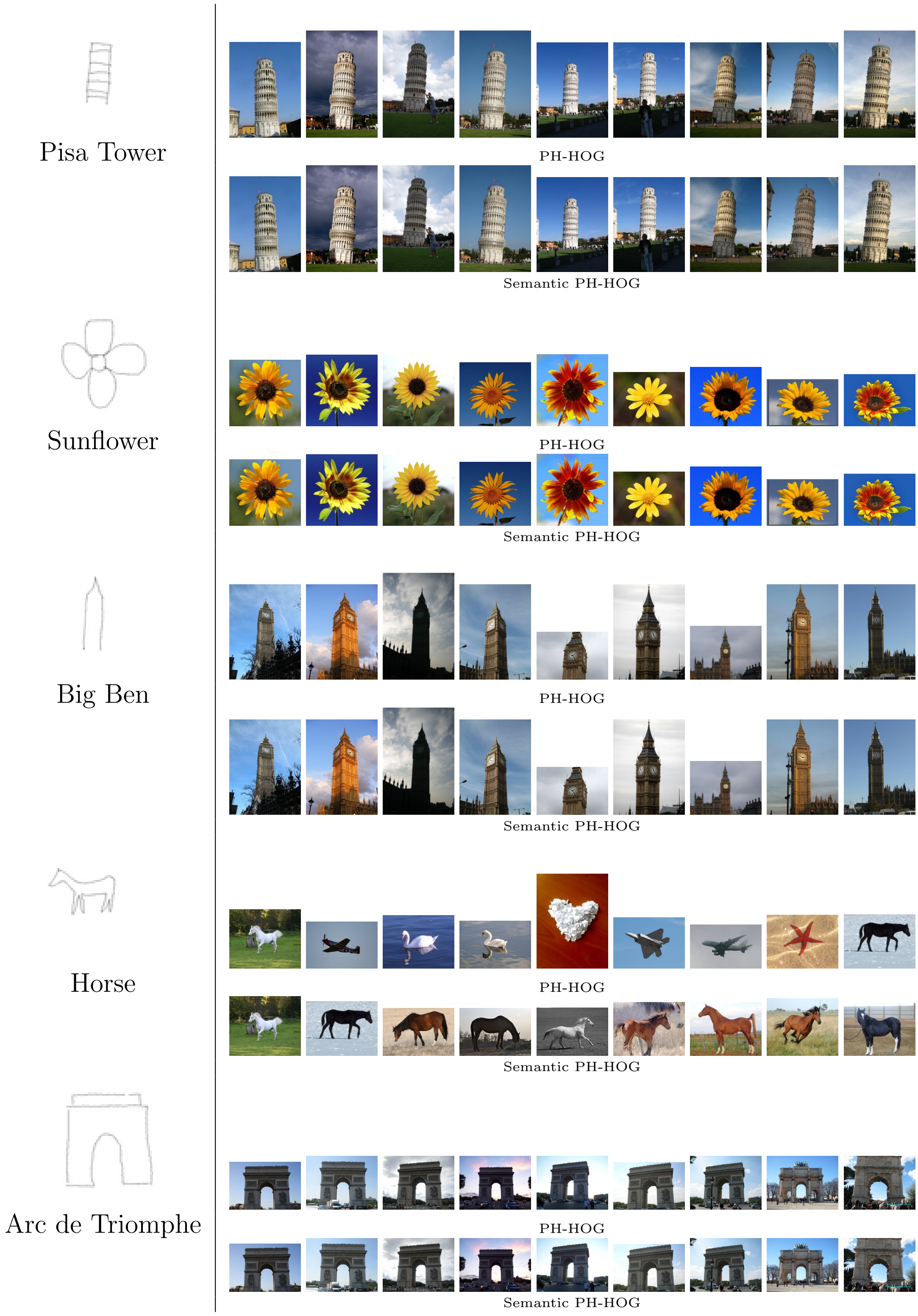


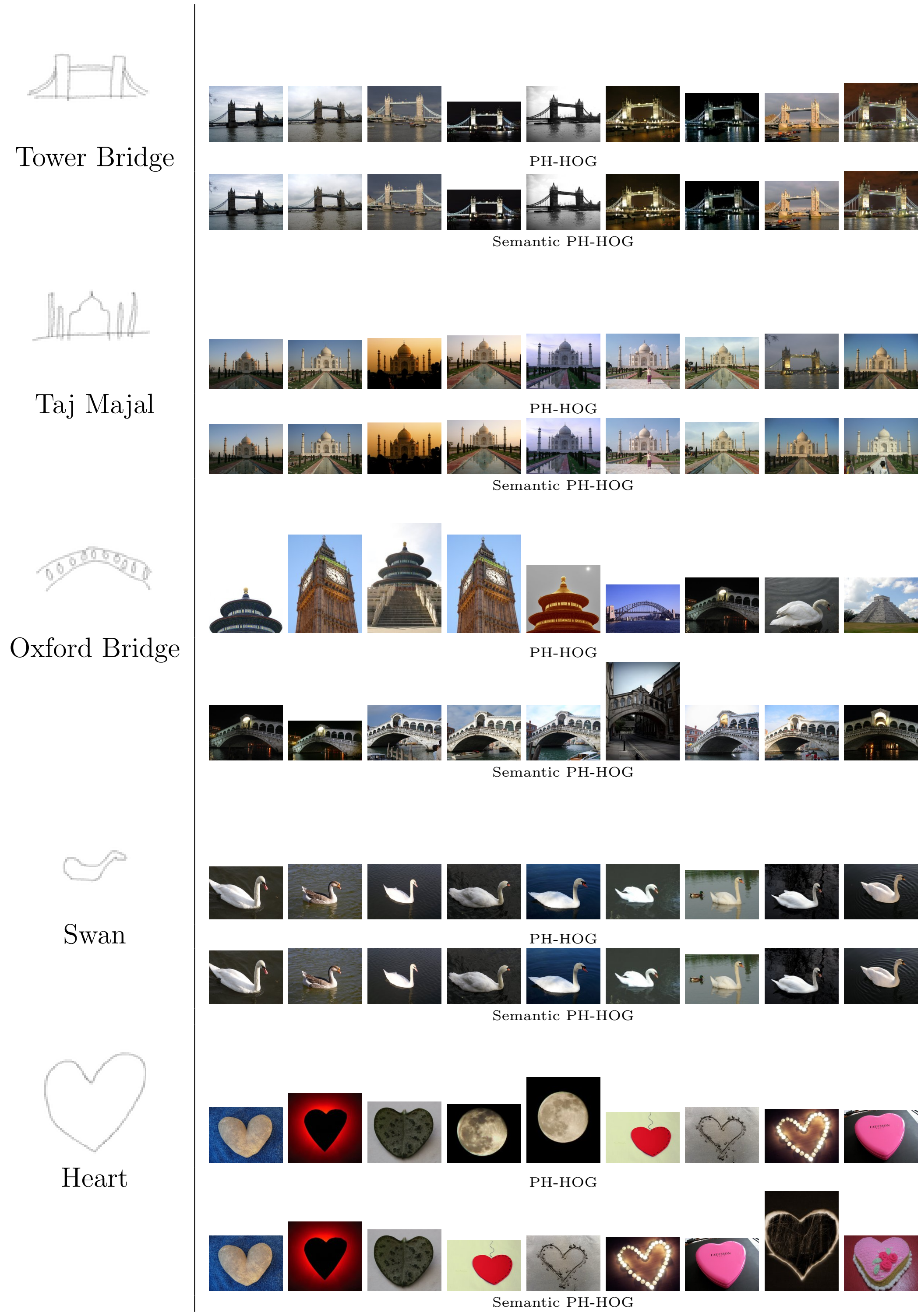
A.2 Flickr15k

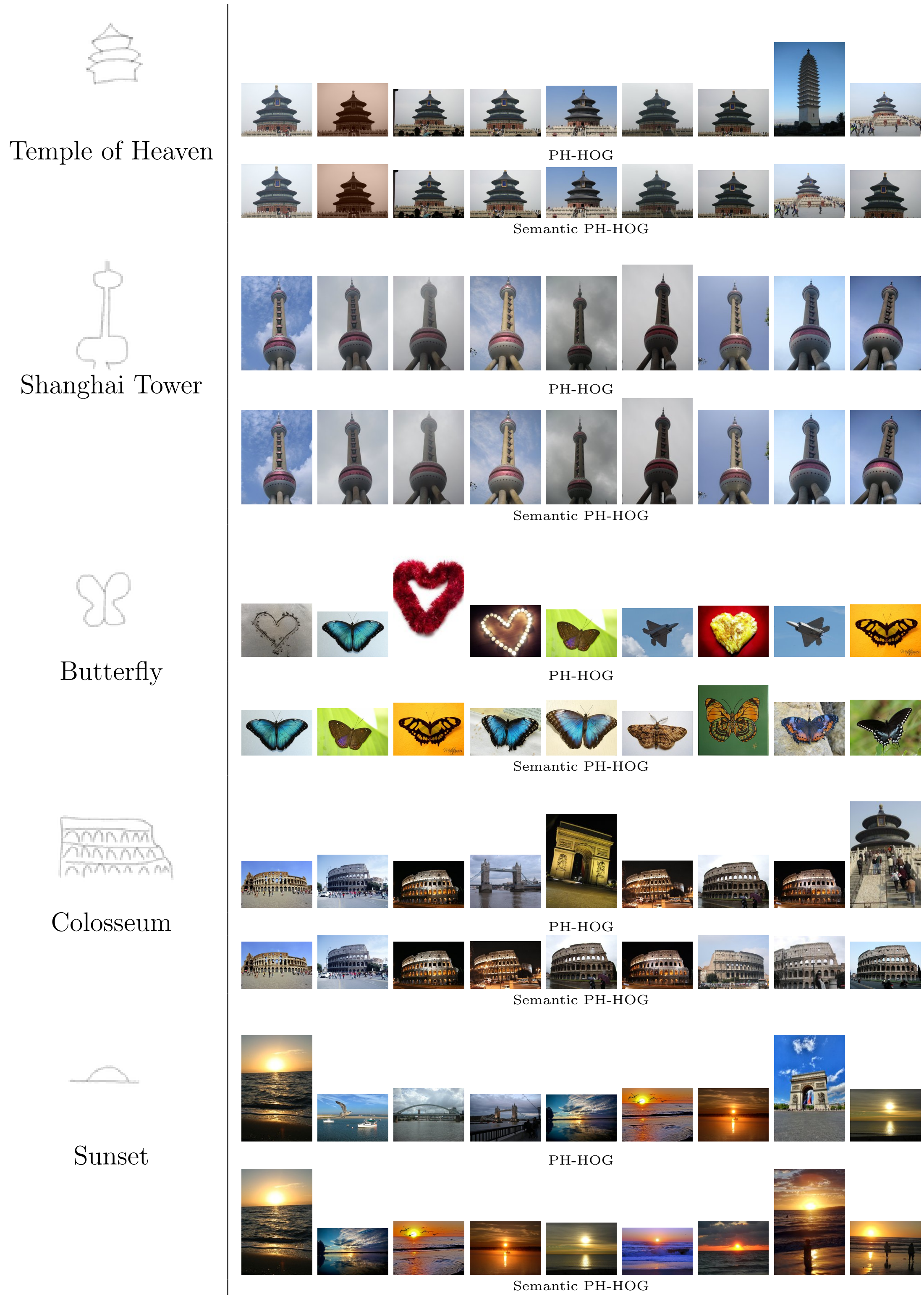
Figure A.2: Top 9 retrieved images in the Flickr15k dataset for all query categories. The PH-HOG method of Chapter 3 is used for retrieval. Semantic PH-HOG indicates results reranking based on unsupervised sketch