NATURAL IMAGE RETRIEVAL WITH SKETCHES

Jinyi Yao, Mingjing Li, Zhiwei Li, Lei Zhang, Wei-Ying Ma

Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, China

ABSTRACT

In this paper, we present a method to retrieve natural images by sketch query. To measure the similarity between the sketch and an image, relevant regions are first located in that image through a multi-resolution search, and a normalized local shape similarity is proposed for image retrieval. Efficiency and other implementation issues are discussed. Experimental results show that it is an effective approach for content-based image retrieval.

1. INTRODUCTION

Content-based image retrieval (CBIR) has drawn widespread research interest. In CBIR systems, color and texture features are commonly used to represent the image content, while shape features are occasionally used. Most CBIR systems assume that a sample image is available to initiate the retrieval process. However, in many cases, it's quite inconvenient for users to find a sample image as query. A better way is to allow users to draw a sketch to represent the search concept. This is the so-called query by sketch.

Sketch contains only shape information. A user survey about cognition aspects of image retrieval shows that users are more interested in retrieval by shape than by color and texture [2]. However, retrieval by shape is still considered one of the most difficult aspects of content-based search [4]. Rajendran et al [3] generates edge signatures for images and sketch query, and compare their curvature-histograms and direction-histograms for shape similarity. In [6], strokes are extracted from images and query, then the spatial order and feature distance of them are considered for shape similarity. Mori et al [1] use "shape context" to quickly prune a search for similar shape, where a shape is a discrete set of points and for each point the shape context is a histogram of relative positions of the remaining points. All these systems can only work on clean images where each contains only one object and is compared to the query image or sketch as a whole.

Natural images are much more complex, often contain multiple and irrelevant objects. The shape similarity of the sketch to images is actually determined by that of the sketch to local relevant regions. But exhaustive search over all regions in each image is unaffordable. Rucklidge [5] proposed an algorithm to efficiently localize objects using Hausdorff distance. We extend his algorithm to image retrieval by sketch in this paper. For each image, we localize a set of regions relevant to the given sketch. Then a normalized local shape similarity measure is proposed to select the most similar one. Some techniques for efficiency are also exploited.

The remainder of this paper is organized as follows. In Section 2, we briefly illustrate the process to localize local relevant regions. In Section 3 we present the normalized local shape similarity and describe the whole retrieval system. Experiments are presented in Section 4. Finally, a short conclusion is given in Section 5.

2. RELEVANT REGIONS LOCALIZATION

Natural images record some objects together with their environments, including background and other neighbor objects. Besides, each image is an observation under some perspective. Relevant regions in an image refer to those with high shape similarity to the sketch that undergo some affine transformation. To perform image retrieval by sketch, we need to solve two problems. One is how to define the shape similarity between the sketch and the relevant region; the other is how to find relevant regions, which is actually a search problem in the affine transformation space.

2.1. Shape Similarity by Hausdorff distance

Shape can be represented by a set of points, for both sketches and images. Hausdorff distance from set A to set B is defined as:

$$h(A,B) = \max_{a \in A} \{ \min_{b \in B} \|a - b\| \} \tag{1}$$

where a and b are points of sets A and B respectively. It will be small when every point of A is near some point of B. In this paper, A is a point set representing an image, and B is a point set representing a sketch. We call $h(B,A)$ the forward distance, and $h(A,B)$ the reverse distance.

Generally $h(A,B) \neq h(B,A)$, and undirected Hausdorff distance can be defined as

$$H(A,B) = \max(h(A,B), h(B,A)) \tag{2}$$

0-7803-9332-5/05/$20.00 ©2005 IEEE

To be robust to noise, partial directed Hausdorff distance $h^f(A,B)$ is considered. f is some value between zero and one. $h^f(A,B)$ is computed by finding the distance from every point of A to the closest point of B, then find the f-th quantile value of these values. When $f=1$, $h^f(A,B) = h(A,B)$.

To consider the shape similarity between the sketch B and a local region in image A, which is in a box $[m_x, M_x] \times [m_y, M_y]$, the box-reverse Hausdorff distance $h_{box}(A,B)$ is defined [5].

$$h_{box}(A,B) = \max_{\substack{(ax,\,ay)\,\in\,A \\ mx \,\le\, ax \,\le\, Mx \\ my \,\le\, ay \,\le\, My}} \{\min_{b\in B} \|(a_x,a_y) - b\|\} \quad (3)$$

As before, the partial box-reverse Hausdorff distance can be defined and denoted by $h^{f_R}_{box}(A,B)$.

