Search by Sketch

Ching-Feng Yeh, Tsung-Hsien Wen, Ching-Chia Wang
National Taiwan University
d00942013@ntu.edu.tw, r00921033@ntu.edu.tw, r00921048@ntu.edu.tw

Figure 1. Example results of the Search by Sketch system, queried by two sample sketches. (http://140.112.21.28/gpu/draw-to-retrieve.html)

Abstract

In this project, we have implemented an interactive sketch-based image retrieval system called "Search by Sketch." (See Fig.1) The client end is a colorful user-interface webpage consisting of a drawing panel and a display panel, while the server end takes charge of the retrieval task. When the user draws an arbitrary sketch and then clicks the search button, the server will retrieve 20 most similar images from our database, which contains approximately 7,000 world landmark photos, in about 0.8 seconds. All the photos in the database are collected from Google and Flickr. The core of our retrieval algorithm, the edgel index algorithm, is proposed by [2].

Keywords: sketch-based image retrieval, edgel indexing

1. Introduction

With the advance of communication technology, multimedia applications are becoming more and more popular today. With the extremely large amount of available information on the internet, image retrieval has become one of the most important tasks in multimedia domain. However, different from text-based retrieval systems, which are conventional and mostly well-developed, image retrieval tasks are much more difficult and require further development. For one, the extracted features representing an image are hard to be defined. Unlike text documents, semantic information of image is not well-defined and usually more complicated to extract. Furthermore, due to the variations of lighting conditions, spatial disorder, and even types of camera, two similar images containing alike semantic information can be very different on pixel level.

In this project, we utilized the concept of matching between sketch and image to minimize the semantic gap which exists in image retrieval tasks. Following the previous works [1,2], we developed a complete system using sketch as query in image retrieval. Different from the conventional image-image matching systems, sketch queries have shown several advantages. First, sketch is drawn according to the contour, which is the most important part of the desired image. As a result, only the relevant part remains and the contribution of the irrelevant parts can be avoided. Besides, we do not consider colors in current implementation, which reduced the mismatch by lighting or camera type. The most important of all, the sketch provides information that cannot be described by text only and therefore is independent of languages, which indicates that sketch might be the closest form to human visual perception.

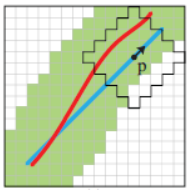

Figure 2. Illustration of a straight line (blue line), a curved sketch line (red line), and its tolerance radius (the colored region). The straight line is regarded as 100% matched with the sketch line.

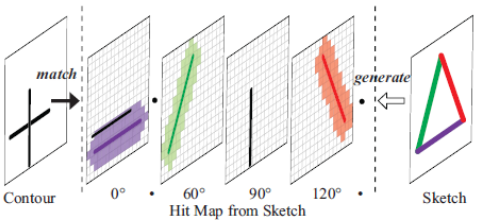

Figure 3. An illustration of splitting sketch image into several hit-maps according to edgel's quantified angles. In the illustration, there are only 4 kinds of quantified edgel angle, but we in practice use 8 angles to get more accurate retrieval results.

2. The Edgel Index Algorithm

2.1 Indexable Oriented Chamfer Matching

We choose the edgel index algorithm [2], which is a modified version of the Oriented Chamfer Matching [4], to calculate the similarity between sketch and images. In the algorithm, an edgel is defined as the edge pixel with its angle in an image. It can be represented by p = (x, y, e), where x and y represent the 2-dimensional position of this edgel in the image, and e is the quantified angle of its gradient direction. A tolerance radius is set for the fact that sketch lines are usually not precisely drawn straight, as shown in Fig. 2.

Every image (photo and sketch) is divided into 8 hit-maps, where each hit-map contains all edgels of one certain quantified angle. The similarity between two images is defined as the number of "hits" between their hit-maps.

The similarity score of a photo D with a sketch Q is calculated from the number of edgel hits between them. An edgel $p \in D$ is considered matched if there's an edgel $q \in Q$ that locates in the range of its tolerance radius and has the same quantified angle:

$$\mathrm{Hit}(p) = \begin{cases} 1, \exists q \in Q \ (\|x_q - x_p\| \le r \ \cap \ \theta_q = \theta_p) \\ 0, \text{otherwise.} \end{cases} \quad (1)$$

where x_q and x_p are the positions of p and q respectively, and r is the tolerance radius. So by (1) we can calculate the similarity score of D as follows:

$$\mathrm{Sim}_D = \frac{1}{|D|} \sum_{p \in D} \mathrm{Hit}(p) \quad (2)$$

where |D| is the total numbers of edgels in D and is normalizing the similarity score. This normalization term compensates the improper advantage of the photo with larger number of edgels.

