Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Information Processing and Management 42 (2006) 484–502
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

XMage: An image retrieval method based on partial similarity

Chang-Ryong Kim *, Chin-Wan Chung

Division of Computer Science, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, 373-1, Kusong-dong, Yusong-gu, Taejon 305-701, South Korea

Received 4 June 2004; accepted 5 January 2005
Available online 17 March 2005

Abstract

XMage is introduced in this paper as a method for partial similarity searching in image databases. Region-based image retrieval is a method of retrieving partially similar images. It has been proposed as a way to accurately process queries in an image database. In region-based image retrieval, region matching is indispensable for computing the partial similarity between two images because the query processing is based upon regions instead of the entire image. A naive method of region matching is a sequential comparison between regions, which causes severe overhead and deteriorates the performance of query processing. In this paper, a new image contents representation, called Condensed eXtended Histogram (CXHistogram), is presented in conjunction with a well-defined distance function CXSim() on the CX-Histogram. The CXSim() is a new image-to-image similarity measure to compute the partial similarity between two images. It achieves the effect of comparing regions of two images by simply comparing the two images. The CXSim() reduces query space by pruning irrelevant images, and it is used as a filtering function before sequential scanning. Extensive experiments were performed on real image data to evaluate XMage. It provides a significant pruning of irrelevant images with no false dismissals. As a consequence, it achieves up to 5.9-fold speed-up in search over the R^*-tree search followed by sequential scanning.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Image-to-image similarity measure; Region-based retrieval; Histogram intersection; Filtering function

1. Introduction

Content-based image retrieval systems search for images that are similar to a query image from a given image database. An efficient image retrieval technology is necessary in diverse applications, such as medical

* Corresponding author. Tel.: +82 32 610 3481; fax: +82 32 610 3488.
E-mail address: ckim@bc.ac.kr (C.-R. Kim).

0306-4573/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2005.01.007

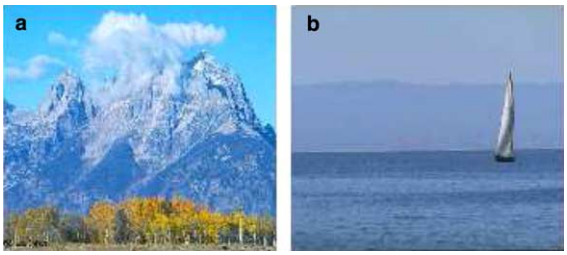

Fig. 1. Two similar images compared by the feature vector of the entire image.

analysis, weather prediction, image searches on the internet, and electronic commerce. Since the use of image data has been widely spread across many application domains, an efficient search on voluminous and complex information, which is the intrinsic characteristic of multimedia data, is becoming increasingly important. Initial systems tackled the content-based image retrieval problem by using color, texture, and shape features. As a result, two images can be seen as similar only because they have similar color distributions even though they are different visually as shown in Fig. 1. Most systems in this class are insensitive to partial similarity because they use features of the entire image.

The results of recent research show that the usage of the region is better than that of the entire image for the granularity of comparison, producing more accurate results (Jing, Mingjing, Zhang, & Zhang, 2002; Jing, Zhang, Lin, Ma, & Zhang, 2002; Natsev, Rastogi, & Shim, 2003; Smith & Li, 1999; Wang, Li, & Wiederhold, 2001). Often it is difficult for a user to select representative regions for the entire image and one or more segmented regions are usually not sufficient to address semantic object representation. Moreover, region matching is indispensable for computing partial similarity between two images in region-based image retrieval. It causes severe overhead and deteriorates the performance of query processing. In addressing these problems mentioned above, the image-to-image similarity measure for partial similarity is of special importance in retrieving similar images.

Coupled with the inaccuracy of automatic segmentation, the retrieved results, in general, do not always match user's intuition, or understanding of the images. Typically, a user's semantic understanding of an image is at a higher level than region representation and an object is composed of multiple segments with varying color and texture patterns. The accuracy of automatic segmentation is an open problem in vision application. We have focused on development a filtering function for region matching using color distribution on the assumption that an image is accurately segmented into a set of regions.

Mathematically, defining a partial similarity measure is equivalent to defining the distance between sets of points in a high-dimensional space, i.e., the feature space. Every point in the space corresponds to the feature vector or the descriptor of a region. Although the distance between two points in a feature space can be easily defined by various measures, such as the *Euclidean* distance, it is not obvious how to define a distance between two sets of feature points for searching two partially similar sets. Since two images may be segmented into different numbers of regions, they cannot be represented in one high-dimensional space. Therefore, the traditional dimension reduction method, such as DFT and Wavelet transformation, cannot be used.

It is straightforward to define distance mathematically between two sets of points in various ways. One common method is to compute the minimum distance between any two points in the two sets and consider that as the distance between the two sets. But the minimum distance between any two points cannot detect partial similarity between two images because they are considered to be partially similar if they are similar in a large portion and different only in certain regions.

This is the motivation for our research. The development of a distance function between two sets of feature points is addressed. The proposed distance function is image-to-image partial similarity measure and is

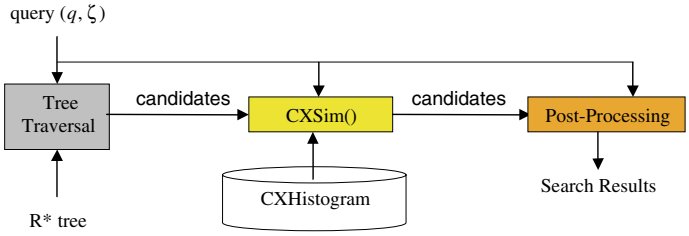

Fig. 2. XMage: block diagram.

used as a filtering function for a fast partial similarity search of image database. Fig. 2 shows the proposed query processing, a brief sketch of which is as follows. It is further explained in Section 4.

- The R^*-tree and the CXHistogram Construction: An appropriate segmentation algorithm generates a set of regions from a raw target image. The R^*-tree and the CXHistogram are constructed by using the extracted multi-attribute feature vector of each region. In our experiments, a 16 color value (RGB) is extracted as a feature vector. The XHistogram is a bar graph of the region intensities. It is proposed for solving the 'boundary value problem' (refer to Section 3.1). The CXHistogram is built by simply clustering the XHistogram using an equalized quantization method. The R^*-tree and the CXHistogram are stored into a database for later processing. The CXHistogram is a representation for each image.
- Query processing: A query image is also segmented into a set of regions and the CXHistogtram of the query image is generated by using the same method mentioned above. After the R^*-tree traversal, all target images contained at least one region similar to a region of query image are retrieved. Then, the proposed similarity measure, CXSim(), compares the partial similarity between the query image and the retrieved target images by comparing the two CXHistograms. The use of the CXSim() minimizes 'false hits' while guaranteeing 'no false dismissals.' The measure selects candidate target images from a database, each of which falls under a given threshold.

The contributions and advantages of this work are summarized as follows:

- The CXSim() is developed as an image-to-image similarity measure between two images for a partial similarity image search. It achieves the effect of comparing regions of two images by simply comparing the two images.
- The CXSim() is a location-independent and noise-tolerant similarity measure. The former means that two images are examined regardless of the location of objects, and the latter means two images are determined to be similar if they are similar except for some regions.
- The CXSim() does not depend on the number of regions of an image because the proposed method can detect the partial similarity of two images. This *is* achieved through *Histogram Intersection*, which drops dissimilar portions of two images.
- XMage achieves up to 5.9-fold speed up in search over the R^*-tree search followed by sequential scanning.

The rest of this paper is organized as follows: Section 2 provides a survey of related works with a discussion of the partial similarity search for image data. In Section 3, we propose a new image representation and an image-to-image partial similarity measure, the CXHistogram and the CXSim(), respectively, for region-based image retrieval including formal definitions and mathematical proofs. Section 4 explains in

detail the CXHistogram construction and a multi-step query processing with a partial similarity search algorithm. Extensive experiments are conducted in Section 5 in order to measure the effectiveness and efficiency of our proposed method, XMage. The conclusion is given in Section 6, along with the planned future works.

