Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Visual Languages and Computing
14 (2003) 151–179
Journal of Visual Languages & Computing
www.elsevier.com/locate/jvlc

Effective and efficient region-based image retrieval

Mario A. Nascimento*, Veena Sridhar, Xiaobo Li

Department of Computing Science, University of Alberta, Alta., Edmonton, Canada

Received 30 May 2002; received in revised form 30 October 2002; accepted 5 November 2002

Abstract

Content-based image retrieval (CBIR) is a challenging task. Current research works attempt to obtain and use the semantics of image to perform better retrieval. Towards this goal, segmentation of an image into regions has been used in recent years, since local properties of regions can help matching objects between images and thereby contribute towards a more effective CBIR. This paper improves on a CBIR technique, called SNL (Sridhar, Nascimento, Li) that utilizes the regional properties of the images. In SNL each image is segmented and features including the color, shape, size and spatial position of the obtained regions are extracted. Regions are then compared using the integrated region matching (IRM) distance measure, which is not a metric, which prevents the use of metric access structures or filtering techniques based on the triangle inequality. We overcome this issue, by using MiCRoM, a true metric distance to compare segmented images. This resulting approach, called SNL*, can be used in conjunction with a filtering technique to reduce substantially the number of images compared. Albeit metric-based, SNL* is computationally expensive. We address this drawback, in the SNL^+ approach, where we replace the expensive metric distance in SNL* by the inexpensive original (non-metric) IRM distance. We found that one can still make use of the same filtering technique, at the expense of little loss in retrieval effectiveness. Thus, the main contribution of this paper is SNL^+, a very effective and highly efficient region-based image retrieval technique.

*Corresponding author. Tel.: +1-780-492-5678; fax: +1-780-492-1071.
E-mail addresses: mn@cs.ualberta.ca (M.A. Nascimento), veena@cs.ualberta.ca (V. Sridhar), li@cs.ualberta.ca (X. Li).

1045-926X/03/$ - see front matter
doi:10.1016/S1045-926X(02)00076-9

1. Introduction and motivation

Image database management and retrieval has been an active research area since the 1970s [1]. With the rapid increase in computer speed and decrease in memory cost, image databases containing thousands or even millions of images are used in many application areas such as medicine, satellite imaging, and biometric databases, where it is important to maintain a high degree of precision. With the growth in the number of images, manual annotation becomes infeasible both time and cost-wise. Content-based image retrieval (CBIR) is a powerful tool since it searches the image database by utilizing visual cues alone. CBIR systems extract features from the raw images themselves and calculate an association measure (similarity or dissimilarity) between a query image and database images based on these features. CBIR is becoming very popular because of the high demand for searching image databases of ever-growing size. Since speed and precision are important, we need to develop a system for retrieving images that is both efficient and effective.

Recent approaches to represent images require the image to be segmented into a number of regions (a group of connected pixels which share some common properties). This is done with the aim of extracting the objects in the image. However, there is no unsupervised segmentation algorithm that is always capable of partitioning an image into its constituent objects, especially when considering a database containing a collection of heterogeneous images. Therefore, an inaccurate segmentation may result in an inaccurate representation and hence in poor retrieval performance.

In [2] we introduced SNL,[1] a region-based CBIR technique. SNL uses a clustering approach [3] to obtain regions of (potential) interest in images. Regions from two images can then be matched using the IRM heuristic [4] which ultimately helps determining the distance between the two images. Although SNL has been shown to be effective, its efficiency has not yet been thoroughly investigated.

SNL's query processing yields a linear search (and image comparison) over the whole database. One way of improving its efficiency would be to reduce the number of image comparisons done at query time. This can be achieved by using a metric access structure (e.g. [5]) or a filtering technique (e.g. [6]). However, these alternatives require the use of a metric distance, which is not the case of the distance implied by IRM. We address this issue by substituting IRM in SNL by the MiCRoM metric distance [7], resulting in the SNL* technique. We then equipped SNL* with the Omni-filter [6], which can potentially reduce the number of image comparisons made at query time.

The main assumption for using the Omni-filter is the employment of a true metric distance, otherwise relevant answers can be mistakenly left out of the answer set. When comparing SNL with SNL* we observed that IRM was a very good approximation for MiCRoM, i.e. IRM is a near-metric distance, and is much less expensive to compute. We then devised the so-called SNL$^+$ technique which simply

[1] SNL stands for the initials in Sridhar, Nascimento and Li who originally proposed the technique in [2].

incorporates the Omni-filter into the original SNL technique and obtained not only effective, but also more efficient query performance.

The remainder of this paper is organized as follows. Section 2 discusses some related work in the field of content-based image retrieval using visual attributes like color, shape, spatial position and also some works related to region-based image retrieval. Section 3 presents our new CBIR approach, SNL, which focuses on a color representation that is not very sensitive to segmentation inaccuracies and also accounts for other features of regions. That section further discusses several experiments which show the effectiveness of the SNL approach. Section 4 describes SNL* and SNL$^+$, an optimal but slow and a nearly optimal but very fast SNL-based techniques respectively along with several experiments pointing to SNL$^+$ as a very good alternative for region-based image retrieval. Finally, the paper is concluded with a summary of our findings and some directions for future work.

2. Related work

2.1. Primitive features

Several features have been used to represent images in CBIR systems. The most commonly used feature is color. Global color histogram (GCH) is a simple and effective way of utilizing the color features [8]. The GCH is an n-dimensional vector $C = \{C(1), C(2), \ldots, C(n)\}$, where each element C_j represents the percentage of pixels of color j in an image. Another color-based approach was proposed in [9], where an image was represented with the help of the first three moments namely the color average, variance and skewness. Pass et al. [10] proposed a new method using the color coherence vectors (CCV). They proposed a histogram-based approach that incorporated some spatial information as well. The image is initially blurred to remove small differences between pixels and then the color space is discretized to n-colors. Pixels within a bucket were classified as either coherent or incoherent depending on whether they were part of a large similar-colored region. Both GCH and CCV are invariant to scaling and rotation and very simple to compute, but take into account only the distribution of colors, disregarding the inherent relation between the bins. Therefore, bin definition or color quantization is a critical issue. Another drawback of approaches based only on a global color representation, such as GCH and CC, is that they do not consider the spatial location of the colors present in an image.

To avoid some of the problems stated above, local color histograms were proposed. An image is partitioned into equal-sized sub-images/blocks and the similarity between two images is based on the histogram distances between corresponding blocks. This method is not capable of handling geometric transformations like rotation and translation and it suffers from problems like cell-cross talk [11] and variance to absolute spatial location. In this context cell/color histograms (CCH) have been recently proposed [12]. It was an effort to elegantly combine the information represented by local histograms in a partition-based approach and global color

histograms. The representation takes advantage of the fact that a low number of distinct quantized colors are usually present in images to lower its space overhead.

Region-based CBIR techniques have become increasingly more important and are reviewed in more detail shortly (Section 2.2). Nonetheless, an extensive review of the current state-of-art in the area of color-based CBIR, can be found elsewhere, e.g. [11,13,14].

Shape, next to colors, is considered an important characteristic in describing the salient objects in images and can help discriminate between two images and therefore in retrieval. Shape extraction involves several steps. The first step is to use a suitable segmentation method to divide the image into regions. Segmentation techniques can be classified into three broad categories: region-based, boundary-based and pixel-based. Region-based segmentation methods include region growing by pixel aggregation, region splitting and merging techniques. Edge detection technique is a common boundary-based method and thresholding is a popular pixel-based segmentation method. Once the image is segmented and regions are obtained, features belonging to the obtained regions should be recorded. Any segmentation technique mentioned above can use any of the representation schemes. Chain codes [16] use the 8-connectivity or the 4-connectivity to represent the line segments that constitute the boundary of a region. Signatures, shape numbers and polygonal approximation are other representation schemes.

The next step is to use appropriate descriptors for these regions so that they can be used while matching regions of different images. Shape descriptors are classified into three types. Boundary-based descriptors define the properties of the boundary (2-D closed curve) or the exterior of a region. Boundary-based techniques mainly use the outline of the region to calculate shape. Fourier descriptor is one of the well-known methods belonging to this category (e.g. [15]). In this technique, the boundary of a given region is obtained and Fourier transformed [16]. The dominant Fourier coefficients are used as the shape descriptors. Other descriptors in this category are shape numbers and moments [16]. Regional descriptors, on the other hand, describe the content or the interior of the region. Moment Invariants [1] is the most commonly used descriptor. Hu [17] proposed seven such moments and there were several papers, e.g. [18,19] that improved upon his idea. Area, calculated as the total number of pixels in a region, minimum/maximum bounding rectangle/circle/ellipse and the ratio between the sides, radii, and length of the radius are other regional descriptors. Compactness, measured as the ratio between the squared perimeter and area, Elongatedness, which is the ratio between the length of the longest chord in the region and the chord perpendicular to it, are also examples of descriptors belonging to this category.