recognition. Sketch recognition is carried out via flip-invariant matching introduced in Chapter 5.

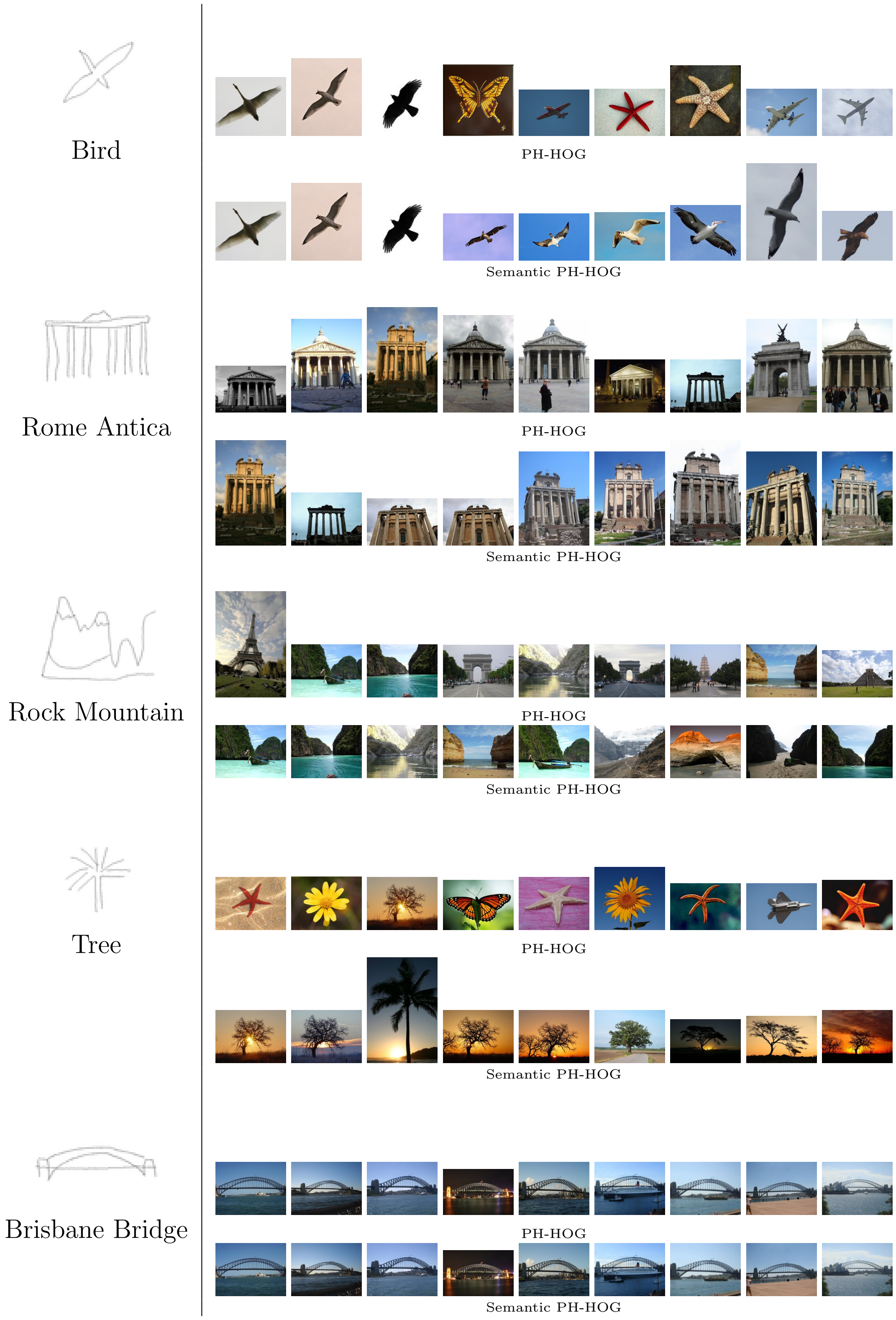


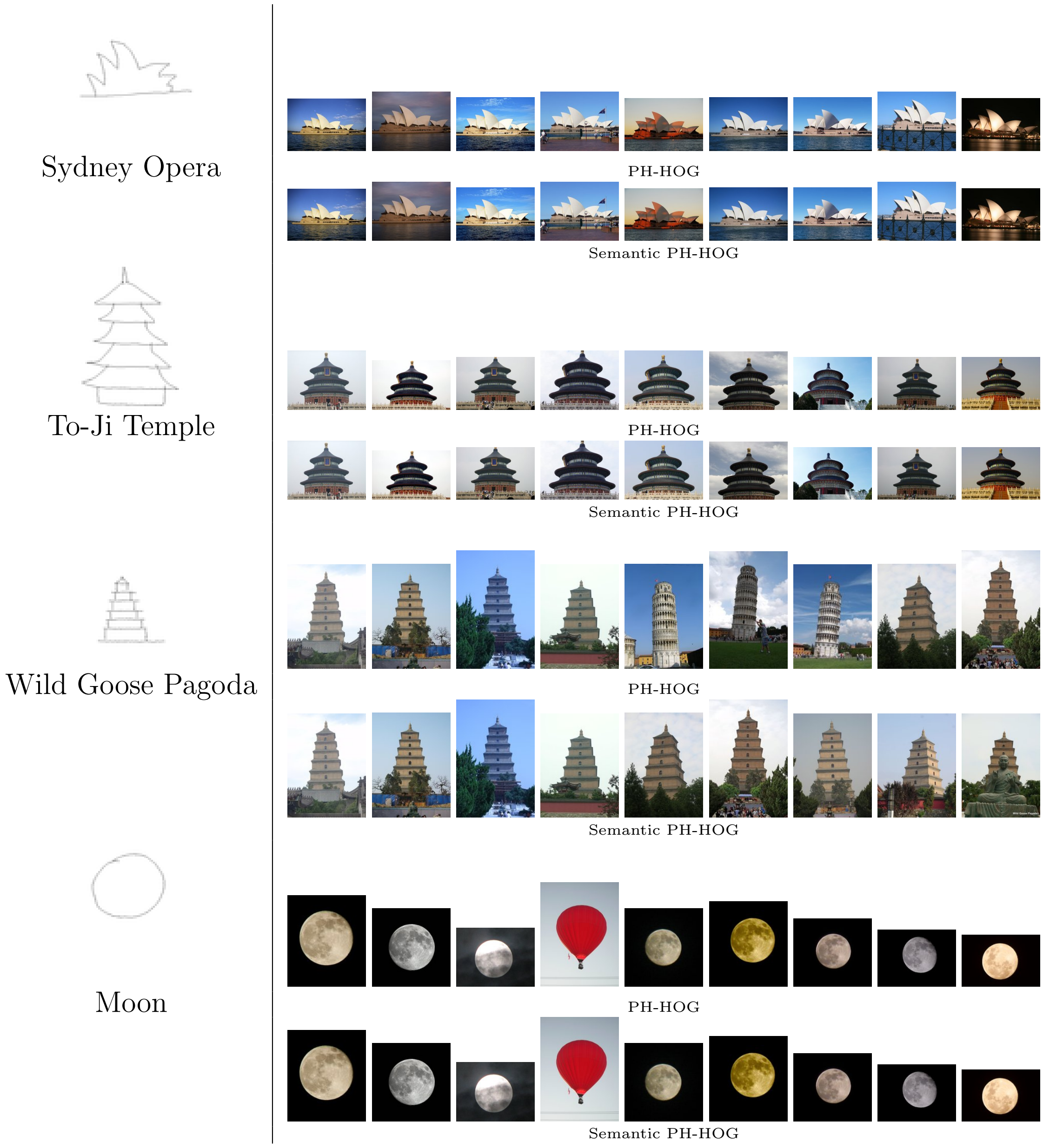

Bicycle
PH-HOG
Semantic PH-HOG
Burj Al Arab
PH-HOG
Semantic PH-HOG
Eiffel Tower
PH-HOG
Semantic PH-HOG
Airplane
PH-HOG
Semantic PH-HOG
Koln Dom
PH-HOG
Semantic PH-HOG


