Image and Vision Computing 46 (2016) 64–73

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Sketch retrieval via local dense stroke features☆

Chao Ma [a,*,1], Xiaokang Yang [a], Chongyang Zhang [a], Xiang Ruan [b], Ming-Hsuan Yang [c]

[a] Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai 200240, PR China
[b] Omron Corporation, Kyoto, Japan
[c] Electrical Engineering and Computer Science, University of California, Merced, CA 95344, United States

ARTICLE INFO

Article history:
Received 13 August 2014
Received in revised form 11 July 2015
Accepted 27 November 2015
Available online 22 January 2016

Keywords:
Sketch retrieval
Stroke feature
Poisson based histogram of orientation

ABSTRACT

Sketch retrieval aims at retrieving the 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 with crucial corners as anchor points, from which local gradients are 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 from a large database based on a hand-drawn query, has received considerable attention in recent years [1–8]. Sketches, originating from the contour or skeleton of an object, have long been proposed as an effective intermediate representation for describing essential shape information of objects [9] with numerous applications. In this work, we define a *sketch* as a collection of hand-drawn stroke lines, which can be closed or open as shown in Fig. 1, to describe an object of interest.

As sketches are hand-drawn with free 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 shapes (see Fig. 1). Second, there exist small inter-class differences, due to loss of visual details (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.

Recent work on sketch retrieval mainly focuses on retrieving natural images (*sketch-to-image*) on large database [10,3,4,5,7,6,8,11,12], while considerably less attention is paid to retrieving sketches (*sketch-to-sketch*). With the increasing capacity of the sketch dataset (e.g., the TU Berlin dataset [8] is created by crowd sourcing with 20,000 sketches of 250 object categories), it is of great importance to resolve the problem of retrieving sketches on large scale database. Due to large intra-class and small inter-class differences between sketches, it is ineffective to retrieve them simply using shape retrieval algorithms [13–16], where shapes are derived from natural objects with regular and simple contours (rather than hand-drawn). Unlike these simple shapes, sketches are hand-drawn with significant disparities on the number and length of stroke lines even for the same class.

In this paper, we propose an algorithm for efficient and effective sketch matching with focus on *sketch-to-sketch* rather than *sketch-to-image* retrieval based on one hand-drawn query. We represent a sketch image by local features that are distributed evenly on stroke lines. For efficient query and match, local features of a sketch image are described by a quantized histogram of gradients and stored hierarchically in a vocabulary tree. Each sketch image is then represented by the index of tree nodes instead of storing all of the local feature descriptors 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 a sketch image 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 [17] to capture spatial information of local features and usage of inverted index on tree nodes to speed up quantization of local features. We evaluate the proposed algorithm on three large data sets of hand-drawn sketches. Experimental results on these 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.

☆ This paper has been recommended for acceptance by Seong-Whan Lee, PhD.

* Corresponding author.
E-mail addresses: chaoma@sjtu.edu.cn (C. Ma), xkyang@sjtu.edu.cn (X. Yang), sunny_zhang@sjtu.edu.cn (C. Zhang), gen@omm.ncl.omron.co.jp (X. Ruan), mhyang@ucmerced.edu (M.-H. Yang).

[1] C. Ma was sponsored by China Scholarship Council and took a two-year study in University of California at Merced.

http://dx.doi.org/10.1016/j.imavis.2015.11.007
0262-8856/

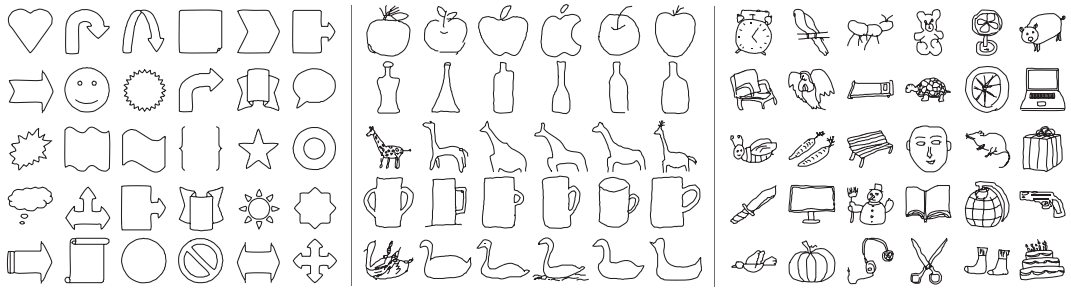

Fig. 1. Sketch images. From left to right: Office icon library, hand-drawn ETHZ shape [1] (apple, bottle, giraffe, mug and swan) and TU Berlin sketch [8] data set. Notice that the office icons have minor inter-class differences (e.g., the right arrows) while the ETHZ shapes have large intra-class differences (e.g., the giraffes and swans).

Compared to early results of this work [18], we show effectiveness of the proposed local features which use edge information of foreground and background regions to better represent sketches (Section 3.2); we analyze the histogram distribution of the number of the stroke points each sketch contains to set the optimal number of dense stroke features (Section 4.1); and we present more experimental results and discuss the application scenarios of the proposed sketch retrieval method (Figs. 9 and 11).

2. Related work and problem context

Existing methods on primal sketches focus on representation schemes based on primitive features such as edges as well as curves. In [19,20], sketches are stored in the form of multiple strokes and retrieved by using the shape of each stroke and the spatial relationship between them. When sketches are simple close-formed hand-drawings, the Fourier transformed boundary is used as shape feature for representation [21,22]. By applying the 2-D Fourier transform on a polar shape image, an adapted Fourier descriptor is proposed to represent sketches in terms of contour [23]. In [10], Rui et al. review sketch-based image retrieval with focus on the contour-based and region-based representation schemes. However, due to simplicity of representation for sketches, these methods are ineffective to represent and index complex sketches of a large scale database.

In recent years, much attention has been paid to sketch retrieval due to its wide applications for intelligent human computer interfaces. Ferrari et al. construct the ETHZ shape database [1] 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 chains. In addition, shape modeling [24], Chamfer matching [2], partial shape matching [11,5], and discriminative latent shape models [3] have 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 thus they are less effective for complex sketch retrieval (i.e., sketch to sketches) when there exist large intra-class and small inter-class differences.