The disadvantage of most of the shape-based retrieval systems is that boundary-based techniques are applicable only to "sketch-databases" i.e. databases with images that contain the sketch of a single object only. For using region-based descriptors, obtaining a region is a major problem. So due to this inaccuracy of the region itself, the descriptors may become ineffective.

Texture is a powerful regional descriptor that helps in the retrieval process. Texture, on its own does not have the capability of finding similar images, but it can

be used to classify textured images from non-textured ones and then be combined with another visual attribute like color to make the retrieval more effective. One of the popular representations of texture feature is the co-occurrence matrix proposed by Haralick et al. [20]. The matrix is based on pixel orientation and inter-pixel distance. Meaningful statistics from the co-occurrence matrix are extracted and represented as texture information. Tamura et al. [21] proposed a method to extract six visual texture properties namely coarseness, contrast, directionality, likeliness, regularity and roughness.

A number of color/texture/shape descriptors have also been designed and tested for similarity retrieval, extraction, storage and representation complexities, and have been approved for the MPEG-7 standard (e.g. [22,23]). The color descriptors include a histogram descriptor coded using Haar transform, and color layout and color structure histogram. The shape descriptors are based on contours and based on moments. These descriptors are expected to become more commonly available in the future.

2.2. *Towards semantic features*

Obtaining the semantics or the meaning of an image is one of the most current research topics in the area of image retrieval. Visual features alone are not enough to distinguish between images. For example, there might be two images—that of a blue sky and the other of a blue sea. Using color, texture and other attributes they might be deemed similar, but semantically they are completely different. Of course, it cannot be denied that without the help of visual features, it is very hard (if possible at all) to derive the semantics of an image, unless they are annotated manually. One of the most important factors in a semantic based retrieval system is to not just look at the image on the whole, but in fact, to look at the objects in the image and to try and find relationships between these objects. Partitioning or segmenting the image into regions may reveal the "true" objects within an image. Local properties of regions could help in understanding these objects, thus contributing to more meaningful image retrieval. For this purpose, it is important to partition the image into its constituent objects. There are several image retrieval systems that adopt a region-based approach.

In Blobworld [24], objects are recognized by segmenting the image into regions that have roughly the same color and texture. Each pixel is then associated with a vector that consists of color, texture and spatial features. A model of the distribution of pixels is developed in an 8-D space. The distance between two images is calculated as the distance between the blobs in terms of color and texture.

Netra [25] is another image retrieval system which segments images into regions of homogeneous color and then uses the color, texture, shape and spatial properties for measuring similarity. Both Blobworld and Netra require the user to select the region of interest from the segmented image and only this region is then compared with regions in other images in the database thus avoiding noise during the matching process. There are however some disadvantages of this method. The user is burdened with the task of selecting his object of interest, when in fact the segmentation may

not have yielded regions close to the human perception of an object. The other problem is that humans often tend to associate objects with the background and other surrounding information to give it some meaning. So depending on the background where a particular object is present, users may perceive the same object differently.

An attempt towards capturing the semantics to help find similar images was made by Wang et al. [26] and Stehling et al. [3]. In Simplicity [26], the authors make use of semantics to classify images into the following categories: Textured vs Non-textured using the well-known χ^2 measure and Graphs vs Photographs using the probability density of wavelet coefficients in high-frequency bands. They first segment the images by dividing the image into 4×4 blocks and then extract a feature vector consisting of six features (three of which are the average color components and the other three indicate the energy in high-frequency bands of wavelet transforms). Then a K-means algorithm [27] is used to cluster these feature vectors. While [26] makes use of the color of each region to find similar images within a category of images, in [3] the color and the spatial position of each region is used. The distance used by both [26] and [3] to compute the similarity between the images is the IRM measure proposed in [4]. The advantage of the IRM distance is that it is not affected by over or under segmentation because it considers all the regions in an image.

3. SNL: a segmentation-based CBIR technique

3.1. Motivation

The SNL technique is a segmentation-based CBIR technique that utilizes a more effective representation of image regions and a more accurate image similarity/distance calculation. SNL attempts to model the properties of objects within an image in a way that is closer to human perception, thus generating a more meaningful association (distance) measure between images.

With all the techniques mentioned in Section 2.2, the segmentation results obtained using a single set of parameters on thousands of images may not always correspond to human perception of objects. This is illustrated with an example shown in Fig. 1, where we see a query image A and two database images, B and C. On careful inspection of the segmented images, we notice that all three contain "something" at the center, surrounded by a green background. Also we notice that the "things" presented in images A and C have a lot of colors in common including black, white and some gray patches. Based on this information, most retrieval techniques would deem images A and C to be more similar than images A and B. However, when one looks at the actual images shown in Fig. 2, we see that A and B are both images of tigers and are certainly more similar than A and C.

It is clear that a correct image segmentation is not enough. An accurate representation of the image regions is important, even critical, in measuring image similarity. Along this line, the proposed SNL technique contains three parts: image segmentation, feature extraction, and similarity calculation.

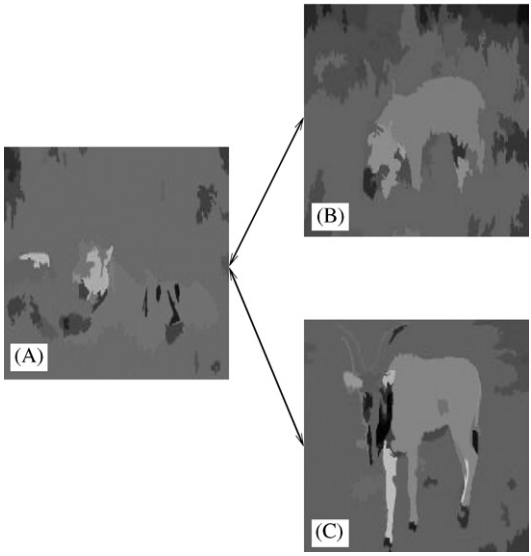

Fig. 1. Motivation for SNL (segmented images).

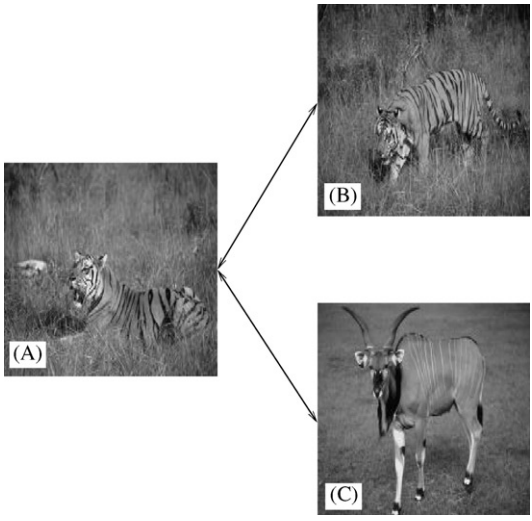

Fig. 2. Motivation for SNL (original images).

3.2. Segmentation

The first step in our retrieval technique is to segment the image into regions that (ideally) would correspond to the objects present in the image. For this purpose, we

need a segmentation algorithm that is effective in rendering homogeneous regions in a short time.

A recent paper by Stehling et al. [3] presents a single-link region growing algorithm, named CBC, used along with a minimum spanning tree. The algorithm does not require the number of output clusters, and can be described as follows. The image is first converted into a graph whose vertices are the pixels in the image and whose edges are neighborhoods of four pixels. The weight of each edge is the Euclidean distance between the colors of the four-pixel neighborhoods. The pixels are clustered using two thresholds: color threshold and size threshold. A set of connected pixels whose color similarity is greater than or equal to the color threshold forms a region. Then, regions less than the given size threshold are considered to be noise and hence merged with the nearest neighbor having the greatest similarity in terms of color. In [3] the color and size thresholds in the CBC's segmentation step were set to be 3 and 0.1, respectively. The authors suggest that this set of parameters result in a good compromise between the number of regions, effectiveness and robustness. The small size threshold in CBC results in many details in the form of small regions. As the size threshold is increased, fewer regions are obtained. As we shall discuss later, the SNL approach uses a different size threshold in order to obtain less regions but retaining more information per region.

The clustering algorithm proposed here is not only automatic, but also uses spatial and color features and takes less time (4 s[2] for a 512×512 image). Hence we decided to use this clustering algorithm to obtain regions in the image.

Nevertheless, one must note that several other segmentation techniques could be used, e.g. those used in the works mentioned in Section 2.2 [24–26]. However, in this paper we do not aim at contributing towards the area of image segmentation itself. Rather, we aim at using metadata obtained from a segmentation process in order to enhance the CBIR task.