Pisa Tower
PH-HOG
Semantic PH-HOG
Sunflower
PH-HOG
Semantic PH-HOG
Big Ben
PH-HOG
Semantic PH-HOG
Horse
PH-HOG
Semantic PH-HOG
Arc de Triomphe
PH-HOG
Semantic PH-HOG


Tower Bridge
PH-HOG
Semantic PH-HOG
Taj Majal
PH-HOG
Semantic PH-HOG
Oxford Bridge
PH-HOG
Semantic PH-HOG
Swan
PH-HOG
Semantic PH-HOG
Heart
PH-HOG
Semantic PH-HOG

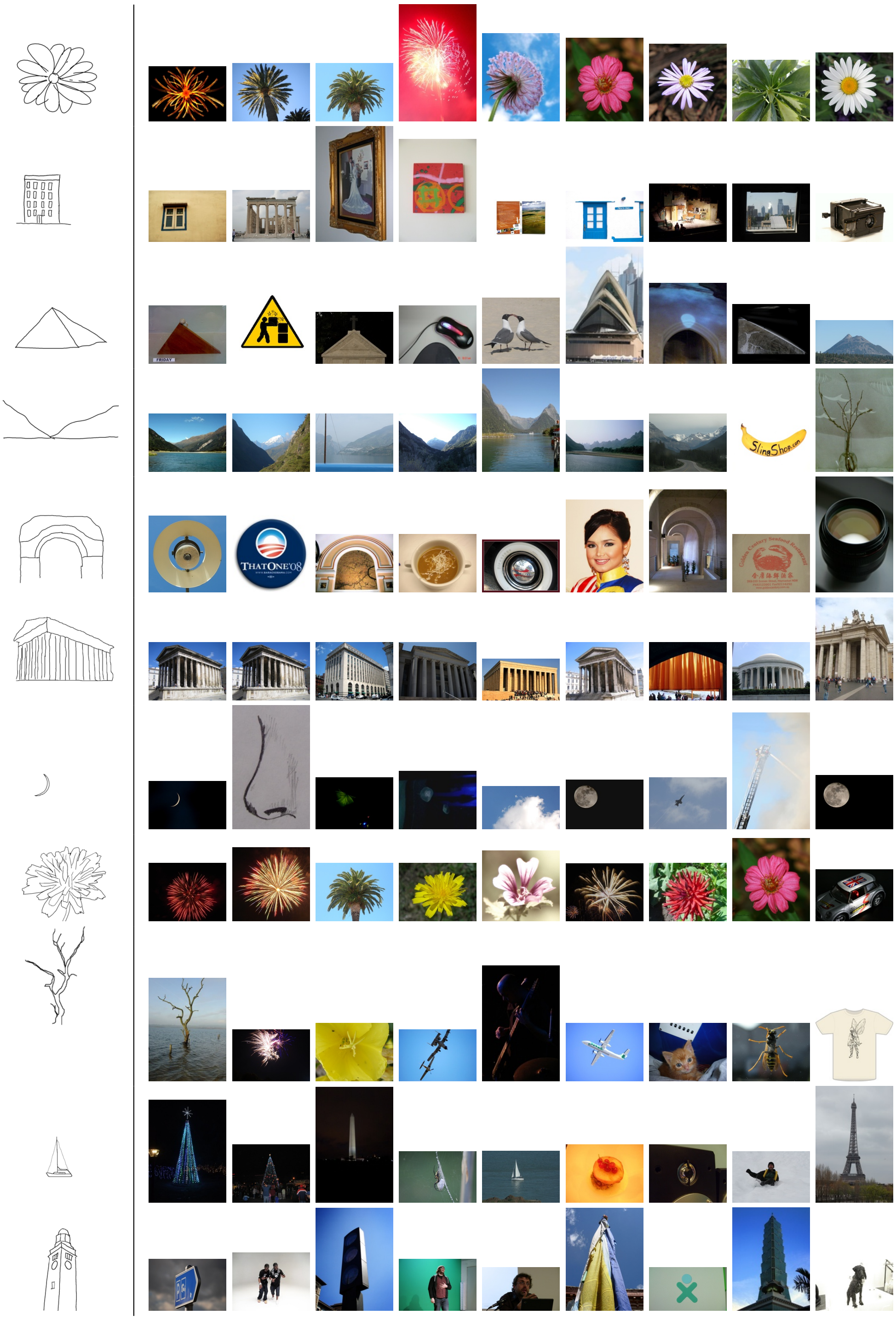

Temple of Heaven
PH-HOG
Semantic PH-HOG
Shanghai Tower
PH-HOG
Semantic PH-HOG
Butterfly
PH-HOG
Semantic PH-HOG
Colosseum
PH-HOG
Semantic PH-HOG
Sunset
PH-HOG
Semantic PH-HOG

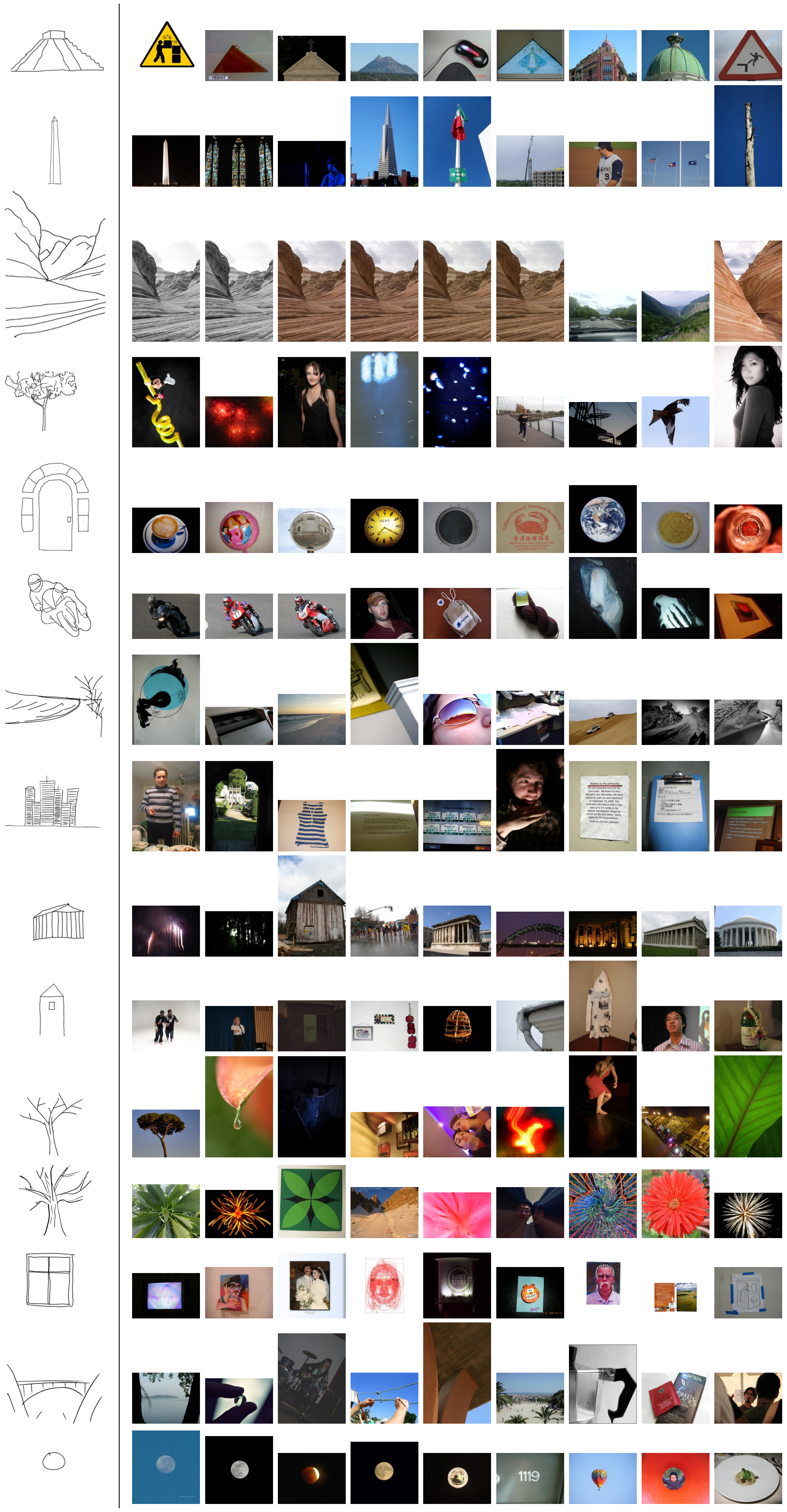

Bird
PH-HOG
Semantic PH-HOG
Rome Antica
PH-HOG
Semantic PH-HOG
Rock Mountain
PH-HOG
Semantic PH-HOG
Tree
PH-HOG
Semantic PH-HOG
Brisbane Bridge
PH-HOG
Semantic PH-HOG