To retrieve object images from a large database, feature descriptors are commonly extracted for indexing and matching sketches. The descriptor-based representations in the literature can be roughly categorized as either global or local. In [25], Chalechale et al. exploit angular and spatial distributions of edge pixels to represent holistic features, which is similar to the shape context information [26]. Shao et al. [27] instead extract key points along stroke lines to account for shape difference between sketches. In [4], Cao et al. propose an edge descriptor for sketch based image retrieval. As the underlying matching method is based on Chamfer distance with focus on global geometric information, the proposed edge-based descriptors are less effective in describing complex sketches. On the other hand, local feature based methods are more robust to represent complex sketches. In [6,7], 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. [12] 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 describing local features via a coarsely quantized histogram of gradients. We note that existing methods focus on sketch *classification* [4,6,12,25] (i.e., sketch to images with object types) or detection (i.e., sketch to images with object locations) [24,1–5], rather than *retrieval* (i.e., sketch to sketches with ranking) based on one query.

3. Proposed algorithm

We present the proposed algorithm for sketch retrieval via stroke features, which consists of three components: selecting the most representative stroke points, describing stroke features using a quantized histogram of gradients, and representing sketch images using a hierarchical vocabulary tree for matching. Fig. 2 shows the main steps of the proposed method. In the training phase, we extract all the local features of sketch images and store these local features in a hierarchical vocabulary tree similar to [28], where each 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 vocabulary tree. For efficient retrieval, this vocabulary tree can be easily integrated with an inverted indexing method, which tallies the identities (labels) of training sketch images that have local features belonging to each node. Retrieval can thus 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 [1,2,3,5]. While several key point detectors such as difference of Gaussian (DoG) [29], Hessian operator [30] and Harris–Laplace detector [31] can be used for locating local features, they are designed mainly for finding salient points. As salient key points are usually sparsely distributed over an image, it is of great importance to capture their spatial relationship for better object representation (rather than simple bag-of-words approaches). Grid-based as well as random sampling methods have also been proposed to locate local features [32,33]. As sketch images consist 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 sampling well (e.g, local features detected by grid points may capture few stroke points). 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 the Harris

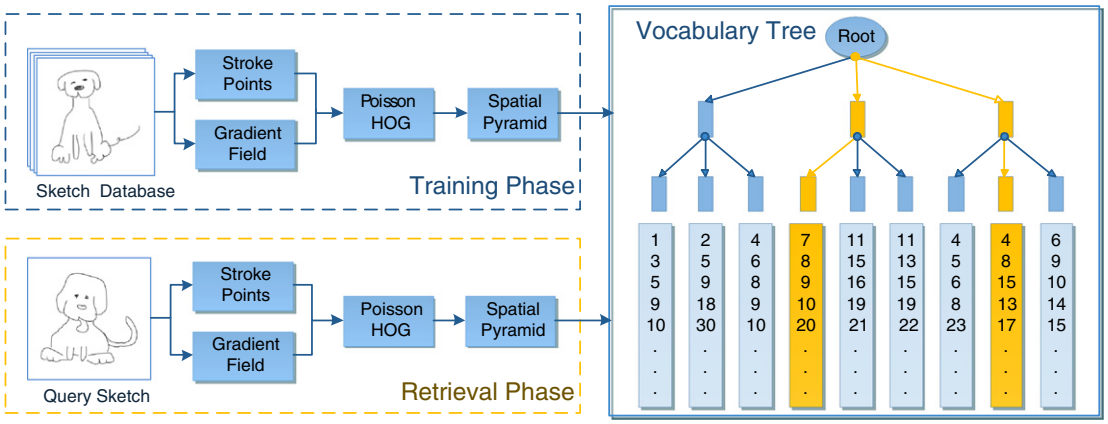

Fig. 2. Main steps of the proposed local feature based sketch retrieval: the vocabulary tree is constructed offline for retrieval. Inverted training image identities are indexed below the tree leaves. Each query sketch is retrieved by the hits of the training identities (orange).

corner detector [34] 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 random points, other than the anchors, that are too close to each other, in order to spread 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. In addition, the fact that not all corner points can be consistently detected from sketch images (due to large intra-class differences between hand-drawn sketches) should be taken into account. Let N be the number of stroke points densely sampled by Algorithm 1. The number of anchor points is $N/4$ and the number of randomly sampled points is $3N/4$.

Algorithm 1. Dense sampling of stroke points.

Input:
Sketch image $I(x; y) = \{0,1\}$, where $\forall_{(x,y)} I(x,y) = 1$ and $|I|$ denote all stroke points and their total number respectively.
Output:
N stroke points.
1: If $|I| \le N$, return the whole stroke points.
2: Compute Harris response using (1) and sort it with descent order to select $N/4$ corners, whose location denoted by Ω_r
3: Randomly select $2N$ stroke points from I, whose locations denoted by Ω_r.
4: $\Omega = \Omega h \cup \Omega_r$.
5: Compute pairwise Euclidean distance D_E of Ω, and set $D_E(\Omega_h,:) = \infty$
6: For each point in Ω, remove its nearest neighbor (only in Ω_r) using rowindex according to the distance matrix D_E until $|\Omega| = N$.
7: Return Ω

Fig. 3 shows one example of the detected key points by different methods. Note that if only salient key points are used to describe sketches (Fig. 3(a)–(d)), the local features are not sufficient to represent sketches well. On the other hand, the feature representation based on random sampling (Fig. 3(e)) is also ineffective due to uneven point distribution. The proposed representation based on anchor points and dense sampling (Fig. 3(f)) is more reliable for locating local features.

3.2. Histogram of dense gradients from stroke points

The histogram of oriented gradients (HOG) descriptor is widely used for object detection [35]. In the HOG formulation, an image is divided into grid cells where gradient orientations are indexed with 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. [12] use the Poisson equation [36] to smooth the gradient field (more details are discussed in [37]). For a sketch image I, a sparse field from the gradients at the edge pixels is computed, i.e., $G[x,y]\mapsto \arctan(\frac{\delta I}{\delta x}/\frac{\delta I}{\delta y})$ for every edge point $I(x,y)=1$. A dense field $\mathcal{G}_\Lambda$ over image coordinates $\Lambda \in \mathcal{R}^2$ is obtained by minimizing the following energy function:

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

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 [36]. The dense gradient field $\mathcal{G}$ captures more edge information (see Fig. 4) than the sparse gradient field G, thereby representing sketch images with more discriminative strength.