2. Related work

The annotation of images is given by domain experts and needs an enormous amount of time for large image datasets. Moreover, it may be different from the view of an end user because a domain expert can take a subjective view of an image. Therefore, there has been active research on automatic parsing in which the extracted features of each image map to a multi-dimensional point. The shorter the distance between two points is, the more similar two images are. The typical systems for content-based image retrieval include QBIC (Faloutsos et al., 1994; Flickner, Sawhney, Niblack, & Ashley, 1995; Niblack et al., 1993), Virage (Gupta & Jain, 1997), and Photobook (Pentland, Picard, & Sclaroff, 1995; Picard & Kabir, 1993), and WBIIS (Wang, Wiederhold, Firschein, & Wei, 1998). The similarity of two images is defined as the *Euclidean* distance between two feature vectors of images (Flickner et al., 1995; Jacobs, Finkelstein, & Salesin, 1995; Niblack et al., 1993; Ogle & Stonebraker, 1995; Wang et al., 1998). The usage of the feature vector of the entire image is sensitive to object extension/reduction and object location. Two images can be considered similar by feature vectors although they are different visually.

The fundamental reasons for the problems of previous research are that the number of features in query processing is insufficient and they are extracted from the entire image. Kastenmuller et al. developed a joint histogram for overcoming the limit of the insufficient features (Kastenmuller, Kriegel, & Seidl, 1998). The joint histogram is composed of various features and generated by the Cartesian product of features. They have shown by various experiments that the joint histogram can obtain more effective results than the color histogram.

The use of regions can override the limitations of the feature extraction from the entire image (Natsev et al., 2003; Smith & Chang, 1999; Smith & Li, 1999). The region is better than the entire image as the granularity of comparison, and it produces more accurate results (Jing, Mingjing, et al., 2002, 2002; Natsev et al., 2003; Smith & Li, 1999; Wang et al., 2001). Guibas considered a quadtree partitioning of an image where only windows with uniformly consistent features are preserved and other windows are discarded as containing multiple region information (Guibas, Rogoff, & Tomasi, 1995). For each window a Fourier signature is computed and compared to that of neighboring windows to determine if they are sufficiently alike. The drawbacks of this approach are that two images can be seen as different even though they are similar except that the same objects are at different locations. In addition, window sizes and locations are restricted, and thus it will not be able to efficiently handle scaling and translation of objects. In Smith and Chang (1999) and Smith and Li (1999), authors proposed a similarity measure based on a variable-sized window. However, this was location-dependent because the spatial information of objects was used.

To provide users with a simple querying interface and to reduce the influence of inaccurate segmentation, several image-to-image similarity measures combining information from all of the regions have been proposed (Greenspan, Dvir, & Rubner, 2000; Jing, Zhang, et al., 2002; Natsev et al., 2003; Petrakis & Faloutsos, 2002; Smith & Li, 1999; Wang et al., 2001). Such systems only require the users to assign a query image, and therefore relieve the users of selecting regions. Many efforts have been made to reduce the query processing time. These can be classified into two categories: the use of the traditional tree structures (Gong, Zhang, Chuan, & Sakauchi, 1994; Natsev et al., 2003; Thomas, Carson, & Hellerstein, 2000), such as the R^*-tree (Beckmann, Kriegel, Schneider, & Seeger, 1990) and the M-tree (Ciaccia, Patella, & Zezula, 1997), and the use of statistical clustering (Ma & Manjunath, 1997; Wang & Du, 2001). An alternative image similarity measure that allows partial image matching is described in Rubner, Thomasi, and Guibas

(1998). They proposed the earth mover's distance (EMD) using liner programming for matching histograms.

Recently, Natsev et al. considered the similarity model WALRUS that is a robust model for scaling and translation of objects within an image (Natsev et al., 2003). Each image is first decomposed into regions. The similarity measure between two images is then defined as the fraction of the area of the two images covered by matching regions. However, WALRUS focuses on the development of a fast and effective segmentation method instead of an image-to-image similarity measure. Consequently, region matching should be necessary before image matching. The authors proposed a greedy heuristic for computing the similar region pair set with the maximum area. The basic idea is to iteratively choose the best pair of matching regions that maximizes the area covered by the regions. The time complexity of the above greedy algorithm is $O(n^2)$, where n is the number of matching pairs obtained by the R^*-tree search. The following is an example of region matching by using a greedy method. In Natsev et al. (2003), authors adopted 0.085 as a normalized region similarity threshold. In this paper, the region similarity threshold is assumed to be 2.21, which was generated 0.085 times the longest distance of the 16 color space.

Example 1. Let $R(A)$ be a representation for a region of an image, where R and A denote region intensity and its area, respectively. Let a target image t and the query image q be $\{t_1, t_2, t_3, t_4\} = \{4(4), 7(2), 8(5), 3(1)\}$, $\{q_1, q_2, q_3\} = \{3(3), 2(4), 1(2)\}$, respectively. Two regions are selected as a matching pair if the distance between them is less than or equal to the given region similarity threshold, $\{(t_1, q_1), (t_1, q_2), (t_4, q_2), (t_4, q_3)\}$, are matching pairs. The similar region pair set with the maximum area is then $\{(t_1, q_1), (t_4, q_2)\}$, the area covered by which is $7 + 5 = 12$.

Time-series analysis is the most similar application to partial similarity image retrieval. If an image is partitioned into n regions and the feature vector of each region is m-dimensional, a region can be mapped to a point in an m-dimensional space and then an image is multi-dimensional time-series data according to the region visiting order. The number of regions is proportional to image complexity. Therefore, the problem of region-based image retrieval can be transformed into the subsequence matching of multi-dimensional time-series data.

The previous work of time-series analysis is mostly similarity search for one-dimensional data (Agrawal, Faloutsos, & Swami, 1993; Agrawal, Lin, Sawhney, & Shim, 1995; Rafiei & Mendelzon, 1997; Yi, Jagadish, & Faloutsos, 1998) and the research for a partial similarity search for time-series was made in Agrawal et al. (1995), Berchtold, Keim, and Kriegel (1997) and Faloutsos, Ranganathan, and Manolopoulos (1994). But previous research on the partial similarity search was made for one-dimensional time series data. Recently, Lee et al. has proposed a, filtering function of a partial similarity measure for multi-dimensional time-series data (Lee, Chun, Kim, Lee, & Chung, 2000). They used a continuous summation of distance of two corresponding points as a similarity measure, which is generally used for partial similarity search of time-series data. But it is not a noise-tolerant measure because it considers two time-series data to be dissimilar if they are different at some locations, or with a small amount of noise at all locations. Other research areas of partial similarity search involve categorical attribute clustering (Guha, Rastogi, & Shim, 1999), and CAD (Berchtold et al., 1997), etc. The partial similarity measure using continuous distance summation can be formulated as Eq. (1) and may be quite sensitive to a few noises.

$$D_{\min}(t,q) = \min_{l \leqslant m \leqslant k-l+1} = \sum_{i=m}^{l+m} |\vec{q}_{i-m+1} - \vec{t}_i|, \quad \text{where } t = \{\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_k\}, \;\; q = \{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_k\} \tag{1}$$

Example 2. Let t, q be a target sequence and a query sequence, respectively, $t = \langle (1,2), (3,5), (2,3), (4,5), (5,5), (2,2) \rangle$, $q = \langle (2,3), (4,5), (1,1), (2,2) \rangle$. We can obtain $\sqrt{21}$, $\sqrt{56}$ and $\sqrt{32}$ from $\sum_{i=m}^{l+m} |\vec{q}_{i-m+1} - \vec{t}_i|$. From the definition of $D_{\min}(t,q)$, the similarity value between t and q is $\sqrt{21}$. $\sqrt{21}$ is the distance between

$\langle(1,2),(3,5),(2,3),(4,5)\rangle$ and $\langle(2,3),(4,5),(1,1),(2,2)\rangle$. But the most partially similar pair is $\langle(2,3),(4,5),(5,5),(2,2)\rangle$ and $\langle(2,3),(4,5),(1,1),(2,2)\rangle$.

We developed an image-to-image similarity measure as a filtering function of a partial similarity measure for region-based image retrieval in order to reduce the overall query time in a large image database. Much previous research on the content-based image retrieval was focused on the development of a filtering function. But, to our knowledge, there has been no approach that is designed to reduce the overall query time in a region-based image search. Measures developed in the time-series area have no noise-tolerant properties even though they can measure the partial similarity of multi-dimensional time-series.

