A Sketch-Based User Interface for Image Search Using Sample Photos

Hitoshi Sugimura, Hayato Tsukiji, Mizuki Kumada, Toshiya Iiba,
and Kosuke Takano(✉)

Department of Information and Computer Sciences,
Kanagawa Institute of Technology, Atsugi, Japan
{s1121157, s1321033, s1321155,
s1221125}@ccy.kanagawa-it.ac.jp,
takano@ic.kanagawa-it.ac.jp

Abstract. In this paper, we present a sketch-based user interface for a content-based image search using sample photos. Our sketch-based user interface diverts a part of the images and their color extracted from sample photos, which are obtained from image resources, such as an image search result, to a sketch, and supports a user to draw a more practical sketch as a query. This feature of the proposed sketch-based user interface allows a user to make a sketch-based query according to the thought and image of the user by painting and decorating the sketch using fragments of images snipped from sample photos and their colors. Thus, the proposed sketch-based user interface can help a user to make a sketch for the image retrieval by leveraging objects and colors that actually appear in images and photos. In the experiment using our prototype, we evaluated the feasibility of our approach.

Keywords: Content-based image retrieval · Interaction design · Query by sketch · User interface · Sample photos

1 Introduction

In the field of the content-based image retrieval (CBIR) [1], a sketch-based query is a query for an image search that is performed by drawing a sketch that represents the user's thought and image. A sketch-based query allows us to obtain the desired images by intuitively sketching what we want to query, even if we cannot have any query keywords.

However, we often redraw our sketch by improving the colors and shapes until a search system returns appropriate images. A user's repeated redrawing would decrease the convenience of the sketch-based query. Furthermore, even if we sketch well what we want to query, we might not find the objective images when there is no image that has the similar features of color and shape as the user's sketch in an image database. For example, we usually draw a sketch using a few colors, not like a photo where various colors appear. Therefore, there is a possibility that appropriate images will not be obtained in the search result when using a sketch-based query because the sketch does not include the colors that appear in target images in the image database (Fig. 5).

S. Yamamoto (Ed.): HIMI 2016, Part I, LNCS 9734, pp. 361–370, 2016.
DOI: 10.1007/978-3-319-40349-6_34

In this study, we present a sketch-based user interface (sketch UI) for an image search using sample photos. Our sketch UI provides the function of diverting a part of the images and their colors extracted from sample photos, which are obtained from image resources, such as an image search result, to a sketch and helps a user to draw a more practical sketch as a query. This feature of the proposed sketch UI allows a user to make a sketch-based query according to the thought and imagination of the user by painting and decorating the sketch using fragments of images snipped from sample photos and their colors. Thus, the proposed sketch UI can help a user to make a sketch for the image retrieval by leveraging objects and colors that actually appear in images and photos.

In this study, we implemented a prototype of an image search system with a sketch-based user interface that is available through a Web browser. To achieve basic functionality for CBIR, we used a histogram intersection method [1] and a perceptual hash method [3] for the color and outline comparison between images, respectively. In the experiment using our prototype, we evaluated the feasibility of our approach. In the experiment, we prepared 97 images and 8 patterns of sketch-based queries to compare our proposed sketch UI with a baseline approach with a simple sketch functionality. The experimental results showed that our proposed sketch-based user interface could reduce the cost of making a sketch-based query and increase the opportunity for a searcher to obtain the desired images from an image database, especially when a color feature is a significant factor for the image search.

2 Motivating Example

The query-by-sketch image retrieval system [2, 6] allows a user to draw a sketch and retrieve the desired images based on the sketch by applying the content-based image retrieval (CBIR) method [4, 5]. One of the merits of query-by-sketch image retrieval is the flexibility to make an image-based query; however, there is a possibility that the features of a sketched image input as a query do not match those of the target images in the database in the retrieval process because a user freely chooses colors and draws various types of shapes in the sketch. As a result, the image retrieval system cannot return the desired images to the user according to the sketch-based query.