As sketches are typically drawn casually by hand with large intra-class differences, it is essential to take such variations into account for 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 coarsely quantized orientations (e.g., $d=4$) with an anti-alias function (5). These two modifications successfully suppress intra-class differences, 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 $\Delta\mathcal{G}$ [36]

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

The dense gradient field $\mathcal{G}$ can be approximated by the convolution of a sketch Image I with $\Delta\mathcal{G}$

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

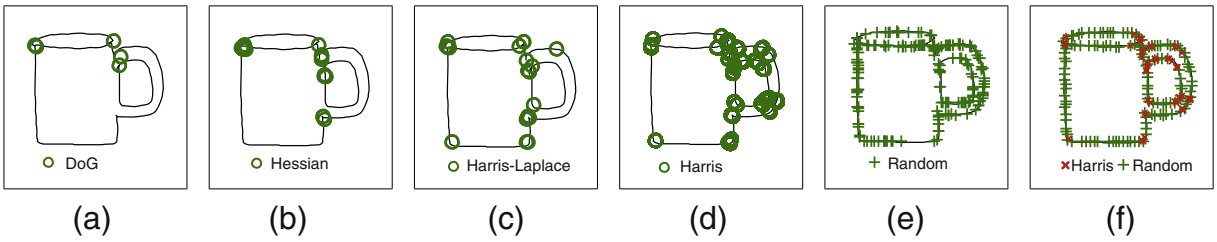

Fig. 3. Different key point sampling results. (a) DoG points [29]. (b) Hessian points [30]. (c) Harris–Laplace points [31]. (d) Harris corners. (e) Randomly sampled points. (f) Proposed dense stroke points. (a)–(c) are sparse salient point detection methods usually used in bag-of-words approaches. The number of Harris corners is more than (a)–(c). The proposed dense stroke points (f) are distributed more evenly than the randomly sampled ones (e).

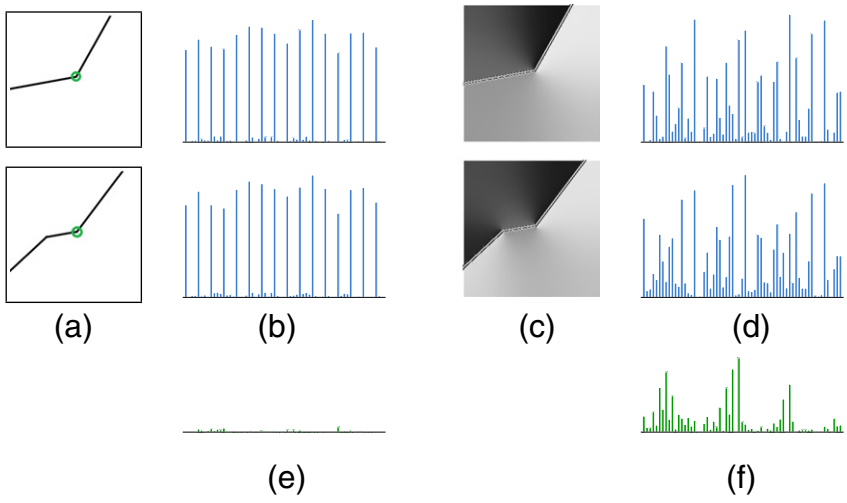

Fig. 4. Histogram of sparse and dense gradients on two patches. (a) Two local patches with different shapes. (b) Histogram of sparse gradients with 4×4 grid cells in 4 directions. (c) Patches in dense gradient field G. (d) Histogram of dense gradients with 4×4 grid cells in 4 directions. (e) Absolute difference of two patches with representation (b). (f) Absolute difference of two patches with representation (d). The patches in (a) can be differentiated with the proposed dense gradients.

Step 2. Select a local square patch around a stroke point p_i in $\mathcal{G}(i \in \Omega)$. 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 weighted gradients with d ($d = 4$) orientations in each cell. For each gradient with orientation θ, the weight factor on its magnitude is:

$$f(\cos(\theta - \alpha_i)^3), s.t. f(t) = \begin{cases} 0, & t < 0 \\ t, & t \geq 0 \end{cases} \quad (5)$$

where α_i ($i = 1,2,\ldots,d$) denotes the i-th orientation bin center. The difference between the anti-alias and hard quantization histogram is shown in Fig. 5.

Fig. 6(a)–(d) shows four examples of the HOG descriptors (second column), dense gradient field $\mathcal{G}$ (third column) as well as the proposed PHOG descriptors (fourth column) for sketch representations. Sketches in the first row (blue) are queries; sketches in the second (green) and third (red) rows represent rank 1 retrievals using the proposed PHOG and HOG descriptors respectively. The proposed PHOG descriptors effectively improve the retrieval precision as a dense gradient field $\mathcal{G}$ captures the global contour of sketches in the foreground and background regions by solving (2). Each local patch S_l encodes richer edge information, which facilitates strengthening the discriminative strength of the PHOG descriptors.

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 [28]. 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 the number of cluster centers K and the depth of tree L. 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

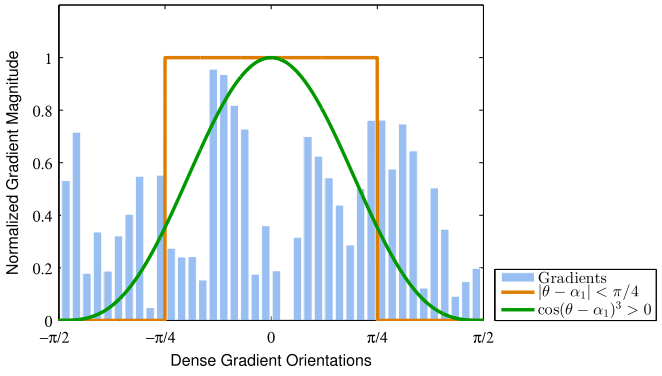

Fig. 5. Histogram of dense gradients using the anti-alias (green) and hard quantization (orange) in the first orientation ($\alpha_1 = 0$). Note that the slope of anti-alias function falls much more slowly than that of hard quantization, which helps suppress the intra-class difference.

