VIEWGLE: Fast Extraction of Similar Partial Images for Querying Viewing Parameters

Ai Gomi* Takayuki Itoh* Koji Koyamada** Shohei Hido**

(*) Ochanomizu University (**) Kyoto University

E-mail: {gomiai, itot}@itolab.is.ocha.ac.jp

Abstract

This paper presents a technique for quickly extracting target parts of pre-stored high-resolution images, where the extracted parts are similar to an input small image. Our technique first quickly extracts thousands of candidates of target parts by rough similarity estimation, and then calculates similarity values between the candidates and input images. The technique then generates response surface interpolating the similarity values on an image space, and finally discovers optimal target parts by searching for the maximum similarity on the response surface. We have not guaranteed the 100% accuracy of the partial image retrieval yet, but the technique realizes nearly real-time retrieval from large-scale images contains millions of pixels. The technique can be used to various applications, but we especially expect to apply for querying viewing parameters.

1. Introduction

Since digital cameras, camera-embedded cellular phones, and the Internet have got drastically popular, it got easier to create or obtain digital images. There are currently huge numbers of large-scale images in our daily life, so recently requirements for efficiently retrieving those images have been increasing. Some popular image retrieving systems refer keywords associated to the pre-stored images [1,2]. However, the systems have some problems, because vocabulary associated from the images strongly depends on sense of person, and it is difficult to associate words to images adequately for every people.

One solution to solve such problem is retrieval of images not only by referring keywords, but also by analyzing visual property such as color distribution and shapes of contents. Many of such technique require input images, then compare the visual property between the input images and pre-stored query images, and finally return the most similar image as output. For example, QBIC[3] is one of the first commercial image query system, which accepts similar images and sketches as input information.

Capability of similarity-based image retrieval techniques depend on definitions of the similarity. Therefore it is desirable to investigate various definition of the similarity. Various visual criteria can be applied to calculate the similarity, such as color signature [4], color characteristics [5][6], and perspective [7]. Some mathematical schemes, such as Wavelet [8] or vector quantization [9], are also useful for image similarity estimation.

Since many of image retrieval techniques focus on similarity estimation between equally-sized images, partial image retrieval is also an interesting topic. Most of partial image retrieval techniques extract rectangular parts from pre-stored large-scale images, and compares the extracted parts with input images. Well-known template matching technique [10] compares RGB values pixel-by-pixel to examine the similarity, but it is very time-consuming for large-scale images. Some of fast template matching techniques, such as sequential similarity detection algorithm [11] and coarse-to-fine search [12], have been also well-known. Kimura et al. archived nearly real-time and accurate partial image retrieval [13], but their test cases are still much smaller than image sizes of recent consumer-purpose digital cameras.

Another issue of template-matching techniques is retrieval of "similar but different" objects. Specific object retrieval techniques [14,15,16] are also interesting topics in the view of this issue, but again, their test cases are not enough large as image sizes of recent digital cameras.

Against well-known template matching approach to find the target partial images, the technique aims the discovery of "similar but different" partial images in the high-resolution images. The technique first generates clusters of adjacent similarly-colored pixels, and then generates triangles connecting the centers of clusters of dominant three colors. Fast triangle similarity calculation successfully extracts small numbers of candidates of target parts. The technique then applies known image similarity calculation technique, and generates a response surface interpolating the similarity values. The surface is used to discover the optimal target part that brings the maximum similarity.

This technique has a potential to be applied to various applications. We are especially interested in query of viewing parameters of the small images, by searching for target parts from high-resolution images placed in virtual 3D spaces.

2. Overview

2.1 Algorithmic overview

Here we formalize the problem of extraction of similar partial images as follows:

1. Let the input small image as I, and the high-resolution image as I'.
2. Let an arbitrarily extracted rectangular part of I' as I", whose upper-left corner locates at *(x,y)*, width is *w*, and height is *h*.
3. Calculate the similarity between I and I", and discover I'' that brings the maximum similarity value.