3.3. Feature extraction and representation

The next phase is the regional feature extraction phase, wherein the segmented images are analyzed and a feature vector is constructed for each region.

One of the most effective features that helps in distinguishing one image from another is color. As mentioned before, the problem with any segmentation/clustering algorithm is that a single set of parameters cannot be applied to all the images in the database, especially when considering a miscellaneous collection. Even within an image, it would make more sense if some objects had a more detailed representation than others. The segmentation algorithms mentioned before, cluster pixels on the basis of the most significant colors present in the image and tend to ignore or merge smaller segments with the larger ones closest to them, either in terms of color or spatial location or some other property. It is definitely true that significant colors help in identifying similarity between images, but they also lead to a lot of false positives. For instance, a yellow sunflower, yellow sun and a yellow ball (of the same

[2] On a 533 MHz Celeron PC with 128 Mb of RAM running Linux 2.2.16.

size) would all be segmented into roughly circular regions with the dominant color which is yellow. In terms of the mean color of the region, size and shape they would all be deemed very similar. But semantically they are not similar at all. In fact, the subtle difference between them can be brought out by the less dominant colors in the region, e.g. the black center in the sunflower and the orange tinge in the sun. Thus, from the above discussion we can infer that while the dominant colors help in finding regions that are similar to each other, less dominant colors help in eliminating false positives. For this reason, we decided to represent the color feature of each segment with its histogram which gives us the distribution of colors in that region. Thus, for each region i in the image I, we have a color histogram representation, $C(i)$.

Other regional features of interest, but not as important from the retrieval perspective as shall be clear in the experiments, are: size, shape, and spatial location. How those are represented and used is discussed next.

3.4. The similarity/distance measure

Next to image representation, similarity measure is one of the key items in the process of image retrieval that decides the effectiveness and the efficiency of the retrieval technique. In the case of retrieval using regions of an image, it is important to choose a similarity measure that is robust to segmentation inaccuracies. It is also important that the measure agrees with the human perception of similarity and is easily computable. Since images have been decomposed into their respective segments, the similarity between two images is in fact the similarity between their constituent segments. As mentioned in the previous section, each region is represented by its spatial location, shape, color and size. Hence, to compare two regions their respective normalized features should be compared.

The distance between the spatial positions of two regions, i of image $I1$ and j of image $I2$, is calculated as the Euclidean distance between the centers of the two regions. This is shown below, i.e.

$$D_S(i,j) = \sqrt{(X(i) - X(j))^2 + (Y(i) - Y(j))^2}, \tag{1}$$

where $X(i)$ and $Y(i)$ are the x and y coordinates of the centers of the regions.

The shape distance is the distance between the height/width ratios of the MBRs[3] enclosing two regions i and j of images $I1$ and $I2$, respectively. It can be computed as

$$D_E(i,j) = |e(i) - e_(j)|, \tag{2}$$

where $e(i)$, $e(j)$ are the height/width ratios.

So far, we measured the distance between two regions in terms of shape which is a boundary feature and in terms of the spatial position. The distance between two regions in terms of their content i.e. color and size, can be calculated using the

[3] Minimum Bounding Rectangle, i.e. the smallest rectangle which would enclosed a given object (an image region in this case).

following equation:

$$D_C(i,j) = \frac{\sum_{k=0}^{k=N} |C(i)[k] - C(j)[k]|}{\sum_{k=0}^{k=N} C(i)[k] + \sum_{k=0}^{k=N} C(j)[k]}, \tag{3}$$

where $C(i)$, $C(j)$ are the color histograms of regions i of $I1$ and j of $I2$ containing N bins each.

The rationale behind using the above measure is explained using an example. Consider two sets A and B with elements (e.g. colored pixels), and that an element from A matches an element of B if they are equal (e.g. both pixels have the same color). Then one can consider the number of unmatched objects as the distance between A and B. More formally we have

$$D_C(A,B) = \frac{\#(unmatched)}{\#(A) + \#(B)}, \tag{4}$$

where $\#(x)$ is the cardinality of x and *unmatched* is the set of objects that are not present in both A and B. We can rewrite the numerator as $\#(unmatched) = \#(A \cup B) - \#(A \cap B) = \#(A) + \#(B) - 2 \times \#(A \cap B)$. Therefore, equation $D_C(A,B)$ can be written as

$$D_C(A,B) = \frac{\#(A) + \#(B) - 2 \times \#(A \cap B)}{\#(A) + \#(B)} = 1 - \frac{2 \times \#(A \cap B)}{\#(A) + \#(B)}. \tag{5}$$

In this last equation, the term $2 \times \#(A \cap B)/(\#(A) + \#(B))$ happens to be Dice's coefficient [28]. More importantly however, is that this distance is a metric since it follows the three metric axioms [28].

Thus, we have separately measured the distance between the spatial position, shape and the content of regions. But to differentiate between the regions, we need a single overall measure, which can be obtained by combining these three distances. Distance between two regions i and j of images $I1$ and $I2$ is defined as

$$D^{SNL}(i,j) = \alpha \times D_C(i,j) + \beta \times D_S(i,j) + \gamma \times D_E(i,j), \tag{6}$$

where D_C is the distance between the region content and α is the weight assigned to it, D_S is the spatial distance with its corresponding weight β and D_E is the shape distance between two regions with weight γ.

In order to measure the similarity between two images the IRM proposed in [26] is used. The IRM measure to calculate the distance $D_I(I1, I2)$, between two images $I1$ with m regions and $I2$ with n regions is calculated as discussed in [4]. The main idea behind the IRM measure is to match images completely. The inter-region distances between all pairs of regions in the two images are computed. The two most similar regions (least inter-region distance) are completely matched, if the regions have the same size, otherwise a partial match occurs and the unmatched portion is matched with some other region. This process is repeated until all the regions are matched completely.

The process of calculating the IRM measure requires quadratic time since we need to compare all segments of image $I1$ with all segments of image $I2$. In our case, however, due to our configuration, we obtain only a few regions (5 regions on an average, for color threshold = 3, size threshold = 1 in the segmentation algorithm)

as opposed to the originally proposed CBC (40 regions on an average, for color threshold = 3, size threshold = 0.1 in the segmentation algorithm). Therefore we can afford to use this measure.

For every query image, the extracted regional features are compared with the meta-data of all the images in the database using the distance formulae and then using the IRM measure, the image similarities are computed. After obtaining the similarities, the database images are re-ranked in the order of decreasing similarity (or increasing distance).

Let us now exemplify the above with an example (Fig. 3), which shall also serve to highlight SNL's strength. For simplicity, let us assume that our color palette consists of only three colors: black, gray and white.

In the first case, we illustrate the fact that SNL is capable of perceiving changes such as rotation and in the second case, we point out the importance of using a histogram representation for the color property of a region. In this example, we compare image Q with images A, B and C. We know that image A is a rotated version of image Q and is assumed to be more similar to Q than B. It is also clear that image C is not the same as image Q because C contains some "candy canes". Therefore, if the human perception of distance between two images i, j is termed as $H(i,j)$, then the assumptions we made earlier are $H(Q,A) < H(Q,B)$ and $H(Q,C) \sim 0$, i.e. small but not null.

The distance between image Q and the other three images, A, B and C is calculated using the above-mentioned techniques and is shown in Table 3.

When GCH is applied, $D_I(Q,A) = D_I(Q,B) = 0$ because the color composition of Q, A and B are the same. Due to difference in color composition, the distance between Q and C determined by the GCH technique is much greater than 0. Thus, GCH does not agree with our assumption of similarity. From this particular case, we can deduce that color composition is important, but it is not enough to differentiate between images where the spatial distribution of colors is different.

For applying SNL and CBC, images need to be segmented. Q and A are segmented into two regions each, $Q1$, $Q2$ and $A1$, $A2$ and B is segmented into four

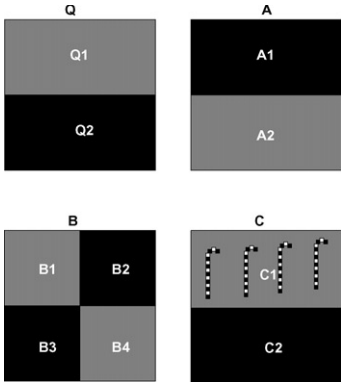

Fig. 3. Sample image set.

regions $B1$, $B2$, $B3$ and $B4$. In C, the smaller regions constituting the "candy canes" are merged with region $C1$ to form a single region with the average color, gray. The second region is $C2$. When CBC technique is applied, $D_I(Q,A) \neq 0$ and $D_I(Q,B) \neq 0$. The reason is that the matching technique takes into account the color and also the spatial location of the region. However, since they do not consider the shape properties of the regions, $D_I(Q,A) > D_I(Q,B)$ as seen from Table 1. Also, $D_I(Q,C) = 0$. This is because, during segmentation the small regions inside $C1$ were merged with it and the average color was represented. It is quite contrary to what human beings would likely perceive.