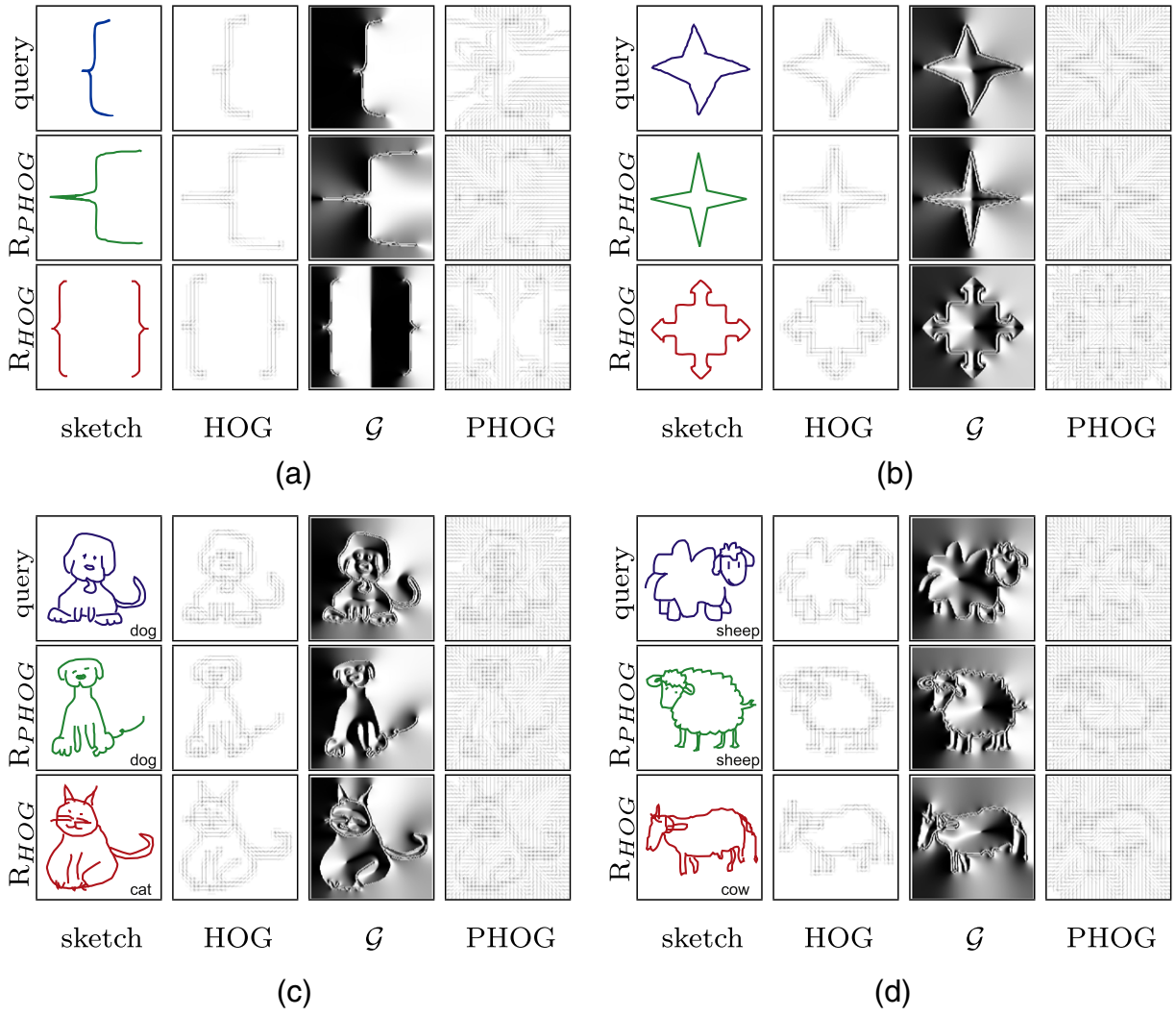

Fig. 6. Visualization results. In (a)–(d), following each sketch (first column) are HOG descriptors (second column), dense gradient field $\mathcal{G}$ (third column) and the proposed PHOG descriptors (fourth column). Sketches in the first row (blue) are queries; sketches in the second (green) and third (red) rows represent rank 1 retrievals using the proposed PHOG and HOG descriptors, denoted by P_{HOG} and R_{HoG} respectively. The dense gradient field $\mathcal{G}$ captures the global contour information of both the foreground and background by solving (2) and increases the discriminative strength of PHOG descriptors as the global shape information is better maintained (best viewed on high-resolution display).

a path from the root node to a leaf node (see Fig. 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 Fig. 2). Sketches can be easily retrieved by counting the hit frequencies between the local features of a query and training images instead of retaining all the feature descriptors. Similar to [28], 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:

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

where N_α is the total number of training images and n_i denotes the number of training images that have local PHOG descriptors belonging to node i. 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, $w = [w_1, w_2, \ldots, w_i, \ldots]$ 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 [38,39]. Given a sketch pair, I_q and I_r, and the corresponding sketch descriptors h_q as well as h_r, we compute their distance by

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

Thus, for each query sketch I_q, we use D_Λ to denote the distance vector of I_q to a subset Λ of the training sketch images, where each training image has local PHOG features in the same bin of the hierarchical tree as the query sketch. Thus, the distance vector computed on the subset Λ

and retrieval can be performed in sub-linear time. The rank k retrievals are based on:

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

4. Experiments

We present experimental results of the proposed algorithm (PHOG-A) with comparisons to state-of-the-art alternative approaches on aforementioned three data sets. The experimental setup including parameter settings, representative baseline studies, and evaluation criteria is first described (Section 4.1). We discuss experimental results on three data sets respectively (Section 4.2), and analyze the performance of evaluated methods (Section 4.3).

4.1. Experimental setup

In order to set the proper number of dense stroke features (N stroke points of Algorithm 1), we analyze the distribution of the number of stroke points from sketch images of three data sets. The number of stroke points typically falls between 1600 and 1800 as shown in Fig. 7. Thus, we use 1800 stroke points in all the experiments. As we detect $N/4$ corner points and $2N$ randomly sampled stroke points (i.e., $N/4 + 2N \le 1800$ in Algorithm 3.1), we set the value of N to 800 as a trade-off between discriminative strength with a sufficient number of stroke points and computational burden. The area of local feature patch S_l is set to 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 determine the optimal number of cluster centers (K) and depth (L) for the hierarchical vocabulary tree on each data set, one simple but effective grid search method is used with 5-fold cross validations. We initialize K and L as integers within the range of 3 to 10, and search for the optimal pair with highest average retrieval accuracy. Note that the total number of tree nodes N_t is

$$N_t = \frac{K\left(K^L - 1\right)}{K - 1}. \quad (10)$$

We further constrain the search pairs N_t to be less than 10^5 since the use of an excessive number of nodes is likely to cause the overfitting problem. The optimal pairs of K and L on three test data sets are shown in Table 1.