3. Partial similarity model

In this section, we define problems to be tackled in region-based image retrieval and propose a partial similarity measure to solve them. An image is segmented into a set of regions, the features of which are extracted by an appropriate algorithm. Table 1 gives the symbols and their corresponding meanings used in this paper.

3.1. Problem definition

If two images are similar in a large portion, and different only in certain regions, they should be considered to be similar (referred to as noise-tolerant similar).

- *Noise problem*: Noise is defined as the regions of an image that are not similar to any region in the other image. Most similarity measures of traditional approaches are via the *Euclidean* distance between two feature vectors, which may not detect partially dissimilar images if they compute similarity at the granularity of the entire image. This problem is referred to as the noise problem.

For this, the granularity for the comparison of two images should be done by regions rather than the entire image. Only the similar region or regions of the two images are counted in computing the partial similarity. The following summarize the problems to be tackled.

- *Different length problem*: The number of regions generally increases with the image complexity. Thus, the number of regions of two images may be different. This problem is defined as the different length

Table 1
Symbols and its meanings

Symbol	Meaning
D	*Euclidean* distance function
$H()$	Region-based histogram
HS	Constructor of HS histogram
$Sim()$	Similarity measure
$XH()XH2()$	XHistogram
$CXH()$, $CXH2()$	CXHistogram
$XSim()$, $CXSim()$	Proposed partial similarity measure
K	Flooring value of the region similarity threshold
τ	Region similarity threshold
ζ	Image similarity threshold
$Area2()$	Function for summation of all frequencies of a histogram
$Area()$	Function for counting of number of pixels in an image

problem. If a single feature vector is computed for each region, several high-dimensional points may represent an image. Region-based image retrieval techniques, therefore, cannot use traditional compression techniques that are popular as dimension reduction methods. To count the common regions disregarding their locations, the elements of two region sets should be compared in a sequential method, that is, pairwise way. A pair of regions is considered to be similar if the *Euclidean* distance between two regions is less than or equal to the given region similarity threshold, τ and *Seq*() is defined as a sequential search measure in Definition 1.

Definition 1 (*Image similarity*). Let t_i, a_m be regions of target image t and q_j, b_m of query image q. Let $(t_i, q_j) \in R, R$: set of matching pairs. Let $(a_m, b_m) \in A, A$: set of maximum matching pairs and $A \subseteq R$. Let *R_Seq*() be region similarity measure of *Seq*(). Two images t and q are said to be similar if there exists a similar region pair set $A = \{(a_1, b_1), (a_2, b_2), \ldots, (a_l, b_l)\}$ of t and q, such that

$$R_Seq(t_i, q_j) = \begin{cases} \dfrac{Area(t_i) + Area(q_j)}{Area(t) + Area(q)} & \text{if } (t_i, q_j) \in A \\ 0 & \text{otherwise} \end{cases}$$

$$Seq(t, q) = \sum_{(t_i, q_j) \in A} R_Seq(t_i, q_j) = \frac{Area(\cup_{i=1}^{l} a_i) + Area(\cup_{i=1}^{l} b_i)}{Area(t) + Area(q)} \geqslant \zeta \tag{2}$$

Seq() is the same as the image similarity measure adopted in Natsev et al. (2003). In the above definition, $Area(\cup_{i=1}^{l} b_i)$ is the number of pixels in the query image q covered by regions $b_1, b_2, \ldots, b_l$. Thus, two images are considered to be similar if the fraction of matched area compared to the total area of the two images is above the user-specified threshold, ζ.

- *Boundary value problem*: Two regions can be seen as dissimilar if they are mapped into different bins of a histogram even though they are similar by a distance-based measure. For example, adopting 2.21 as a threshold for similarity, two regions ⟨0,0,1⟩ and ⟨0,1,0⟩ are determined as similar by the *Euclidean* distance measure. However, they may be located in different bins such as b and c, and two dissimilar regions may be located in the same bin such as a and b. Then a and b are determined as similar, whereas b and c are not as shown in Fig. 3. This problem is defined as the boundary value problem.

Noise problem and different length problem are remedied by HS histogram with histogram intersection (refer to Lemma 1), while boundary value problem are remedied by the XHistogram approach. The following example illustrates the region and noise. An image is partitioned into many regions, each feature vector of which is extracted from a raw image.

Example 3. We consider the three-dimensional space. Let a target image be $\{\langle 1,2,4\rangle, \langle 3,5,2\rangle, \langle 2,3,5\rangle, \langle 4,5,1\rangle, \langle 1,3,2\rangle, \langle 2,4,3\rangle\}$ and the query image be $\{\langle 2,3,5\rangle, \langle 4,5,1\rangle, \langle 2,5,2\rangle, \langle 2,2,1\rangle\}$, where each element of two sets represents a region feature vector. The number of regions of the target image is 6 while the query image has four regions. The noises of two images are $\langle 1,2,4\rangle$, $\langle 3,5,2\rangle$, $\langle 1,3,2\rangle$, $\langle 2,4,3\rangle$, $\langle 2,5,2\rangle$ and $\langle 2,2,1\rangle$.

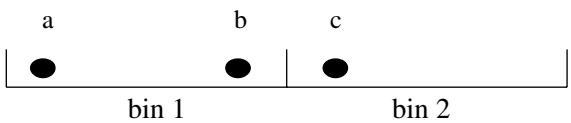

Fig. 3. Boundary value problem.

3.2. Partial similarity based on histogram intersection

The image histogram is a valuable tool used to view the intensity profile of an image. It is a bar graph of the pixel intensities. The pixel intensities are plotted along the x-axis and the numbers of pixels for each intensity along the y-axis. The histogram represents the occurrences of intensity of an image. We define a new histogram for a region-based image search named region-based histogram, which is a bar graph of the region intensities. All region intensities are plotted along the x-axis and the area of each region is plotted along the y-axis. A region area is defined as the number of pixels within the region. One-dimensional region intensity is considered; however, it can be easily extended to a multi-dimensional region intensity.

Example 4. Let $R(A)$ be a representation for a region of an image, where R and A denote region intensity and its area, respectively. Let a target image t and the query image q be {4(4), 7(2), 8(5), 3(1)}, {3(3), 2(4), 1(2)} as in Example 1 and the domain of the bins be [1, 9]. The region-based histogram, $H(t)$ and $H(q)$ for the image t and q are built as follows: $H(t) = \langle 0,0,1,4,0,0,2,5,0\rangle$, $H(q) = \langle 2,4,3,0,0,0,0,0,0\rangle$.

In Swain and Ballard (1991), the authors proposed a new histogram for computing similarity between two images, whose number of pixels for each intensity is computed by a histogram intersection method. Let the histograms of a target image t and the query image q be $H(t) = \langle t_1, t_2, t_3, \ldots, t_n\rangle$ and $H(q) = \langle q_1, q_2, q_3, \ldots, q_n\rangle$, respectively. A new proposed histogram is built using the histogram intersection and is defined as follows:

$$\langle \min(t_1, q_1), \min(t_2, q_2), \ldots, \min(t_n, q_n)\rangle.$$

We introduce the *Histogram Summation* (HS) histogram, which is generated by histogram intersection for the partial similarity comparison of two images. HS function has two histogram sets as its domain and range, respectively.

Definition 2 (*HS histogram*). Let histograms of a target image t and the query image q be $H(t) = \langle t_1, t_2, t_3, \ldots, t_n\rangle$ and $H(q) = \langle q_1, q_2, q_3, \ldots, q_n\rangle$. HS histogram is built using the histogram intersection and defined as

$$HS(H(t), H(q)) = \langle isum(t_1, q_1), isum(t_2, q_2), \ldots, isum(t_n, q_n)\rangle,$$

where

$$isum(t_i, q_i) = \begin{cases} t_i + q_i & \text{if } \min(t_i, q_i) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Example 5. This shows a case in which histograms are $\langle 0,0,1,4,0,0,2,5,0\rangle$ and $\langle 2,4,3,0,0,0,0,0,0\rangle$, as in Example 4. From Definition 2, the HS histogram of two images is $\langle 0,0,4,0,0,0,0,0,0\rangle$. It could be seen that the HS function disregards the dissimilar regions by the histogram intersection.

