Sketch Retrieval via Dense Stroke Features

Chao Ma[1]
chaoma@sjtu.edu.cn
Xiaokang Yang[1]
xkyang@sjtu.edu.cn
Chongyang Zhang[1]
sunny_zhang@sjtu.edu.cn
Xiang Ruan[2]
gen@omm.ncl.omron.co.jp
Ming-Hsuan Yang[3]
mhyang@ucmerced.edu

[1] Institute of Image Communication and Network Engineering
Shanghai Jiao Tong University
Shanghai, China

[2] Omron Coporation
Kyoto, Japan

[3] Electrical Engineering and Computer Science
University of California, Merced
California, USA

Abstract

Sketch retrieval aims at retrieving most similar sketches from a large database based on one hand-drawn query. Successful retrieval hinges on an effective representation of sketch images and an efficient search method. In this paper, we propose a representation scheme which takes sketch strokes into account with local features, thereby facilitating efficient retrieval with codebooks. Stroke features are detected via densely sampled points on stroke lines from which local gradients are further enhanced and described by a quantized histogram of gradients. A codebook is organized in a hierarchical vocabulary tree, which maintains structural information of visual words and enables efficient retrieval in sub-linear time. Experimental results on three data sets demonstrate the merits of the proposed algorithm for effective and efficient sketch retrieval.

1 Introduction

Sketch-based image retrieval, which deals with the problem of retrieving similar images in a large database based on a hand-drawn query, has received considerable attention in recent years [3, 8, 11, 13, 18]. Sketches, originated from the contour or skeleton of an object, has long been proposed as an effective intermediate representation for describing essential shape information of objects [14] with numerous applications. In this work, we define a *sketch* as a collection of hand-drawn stroke lines, which can be close or open as shown in Figure 1, to describe an object of interest.

As sketches are hand-drawn with different styles to represent objects, sketch retrieval is challenging due to several factors. First, there exist large intra-class differences, as a result of experiential and cognitive differences among individuals, e.g., giraffe sketches drawn by two individuals are likely to be significantly different in terms of their shapes (See Figure 1). Second, there exist small inter-class differences, due to their loss of visual information (i.e., texture and appearance), e.g., the sketch of an apple may look similar to that of an orange. Therefore, the key issue for sketch retrieval lies in an effective scheme to represent sketches that takes both inter-class and intra-class differences into consideration.

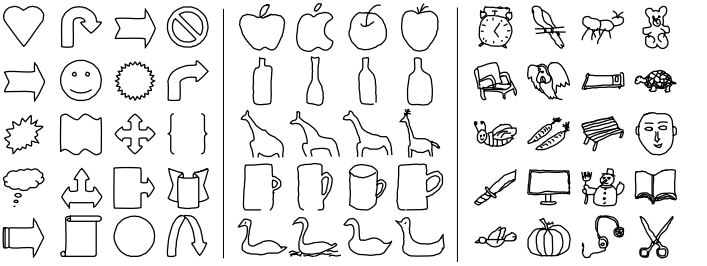


Figure 1: Sketch images. Office icon library (left), hand-drawn ETHZ shape [8] (middle) and TU Berlin sketch [7] data set (right).

In this paper, we propose an algorithm for efficient and effective *sketch retrieval* with focus on *sketch-to-sketch* rather than *sketch-to-image* retrieval based on one hand-drawn query. We represent a sketch image via local features that are distributed evenly on stroke lines. For efficient query and match, local features of sketches are described by a quantized histogram of gradients and stored hierarchically in a vocabulary tree. Each sketch is then represented by the index of tree nodes instead of storing all of them in a long vector. We show that a straightforward bag-of-words approach with local corner features for sketch retrieval is not effective. Instead, the proposed algorithm focuses on stroke lines of sketches with crucial corner points and evenly sampled points, which performs more robustly for sketch retrieval. In addition, the proposed representation scheme facilitates integration with other spatial kernels [10] to capture spatial information of local features. We evaluate the proposed algorithm on three large data sets of hand-drawn sketches. Experimental results on three data sets with more than 20,000 sketch images show that the proposed algorithm performs favorably against state-of-the-art methods in terms of retrieval accuracy and execution time.