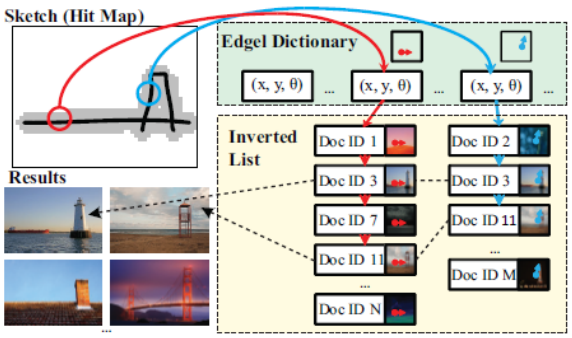

Figure 4. Illustration of the edgel dictionary and the matching process. By example, if the pixel in the red circle in the sketch matches one entry of the edgel dictionary, the similarities of images under the entry will increase by 1.

2.2 Indexing for Sketch-Image Matching

The algorithm above is very suitable for indexing. The similarities of images with the sketch can be calculated in a short time through an inverted index strategy as proposed in [2]. An edgel p = (x, y, e) from an image could have a hit only when there's an edgel q = (x, y, e) in the sketch. Thus, if considering each edgel as a "word", we could build an edgel dictionary for the whole database, in which each entry is represented by a triple (x, y, e). In our work, the resolution of our sketch panel is 200 × 200, and the angles of edgels are equally quantified into 8 categories, i.e. 0°~22.5°/180°~202.5°, 22.5°~45°/202.5°~225°, …, and 157.5°~180°/337.5°~360°. Therefore, our dictionary has sketch comes in, if an edgel q in the sketch matches an edgel entry p in the dictionary, the similarity scores of all photos under p will increment by 1.

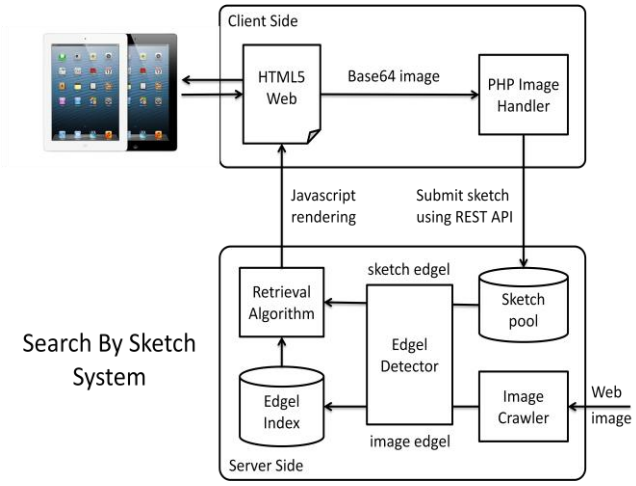

Figure 5. Search by Sketch system framework, including a web-based client side and a "Restful" server side.

3. System Overview

3.1 Client Side

The HTML code of the client side is based on an HTML5 canvas sample [5]. We modified it for encoding the sketch into base64 and submit it to server side. The PHP image handler is to help the uploading process. After we submit the sketch to server side, client side will wait for server instruction for a few seconds. The loading view is enabled now. The Javascript code in the client side is used to receive the retrieval result (JSON format returned from server side), and render it onto the html5 web page display.

3.2 Server Side

Our server API follows the Restful standard to make the communication platform independent. We have an image crawler at server side to crawl necessary images from web. We implemented two crawlers: (1) Google Image Crawler. (2) Flickr Crawler. After that, we run the edgel detector to process all the images we have, then built an edgel indexing for online fast retrieval.

Now once the sketch been submitted to the server, we will process the sketch by running edgel detector as well, match it with all the database entries to find similar images based on edgel similarity, and then return the retrieved result back to the client using JSON format.