The following definition gives a partial similarity measure based on histogram intersection. Let $Area2()$ be a function for the summation of all frequencies of a histogram. If the histogram of an image is $H(t) = \langle t_1, t_2, t_3, \ldots, t_n\rangle$, then $Area2(H(t)) = \sum_{i=1}^{n} t_i$.

Definition 3 (*Partial similarity measure*). A partial similarity measure, $Sim()$, for the target image t and the query image q is defined as follows. $Sim()$ can have two regions as its parameters because a region is a subimage of an image:

$$Sim(t, q) = \frac{Area2(HS(H(t), H(q)))}{Area2(H(t)) + Area2(H(q))} \tag{3}$$

Lemma 1. *Sim*() *removes noises via the histogram intersection.*

Proof. Let histograms of a target image t and the query image q be $H(t) = \langle t_1, t_2, t_3, \ldots, t_n \rangle$, $H(q) = \langle q_1, q_2, q_3, \ldots, q_n \rangle$, respectively. Let an HS histogram be $HS(H(t),H(q)) = \langle a_1, a_2, a_3, \ldots, a_n \rangle$. From Definition 2, when $t_i = 0$, noises of the query image are removed, and then $a_i = 0$. When $q_i = 0$, noises of a target image are removed by the same method. Therefore, *Sim*() removes noises via the histogram intersection, consequently *Sim*() solves the different length problem. □

The more similar a target image is to the query image, the more likely *Sim*() approaches 1. *Sim*() = 0 means that a target image and the query image are absolutely different. The following example shows the meanings of *Sim*().

Example 6. In the case of Examples 5 and 1, *Sim*() and *Seq*() are 0.19 and 0.43, respectively.

$$Sim(t,q) = \frac{Area2(HS(H(t),H(q)))}{Area2(H(t)) + Area2(H(q))} = \frac{4}{12+9} = 0.19$$

$$Seq(t,q) = \frac{Area(\cup_m a_m) + Area(\cup_n b_n)}{Area(t) + Area(q)} = \frac{5+7}{12+9} = 0.43$$

This means that the similarity degree of two images by *Sim*() is 19%. $Sim(t,q)$ and $Seq(t,q)$ are different because a similarity measure based on the histogram intersection is an exact match rather than a similar match since it computes the summation value of two corresponding bins. A new histogram, the CXHistogram, is proposed to overcome the problem mentioned above.

3.3. CXHistogram and CXSim( )

The CXHistogram is introduced as a new image representation, and the CXSim() as a correct and efficient image-to-image similarity measure that aims at pruning irrelevant target images from a database.

For two arbitrary regions, t_i and q_j if $D(t_i, q_j) \leqslant \tau$, then the two regions are considered to be similar. However, they can be located in different bins and then $Sim(t_i, q_j) = 0$ even though $D(t_i, q_j) \leqslant \tau$. From Definition 3, they are determined to be dissimilar. The following Example 7 is considered for solving this problem: We assume that the region similarity threshold is 2.21 and a region feature vector is one-dimensional and its domain is $[1, 9]$. All domains of each dimension are extended by two times of the flooring value of τ on both the right and left sides. For example, if $\tau = 2.21$, all domains are extended by four. Correspondingly, if the domain of the bins is $[1, 9]$, it is extended to $[-3, 13]$. Our design objective is that *Sim*() have 1 as its value in the case that two images are similar in the vector space. Based on the above design objective, we can adopt an alternative value for the dimension extension. From the several experiments, we can discover that two times of the ceiling value of τ cannot satisfy the design objective but three times of the flooring value of $\frac{\tau}{2}$ is good at alternative value.

Example 7. Let the two images be 4(4) and 2(4). The histograms of the two images are modified as follows and named as *XH*2 and *XH*. *XH*2 is constructed by distributing the area of a region not only in the corresponding bin but also among the following nearby bins. The bins fall less than or equal to twice the flooring value of the given region similarity threshold, τ, both on the right side and left side. For *XH*, 1 time the flooring value of τ is applied. The area of the region is put in a parenthesis at its original intensity.

$XH2(t) = \langle 0,0,0,4,4,4,4,(4),4,4,4,4,0,0,0,0,0 \rangle$;
$XH(t) = \langle 0,0,0,0,0,4,4,(4),4,4,0,0,0,0,0,0,0 \rangle$;
$XH(q) = \langle 0,0,0,4,4,(4),4,4,0,0,0,0,0,0,0,0,0 \rangle$.

Then,

$$Sim(t,q) = \frac{Area2(HS(H(t),H(q)))}{Area2(H(t)) + Area2(H(q))} = \frac{5 \times 8}{20 + 20} = 1$$

$$Seq(t,q) = \frac{Area(\cup_m a_m) + Area(\cup_n b_n)}{Area(t) + Area(q)} = \frac{4+4}{4+4} = 1$$

Example 7 can be summarized in the following.

Observation. The histograms of two images are extended. The area of each regions of a target image is not only plotted in the corresponding bin but is also distributed among all nearby bins of that bin, which fall less than or equal to *twice* the flooring value of the given region similarity threshold, τ, both on the right side and left side of the corresponding bin. For a query image, 1 time the flooring value of τ is applied.

In Example 7, the cause of applying the HS function to $XH2(t)$ and $XH(q)$ is as follows: Let two regions be $t_i = a(4)$ and $q_j = b(3)$. If the region similarity threshold is 2.21, the flooring value of the given region similarity threshold, K is 2. If $D(t_i, q_j) \leqslant \tau$, the two regions are determined to be similar. $Sim()$ is to be used as the same purpose measure. In the case that two regions are similar in the vector space (i.e., the *Euclidean* distance between the two regions is within the threshold), $Sim()$ is designed to have 1 as its value. If $Sim()$ is less than 1, the two regions are considered as dissimilar. From Fig. 4, numerators of $Sim(t_i, q_j)$ are $(K+1) \times 7 = 21$, $(K+2) \times 7 = 28, \ldots, (2 \times K + 1) \times 7 = 35$, whereas the value of the denominator is 35. Consequently, although t_i and q_j are similar when $D(t_i, q_j) = 2$ and $D(t_i, q_j) = 1$, corresponding $Sim()$ values are $\frac{21}{35}$ and $\frac{28}{35}$, respectively, which are less than 1. If HS function is applied to $XH2(t_i)$ and $XH(q_j)$ or $XH(t_i)$ and $XH2(q_j)$ as shown in Fig. 5, then $Sim(t_i, q_j)$ has 1 as its value when $D(t_i, q_j) = 2$ and the number of overlapping bins is $2 \times K + 1$.

The XHistogram and the $XSim()$ are introduced as a new image representation and a new similarity measure, respectively. *2K-neighbors* and *K-neighbors* of a bin are defined as a set of all nearby bins whose distance from that bin is less than or equal to $2 \times K$ and equal to K, respectively. For example, if the region similarity threshold is 2.21, 2K-neighbors of 0 is $0, 1, 2, 3, 4$.

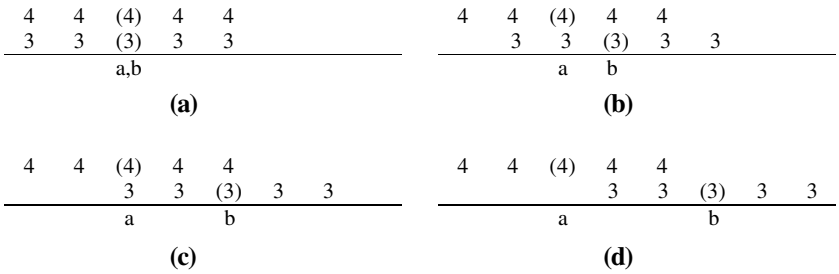

Fig. 4. Some examples for $HS(XH(t), XH(q))$: (a) $D(t_i, q_j) = 0$; (b) $D(t_i, q_j) = 1$; (c) $D(t_i, q_j) = 2$; (d) $D(t_i, q_j) = 3$.