2 Related Work and Problem Context

Early works on primal sketch focus on representation schemes based on primitive features such as edges as well as curves, and considerably less attention is paid to sketch retrieval until recent years. Ferrari et al. construct the ETHZ shape database [8] and k-adjacent segments to detect objects in images based on hand-drawn examples where image edges are partitioned into contour segments and organized in a chain. In addition, directed chamfer matching [11], partial shape matching [5, 13], latent support vector machines [18] have also been applied to sketch-based object detection and localization. However, these methods mainly focus on retrieving objects in images using one good query sketch (i.e., sketch to images), and they are less effective for complex sketch retrieval (i.e., sketch to sketches), especially when there exist large intra-class differences and small inter-class differences.

To retrieve object images in a large database, descriptors based on edge pixels are proposed for indexing and matching sketches [3]. As the underlying matching mechanism is based on chamfer distance, the proposed algorithm based on edgel descriptors are less effective in describing complex sketches. In [6], Eitz et al. leverage the bag-of-words formulation with SIFT descriptors for sketch-based image retrieval (i.e., sketch to images). Hu et al. [9] also present a bag-of-words approach based on multiple descriptors and histogram of image gradients for sketch-based image retrieval. Both these methods use grid-based sampling methods to locate local features and the K-means clustering algorithm to learn codebooks for following indexing scheme. In contrast, we focus more on selecting the most representative local features that are evenly distributed on strokes including crucial corner points and de-

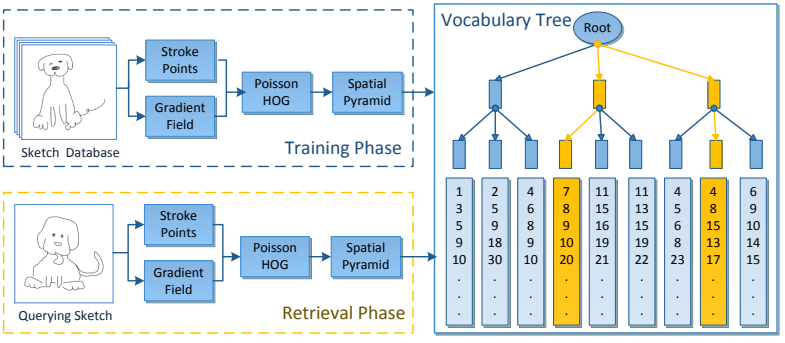


Figure 2: Main steps of the proposed local feature based sketch retrieval: the vocabulary tree is constructed offline for retrieval. Inverted training image identities are indexed below the tree leaves. Each query sketch is retrieved by the hits of the training identities (orange).

scribing local features via a coarsely quantized histogram of gradients. We note that existing methods focus on sketch *classification* [3, 6, 9] (i.e., sketch to images with object types) or detection (i.e., sketch to images with object locations) [3, 8, 11, 13, 18], rather than *retrieval* (i.e., sketch to sketches with ranking) based one query.