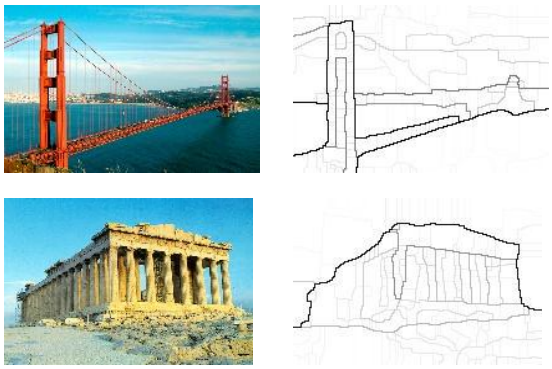

Figure 6. Examples of the down sampled photos and their contour images. Notice that there are stronger contours (black edges) and weaker contours (gray edges). The darker edges have higher possibility of being a significant contour. Thus, we use a threshold of pixel intensity to preserve the key edges that contain contours of better quality, and to filter out the grayer edges which are actually noises for the retrieval stage. Our experiment shows that preserving the top 1.5% of the strongest edgels in each contour image yields the best retrieval results.

4. Retrieval Process

The retrieval process can be divided into two stages, the preprocessing stage and the online stage. The preprocessing stage creates the edgel dictionary, and the online stage matches every edgel in the sketch with the entries of the dictionary to retrieve photos.

4.1 The Preprocessing Stage

In the preprocessing stage, we collected thousands of world landmark photos from Google and from Flickr. These colorful photos were first down sampled to a size whose width or length is at most 200 pixels. This method accelerates the retrieval time while preserving sufficient details for matching sketches. Then, these small photos were converted into image contours by the Berkeley Contour Detector [3]. The Berkeley Contour Detector not only extracts contour edges from the photo, but also preserves the angle of each edgel, which we can directly use to create the entries of the edgel dictionary. Finally, after filtering out noisy edges in the contours images, we built the edgel dictionary as described in Section 2 with the remaining contour edgels of greater significance (Fig.6). All the tasks here were done by codes in C++/OpenCV or Matlab.

4.2 The Online Stage

When a user draws a sketch on the drawing panel in our webpage and clicks the search button, the client end will transfer the sketch image to the server. Next, the server extracts all the edgels in the sketch, matches them with the entries of the edgel dictionary, and calculates the similarity score of each photo in the database as described in Section 2. The code here is written in HTML 5 and Python/OpenCV. After optimizing the code, the retrieval time is about 0.8 seconds.

5. Conclusion

We implemented a complete system capable of retrieving images by sketch query. The database we collected consists of worldwide famous landmarks and contains about 7,000 images and 200 landmarks. The performance depends primarily on the quality of the sketch by the user, but the chance of correct matching was very high during the live demo section. The most important feature of this system is that it provides a retrieval interface which breaks the limitation of texting, and therefore enables the user to find the images in his/her mind more directly and intuitively. In addition, according to our observation, it takes a user 5 seconds in average to draw a sketch, which is very close to the time required by typing a description of an image, and the retrieval results are shown on the webpage in about 2 seconds. As a result, the Search by Sketch system is very worthy of investigation and further development. And there is absolutely more fun in drawing sketches compared with typing texts.

In addition, the algorithm we implemented is very suitable for GPU acceleration. For example, we can divide the entries of the dictionary into groups by the number of blocks. (Notice that the maximum number of entries is 200 × 200 × 8 = 320,000) Each entry is computed by one thread. The similarity score array is allocated at the global memory, and, in order to avoid the conflicts of the write operation by the threads, its length will be the same as the sum of the sizes of all inverted lists in the dictionary. This part of GPU acceleration is left for future work.

Reference

[1] Yang Cao, Hai Wang, Changhu Wang, Zhiwei Li, Liqing Zhang, and Lei Zhang, MindFinder: Interactive Sketch-based Image Search on Millions of Images, in ACM Multimedia 2010 International Conference, 2010.
[2] Yang Cao, Changhu Wang, Liqing Zhang, and Lei Zhang, Edgel Inverted Index for Large-Scale Sketch-based Image Search, in CVPR 2011.
[3] Contour Detection and Hierarchical Image Segmentation. P. Arbelaez, M. Maire, C. Fowlkes and J. Malik.IEEE TPAMI, May 2011.
[4] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Modelbased hand tracking using a hierarchical bayesian filter. PAMI, 2006.
[5] Create a Drawing App with HTML5 Canvas and JavaScript, William Malone.
http://www.williammalone.com/articles/create-html5-canvas-javascript-drawing-app/