SNL determines the distance between Q and A to be smaller than the distance between Q and B (see Table 1) since SNL uses the color, size, spatial location and the shape of each region. SNL is also capable of distinguishing Q from C despite the disadvantage of the segmentation process, since it retains the content (i.e. color histogram) of the region—recall that CBC keeps only the quantized average color which may be not enough to distinguish the two regions. SNL satisfies both the assumptions made earlier and is therefore better suited to represent human perception of similarity. Thus, using some example figures we have illustrated that we combine the advantages of GCH and CBC to make our technique more similar to human perception.

Earlier, in Fig. 2, we gave three examples of real-life images. Images A and B are more related (they are both pictures of tigers) than A and C. Fig. 1 also shows the segmented images A', B' and C'. In this example, we shall compare the distance between image A and the other two images calculated using GCH, CBC and SNL as shown in Table 2. Here again, we know that human beings would perceive images A and B to be more similar than images A and C, since A and B are images of tigers in a similar background. The assumption made here can be stated as $H(A,B) < H(A,C)$.

Table 1
Distance calculation using the three techniques

Techniques	$D_I(Q,A)$	$D_I(Q,B)$	$D_I(Q,C)$
GCH	0	0	0.2
CBC	0.062	0.048	0
SNL	0.075	0.308	0.07

Table 2
Distance calculation using the three techniques

Techniques	$D_I(A,B)$	$D_I(A,C)$
GCH	0.316	0.220
CBC	0.043	0.038
SNL	0.137	0.150

While GCH and CBC determine images A and C to be more similar than A and B, SNL deems B to be more similar to A than C, thus agreeing with our assumption on human notion of similarity. Thus, we see how false positives can be avoided by SNL.

3.5. Experiments and results

In this section, we discuss about the evaluation measures used and the experiments performed. Three sets of experiments were conducted to observe and measure the performance of the proposed retrieval technique. The first experiment relates to the quantization scheme to be applied to the RGB color space. Similar experiments were also performed for the HSV color space, and can be found elsewhere [2,29]. In the second set of experiments, weights to be assigned to the content, spatial and shape features of each region are determined. The third set of experiments, presents the performance of SNL technique in comparison to the GCH and the CBC technique proposed by Stehling et al. [3]. The experiments were performed on two heterogeneous database containing 10,000 and 50,000 images with 15 query images.[4] Each of these 15 query images have a set of relevant images that are similar in color distribution and are semantically related to it. These images were originally used in [30], and manually assembled using a different dataset than the one we use for our experiments—we believe that this minimizes the chances of biasing the result sets. One should also note that our proposed approach does not manipulate the images' raw data but rather their metadata (regions and color histograms for those regions), thus query processing time does not depend on the size of the database images.

The most popular way to evaluate the performance of a retrieval system is to calculate the percentage of relevant documents retrieved and also their relative order. Ideally, a system should retrieve all the relevant documents first keeping the number of non-relevant documents that are retrieved before the relevant ones, as minimum as possible. Recall [31] is the percentage of the total relevant documents retrieved and is defined as

$$\text{Recall} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}}.$$

Precision refers to the capability of the system to retrieve only the relevant documents. Precision can be expressed as

$$\text{Precision} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}.$$

The first experiment was done to select a good quantization scheme for the RGB color space. We used about 10,000 images to test the performance of the RGB color spaces for various quantization levels. The color property of each region is represented with a uniformly quantized color histogram consisting of 27, 64 and 125 bins in the RGB color space with all color channels being equally important.

In Fig. 4, it is seen that the performance of the 64-color quantization is the best and the curve is drastically pulled down by an increase in the quantization space.

[4] http://www.cs.ualberta.ca/~mn/CBIRone/.

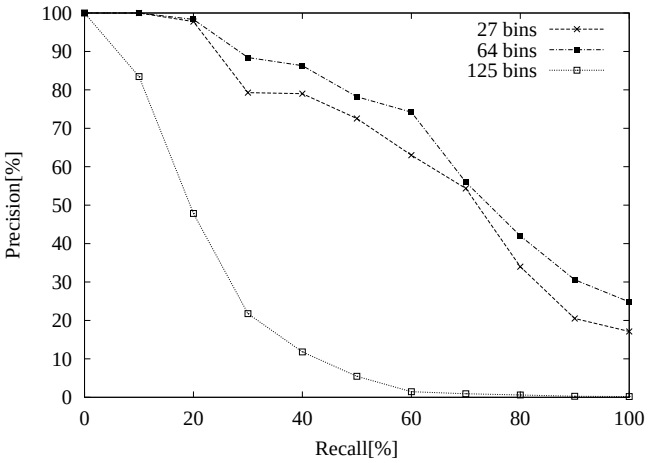

Fig. 4. Performance variation with various quantization levels in the RGB space.

This is because two colors which are very similar to each other can be classified into two different bins and since only a one-to-one difference between the bins is calculated, the distance between two similar colors is increased. Decreasing the number of bins also affects the performance because with just 27 bins the separability between colors is reduced. The performance is not affected as much due to the fact that the regions obtained from segmentation are homogeneous in color to some extent and 27 colors are sufficient to represent the colors within such a homogeneous region. Since the 64 color quantization scheme in the RGB color space was the best, we adopt it for the rest of our future experiments.

In the previous section, we discussed about calculating the distance between two regions. This distance is a weighted sum of the region content distance, shape distance and the spatial distance between any two regions. The second experiment was done to decide on the values to be assigned to α, β and γ. Again a set of 10,000 images was considered and the importance of each of these 3 features was studied by assigning different values for α, β and γ. In Fig. 5, we observe that color is clearly the most important feature that affects the retrieval performance. Shape and spatial properties do not account for the performance very much. Thus, we know that the value of α has to be higher than both β and γ. To further refine these weights, we decided to consider a few sample points to calculate the average precision for all recall values in a database of 10,000 images. The graph corresponding to this experiment (Fig. 6) indicated that an α value of 0.7, and β and γ equal to 0.15 each yielded a very good result in terms of effectiveness.

The third experiment compared SNL with the CBC technique [3] and GCH. Since CBC was proposed recently and claimed to perform better than CCV [10] and Color Moments [9], and those, in turn, were claimed to outperform earlier techniques we believe that CBC represents an effective and representative region-based CBIR approach. The reason for comparing also with GCH is to have a well-known baseline which is also used in virtually every published work in the related literature.

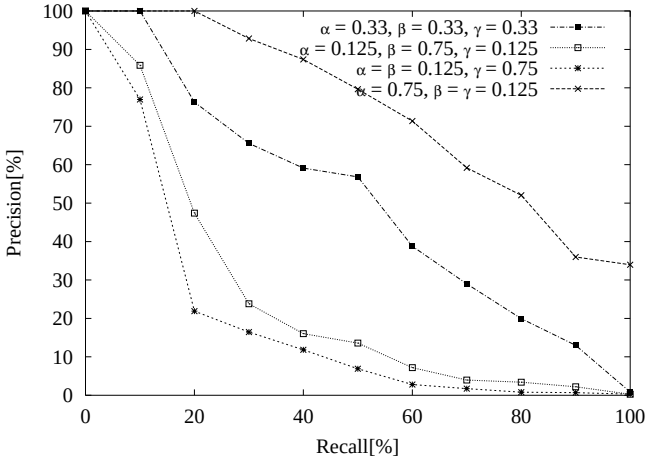

Fig. 5. Performance variation with varying importance to different features.

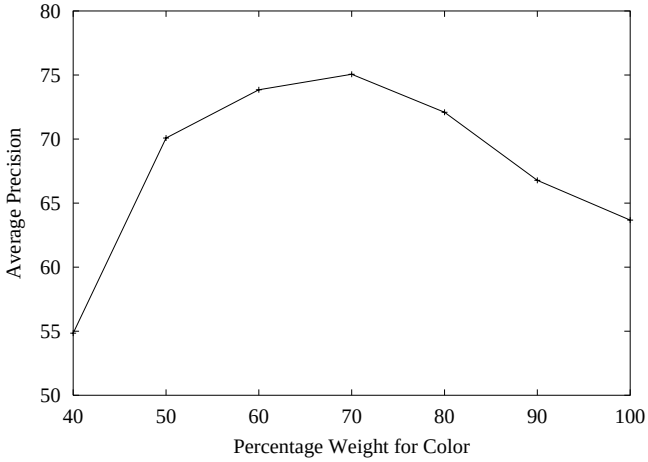

Fig. 6. The average precision at varying color weights for 10,000 images.

