Sketch-based Image Retrieval on Mobile Devices Using Compact Hash Bits

Kai-Yu Tseng
National Taiwan University
Taipei, Taiwan
cfthn0357@gmail.com

Yen-Liang Lin
National Taiwan University
Taipei, Taiwan
r96944029@ntu.edu.tw

Yu-Hsiu Chen
National Taiwan University
Taipei, Taiwan
dennis2030@gmail.com

Winston H. Hsu
National Taiwan University
Taipei, Taiwan
winston@csie.ntu.edu.tw

ABSTRACT

The advent of touch panels in mobile devices has provided a good platform for mobile sketch search. However, most of the previous sketch image retrieval systems usually adopt an inverted index structure on large-scale image database, which is formidable to be operated in the limited memory of mobile devices. In this paper, we propose a novel approach to address these challenges. First, we effectively utilize distance transform (DT) features to bridge the gap between query sketches and natural images. Then these high-dimensional DT features are further projected to more compact binary hash bits. The experimental results show that our method achieves very competitive retrieval performance with MindFinder approach [2] but only requires much less memory storage (e.g., our method only requires 3% of total memory storage of MindFinder in 2.1 million images). Due to its low consumption of memory, the whole system can independently operate on the mobile devices.

Categories and Subject Descriptors

I.3.8 [**Computing Methodologies**]: Computer Graphics – *Applications*

General Terms

Performance, Experimentation.

Keywords

Sketch, Hash.

1. INTRODUCTION

Recently, image retrieval over large-scale photo collections is one of the key techniques for managing the exponentially growing media collections. However, most images in the websites or user collections do not contain tags so content-based image retrieval (CBIR) was proposed to solve this problem. In CBIR systems, users have to provide a sample image for searching and the use of touch screens on mobile devices has become increasingly common, which offers a big opportunity to search images based on those simple sketches as sample images (cf. Figure 1). Besides, users may have thousands of photos in a local album and are not willing to upload to any cloud computing service. The traditional sketch retrieval systems are provided on servers and usually use the inverted index structure, which is formidable to be operated in the limited memory of mobile devices (e.g., memory storage is about 1GB and 512MB for tablets and smartphones respectively).

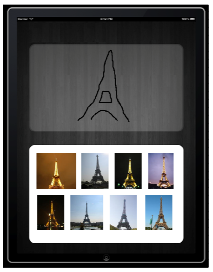


Figure 1: Example results of our designed system. We propose a mobile sketch search system using compact hash bits. Due to its low consumption of memory, the whole system can independently operate on the mobile devices.

To overcome these problems, we propose a mobile sketch system[1] that not only reduces memory storage but also ensures its competitive search accuracy and efficiency. First, we use distance transform (DT) features to approximate the shape information of images. Second, we project these high-dimensional DT features to more compact binary hash bits by very sparse random projection (VSRP) method. Due to its low consumption of memory, the system can efficiently operate on the mobile platform alone. Otherwise, even though the images are stored on server, our system can allocate some computations to the mobile terminal will save the operational cost of server and compress the features to compact hash bits for efficient transmission.

In Section 2, we introduce the previous works and point out the differences between them. Our observations and methods are described in Section 3. Section 4 will show our experimental results, followed by the conclusions in Section 5.

2. RELATED WORK

Most of the previous researches on sketch-based image retrieval mainly focus on the study of extracting effective features to better match a rough sketch to natural images. M. Eitz et al. [1] survey many different sketch features and compare their performances. However, the memory storage issue is not well discussed. Y. Cao et al. [2] propose the MindFinder system to index large-scale image dataset. They use every edge pixel in the dataset images as an index point and consider an edge pixel locating on the boundary as a hit. In order to endure slight translation, they divide

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

[1] Demo video is available at: http://youtu.be/wK6dlt7w4E8

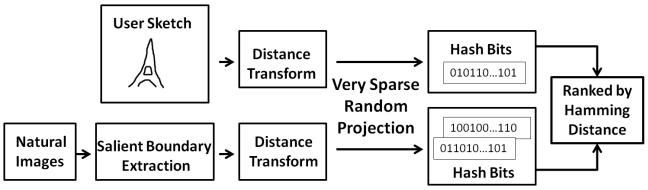


Figure 2: Framework of the proposed system. In the offline stage, we obtain the compact hash bits of all images in the dataset and store them on the mobile devices. In the online stage, we can perform the similar steps to obtain the hash bits of user sketch. Then, we can rank the dataset images by their hamming distance.