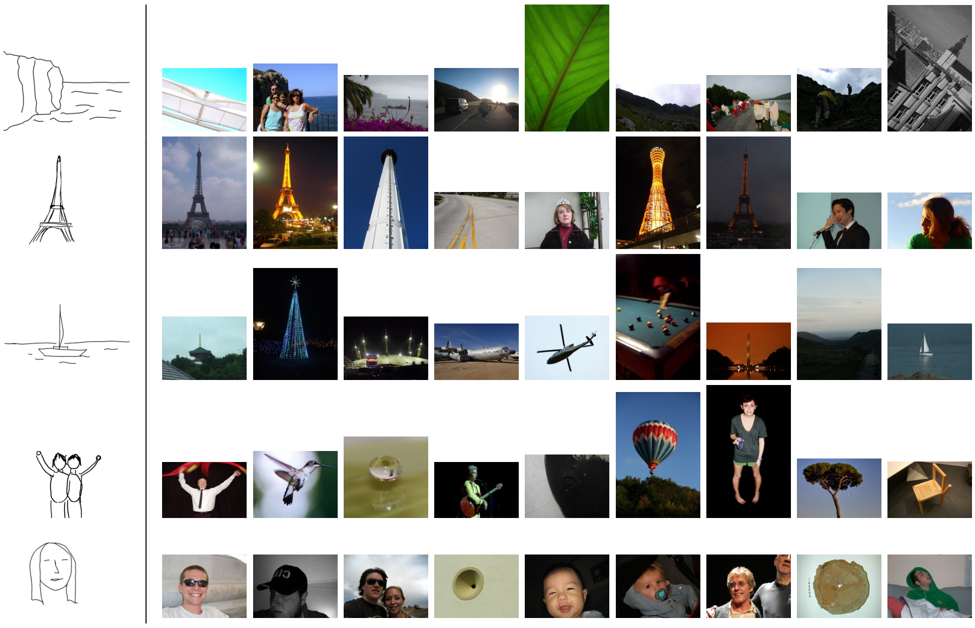

Sydney Opera
PH-HOG
Semantic PH-HOG
To-Ji Temple
PH-HOG
Semantic PH-HOG
Wild Goose Pagoda
PH-HOG
Semantic PH-HOG
Moon
PH-HOG
Semantic PH-HOG

A.3 EitzSBIR

Figure A.3: Top 9 retrieved images for all queries of the EitzSBIR dataset. The PH-HOG method of Chapter 3 is used for retrieval.





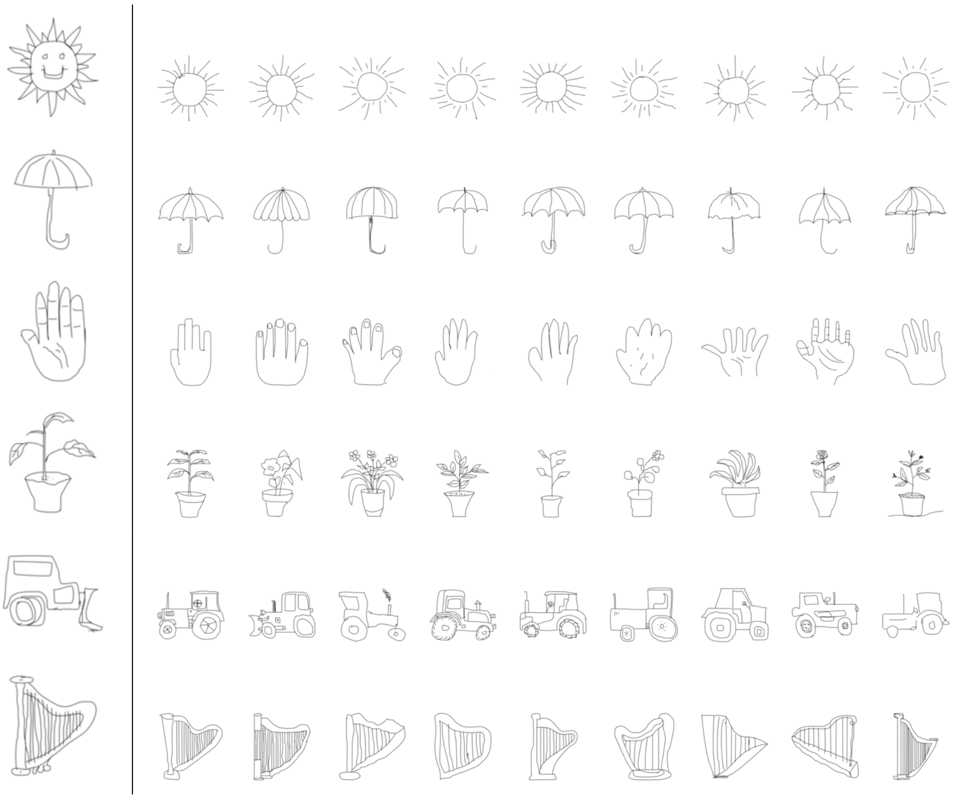


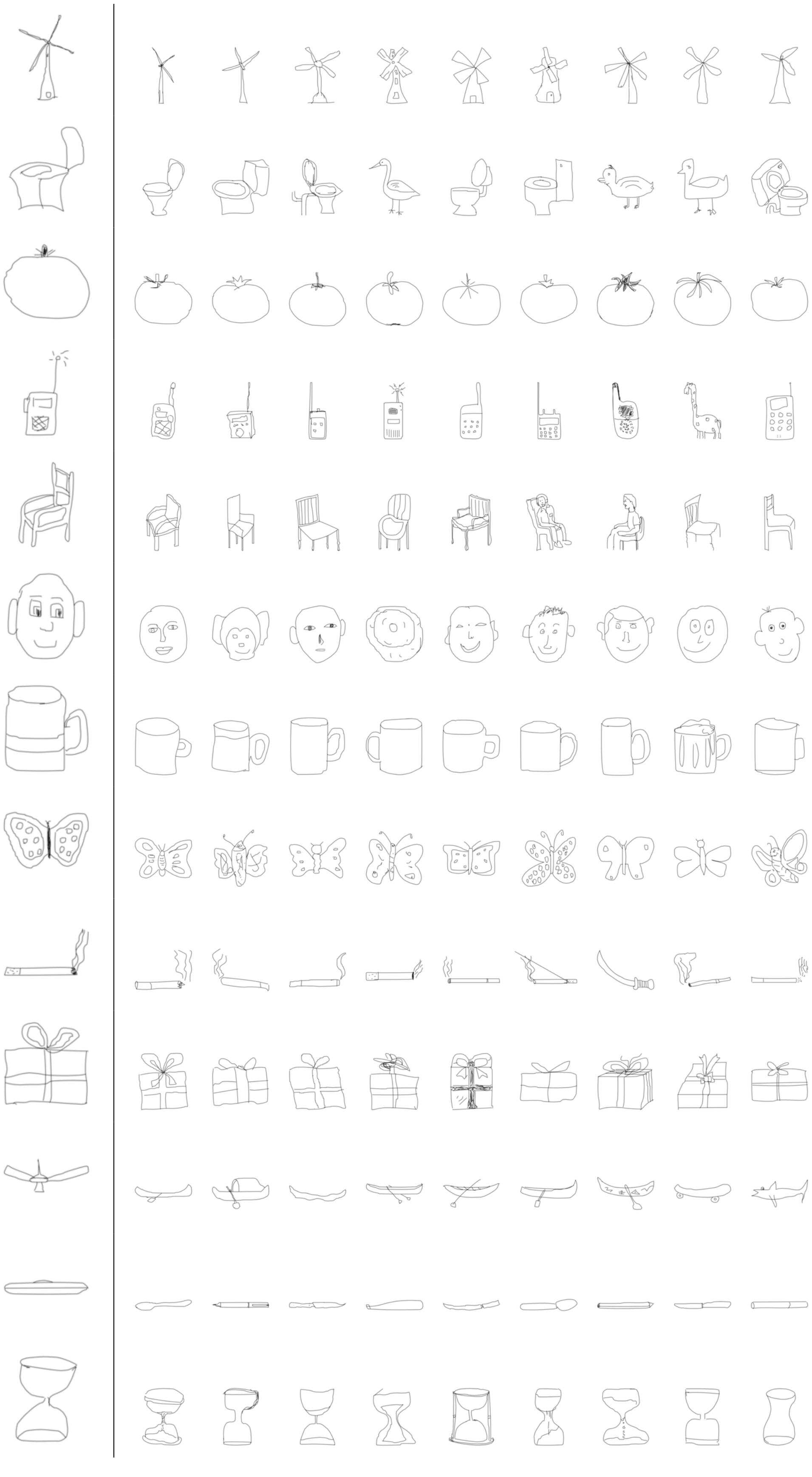
Appendix B
Sketch-to-Sketch Retrieval Results