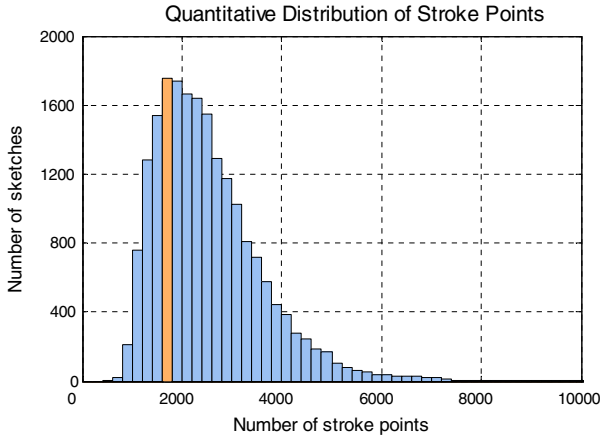

Fig. 7. Histogram distribution of the number of stroke points each sketch contains over three data sets. Each sketch is resized to 256 × 256 pixels and raster scanned to a bitmap image with brush width of 2 pixels.

Table 1
Optimal grid search results about the hierarchical tree parameters (K,L) on three test data sets.

	K	L
Office Icon Library	6	4
ETHZ Shape Data Set	5	4
TU Berlin Sketch Data Set	9	4

To capture spatial information of local features, we use a two-level spatial pyramid kernel. First, we partition each sketch image into two parts relative to the centroid of sampled stroke points horizontally and vertically. Next, 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 structural information without increasing intra-class differences.

4.2. Experimental results

4.2.1. Office Icon Library

We collect a set of 78 Office icons for flow chart creation for training and create a set of 38 hand-drawn sketches as the test set. Some icons in the training set are rather similar with minor shape difference, e.g., arrows shown in Fig. 8. Meanwhile, the hand-drawn queries contain large shape variation when compared with the counterparts in the training set. We train a hierarchical tree of depth 4 with 6 clusters ($K = 6$ and $L = 4$). The retrieval results are evaluated by human subjects (similar to the setup in [6]), and we present the retrieval results and compute the CMA. The results of the proposed algorithm with comparisons to the alternative SPM, SCM and DCM methods are presented in Fig. 8, and the CMA (rank 1 to 6) is shown in Table 2.

4.2.2. ETHZ shape data set

We use 5 hand-drawn shapes from the ETHZ data set [1] (i.e., apple, giraffe, swan, bottle and mug shown in Fig. 1) which consists of 1050 sketches drawn by 50 different subjects (210 sketches per category) for experiments. We train a vocabulary tree ($K = 5$ and $L = 4$) and choose each sketch as query sketch, and compare the CMA of rank 1 to 6 retrieval results as the training set is large.

In addition to comparisons with the SPM, SCM and DCM methods, in Table 3 we present 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 descriptors as well as a hierarchical tree (KHOG-T). The proposed sketch retrieval algorithm (PHOG-A) consists of stroke features from Poisson-based HOG descriptors, a hierarchical tree and a spatial pyramid kernel. Note that the PHOG-A algorithm differs from the PHOG-K method by the use of a hierarchical tree. On the other hand, the PHOG-A algorithm differs from the PHOG-T method by the use of a spatial pyramid.

We show all the failed retrieval cases (7 out of 1050 sketches) from rank 1–6 in Fig. 9. We note that the first five failure cases can be mainly attributed to small inter-class differences, e.g., the query hand-drawn swan in the first row of Fig. 9 is more similar to the retrieved apples or bottle. On the other hand, large intra-class differences are the prime reasons for the last two failure cases, i.e., two bottles in eighth and ninth row are both quite different from the most common vertical bottles (one placed horizontally and the other with 45 degree inclination) in the training set.

4.2.3. TU Berlin Sketch data set

Eitz et al. [8] collect 20,000 sketches representing 250 different objects (each object has 80 different sketches) via crowd sourcing. In [8], the goal is to analyze and compare hand-drawn sketch recognition capabilities of humans and computers. The bag-of-words approach with

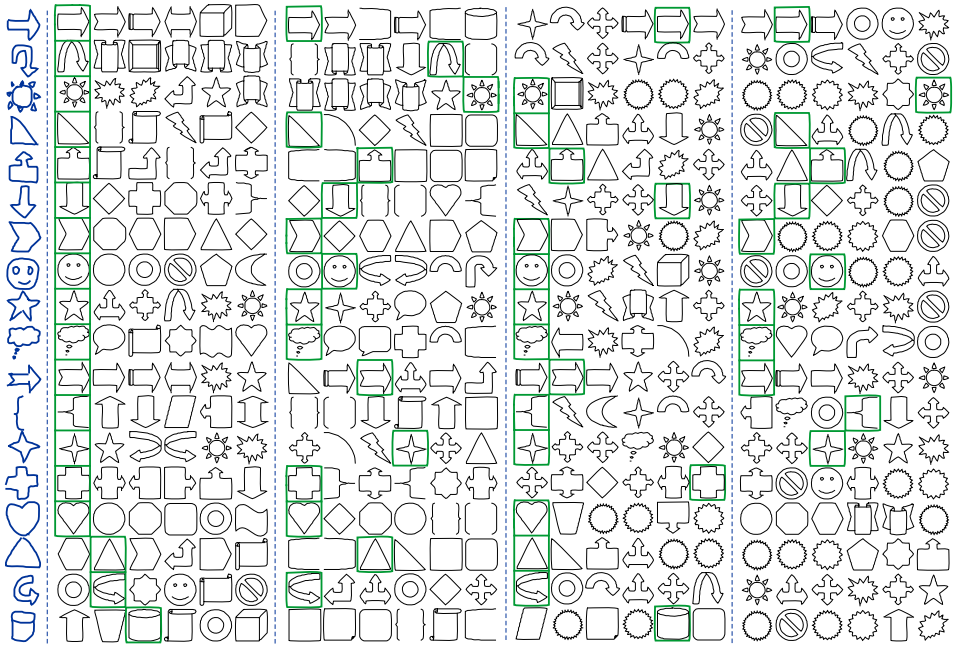

Fig. 8. Rank 6 Office icon retrieval results. From left to right: sampled hand-drawn queries, sketch retrievals by the proposed algorithm, SPM, SCM and DCM methods, respectively. The most similar retrievals are marked by green squares.

dense SIFT features and a codebook with k-means clustering is used for sketch representation, which is the same as the above-mentioned SPM method except that the spatial pyramid kernel is not used as it is shown to provide little performance improvement [8]. We use the last 20 sketches per category as the query set, and use the others to train the vocabulary tree ($K = 9$ and $L = 4$) for experiments. We additionally compare with the CSGC method [40], where the stroke lines are linked as chains to compute the similarity scores for sketch retrieval. Fig. 10 shows the CMA and CBMA curves of all evaluated methods. Some retrieved sketches by the proposed PHOG-A are presented in Fig. 11. To demonstrate the effectiveness of the Harris corner points for reducing randomness of retrieval results, we carry out experiments 100 times respectively on the TU-Berlin Sketch data set using: (1) only the randomly selected points and (2) the Harris corners together with randomly selected points. We report the mean and variance of cumulative match accuracy with rank 16 (a larger rank value is more sensitive to such randomness) as 81.6% $\pm$ 0.058 (only randomly selected points) against to 83.3% $\pm$ 0.010 (Harris corners with randomly selected points). These experimental results show that the proposed key points not only improve the retrieval accuracy but also reduce randomness of retrieval results.

