Sketch-based Image Retrieval on a Large Scale Database

Rong Zhou, Liuli Chen and Liqing Zhang
MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems
Department of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, China
{rongzhou, chenliuli}@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn

ABSTRACT

The paper presents a simple and effective sketch-based algorithm for large scale image retrieval. One of the main challenges in image retrieval is to localize a region in an image which would be matched with the query image in contour. To tackle this problem, we use the human perception mechanism to identify two types of regions in one image: the first type of region (the main region) is defined by a weighted center of image features, suggesting that we could retrieve objects in images regardless of their sizes and positions. The second type of region, called region of interests (ROI), is to find the most salient part of an image, and is helpful to retrieve images with objects similar to the query in a complicated scene. So using the two types of regions as candidate regions for feature extraction, our algorithm could increase the retrieval rate dramatically. Besides, to accelerate the retrieval speed, we first extract orientation features and then organize them in a hierarchal way to generate global-to-local features. Based on this characteristic, a hierarchical database index structure could be built which makes it possible to retrieve images on a very large scale image database online. Finally a real-time image retrieval system on 4.5 million database is developed to verify the proposed algorithm. The experiment results show excellent retrieval performance of the proposed algorithm and comparisons with other algorithms are also given.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query formulation, Search process

General Terms

Algorithms, Experimentation

Keywords

Sketch, contour, saliency, hierarchical retrieval

MM'12, October 29–November 2, 2012, Nara, Japan.

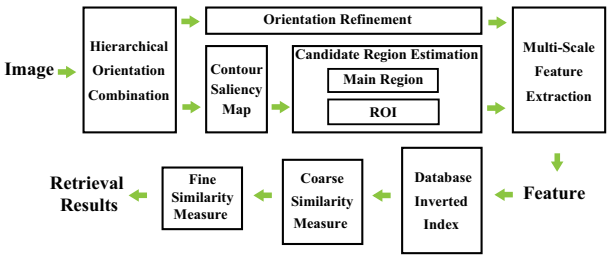

Figure 1: Framework of the whole retrieval procedure.

1. INTRODUCTION

During the image process of human visual system, contour is a core feature for human beings to recognize or distinguish objects from an image or a scene. There have been a lot of studies in contour-based image retrieval system recently [4]. But due to lacking of universal feature extraction and effective database indexing, most of them are only able to search images on small scale image databases [1]. For large scale databases, Eitz et al. [5] presented an algorithm which divides an image into a fixed number of cells, and each cell corresponds to a tensor descriptor. Because of no database index structure, Eitz's algorithm must scan the whole database for each query. Different from it, the method presented by Cao et al. [2] is an index-able oriented chamfer matching method, it focuses on how to build an effective index structure for an image database. However, both methods have same limitations: they only could retrieve images whose objects almost have the same sizes and positions as the object in the query image, thereby resulting in a low recall rate in image retrieval.

In order to develop an image retrieval system that is able to find out retrieved images with objects similar to the query, regardless of their sizes and positions, we define two types of regions: the main region and region of interests (ROI). The main region is defined to tackle the problem that one image only contains one scene (or object) similar to the query but different in size and position; ROI deals with one object similar to the query saliently appears in a complicated background. Thus using the two regions as candidate regions for feature extraction is helpful to improve the retrieval performance dramatically.

Moreover, we build a hierarchical inverted index and split the next process into coarse-to-fine similarity measure. The whole process will filter out a large number of irrelevant images quickly and make it possible to perform the real-time

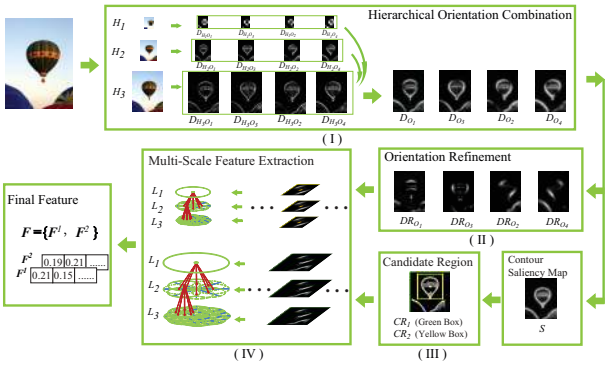

Figure 2: Illustration for feature extraction.