Since known image similarity calculation techniques are somewhat costly, the technique may take huge calculation time if we apply every possible values of *x*, *y*, *w*, and *h*, to extract I". This paper discusses the technique for quick extraction of the optimal I", consists of the following two steps.

[Step 1] This step quickly extracts candidates of target parts by rough similarity estimation.

[Step 2] This step calculates similarity values between I and multiple the candidates of I", and generates a response surface interpolating the similarity values. The response surface brings the optimal I" that brings the maximum similarity value.

Here, features of our technique are as follows:

- Low computation time for large-scale images
- Discovery of "similar but different" parts, such as slightly larger, smaller, brighter, or darker parts.

We have not guaranteed the accuracy of our technique yet, but the next section appeals that partial image retrieval techniques are useful in some scenarios even though they are not 100% accurate.

2.2 Scenarios

Partial image retrieval techniques have potentials to use for variety of applications, but we especially expect to use for the following scenarios:

- If the technique is implemented to digital cameras, it can focus on specific objects or persons, even though the cameras are intermittently used and therefore well-known video-based retrieval technique cannot be used.
- Collecting images taken by various people at the same place and comparing with image database, users can specific which camera angles were preferable there. This application is especially interesting for some specific business, including civil developers and tour conductors.
- Visibility of civil objects, such as advertising displays or traffic signs, can be examined by inputting images of displays or signs. This application is interesting for traffic analysts or map developers.

We think that such scenarios queries viewing parameters from large-scale (ex. panoramic) images. Accelerated retrieval techniques are desirable for such purposes because the scenarios assume real-time usage or huge number of images. However, these scenarios do not assume fully-automated processes, so we think it is still useful even if the techniques do not guarantee 100% accuracy of retrieval results. That is why we aimed computation time rather than 100% accuracy.

3. Implementation

3.1 Color reduction

Various known image similarity calculation methods can be applied to partial image retrieval, but currently we apply a color-based similarity calculation algorithm. It first generates the clusters of similarly-colored pixels by K-means method which classifies them into K clusters. The algorithm we applied is as follows:

1. Spread all pixels in an image onto a RGB-space.
2. Reasonably choose the centers of K clusters.
3. Classify all pixels into the cluster whose center is the closest in the RGB space.
4. Recalculate the positions of centers of generated clusters.
5. Terminate if the centers of new clusters are enough close to the previous positions, otherwise return to 3.

Figure 1. (Upper) input images. (Lower) posterized images by k-means method. (Left) a scene of shining day. (Right) a scene of rainy day.

Currently we assume that K, number of clusters, are pre-defined or given by a user. After terminating the above algorithm, our implementation assigns same color IDs to the pixels belonged to each cluster, and then decomposes the clusters according to adjacency of pixels and assigns labels to the decomposed clusters, to obtain same-colored regions of the image.

The process can bring same clustering results from images which have slightly different tones, such as same scenes where weather or sunshine direction varies. Figure 1 shows the example of the veriety of tones, where upper images are input images and lower images show the clustering results by posterinzing the images as 15 colors. The upper-right image is taken by a shiny day and the upper-left is rainy. Comparing lower-left and lower-right images, it can be observed that same objects in both images are categorized into the same clusters.

As described in Section 2.1, our technique first roughly extracts candidate target parts from the large-scale images, and then calculates the similarity values between the input images and the candidate parts. Posterized images are used for the both processes. In the second process, our technique calculates distances between same-colored regions of I and I'', and obtains average distance values as errors between I and I''. We use the inverse number of the average distances as similarity values. The algorithm is similar to the implementation introduced in [8].

3.2 Rough similarity calculation

Step 1 of the technique applies rough similarity calculation scheme. It first applies clustering and labeling described in 2.2 to I and I', and generate a triangle T, connecting centers of the largest three regions, R_1, R_2, and R_3, as shown in Figure 1(left). It then extracts regions from I', which have same colors with R_1, R_2, and R_3. It then generates all possible triangles T'_i, connecting three regions in I', where those combination of three colors are same as T, as shown in Figure 1(right). It calculates similarity between T and each of T'_i, by simply calculating angles between each edge of T and each of T'_i. Though the number of T'_i may be very large, its computation time is generally small. If T'_i is enough similar to T, the technique extracts a rectangular region around T'_i as I'', a candidate of the target part.

