Adaptive Color Histogram Indexing

Vincent Ng[1], David Cheung[2], Ada Fu[3]
[1]Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong
[2]Department of Computer Science, Hong Kong University, Hong Kong
[3]Department of Computer Science, Hong Kong Chinese University, New Territories, Hong Kong

ABSTRACT

In this paper, we develop an indexing scheme for medical images. In general, for a given medical image, there are objects which are clinically important amongst the rest. We name the objects as the *dominant* objects. Our proposed index is composed of three parts: (1) dominant objects in images are located; (2) each image will have an associated R-tree which is constructed by its dominant objects; and (3) an R-tree that clusters similar images together. To demonstrate the effectiveness of the index developed, we use images of skin lesions as the image data. Our initial experiments give promising results for image retrieval.

Keywords: Color histograms, R-trees, Image databases

1 Introduction

Traditional database management systems are unable to provide efficient retrieval for image data. In most systems, images are either unstructured or structured by using keys such as title, author or date. Recently, content-based or feature-based methods have been proposed to provide efficient retrieval of images. Image access is done through a high-level abstraction of the image properties. Example queries are "Retrieve all red circular lesions on a tanned skin" or "Find all mammograms which has a bright cluster".

In the past, many indexing methods are either based on the images directly or the features of objects within images. In 1991, Kato has developed a system to retrieve images based on sample images or sketches[1]; while Niblack[2] proposed to use the R*-tree for color indices. Recently, Gong[3] suggests to use B^+-tree in which each key is made up of different image features. Of the many methods, color histograms have shown to be an effective way to compare color images.[4] There are currently two approaches to utilize the color histograms besides the choices of colors used in histogram computation. The first approach uses a single histogram for the entire image.[4] It suffers from a drawback where the positional information of color is lost. A better approach, reported by Lu, uses a quadtree structure to decompose the image recursively into quadrants and each node is associated with a histogram.[5] However, it has no concept of interesting objects in a given image.

In our work, we are interested to develop an indexing scheme for medical images. In many medical images, there are usually a background and a number of interesting or *dominant* objects. The background is relatively uniform in colors, photographing or digitizing positions are standard and each dominant object would have its own set of major colors. For example, in an image of a pigmented lesion, the dominant object is a mole (see Figure 1; while in a MRI (Magnetic Resonance Image) , there are dominant objects such as the liver and the spine. Hence, to facilitate image queries, a global description of color composition of an image is not appropriated.

 0-8194-1970-2/95/$6.00

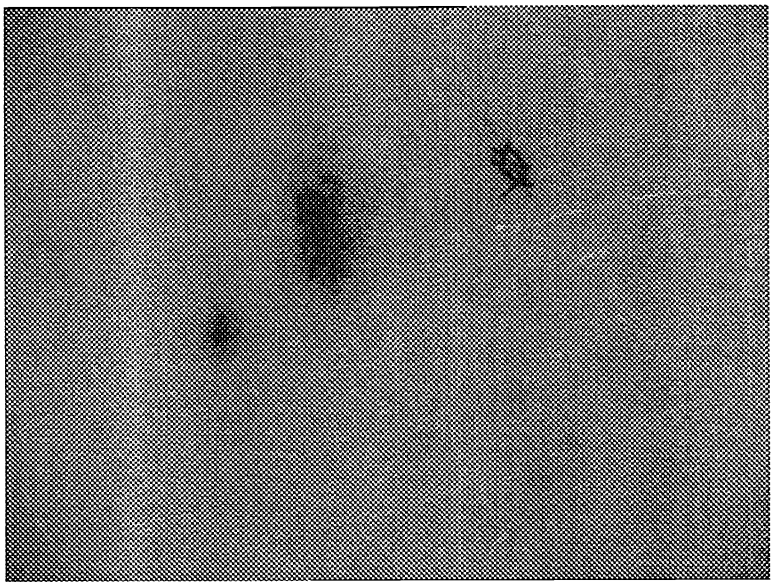

Figure 1: An image with two skin lesions.