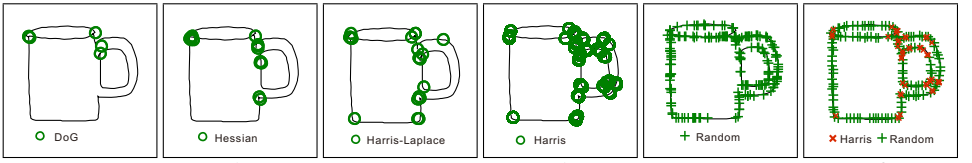
3 Proposed Algorithm

We present the proposed algorithm for sketch retrieval via stroke features, which consists of three components: selecting most representative stroke points, describing stroke features using a quantized histogram of gradients, and representing sketch images using a hierarchical vocabulary tree for matching. Figure 2 shows the main steps of the proposed method. In the training phase, we extract all the local features of training sketches and store these local features in a hierarchical vocabulary tree similar to [16], where each training sketch is indexed by the frequency of tree nodes to which its local features belong. In the retrieval phase, each query sketch is represented via its stroke features and the same tree. For efficient retrieval, this vocabulary tree can be easily integrated with the inverted indexing method, which tallies the identities (labels) of training sketch images that have local features belonging to each node. Therefore, retrieval can be carried out by counting the hit frequency between a query sketch and the inverted list with identities of training images.

3.1 Densely Sampled Stroke Points

In this work, we use local stroke features to represent sketches instead of contour segments [8, 11, 13, 18]. While several key point detectors such as difference of Gaussian (DoG) [12], Hessian operator [1], and Harris-Laplace detector [15] are available for locating local features, they are designed mainly for finding salient points. Salient key points are sparsely distributed over an image, and it is of great importance to capture their spatial relationship for better object representation (instead of simple bag-of-words approaches). Grid-based as well as random sampling methods have also been proposed to locate local features. As sketch images consists of strokes with no textural information, it is essential to select the most representative stroke points for local features. Since corners and end pixels of strokes always encode important geometric information of a sketch, they are used as anchor positions for dense sampling to encode shape information properly.

In addition, sketches with complex shapes are not compactly represented by grid-based



(a) (b) (c) (d) (e) (f)

Figure 3: Different key point sampling results. (a) DoG points [17]. (b) Hessian points [1]. (c) Harris-Laplace points [15]. (d) Harris corners. (e) Randomly selected points. (f) Proposed stroke points. (a)-(c) are sparse salient point detection methods usually used in bag-of-words approaches. The number of Harris corners is more than (a)-(c). The proposed stroke points (f) are distributed more evenly than the randomly sampled ones (e).

sampling very well. Thus, we propose to extract evenly distributed stroke points based on anchor corners. For sketch retrieval, each image is normalized to a canonical size and thus the Harris corner detector is adopted for computational efficiency. We compute the corner response of a sketch image I by:

$$E(x,y) = \sum_{u,v} w(u,v)[I(x+u,y+v) - I(x,y)]^2, \tag{1}$$

where $w(u,v) = \exp(-(u^2+v^2)/\sigma^2)$ is the Gaussian kernel. We use these corners as anchors and add a number of points (e.g., twice the desired number of points) randomly sampled on the strokes. We next remove those points, other than the anchors, that are too close to each other, in order to spare the points evenly (i.e., points with large spreads are preferred). This greedy pruning method performs well in practice in terms of speed and distribution. The proposed stroke point detection method is summarized in Algorithm 1 and key point detection results of different methods are shown in Figure 3.

Algorithm 1 Dense sampling of stroke points.

Input:

Sketch image $I(x,y) = \{0,1\}$, where $\forall_{(x,y)} I(x,y) = 1$ denotes all stroke points.

Output:

N stroke points.

1: Compute Harris response using (1) and select $N/4$ corners, whose location denoted by Ω_h.
2: Randomly select $2N$ points from I, where $I(x,y) = 1$, whose locations denoted by Ω_r.
3: $\Omega = \Omega_h \cup \Omega_r$.
4: Compute pairwise Euclidean distance D_E of Ω, and set $D_E(\Omega_h, :) = \infty$.
5: For each point in Ω, remove its nearest neighbor (only in Ω_r) until $|\Omega| = N$.
6: **return** Ω.

