JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Physical Storage Design Method for Access Structures of Image Information Systems

Jung-A Lee* and Jong-Hak Lee*

Abstract

This paper presents a physical storage design method for image access structures using transformation techniques of multidimensional file organizations in image information systems. Physical storage design is the process of determining the access structures to provide optimal query processing performance for a given set of queries. So far, there has been no such attempt in the image information system. We first show that the number of pages to be accessed decreases as the shape of the given retrieval query region and that of the data page region become similar in the transformed domain space. Using these properties, we propose a method for finding an optimal image access structure by controlling the shapes of the page regions. For the performance evaluation, we have performed many experiments with a multidimensional file organization using transformation techniques. The results indicate that our proposed method is at least one to maximum five times faster than the conventional method according to the query pattern within the scope of the experiments. The result confirms that the proposed physical storage design method is useful in a practical way.

1. Introduction

The efficient management of images with size in space is required in the new storage structure application such as image retrieval system and VLSI design, etc. [1]. In recent years, researches on image information storage systems have been actively conducted, and studies on data modeling and access methods for images are being carried out [2].

As an access method for image data, a new image access method (IAM) is needed considering the inherent characteristics of image search queries such as retrieving object images adjacent to each other in space. Since the image for an object has a size in space, the IAM must have a mechanism for managing the size of the image. Most IAMs are extensions of the multidimensional point access method (PAM) [3,4], which has been studied since the late 1970s, by adding features that can process image sizes. IAMs can be classified into space filling curve techniques, image clipping or image duplication techniques, region overlapping techniques, and transformation techniques depending on how the image size is managed [5-7].

Transformation techniques, a class of IAMs, transform spatial images into point images within a

transform space using parameters that can represent the size of spatial images in the original space, and manage point images using a multidimensional file structure [6]. Two representative transformation techniques are corner transformation and center transformation [6]. These techniques index the image using a minimum bounding rectangle (MBR), which is a rectangle whose sides are parallel to the x and y axes and minimally enclose the image shape. In the two-dimensional source space, the corner point transformation method expresses the image as a point in the four-dimensional transform space using the lower-left point and the upper-right point of the MBR as parameters.

In general, the most important factor for evaluating the performance of a storage management system (SMS) is the efficiency of retrieval processing, i.e., how quickly the SMS can respond to users' retrieval queries [8]. Retrieval processing is the process of searching the records that satisfy retrieval conditions from the entire records. To improve the performance of SMS, a physical storage design technique [9,10] is used. Physical storage design is the process of determining the access structures to provide optimal query processing performance for supporting efficient retrieval processing based on precollected information about the queries [9]. This physical storage design is also required for access structures for image information storages. So far, there has been no such attempt in the image information system. In this paper, we propose a physical storage design method for image information storages related to the multidimensional clustering property and image information access structures using corner transformation techniques.

When a region where a range retrieval query is located within a transform space is called a retrieval region, all image retrieval queries given to the original space in the image information access structure using transformation techniques are transformed into one type of range retrieval query in the transform space [3] and therefore, the user retrieval query of the image information system can be interpreted as an operation of searching for images included in the retrieval region in the transform space. If the region on the transform space corresponding to the data page of the image information access structure is a page region, the problem of designing the storage structure of the image information system is to minimize the number of page regions intersect with a series of user retrieval regions.

In this paper, we define the ratio of the interval size of each axis constituting a region as an interval ratio, and express the shape of a region as the interval ratio. Based on the information about the types of retrieval regions appearing in the image information user retrieval query pattern, the optimal interval ratio of the page region that minimizes the number of page regions intersect with retrieval regions is determined and the optimal image information access structure can be constructed by using a region-bisecting method that makes page regions with such an optimal interval ratio as possible.