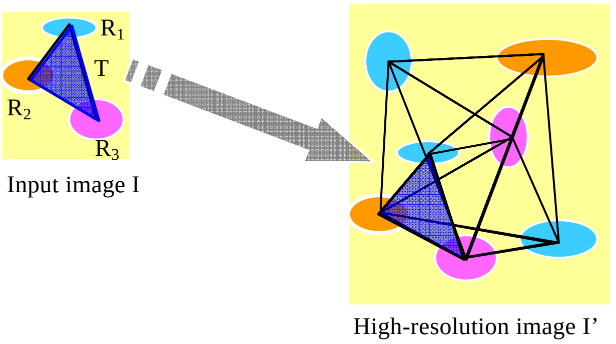

Figure 2. Triangle T in the input image I, and triangle T'i in high-resolution image I'.

- Algorithm for comparing triangles

Each center of gravity of the triangle T in the input image I is defined as $R_1(x_{R1},y_{R1})$, $R_2(x_{R1},y_{R2})$, and $R_3(x_{R3},y_{R3})$, in the image coordinate of I. Similarly, each center of gravity of the triangle T' in the retrieved image I' is defined as $R_{1'i}(x_{R1'i},y_{R1'i})$, $R_{2'i}(x_{R2'i},y_{R2'i})$, and $R_{3'i}(x_{R3'i},y_{R3'i})$, in the image coordinate of I'. In this case, R_1 and $R_{1'i}$, are the centers of the same-colored clusters. Similarly, R_2 and $R_{2'I}$, and R_3 and $R_{3'I}$, are the centers of the same-colored clusters. At first, our implementation compares each of 'x and y'-coordinate value, as shown in Figure 3.

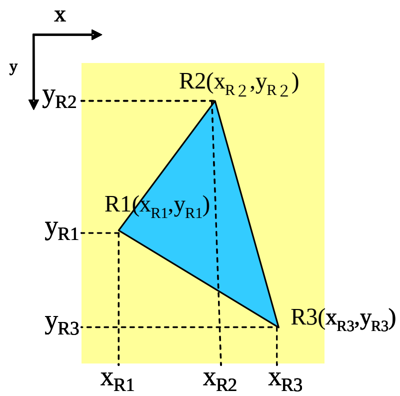

Figure 3. The position of the centersR.

If each coordinate x and y of R_1, R_2, and R_3 in I are ordered as the inequation (1),

$$x_{R1} \le x_{R2} \le x_{R3}\ ,\ y_{R1} \le y_{R2} \le y_{R3} \qquad (1)$$

the triangles T'i which are satisfied the inequation (2):

$$x_{R1'i} \le x_{R2'i} \le x_{R3'i}\ ,\ y_{R1'i} \le y_{R2'i} \le y_{R3'i} \qquad (2)$$

And then similar triangles T'I, satisfying the inequation (3), are extracted from them. Here value S is a threshold specified by a user.

$$\left| \frac{x_{R2} - x_{R1}}{x_{R1} - x_{R3}} - \frac{x_{R2'_i} - x_{R1'_i}}{x_{R1'_i} - x_{R3'_i}} \right| < S \ ,$$

$$\left| \frac{y_{R1} - y_{R2}}{y_{R3} - y_{R2}} - \frac{y_{R1'_i} - y_{R2'_i}}{y_{R3'_i} - y_{R2'_i}} \right| < S \qquad (3)$$

Since triangle comparison is dealt with only by the combination of four simple inequations, our implementation can quickly find similar triangles.