3.2 Histogram of Dense Gradients from Stroke Points

The histogram of gradient (HOG) descriptor is widely used for object detection [4]. In the HOG formulation, an image is divided into grid cells where gradient orientations are indexed into a histogram of d bins (weighted by its magnitude). To further improve the performance of HOG by using local geometric information of each cell, Hu et al. [9] propose to use the Poisson equation [17] for smoothing the gradient field. For a sketch image I, a sparse field from the gradient of edge pixels is computed, $G[x,y] \mapsto \arctan\left(\frac{\delta I}{\delta x} / \frac{\delta I}{\delta y}\right)$ for $I(x,y) = 1$ (i.e.,

edge point). A dense field $\mathcal{G}_\Lambda$ over image coordinates $\Lambda \in \mathcal{R}^2$ is obtained by minimizing the following energy function:

$$\arg\min_{\mathcal{G}} \int\int_{\Lambda} (\nabla\mathcal{G} - G)^2 \quad \text{s.t. } \mathcal{G}|_{\delta\Lambda} = G|_{\delta\Lambda}, \tag{2}$$

where ∇ is the gradient operator and $\delta\Lambda$ denotes the boundary condition. This equation can be solved by a discrete Poisson solver with the Dirichlet boundary conditions [17]. The dense gradient field $\mathcal{G}$ captures more edge information than the sparse gradient field G, thereby better representing sketch images which consist of open and close stroke lines.

As sketch are always drawn casually by hand with great intra-class differences, it is essential to consider such variations in sketch retrieval. Thus, we adapt the HOG formulation by first discarding the central distance weights (i.e., we do not compute the distance voting to each grid cell center) and then computing a histogram with coarse quantized orientations (e.g., $d = 4$) with an anti-alias function (5). These two strategies can successfully suppress intra-class difference locally area, and help achieve better retrieval performance (See Section 4). We summarize the proposed Poisson-based HOG (PHOG) feature descriptor as follows:

Step 1: Compute the dense gradient field $\mathcal{G}$ from a sketch image by solving (2) with a Laplace of Gaussian operator $\Delta\mathcal{G}$ [17],

$$\Delta\mathcal{G}(x,y) = -\frac{1}{\pi\sigma^2}[1 - \frac{x^2+y^2}{2\sigma^2}]e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{3}$$

The dense gradient field $\mathcal{G}$ can be approximated by convolution through $\Delta\mathcal{G}$ and sketch Image I:

$$\mathcal{G}(x,y) = \sum_{u,v} I(x,y)\Delta\mathcal{G}(x-u, y-v). \tag{4}$$

Step 2: Select a local square patch around a stroke point $p_i (i \in \Omega)$ in $\mathcal{G}$. The patch area is denoted by S_l.
Step 3: Divide the patch S_l into $n \times n$ grid cells evenly.
Step 4: Compute the histogram of gradients with d (d=4) orientations in each cell using:

$$\cos(x - \alpha_i)^3 > 0, \tag{5}$$

where x denotes the gradient orientation weighted by its magnitude, and α_i denotes the i-th bin center.

3.3 Hierarchical Vocabulary Tree

In general, sketch retrieval methods depend heavily on how features can be efficiently indexed in a codebook. In this work, we use a hierarchical tree to train a codebook in spirit similar to the vocabulary tree [16]. This tree effectively retains structure information of visual words, which accelerates not only the indexing process but also sketch retrieval in conjunction with the inverted training identity indexing scheme.

A hierarchical tree can be defined by two parameters, K and L, where K is the number of cluster centers and L denotes the depth of tree. We iteratively use the K-means clustering algorithm at each level until the tree grows to the pre-defined level L. The nodes of the tree represent the cluster centers, and each local PHOG feature descriptor of a sketch image can be effectively represented by a path from the root node to a leaf node (See Figure 2).