It is more interesting and important to describe the color compositions of dominant objects. In this paper, we suggest a color index which is composed of three parts:

- Locating dominant objects in images;
- Associating each image with an R-tree which contains the color composition information of its dominant objects;
- Constructing an R-tree that clusters similar images together.

The rest of the paper is organized according to the three parts above. We will first discuss the method of finding dominant objects in a given image in Section 2, where we will also describe the use of the HSI color scheme. In Section 3, we present how to create the R-image trees and to use them to compute the similarities between images. Section 4 discusses the construction of the R-tree to group images together, and Section 5 presents results of our preliminary experiments.

2 Dominant Objects

In general, when pictures are digitized into images, colors of the pixels are coded in terms of the RGB (Red, Green, Blue) values. However, it has been known that the R, G, B values are correlated and depended on the light intensity changes. A better way to perform color analysis is to use the HSI scheme where the hue, H, the saturation, S and the intensity, I can be obtained with the the following equations[6]:

$$I = R + G + B \tag{1}$$

$$S = 1 - 3min(R, G, B)/I \tag{2}$$

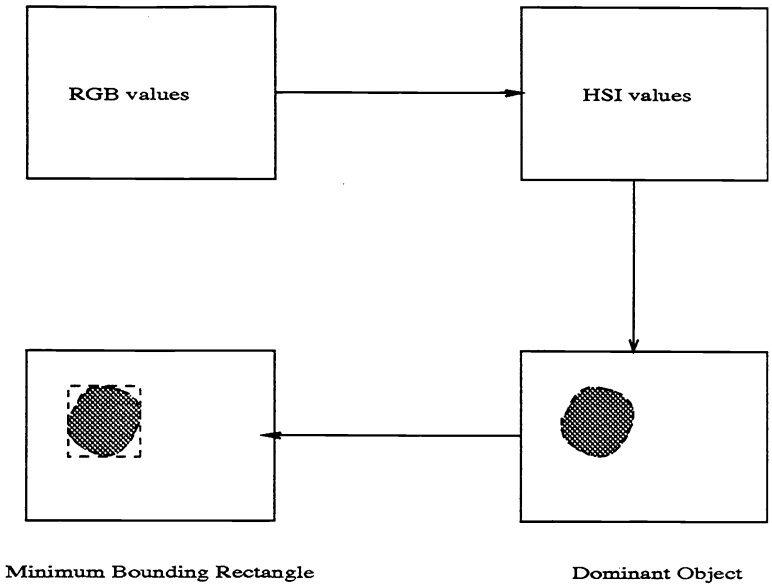

Figure 2: Locating dominant objects.

$$H_w = cos^{-1} \frac{0.5((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-B)(G-B)}} \tag{3}$$

$$H = \begin{cases} H_w & \text{if } B \leq G \\ 2\pi - H_w & \text{if } B > G \end{cases} \tag{4}$$

The saturation measures the amount of white inside a considered color and has a value between 0% to 100%. The hue has a value from 0 to 360 degrees. It is the red color at 0 degree, or the green color at 120 degree, or the blue color at 240 degree. If discrete values of hue and saturation are used, even then, there can be up to 36,000 distinct color indexes at one intensity level. In order to reduce the number of color indexes into a manageable set, we group adjacent hue values, saturations and intensities together. We divide the HSI color space into 36 groups of hue $(H_1, \ldots, H_{36})$, 10 groups of saturation $(S_1, \ldots, S_{10})$ and 5 groups of intensity levels $(I_1, \ldots, I_5)$. In total, this represents 360 possible color indexes at one intensity level.

We plan to experiment with the images of skin lesions from a clinical study.[7] In that study, we have proposed a multistage algorithm to segment the images. However, when we start working on this project, we observe that the segmentation results are not critical for comparing different images. The comparison should be based on the color content but not on the exact boundary information of the dominant objects. Furthermore, a dominant object is usually composed of similar colors. Therefore, we will only need a simple method to find out regions that contain dominant objects. The idea is to join neighborhood pixels of the same or similar color together.