3.3 Optimal part discovery using response surfaces

Step 2 calculates image similarity between I and I” using the known technique described in 2.2. Since the candidates are just roughly extracted, it often happens that the optimal I” is located among the candidates. The known image similarity calculation is somewhat costly, and therefore it is not preferable to calculate the similarity between I and every possible I” around the candidates. Here we discuss how to obtain the optimal I” by the small number of image similarity calculation and its interpolation.

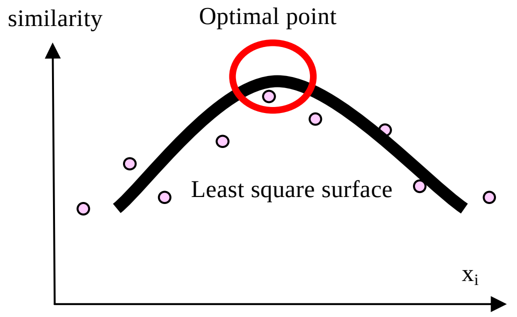

Figure 4. Response surface method.

Response surface method is a well-known technique to discover the optimal solution from small number of sampling values. Usually the method approximates the function of parameters and the response value as two-dimensional polynomial equation described as equation (4).

$$y = \beta_0 + \sum_{i=0}^{n} \beta_i x_i + \sum_{i=0}^{n} \beta_{ii} x_i^2 + \sum_{i<j}^{n} \beta_{ij} x_i x_j \quad (4)$$

Calculating the similarity between I and I”, the technique substitutes the parameters and the similarity values to (4). Here y is the similarity value, and x_0 to x_3 are parameters (x,y,w,h) of I”. The values of $\beta, \beta_i, \beta_{ii}, \beta_{ij}$ can be calculated because (1) becomes simultaneous equations of these variables. The response surface generally has a peak of the response value, as shown in Figure 4, which is to be easily discovered. The technique returns the output image with parameters (x,y,w,h) that brings the maximum similarity value.

4. Experiments

Figure 5 shows an example of the input(131×189pixels), retrieval(326×375pixels), and output (113×164pixels) images. Step 1 generated 54621 triangles in the retrieval image; however, computation time for similarity calculation between the triangles was only 2.7 seconds. The step selected only 11 triangles as candidate positions. Figure 6 shows the images extracted at the candidates.

Figure 5. (Upper-left) Input image I. (Lower-left) Output image I”. (Right) Retrieval image I’.

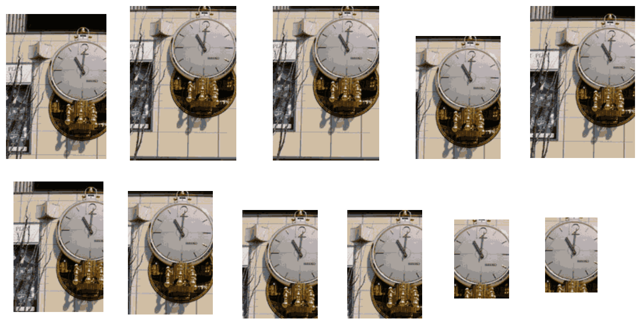

Figure 6. Similar images at the candidate positons.

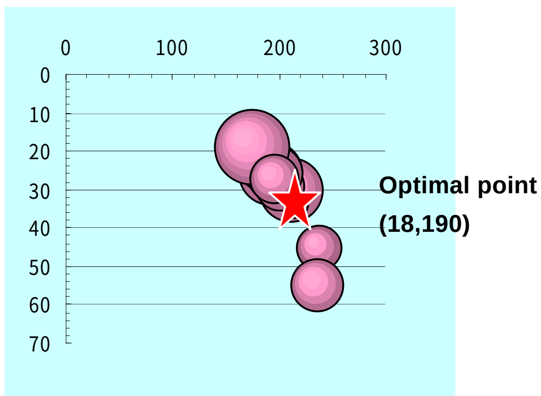

Figure 7. Discovery of optimal point using response surface.