Suppose that t is an affine transformation. It is a six-tuple $(m_{00}, m_{01}, m_{10}, m_{11}, t_x, t_y)$, representing a mapping

$$(x,y) \rightarrow (m_{00}x + m_{01}y + t_x, m_{10}x + m_{11}y + t_y) \quad (4)$$

Let $t(B)$ denotes the result of applying t to B. Its boundary decides a local region in A. The partial forward distance $d(t)$ and partial box reverse distance $d'(t)$ can be defined as [5]:

$$d(t) = h^{f_F}(t(B), A) \quad (5)$$

$$d'(t) = h^{f_R}_{box}(A, t(B)) \quad (6)$$

There are two criteria [5] to evaluate t, which is actually to evaluate the shape similarity between $t(B)$ and the corresponding local region in A.

1. **Forward Criterion.** $d[t] \le \tau_F$.
2. **Reverse Criterion.** $d'[t] \le \tau_R$.

An equal and computationally easier method to verify these two criteria is to calculate the fraction of points whose forward distances are less than τ_F, and the fraction of points whose box-reverse distances are less than τ_R. They are called forward fraction $f[t]$ and box-reverse fraction $f'[t]$ respectively. The corresponding thresholds are f_F and f_R

2.2. Multi-resolution Search

To locate the relevant regions in an image, we adopt Rucklidge's method [5] to search through the transformation space to find all transformations that satisfy the above criteria. Here we briefly describe his algorithm.

We first explain two approximations techniques in this algorithm. The transformation space is rasterized so that only transformations $(i_1/M_x, i_2/M_y, i_3/M_x, i_4/M_y, i_5, i_6)$ for integer values of $i_1 \cdots i_6$ are considered. $[M_x, M_y]$ is the sketch size. According to equation 4, changing any one of these integer parameters by ± 1 changes the location of any transformed sketch point by at most one unit. The transformation is alternatively represented by $[i_1 \cdots i_6]$, using square brackets to indicate the raster basis is being used. Another approximation technique is rounding transformed points into integral coordinates. $t[B]$ denotes the result of rounding each point of $t(B)$. $d[t]$ and $d'[t]$ are defined by replacing $t(B)$ with $t[B]$ in equations 5 and 6.

For efficiency, the whole process is divided into two phases. The first one is to find all transformations that satisfy the forward criterion. The second one is to verify them on the reverse criterion one by one.

The fist phase is implemented by a cell decomposition strategy. "Cells" refer to rectilinear regions of transformation space. The whole transformation space is initially divided into a set of cells with equal size. Then each cell is tested to see whether it's possible to contain any transformation satisfying the forward criterion. If not, it's dropped. Otherwise this cell is further divided into a set of cells with finer resolution. We repeat testing and dividing until the finest resolution is reached where each cell contains only one transformation. The key technique is to efficiently test whether a given cell possibly contains a transformation that $d_t \le \tau_F$

For $d[t]$, we need to compute the closest point in A for each point of $t[B]$. Since all the points of $t[B]$ have integral coordinates, we can compute the closest point in A for all integral points beforehand. The result is the distance transform of A:

$$\Delta(x,y) = \min_{a\in A} \|(x,y) - a\| \quad (7)$$

Now given t, we can probe $\Delta(\cdot)$ for all points of $t[B]$, of which the f-th quantile value is $d[t]$.

Given a cell R, define $d[R] = \min_{t\in R} d[t]$. The ideal rule to reject a cell is to decide whether $d[R] > \tau_F$ holds. It's approximated by the following method to efficiently estimate a lower bound of $d[R]$.

Denote t^l as the transformation whose parameters all have their lowest value in R and t^h having highest values.

$$t^l = [m^l_{00}, m^l_{01}, m^l_{10}, m^l_{11}, t^l_x, t^l_y]$$

$$t^h = [m^h_{00}, m^h_{01}, m^h_{10}, m^h_{11}, t^h_x, t^h_y]$$

Given any point b, as t varies within the cell R, $t[b]$ varies within a box whose top left corner is at $t^l[b]$ and bottom right corner is at $t^h[b]$. The box size is d_x by d_y:

$$d_x = (m^h_{00} - m^l_{00}) + (m^h_{01} - m^l_{01}) + (t^h_x - t^l_x)$$

$$d_y = (m^h_{10} - m^l_{10}) + (m^h_{11} - m^l_{11}) + (t^h_y - t^l_y)$$

Define the box distance transform of A:

$$\Delta'(x,y) = \min_{\substack{0 \,\le\, x' \,\le\, dx \\ 0 \,\le\, y' \,\le\, dy}} \Delta(x+x', y+y') \quad (8)$$

Obviously $\Delta'(t^l[b]) = \min_{t \in R}(\Delta(t[b]))$ holds for any point b. Therefore the f-th quantile value of $\Delta'(t^l[\cdot])$ for all points in B is a lower bound of $d[R]$. The box size d_x by d_y is determined only by the size of R , so $\Delta'(\cdot)$ can be computed beforehand for all integral points on each level of resolution.