image retrieval. The whole retrieval procedure is illustrated in Fig. 1. Furthermore, we develop a retrieval prototype system which can manipulate image retrieval on 4.5 million images. Finally we provide computer simulations and comparisons with other methods, demonstrating the developed image retrieval system can achieve satisfactory retrieval performance.

2. FEATURE EXTRACTION

In this section, we propose a framework of orientation feature extraction based on the contour saliency map, as shown in Fig. 2.

2.1 Hierarchical Orientation Combination

Human visual system processes images in a hierarchical structure. According to this mechanism, the hierarchical orientation combination is proposed first.

We use the following notations in the rest of the paper. H_i denotes ith level image resolution, H_{i+1} is higher than H_i. And O_j denotes jth orientation, C_q denotes qth RGB color component, D is difference image.

The orientation information of an image is computed by:

$$D_{H_iO_j} = \max_q\{D_{H_iO_jC_q}\}, \quad D_{O_j} = \sum_{i=1}^{M}[D_{H_iO_j}]_{m\times n} \qquad (1)$$

where $D_{H_iO_jC_q}$ is the difference image at ith level resolution, jth orientation and qth RGB color component. $\max_q\{\cdot\}$ is the maximum value over three RGB components. $m \times n$ is highest (H_M) level resolution. $[\cdot]_{m\times n}$ means re-sampling to $m \times n$ resolution proportionably, D_{O_j} is all level resolution's combination of difference images at jth orientation (in our experiment, $M = 3$, $H_{i+1} = 2H_i$, O_j denotes 0, $\frac{\pi}{4}$, $\frac{\pi}{2}$, and $\frac{3}{4}\pi$ orientation respectively, $m \times n$ is not more than 128×128). The contour saliency map S could be obtained by:

$$S = \sum_{j=1}^{N} D_{O_j} \qquad (2)$$

Then the contour saliency map S is normalized to between 0 and 1.

2.2 Orientation Refinement

Fig. 2 (I) shows the appearance of D_{O_j}, we could see D_{O_j} contains not only jth but also other orientation contour information. The redundant information weakens its ability to describe the contour of an object. Considering 0 and $\frac{\pi}{4}$, $\frac{\pi}{2}$ and $\frac{3}{4}\pi$ are orthogonal respectively, we perform the refining operation:

$$\begin{aligned} DR_{O_1} &= \lfloor D_{O_1} - D_{O_3} \rfloor_T, \quad DR_{O_3} = \lfloor D_{O_3} - D_{O_1} \rfloor_T \\ DR_{O_2} &= \lfloor D_{O_2} - D_{O_4} \rfloor_T, \quad DR_{O_4} = \lfloor D_{O_4} - D_{O_2} \rfloor_T \end{aligned} \qquad (3)$$

where DR_{O_j} denotes refined orientation contour information. $\lfloor\cdot\rfloor_T$ is a truncation function, and $T = 0$.

From Fig. 2 (II) we could see, after the refining operation, redundant information has been already removed and DR_{O_j} just remains the jth orientation contour information. So the procedure would enhance the feature's ability to describe thereby leading to higher retrieval precision.

To verify the above assertion, we visualize the principal component analysis (PCA) distributions of features from six different image categories (bag, car, mug, starfish, T shirt, tiger, shown with different colors). Fig. 3 (I) shows the first 3 principal components of features which are extracted from D_{O_j}, and Fig. 3 (II) is corresponding principal components of features extracted from DR_{O_j}. From the distribution we could see, the feature's ability to distinguish become better after orientation refining operation. It explains the rationality of orientation refinement.