In CBC's original paper [3] the color and size thresholds in the CBC's segmentation step were set to be 3 and 0.1, respectively. We felt that for a region to be more meaningful, its size needed to be at least 1% of the total image size. Hence in the case of SNL, we set the color and the size thresholds to be 3 and 1, respectively. Moreover, since SNL is robust to segmentation inaccuracies a higher size threshold does not affect the results and in fact leads to a smaller number of regions that need to be compared during query time. Recall that unlike CBC which retains only the average color of the obtained regions, SNL retains a whole color histogram for the regions. We should also stress that this is not a modification per se of CBC's segmentation algorithm, rather it is just a customized setting that suits

better SNL's goals. Since it is important to see how well our technique scales up, we also used a set of 50,000 images in addition to the set of 10,000 images used earlier. Figs. 7 and 8 indicate the performance of the three compared techniques, where one can clearly see that the SNL technique performs better than both the GCH and CBC.

Thus we infer that the SNL technique in the RGB (as well as the HSV color space, not shown here) scales up well. The performance of the SNL technique in the previous three graphs also indicates that it is able to handle false positives well. As the database size increases, the number of false positives also increase proportionally. Since the performance of the SNL technique did not decrease with the increase in false positives, SNL seems to be a better technique when compared to CBC and GCH.

Storage space is also an important measure for the efficiency of a technique. Even though it is no longer as critical an issue as it used to be about 10 years ago, it is nevertheless not negligible. The storage requirements of GCH, CBC and SNL are listed in Table 3. In GCH, each image uses about 81 integers (assuming a 81 bin uniform quantization in the HSV color space). Thus it requires only about 162 bytes (assuming two bytes per integer). In the case of CBC, an average of 40 regions are obtained and each region stores three float values for color, two float values for the spatial position and one float value for the size of the region and therefore requires about 960 bytes (assuming four bytes per float value). In SNL in the RGB color space, on an average, we obtain about five regions and each region requires about 64 integers for the histogram, one float value for the size of each region, one float value for the shape of each region and two float values for the position of each region. Therefore it uses a total of 720 bytes, i.e. it is not only more effective than CBC but also more economical in terms of storage requirement. As one can see, SNL is not nearly as economical as GCH but it is not only conceptually more elaborate but also much more effective.

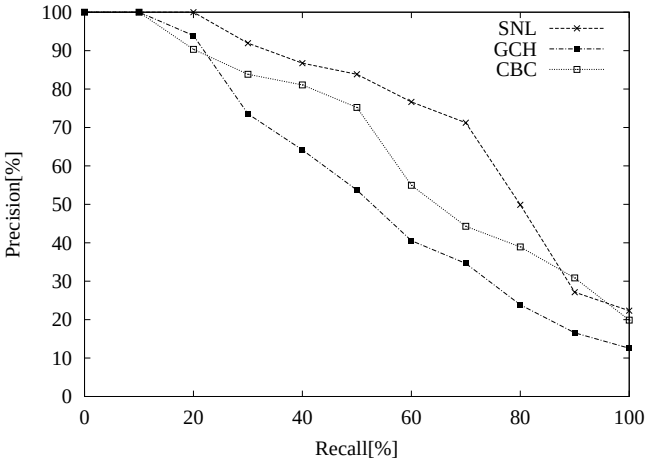

Fig. 7. Comparing different techniques using a database of 10,000 images.

Table 3
Storage space for the three techniques

Techniques	Storage (in bytes)
GCH	162
CBC	960
SNL	720

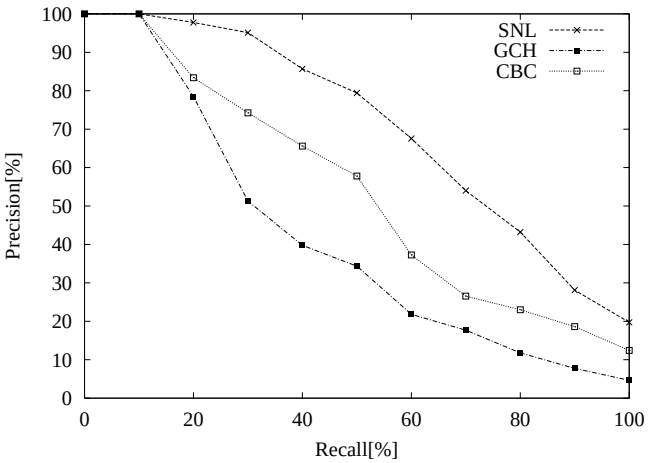

Fig. 8. Comparing different techniques using a database of 50,000 images.

4. SNL* and SNL$^+$: metric-based and fast SNL versions

The proposed SNL technique is effective from the precision-recall point of view. However, there are two issues that need to be further addressed: (1) IRM is an heuristic (non-metric) distance, and (2) efficiency, i.e. query processing time. This section presents two versions of SNL: SNL* and SNL$^+$. SNL* substitutes the IRM distance by MiCRoM [7], which is a true-metric distance. This allows the use of a filtering technique, called Omni [6], which reduces substantially the number of image comparison that need to be done. However, as we shall see next, the MiCRoM metric is computationally expensive, making it a non-practical choice. To overcome this we investigate SNL$^+$, which is built on the idea of using the original IRM distance, even though it is not a metric (i.e. the result set may not be complete), in conjunction with the Omni technique. Our extensive experiments show that SNL$^+$ is a technique just as effective as SNL and SNL*, and much more efficient than those.

4.1. SNL* = SNL − IRM + MiCRoM + Omni

It has been recently shown by Stehling et al. [7] that the IRM distance measure is not a metric, i.e. in particular it does not enforce triangle inequality.[5] In the context of metric access structures (e.g. [5]) and filtering techniques (e.g. [6]), the triangle inequality property can used to prune the search space and thereby reduce the number of complex distance calculations. This property guarantees that during space reduction, the pruning process will not filter out any of the relevant images.

In the SNL* technique we replace the IRM (non-metric) distance by the MiCRoM distance proposed by Stehling et al. [7]. In MiCRoM, the distance between two images A and B is modeled as a network flow transportation problem [32] as follows. Each region A_i in image A is a producer, capable of producing $P(A_i)$ colored pixels. Conversely, each region B_j in B is a consumer able to consume $C(B_j)$ colored pixels. Every pair of regions is connected by an edge which can transport a number of pixels limited by the size of the smallest of the regions, at a cost given by the distance between those. This distance is the same distance $D^{SNL}(i,j)$ defined in Section 3.4 which is a metric. It has been shown in [7] that the solution of the network flow problem is not only an optimal version of the IRM distance computation, but it is also a metric distance.

Recall that in SNL a linear scan had to be performed during query processing, i.e. the query image was compared to all other images in the database. The advantage of equipping SNL with the MiCRoM metric distance is that a filter (e.g. the Omni technique detailed next) can be used to prune the search space and reduce the number of image comparisons done.

To explore the filtering issue, and further reduce query processing time, we adopted the Omni approach [6]. The overall goal for the filter is to reduce the number of distance calculations required to answer similarity queries. This technique assumes that the distance function used to calculate the similarity between two images is a metric, e.g. MiCRoM.

In the Omni approach, a set of global representative points are initially chosen from the database. These representative points are static in nature, i.e. they are only selected once. These set of points are called the foci and the set of foci of a database forms the Omni-foci base. Each object (image representation) is mapped on to a lower dimension space and in this space they are represented by the Omni-coordinates. Coordinate values of objects in the Omni-coordinate system are actually the original distance between objects and the foci. Whenever a new object is inserted, the Omni-coordinates of this object are calculated and stored. While querying, the Omni-coordinates of the query image are calculated, i.e. the query image is also mapped onto a lower dimension space. In this space, the triangle inequality property is extensively used to prune many distance calculations. Fig. 9 illustrates the Omni approach with a single focus point f, q being the query image and r_q being the query radius. Each focus point defines a metric sub-space ring called the mbOr as indicated by the area between the two rings in Fig. 9. An mbOr includes

[5] Given a set of objects o_i, o_j and o_k, $d(o_i,o_j) \leqslant d(o_i,o_k) + d(o_k,o_j)$.

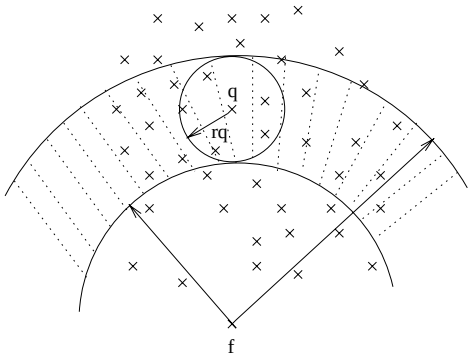

Fig. 9. Pruning using a single focus (adapted from [6]).

all the objects that the Omni-coordinates identify as part of the answer set. All points outside the ring are filtered. Points inside the ring cannot be pruned by the focus and hence for this subset of images, the original distance needs to be calculated. Nevertheless, the cardinality of such subset is (usually) much smaller then the whole dataset.