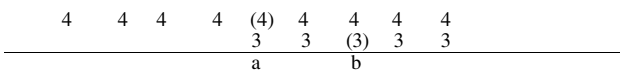

Fig. 5. An example for $HS(XH2(t), XH(q))$ when $D(t_i, q_j) = 2$.

Definition 4 (*XHistogram*, *XSim*()). The XHistogram is a region-based histogram. The area of each region is distributed among all the elements of its 2*K*-neighbors or *K*-neighbors. A partial similarity measure on the XHistogram, *XSim*(), for a target image t and the query image q is defined as follows:

$$XSim(t,q) = \frac{Area2(HS(XH2(t), XH(q)))}{Area2(XH(t)) + Area2(XH(q))} \tag{4}$$

2*K*-neighbors has a limitation (the same for *K*-neighbors). The number of elements of each 2*K*-neighbors may be different. Let the region similarity threshold and the domain of bins be 2.21 and [1, 9], respectively. 2*K*-neighbors of 1 is $\{1,2,3,4,5\}$, whereas 2*K*-neighbors of 5 is $\{1,2,3,4,5,6,7,8,9\}$. In solving this problem, the domain of bins is extended from [1, 9] into [−3, 13]. All domains of each dimension should extend to two times the *K* value on both the right and left sides.

Example 8 (XHistogram). Let a target image t and the query image q be $t = \{4(4), 7(2), 8(5), 3(1)\}$, $q = \{3(3), 2(4), 1(2)\}$, where a region similarity threshold is 2.21 and the domain of the bins [−3, 13]. Fig. 6 is the XHistogram for the target image t and can be rewritten in the following simple form:

$XH2(t) = \langle 0,0,1,5,5,5,7,12,12,12,12,11,7,7,7,5,0 \rangle$.

The XHistogram of $XH(q)$, $XH(t)$ can be built in the same way:

$XH(q) = \langle 0,0,2,6,9,9,9,7,3,0,0,0,0,0,0,0,0 \rangle$;
$XH(t) = \langle 0,0,0,0,1,5,5,5,7,11,7,7,7,5,0,0,0 \rangle$.

From the above $XH2(t)$, and $XH(q)$, the HS histogram, can be computed:

$HS(HX2(t), XH(q)) = \langle 0,0,3,11,14,14,16,19,15,0,0,0,0,0,0,0,0 \rangle$.

Therefore, the similarity of the two images is

$$XSim(t,q) = \frac{Area2(HS(XH2(t), XH(q)))}{Area2(XH(t)) + Area2(XH(q))} = \frac{92}{60+45} = 0.87$$

The following theorem indicates that the *XSim*() is a necessary condition of *Seq*().

Theorem 1. *The XSim() is a no false dismissals measure of Seq().*

Proof. Let t_i, q_j be region of target image t and query image q, respectively. We show that $Seq(t,q) \geqslant \zeta \Rightarrow XSim(t,q) \geqslant \zeta\ \forall (t_i, q_j) \in R$, R:set of region pairs. Let $(a_m, b_n) \in A$, A:set of maximum matching region pairs and $A \subseteq R$. Let *R_XSim*() be region similarity measure of *XSim*() and *R_Seq*() and *Seq*() is defined in Definition 1.

$$R_XSim(t_i, q_j) = \frac{Area2(HS(XH2(t_i), XH(q_j)))}{Area2(XH(t)) + Area2(XH(q))}$$

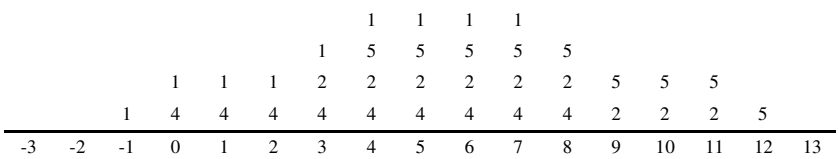

Fig. 6. XHistogram.

(a) $D(t_i, q_j) \leqslant \tau$ and $(t_i, q_j) \in A$,

$$\begin{aligned} R_XSim(t_i, q_j) &= \frac{Area2(HS(XH2(t_i), XH(q_j)))}{Area2(XH(t)) + Area2(XH(q))} \\ &= \frac{(2K+1) \times (Area((t_i)) + Area(q_j))}{(2K+1) \times (Area2(H(t)) + Area2(H(q)))} \\ &= \frac{Area(t_i) + Area(q_j)}{Area(t) + Area(q)} = R_Seq(t_i, q_j) \end{aligned}$$

(b) otherwise,

$$\begin{aligned} R_XSim(t_i, q_j) &= \frac{Area2(HS(XH2(t_i), XH(q_j)))}{Area2(XH(t)) + Area2(XH(q))} \\ &= \frac{r \times (Area((t_i)) + Area(q_j))}{(2K+1) \times (Area(t) + Area(q))}, \\ &\quad \text{where } r \in \{0, 1, 2, \ldots, 2K\} \geqslant 0 = R_Seq(t_i, q_j) \end{aligned}$$

From (a) and (b)

$$XSim(t, q) = \sum R_XSim(t_i, q_j) \geqslant \sum R_Seq(t_i, q_j) = Seq(t, q)$$

$\therefore Seq(t, q) \geqslant \zeta \Rightarrow XSim(t, q) \geqslant \zeta$. □

However, the XHistogram has a large number of bins and thus takes a very large space. For example, the number of bins of an image whose color model is RGB and whose resolution is 24, is 13,824 (that is, 24^3). The CXHistogram is used for solving the above limitation.

Definition 5 (*CXHistogram*, *CXSim*()). The CXHistograms are, denoted by *CXH*2() and *CXH*(), regions-based histograms and are constructed by simply clustering the XHistograms, *XH*2() and *XH*(), respectively. The XHistogram is clustered by the equalized quantization which combines the same number of adjacent bins into one bin. *CXSim*(*t*,*q*) is the similarity measure of two images based on the CXHistogram and is defined as

$$CXSim(t, q) = \frac{Area2(HS(CXH2(t), CXH(q)))}{Area2(CXH(t)) + Area2(CXH(q))} \tag{5}$$

The CXHistogram of *XH*2(*t*) in Example 8 is ⟨1,15,31,35,21,5⟩ if three adjacent bins are clustered into 1 bin.

Theorem 2. *The CXSim() is a no false dismissals measure of the XSim().*

Proof. We show that $XSim(t, q) \geqslant \Rightarrow CXSim(t, q) \geqslant \zeta$, where ζ: image similarity threshold. Let *K_HS*(*CXH*2(*t*), *CXH*(*q*)) be *K*th value of HS histogram for *CXH*2(*t*) and *CXH*(*q*). Let *ij_HS*(*XH*2(*t*), *XH*(*q*)) be the sum of *i*th and *j*th value of HS histogram for *XH*2(*t*) and *XH*(*q*). Let $XH2(t) = \langle t_1, t_2, \ldots, t_m \rangle$, $XH(q) = \langle q_i, q_2, \ldots, q_m \rangle$, $CXH2(t) = \langle T_1, T_2, \ldots, T_n \rangle$, $CXH(q) = \langle Q_1, Q_2, \ldots, Q_n \rangle$, where $m \geqslant n$. Without loss of generality, we assume that $T_k = t_i + t_j$, $Q_k = q_i + q_j$:

(i) $t_i = q_i = t_j = q_j \neq 0$, $K_HS(CXH2(t), CXH(q)) = t_i + t_j + q_i + q_j = ij_HS(XH2(t), XH(q))$;
(ii) $t_i = 0$, $K_HS(CXH2(t), CXH(q)) = t_j + q_i + q_j \geqslant t_j + q_j = ij_HS(XH2(t), XH(q))$;
(iii) $t_j = 0$, $K_HS(CXH2(t), CXH(q)) = t_i + q_i + q_j \geqslant t_i + q_i = ij_HS(XH2(t), XH(q))$;

(iv) $q_i = 0$, $K_HS(CXH2(t), CXH(q)) = t_i + t_j + q_j \geqslant t_j + q_j = ij_HS(XH2(t), XH(q))$;
(v) $q_j = 0$, $K_HS(CXH2(t), CXH(q)) = t_i + t_j + q_i \geqslant t_i + q_i = ij_HS(XH2(t), XH(q))$;
(vi) $Area2(CXH(q)) = Area2(XH(q))$, $Area2(CXH(t)) = Area2(XH(t))$ from (i) to (v).

$$CXSim(t,q) = \frac{\sum_K Area2(K_HS(CXH2(t), CXH(q)))}{Area2(CXH(t)) + Area2(CXH(q))} \geqslant \frac{\sum_{i,j} Area2(ij_HS(XH2(t), XH(q)))}{Area2(XH(t)) + Area2(XH(q))} = XSim(t,q)$$

$\therefore XSim(t,q) \geqslant \zeta \Rightarrow CXSim(t,q) \geqslant \zeta$. □

Corollary 1. *The CXSim() is a no false dismissals measure of Seq().*

Proof. From Theorems 1 and 2, the CXSim() is a no false dismissals measure of *Seq*(). □

Corollary 1 means that the CXSim() can be used as a filtering function for *Seq*() and its performance is measured in Section 5.