A relevance feedback method for the query-by-sketch image retrieval is proposed, where a searcher obtains objective images by selecting relevant images in the search result and adding their features to the original query [2, 7, 8]. In [2, 8], the semantics extraction from query images using the relevance feedback was studied. In [2], a query-by-sketch image retrieval method was proposed to reduce the semantic gap between low-level features and high-level semantics by adopting relevance feedback. In addition, He et al. presented a leaning method for capturing the semantics of images through the relevance feedback [8]. Furthermore, in [7], Shen et al. proposed a method for optimizing the relevance feedback search by query prediction based on the previously executed iterations' queries. However, these methods cannot always allow us to represent our thoughts and images as a query because the composed image query consists of the features of whole images that the searcher selects as relevant. For example, suppose that a searcher inputs a sketch of an apple as a query and selects an image where one apple appears as a relevant image from the search result through the relevance feedback process. In this case, the image retrieval system will return images that are similar to the selected apple image in the second search result. In this example, when the searcher wants to obtain an image of a tree with many apples, the desired images will not appear in the higher ranking of the search result, unless an image with a feature of a tree is also associated with the composed image query in the relevance feedback.

We incorporate two functions into our sketch UI using sample photos in a target image database: (1) a function of the color palette creation based on colors appearing in the sample photos, and (2) a function of the sketch patch using image fragments snipped from the sample photos. Figures 1 and 2 show examples of these two functions. In the upper illustration of Fig. 1, a user chooses colors in his/her head to sketch an apple; however, as a result, a system might return irrelevant images in the search result. On the other hand, in the lower illustration of Fig. 1, we can see that the user can obtain the proper images of apples in the search result by painting the sketch in a red color, and a shine color appeared in a sample photo of an apple. To support this function, the proposed sketch UI dynamically creates a color palette by extracting colors appearing in sample photos so that a user can choose the colors in the photo for the sketch.

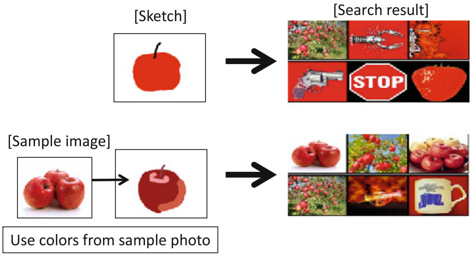


Fig. 1. Example of color extraction from the sample image (Color figure online)

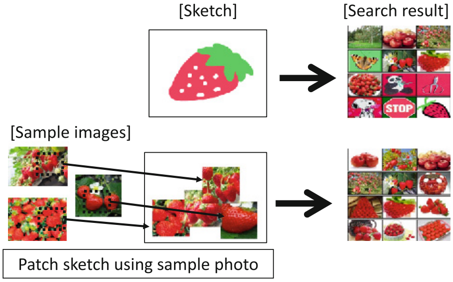


Fig. 2. Example of the sketch patch using a sample image (Color figure online)

When a user does not have a talent for painting, the user may not draw a good sketch and cannot obtain the desired images in the retrieval process, as shown in the upper illustration of Fig. 2. To help a user to draw an expressive sketch, our sketch UI provides the sketch patch function to extract what the user wants to draw by snipping parts of images from sample photos and adding them to the user's sketch. The lower illustration of Fig. 2 shows an example of the sketch patch function. In this example, the desired images of strawberries are retrieved in the search result by extracting partial image fragments of strawberries from three sample photos and adding them to the sketch.

3 Proposed System

Figure 3 shows an overview of our image search system with the proposed sketched-based user interface (sketch UI). Our image search system allows a user to input a sketch-based image as a query and return the search result from an image database by applying the conventional CBIR method. Figure 4 shows the design of our sketch UI. The sketch UI mainly consists of (A) color palette, (B) sketch canvas, (C) search result, (D) image pool for sample photos, and (E) query history.

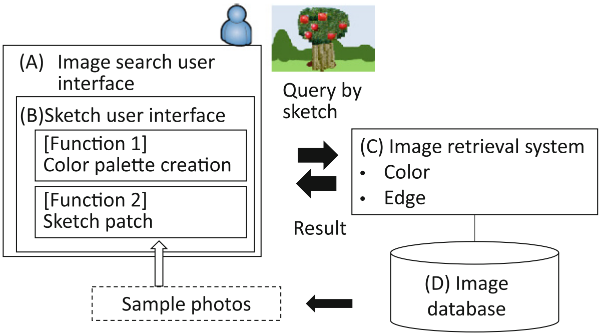

(A) Image search user interface
(B)Sketch user interface
[Function 1]
Color palette creation
[Function 2]
Sketch patch
Query by sketch
(C) Image retrieval system
- Color
- Edge
Result
(D) Image database
Sample photos