As showed in Figure 2, each image is processed in two major phases before its associated R-tree is constructed. In the first phase, the RGB values of the image are first mapped to the HSI values with Equations (1)-(4). In the second phase, each pixel is masked by a 5x5 grid and it is labeled as color k if there are sufficient (say more than half) pixels of the k^{th}-color in the grid. Adjacent or close-by regions of the same color index are merged together to form larger regions. Color indexes are then further grouped until the sizes of all regions are larger than a preset threshold value. Among all the regions, the largest region is considered as the background of the image and the remaining regions become dominant objects. For each dominant object, we calculate its enclosing minimum bounding rectangle. The color histogram of the corresponding rectangular area is then found and it

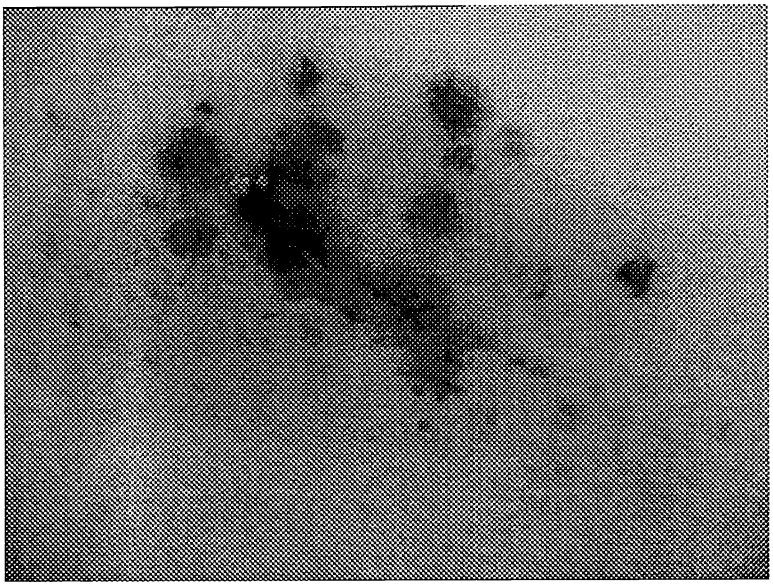

Figure 3: An image with multiple skin lesions.

records the number of pixels of each color inside the rectangle. To illustrate the steps, in Figure 3, we show an image with multiple skin lesions. The image is then processed to find out its dominant objects (see Figure 4) and their minimum bounding rectangles (see Figure 5 for the bounding rectangle of the largest dominant object in the image).

3 R-image Trees

Once we have identified the dominant objects of the images, the second part is to develop a similarity measurement between them. We adopt the use of color histograms which are effective in characterizing the color of an image.[4,2] A color histogram of an image represents the distribution of different discrete colors appeared in the image. In our work, we take advantage of the spatial information of the dominant objects to improve query performance.

An R-tree[8] is created by inserting the bounding rectangles of the dominant objects. We named the tree as the **R-image** tree. At each internal node of an R-image tree, there is an associated color histogram. As shown in Figure 6, we join two bounding rectangles (R, R') of two leaf nodes together to form a new internal node with a bounding rectangle equals to A. Inside A, there are four possible sub-regions: the overlapping region (M), the distinct regions of R and R' (R_d and R'_d correspondingly), and the background region (BG). In order to find out the color histogram associated with A, a simple but inefficient method is to re-compute the color histogram of the region A from scratch. To reduce the computational effort, we propose to use a heuristic method to find out A's color histogram by utilizing the color histograms of R and R'. Suppose A_i, R_i, R'_i and BG_i are the number of pixels of the i^{th}-color inside region A, rectangles R and R', and the background. We can then approximate A_i as

$$A_i = R_i + R'_i + BG_i \tag{5}$$

This method can be easily extended by simply adding up the corresponding color information when there are more than two overlapping rectangles in the region A. With this method, at each internal node of the R-image tree, we would only need to find out the color distribution of the pixels in the background enclosed by the bounding

Figure 4: Dominant objects found in the image with multiple skin lesions.