4. XMage query processing()

This section presents the overall process of XMage which consists of the *R**-tree filtering, CXSim() filtering, and refinement step. The query processing of XMage is a range query based on image similarity threshold and pre-processing for image segmentation and building of the CXHistogram should be made for all target images for later processing (Fig. 7):

- *Image parsing*: We use the sliding windows for obtaining a set of regions of an image (Faloutsos, Ranganathan, et al., 1994; Natsev et al., 2003). Some sliding windows are clustered into a region using the modified video shot detection algorithm (Lienhart, Pfeiffer, & Effelsberg, 1997). If the Euclidean distance of feature vectors of two sliding windows exceeds a threshold, a new region is generated. The similar regions are clustered afterwards. The region detection algorithm is different from the shot detection algorithm with respect to spatial connectivity. For example, the window, (left, top, right, bottom) = $(0,0,9,9)$, may be clustered into a region with the window $(2,4,11,13)$. That is, all windows connected spatially are considered to be neighboring. The average of RGB color values and areas of regions for all target images are saved into databases. An image parsing for the query image is executed in query time.
- *Color space*: A color space is a way of representing colors and their relationship to each other. Different image processing systems use different color models for different reasons. The color picture publishing

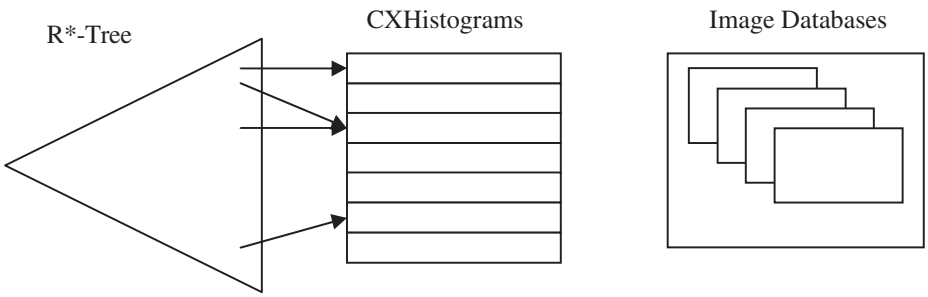

Fig. 7. File structure for the proposed method.

industry uses the CMY color model. Color CRT monitors and most computer graphics systems use the RGB color model. Systems that must manipulate hue, saturation, and intensity separately use the HIS color model. Human perception of color is a function of the response of three types of cones. Because of that, color systems are based on three numbers (called tristimulus values). It is well-known that RGB space is not perceptually uniform, i.e., perceptually small visual color differences do not always correspond to small RGB color distance. There exist color spaces that are much more perceptually uniform such as the CIE L^*a^*b/L^*u^*v spaces, which are known to improve the perceptual non-uniformity of RGB space by a factor of 80:6. Therefore, it is self-explaining that the usage of the CIE L^*a^*b/L^*u^*v spaces is better than that of the RGB space, producing more accurate results.

- *Color clustering*: Most color-based image retrieval systems adopt the color histogram as the representation of the distribution of color in the image. One of the most important steps in these systems is to reduce histogram dimensions with the least loss in color content (Wang et al., 1997). In this paper, we perform the equalized quantization to the RGB color space in order to change color levels from 256 to 16 in each axis.
- *Determine the K value*: The K value is region similarity threshold and used for building the CXHistogram. It is determined via an experimental analysis by a domain expert. In our experiments, 2.21 was adopted as the K value. In Natsev et al. (2003), authors adopted 0.085 as a normalized region similarity threshold. In this paper, the region similarity threshold, K is assumed to be 2.21, which was generated 0.085 times the longest distance of the 16 color space.
- *Determine 2K-neighbors*: 2K-neighbors of a bin is defined as a set of all nearby bins whose distance from that bin is less than or equal to $2 \times K$. Each region intensity has its own 2K-neighbors. 2K-neighbors will be used when the XHistogram is built.
- *Build up the CXHistogram*: All areas of each region of the image is added to the areas of all the elements of 2K-neighbors of the bin corresponding to the region, as shown in Example 8. The XHistograms and the CXHistograms are then generated.

After the CXHistograms of all target images have been constructed, partial similarity search is performed against a given query image. The following is an algorithm for XMage in a large image database using the R^*-tree and the CXSim() (Fig. 8).

Procedure PartialImageSearch()
input: query image q, target image t, the R^*-Tree of a set of target image regions $\{t_i\}$
output: target images set ψ partially similar to the query image
begin
1. $\psi := \emptyset$ /* answer set of target images */
2. Step 0 : Decompose q into $\{q_1, q_2, q_3, \ldots, q_m\}$ regions
3. Step 1 : Build up the $CXHistogram$ of the query image q
4. Step 2 : (Filtering step)
5. 1) R^*-tree filtering step
6. **for** each region of query image, q_j **do**
7. **if** $D(t_i, q_j) \leq \tau$ **then** store selected candidate images containing t_i into α_j
8. compute $\alpha = \cup_{j=1}^{m} \alpha_j$
9. 2) $CXSim()$ filtering step
10. **For** each target image t in α **do**
11. **if** $CXSim(t, q) \geq \zeta$ **then** $\psi \leftarrow \psi \cup \{t\}$
12. Step 3 : (Refinement step)
13. **For** each selected image t in the set ψ **do**
14. **if** $Seq(t, q) < \zeta$ **then** $\psi \leftarrow \psi - \{t\}$
15. Step 4 : Return the answer set ψ
end

Fig. 8. Algorithm for partial similarity image search.

5. Experimental results

In order to measure the effectiveness of the proposed method, extensive experiments were conducted over an environment containing a large volume of image data. The experiments focused on showing the accuracy and the efficiency of the CXSim(). In WALRUS, the authors examined a two-step query processing, i.e., the R^*-tree filtering and sequential search, where our proposed method consists of the, R^*-tree and the CXSim() filtering and sequential search (Fig. 2). The two-step query processing mentioned above is called *IndexSeq*. A comparison was made between XMage and IndexSeq. In order to evaluate the quality of image retrieval, a real-life image dataset called *misc*, which was downloaded from the website of the image search engine WBIIS at Stanford University, was used. The *misc* database contained 10,000 images stored in the JPEG format. The sizes were either 85×128, 96×128 or 128×85. For the experiments, 1–2000 images were selected from the *misc* database for our experiments. The system was written in Microsoft Visual C++ and run under Windows 2000 on a Pentium IV processor.