Fig. 3. System overview

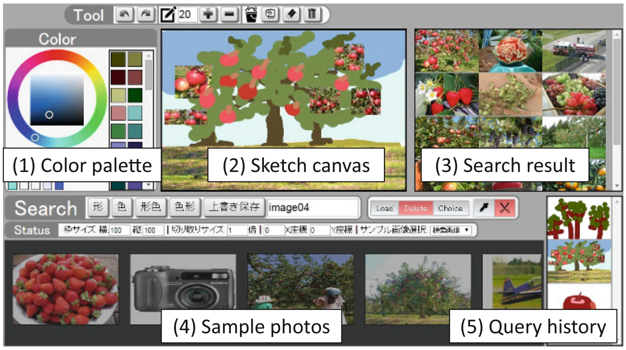


Fig. 4. Image search interface

Using the sketch UI, a user first draws a sketch on the sketch canvas by choosing colors from the color palette. Then, an image search result is shown in the search result area according to the user's sketch. If there are images that the user wants to divert in the search result, those images can be copied to the image pools for use as sample photos. Here, the color list on the color palette is updated based on the colors in the sample photos. Furthermore, the user extracts image fragments that he/she wants to use by snipping them from the sample photos and adding them to the sketch. By repeating these sketch steps, the user can obtain the desired images from the image database.

3.1 A Function of Color Palette Creation

When a user draws a sketch for the image search, the user does not know what colors should be used to efficiently obtaining the objective images. The function of the color palette creation helps a user to choose proper colors for drawing a sketch by extracting colors actually appearing in a set of sample photos in advance and showing them in the color palette.

The color palette consists of a basic color list and a sample photo's color list. The sample photo's color list on the color palette is updated based on the colors appearing in the sample photos. Each sample photo in the image pool is extracted to n colors c1, c2, c3,…, cn. Then, the number of pixels for each n colors is calculated for all sample photos, and the colors on the sample photo's color list are sorted according to the pixel number. Colors that are not used in the sample photos do not appear in the sample photo's color list. In addition, the initial colors on the sample photo's color list are extracted from all target images in the image database.

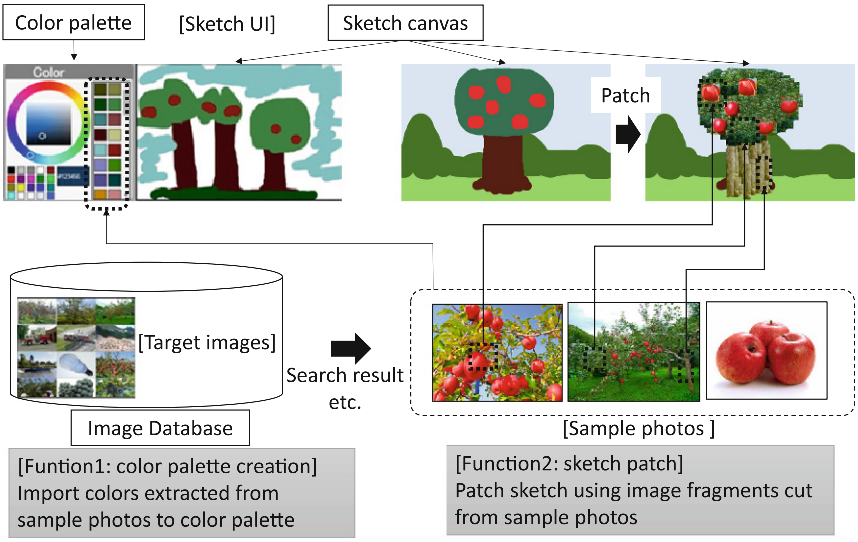
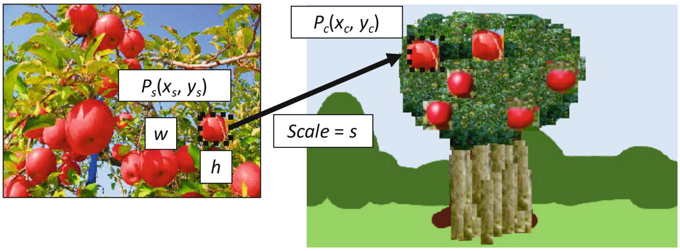