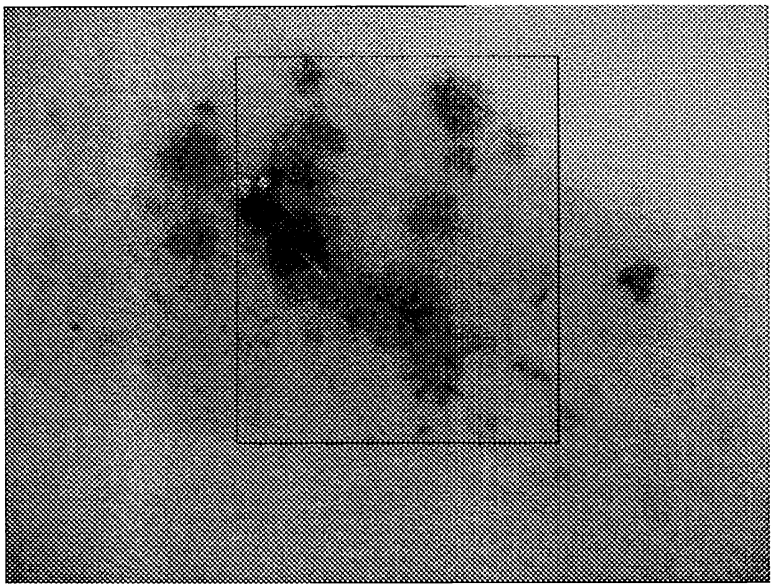

Figure 5: The bounding rectangle of a dominant object in the image with multiple skin lesions.

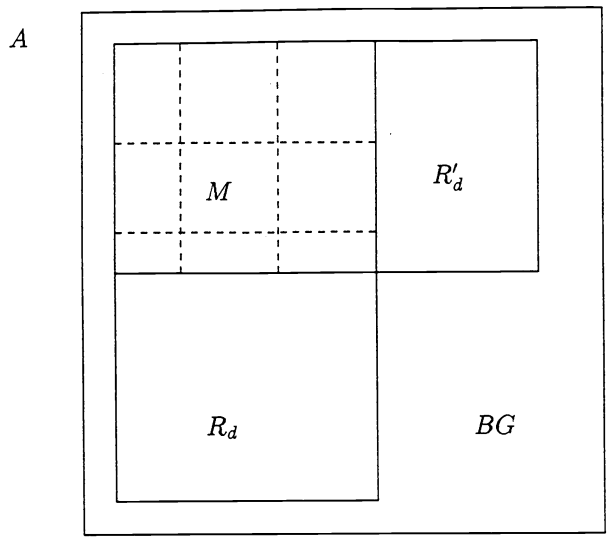

Figure 6: Updating the color histogram associated with an internal node.

rectangle of the internal node.

It has been known that the construction of an R-tree depends on the insertion order of the objects.[8] Therefore, dominant objects of an image are sorted with respect to their sizes and positions before inserting into its corresponding R-image tree. By using the heuristic method above, at each node of the R-image tree, we update its associated color histogram level-by-by up to the root node during insertions. However, the bounding rectangle of the root node may not cover the complete image. A separated color histogram is therefore computed for the complete image. We call the color histogram at the root node as the *root histogram* and the other one as the *complete histogram.*

Suppose we have two bounding rectangles R and R' from two different images. If R equals R', the similarity between them can be obtained directly by comparing their corresponding color histograms. If R and R' do not overlap, the two images cannot be the same. However, if R and R' overlap as showed in Figure 6, we would need a similarity measurement which will not reject positive matches. The difference method proposed by Swain will fail because of the unknown color distribution in the region M.[4] We observe that if there are m pixels in the region M and the overlapping area from the two rectangles are of the same color distribution, then the sum of the minimum number of pixel for each matched color, $SA_{RR'}$, cannot be less than m. That is

$$SA_{RR'} = \sum_i min(R_i, R'_i) \geq m \tag{6}$$

where R_i, R'_i are the number of pixels of the i^{th}-color in the color histogram of R and R' correspondingly. Therefore the similarity between the two rectangles can be defined as

$$S_{RR'} = \begin{cases} m/U & \text{if } SA_{RR'} \geq m; \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

where U is the union of the areas of R and R'. Note that the heuristic method defined in Equation 5 does not violate the above similarity measurement. It would only increase the value of $SA_{RR'}$ without affecting the value of $S_{RR'}$ at all.