This paper is organized as follows. Section 2 describes the construction process of the image information storage structure. The process of extracting the attribute values of the object image information is explained, and the image information access structure using the multidimensional file structure to be constructed as these attribute values is introduced. Section 3 describes a physical storage design method for constructing an image information access structure that can process image retrieval queries the most efficiently. Section 4 presents the experimental environment and experimental results for performance evaluation. Finally, Section 5 makes a conclusion.

2. Construction of Image Information Storage

The attribute values used to store and retrieve information about the object image lying in the space

include the size of the MBR region of the object image, the dominant hue value, and the RGB color average ratio. In this section, we briefly introduce methods for extracting these attribute values for constructing the image information storage. We introduce the MBR-multilevel grid file (MBR-MLGF) [11], which is a multidimensional file structure using the corner transformation technique to be used as the image information storage.

2.1 Extraction of MBR Region Size

It is difficult to extract the exact feature values from the object image acquired from the camera since the background ratio occupies a high proportion in the object source image and the background image information changes with the illumination intensity variation [12]. In this paper, in order to increase the recognition rate of the object image, unnecessary background information is removed and at the same time, the width and height size attribute values of the MBR region of the object are extracted.

First, we obtain images separated by red, green, and blue channels for the RGB original image of the object. And the adaptive binary threshold inverse [12] is used for the image of each channel to obtain a new object image which is represented as 1 if it is smaller than the threshold value and 0 if it is larger. Then, after performing the multiplication operation for the extracted image for each RGB channel, one object region is obtained by using the binary threshold inverse. This is calculated as in Eq. (1).

$$M(i,j) = \begin{cases} 0 & if \ R_{ij} = G_{ij} = B_{ij} = 255 \\ 255 & \text{otherwise} \end{cases} \tag{1}$$

The image obtained by Eq. (1) represents a monochrome image. When the values of red, green, and blue are all 255 (8 bits are all 1) as shown in the background part, black (value is 0) is displayed. Otherwise, white (value is 255) is displayed.

Finally, in order to calculate the MBR region, we obtain an image by applying the contour algorithm [12] on the image obtained by Eq. (1) to extract the contour lines of the object region with intensity 255. However, in some cases, contour lines are generated in the background region, not in the object region, depending on the illuminance change of lighting. In order to find the object region, the longest contour lines among the horizontal and vertical contours are searched for, and the width and height are calculated based on the intersections of them and used as the width and height size attribute values for the MBR region of the object image.

2.2 Extraction of Color Feature Values

2.2.1 HSV color model

The HSV color model is represented by a hue component H representing a pure color, a saturation component S representing the darkness and lightness of the color, and a value component V representing the degree of brightness of the color [13]. The HSV color model, which has characteristics similar to human visual capabilities, is relatively insensitive to changes in illumination intensity during image acquisition compared to the RGB color model [14] in which each color is represented by the values of red, green, and blue which are primary color spectrum component values [12]. Fig. 1 represents the HSV color model.
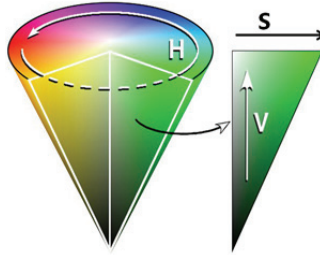
Fig. 1. HSV color model.

As can be seen in Fig. 1, H starts with red, usually at an angle of 0, and makes a circle in a counterclockwise direction. S represents the vector length from the center of a circle to an arbitrary point. If it is in the center of a circle, it indicates low saturation. If it is located outside the circle, it indicates high saturation. V is a vertical component of the inverted cone shape, representing the degree of lightness from the lower dark portion to the higher light portion.

The RGB color model expresses each color as a single point in a cube with the values of R (red), G (green), and B (blue), which are primary color spectrum component values as the axes. When the values of each axis are normalized to the range [0, 1], the RGB three primary color values are located at three corners of the coordinates—R (1,0,0), G (0,1,0), B (0,0,1)—and black is located at (0,0,0), which is the origin of coordinates, and white is at (1,1,1), the farthest corner from the origin. Since the RGB model creates color by combining the three basic colors, the cross-correlation of each color component of R, G, and B is large, and the RGB size varies with the variation of the external illumination intensity, which may lead to unstable results.