the image into six directions and find the hit points within a predefined tolerance. Their method preserves the spatial information and uses an inverted index in large-scale image dataset for efficient retrieval. Y. J. Lee et al. [7] propose the ShadowDraw system that guides user to draw better sketches. They extract BiCE binary descriptors from images and use min-hash function to randomly permute the descriptors. Each descriptor is translated to *n* min-hash values of size *k* and indexed by an inverted look-up table.

However, those prior works are usually based on the inverted index structure, which grows rapidly in data size. Therefore, they usually put the inverted index on the server side. Different from those prior works, we use more compact hash bits to reduce memory usage while keeping the advantage from distance transform features to effectively capture the shape information of the salient objects. The experimental results show that our method is comparable to prior works (e.g., MindFinder) but consumes much less memory.

3. METHOD

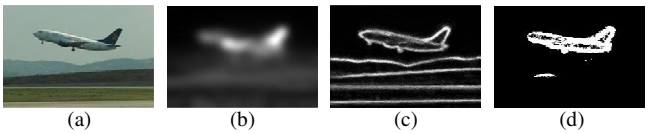
The overall framework of the proposed method is illustrated in Figure 2. First, in order to bridge the gap between sketches and natural images, we extract the salient boundaries of natural images. Second, we extract the DT features from salient boundary images. Third, we apply very sparse random projection to transform high-dimensional DT features to compact hash bits. Finally, we rank the dataset images based on their hamming distances to the query. Due to the lightweight property of hash bits, we can efficiently store them in the limited memory of mobile devices.

3.1 Extract Salient Boundary

When asked to draw the Eiffel Tower, most people will only draw the most impressive parts and ignore those detailed background such as the horizon, trees or buildings (cf. Figure 1). However, natural images usually contain cluttered background and complex texture. To better compare rough sketches to natural images, extracting image boundaries is necessary, and we also extract salient parts from natural images to remove background noise.

Given a natural image, we first resize the image with the longer side being 200 pixels, and then extract salient part *S* by method [5] (cf. Figure 3(b)) and boundary *B* by method [3] (cf. Figure 3(c)). To further remove the background noise in *B*, we set a threshold for salient parts (cf. Eq. (1)) to get image *B'* that only retains main boundaries of an object and binarize *B'* to get image *I* (cf. Figure 3(d))

$$I = \begin{cases} 1 & \text{if } B' \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{that } B' = \begin{cases} B & \text{if } S > \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$



(a) (b) (c) (d)

Figure 3: Examples of extracting salient boundaries from an airplane image. (a) Original natural image. (b) Salient image, where salient parts appear brighter. (c) Boundary image of (a). We filter out those low values in (c) and perform binarization to obtain the salient boundary image (d).

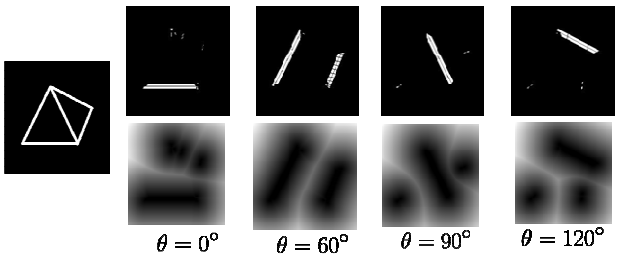


Figure 4: Examples of distance transform in different directions. Due to space limitation, we only show 4 dominant directions (first row) and their corresponding DT features (second row). Darker values in DT imply smaller distances to the nearest boundary points.