We are now ready to describe how to measure the similarity, S_{XY}, between two given images, say X and Y. Initially, we compute the similarity by using the root histograms of the R-image trees from X and Y. A tolerance,

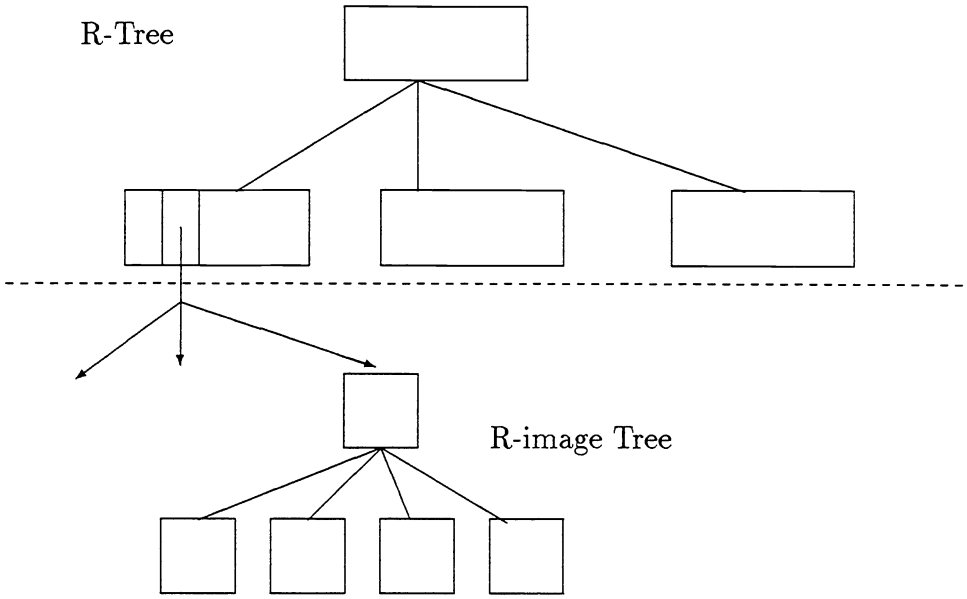

Figure 7: A two-layer index.

ϵ_1, is used to allow a smaller value, $m(1-\epsilon_1)$, at the right-hand-side of Equation 6. If S_{XY} equals to 0, then X and Y are two different images; otherwise the tree nodes in the second level of the corresponding R-image trees of X and Y are utilized. Suppose $X_1, \ldots, X_k$ are the bounding rectangles of the nodes in the second level of the X's R-image tree, and $Y_1, \ldots, Y_l$ for the Y's R-image tree correspondingly. We can then define that X and Y are similar if

$$S_{XY} = \sum_i \sum_j S_{X_i,Y_j} \geq (1-\epsilon_2) \tag{8}$$

where U in Equation 7 becomes the union of areas of $X_1, \ldots, X_k$ and $Y_1, \ldots, Y_l$ and ϵ_2 is another factor of tolerance. If S_{XY} satisfies the inequality in Equation 8, it is repeated for lower levels in the R-image trees. When the computation reaches the leaf nodes of the R-image trees and the inequality is satisfied, the two images are said to be similar within the tolerance ϵ_1 and ϵ_2.

4 A Two-Layer Index

We use a two-level index as shown in Figure 7 for the image database. Layered on the top of the R-image trees, we use an R-tree to index the color histograms at the root nodes of the R-image trees. During the construction of the R-tree, we realize a dimensionality problem. In our image database of lesions, there can be 360 bins in a color histogram at one level of intensity. Indexing on such a high-dimensionality is computational intensive and even infeasible. To solve the problem, one approach is to use a few dominant colors instead of using all colors.[5] For many images, this may not be possible since a small number of colors may not occupy a sufficient portion of an image. A second approach is to use average color indices.[9] However, its formation does not depend on the data and can result with poor matching performance.