3. IMAGE RETRIEVAL

Before doing the search, we do not know how well a given sketch represents the objects that the user intends to search for. Therefore, we can not set the forward criteria too strictly. Otherwise we may fail to find satisfying transformations in all images. Therefore with the criteria being not too strict, we have to handle a number of satisfying transformations for many images in general cases. For each image, we need to find the best one from the candidate transformations to represent the shape similarity between the whole image and the sketch.

Rucklidge [5] extended his method to find the best transformation which he defined as the one with minimum $d[t]$. But it's not suitable for general image retrieval by sketch. Because $d[t]$ is a biased shape similarity measure, which we'll explain next.

3.1. Normalized Local Shape Similarity

We derive a measure from previous criteria to compute the shape similarity between region and sketch. Due to the following two kinds of bias, the aforementioned criteria can not be directly used as shape similarity measure for image retrieval.

1. **Detail Bias** The forward fraction $f[t]$ is biased toward those that transform the sketch into regions with more details. In the extreme case, $f[t]$ can be very close to 1 for a region full with edge points, whatever the sketch is.

2. **Scale Bias** Both the forward fraction $f[t]$ and the box-reverse fraction $f'[t]$ are biased toward those that transform the sketch into small regions. In the extreme case, $f[t]$ and $f'[t]$ can be very close to 1 when the scale value of a transformation t is very small, whatever the sketch and the image are.

These two kinds of bias are very common for image retrieval by sketch. For the detail bias, sketch is supposed to be more concise than images. And images often vary in the magnitude of details. Those with more details are more possible to have a higher value on $f[t]$. For the scale bias, sketch is not in the same scale with objects in images. And images also vary in the scale level.

So we propose to use a normalized $f'[t]$ to compute the shape similarity. We note that $f[t]$ is still necessary because candidate transformations are generated by passing the forward criteria.

As mentioned before, $f'[t]$ is the fraction of points whose partial box reverse distances are less than τ_R:

$$f'[t] = \frac{\#(\{a \in A^t_{box} \mid \min_{b \in B} \|a - t[b]\| < \tau_R\})}{\#(A^t_{box})} \quad (9)$$

where A^t_{box} denotes the point set of the local region in A corresponding to t and $\#$ is the size function, $\#(A^t_{box})$ is the number of points in A^t_{box}.

To make compensation for Scale Bias, the normalized box-reverse fraction $f'_N[t]$ is defined as:

$$f'_N[t] = f'[t] \times \sqrt[2]{t^2_{sx} + t^2_{sy}} \quad (10)$$

Where t_{sx}, t_{sy} are the scale coefficients of t in x and y dimensions respectively. $f'_N[\cdot]$ can help to find an appropriate one from the candidate transformations, instead of biasing toward small scale.

Other existing shape similarity measure can also be used instead of $f'_N[\cdot]$. But they require much more extra computation effort.

3.2. Retrieval System

Now we summarize the whole retrieval system. Edges are extracted for all images beforehand. For each image, A is the set of its edge points. B is the set of points sampled directly from the sketch drawing. Given these two point set, we find all relevant regions in A whose corresponding transformations satisfy the forward criterion, then select the best one t^* according to $f'_N(\cdot)$. At last we sort all the images according to $f'_N(t^*)$. Those with high values on $f'_N(t^*)$ are returned.

In this system, the efficiency bottleneck is to localize relevant regions. The cost time varies much across images. It depends on threshold parameters: τ_F and f_F. In the extreme case of $f_F = 0$, all transformations can satisfy the forward criterion. Then it's an exhaustive search. Besides, even with hand tuned parameters, it may still costs more than ten hours to complete the search process for some images. The extreme sample is an image full with edge points, where all transformations can satisfy the forward criterion.

We take two approximated techniques to prevent from wasting too much time on one image. The first one is to apply a timing interruption mechanism. When the cost time is beyond 2 minutes on one image, the search process is stopped. The second one is to stop when the number of probed cells is more than 2,400,000. It doesn't mean that we drop the image when the search process is forced to stop. Actually before it is stopped, we may have already found many satisfying transformations for this image. Then we can still get an approximation similarity value between this image and the sketch.