Thus, the histogram of all the paths of local PHOG descriptors is the signature of a sketch. Similar to the inverted indexing scheme, we assign each leaf node a list with image identities (labels) which contain the same PHOG feature descriptor (See Figure 2). Sketches can be easily retrieved by counting the hit frequencies between local features of a query and training images instead of retaining all the feature descriptors. Similar to [16], we compute the weight of each node by the average entropy (i.e., a node becomes less distinctive when more training images are included) as follows:

$$w_i = \ln \frac{N}{n_i}, \tag{6}$$

where N is the total number of training images and n_i denotes the number of training images which have local PHOG descriptors belonging to node i. Finally, we compute the sketch descriptor (representation) h_s for a sketch image by

$$h_s = w \otimes h, \tag{7}$$

where $\otimes$ denotes the dot product, and h is the histogram of paths in the hierarchical tree with respect to all PHOG feature descriptors.

3.4 Distance Metric

Given a query, we find the closest sketches via the sketch descriptor h_s based on their distance using the χ^2 kernel [19, 20]. Given a sketch pair, I_q and I_r, and their corresponding sketch descriptors h_q as well as h_r, we compute their distance as follows:

$$D(h_q, h_r) = \frac{1}{2} \sum_{i=1}^{n} \frac{[h_q(i) - h_r(i)]^2}{h_q(i) + h_r(i)}. \tag{8}$$

Thus, for each query sketch I_q, we use D_Λ to denote the distance matrix of I_q to a subset Λ of the training sketch images, in which each training image has local PHOG features in the same bin as the query sketch in the hierarchical tree. Thus, distance matrix computed on the subset Λ and retrieval can be performed in sub-linear time. The rank k retrievals are based on:

$$rank(k) = \arg\min_{1,\ldots,k} D_\Lambda. \tag{9}$$

4 Experimental Results

In this section, we present the experimental results of the proposed algorithm with comparisons to alternative approaches for sketch retrieval on three data sets. In all experiments, we select 800 (N=800) stroke points and set the area of local feature patch S_l as 1/8 of the input sketch image. In each patch, we compute the Poisson-based HOG descriptor in 4×4 grid cells with 4 (d=4) orientations. Thus, each stroke point corresponds to one 64 dimensional feature vector. To capture spatial information of local features, we use two-level spatial pyramid kernel. That is, we partition each sketch image into two parts according to the centroid of sampled stroke points horizontally and vertically respectively. Then we index each part with the learned hierarchical tree and concatenate the four histograms as the final representation of a sketch image. This spatial kernel is effective as it retains most structure information but not at the expense of increasing intra-class differences. We use the cumulative matching

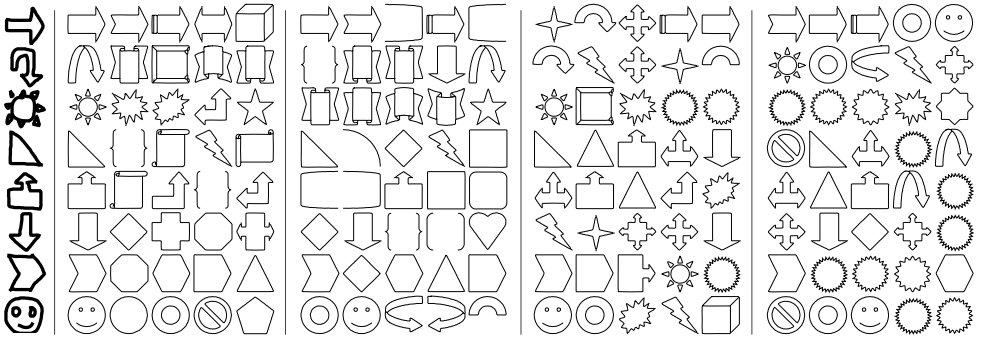


Figure 4: Rank 5 office icon retrieval results. From left to right: sampled hand-drawn queries, sketch retrievals by the proposed algorithm, SPM, SCM and DCM respectively.