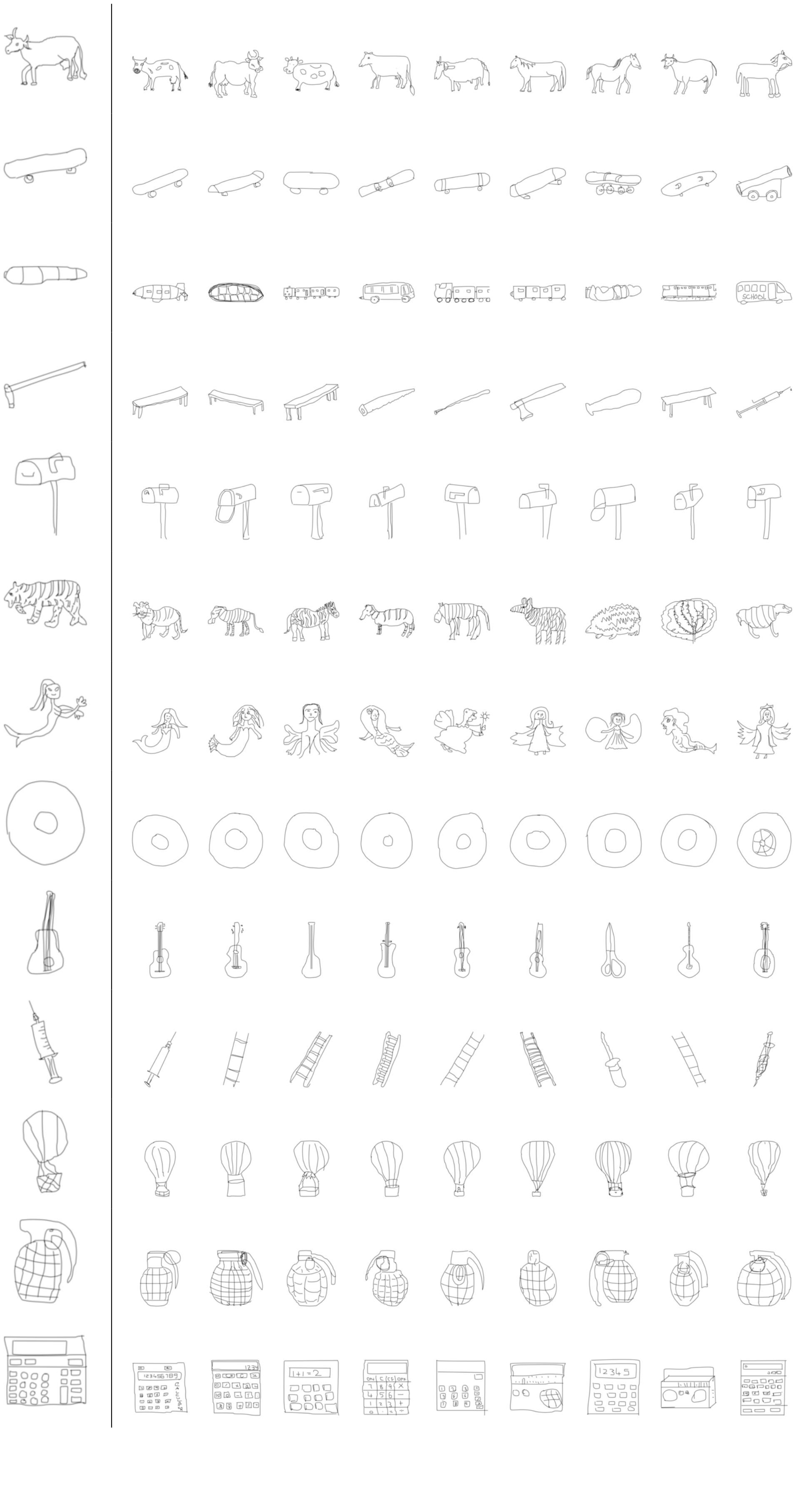
The results following are top ranked sketch retrievals from the EitzSKETCH and Flickr15k datasets.

B.1 EitzSKETCH

Figure B.1: Top 9 retrieved sketches in EitzSKETCH dataset. The SVM-PH-HOG-flip method of Chapter 5 is used for retrieval. EitzSKETCH contains 20,000 sketches evenly allocated in 250 categories.

