

SIMILARITY-BASED PARTIAL IMAGE RETRIEVAL GUARANTEEING SAME ACCURACY AS EXHAUSTIVE MATCHING

Akisato Kimura, Takahito Kawanishi and Kunio Kashino

NTT Communication Science Laboratories, NTT Corporation
3-1, Morinosato-Wakamiya, Atsugi-shi, Kanagawa, 243-0198, Japan
E-Mail : {akisato, kawanisi, kunio}@eye.brl.ntt.co.jp

ABSTRACT

We propose a new framework for quick and accurate partial image retrieval from a huge number of images based on a predefined distance measure. Finding partial similarities generally requires a huge amount of storage space for indexes due to the large number of portions of images. The proposed method extracts portions from each database image at a constant spacing, while it extracts all possible portions from a query image. In this way, the proposed method can greatly reduce the size of indexes while theoretically guaranteeing the same accuracy as exhaustive matching.

1. INTRODUCTION

Advances in image-capturing devices, storage devices, and infrastructures for broadband networks have made a huge number of digital images easily accessible. Hence, techniques for quick image retrieval from large image databases are strongly needed. A common approach is based on keywords. However, providing a keyword for each image is a difficult task for computers and time-consuming for humans.

Therefore, content-based image retrieval (CBIR) that finds similar images based on quantitative features such as colors, shapes, and textures, has been investigated. Examples of CBIR algorithms are QBIC [1] and ExSight [2], which search images including specific objects, and VisualSEEK [3], which utilizes rough shapes and positions as queries. In recent years, region-based approaches [4, 5], which retrieve partial images similar to a query, have been widely investigated. Partial image retrieval broadens the application areas in comparison with whole image retrieval. One of the crucial points in this approach is how to extract portions of images. Some recent approaches [6, 7, 8] have attempted to decompose an image into individual objects. However, they are not always accurate in identifying objects, and therefore most of them use domain-specific constraints to improve the accuracy. Here, we consider a domain-independent approach which involves extracting all possible partial images

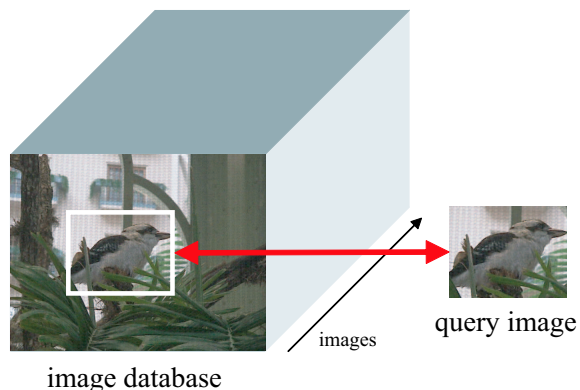


Fig. 1. Outline of partial image retrieval

and then calculating similarities. A major problem in this approach has required a huge amount of calculation and storage space for indexes.

Here, as one of the most important elements of region-based CBIR, we propose a new framework to retrieve partial images similar to a given image (a *query*) from a huge number of images (a *database*) based on a predefined distance measure $d(\cdot, \cdot)$ (Fig. 1). Specifically, our task is to find similar partial images such that distance values between a query fall below a predefined value (a *search threshold*) θ , that is,

$$d(D_i(x, y), Q) \leq \theta, \quad (1)$$

where $D_i(x, y)$ is a portion of an image D_i in the database $\mathbf{D} = \{D_i\}_{i=1}^M$ at position (x, y) and Q is a query image. The framework of the proposed method is similar to DualMatch [9] and GeneralMatch [10], which are methods for quick matching of time-series signals; portions at a constant spacing are firstly extracted from each database image, while all possible portions are extracted from a query image. In this way, the proposed method theoretically guarantees the same accuracy as exhaustive matching.

