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

Sketch retrieval aims at retrieving most similar sketches from a large database based on one hand-drawn query. 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. Second, there exist small inter-class differences, due to their loss of visual information (i.e., texture and appearance). Thus, successful retrieval hinges on an effective representation of sketch images and an efficient search method.

In this paper, we propose an effective representation scheme which takes sketch strokes into account with local features, thereby facilitating efficient retrieval with codebooks. The proposed algorithm 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 1 shows the main steps of the proposed method.

In this work, we use local stroke features to represent sketches instead of contour segments [1, 3, 6]. 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. 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 stroke point detection method is summarized in Algorithm 1.

Inspired by [2], we compute the histogram of orientations on the dense gradient field $\mathcal{G}$ from a sketch image, where $\mathcal{G}$ can be approximated by convolution through $\Delta\mathcal{G}$ and sketch Image I [5]:

$$\mathcal{G}(x,y) = \sum_{u,v} I(x,y)\Delta\mathcal{G}(x-u,y-v), \tag{2}$$

where $\Delta\mathcal{G}$ denotes Laplace of Gaussian operator,

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

Then, we adapt the Poisson-based HOG (PHOG) 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 as follows:

$$\cos(x-\alpha_i)^3 > 0, \tag{4}$$

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

Finally, we use a hierarchical tree to train a codebook in spirit similar to the vocabulary tree [4]. A codebook is organized in a hierarchical vocabulary tree, which maintains structural information of visual words and enables efficient retrieval in sub-linear time. A hierarchical tree can be defined by two parameters, K and L, where K is the number of cluster center and L denotes the depth of tree. We iteratively use the K-means

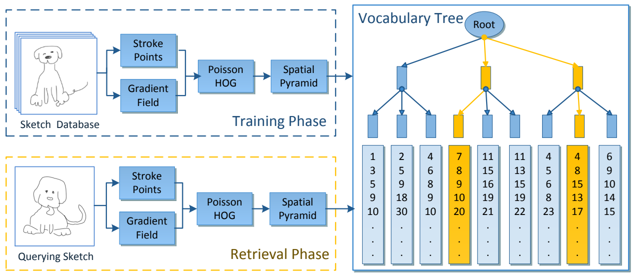

Figure 1: Main steps of the proposed local feature based sketch retrieval: the vocabulary tree is constructed offline for retrieval. Inverted training image identities are indexed below the tree leaves. Each query sketch is retrieved by the hits of the training identities (orange).

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 1). Thus, the histograms 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 1).

Given a query, we find the closest sketches via the sketch descriptor based on their distance using the χ^2 kernel [7]. 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:

$$D(h_q,h_r) = \frac{1}{2}\sum_{i=1}^{n}\frac{[h_q(i)-h_r(i)]^2}{h_q(i)+h_r(i)}. \tag{5}$$

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:

$$rank(k) = \arg\min_{1,\dots,k} D_\Lambda. \tag{6}$$

Experimental results on three data sets with more than 20,000 sketch images demonstrate the merits of the proposed algorithm for effective and efficient sketch retrieval, which performs favorably against state-of-the-art methods in terms of retrieval accuracy and execution time.

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

[2] R. Hu, M. Barnard, and J. Collomosse. Gradient field descriptor for sketch based retrieval and localization. In *ICIP*, 2010.

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

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

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

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

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