A crucial point in the Omni technique is the choice of the foci objects. The authors have proposed the so-called HF-algorithm [6] to select the foci, where the general goal is to always chose a focus that minimize the notion of cumulative error, i.e. the sum of the distance between the candidate focus and the set of current foci.

The logic behind the HF-algorithm is illustrated in Fig. 10. The objects are represented as circles and the distance between these objects is the difference between their colors in the RGB space. In this algorithm, a random object, say A, is chosen initially. The object farthest to this random object A, becomes the first focus f_1 and the object farthest to the first focus f_1 becomes the second focus f_2 as shown in Fig. 10. Then the next step is to calculate the edge which is the original distance between objects f_1 and f_2 as shown in Fig. 10. Using f_1, f_2 and the edge, the next focus is calculated by computing the cumulative error. This makes A the next focus.

However, the main idea behind the choice of foci points is that they should be as far apart as possible, so that each one significantly contributes in the pruning process. In this example, object B is a better candidate for a focus than object A. Hence, we modified the HF algorithm, calling it HF$'$, to find foci that are spread far apart in the object space.

In the HF$'$ algorithm, the first two foci are determined exactly as in the HF algorithm. For all other foci, instead of calculating the cumulative difference in the distance between each candidate object and the set of foci found so far, the minimum distance between a candidate object and the set of foci found so far, is calculated. Then, the maximum of the minimum distances is found and the corresponding candidate object becomes the next focus. In the illustration shown in Fig. 10, objects f_1 and f_2 are the first two foci. For the third focus point, there are two candidate objects namely A and B. The minimum of the distances between A and the two foci f_1 and f_2 is 0.261. For object B, the minimum distance is 0.331. Since the distance from

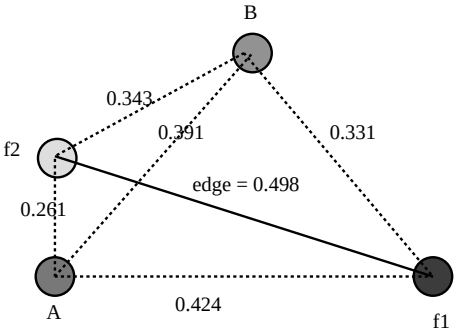

Fig. 10. Illustration for HF and HF′ algorithms.

object B is greater, it becomes the next focus point. Thus, HF′ finds foci that are spread further apart in space.

While the HF′ algorithm does initially find foci spread farther apart, the main difference between the original and the improved algorithm after a few iterations was the order of the foci images obtained. In practical terms, the qualitative difference between Omni's performance was nearly not noticeable.

The authors of [6] have shown that the Omni approach can be used with a sequential scan algorithm, or more elaborated access structures, e.g. R-trees [33]. Since our aim was to investigate the effect of incorporating the Omni technique into SNL using the MiCRoM distance (metric). To execute a range query with radius r_q using this algorithm, first of all the original distance between the query object o_q and each foci $f_k \in F$, $df_k(o_q)$ is calculated, thereby creating Omni-coordinates for the query object. Then, for each object o_j in the database, if $|df_k(o_j) - df_k(o_q)| > r_q$, for each focus $f_k \in F$ then the original distance calculation between objects o_q and o_j is skipped. Otherwise, the original distance is computed to check if the object o_j lies within the radius r_q of the query object o_q.

4.2. *Experiments with* SNL*

This section presents the experiments that were conducted to effectively adopt the Omni approach with the MiCRoM distance in the SNL* technique. The database images and the 15 query images that were used for experiments in this section are described in Section 3. Experiments were conducted to select a suitable number of foci and also to decide on the query radius of our range queries.

In order to select the foci, the HF′ algorithm is used. The number of foci is critical because for every focus, there is an extra dimension added which involves some computation. Unless this additional focus is good enough to filter a fairly large fraction of images and thereby save us some query processing time, the overhead of the space and computation time is not worth it. Hence it is very important to choose a good number of focus points. An experiment was performed wherein the number

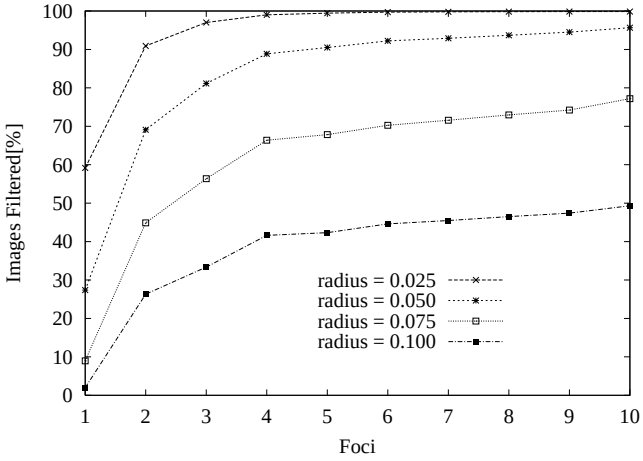

Fig. 11. Variation of images filtered with foci (using 10,000 images).

of foci was varied from 1 to 10 and the percentage of images filtered was noted down. In Fig. 11, we see that there is a sharp rise in the percentage of images filtered as the number of foci is varied from 1 to 4 and from then on the curve does not show too much variation. In addition, we have observed that while the HF′ algorithm seems to be more effective than the original HF algorithm for a small number of foci (up to 6), both tend to perform equally well for larger numbers of foci. We also observed that when the radius is 0.025 and the number of foci was, say 10, only about 2% of the database is read, whereas for the same number of foci when the radius is 0.1, about 50% of the database is read. Thus, the number of images retrieved can be somehow controlled by the query radius.

The aim of proposing the SNL* technique is to reduce the query processing time. The number of foci is a factor that directly affects the query processing time. Hence before selecting the number of foci, the behavior of the query processing time with respect to change in the number of foci was noted as shown in Fig. 12. In this graph, we see that there is a decrease in query processing time when the number of foci is between 1 and 4 and then steadies down between 4 and 10. Hence, from the above two graphs, it seems reasonable to select the number of foci to be four.

The query radius is also another factor that affects the processing time. There is a trade off between the number of relevant images retrieved and the total number of images retrieved (the larger the number of images retrieved, the greater the processing time). In Fig. 13, we can see that when the query radius is 0.1, all the relevant images are retrieved. But to achieve this, we need to process about 60% of the database. Whereas for a query radius of 0.075, about 96% of the relevant images are retrieved and only 35% of the database needs to be processed. Therefore, a radius of 0.075 was selected for our range query, since it was sufficient to retrieve almost all the relevant images corresponding to the query images. This is of course tunable, depending on the percentage of relevant images one is willing to, potentially, not retrieve.

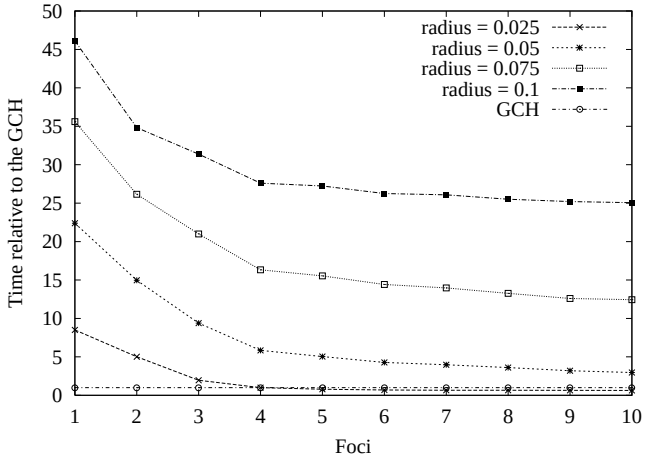

Fig. 12. Variation of query processing time with foci.

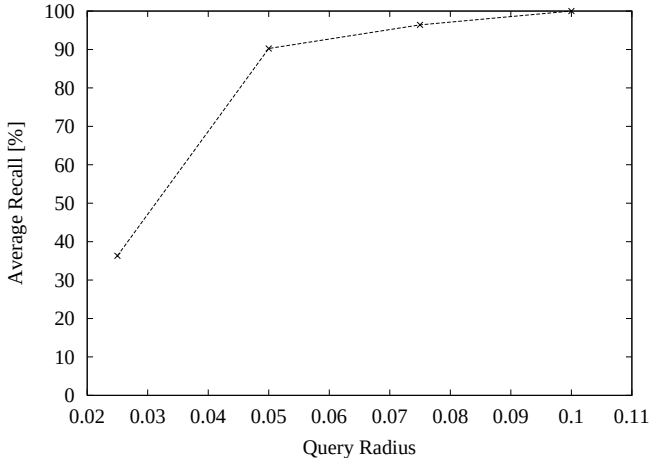

Fig. 13. Variation of recall with query radius.