Step 2 calculates the similarity of images between I and 11 images shown in Figure 6, and then applies the response surface method to obtain the output image. Figure 7 shows the similarity values of 11 images, and the optimal point obtained from the response surface. Here sizes of spheres denote the similarity values, and axes denote x- and y-coordinates of I”. Figure 5 (Lower-left) shows the output image at the optimal point shown in Figure 7. The result shows that the technique successfully discovered an enough similar partial image by small number of similarity calculation and optimal point finding, in a small computation time.

Figure 8 shows the input (67x272pixels), retrieval (1037x691pixels), and output (71x291pixels) images, which attempted the retrieval of a specific person from a large-scale image. Step 1 generated 324 triangles and computation time for similarity was only 0.25 seconds. 23 images at the selected triangles advanced for step 2.

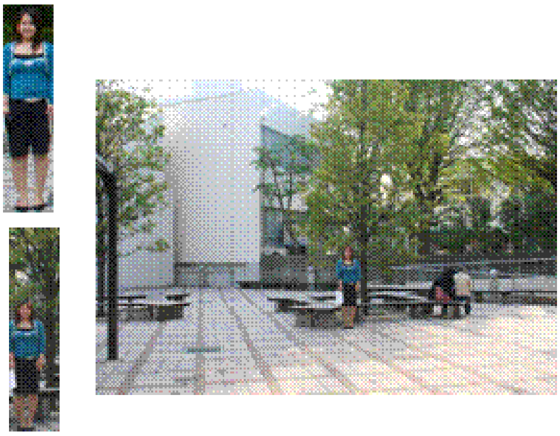

Figure 8. (Upper-left) Input image I. (Lower-left) Output image I”. (Right) High-resolution image I’.

Figure 9. Similar images at the candidate positons.

In this case, colors of cloth worn by person must be different from background colors, but it may conditionally fail. To solve this problem, we would like to extract not only by colors of object but also by the shape, as a future work. As shown in Figure 10, candidate positions in this test divide into three areas. We should independently apply the response surfaces for the three areas, but current our implementation requires manual work for the division. Its automation is also our future work.

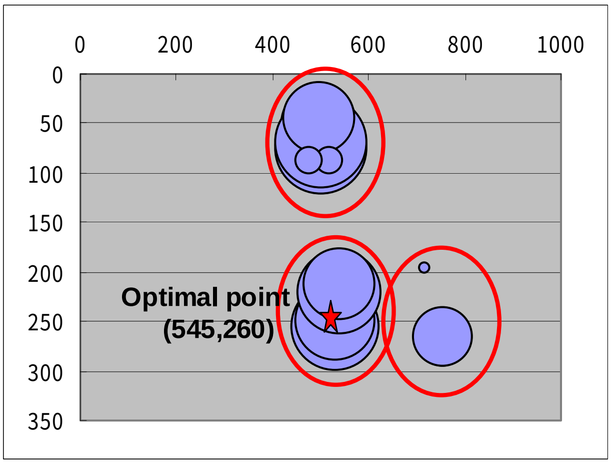

Figure 10. Discovery of optimal point using response surface.

Figure 11. (Upper-left) Input image I. (Lower-left) Output image I”. (Right) High-resolution image I’.

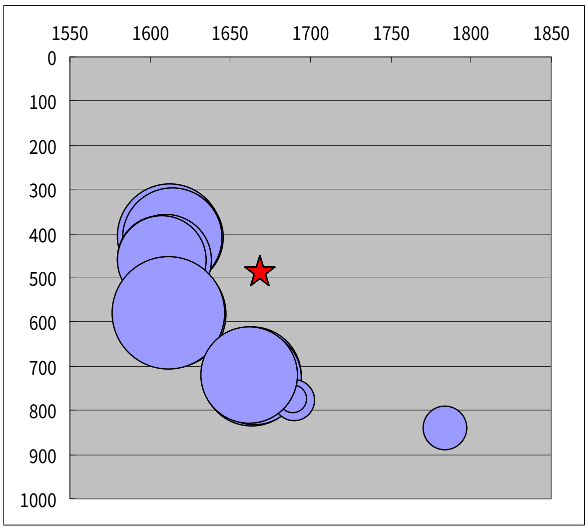

Figure 12. Discovery of optimal point using response surface.