4.3. Discussion

The DCM, SCM and CSGC methods are based on holistic representations which capture global spatial information of sketch images. For sketches with simple shapes, the layout information becomes more important, and these holistic methods are effective for sketch retrieval. As shown in the experiments with the office icon and ETHZ data sets, these methods achieve comparable results with 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 and thus it cannot be applied to large-scale sketch retrieval.

The proposed algorithm and the SPM methods are based on local features with a codebook 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 [17,33]. However, hand-drawn sketches consist of strokes with no texture information. We sample local features on stroke lines instead of uniformly sampling over the entire image, and use Poisson-based HOG descriptors with coarsely quantized histograms. As shown in Table 3, the PHOG-K method performs better (rank 1) than the SPM method which demonstrates the sampled stroke features are more effective for sketch representation. On the other hand, the comparisons of the PHOG-T and KHOG-T methods demonstrate the effectiveness of coarse quantization (PHOG) in accounting for large shape variation of sketches.

We note that 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 [8] based on a bag-of-words approach. Experimental results show that better accuracy can be achieved by the proposed two-level spatial pyramid kernel especially for rank 1 tests. Table 3 and Fig. 10 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 structural information of visual

Table 2
Cumulative matching accuracy on Office Icon Library from rank 1–6 (%). Bold data highlight the best results.

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
DCM [2,4]	36.84	52.63	65.79	71.05	71.05	73.68
SCM [26]	71.05	78.95	81.58	81.58	92.11	94.74
SPM [17]	52.63	63.16	73.68	81.58	86.84	86.84
PHOG-A	**76.32**	**92.11**	**97.37**	**100**	**100**	**100**

Table 3
Cumulative matching accuracy on ETHZ Shape data set from rank 1–6 (%). Bold data highlight the best results.

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
DCM [2,4]	95.14	97.33	97.71	98.19	98.38	98.48
SCM [26]	92.86	96.38	97.81	98.38	98.57	98.57
SPM [17]	96.19	97.52	98.10	98.29	98.38	98.48
PHOG-K	96.48	97.52	97.90	98.10	98.48	98.67
KHOG-T	96.48	97.81	98.19	98.48	98.67	99.05
PHOG-T	96.67	98.38	**98.67**	**98.67**	**99.14**	**99.14**
PHOG-A	**97.14**	**98.48**	**98.67**	**98.67**	99.05	**99.14**

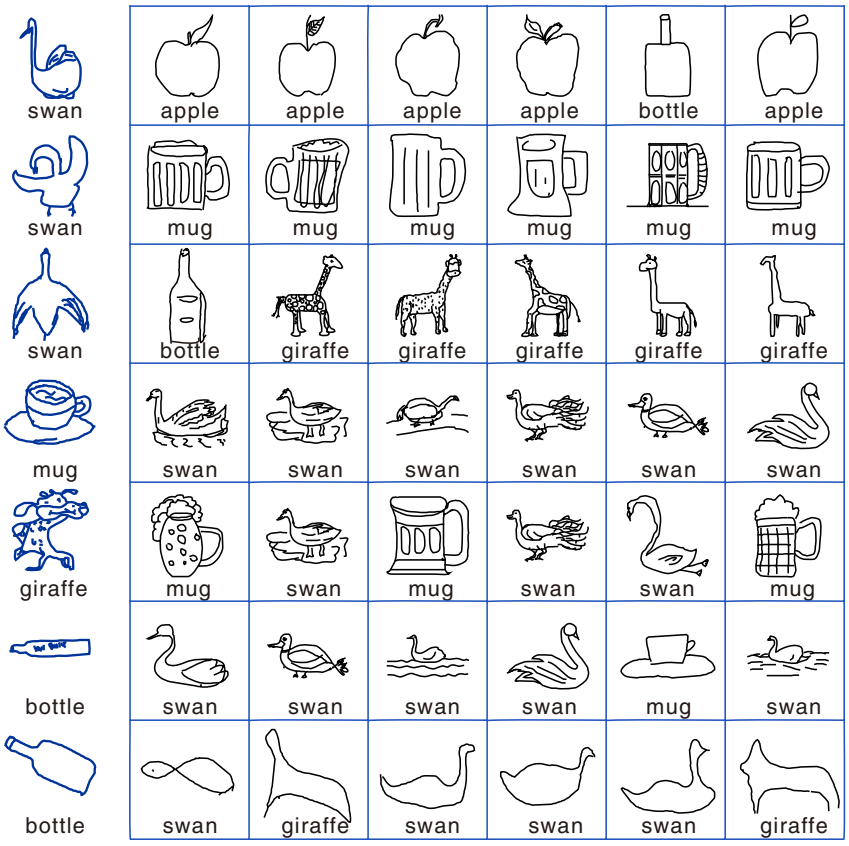

Fig. 9. Rank 6 failure cases (7 out of 1050 sketches) on the ETHZ shape data set. In the first column are query sketches (blue) and in the remaining columns are the corresponding failure retrievals (rank 1–6).

words. In Table 3, the results of the PHOG-T method over the PHOG-K approach demonstrate that the use of a vocabulary tree helps improve retrieval accuracy.

Since users are not aware of the retrieval contents of the training dataset, the query sketches may be significantly different or with incomplete shapes. We attribute this issue to the large intra-class difference between query and retrieval pairs. We qualitatively discuss two examples, where large intra-class differences exist in the query and retrieved sketch pairs. The first one is shown in Fig. 6(a) and Fig. 8 (12th row): the query sketch of a left bracket is not drawn well, i.e., the long sharp curve is missing (which may be considered as an incomplete query). Fig. 9 shows another example that the giraffes and swans on the last row have fewer stroke lines compared to those on the third row. For these cases, the proposed algorithm exploits a hierarchical tree to index the discriminative local features and thus effectively suppresses such intra-class difference and retrieves the most holistically similar sketches.