Summarizing, we learned that using SNL* with query radius of 0.075 and an Omni-foci base of six images one could retrieve over 90% of the relevant images by calculating the original (expensive) distance on less than 40% of the dataset.

As one can clearly see from the discussion above, an in particular from Fig. 12, the time needed to process a query using SNL* is much larger than our baseline (the fast, but not effective, GCH). This is due the computational cost of solving the network problem inherent to the MiCRoM distance. One question that arises at this point is the following: how important is it for the image distance measure to be a true metric distance? We address this next.

4.3. $SNL^+ = SNL^* - MiCRoM + IRM = SNL + Omni$

One, e.g. [11], can argue that it is just yet another approximation step brought into a process where there are several approximations such as:

- The visual features that are used to represent and compare images is an approximation of the visual content of the images;
- The distance measure devised to calculate the similarity between images is an approximation of the human perception of similarity;
- The weights assigned to the features extracted are also an approximation of what would be perceived as most important;
- The retrieval threshold that is used in the query processing phase is an approximate estimation of the similarity between relevant images.

Therefore, it might be acceptable to lose a small number of relevant images in exchange for a much faster query processing. The IRM distance which is used to measure the similarity between images is a very good heuristic to approximate the MiCRoM distance. Thus, in SNL^+, we decided to investigate the effect of using a non-metric distance (namely the original IRM using in SNL) with the Omni approach. The trade-off is that since IRM is not a metric distance, there is no guarantee that relevant images are not left out of the answer set. The obtained results regarding this and other issues are discussed next.

4.4. *Experiments with* SNL^+

In order to select the foci, the HF′ algorithm is used. An experiment similar to the one for SNL^* was performed wherein the number of foci was varied from 1 to 10 and the percentage of images filtered was noted down. In Fig. 14, we see that there is a sharp rise in the percentage of images filtered as the number of foci is varied from 1 to 6 and from then on the curve does not show too much variation. The change in query processing time with respect to the number of focus points was plotted as shown in Fig. 15. In this graph, we see that there is a decrease in query processing time when the number of foci is between 1 and 6. The time steadies down between 6 and 7 and then starts increasing gradually. This happens because the increase in the size of the Omni-foci base is not able to further prune the dataset, and at the same time, the increase in the number of Omni-coordinates makes the distance computation (in the Omni-space) more expensive. From these two graphs, the foci cardinality was chosen to be 6.

The percentage of relevant images retrieved was plotted against various radii to select the query radius of the range queries. From Fig. 16, a query radius of 0.075 was selected as the best compromise (using an argument similar to the one for SNL^*).

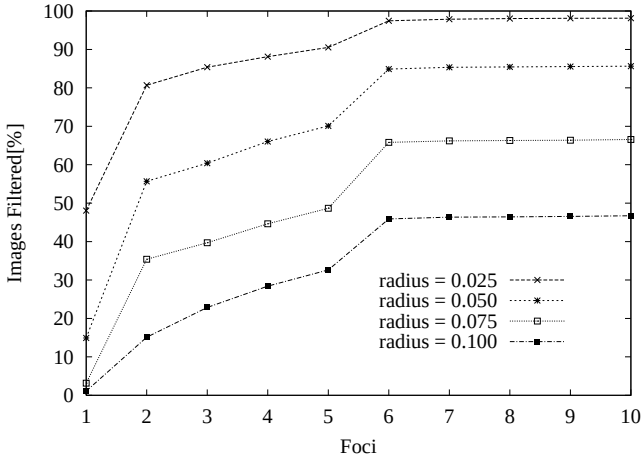

Fig. 14. Variation of images filtered with foci.

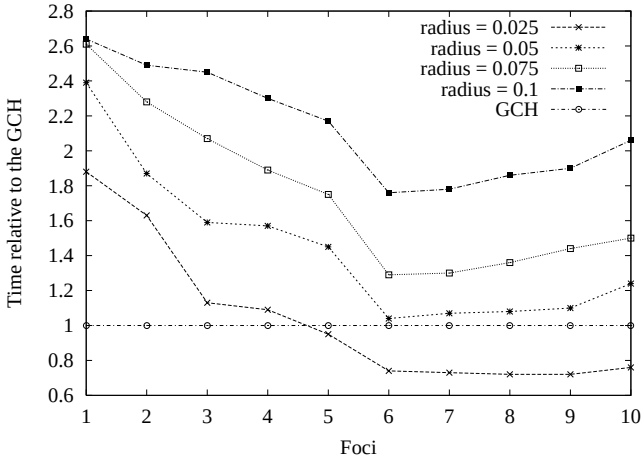

Fig. 15. Variation of query processing time with foci.

4.5. SNL *vs.* SNL* *vs.* SNL$^+$

As discussed earlier, the IRM measure used to calculate the similarity between images is not a metric and despite this fact, SNL$^+$ uses it with the Omni-sequential approach, which by default works on metric distances. Using non-metric distances makes the SNL$^+$ approach liable to losing some of the relevant images. In order to determine the amount of relevant images lost, the precision and recall values for 10,000, and 50,000 images using all three variations of the SNL technique are measured. The number of foci was fixed as 6 (4 for SNL*) and a radius of 0.075 was

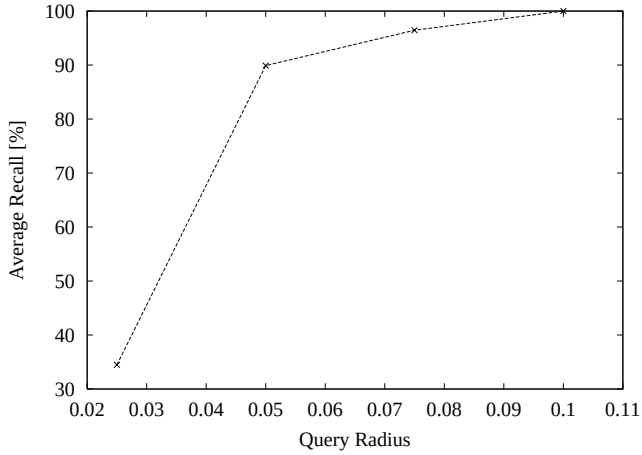

Fig. 16. Variation of recall with query radius.

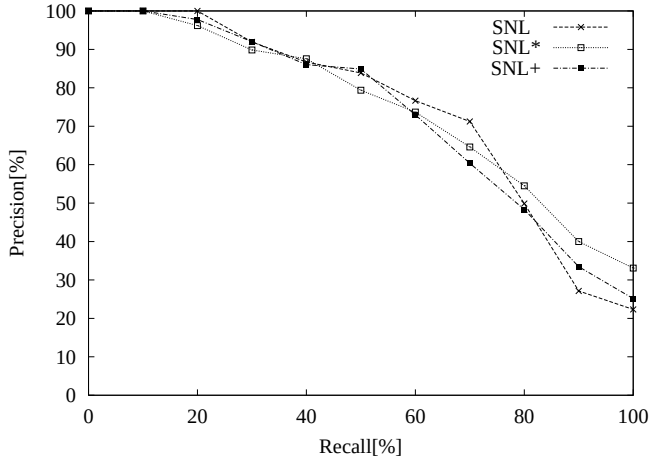

Fig. 17. Comparing SNL, SNL* and SNL$^+$ with 10,000 images.

used. The results of this experiment are shown in Figs. 17 and 18. The graphs show that the curves are very close to each other indicating that the loss of relevant images that occurs by approximating the Omni-sequential algorithm with a non-metric distance is acceptable. At certain retrieval points, the SNL$^+$ and SNL* have a higher precision compared to the SNL technique. This is due to the fact that in SNL* and SNL$^+$ the original distance is calculated only for a very small fraction of the database thus further decreasing the room for false positives.

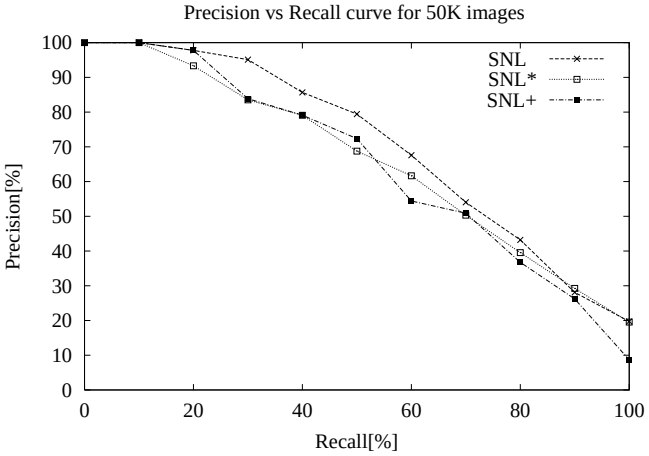

Fig. 18. Comparing SNL, SNL* and SNL$^+$ with 50,000 images.