3.2 Distance Transform Features

Distance transform (DT) has been shown effective in representing the contours of objects. The DT features is a map where the each pixel records the distance from its nearest edge point. Therefore, we utilize DT features to effectively represent the shape information. However, DT is very sensitive to rotations and translations and most people are not experts of painting. For enduring more tolerance, we quantize the salient boundary image into six directions (cf. Figure 4). The DT features of each direction can be written as Eq. (2).

$$DT_{I_\theta}(p) = \min_{p' \in I_\theta} dist(p, p') \tag{2}$$

Where I_θ is the θ-th direction image, $\theta \in \{0°, 30°, ..., 150°\}$. For each point $p \in (x, y)$, we compute its L_2 distance to the nearest boundary point p' in θ-th direction image. DT features of each direction can be represented by a single row vector and the final DT features are the concatenation of six row vectors.

The experimental results show performance of using DT features is competitive with MindFinder approach (c.f. Figure 6). However, the high dimensionality of DT features is still a problem (e.g. if the image size is 200×200 and has six planes, the dimension of DT features is about 240,000). Keeping these high dimensional features in the memory is not feasible and cannot perform real-time search in large-scale photos database without indexing. In next section, we will review the problem and explain the proposed method in details.

3.3 Hash Bits

To overcome the problem of restricted amounts of memory on mobile devices, we project the high-dimensional DT features to more compact hash bits and shrink the size of the projection matrix enough to save memory consumption.

Given an original *D*-dimensional X_D, we can reduce the dimension of X_D to *K*-dimensional R_K ($K \ll D$) by multiplying a $K \times D$

dimensional projection matrix $P_{K\times D}$. The projection can be formulated as Eq. (3).

$$R_K = P_{K\times D} X_{D,} \tag{3}$$

where matrix $P_{K\times D}$ is generated by random projection (RP), whose elements are generated from a Gaussian distribution. We choose RP instead of principal component analysis (PCA) because most directions are orthogonal in a high dimensional space [6] and PCA is computationally expensive and memory exhausted.

We further binarize the values of R_K vector to get compact hash bits. Although the high dimensional features are translated to compact hash bits, the dimension of the random projection matrix is still too large to keep it in the memory (e.g., in our case, the original dimension D is 240,000 and the dimension of hash bits K is 540. Then the dimension of random projection matrix will be $O(240,000\times 540)$, i.e., 494MB).

For reducing the size of random projection matrix, we use very sparse random projection instead of random projection. P. Li et al. [4] show that the Gaussian distribution can be approximated by a zero mean, unit variance distribution. Therefore, each element of the very sparse random projection matrix is generated by Eq. (4):

$$P_{ij} = \sqrt{D} \times \begin{cases} +1 & \text{with probability } 1/2\sqrt{D} \\ 0 & \text{with probability } 1-1/\sqrt{D} \\ -1 & \text{with probability } 1/2\sqrt{D} \end{cases} \tag{4}$$

Most of the elements in the projection matrix are almost zero, thus we can store it with a very sparse matrix representation. It is only $1/\sqrt{D}$ times the size of the original random projection matrix (e.g., the size of the projection matrix will be reduced from $O(240,000\times 540)$ to $O(490\times 540)$, i.e., 494MB to 1MB).

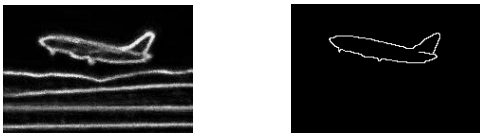
Finally, we assign the positive values of R_K to 1 and negative values to 0. Every image can be represented by a sequence of binary codes. Then, we can rank dataset images by their hamming distances to the query sketch.

4. Experiments

To evaluate the performance of proposed features and matching algorithms, we experiment on two self-constructed datasets and compare with two different types of queries; natural boundary and manual sketch (cf. Figure 5). To validate the usefulness of distance transform features, we first evaluate the performance by using natural boundaries extracted by algorithm [3]. However, there are differences between natural boundaries and real user sketches. In order to better approximate real user sketches, manual sketches (cf. Figure 5(b)) is generated by manually tracing the contours of salient objects in the natural boundary images.

100 images from Caltech: The images in this dataset are selected from Clatech256. There are ten classes and each class contains 10 images without cluttered background. In this dataset, we do not extract salient parts because the objects occupy the main portion of the images. We evaluate the retrieval performance via mean average precision (MAP) of top 10 results.