This paper is organized as follows: Section 2 outlines the proposed method. Section 3 details the procedure for retrieving partial images. Section 4 presents the experimen-

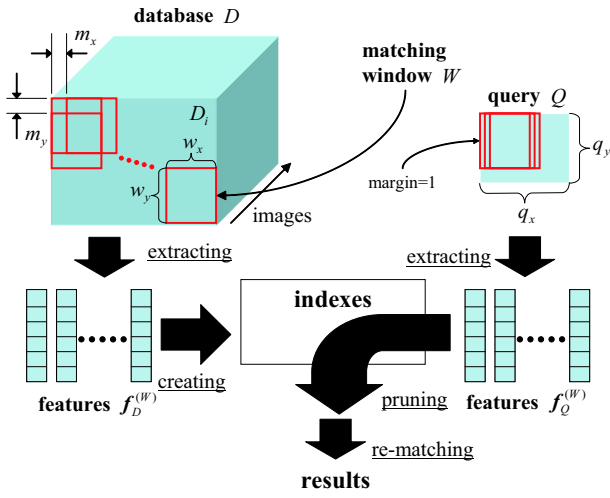


Fig. 2. Outline of the proposed method

tal results. Finally, section 5 gives conclusions.

2. OUTLINE

Fig. 2 shows the outline of the proposed method.

Firstly, a block B with a predefined size $b_x \times b_y$ pixels is picked up from each image D_i of a database D , and a feature $\mathbf{f}_{D_i}^{(B)}(x, y)$ is extracted. Blocks are extracted while sliding 1 pixel each. Various sorts of features are usable, such as, average colors of blocks, DFT (Discrete Fourier Transform) coefficients, DCT (Discrete Cosine Transform) coefficients, color histograms, or wavelet coefficients. Secondly, a *matching window* W with a predefined size $w_x \times w_y$ pixels is placed onto each image D_i of a database D . A portion $D_i^{(W)}(x_{Di}, y_{Di})$ is extracted from the matching window W . We call this portion the *matching region*, where (x_{Di}, y_{Di}) represents the location of the matching region in image D_i . Matching regions are extracted at a predefined spacing of (m_x, m_y) pixels. Therefore, not all possible matching regions are extracted. We call (m_x, m_y) a *margin*. Next, a feature $\mathbf{f}_{D_i}^{(W)}(x_{Di}, y_{Di})$ is extracted from each matching region, which is a set of features extracted from disjoint blocks in the matching region $D_i^{(W)}(x_{Di}, y_{Di})$. For simplicity, let us suppose that the size of a matching window (w_x, w_y) is an integer multiple of the size of a block (b_x, b_y) . Extracted features are stored in an index structure. Many kinds of indexing algorithms are applicable, including those based on spatial index structures, such as R*-tree [11] and SR-Tree [12], and also those based on vector quantization [13]. The procedure described above does not need a query image, and can therefore be done beforehand.

After a query image Q is given, a matching window W of the same size as that of the database is placed onto

the query image and features $\mathbf{f}_Q^{(W)}(x_Q, y_Q)$ are extracted. Unlike the database case, all possible matching regions are extracted from the query. This means that margins for the query equal 1. Secondly, matching regions that have the possibility to be included in partial images similar to the query are extracted from the database by using the index mentioned earlier. Details will be described in Section 3. Lastly, partial images $D_i(x_i, y_i)$ corresponding to the selected matching regions are matched with the query Q as follows:

$$\begin{aligned}
 d(D(x, y), Q) & \\
 &\stackrel{\text{def}}{=} d(\mathbf{f}_D(x, y), \mathbf{f}_Q) \\
 &\stackrel{\text{def}}{=} \sum_{i=0}^{\frac{q_x}{b_x}-1} \sum_{j=0}^{\frac{q_y}{b_y}-1} d(\mathbf{f}_D^{(B)}(x + ib_x, y + jb_y), \mathbf{f}_Q^{(B)}(ib_x, jb_y)).
 \end{aligned} \tag{2}$$

Equation (2) means that the distance between a partial image $D_i(x_i, y_i)$ and a query Q is defined as the sum of the distances between corresponding block features. Various kinds of distance measures are available, such as, Euclid distance, Manhattan distance, inner product, or normalized cross correlation.