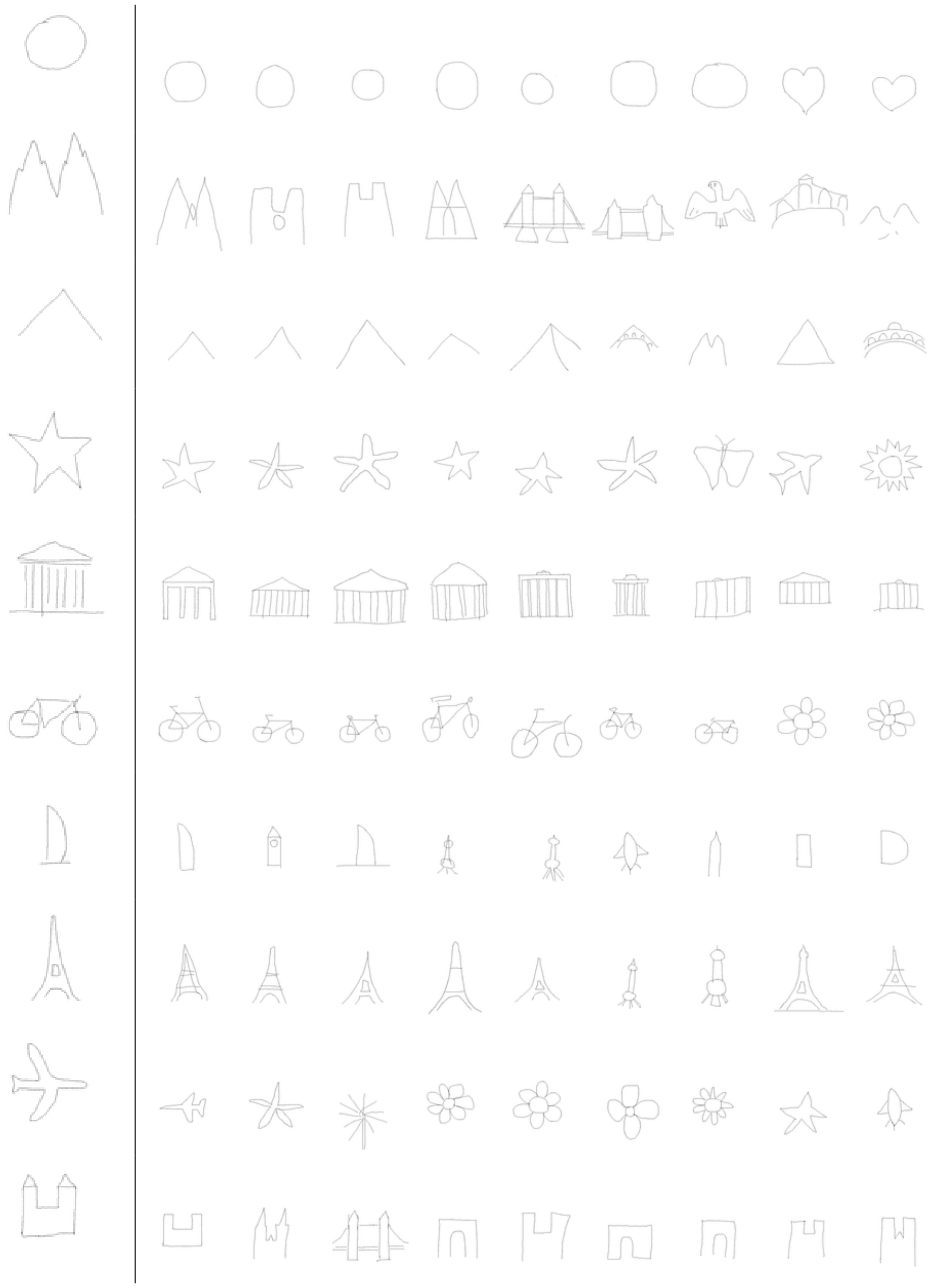


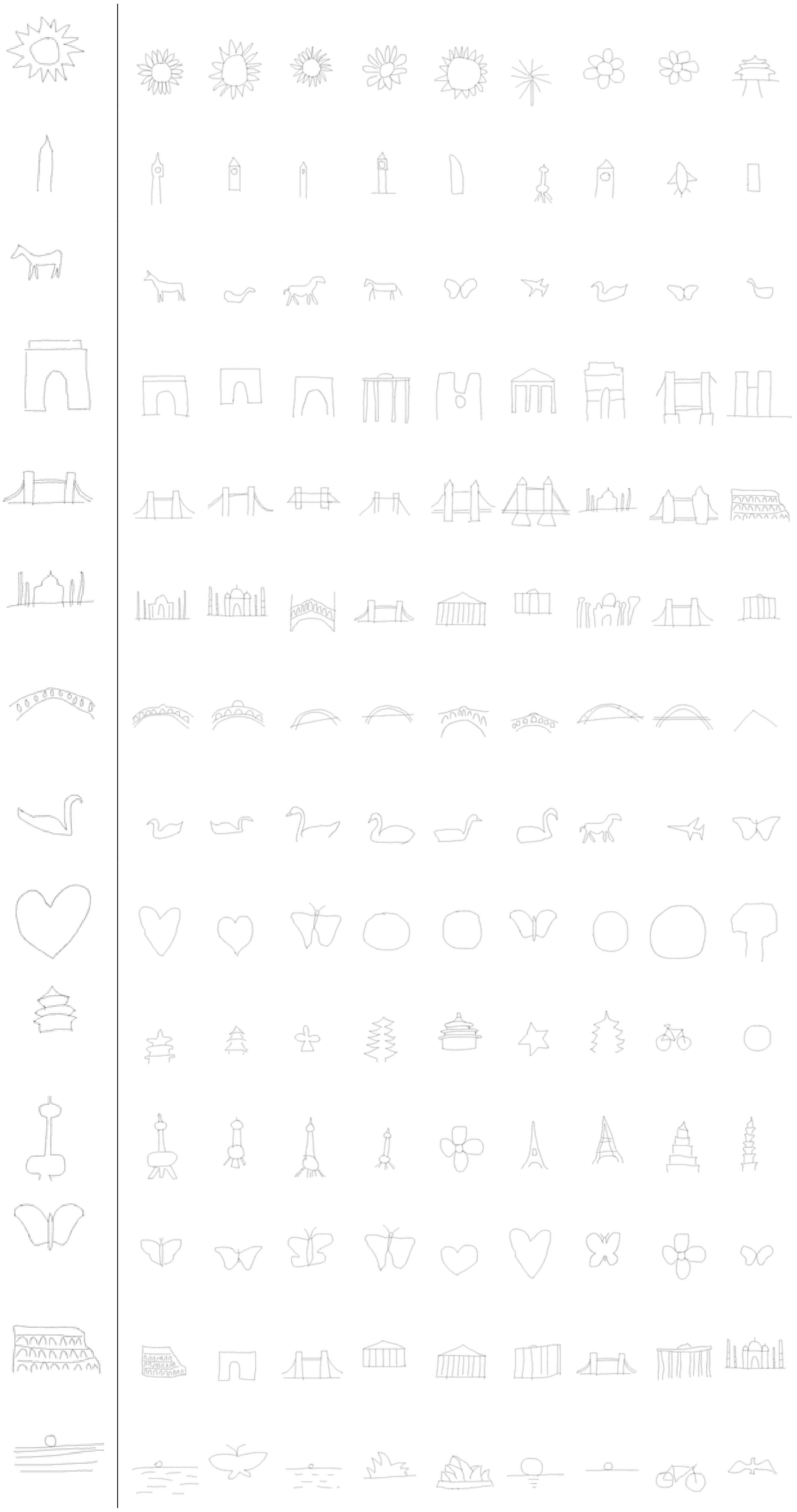


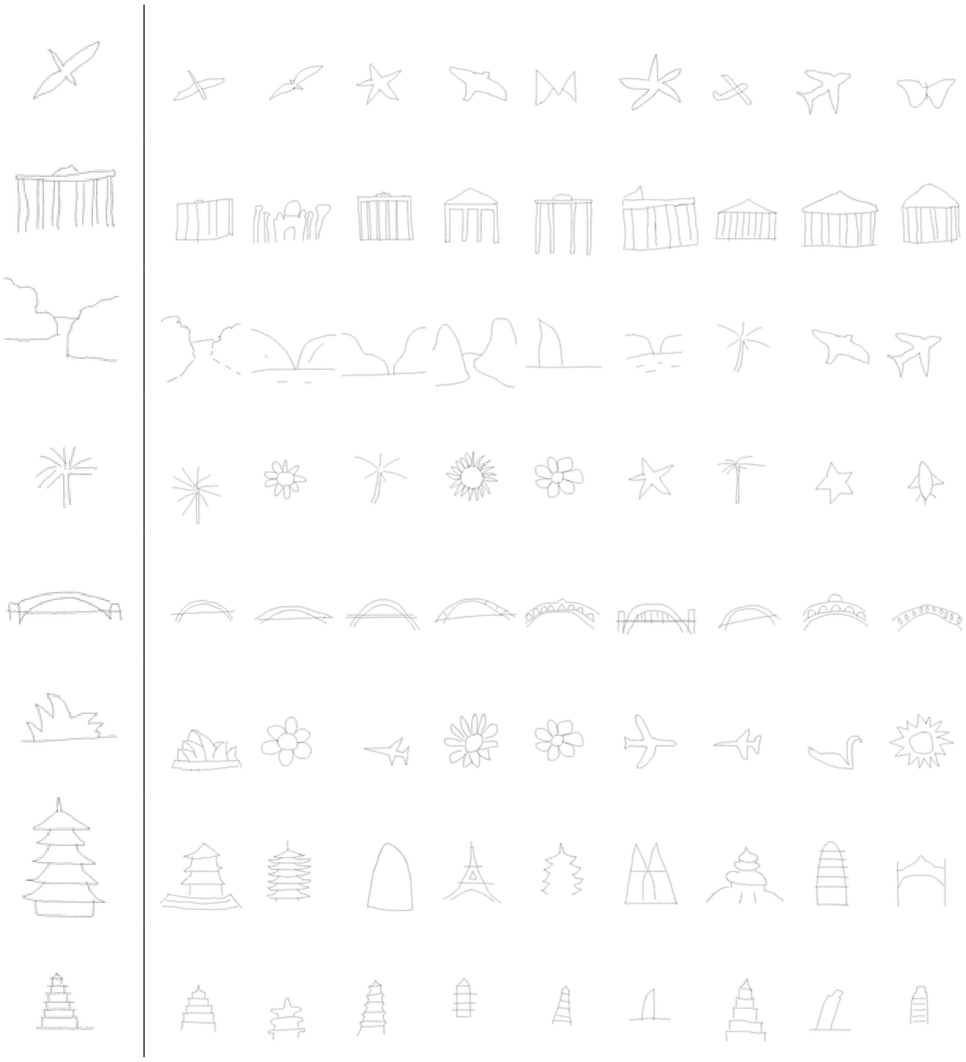

SCHOOL
1+1=2
12345

B.2 Flickr15k

Figure B.2: Top 9 retrieved sketches in Flickr15k sketch set. The SVM-PH-HOG-flip method of Chapter 5 is used for retrieval. Flickr15k contains 330 sketches evenly allocated in 33 categories.







Publications

[1] K. Bozas and E. Izquierdo. Large scale sketch based image retrieval using patch hashing. In *Advances in Visual Computing*, volume 7431 of *Lecture Notes in Computer Science*, pages 210–219. Springer Berlin Heidelberg, 2012.

[2] K. Bozas and E. Izquierdo. Discriminant pairwise local embeddings. In *IEEE International Conference on Multimedia and Expo (ICME)*, Short Paper. IEEE, 2013.

[3] K. Bozas and E. Izquierdo. Horizontal flip-invariant sketch recognition via local patch hashing. In ***Submitted*** *in International Conference on Acoustics, Speech, and Signal Processing*, 2014.

[4] K. Bozas and E. Izquierdo. Scalable sketch based image retrieval via patch hashing. ***Under major revision*** *in ACM Transactions on Multimedia Computing, Communications and Applications*, –:–, 2014.

Bibliography

[5] M. Albanesi and S. Bertoluzza. Human vision model and wavelets for high-quality image compression. In *Image Processing and its Applications, 1995., Fifth International Conference on*, pages 311–315, Jul 1995. (Cited on page 11.)

[6] L. Albertazzi. *Shapes of Forms: From Gestalt Psychology and Phenomenology to Ontology and Mathematics*. Synthese Library. Springer, 1999. (Cited on page 26.)

[7] C. Alvarado and R. Davis. Sketchread: A multi-domain sketch recognition engine. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 23–32, New York, NY, USA, 2004. ACM. (Cited on pages 42 and 94.)

[8] R. Arandjelović and A. Zisserman. All about VLAD. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. (Cited on page 25.)

[9] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, Apr 2002. (Cited on pages 24 and 95.)

[10] Y. Bengio, J.-F. Paiement, and P. Vincent. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *In Advances in Neural Information Processing Systems*, pages 177–184. MIT Press, 2003. (Cited on page 113.)

[11] S. Bileschi and L. Wolf. Image representations beyond histograms of gradients: The role of gestalt descriptors. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007. (Cited on pages 26 and 113.)

[12] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005. (Cited on page 87.)

[13] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, pages 401–408, New York, NY, USA, 2007. ACM. (Cited on page 48.)

[14] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60:327–336, 1998. (Cited on pages 47 and 53.)

[15] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *Communications, IEEE Transactions on*, 31(4):532–540, Apr 1983. (Cited on page 48.)

[16] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2012. (Cited on pages 25, 62 and 99.)

[17] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial-bag-of-features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3352–3359, June 2010. (Cited on page 95.)

[18] Y. Cao, C. Wang, L. Zhang, and L. Zhang. Edgel index for large-scale sketch-based image search. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 761–768, June 2011. (Cited on pages 34, 37, 44 and 47.)

[19] Y. Cao, H. Wang, C. Wang, Z. Li, L. Zhang, and L. Zhang. Mindfinder: Interactive sketch-based image search on millions of images. In *Proceedings of the International Conference on Multimedia*, MM '10, pages 1605–1608, New York, NY, USA, 2010. ACM. (Cited on pages 5 and 37.)

[20] G. Carneiro. Graph-based methods for the automatic annotation and retrieval of art prints. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, pages 32:1–32:8, New York, NY, USA, 2011. ACM. (Cited on page 4.)