Therefore, we use H, S, V values of HSV color model as attribute values of the object image. When the image of the RGB color model is given, the H, S, and V component values of the HSV color model can be obtained from the values of R, G, and B pixels using the following equations (2), (3), and (4) [13].

$$H = \begin{cases} \theta, & B \leq G \\ 360 - \theta, & B > G \end{cases}$$

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}} \right\} \tag{2}$$

$$S = 1 - \frac{(R+G+B)}{3} \left[\min(R,G,B) \right] \tag{3}$$

$$V = \frac{1}{3}(R+G+B) \tag{4}$$

2.2.2 Creation of color attribute values

Among the color attribute values, the Hue value of the HSV and the average ratio between the RGB channels are robust to the illumination intensity variation [12]. The following procedure is used to generate attribute values such as the dominant hue value and the average ratio between RGB channels.

Step 1: The color region is separated from the image acquired from the camera so that the attribute values can be extracted only in the color region. Since the difference of the red, green, and blue channel values in the color region of the image is larger than that in the white region, the color region can be separated from the image as the difference of the values.

Step 2: A dominant hue attribute value is extracted from the color region. Since the color region extracted in Step 1 is the RGB color model, the inherent color information easily changes by the illumination change of the external lighting. On the other hand, the HSV color model, which has characteristics similar to the human visual ability, has the advantage of insensitivity to the illuminance variation during the image acquisition. Therefore, in this paper, we use the H (hue) component among the HSV color model as an attribute value of image information. That is, the extracted color image is converted into HSV color model, and a histogram of the H component of the divided image is obtained. The size of the H component having the largest accumulated intensity is used as an attribute value, except for the background portion in which the intensity is 0 in the histogram.

Step 3: Extract the attribute value of the average ratio of RGB colors. The attribute value of the color average ratio is obtained by averaging each color region for an image divided into R, G, and B channels, and the ratio of the second largest value to the largest value is used as an attribute value.

2.3 MBR-Multilevel Grid File

In this section, we introduce the MBR-MLGF [11], which is an image information access structure using corner transformation techniques to be used as a storage structure for image information systems. we first introduce the structure of MLGF [4,15], which is the basic structure of MBR-MLGF, and introduce the definition of MBR-MLGF and its structural characteristics and features.

MLGF [4] consists of directories and data pages. The directory has a balance tree structure, and each step directory reflects the division state of the entire space. An entry in the lowest level of a directory not only points to a data page, but also represents a region allocated to the page. One data page stores only data records belonging to the region of the last directory entry. The directory entry at the upper level refers to the directory page at the next lower level and represents the region indicated by the directory page.

MBR-MLGF [11] improved MLGF to maintain additional MBR information in the MLGF directory entry in order to efficiently manage the image information. That is, in order to express the MBR of the images in the original space, each directory entry of the MBR-MLGF includes a minimum left point (min-lx) value and a maximum right point (max-lx) value of the MBR including the images existing in the region corresponding to the directory entry.

The hash value for the lx axis among the region vectors for one region in the MBR-MLGF becomes a prefix of the lx values of the images located in the region. This value is also the prefix of the extreme value for this region. Fig. 2(a) shows the corresponding region and extreme value (thin line) of a directory entry of MBR-MLGF. In this example, it is assumed that the maximum precision for the hash value of each axis of the MBR-MLGF is represented by 8 bits. In this figure, min-lx for images belonging to this region is '10010000', the lx axis hash value of the region vector is '10', and '10' is the prefix of '10010000'.