Table 4
Query processing time relative to GCH for SNL, SNL* and SNL$^+$

Techniques	10K	20K	50K
SNL$^+$	1.28	1.36	1.45
SNL*	16.33	20.13	29.95
SNL	2.55	2.72	3.05

Finally, Table 4, compares the query processing time required by the three techniques namely SNL, SNL* and SNL$^+$ relative to the time needed by using GCH. The actual query processing time of the SNL$^+$ technique in a database of 10K images using six foci and 0.075 as the query radius is only about 6 s. It can be seen that the processing time has reduced by almost 50% using the SNL$^+$ technique instead of the original SNL (which required a linear scan over the dataset) and one order of magnitude when compared to the SNL*. Even when the database size is changed from 10K to 50K, the query processing time for SNL$^+$ is still small and in fact comparable with GCH's processing time. The query processing time for SNL* that uses the metric distance is about 8 times that of SNL indicating that finding the optimum solution to the matching problem is a time consuming process and hardly feasible. A greedy approach IRM which approximates the MiCRoM distance works just as well and takes much less time. Using the Omni approach has helped to reduce the query processing time.

5. Conclusions and future work

This paper revisited a region-based CBIR technique, SNL [2], and improved on it, addressing two important issues: effectiveness of its distance measure and retrieval

efficiency. SNL's original distance measure, obtained using the IRM [4], is not a metric distance, thus preventing one of using techniques, e.g. metric access structures, to speedup query processing.

The distance measure issue was addressed by using the MiCRoM metric distance proposed in [7]. While on the one hand this allowed the use of a filtering technique [6], reducing the number of image comparisons in the original metric space, on the other hand MiCRoM itself was too expensive to compute, hence resulting in a non-practical solution. In order to make query processing more efficient we experimented using the original IRM distance, even though it was not a metric distance, along with the Omni approached. The potential loss in query effectiveness was minimal, and the query processing time was improved greatly. In fact, when compared with the traditional GCH, we were able to achieve a retrieval up to twice as much more effective (in terms of precision and recall) at the expense of less than 50% more query processing time.

Some of the future directions for exploring the SNL technique are to improve the shape and position representations of the regions, to improve color representation with non-uniform quantization, and to explore other distance measures. On the efficiency and scalability issues, other Omni-based techniques, such as the Omni-R-tree could be applied as well. As well, in order to optimize the histogram storage space, one could investigate the use of the binary signatures proposed in [30]. Finally, a detailed study comparing our approach with that using MPEG-7 low-level image property descriptors on entire images and on individual regions should also be conducted when more standard image databases become available.

Acknowledgements

This work was supported in part by the Canadian Natural Sciences and Engineering Research Council.

References

[1] T. Huang, Y. Rui, Image retrieval: Past, present, and future, in: Proceedings of the International Symposium on Multimedia Information Processing, 1997, pp. 1–23.

[2] V. Sridhar, M.A. Nascimento, X. Li, Region-based image retrieval using multiple-features, in: Proceedings of the 2002 Visual Information Systems Conference (VISUAL'02), 2002, pp. 61–75.

[3] R.O. Stehling, M.A. Nascimento, A.X. Falcão, An adaptive and efficient clustering-based approach for content based image retrieval in image databases, in: Proceedings of the International Data Engineering and Application Symposium, 2001, pp. 356–365.

[4] J. Li, J.Z. Wang, G. Wiederhold, IRM: integrated region matching for image retrieval, in: Proceedings of the Eighth ACM Multimedia Conference, 2000, pp. 147–156.

[5] P. Ciaccia, M. Patella, P. Zezula, M-tree: an efficient access method for similarity search in metric spaces, in: Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB'97), 1997, pp. 426–435.

[6] R.F. Santos-Filho, A. Traina, C. Traina Jr., C. Faloutsos, Similarity search without tears: the omni family of all-purpose access methods, in: Proceedings of the 17th International Conference on Data Engineering (ICDE 2001), 2001, pp. 623–630.
[7] R.O. Stehling, M.A. Nascimento, A.X. Falcão, MiCRoM: a metric distance to compare segmented images, in: Proceedings of the 2002 Visual Information Systems Conference (VISUAL'02), 2002, pp. 12–23.
[8] M. Swain, D. Ballard, Color indexing, International Journal of Computer Vision 7 (1) (1991) 11–32.
[9] M.A. Stricker, M. Orengo, Similarity of color images, in: Proceedings of the Storage and Retrieval for Image and Video Databases (SPIE)-III, Vol. 2420, 1995, pp. 381–392.
[10] G. Pass, R. Zabih, J. Miller, Comparing images using color coherence vectors, in: Proceedings of the Fourth ACM Multimedia International Conference, 1996, pp. 65–73.
[11] R.O. Stehling, M.A. Nascimento, A.X. Falcão, Techniques for color-based image retrieval, in: C. Djeraba (Ed.), Multimedia Mining—A Highway to Intelligent Multimedia Documents, Kluwer Academic, Dordrecht, 2002 (Chapter 4).
[12] R.O. Stehling, M.A. Nascimento, A.X. Falcão, Cell histograms versus color histograms for image representation and retrieval, Knowledge and Information Systems (KAIS) Journal, 2003, to appear.
[13] A.D. Bimbo, Visual Information Retrieval, Morgan Kaufmann, Los Altos, CA, 1999.
[14] G. Lu, Multimedia Database Management Systems, Artech House, Norwoord, MA, 1999.
[15] Y. Rui, A. She, T. Huang, Modified Fourier descriptors for shape representation—a practical approach, in: Proceedings of the First International Workshop on Image Databases and Multimedia Search 1996, pp. 22–23.
[16] R.C. Gonzalez, R.E. Woods, Digital Image Processing, 3rd Edition, Addison-Wesley, Reading, MA, 1992.
[17] M.K. Hu, Visual pattern recognition by moment invariants, IRE Transactions on Information Theory IT-8 (1962) 179–187.
[18] L. Yang, F. Albregtsen, Fast computation of invariant geometric moments: a new method giving correct results, in: Proceedings of the 12th International Conference on Pattern Recognition, 1994, pp. 201–204.
[19] D. Kapur, T. Saxena, Y.N. Lakshman, Computing invariants using elimination methods, in: Proceedings of the IEEE International Symposium on Computer Vision, 1995, pp. 97–102.
[20] R. Haralick, K. Shanmugam, I. Dinstein, Texture feature for image classification, IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (6) (1973) 610–621.
[21] H. Tamura, S. Mori, T. Yamawaki, Texture features corresponding to visual perception, IEEE Transactions on Systems, Man, and Cybernetics SMC-8 (6) (1978) 460–473.
[22] B.S. Manjunath, et al., Color and texture descriptors, IEEE Transactions on Circuits and Systems for Video Technology 11 (6) (2001) 703–715.
[23] M. Bober, MPEG-7 visual shape descriptors, IEEE Transactions on Circuits and Systems for Video Technology 11 (6) (2001) 716–719.
[24] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, J. Malik, Blobworld: a system for region-based image indexing and retrieval, in: Proceedings of the Third International Conference on Visual Information Systems, 1999, pp. 509–516.
[25] W.Y. Ma, B.S. Manjunath, Netra: a toolbox for navigating large image databases, Multimedia Systems 7 (3) (1999) 184–198.
[26] J.Z. Wang, J. Li, G. Wiederhold, Simplicity: semantics-sensitive integrated matching for picture libraries, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (9) (2001) 947–963.
[27] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, Los Altos, CA, 2000.
[28] C.J. Van Rijsbergen, Information Retrieval, 2nd Edition, Butterworths, London, 1979.
[29] V. Sridhar, Region-based image retrieval using multiple-features, Master's thesis, University of Alberta, 2002, ftp://ftp.cs.ualberta.ca/pub/TechReports/2002/TR02-10/TR02-10.ps.gz.
[30] M.A. Nascimento, V. Chitkara, Color-based image retrieval using binary signatures, in: Proceedings of the 2002 ACM Symposium on Applied Computing, 2002, pp. 687–692.

[31] I. Witten, A. Moffat, T. Bell, Managing Gigabytes, 2nd Edition, Morgan Kaufmann, Los Altos, CA, 1999.
[32] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[33] N. Beckmann, H.P. Kriegel, R. Schneider, B. Seeger, The R*-tree: an efficient and robust access method for points and rectangles, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 1990, pp. 323–331.