[21] A. Chalechale, G. Naghdy, and A. Mertins. Sketch-based image matching using angular partitioning. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(1):28–41, 2005. (Cited on page 30.)

[22] D. Chang, L. Dooley, and J. E. Tuovinen. Gestalt theory in visual screen design - a new look at an old subject. In A. McDougall, J. Murnane, and D. Chambers, editors, *WCCE2001 Australian Topics: Selected Papers from the Seventh World Conference on Computers in Education*, volume 8 of *CRPIT*, pages 5–12, Copenhagen, Denmark, 2002. ACS. (Cited on page 26.)

[23] N. S. Chang and K. S. Fu. Query-by-pictorial-example. In *Proc. IEEE Computer Society's Third Int. Computer Software and Applications Conf COMPSAC 79*, pages 325–330, 1979. (Cited on page 27.)

[24] H.-T. Chen, H.-W. Chang, and T.-L. Liu. Local discriminant embedding and its variants. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 846–853, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on pages 75, 76 and 86.)

[25] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: internet image montage. *ACM Trans. Graph.*, 28:124:1–124:10, December 2009. (Cited on page 35.)

[26] Y. Chen, V. Roussev, I. Richard, G., and Y. Gao. Content-based image retrieval for digital forensics. In M. Pollitt and S. Shenoi, editors, *Advances in Digital Forensics*, volume 194 of *IFIP, The International Federation for Information Processing*, pages 271–282. Springer US, 2005. (Cited on page 4.)

[27] O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *Proceedings of the British Machine Vision Conference*, 2008. (Cited on pages 47, 52, 53 and 55.)

[28] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7432–7437, 2005. (Cited on page 14.)

[29] F. Cole, A. Golovinskiy, A. Limpaecher, H. S. Barros, A. Finkelstein, T. Funkhouser, and S. Rusinkiewicz. Where do people draw lines? *Commun. ACM*, 55(1):107–115, Jan. 2012. (Cited on pages 2 and 5.)

[30] F. Cole, K. Sanik, D. DeCarlo, A. Finkelstein, T. Funkhouser, S. Rusinkiewicz, and M. Singh. How well do line drawings depict shape? *ACM Trans. Graph.*, 28(3):28:1–28:9, July 2009. (Cited on pages 2 and 5.)

[31] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001. (Cited on page 71.)

[32] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Sept. 1995. (Cited on page 97.)

[33] I. J. Cox, M. Miller, T. Minka, T. Papathomas, and P. Yianilos. The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *Image Processing, IEEE Transactions on*, 9(1):20–37, Jan 2000. (Cited on page 13.)

[34] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2005*, volume 1, pages 886–893, 2005. (Cited on pages 12, 32, 47 and 63.)

[35] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):5:1–5:60, May 2008. (Cited on pages 1, 2, 4, 11 and 12.)

[36] J. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20(10):847–856, 1980. (Cited on page 11.)

[37] A. Del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(2):121–132, 1997. (Cited on pages 29, 30 and 94.)

[38] D. L. Donoho. High-dimensional data analysis: the curses and blessings of dimensionality. In *American Mathematical Society Conf. Math Challenges of the 21st Century*. American Mathematical Society, 2000. (Cited on pages 16 and 19.)

[39] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *ACM Transactions on Graphics (Proceedings SIGGRAPH)*, 31(4):44:1–44:10, 2012. (Cited on pages 42, 43, 44, 74, 87, 88, 90, 93, 94, 98, 99, 100 and 103.)

[40] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. A descriptor for large scale image retrieval based on sketched feature lines. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '09, pages 29–36, New York, NY, USA, 2009. ACM. (Cited on pages 32 and 44.)

[41] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. An evaluation of descriptors for large-scale image retrieval from sketched feature lines. *Computers & Graphics*, 34(5):482–498, 2010. (Cited on pages 24, 25, 54, 62 and 65.)

[42] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *Visualization and Computer Graphics, IEEE Transactions on*, 17(11):1624–1636, Nov 2011. (Cited on pages 25, 33, 38, 39, 47, 51, 54, 62, 65, 66 and 95.)

[43] H. Feichtinger and T. Strohmer. *Gabor Analysis and Algorithms: Theory and Applications*. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 1998. (Cited on page 11.)

[44] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(1):36–51, Jan 2008. (Cited on page 25.)

[45] C. Fisher and M. Bornstein. Identification of symmetry: Effects of stimulus orientation and head position. *Perception and Psychophysics*, 32(5):443–448, 1982. (Cited on page 93.)

[46] C. Forlines, D. Wigdor, C. Shen, and R. Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 647–656, New York, NY, USA, 2007. ACM. (Cited on page 2.)

[47] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990. (Cited on page 74.)

[48] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. (Cited on page 43.)

[49] T. Hammond and R. Davis. Tahuti: A geometrical sketch recognition system for uml class diagrams. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM. (Cited on pages 42 and 94.)

[50] X. He. Incremental semi-supervised subspace learning for image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 2–8, New York, NY, USA, 2004. ACM. (Cited on page 75.)

[51] X. He, D. Cai, and J. Han. Learning a maximum margin subspace for image retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, 20(2):189 –201, feb. 2008. (Cited on page 75.)

[52] X. He and P. Niyogi. Locality preserving projections. In *In Advances in Neural Information Processing Systems 16*. MIT Press, 2003. (Cited on pages 75, 76 and 86.)

[53] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):657–662, April 2006. (Cited on page 25.)

[54] K. Hirata and T. Kato. Query by visual example - content based image retrieval. In *Proceedings of the 3rd International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '92, pages 56–71, London, UK, 1992. Springer-Verlag. (Cited on page 27.)

[55] R. Hu, M. Barnard, and J. Collomosse. Gradient field descriptor for sketch based retrieval and localization. In *Proc. 17th IEEE Int Image Processing (ICIP) Conf*, pages 1025–1028, 2010. (Cited on pages 25, 31, 38, 40, 47, 54, 61, 62 and 65.)

[56] R. Hu and J. Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *Comput. Vis. Image Underst.*, 117(7):790–806, July 2013. (Cited on pages 25, 31, 32, 33, 38, 41, 47, 54, 61, 62, 65, 67, 71, 95 and 98.)

[57] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Netw.*, 13(4-5):411–430, May 2000. (Cited on page 75.)

[58] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM. (Cited on page 52.)

[59] A. Ishai, L. G. Ungerleider, A. Martin, and J. V. Haxby. The representation of objects in the human occipital and temporal cortex. *J. Cognitive Neuroscience*, 12(Supplement 2):35–51, Nov. 2000. (Cited on page 73.)

[60] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 277–286, New York, NY, USA, 1995. ACM. (Cited on pages 5 and 28.)

[61] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233 – 1244, 1996. (Cited on pages 5, 13 and 29.)

[62] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3304–3311, jun 2010. (Cited on page 25.)

[63] I. Jolliffe. *Principal Component Analysis.* Springer, 2nd edition, 2002. (Cited on page 74.)

[64] J. P. Jones and L. A. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258, 1987. (Cited on page 11.)

[65] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proc. Tenth IEEE Int. Conf. Computer Vision ICCV 2005*, volume 1, pages 604–610, 2005. (Cited on page 17.)

[66] B. Klare, Z. Li, and A. Jain. Matching forensic sketches to mug shot photos. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):639–646, March 2011. (Cited on page 42.)

[67] B. Ko and H. Byun. Integrated region-based image retrieval using region's spatial relationships. In *Proc. 16th Int Pattern Recognition Conf*, volume 1, pages 196–199, 2002. (Cited on page 19.)

[68] K. Koffka. *Principles of Gestalt psychology.* Harcourt, New York, 1935. (Cited on page 26.)

[69] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. (Cited on page 34.)

[70] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:49–86, 1951. (Cited on page 20.)

[71] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc.*

IEEE Computer Society Conf. Computer Vision and Pattern Recognition, volume 2, pages 2169–2178, 2006. (Cited on pages 15 and 95.)

[72] Y. J. Lee, C. L. Zitnick, and M. F. Cohen. Shadowdraw: real-time user guidance for freehand drawing. *ACM Trans. Graph.*, 30(4):27, 2011. (Cited on pages 6 and 47.)

[73] S. Li, M.-C. Lee, and C.-M. Pun. Complex zernike moments features for shape-based image retrieval. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(1):227–237, Jan 2009. (Cited on page 25.)

[74] Y. Li, Y.-Z. Song, and S. Gong. Sketch recognition by ensemble matching of structured features. In *In British Machine Vision Conference (BMVC)*, 2013. (Cited on pages 42, 43, 94, 95, 100 and 101.)

[75] S. Liang, Z. Sun, and B. Li. Sketch retrieval based on spatial relations. In *Proc. Int Computer Graphics, Imaging and Vision: New Trends Conf*, pages 24–29, 2005. (Cited on page 31.)

[76] Y.-Y. Lin, T.-L. Liu, and H.-T. Chen. Semantic manifold learning for image retrieval. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 249–258, New York, NY, USA, 2005. ACM. (Cited on pages 75 and 79.)

[77] H. Ling and D. Jacobs. Shape classification using the inner-distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):286–299, Feb 2007. (Cited on pages 12 and 95.)

[78] C. Liu, D. Wang, X. Liu, C. Wang, L. Zhang, and B. Zhang. Robust semantic sketch based specific image retrieval. In *Proc. IEEE Int Multimedia and Expo (ICME) Conf*, pages 30–35, 2010. (Cited on pages 5, 35 and 36.)