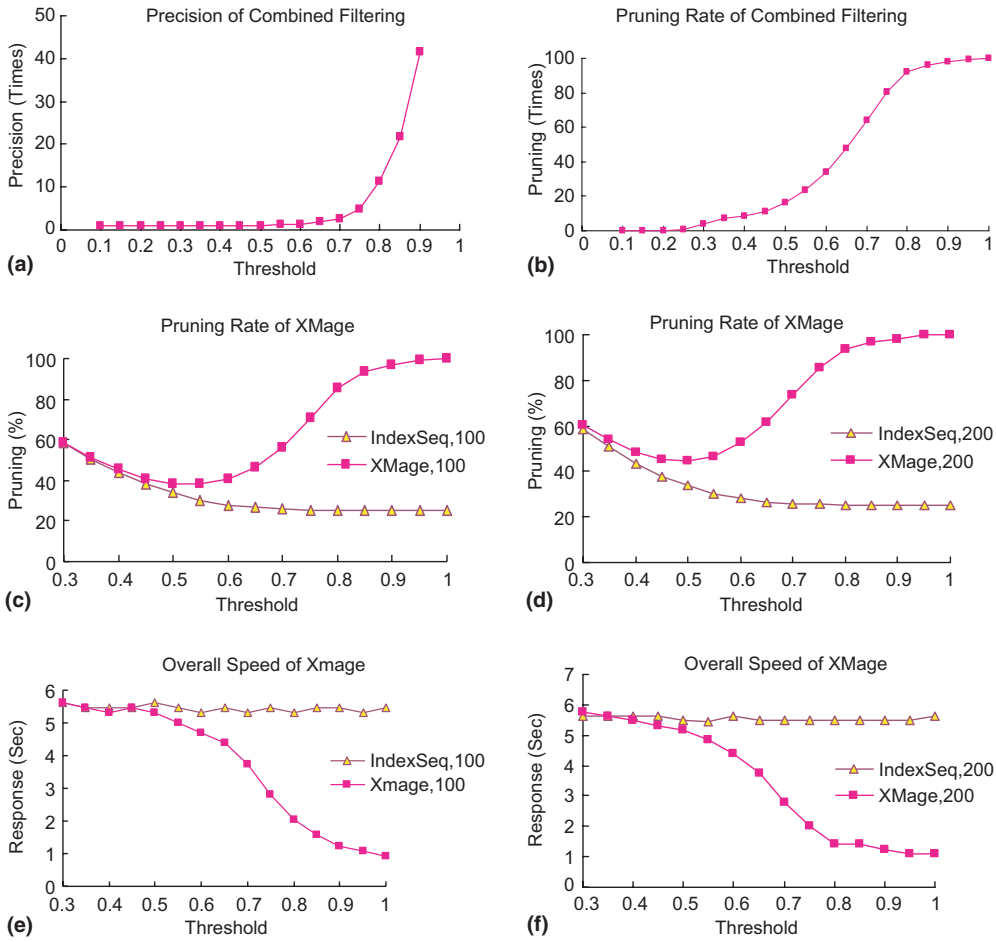

Fig. 9. Performance of XMage: (a) the ratio of the precision of the combined filtering to that of the R^*-tree with 200 bins; (b) the ratio of the pruning rate of the combined filtering to that of the R^*-tree with 200 bins; (c) pruning rate for XMage and IndexSeq with 100 bins; (d) pruning rate for XMage and IndexSeq with 200 bins; (e) overall response time for XMage and IndexSeq with 100 bins; (f) overall response time for XMage and IndexSeq with 200 bins.

The overall pruning efficiency and query processing time of XMage were examined by varying the image similarity threshold and we also examined the recall/precision of the filtering step for the R^*-tree and the CXSim(). To measure the pruning efficiency, the pruning rate (PR) was used as follows:

$$\mathrm{PR} = \frac{|\text{images actually pruned}|}{|\text{images to be pruned}|} = \frac{|\text{total images}| - |\text{retrieved images} \cup \text{relevant images}|}{|\text{total images}| - |\text{relevant images}|} \tag{6}$$

where $|S|$ denotes the cardinality of the set S.

5.1. Effectiveness of CXSim()

In XMage, the CXSim() reduces the false alarms occurred after the R^*-tree search. The recall and precision of the combined filtering of the R^*-tree and the CXSim() were examined. Based on the results, it could be observed that the recall of the combined filtering was always 1. This indicates that the combined filtering has the 'no false dismissals' property. Fig. 9(a) shows the ratio of the precision of the combined filtering to that of the R^*-tree. In Fig. 9(a), the precision of the combined filtering was 1–41 times higher compared to that of the R^*-tree search for the image similarity threshold range $[0.1, 0.9]$ where the number of bins of the histogram was set at 200. When the image similarity threshold was 0.95, the ratio is 214. This indicates that the combined filtering effectively reduced the false alarms that occurred during the R^*-tree search. Fig. 9(b) shows the pruning efficiency of the combined filtering. It could be observed that the combined filtering prunes 0–98% false alarms that are contained in the result of the, R^*-tree search for the image similarity threshold range $[0.1, 1.0]$ and the same number of bins.

Fig. 10. Images found by XMage when 0.7 image similarity threshold.

5.2. Efficiency of the proposed method, XMage

XMage is based on an assumption that there is ground truth available to evaluate item relevance for a specific query. Our primary objective is to improve the efficiency of query processing and the image matching of WALRUS is a fairly thorough comparison method, we consider the query result obtained WALRUS approach as query ground truth. They use a greedy algorithm for aggregation of region matches into image matches.

Fig. 9(c) shows the pruning efficiency, where the number of bins was set at 100. The pruning rate of XMage was 38–100% while that of IndexSeq was 25–58%, for the image similarity threshold range $[0.3, 1.0]$. When the number of bins was set at 200, the pruning rate of XMage was 45–100% and that of IndexSeq was 25–58% (Fig. 9(d)). Fig. 9(e) and (f) shows the overall response time. We observe that XMage requires 0.9–5.6 and 1.1–5.7 s response time for the image similarity threshold range $[0.3, 1.0]$ when the numbers of bins of the histogram was set at 100 and 200, respectively. The response time of XMage was 1.1–5.9 times faster than that of IndexSeq for the image similarity thresholds with $[0.3, 1.0]$.

To show the accuracy of XMage, the visual similarity of images retrieved by XMage were compared to a given query image. The query image with an ID of 866 in the database was selected. The top 15 images were found by XMage (Fig. 10) from 2000 images and were all visually similar to the query image except for

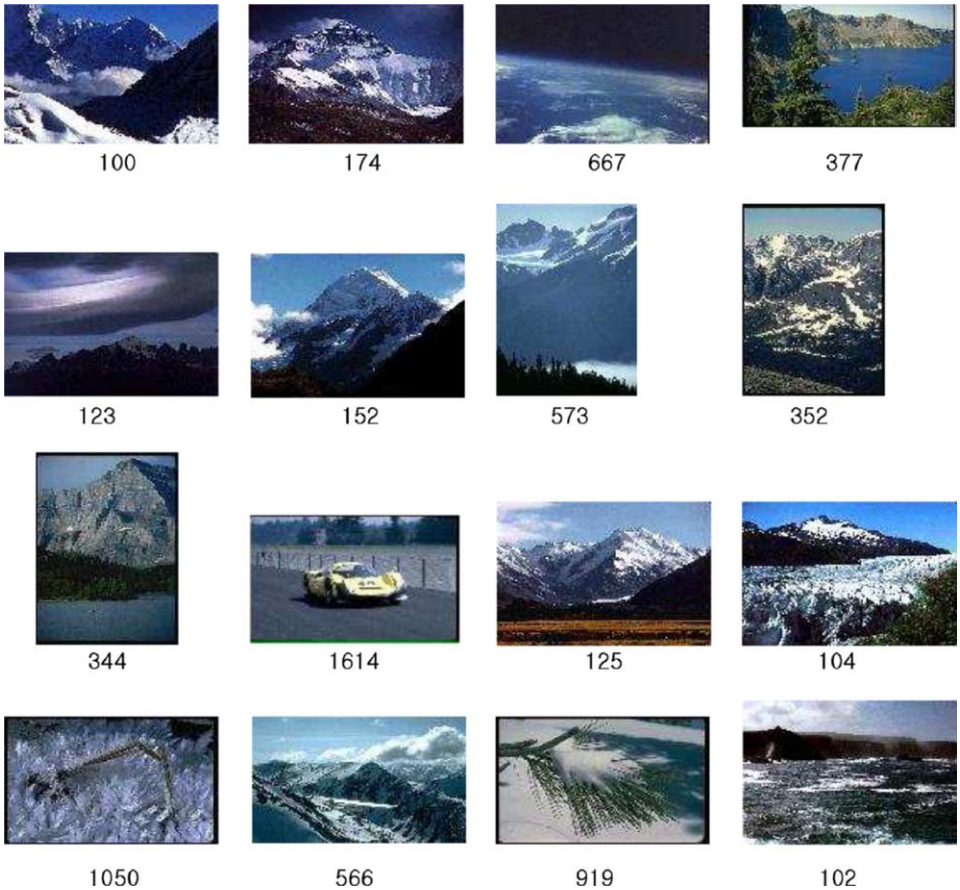

Fig. 11. Images found by XMage when 0.7 image similarity threshold.

image 1605 image. The images are ordered top to bottom, left to right, in decreasing order of similarity. The top 15 similar images to the query image with an ID of 100 were shown in Fig. 11. The images were obtained by using 2.21 and 0.7 as the region similarity threshold and image similarity threshold, respectively.