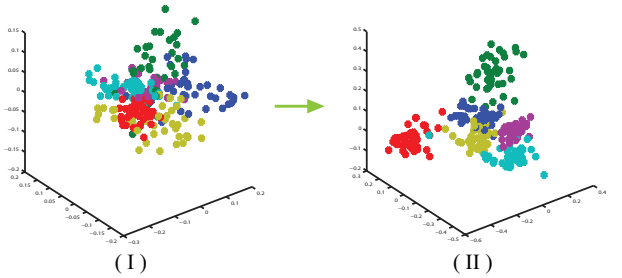

Figure 3: PCA distribution of features from: (I) D_{O_j} vs. (II) DR_{O_j}.

2.3 Candidate Region Estimation

We define two candidate regions on one image: the main region and ROI. the main region is aimed at the problem that the object (or scene) is similar to the query but different in size and position. ROI deals with the problem finding the object which is similar to the query and has most salient contour in a complicated background. The algorithm estimating main region is as follows:

Algorithm 1: Main Region Estimation

1. Remove the tiny contour from the saliency map S:
$$S_1 = \lfloor S \rfloor_{T_1}$$
where $T_1 = 0.25$ in our experiment.

2. Compute the center (x_1, y_1) of the main region:
$$(x_1, y_1) = \arg\max\{S_1 \star G\}$$
where G is the Gaussian kernel which size is $s \times s$, s is the maximum side length of S, and $\star$ denotes convolution. (x_1, y_1) is the coordinates of the convolution maximum value.

3. Let $\delta(x_1, y_1; \gamma_1)$ denotes the square region of S_1 which

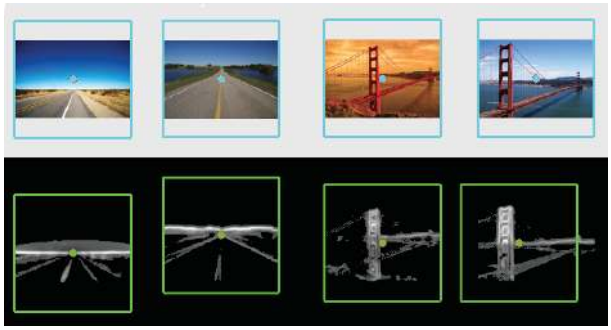

Figure 4: Examples of the main region (green box). Green point is their center; Blue box's center is the center of the image.

center is (x_1, y_1) and side length is $2\gamma_1$. Update $\gamma_1 = \gamma - \Delta\gamma_1$ till the sum of contour value in region $\delta(x_1, y_1; \gamma_1)$ is less than $\alpha \cdot \Im$, $\alpha = 1$, $\Im$ is the sum of contour value in S_1, $\gamma = \frac{1}{2}s$, $\Delta\gamma_1 = \frac{1}{10}\gamma$.

4. So the main region is:

$$CR_1 = \delta(x_1, y_1; \gamma_1).$$

Fig. 4 shows the main region of two kinds of scenes. From the figure we could see, the main region's center could be matched better than the image's center in the case of same scenes. That will be useful to improve the retrieval performance of the algorithm.

Next we regard the most salient contour region of an image as its ROI, as shown in Fig.5. The algorithm estimating ROI is as follows:

Algorithm 2: ROI Estimation

---

1. Remove the tiny contour from the saliency map S further:

$$S_2 = \lfloor S \rfloor_{T_2}$$

where $T_2 = 0.5$ in our experiment.