To avoid false dismissals, the margin (m_x, m_y) , size of a matching window (w_x, w_y) , and size of a query image (q_x, q_y) must satisfy the following conditions:

$$q_x \geq w_x + m_x, \quad q_y \geq w_y + m_y, \tag{3}$$

$$m_x \leq w_x, \quad m_y \leq w_y. \tag{4}$$

Condition (3) means that a window with the same size as the query must include at least one matching region. Condition (4) means that each pixel must be included in at least one matching region. We need to set the margin (m_x, m_y) and the size of a matching window (w_x, w_y) so as to satisfy conditions (3) and (4) to preclude false dismissals.

3. PRUNING MATCHING REGIONS

This section describes how relevant matching regions are selected by utilizing the index created from the database.

Firstly, a feature $\mathbf{f}_Q^{(W)}(x_Q, y_Q)$ of each matching region $Q^{(W)}(x_Q, y_Q)$ is extracted. Next, all relevant features $\mathbf{f}_D^{(W)}(x_D, y_D)$ of the database such that the distance between the feature $\mathbf{f}_Q^{(W)}(x_Q, y_Q)$ falls below a predefined value (a *pruning threshold*) $\hat{\theta}$ are selected.

$$d(\mathbf{f}_D^{(W)}(x_D, y_D), \mathbf{f}_Q^{(W)}(x_Q, y_Q)) \leq \hat{\theta}. \tag{5}$$

The distance between features is calculated in the same way as Eq. (2).

$$d(\mathbf{f}_D^{(W)}(x_D, y_D), \mathbf{f}_Q^{(W)}(x_Q, y_Q))$$

$$\stackrel{\text{def}}{=} \sum_{i=0}^{\frac{w_x}{b_x}-1} \sum_{j=0}^{\frac{w_y}{b_y}-1} d\left(\mathbf{f}_D^{(B)}(x_D + ib_x, y_D + jb_y), \mathbf{f}_Q^{(B)}(x_Q + ib_x, y_Q + jb_y)\right) \quad (6)$$

In a basic implementation, Eq. (5) is calculated for all matching regions $Q^{(W)}(x_Q, y_Q)$. We can utilize MBSs (Minimum Bounding Spheres) instead of all matching regions for efficient selection of relevant matching regions. Firstly, a representative feature $\bar{\mathbf{f}}_Q$ of the query is calculated, and then the maximum distance between the features of the query and the representative feature,

$$\bar{d} = \max_{(x,y)} d\left(\bar{\mathbf{f}}_Q, \mathbf{f}_Q^{(W)}(x,y)\right), \quad (7)$$

is calculated. A MBS is composed of a center $\bar{\mathbf{f}}_Q$ and a radius \bar{d} . We can use, for example, a feature at a specific position or a centroid of all features, as a representative feature. Then, all relevant features are selected by determining whether the following inequality is satisfied:

$$d\left(\mathbf{f}_D^{(W)}(x_D, y_D), \bar{\mathbf{f}}_Q\right) \leq \hat{\theta} + \bar{d}. \quad (8)$$

It is important to determine an appropriate pruning threshold so as not to cause false dismissals. We set the pruning threshold as follows:

$$\hat{\theta} = \frac{\theta}{N}, \quad (9)$$

where N is the minimum number of disjoint matching regions in the query, i.e., the minimum number of matching regions in the query such that regions do not overlap with each other. For simplicity, suppose that the size of a matching window (w_x, w_y) is an integer multiple of the margin (m_x, m_y) . Then, N is calculated from the size of the query image (q_x, q_y) and the size of the matching window (w_x, w_y) as follows:

$$N = N_x \cdot N_y, \quad (10)$$

$$N_x = \max\left(1, \left\lfloor \frac{q_x + 1}{w_x} \right\rfloor - 1\right), \quad (11)$$

$$N_y = \max\left(1, \left\lfloor \frac{q_y + 1}{w_y} \right\rfloor - 1\right). \quad (12)$$