Failure retrieval cases using the proposed algorithm are presented in Fig. 9 and 11. The proposed PHOG-A method for sketch retrieval is based on a local representation scheme. Although a hierarchical tree retains structural information, some useful holistic information is not exploited.

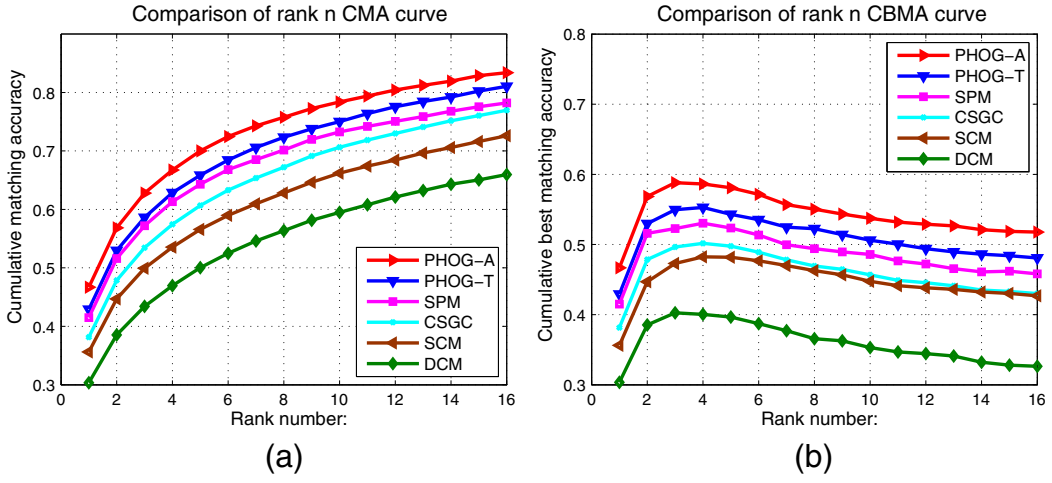

Fig. 10. Rank *n* CMA (a) and CBMA (b) curves with the TU Berlin sketch data set.

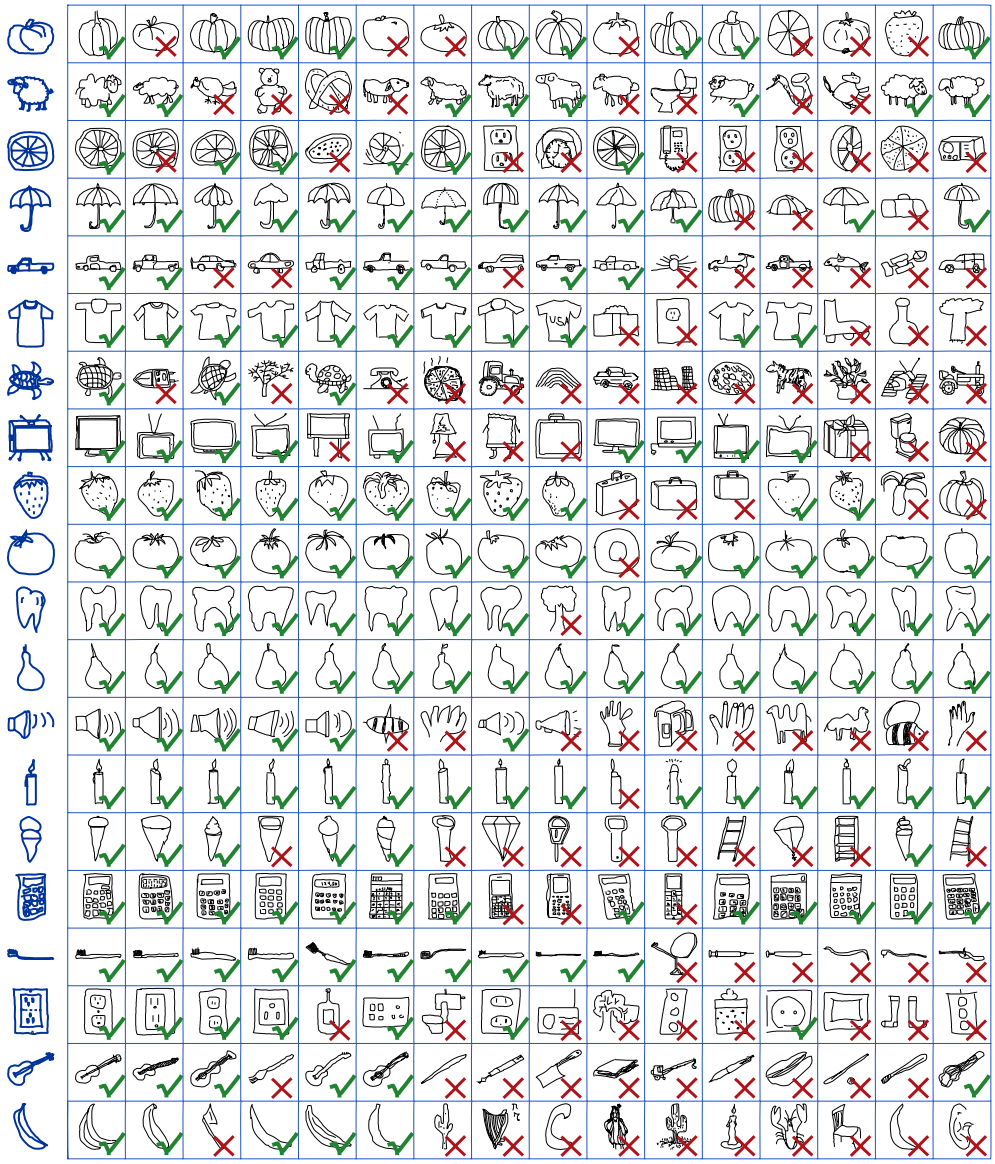

Fig. 11. Sample retrieval results on the TU Berlin sketch database. In the left column are the query sketches (blue) and in the right columns are the corresponding rank 16 retrievals. The retrievals with the same category as the corresponding query sketch are marked by √ otherwise marked by ×.

For example, the giraffe query sketch in the fifth row of Fig. 9 has few spots, while the first, third and fifth ranked retrievals have many spots (i.e., the holistic shape difference is not exploited by the local representation scheme). Likewise, the query pumpkin sketch in the first row of Fig. 11 has some vertical stripes, while the second, sixth and seventh ranked retrievals have few stripes. In addition, sketches with considerably smaller inter-class differences are readily categorized as the same class, e.g., the query sketch in fifth row in Fig. 11 is a pickup truck, while the third, fourth and twelfth retrievals are race cars; the eighth and sixteenth retrievals are SUVs; and the thirteenth retrieval is a truck.