6. Conclusions and future works

This paper focused on developing an image-to-image similarity measure, CXSim(), based on the Histogram Intersection in order to reduce the overall query time while retrieving partially similar images from a large image database. The proposed measure supports location-independent and noise-tolerant searches of images data. It also retrieves candidate images satisfying the criteria of no false dismissals. The major contributions of this work include the introduction of a new image representation CXHistogram, and the use of the CXSim() as an image-to-image similarity measure. The CXSim() achieves the effect of comparing regions of two images by simply comparing the two images. The experiment on real-life images, the CXSim() is an efficient measure to prune irrelevant images, being approximately 1.1–5.9 times faster than IndexSeq for the image similarity thresholds with 0.3–1.0, when the number of bins is set at 100. The CXSim() will be revised for use as a refinement function with very few false dismissals or a more efficient filtering function with no false dismissals.

Acknowledgements

This research was supported by the Agency for Defense Development, Korea, through the Image Information Research Center at Korea Advanced Institute of Science & Technology.

References

Agrawal, R., Faloutsos, C., & Swami, A. (1993). Efficient similarity search in sequence databases. In *Proc int'l conf. on foundations of data organization and algorithm* (pp. 69–84).

Agrawal, R., Lin, K. l., Sawhney, H. S., & Shim, K. (1995). Fast similarity search in the presence of noise, scaling, and translation in time series databases. In *The VLDB con.* (pp. 490–501).

Beckmann, N., Kriegel, H.-P., Schneider, R., & Seeger, B. (1990). The *R**-tree: An efficient and robust access method for points and rectangles. In *Proc. of the ACM SIGMOD Int'l conference on management of data* (pp. 322–331).

Berchtold, S., Keim, D. A., & Kriegel, H.-P. (1997). Using extended feature objects for partial similarity retrieval. In *Proc. of the ACM SIGMOD int'l conference on management of data* (pp. 333–348).

Ciaccia, P., Patella, M., & Zezula, P. (1997). M-tree: An efficient access method for similarity search in metric spaces. In *Proc. of the int'l conference on very large data bases*.

Faloutsos, C. et al. (1994). Efficient and effective querying by image content. *Journal of Intelligent Information System*, 231–262.

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequences matching in time-series databases. In *Proc. of the ACM SIGMOD int'l conference on management of data* (pp. 419–429).

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., et al. (1995). Query by image and video content: the QBIC systems. *IEEE Computer, 28*(9), 23–32.

Gong, Y., Zhang, H. J., Chuan, H. C., & Sakauchi, M. (1994). An image database system with content capturing and fast image indexing abilities. In *Proc. of IEEE int'l conf. on multimedia computing and systems* (pp. 121–130).

Greenspan, H., Dvir, G., & Rubner, Y. (2000). Region correspondence for image matching via EMD flow. In *IEEE workshop on content-based access of image and video libraries*.

Guha, S., Rastogi, R., & Shim, K. (1999). ROCK: A robust clustering algorithm for categorical attributes. In *Proc. of 15th int'l conf., on data engineering* (pp. 512–521).

Guibas, L., Rogoff, B., & Tomasi, C. (1995). Fixed-window image descriptors for image retrieval. In *SPIE symposium on electronic imaging: Science and technology* (pp. 352–362).

Gupta, A., & Jain, R. (1997). Visual Information Retrieval. *Communications of the ACM, 40*(5), 70–79.

Jacobs, C. E., Finkelstein, A., & Salesin, D. H. (1995). Fast multiresolution image querying. In *Proc. ACM SIGGRAPH*.

Jing, F., Mingjing, L., Zhang, H. J., & Zhang, B. (2002). An effective region-based image retrieval framework. In *Proc. ACM multimedia*.

Jing, F., Zhang, B., Lin, F. Z., Ma, W. Y., & Zhang, H. J. (2002). A novel region-based image retrieval method using relevance feedback. In *Proc. 3rd ACM int'l workshop on multimedia information retrieval* (MIR).

Kastenmuller, G., Kriegel, H.-P., & Seidl, T., 1998. Similarity search in 3D protein databases. In *Proc. German conference on bioinformatics (GCB'98)*, Koln.

Lee, S.-L., Chun, S.-J., Kim, D.-H., Lee, J.-H., & Chung, C.-W. (2000). Similarity search for multidimensional data sequences. In *Proc. of 20th int'l conf. on data engineering* (pp. 599–608).

Lienhart, R., Pfeiffer, S., & Effelsberg, W. (1997). Video abstraction. *Communication of the ACM, 40*(12), 54–62.

Ma, W. Y., & Manjunath, B. S. (1997). NETRA: A toolbox for navigating large image database. In *Proc. IEEE Int'l conf. on image processing* (pp. 568–571).

Natsev, A., Rastogi, R., & Shim, K. (2003). WALRUS: A similarity retrieval algorithm for image databases. *IEEE Transactions on Knowledge and Data Engineering, 15*(5), 1–16.

Niblack, W., et al. (1993). The QBIC project: Query image by content using color, texture and shape. In *In storage and retrieval for image and video databases (SPIE)* (pp. 173–187).

Ogle, V. E., & Stonebraker, M. (1995). Chabot: Retrieval from a relational database of image. *IEEE Computer, 28*(9), 40–48.

Pentland, A., Picard, R. W., & Sclaroff, S. (1995). Photobook: Tools for content-based manipulation of image databases. In *In Storage and retrieval for image and video databases (SPIE)* (pp. 34–47).

Petrakis, E. G. M., & Faloutsos, C. (2002). ImageMap: An image indexing method based on spatial similarity. *IEEE Transactions of Data Engineering, 14*(5), 979–987.

Picard, R. W., & Kabir, T. (1993). Finding similar patterns in large image databases. In *IEEE ICASSP* (pp. 161–164).

Rafiei, D., & Mendelzon, A. (1997). Similarity-based queries for time series data. In *Proc. of the ACM SIGMOD int'l conference on management of data* (pp. 13–25).

Rubner, Y., Thomasi, C., & Guibas, L. J. (1998). The earth movers's distance as a metric for image retrieval. In *Technical Report STAN-CS-TN-98-86*. Computer Science Department, Stanford University.

Smith, J. R., & Chang, S.-F. (1999). Integrated spatial and feature image query. *Multimedia Systems Journal, 7*(2), 129–140.

Smith, J. R., & Li, C.-S. (1999). Image classification and querying using composite region templates. *Journal of Computer Vision and Pattern Recognition, 75*(1/2), 165–174.

Swain, M. J., & Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision, 7*(1), 11–32.

Thomas, M., Carson, C., & Hellerstein, J. (2000). Creating a customized access method for Blobworld. In *Proc. of the 16th int'l conf. on data engineering*.

Wang, J. Z., & Du, Y. P. (2001). Scalable integrated region-based image retrieval using IRM and statistical clustering. In *Proc. ACM and IEEE joint conf. on digital libraries*.

Wang, J. Z., Li, J., & Wiederhold, G. (2001). SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*, 947–963.

Wang, J. Z., Wiederhold, G., Firschein, O., & Wei, S. X. (1998). Content-based image indexing and searching using daubechies' wavelets. *International Journal of Digital Libraries (IJODL), 1*(4), 311–328.

Wang, J., Yang W.-J., & Acharya, R. (1997). Color clustering techniques for color-content-based image retrieval from image database. In *Proc. of int'l conf. on multimedia computing and systems*.

Yi, B. K., Jagadish, H. V., & Faloutsos, C. (1998). Efficient retrieval of similar time sequences under time wraping. In *Proc. of the 14th int'l conf. on data engineering* (pp. 201–208).