Flickr 10k: We crawl 10k images from Flickr website and equally divide them into 10 classes. Due to the cluttered background within these images, we extract the salient boundaries from them. We randomly select 10 images from every class and totally have 100 query images. We evaluate the retrieval performance via mean average precision (MAP) of top 50 results.



(a) (b)

Figure 5: Two query types. (a) Natural boundary extracted by algorithm [3]. (b) Manual sketch generated by manually tracing the contour of salient objects of (a).

4.1 Performance of Distance Transform

To verify the usefulness of distance transform features, we compare the performance between MindFinder and distance transform features, of which the results are ranked with L1 distances.

In the 100-images dataset (cf. Figure 6(a)), we find that the distance transform is better than the prior work. Because of the absence of background influences, distance transform can be more accurate to capture the shape information. The performance of manual sketch is slightly higher than natural boundary because the manual sketch only retains the most salient parts of the objects.

In the Flickr 10k dataset (cf. Figure 6(b)), the performance of distance transform is still better than the prior work, since we have extracted the salient parts from images to avoid the disturbance from the background noises.

Such experimental results verify that distance transform is a useful feature for sketch image retrieval.

4.2 Performance of Random Projection

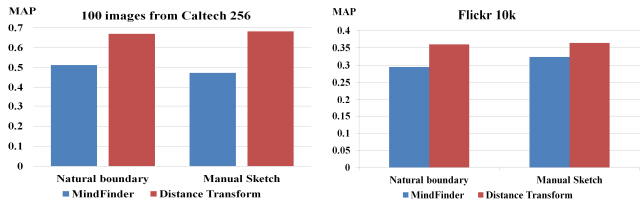
We compare the performance of original high-dimensional DT features with L1 distance and different lengths of hash bits (e.g., 60~540 bits) generated by random projection (RP) or very sparse random projection (VSRP) in the two datasets (cf. Figure 7). The curves of RP and VSRP are not smooth because the elements of projection matrixes are re-generated for each length of hash bits. We observe that RP and VSRP approaches gradually match the performance of DT (L1) with the growth of bit number, which supports the feasibility of using more compact hash bits to approximate the performance of high dimensional DT features.

4.3 Size Reduction

We compare the memory usage of different methods: inverted index used in MindFinder, random projection (RP) and very sparse random projection (VSRP) (cf. Table 1).

Table 1. Comparisons of the memory usage for different methods. MindFinder system needs 292*MB* and 3.7*GB* memory space to store inverted index in 10k images and 2.1 million images respectively. However, our method (rightmost column) only requires 1.64*MB* and 136*MB* memory usage respectively. The result size of RP and VSRP contains two parts: the size of hash bits and the size of projection matrix.

Dataset	Mind Finder[2]	RP	VSRP
10*k* images	292*MB*	0.64*MB* + 494*MB* = 494.64*MB*	0.64*MB* + 1*MB* = **1.64*MB***
2.1million images	3.7*GB*	135*MB* + 494*MB* = 629*MB*	135*MB* +1*MB* = **136*MB***



(a) (b)

Figure 6: Performance comparison of MindFinder and DT. The MAP of DT is higher than MindFinder on two datasets. The relative improvements are 16% on 100 images from Caltech 256 (a) and 4% on Flickr 10k dataset (b) respectively. The performance of manual sketch is higher than the other query types, because the manual sketch only retains the most salient parts of the objects.

In the Flickr 10k dataset, there are 76,674,976 individual points (*Edgels*) in the inverted index structure of MindFinder system. For each index point, they store a list of all corresponding image IDs. Each image IDs is stored by 4 bytes integer. Therefore, the total size of the inverted index is $76,674,976 \times 4 / (1,024^2) \approx 292MB$. In our method, the cost contains two parts, hash bits and projection matrix. Every image is represented by 540 hash bits so the total size of hash bits is $540 \times 10k \approx 0.64MB$. Due to the high dimension of original DT features, the size of RP projection matrix is larger than MindFinder ($540 \times 240k \times 4 / (1,024^2) \approx 494MB$). To overcome this problem, we use very sparse random projection instead to further reduce the size of projection matrix to $490 \times 540 \times 4 \approx 1MB$. In 2.1 million images dataset, the storage of MindFinder is 3.7*GB* reported from [2], but our method only requires 136 *MB*.