Table 1: Cumulative matching accuracy on ETHZ Shape data set from rank 1-5 (%).

	rank 1	rank 2	rank 3	rank 4	rank 5
DCM [3, 11]	95.14	97.33	97.71	98.19	98.38
SCM [2]	92.86	96.38	97.81	98.38	98.57
SPM [10]	96.19	97.52	98.10	98.29	98.38
PHOG-K	96.48	97.52	97.90	98.10	98.48
KHOG-T	96.48	97.81	98.19	98.48	98.67
PHOG-T	96.67	98.38	**98.67**	**98.67**	**99.14**
PHOG-A	**97.14**	**98.48**	**98.67**	**98.67**	99.05

accuracy (CMA) as the evaluation criteria. The CMA curve represents the top n retrieval images containing the same (correct) sketch category as the query. In addition, we also define the cumulative best matching accuracy (CBMA) curve as that the correct retrieval sketches that account for the most of the top n retrieved sketches.

We evaluate the proposed algorithm with three alternative approaches for sketch retrieval using the directed chamfer matching (DCM) method [3, 11] which has been shown to be effective for large scale sketch-based image retrieval problem [3], the shape context matching (SCM) method [2] which is effective for shape matching and object recognition based on features in clean backgrounds, and the spatial pyramid matching (SPM) method [10] which has been shown to be effective for object categorization based on dense SIFT descriptors. For each of these three methods, we empirically determine the parameters for the best performance. More results are available in the supplementary document.

Office Icon Library: We collect the 78 office icons used for flow chart creation and create 38 hand-drawn query sketches as the test set. Some icons in the training set are rather similar with minor shape difference, e.g., arrows shown in Figure 4. In the meanwhile, the hand-drawn queries contain large shape variation when compared with the counterparts in the training set. We train the hierarchical tree of depth 4 with 6 clusters ($K = 6$ and $L = 4$). On this data set, the retrieval results are subjective to human subjects (similar to the setup in [3]), and we present the raw retrieval results instead of the CMA and CBMA curves. The results of the proposed algorithm with comparisons to the alternative SPM, SCM and DCM methods are presented in Figure 4, and the remaining 30 retrieval results are available in the supplementary material.

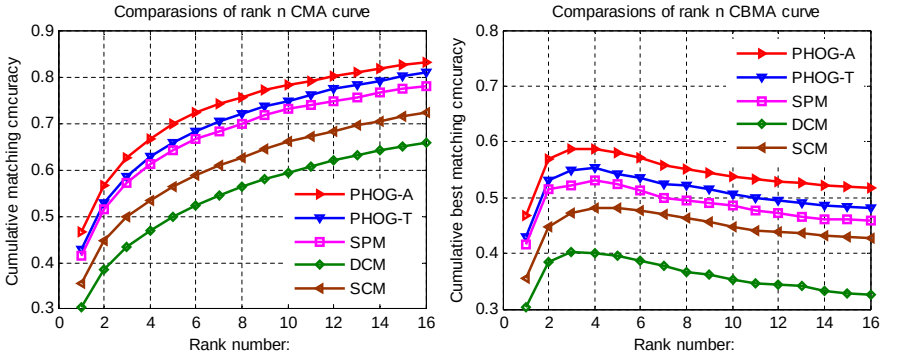


Figure 5: Rank n CMA and CBMA curves with the TU Berlin sketch data set.

ETHZ Shape Data Set: We use 5 hand-drawn shapes from the ETHZ data set [8] (i.e., apple, giraffe, swan, bottle and mug shown in Figure 1) with 1,050 sketches drawn by 50 different painters (210 sketches per category) for experiments. We train the vocabulary tree ($K = 5$ and $L = 4$) and choose each sketch as query sketch, and only compare the CMA of rank 1 to 5 retrieval results as the training set is large.

In addition to comparisons with the SPM, SCM and DCM methods, in Table 1 we present several experimental results using variants of the algorithmic components including Poisson-based HOG and K-means (PHOG-K, where $k = 500$), Poisson-based HOG and a hierarchical tree (PHOG-T), and proposed stroke points as key points with HOG descriptor as well as a hierarchical tree (KHOG-T). The proposed sketch retrieval algorithm (PHOG-A) consists of using the Poisson-based HOG based from stroke features, hierarchical tree and spatial pyramid kernel. Note that the PHOG-A method differs from the PHOG-K alternative by the use of a hierarchical tree. On the other hand, the PHOG-A method differs from the PHOG-T alternative by the use of a spatial pyramid.