Third experiment is shown in Figure 11, where input (183x380pixels), hight-resolution (3456x2304pixels), and output (349x919pixels) images. Step 1 generated 72930 triangles in the high-resolution image, but the computation time for similarity calculation was only 0.79 seconds. The step selected only 16 triangles as candidate positions. Figure 12 shows the similarity values of the candidates and the optimal point obtained from the response surface.

5. Conclusion

This paper presented a technique for quick extraction of similar partial images from high-resolution images.

Our experiment proofed that the technique is very quick, but not sometimes accurate as s shown in Figure 7. One challenge is improvement of accuracy while the technique preserves the computational efficiency. Reasonable combination with other techniques, such as shape-based similarity calculation, frequency analysis, and specific object extraction, is an interesting future work for us.

As mentioned in the Section 1, we are interested in using this technique for query of viewing parameters of images. If the high-resolution image is located in a virtual 3D space (i.e. 360-degree cylinder image for QuickTime VR [17]) and there are actually similar parts to the input image, the technique can suggest the viewing parameters of the input image. Also, if we have high-resolution images taken at near points, we can obtain intermediate images by applying View Morphing [18]. Such application should be challenging but interesting future work for us.

References

[1] http://bsearch.goo.ne.jp/

[2] http://images.google.com/

[3] C. Faloutsos, R. Barber, M. Flickner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and Effective Querying by Image Content. Journal of Intelligent Information Systems, Vol. 3, No. 3, pp. 231-262, 1994.

[4] Y. Rubner., Perceptual Metrics for Image Database Navigation, PhD thesis, Stanford University, May 1999.

[5]http://chihara.aist-nara.ac.jp/people/95/masano-h/research-j.html (in Japanese)

[6]http://criepi.denken.or.jp/jp/pub/annual/1998/98seika30.pdf

[7]http://www.simplex.t.u-tokyo.ac.jp/theses/2001m-yatagawa_eiji.pdf (in Japanese)

[8]http://www.taf.or.jp/publication/kjosei_18/pdf/071.pdf (in Japanese).

[9] K. Inoue, K. Urahama, Similarity Search of Videos by Using Reference Images, Journal of the Institute of Electronics, Information, and Communication, Vol. J84-D-II, No. 7, pp. 1533-1536, 2001. (in Japanese)

[10]http://www.netnam.vn/unescocourse/computervision/861.htm

[11] E. I. Barnea and H.F.Silverman, A class of Algorithms for Fast Digital Image Registration, IEEE Transactions on Computers, Vol. C-21, pp. 179-186, Feb 1972.

[12]http://www.cs.ou.edu/~atiq/papers/mhtc-ijamt.pdf

[13] A. Kimura, T. Kawanishi, K. Kashino, Similarity-based Partial Image Retrieval Guaranteeing Same Accuracy as Exhaustive Matching, IEEE 2004 International Conference on Multimedia & Expo. (ICME2004), 2004.

[14] T. Hattori, H. Kitajima, T. Yamasaki, Face Pattern Recognition And Extraction From Multi Persons Scene, Proceedings of ICEIS 2003 (Fifth International Conference on Enterprise Information Systems), ACM, AAAI and IEEE, pp.92-99, 2003.

[15] R. Fergus, P. Perona, A. Zisserman, Object Class Recognition by Unsupervised Scale-Invariant Learning, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 264-271, 2003.

[16] S. Helmer, D. G. Lowe, Object Recognition with Many Local Features, Generative Model Based Vision 2004 (GMBV 2004).

[17] S. E. Chen, QuickTime VR: An Image-based Approach to Virtual Environment Navigation, ACM SIGGRAPH ’95, pp. 29-38, 1995.

[18] S. M. Seitz, C. R. Dyer, View Morphing, ACM SIGGRAPH ’96, pp. 21-30, 1996.