[79] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262 – 282, 2007. (Cited on page 6.)

[80] Y.-S. Liu, Q. Li, G.-Q. Zheng, K. Ramani, and W. Benjamin. Using diffusion distances for flexible molecular shape comparison. *BMC Bioinformatics*, 11(1):480, 2010. (Cited on page 13.)

[81] D. Lopresti and A. Tomkins. Computing in the ink domain. In K. O. Yuichiro Anzai and H. Mori, editors, *Symbiosis of Human and Artifact - Future Computing and Design for Human-Computer Interaction, Proceedings of the Sixth International Conference on Human-Computer Interaction, (HCI International '95)*, volume 20 of *Advances in Human Factors/Ergonomics*, pages 543 – 548. Elsevier, 1995. (Cited on pages 27 and 28.)

[82] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. (Cited on page 15.)

[83] C. Ma, X. Yang, C. Zhang, X. Ruan, and M.-H. Yang. Sketch retrieval via dense stroke features. In *British Machine Vision Conference*. British Machine Vision Association, 2013. (Cited on pages 42, 43, 94, 95, 100, 101 and 103.)

[84] T. Ma and L. Latecki. From partial shape matching through local deformation to robust global shape similarity for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1441–1448, June 2011. (Cited on page 94.)

[85] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989. (Cited on page 11.)

[86] D. Marr. Early Processing of Visual Information. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 275(942):483–519, 1976. (Cited on page 15.)

[87] D. Marr. Theory of Edge Detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 207(1167):187–217, 1980. (Cited on page 15.)

[88] D. Marr. *Vision*. W. H. Freeman & Company, New York, 1982. (Cited on pages 11 and 15.)

[89] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004. (Cited on pages 54 and 110.)

[90] S. Matusiak, M. Daoudi, T. Blu, and O. Avaro. Sketch-based images database retrieval. In *Proceedings of the 4th International Workshop on Advances in Multimedia Information Systems*, MIS '98, pages 185–191, London, UK, 1998. Springer-Verlag. (Cited on pages 29 and 30.)

[91] F. Mokhtarian and A. K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:789–805, August 1992. (Cited on page 29.)

[92] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, November 2005. (Cited on page 12.)

[93] H. Muller, N. Michoux, D. Bandon, and A. Geissbuhler. A review of content-based image retrieval systems in medical applications, clinical benefits and future directions. *International Journal of Medical Informatics*, 73(1):1 – 23, 2004. (Cited on page 4.)

[94] N. Negroponte. Recent advances in sketch recognition. In *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, AFIPS '73, pages 663–675, New York, NY, USA, 1973. ACM. (Cited on pages 41 and 94.)

[95] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168, 2006. (Cited on page 17.)

[96] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, Jul 2002. (Cited on pages 25, 62 and 99.)

[97] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quenot. Trecvid 2013, an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*. NIST, USA, 2013. (Cited on pages 23 and 61.)

[98] G. Papari and N. Petkov. Adaptive pseudo dilation for gestalt edge grouping and contour detection. *Image Processing, IEEE Transactions on*, 17(10):1950–1962, Oct 2008. (Cited on page 26.)

[99] B. Paulson and T. Hammond. Paleosketch: Accurate primitive sketch recognition and beautification. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI '08, pages 1–10, New York, NY, USA, 2008. ACM. (Cited on page 42.)

[100] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR '07*, pages 1–8, 2007. (Cited on page 17.)

[101] R. Plamondon and S. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):63–84, Jan 2000. (Cited on page 31.)

[102] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, May 2008. (Cited on page 43.)

[103] J. Saavedra and B. Bustos. Sketch-based image retrieval using keyshapes. *Multimedia Tools and Applications*, pages 1–30, 2013. (Cited on pages 33 and 62.)

[104] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. (Cited on page 18.)

[105] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138–142, Dec. (Cited on page 85.)

[106] T. Sawada and Z. Pizlo. Detection of skewed symmetry. *Journal of Vision*, 8(5), 2008. (Cited on page 108.)

[107] B. Sayim and P. Cavanagh. What line drawings reveal about the visual brain. *Frontiers in Human Neuroscience*, 5(118), 2011. (Cited on page 73.)

[108] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2001. (Cited on pages 79 and 81.)

[109] T. M. Sezgin and R. Davis. Hmm-based efficient sketch recognition. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, IUI '05, pages 281–283, New York, NY, USA, 2005. ACM. (Cited on pages 42 and 94.)

[110] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *IEEE Conference on Computer Vision and Pattern Recognition 2007 (CVPR'07)*, June 2007. (Cited on page 24.)

[111] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Shape Modeling International*, June 2004. (Cited on page 43.)

[112] M. Shneier and M. Abdel-Mottaleb. Exploiting the jpeg compression scheme for image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8):849–853, Aug 1996. (Cited on page 13.)

[113] T. Sikora. The mpeg-7 visual standard for content description-an overview. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):696–702, Jun 2001. (Cited on page 32.)

[114] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proc. Ninth IEEE Int Computer Vision Conf*, pages 1470–1477, 2003. (Cited on pages 16 and 95.)

[115] sketch. Oxford English Dictionary Online, 2nd edition. `http://www.oed.com/`, July 2003. (Cited on page 5.)

[116] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, Dec. 2000. (Cited on pages 1, 4 and 13.)

[117] M. Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of 23rd International Conference on*

Machine Learning, pages 905–912. ACM, 2006. (Cited on pages 75, 79 and 86.)

[118] X. Sun, C. Wang, C. Xu, and L. Zhang. Indexing billions of images for sketch-based retrieval. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 233–242, New York, NY, USA, 2013. ACM. (Cited on pages 34 and 113.)

[119] Z. Sun, C. Wang, L. Zhang, and L. Zhang. Free hand-drawn sketch segmentation. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, volume 7572 of *Lecture Notes in Computer Science*, pages 626–639. Springer Berlin Heidelberg, 2012. (Cited on page 31.)

[120] V. Takala, T. Ahonen, and M. Pietikäinen. Block-based methods for image retrieval using local binary patterns. In *SCIA*, pages 882–891, 2005. (Cited on page 15.)

[121] J. B. Tenenbaum, V. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000. (Cited on pages 19 and 75.)

[122] C. Theoharatos, N. Laskaris, G. Economou, and S. Fotopoulos. A generic scheme for color image retrieval based on the multivariate wald-wolfowitz test. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):808–819, June 2005. (Cited on page 13.)

[123] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999. (Cited on page 75.)

[124] L. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review, 2008. (Cited on page 74.)

[125] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008. (Cited on page 61.)

[126] J. Wagemans. Detection of visual symmetries. *Spatial Vision*, 9:9–32, 1995. (Cited on page 93.)

[127] D. B. Walther, B. Chai, E. Caddigan, D. M. Beck, and L. Fei-Fei. Simple line drawings suffice for functional mri decoding of natural scene categories. *Proceedings of the National Academy of Sciences*, 108(23):9661–9666, 2011. (Cited on page 73.)

[128] X. Wang, T. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39, Sept 2009. (Cited on page 25.)

[129] X. Wang and X. Tang. Face photo-sketch synthesis and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(11):1955–1967, Nov 2009. (Cited on page 42.)

[130] P. Wenderoth. The salience of vertical symmetry. *Perception*, 23:221–236, 1994. (Cited on page 93.)

[131] J. Wu, W. chian Tan, J. M. Rehg, and B. Taskar. Efficient and effective visual codebook generation using additive kernels. *Georgia Institute of Technology*, 2011. (Cited on page 17.)

[132] J. Wu and L. Zhang. Gestalt saliency: Salient region detection based on gestalt principles. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 181–185, Sept 2013. (Cited on page 26.)

[133] S. Xiang, F. Nie, C. Zhang, and C. Zhang. Nonlinear dimensionality reduction with local spline embedding. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1285–1298, Sept 2009. (Cited on page 113.)

[134] J. Yang, D. Zhang, A. Frangi, and J.-Y. Yang. Two-dimensional pca: a new approach to appearance-based face representation and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(1):131–137, Jan 2004. (Cited on page 91.)

[135] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2008*, pages 1–8, 2008. (Cited on page 17.)

[136] J. Ye, R. Janardan, and Q. Li. Gpca: An efficient dimension reduction scheme for image compression and retrieval. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 354–363, New York, NY, USA, 2004. ACM. (Cited on page 91.)

[137] S. M. Yoon and A. Kuijper. Query-by-sketch based image retrieval using diffusion tensor fields. In *Image Processing Theory Tools and Applications (IPTA), 2010 2nd International Conference on*, pages 343 –348, july 2010. (Cited on page 33.)

[138] W. Yu, X. Teng, and C. Liu. Face recognition using discriminant locality preserving projections. *Image and Vision Computing*, 24(3):239 – 248, 2006. (Cited on page 75.)

[139] C. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *Computers, IEEE Transactions on*, C-20(1):68–86, Jan 1971. (Cited on page 26.)

[140] D. Zhang and G. Lu. Shape-based image retrieval using generic fourier descriptor. *Signal Processing: Image Communication*, 17(10):825 – 848, 2002. (Cited on page 25.)

[141] W. Zhang, X. Wang, and X. Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 513–520, June 2011. (Cited on page 42.)

[142] X. Zhou, K. Yu, T. Zhang, T. S. Huang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV (5)*, pages 141–154, 2010. (Cited on page 15.)