In order to express the hash value for the i-th axis in the region vector for one region, MLGF records the common prefix of the i-th attributes of this region and its length in the entry. Fig. 2(b) shows a data

structure expressing the hash value for the lx axis in the directory entry of MLGF. The numbers at the beginning of this data structure indicate the length of the common prefix, and the last part shows the hash value in the bit unit. The hash value '10' for the lx axis of the directory entry is displayed at the end of this data structure and this hash value is composed of 2 bits, so 2 is displayed at the beginning. The 6 bits expressed as the remainder 'x' are not currently used for this entry.
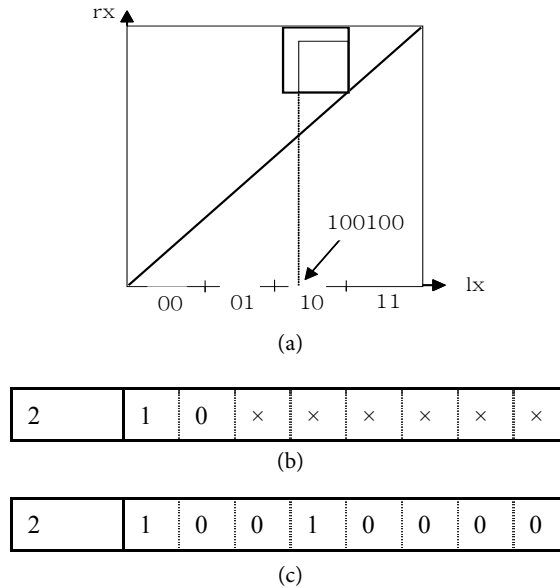

Fig. 2. Expression of extreme value in MBR-MLGF. (a) A region of transform space. (b) Expression of lx-axis hash value in MLGF. (c) Simultaneous expression of lx-axis hash value and extreme value in MBR-MLGF.

Since the extreme values in each directory entry of MBR-MLGF can be expressed using the portion not used in the hash value structure for each axis of MLGF, the size of the directory entry does not change even when extreme value information is added. In Fig. 2(c), the extreme value '10010000' for the lx axis can be expressed using all 8 bits. The two preceding bits represent the hash value for the lx axis of the region vector as it is. Therefore, MBR-MLGF can maintain the extreme value with no extra storage space overhead with the same size data structure as MLGF's directory entry.

3. Physical Storage Design for Image Information System

We now present a physical storage design method for image information access structures using the transformation technique. In this section, we describe a feature that all image search queries given to the original space are transformed into one type of range retrieval query in the transform space. And then, we propose an optimal condition for the access structure that minimizes the number of average page accesses occurring during query processing and a bisecting method for the image information access structure that satisfies this condition. Finally, we present a physical storage design algorithm to provide optimal image retrieval query performance.

3.1 Characteristics of Image Retrieval Queries

Image retrieval queries are queries that retrieve images with a specific spatial relationship to a given retrieval region. Although a standard set of image retrieval queries that satisfy all the requirements for image data processing for objects has not been established yet, basic types for image retrieval queries were proposed in several studies [6,16] and these can be summarized as follows:

- Region intersection query: Retrieves all images that intersect a given retrieval region.
- Region containment query: Retrieves all images contained in a given retrieval region.
- Region enclosure query: Retrieves all images containing a given retrieval region.
- Point query: Retrieves all images containing a given query point. This can be seen as a special case of the region enclosure query.
- Nearest neighbor query: Retrieves the nearest image at the given query point.

Of these basic types, region intersection query, region containment query, region enclosure query, and point query are queries that retrieve all the images satisfying the specific spatial relation to a given retrieval region (query point in the case of point query), so these are collectively referred to as a region query. When a query that cannot be processed by the basic type of image retrieval queries is required, these region queries can be sequentially performed to process the desired query.
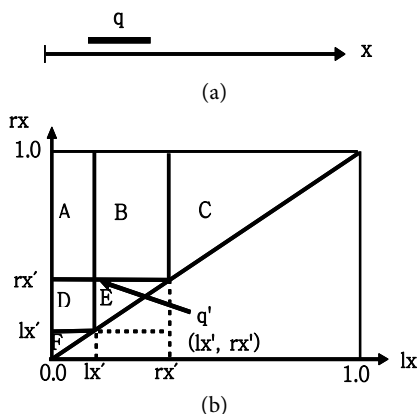

Fig. 3. Region classification by spatial relation of transformation technique. (a) Retrieval region q in the original space. (b) Region classification by spatial relation in the transform space.