Color palette
[Sketch UI]
Sketch canvas
Color
Patch
[Target images]
Search result
etc.
Image Database
[Sample photos]
[Funtion1: color palette creation]
Import colors extracted from
sample photos to color palette
[Function2: sketch patch]
Patch sketch using image fragments cut
from sample photos

Fig. 5. Main functions of the sketch user interface

3.2 Function of the Sketch Patch

The sketch patch helps a user draw a practical sketch by extracting regions that the user wants to copy by snipping them from a sample photo and adding them to the user's own sketch. First, the user selects a point (x_c, y_c) where the user wants to patch the sketch on the canvas; then, the user selects a region from a sample photo. Figure 6 shows an example where a user selects a rectangular region, where top-left coordinate is a point (x_s, y_s) and the width and height are w and h, from a sample photo. In this example, the user patches the snipped regions of apples with a scale s to the specified point (x_c, y_c) on the canvas.



[sample photo on image pool] [sketch on canvas]

Fig. 6. Example of the sketch patch using a sample photo

3.3 Image Retrieval Method

We apply a conventional image retrieval method for the CBIR. First, we calculate the edge similarity using a histogram intersection method [1] as follows:

Step 1: Images are reduced to n-colors c_1, c_2, c_3,..., c_n

Step 2: A histogram h is created for each image based on the number of pixels for the n-colors

Step 3: The similarity of colors between a sketch image I_s and a target image I_t is calculated using the histogram intersection method. Suppose that the score of the similarity of colors is S_c, and the histograms for I_s and I_t are h_{s} and h_{t}, respectively:

$$S_c(I_s, I_t) = \frac{\sum_{i=0}^{n} \min(h_s[i], h_t[i])}{\sum_{i=0}^{n} h_s[i]} \tag{1}$$

Next, to calculate the edge similarity between two images, we apply an average hash method [3] and a hamming distance, as follows:

Step 1: An image is resized to a $n \times n$ pixel image, and the color of the image is converted to a grayscale color image

Step 2: The average value of the grayscale color values is calculated for all pixels in the image

Step 3: When the grayscale value of a pixel in the image is greater than the average value, the pixel is set to 1; otherwise, the pixel is set to 0. Thus, an $n \times n$ bit array is created for the $n \times n$ pixel image

Step 4: The hamming distance between two bit arrays B_s and B_t for a sketch image I_s and a target image I_t is applied to calculate the score S_e for the edge similarity between I_s and I_t:

$$S_e(I_s, I_t) = hamming\_distance(B_s, B_t) \tag{2}$$

Finally, we calculate the similarity score $S_{c,e}$ between a sketch image I_s and a target image I_t in terms of color and edge using the color score S_c and edge score S_e:

$$S_{c,e}(I_s, I_t) = \alpha S_c + (1 - \alpha) S_e \tag{3}$$

where $\alpha (0 \leq \alpha \leq 1)$ is a parameter for weighting each score.

4 Experiments

4.1 Experimental Environment and Method

In the experiment using our prototype, we evaluated the feasibility of our proposed method. We prepared 97 images, including 10 correct images of a tree with apples, for the evaluation. In addition, using three sample photos, as shown in Fig. 6, we sketched 8 patterns for image queries (Fig. 7). To make each sketch, we did not change the shapes of a tree with apples and the other background objects.