4. EXPERIMENTS

4.1. Conditions

We tested the proposed method in terms of index size and search speed. The tests were carried out on a PC (Intel Xeon, 2.8 GHz). In the experiments, we used a real-life image data set [14, 15] from Pennsylvania State University. The data set contains 1000 images stored in JPEG

format, with a size of 384×256 pixels. We chose 10 portions with a size of $q_x \times q_y = 80 \times 80$ pixels from the images as queries. Matching regions were those with a size of $w_x \times w_y = 64 \times 64$ pixels and were placed at a spacing of $m_x \times m_y = 16 \times 16$ pixels in each database image. We used average RGB values of $b_x \times b_y = 8 \times 8$ pixel blocks as features. Thus, the dimensionality of features was $192 (= 8 \times 8 \times 3)$. We applied Global Pruning [13] based on vector quantization as an indexing and pruning algorithm. We used all matching regions of the query, not MBSs, for pruning. The distance measure was Euclid distance. The search threshold was set to 1000.

4.2. Index size

We examined the relationship between the margin and index size. Fig. 3 shows the results. The index size decreased exponentially as the margins increased.

4.3. Search time

Next, we examined the relationship between the margins and search time. Two measures were used: time for matching calculations and time for pre-processing. The number of classifications (*clusters*), which is a parameter for vector quantization in Global Pruning, was 1024. Fig. 4 shows the results. We can see that the margin of 8 gave the best performance. It should be noted that the search time of exhaustive matching was 117.54 seconds and the proposed method is significantly faster.

Lastly, we examined the relationship between the number of clusters and search time. The margins were set to 16 ($m_x = m_y = 16$). Fig. 5 shows that the search time decreased as the number of clusters increased.

5. CONCLUSIONS

We have proposed a fundamental framework for quick partial image retrieval, which is a task of finding images partially similar to a query in a large number of database images. The proposed method does not have to index all possible matching regions of the database images. This greatly reduces the size of indexes and the number of file accesses in searching while guaranteeing the same accuracy as exhaustive matching. Experiments showed significant reduction of the index size and search speed. The proposed method is expected to be able to deal with changes of brightness, scale, and rotation by utilizing other features [16, 17], distance measures [17], and algorithms for matching images [17, 18]. Future work includes detailed investigations of index structures, features, and distance measures appropriate for partial image retrieval.

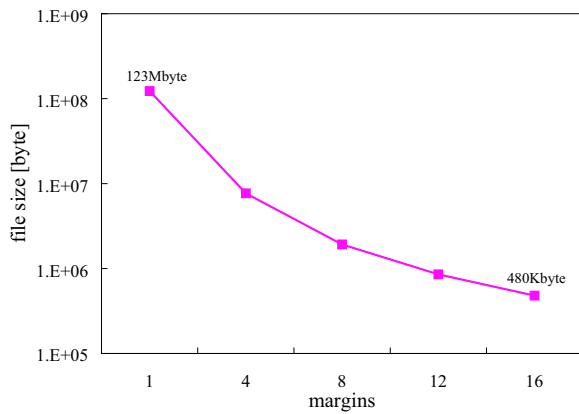


Fig. 3. Relationship between the margin and the size of indexes.

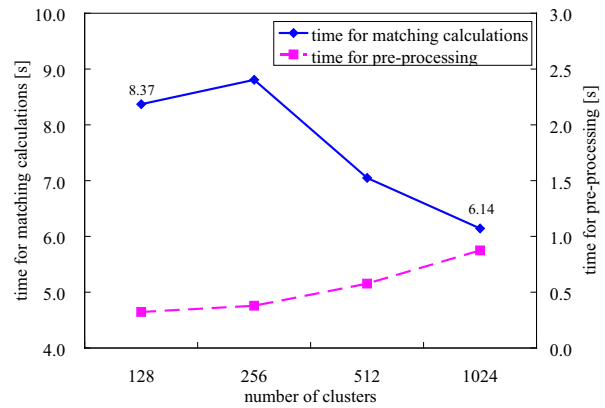


Fig. 5. Relationship between the number of clusters and search time.