As the updating of the image database is infrequent, we decide to build an R-tree by considering the colors used in the root histograms of the R-image trees. The idea is to merge colors into color zones such that each color zone will have a similar usage frequency. Suppose there are n discrete colors, $C_1, C_2, \ldots, C_n$ amongst all

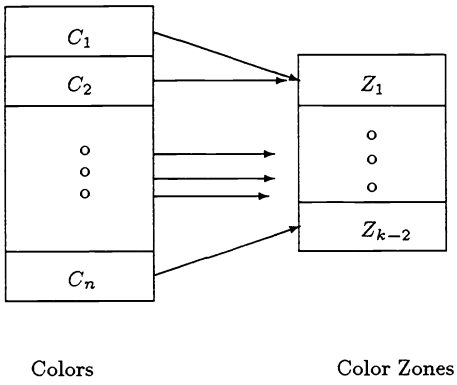

Figure 8: Mapping color indexes into color zones.

the histograms and a color, C_i, can be defined by its hue, saturation and intensity components as (H_i, S_i, I_i). The usage of C_i, f_i, can be approximated by the number of non-empty bins of C_i in the histograms. A large value of f_i suggests that C_i is being used more frequently. After obtaining all the f_i's, we sort the n colors with respect to their (H, S, I) values. Accumulative values of f_i's are then computed along with the sorted list and are represented as a_i's. If k-dimensional hyper-rectangles are to be used in the R-tree, the a_i's are used to divide the colors into $k-2$ zones and a conversion table is built as showed in Figure 8. The value of each color zone is obtained by summing up the values of the corresponding colors in a color histogram.

Each R-image tree is represented as an entry in a leaf node of the R-tree. The entry is a k-dimensional hyper-rectangle whose first $k-2$ dimensions are for the color zones and the last 2 dimensions are for the bounding rectangle at the root node of the corresponding R-image tree. In the hyper-rectangle, the i^{th}-dimension has the range $[Z_{ir}, Z_{ic}]$ where Z_{ir} is the value coming from the root histogram and Z_{ic} is from the complete histogram and $i \leq k-2$. The ranges for the last 2 dimensions are $[x_1, x_2], [y_1, y_2]$ where x_1, x_2, y_1, y_2 are the lower and upper values of the corresponding x- and y-dimensions of the bounding rectangle of the root node at the R-image tree. After all the hyper-rectangles are generated, they are then inserted to form the new R-tree.

To perform an image query, a user provides a query image and a tolerance value. The query image is processed to locate its dominant objects, and its associated R-image tree will be calculated. Together with the tolerance value and the color histograms at the root node of the R-image tree, it forms a search window for the R-tree. The search results are the root nodes of R-image trees in the database. Similarities between the R-image trees and that of the query image are calculated as discussed in the previous section. For those values within the tolerance, their corresponding images are returned to the user.

5 Experiments

To evaluate the proposed method, we have tested it with images of skin lesions. The images are collected from a clinical study in Vancouver, Canada.[7] At the time of writing, there have been 219 images digitized. Each image has a background area and at least one lesion. To simplify our work, we have excluded images containing hairs. This has left us with 185 images where 20 of them have multiple lesions and 165 of them contain only one lesion. From each original image, we created two additional ones. One is created by randomly re-distributing 5% of the pixels in the image and the other one is done similarly by re-distributing 20% of the pixels. Therefore, the total number of images becomes 555.