2. Generate a series of connected point sets $\{P(i)\}$ on S_2;

3. Sort $\{P(i)\}$ to $\{P_s(i)\}$ in descending order by their point numbers (that is, $\{P_s(1)\}$ is the biggest connected point set, as green point sets shown in Fig.5);

4. In general, choose $\{P_s(1)\}$. In the special case, e.g., as shown in Fig.5 (II), if $\{P_s(1)\}$ is not closed, but $\{P_s(2)\}$ is, then choose $\{P_s(2)\}$;

5. The rectangle region that the chosen point set covers is marked by S_p, then let S_1' denotes remaining the value of S_1 in S_p and setting the others out of S_p to zeros;

6. Compute the center (x_2, y_2) and half of side length γ_2 of ROI similar to step 2, 3 of algorithm 1:

$$(x_2, y_2) = \arg\max\{S_1' \star G'\}$$

where the size of Gaussian kernel G' is $s' \times s'$, s' is the maximum side length of S_p. Update $\gamma_2 = \gamma - \Delta\gamma_2$ till the sum of contour value in square region $\delta(x_2, y_2; \gamma_2)$ of S_2 is less than $\alpha \cdot \Im'$, $\Im'$ is the sum of contour value in S_1', $\Delta\gamma_2 = \frac{1}{20}\gamma$. So ROI is $CR_2 = \delta(x_2, y_2; \gamma_2)$.

7. If $\gamma_2 = \gamma$, i.e., ROI may cover the whole saliency map, then go to perform algorithm 1 with $T_1 = 0.5$, $\alpha = 0.9$. Then the resultant region is defined as ROI, Fig.5 (IV) is an example in this case.

Figure 5: Examples of ROI (yellow box)

With algorithm 2, we can only find one ROI in an image. Actually, it could be performed repeatedly to identify more ROIs of the image. If one image has N ROIs and one main region, it will have $N + 1$ features, then one million image database will have $N + 1$ million features. So considering the storage of a large scale database, we just compute one ROI for each image.

2.4 Multi-Scale Feature Extraction

In this subsection, k denotes in the case of different region, $k = 1$ means in the case of the main region CR_1, $k = 2$ means in the case of ROI CR_2. First we define a series of patches $\{X^k(t)\}$ on CR_k, which are hierarchical and overlapping, as shown in Fig. 2 (IV). t is the patch's ID from 1 to 73 ($= 1 + 8 + 8 \times 8$). In L_1, t = 1, the center and the size of patch $X^k(1)$ are equal the ones of CR_k. In L_2, t = 2,...,9, the size of $X^k(t)$ is half of the size of CR_k, and $X^k(t)$'s center is one of eight points which distance to the center of CR_k is one quarter of the side length of CR_k. The rest $X^k(t)$ can be computed in the similar manner. $G^k(t)$ is a Gaussian kernel with the same size as $X^k(t)$.

The features are defined as follows:

$$F_{O_j}^k(t) = \sum [DR_{O_j}]_{X^k(t)} \cdot G^k(t)$$
$$F^k = \{F_{O_j}^k(t)\}, \quad F = \{F^k\} \tag{4}$$

where $[DR_{O_j}]_{X^k(t)}$ denotes the values of DR_{O_j} in patch $X^k(t)$. The final feature F for each image is a group of two features $\{F^k\}$ and each F^k has $73 \times 4 = 292$ dimensions. From equation(4), we could see the feature extraction is hierarchical, these characteristics could be used for fast database index.

The similarity measure of two final features $sim(\cdot)$ could be any similarity measure, e.g., Euclidean distance or cosine similarity.

3. DATABASE INDEX STRUCTURE

The retrieval process for a large-scale database consists of two main steps: the first step is generating the inverted index of the database, which could filter out a large number of irrelevant images quickly and reduce the number of candidate images, the second step is coarse-to-fine similarity

Figure 6: Examples of hand-drawn sketches as query images

measure, that is matching candidate images to the query in more details.

Firstly, we generate an index list for the database with first 36 components of feature F^k. These components correspond to 4 components from L_1 level and 32 components from L_2 level in Fig. 2 (IV)), and contain most of important contour information. For each component, we divide it into N_{L_p} bins, and for each bin, there is an inverted list of images, which is generated by the following method: when the corresponding feature component of an image falls into the bin, the image's ID will be counted to the list. Then we perform inverted lists' intersection operation repeatedly till the number of candidate results is less than threshold N_1.