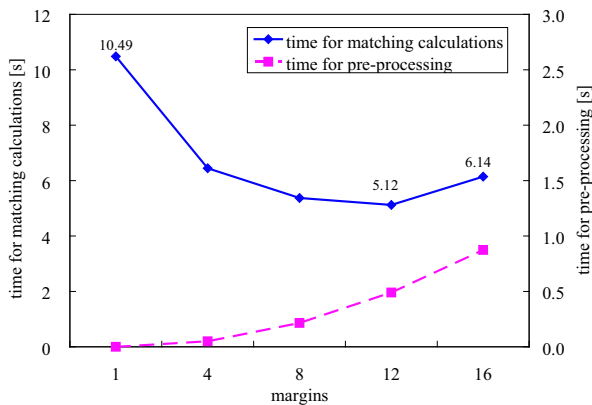


Fig. 4. Relationship between the margin and search time.

6. REFERENCES

- [1] M. Flicker et al.: "Query by image and video content : The QBIC system", *IEEE Computer*, Vol.28, No.9, pp.23-32, 1995.
- [2] K. Kushima et al.: "ExSight : Highly accurate object based image retrieval system enhanced by redundant object extraction", *Proc. of WAIM*, pp.331-343, 2000.
- [3] J. R. Smith et al.: "VisualSEEK : a fully automated content-based image query system", *Proc. of ACM Multimedia*, pp.87-93, 1996.
- [4] A. Natsev et al.: "WALRUS : A similarity retrieval algorithm for image databases", *Proc. of ACM SIGMOD*, pp.395-406, 1999.
- [5] C. Kim et al.: "Partial similarity searching in large image databases", *KAIST CS Technical Report*, CS-TR-2001-165, 2001.
- [6] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein and J. Malik: "Blobworld : A system for region-based image indexing and retrieval", *Proc. of International Conference on Visual Information Systems*, pp.509-516, 1999.
- [7] W. Y. Ma and B. S. Manjunath: "NETRA : A toolbox for navigating large image databases", *Proc. of ICIP*, Vol.1, pp.568-571, 1997.
- [8] M. Das, E. M. Riseman and E. M. Draper: "FOCUS : Searching for multi-colored objects in a diverse image database", *Proc. of CVPR*, pp.756-761, 1997.
- [9] Y. Moon et al.: "Duality-based subsequence matching in time-series databases", *Proc. of ICDE*, pp.263-272, 2001.
- [10] Y. Moon et al.: "GeneralMatch : A subsequence matching method in time-series databases based on generalized windows", *Proc. of ACM SIGMOD*, pp.382-393, 2002.
- [11] N. Beckman et al.: "The R*-tree : an efficient and robust access method for points and rectangles", *Proc. of ACM SIGMOD*, pp.322-331, 1990.
- [12] N. Katayama et al.: "The SR-tree : an index structure for high-dimensional nearest neighbor queries", *Proc. of ACM SIGMOD*, pp.369-380, 1997.
- [13] A. Kimura et al.: "Very quick audio searching : introducing global pruning to the Time-series Active Search", *Proc. of ICASSP*, Vol.3, pp.1429-1432, 2001.
- [14] "Test image database used in SIMPLicity paper", <http://wang.ist.psu.edu/IMAGE/>.
- [15] J. Z. Whang et al.: "SIMPLicity : Semantics-sensitive integrated matching for picture libraries", *IEEE Trans. PAMI*, Vol.23, No.9, pp.947-963, 2001.
- [16] M. Uenohara et al.: "Use of Fourier and Karhunen-Loeve decomposition for fast pattern matching with a large set of templates", *IEEE Trans. PAMI*, Vol.19, No.8, pp.891-898, 1997.
- [17] V. V. Vinod et al.: "Focused color intersection with efficient searching for object extraction", *Pattern Recognition*, Vol.30, No.10, pp.1787-1797, 1997.
- [18] T. Kawanishi et al.: "Skipping template matching guaranteeing same accuracy as exhaustive search", *Proc. of ICAPR*, pp.209-212, 2003.