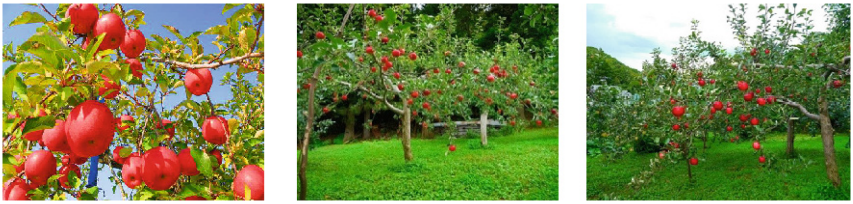
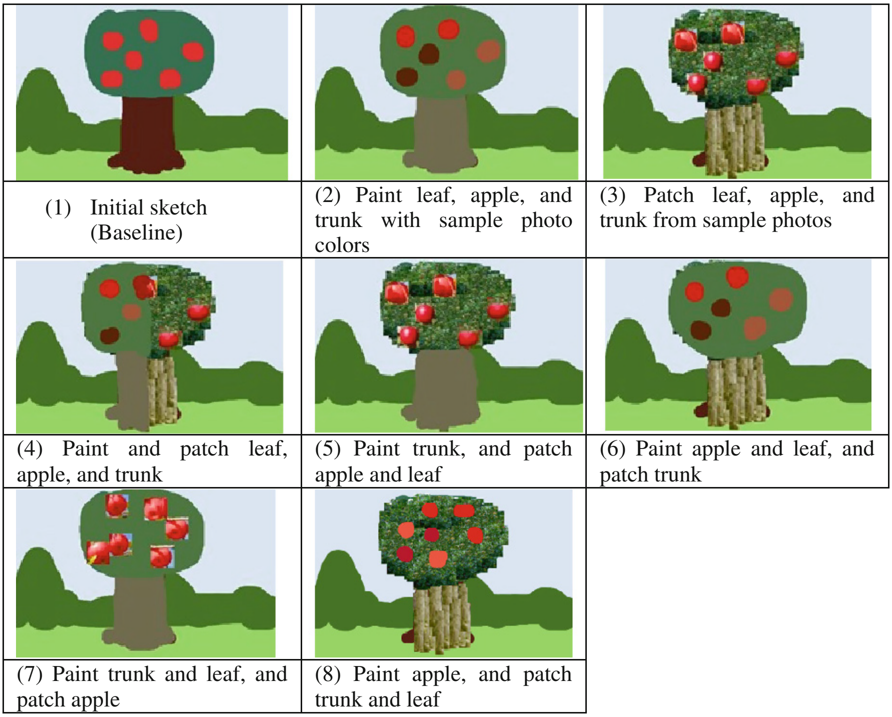


Fig. 7. Sample images

In Fig. 7, the pattern (1) is an initial sketch image that is used as a baseline for the comparison, and the patterns (2) to (8) are drawn based on the combination of the proposed method. We compare each pattern of the sketch-based queries in terms of the recall-precision and the drawing cost of the sketch. In addition, we set the weighting parameter α to 0.2, 0.4, 0.6, and 0.8 in the Eq. (1).



(1) Initial sketch (Baseline)