4. EXPERIMENTS

The image database consists of 1,100 from 11 categories of Corel database, each of which contains 100 images. The categories include Balloon, Beach, Bird, Bobsled, Bonsai, Building, Bus, Butterfly, Car, Cat and Cougar. It is hard to draw representative shape for Beach and Cougar. For each of the other 9 categories, we draw one sketch as query. All images are preprocessed by canny edge extraction. For time reason, we only do experiments in the 4D transformation space: translation and scale. The scale for x and y dimension ranges from 1 to 1.5. The translation range is limited only by images' size. The four threshold parameters are set as following: $\tau_F = 10, f_F = 70\%, \tau_R = 10, f_R = 40\%$. All experiments are conducted in a PC with Xeon 3.0 GHz CPU and 2.0GB RAM.

4.1. Accuracy

To evaluate system accuracy, average precision-scope measure is applied. Scope specifies the number of images returned. Precision is defined as the number of retrieved relevant images over the value of scope. Those images in the same category as the query are deemed to be relevant. The result is shown in Figure 1, benchmark refers to random selection strategy. Among all images, there are about 9% in the same category with a given sketch, so the expected precision of random selection is about 9%.

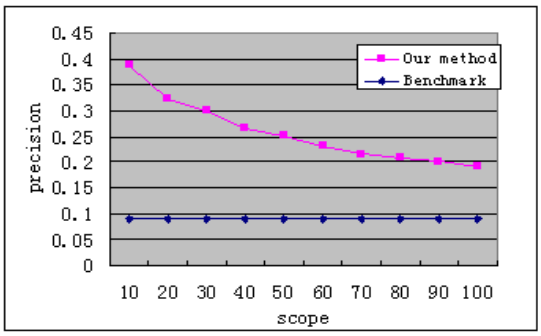

Fig. 1. Average Retrieval Accuracy

4.2. Efficiency

The running time for one sketch query "Balloon" is shown in Figure 2. The x axis is the cost time in seconds for one image. The y axis is the ratio of images whose running times are longer than the corresponding x value. The timing interruption mechanism is not very accurate so that it may cost a little more than 2 minutes in a few cases.

The average running time to compare a sketch to each image is less than 40 seconds.

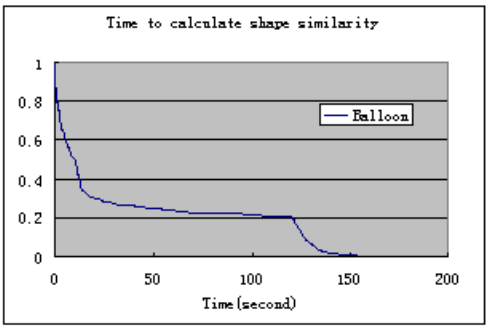

Fig. 2. Time to Searching through Images

5. CONCLUSION

In this paper, we present a method to retrieve natural images by sketch. Local regions in each image are located to match the given sketch. A Hausdorff distance based criterion is taken as the shape similarity measure. This method is robust to affine transformation. The major drawback of this method is that its time cost is too high. More efficient approximation algorithm to locate sketch in images is worthy further investigation.

6. REFERENCES

[1] Gerg Mori, Serge Belongie and Jitendra Malik, "Shape contexts enable efficient retrieval of similar shapes" *Computer Vision and Pattern Recognition*, 2001.

[2] Lambert Schomaker, Edward de Leau and louis Vuurpijl. "Using pen-based outlines for object-based annotation and image-based queries" *Visual Information and Information Systems - Proceedings of the Third International Conference VISUAL'99*, 1999

[3] Raj Kumar Rajendran and Shih-Fu Chang, "Image Retrieval with Sketches and Compositions." *IEEE International Conference on Multimedia and Expo (ICME)*, New York, July 2000.

[4] Remco C. Veltkmap and Michiel hagedoorn, "State of the Art in Shape Matching" *Principles of Visual Information Retrieval, pages 87-119. Springer*, 2001.

[5] William J. Rucklidge, "Efficiently Locating Objects Using the Hausdorff distance." *International Journal of Computer Vision*, 24:251-270, 1997.

[6] Wing Ho Leung and Tsuhan Chen, "Trademark Retrieval using contour-skeleton stroke classification" *Proc. Int. Conf. on Multimedia and Expo, volume 2, pages 517-520. IEEE*, 2002.