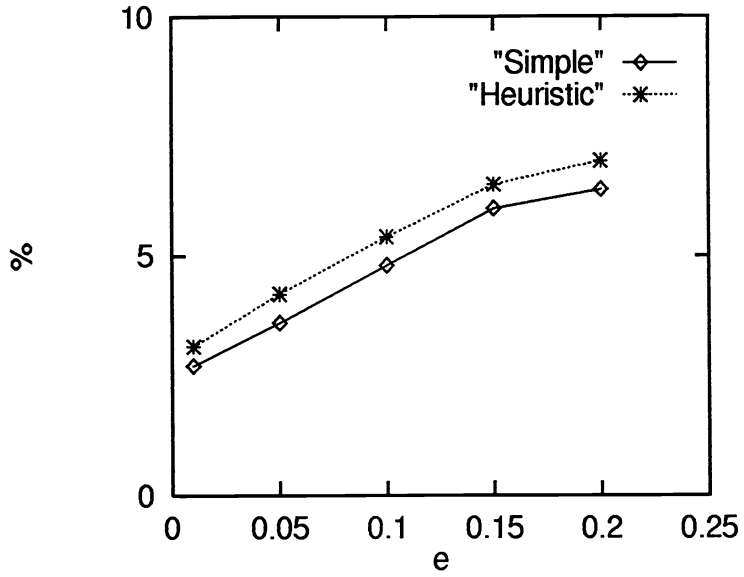

Figure 9: Average percentages of retrieved images versus ϵ.

The two-level index has been implemented in C under the Unix operating system. Each histogram associating with an image can have at maximum 1800 bins which represent 36 groups of hue values, 10 groups of saturation values and 5 groups of intensity values. Before conducting the experiments, we observed that there are only about 20% of the 1800 color indexes are used amongst the images in our database. Hence, we decide to use 322 bins for our color histograms and 12-dimensional hyper-rectangles to represent the root nodes of the R-image trees. In order to compare the results, we use the percentage of retrieval, PR, as a performance metric for image queries that is defined below.

$$PR = \frac{n}{T} \times 100 \tag{9}$$

where n is the number of retrieved images for the query and T is the total number of images in the image database.

The experiments are conducted by randomly selecting 10 images from the database as the query images. They are done in a dedicated Sun SparcStation in order to minimize the deviations due to the operating environment. In the experiment, R-image trees of the query images are first created and the average query result of the 10 images is reported. There are 5 different values of ϵ's: 0.01, 0.05, 0.1, 0.15, 0.2. The same ϵ is used both in the Equations (7) and (8). The heuristic method as described in Section 3 has been implemented together with the simple method. The results are showed in Figure 9. We observe that the two-layer index has a relatively good performance even at higher values of ϵ's. We believe this is due to the early rejections of candidate images by exploiting the spatial information. Furthermore, it is noteworthy that the heuristic method is compared favorably with the simple method in our results for all values of ϵ's. This suggests less computational effort is needed to construct the R-image tree of a query image.

6 REFERENCES

[1] T. Kato, T. Kurita and H. Shimogaki. "Intelligent Visual Interaction with Image Database Systems- Toward the Multimedia Personal Interface". *Journal of Information Processing of Japan*, Vol. 14, No. 2, pp. 134-143, 1991.

[2] W. Niblack et al. "The QBIC project: Querying Images by content using color, texture, and shape." *In SPIE*

1908, Storage and Retrieval for Image and Video Databases, Feb. 1993.

[3] Y. Gong, H. Zhang, H.C. Chuan, M. Sakauchi. "An Image Database System with Content Capturing and Fast Image Indexing Abilities", *Proc. of International Conference on Multimedia Computing and Systems*, Los Alamitos, U.S.A., 1994, pp. 121-130.

[4] M.J. Swain and D.H. Ballard. "Color indexing". *International Journal of Computer Vision*, Vol. 7, No. 1, pp. 11-32, 1991.

[5] H. Lu, B.C. Ooi, KL Tan. "Efficient Image retrieval By Color Contents". *Proc. of Applications of Databases, First International Conference*, June 1994, pp. 95-108.

[6] D.H. Ballard, C.M. Brown. *Computer Vision*, 1982, Prentice Hall.

[7] T. Lee, V. Ng, D. McLean, A. Coldman, R. Gallagher, J. Sale. "A Multi-Stage Segmentation Method for Images of Skin Lesions", *Proc. of IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, May 17-19, 1995, Victoria, Canada.

[8] A. Guttman. "R-Trees: A Dynamic Index Structure for Spatial Searching", *Proc. of SIGMOD*, 1984, pp. 47-57.

[9] H.S. Sawhney and J.L. Hafner. "Efficient Color Histogram Indexing", *Proc. of IEEE International Conference on Image Processing*, November 1994.