Overall, the proposed algorithm PHOG-A performs favorably against other state-of-the-art methods and alternatives. Implemented in MATLAB on a desktop computer with a 3.1 GHz CPU and 4 GB memory, each retrieval takes less than 0.01 s on the ETHZ data set which is nearly the same as the 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 s for each retrieval as opposed to other methods (PHOG-T: 0.08, SPM: 0.06, DCM: 1.04, and SCM: 2.57 s).

5. Conclusion

In this paper, we propose a novel representation for hand-drawn sketches based on stroke features. Local features are detected via densely sampled stroke points and described by a quantized histogram of gradients interpolated by the Poisson equation. 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 benchmark data sets demonstrate that the proposed algorithm performs favorably against other state-of-the-art methods for sketch retrieval.

References

[1] V. Ferrari, T. Tuytelaars, L. Gool, Object detection by contour segment networks, Proceedings of European Conference on Computer Vision 2006, pp. 14–28.
[2] M. Liu, O. Tuzel, A. Veeraraghavan, R. Chellappa, Fast directional Chamfer matching, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2010, pp. 1696–1703.

[3] P. Srinivasan, Q. Zhu, J. Shi, Many-to-one contour matching for describing and discriminating object shape, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2010, pp. 1673–1680.
[4] Y. Cao, C. Wang, L. Zhang, L. Zhang, Edgel index for large-scale sketch-based image search, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2011, pp. 761–768.
[5] T. Ma, L. Latecki, From partial shape matching through local deformation to robust global shape similarity for object detection, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2011, pp. 1441–1448.
[6] M. Eitz, K. Hildebrand, T. Boubekeur, M. Alexa, Sketch-based image retrieval: benchmark and bag-of-features descriptors, IEEE Trans. Vis. Comput. Graph. 17 (11) (2011) 1624–1636.
[7] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, M. Alexa, Sketch-based shape retrieval, Proceedings of SIGGRAPH, 31 2012, pp. 31:1–31:10.
[8] M. Eitz, J. Hays, M. Alexa, How do humans sketch objects? Proceedings of SIGGRAPH 2012, pp. 44:1–44:10.
[9] D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, W. H. Freeman, 1982.
[10] Y. Rui, T.S. Huang, S.-F. Chang, Image retrieval: current techniques, promising directions, and open issues, J. Vis. Commun. Image Represent. 10 (1) (1999) 39–62.
[11] M. Donoser, H. Riemenschneider, H. Bischof, Efficient partial matching of outer contours, Proceedings of Asian Conference on Computer Vision, 2009.
[12] R. Hu, M. Barnard, J. Collomosse, Gradient field descriptor for sketch based retrieval and localization, Proceedings of IEEE International Conference on Image Processing 2010, pp. 1025–1028.
[13] L.J. Latecki, R. Lakämper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2000, pp. 1424–1429.
[14] T.B. Sebastian, P.N. Klein, B.B. Kimia, Recognition of shapes by editing their shock graphs, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 550–571.
[15] C. Aslan, S. Tari, An axis-based representation for recognition, Proceedings of IEEE International Conference on Computer Vision 2005, pp. 1339–1346.
[16] X. Bai, L.J. Latecki, Path similarity skeleton graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 30 (7) (2008) 1282–1292.
[17] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2006, pp. 2169–2178.
[18] C. Ma, X. Yang, C. Zhang, X. Ruan, M.-H. Yang, Sketch retrieval via dense stroke features, Proceedings of the British Machine Vision Conference, 2013.
[19] W.H. Leung, T. Chen, Retrieval of sketches based on spatial relation between strokes, Proceedings of IEEE International Conference on Image Processing 2002, pp. 908–911.
[20] W.H. Leung, Representations, feature extraction, matching and relevance feedback for sketch retrieval, Doctral Dissertation of Carnegie Mellon University, 2003.
[21] C.T. Zahn, R.Z. Roskies, Fourier descriptors for plane closed curves, IEEE Trans. Comput. C-21 (3) (1972) 269–281.
[22] E. Persoon, K.-S. Fu, Shape discrimination using Fourier descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 8 (3) (1986) 388–397.
[23] D. Zhang, G. Lu, Generic fourier descriptor for shape-based image retrieval, Proceedings of IEEE International Conference on Multimedia and Expo 2002, pp. 425–428.
[24] V. Ferrari, F. Jurie, C. Schmid, From images to shape models for object detection, Int. J. Comput. Vis. 87 (3) (2010) 284–303.
[25] A. Chalechale, G. Naghdy, A. Mertins, Sketch-based image matching using angular partitioning, IEEE Trans. Syst. Man Cybern. 35 (1) (2005) 28–41.
[26] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.
[27] T. Shao, W. Xu, K. Yin, J. Wang, K. Zhou, B. Guo, Discriminative sketch-based 3d model retrieval via robust shape matching, Comput. Graphics Forum 30 (7) (2011) 2011–2020.
[28] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2006, pp. 2161–2168.
[29] D. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.
[30] H. Bay, T. Tuytelaars, L. Gool, Surf: Speeded up robust features, Proceedings of European Conference on Computer Vision 2006, pp. 404–417.
[31] K. Mikolajczyk, C. Schmid, Scale and affine invariant interest point detectors, Int. J. Comput. Vis. (2004) 63–86.
[32] E. Nowak, E. Jurie, B. Triggs, Sampling strategies for bag-of-features image classification, Proceedings of European Conference on Computer Vision 2006, pp. 490–503.
[33] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2009, pp. 1794–1801.
[34] C. Harris, M. Stephens, A combined corner and edge detector, in: In Proc. of Fourth Alvey Vision Conference, 1988, pp. 147–151.
[35] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2005, pp. 886–893.
[36] P. Perez, M. Gangnet, A. Blake, Poisson image editing, Proceedings of SIGGRAPH 2003, pp. 313–318.
[37] R. Hu, J.P. Collomosse, A performance evaluation of gradient field hog descriptor for sketch based image retrieval, Comput. Vis. Image Underst. 117 (7) (2013) 790–806.
[38] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2010, pp. 3539–3546.
[39] A. Vedaldi, B. Fulkerson, VLFeat: an open and portable library of computer vision algorithms, http://www.vlfeat.org 2008.
[40] S. Parui, A. Mittal, Similarity-invariant sketch-based image retrieval in large databases, Proceedings of European Conference on Computer Vision 2014, pp. 398–414.