TU Berlin Sketch Data Set: Eitz et al. [7] collect 20,000 sketches representing 250 different objects (each object has 80 different sketches) using crowd sourcing. In [7], the goal is to analyze and compare hand-drawn sketch recognition abilities between humans and computers. The bag-of-words approach with dense SIFT features and a codebook with K-means clustering is used for sketch representation, the same as the above-mentioned SPM method, with one difference that the spatial pyramid kernel is not used as it is shown to gain little performance improvement [7]. We choose last 20 sketches per category as the querying set, and train a vocabulary tree ($K = 9$ and $L = 4$) for experiments. Figure 5 shows the CMA and CBMA curves of all evaluated methods. We present some retrieved sketches by the proposed PHOG-A in Figure 6 and more results are available in the supplementary document.

Discussion: The DCM and SCM methods are based on holistic representations which capture global spatial information of sketch images. For sketches with simple shapes, their layout information becomes more important, and thus these two holistic methods are effective for sketch retrieval. As shown in the experiments with the office icon and ETHZ data sets, these methods achieve comparable performance to the proposed algorithm. However, they do not perform well for more complex sketches (e.g., the ones in the TU Berlin data set). Another issue with the SCM method is the high computational load, which cannot be applied to large-scale sketch retrieval.

The proposed algorithm and the SPM methods are based on local features with a code-

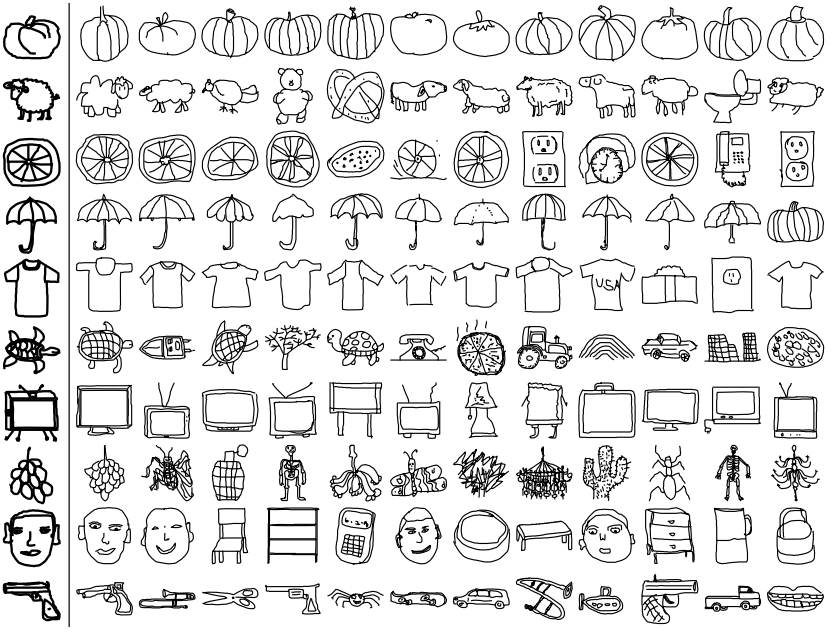


Figure 6: Sample retrieval results on the TU Berlin sketch database. The left column are the querying sketches and the right column are the corresponding rank 12 retrievals.

book indexing scheme. In the SPM method, local features are dense SIFT descriptors and the codebook is trained with the K-means clustering algorithm. The SPM method has been shown to be effective for representing object images with rich appearance and texture information [10]. However, hand-drawn sketches consist of strokes with no texture information. We sample local features on stroke lines instead of uniformly sampling over the image surface, and use Poisson-based HOG descriptors with coarse quantized histograms. In Table 1, the PHOG-K method performs better (rank 1) than the SPM method which demonstrates that the sampled stroke features are indeed more effective for sketch representation. On the other hand, PHOG-T method over KHOG-T method demonstrates the effectiveness of coarse quantization (PHOG) in accounting for large shape variation of sketches.