Then coarse similarity measure is performed still with these 36 components of F^k, and N_2 candidate results are selected. Finally based on all components of F^k, fine similarity measure is used to rank the final N_2 retrieval results. Thus we could build a hierarchical retrieval structure. In our experiment, $N_{L_1} = 20$, $N_{L_2} = 40$, $N_1 = 50000$, $N_2 = 2000$.

4. EXPERIMENT AND RESULT

To evaluate our retrieval algorithm, we develop a prototype system with a database of 4.5 million images from Flickr and Google, and the whole feature database has 9 million features. We run it on the server with 2 Intel Xeon 2.66GHz Six-Core processors and 64GB memory. The average retrieval time is about between $3 \sim 5$ seconds.

We invited 5 subjects to conduct 100 sketch-based retrieval tasks which correspond to 100 different shape objects. So there are totally 100 sketches as query images, some examples of them are shown in Fig. 6.

To verify the proposed algorithm, we show the 100 sketch queries' top 20 average retrieval precision comparison with GIST [6], HOG [3], as shown in Fig. 8. It should be noted that GIST and HOG have no inverted index so they have to scan the whole database for each query. Fig. 7 (I) displays some sketch-based retrieval results of our algorithm. Besides, our system could also handle natural images. Fig. 7 (II) is an example of a natural image as the query.

5. CONCLUSIONS

Developing a practical image retrieval system is still a challenging task. The accuracy and speed are still two key issues in this field. To achieve the goal, we propose a sketch-based algorithm for large scale image retrieval and develop a practical prototype system which can search the results from

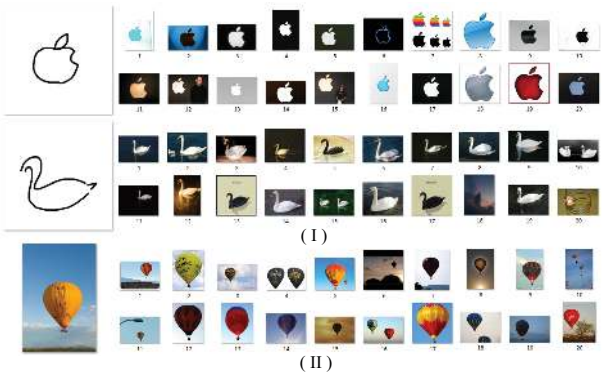

Figure 7: Examples of retrieval results from 4.5 million images: (I) A hand-drawn sketch images as a query; (II) A natural image as a query.

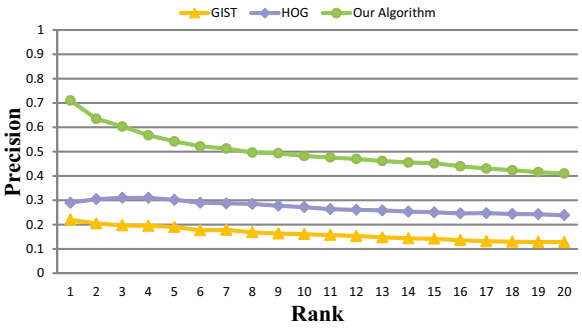

Figure 8: Precision comparison with GIST and HOG

4.5 million images quickly. The experiment results show better precision and efficiency than existing methods.

6. ACKNOWLEDGEMENTS

The work was supported by the National Natural Science Foundation of China (Grant No. 90920014 and 91120305).

7. REFERENCES

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

[2] Y. Cao, C. H. Wang, L. Q. Zhang, and L. Zhang. Edgel index for large-scale sketch-based image search. In *CVPR*, 2011.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.

[5] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. An evaluation of descriptors for large-scale image retrieval from sketched feature lines. *Computers & Graphics*, 34:482–498, 2010.

[6] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.