Additionally, we also compare our methods with ShadowDraw in their 30k-image case [7]. ShadowDraw needs 850*MB* memory to store an inverted look-up table, but ours only needs 1.9*MB* memory to store hash bits.

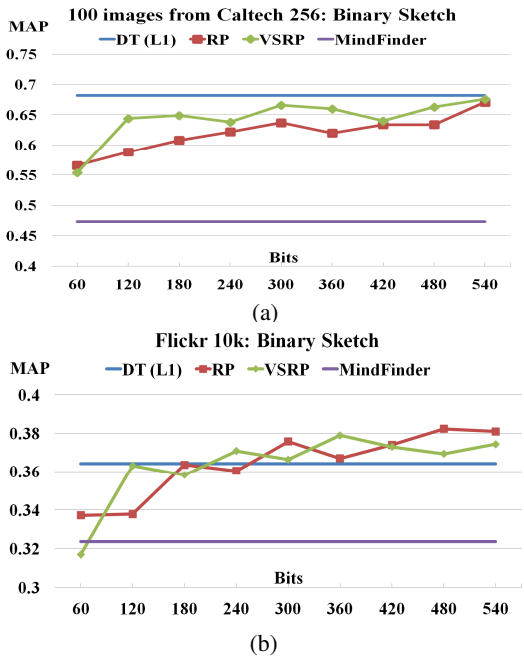
Compact hash bits and very sparse random projection reduce the memory usage significantly, which makes the system feasible to operate on the mobile platform.

4.4 Re-rank

Spatial constraint exists in distance transform. If the salient object is small or have large range translation, DT features will obtain poor matching results. To handle this problem, we re-rank top 100 searching results via fast chamfer matching methods [8] based on a sliding window approach. In Flickr 10k dataset, the performance (MAP) is increased by 0.18% and 2.58% in natural boundary and manual sketch query respectively.

5. Conclusions and Future Work

In this paper, we build a sketch-based image search framework on mobile devices. We propose to use distance transform features (DT) to precisely capture the shape information of the sketch image, and reduce the memory storage by further projecting the high-dimensional DT to compact hash bits. The experimental results show that compact hash bits achieve higher performance than the prior works and consume much less memory storage. We also verify distance transform is a feasible feature for sketch image retrieval and analyze the performance of varying bits of binary hash bits.



(a) (b)

Figure 7: Performance comparison of DT, RP, VSRP and MindFinder. (a) Results on 100 images from Caltech dataset. (b) Results on Flicker 10k dataset. As shown in (a) and (b), RP and VSRP get better performance as the number of bits growths and eventually approach to the performance of DT (L1).

For the future work, we are to seek further improvements by knowledge-based hashing methods, and investigate others features potential to be invariant to translation, rotation and scale variations.

6. REFERENCES

[1] M. Eitz, K. Hildebrand, T. Boubekeur and M. Alexa, An evaluation of descriptors for large-scale image retrieval from sketched feature lines, in Computers & Graphics, 2010

[2] Y. Cao, C. Wang, L. Zhang, and L. Zhang, Edgel Inverted Index for Large-Scale Sketch-based Image Search, in CVPR 2011.

[3] P. Dollar, Z. Tu, and S. Belongie, Supervised Learning of Edges and Object Boundaries, in CVPR 2006.

[4] P. Li, T. J. Hastie and K. W. Church, Very sparse random projections, in KDD 2006.

[5] J. Harel, C. Koch, and P. Perona, Graph-Based Visual Saliency, in NIPS 2006.

[6] R. Hecht-Nielsen. Context vectors: general purpose approximate meaning representations self-organized from raw data, in Computational Intelligence: Imitating Life, 1994

[7] Y. J. Lee, C. L. Zitnick, and M. Cohen, ShadowDraw: Real-Time User Guidance for Freehand Drawing, in SIGGRAPH 2011.

[8] M. Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, Fast Directional Chamfer Matching, in CVPR 2010.