(2) Paint leaf, apple, and trunk with sample photo colors

(3) Patch leaf, apple, and trunk from sample photos

(4) Paint and patch leaf, apple, and trunk

(5) Paint trunk, and patch apple and leaf

(6) Paint apple and leaf, and patch trunk

(7) Paint trunk and leaf, and patch apple

(8) Paint apple, and patch trunk and leaf

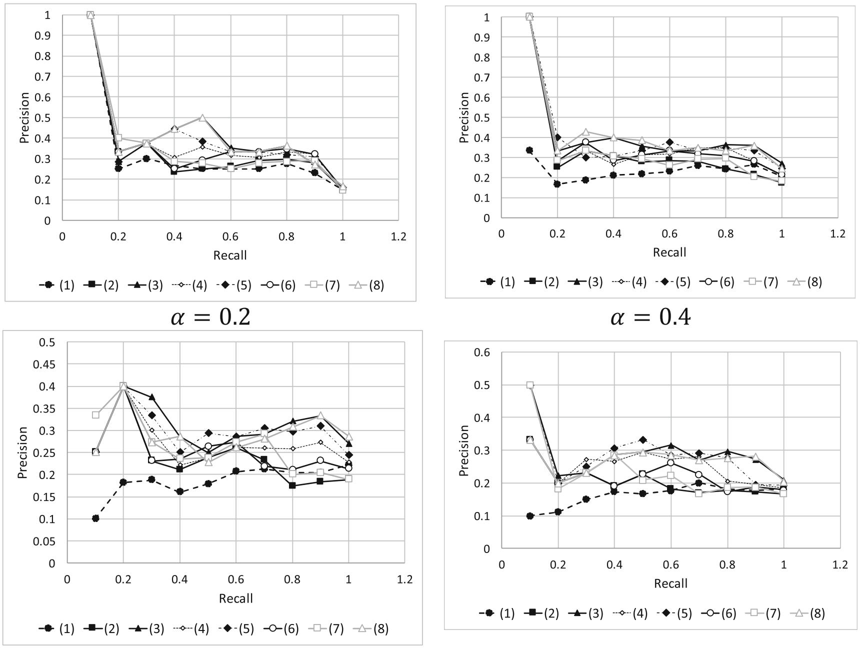
Fig. 8. Query sketch images

4.2 Experimental Results

Figure 9 shows the search results of a recall-precision graph for $\alpha = 0.2, 0.4, 0.6, 0.8$ using the 8 query sketch images shown in Fig. 8. Figure 9 shows that the precision of the search results using the query images (2) to (8) is better than the baseline query image (1) for all four cases.

In addition, when α is a larger value, the tendency of each recall-precision graph varies and the precision of the sketch images using the proposed method is much better than the baseline query image. This result indicates that the proposed sketch patch function is useful when we consider color a more important factor for the image search. At $\alpha = 0.2$ and 0.4, the precision of sketches (3), (5), and (7) is better; therefore, the color red of the apple images from the sample photos positively effects the search results. Meanwhile, when α is a smaller value, the tendency of each recall-precision graph becomes similar, and the shape in all sketched is almost the same.

Furthermore, in terms of the drawing cost, we found that it took a long time to complete a sketch when we used the sketch patch function. On the other hand, it took less time when we drew a sketch by choosing colors from sample photos. Therefore, to find sample photos that have effective colors, a user should draw a shape and paint it using some basic colors and then gradually increase the amount of image patches from



$\alpha = 0.2$ $\alpha = 0.4$

$\alpha = 0.6$ $\alpha = 0.8$

Fig. 9. Search results

the sample photos by leveraging our sketch patch function for retrieving the desired images in a shorter time.

5 Conclusion

In this study, we proposed a sketch-based query user interface (sketch UI) using sample photos to support a user's drawing for an image search. Our sketch UI provides two functions: color palette creation and sketch patching using sample photos. These two functions allow a user to efficiently draw a sketch using colors and parts of images extracted from sample photos. The experimental results using our prototype show that our sketch UI can improve the precision of the initial image search result. In terms of the cost of drawing a sketch, we confirmed that we can draw a sketch in a short time and can find the desired images using both colors and parts of the image from sample photos. In addition, a user can obtain the desired images in a shorter time by initially using some basic colors and then gradually increasing the amount of image patches from sample photos.

In our future work, we will improve the usability of the proposed sketch UI. In addition, we will evaluate the effectiveness of our sketch UI based on comparison with other image retrieval systems.

References

1. Swain, M.J., Ballard, D.H.: Color indexing. Int. J. Comput. Vis. **7**(1), 11–32 (1991)
2. Ohashi, G., Hisamori, T., Mochizuki, K.: Query-by-sketch image retrieval system using relevance feedback. J. Jpn. Soc. Fuzzy Theory Intell. Inform. **19**(5), 537–545 (2007)
3. Klinger, E., Starkweather, D.: The open source perceptual hash library. http://www.phash.org. (Accessed Dec 2015)
4. Ashley, J., Flickner, M., Hafner, J., Lee, D., Niblack, W., Petkovic, D.: The query by image content (QBIC) system. In: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data (SIGMOD 1995), p. 475 (1995)
5. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: state of the art and challenges. ACM Trans. Multimedia Comput. Commun. Appl. **2**(1), 1–19 (2006)
6. Springmann, M., Kopp, D., Schuldt, H.: QbS: searching for known images using user-drawn sketches. In: Proceedings of the International Conference on Multimedia Information Retrieval (MIR 2010), pp. 417–420 (2010)
7. Shen, H.T., Jiang, S., Tan, K.-L., Huang, Z., Zhou, X.: Speed up interactive image retrieval. Int. J. Very Large Data Bases Arch. (The VLDB J.) **18**(1), 329–343 (2009)
8. He, X., Ma, W.-Y., King, O., Li, M., Zhang, H.: Learning and inferring a semantic space from user's relevance feedback for image retrieval. In: Proceedings of the Tenth ACM International Conference on Multimedia (MULTIMEDIA 2002), pp. 343–346 (2002)