We also note it is of great importance to properly capture spatial information of local features for sketch retrieval although Eitz et al. show that the use of spatial layout information does not improve the performance in sketch recognition [7] based on a bag-of-words approach. Experimental results show that better accuracy can be achieved by the two-level spatial pyramid kernel especially for rank 1 tests. Table 1 and Figure 5 show that the PHOG-A method outperforms the PHOG-T method due to the use of spatial information. In addition, the hierarchical tree facilitates retaining data structural information of visual words. In Table 1, the results of the PHOG-T method over the PHOG-K approach demonstrate that the use of a vocabulary tree helps to improve the accuracy of sketch retrieval.

Overall, the proposed PHOG-A algorithm performs favorably against other alternatives. Implemented in MATLAB with 3.1 GHz processors and 4 GB memory, each retrieval takes less than 0.01 second on the ETHZ data set which is nearly the same as PHOG-K, KHOG-T, PHOG-T and SPM methods and an order magnitude faster than DCM (0.21) and SCM (0.56) approaches. For the TU Berlin data set, it takes less than 0.1 second for each retrieval as opposed to other methods (PHOG-T: 0.08, SPM: 0.06, DCM: 1.04, and SCM: 2.57 seconds).

5 Conclusion

In this paper, we propose a novel representation of hand-drawn sketch based on stroke features. Local features are detected via densely sampled stroke points and described by a quantized histogram of gradients interpolated by a Poisson formulation. A codebook is organized in a hierarchical tree, which maintains structural information of visual words and enables efficient retrieval in sub-linear time. Experimental results on three test data sets demonstrate the proposed algorithm performs favorably against other alternatives for sketch retrieval.

Acknowledgements

C. Ma, X. Yang and C. Zhang are supported by the NSFC Grant (60932006, 61025005, 61129001, 61221001), 973 Program (2010CB721406), and STCSM (13511504500). M.-H. Yang is supported in part by the NSF CAREER Grant 1149783 and NSF IIS Grant 1152576.

References

[1] H. Bay, T. Tuytelaars, and L.V. Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

[3] Y. Cao, C. Wang, L. Zhang, and L. Zhang. Edgel index for large-scale sketch-based image search. In *CVPR*, pages 761–768, 2011.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[5] M. Donoser, H. Riemenschneider, and H. Bischof. Efficient partial matching of outer contours. In *ACCV*, 2009.

[6] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *TVCG*, 17(11):1624–1636, 2011.

[7] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *SIGGRAPH*, pages 44:1–44:10, 2012.

[8] V. Ferrari, T. Tuytelaars, and L.V. Gool. Object detection by contour segment networks. In *ECCV*, pages 14–28, 2006.

[9] R. Hu, M. Barnard, and J. Collomosse. Gradient field descriptor for sketch based retrieval and localization. In *ICIP*, pages 1025–1028, 2010.

[10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[11] M. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *CVPR*, pages 1696–1703, 2010.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60: 91–110, 2004.

[13] T. Ma and L.J. Latecki. From partial shape matching through local deformation to robust global shape similarity for object detection. In *CVPR*, pages 1441–1448, 2011.

[14] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, 1982.

[15] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, pages 63–86, 2004.

[16] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

[17] P. Perez, M. Gangnet, and A. Blake. Poisson image editing. *SIGGRAPH*, pages 313–318, 2003.

[18] P. Srinivasan, Q. Zhu, and J. Shi. Many-to-one contour matching for describing and discriminating object shape. In *CVPR*, pages 1673–1680, 2010.

[19] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[20] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, pages 3539–3546, 2010.