In the image information access structure using the transformation technique, all the image retrieval queries given to the original space are converted into one type of range retrieval query in the transform space. Fig. 3 shows a retrieval region q in a one-dimensional source space as a query point q' in a two-dimensional transform space, and the transform space divided into six regions from A to F according to the relative spatial relation to the query point q'. The characteristics of each region are as follows:

- Region A: All images containing q in the original space are located. This is because the lx value of all the images in the region A is smaller than lx' (the lx value of q') and the rx value is larger than rx' (the rx value of q').
- Region B: All images crossed by the right point of q in the original space are located. This is because the lx values of all the images in the region B are between lx' and rx' and the rx value is larger than rx'.

- Region C: All images existing on the right side of *q* in the original space are located. This is because the *lx* value of all images in region C is larger than *rx′*.
- Region D: All images crossed by the left point of *q* in the original space are located. This is because the *lx* value of all the images in the region D is smaller than *lx′* and the *rx* value is between *lx′* and *rx′*.
- Region E: All images included in *q* in the original space are located. The *lx* value of all the images in the region E is larger than *lx′* and the *rx* value is smaller than *rx′*.
- Region F: All images existing on the left side of *q* in the original space are located. This is because the *rx* value of all the images in the region F is smaller than *rx′*.

Using the above region classification, various region queries in the original space can be transformed into a range retrieval query in the transform space. The following shows which range queries for a given retrieval region *q* are transformed into certain range retrieval queries.

- Region intersection query: The parts that need to be retrieved from the transform space of Fig. 3(b) are A, B, D and E regions. Therefore, this query is a range retrieval query whose range to retrieve in the transform space is *lx ≤ rx′ AND rx ≥ rx′*.
- Region containment query: The part that needs to be retrieved from the transform space of Fig. 3(b) is the E region. Therefore, this query is a range retrieval query whose range to retrieve in the transform space is *lx′ ≤ lx ≤ rx′ AND lx′ ≤ rx ≤ rx′*.
- Region enclosure query: The part that needs to be retrieved from the transform space of Fig. 3(b) is the A region. Therefore, this query is a range retrieval query whose range to retrieve in the transform space is *lx ≤ lx′ AND rx ≥ rx′*.
- Point query: If the coordinate of the query point q_p given in the original space is $q_p.x$, this query will be a range retrieval query whose range to retrieve in the transform space is *lx ≤ $q_p.x′$ AND rx ≥ $q_p.x′$*.

3.2 Basic Design Principles of Image Storage Structure

In this section, we represents the effect of the retrieval performance according to the shape of the page regions in the two-dimensional domain space. Fig. 4 shows three types of two-dimensional domain spaces partitioned into page regions for two organizing attributes X and Y. Each domain space has different shape of page regions with the same size 16. That is, Fig. 4(a)–(c) show the domain spaces partitioned into page regions whose interval ratios of Y over X are 1/16, 16, and 1, respectively. And, Fig. 4 shows that the number of page regions—these are the same as the number of data pages accessed to process the retrieval query—crossed by a retrieval region given at an arbitrary location in each domain space. We use three types of retrieval region *A*, *B*, and *C* of the same sizes having the interval ratio 1/4, 4, and 1, respectively. Here, the interval ratio is the ratio of intervals for the attributes in the multidimensional domain space and use it to represent the shape of the region.

In Fig. 4, we can see that the number of page accesses that the query will incur depends on how the domain space is partitioned into page regions and how similar the shape of the retrieval region is to that of the page region. The number of page regions that intersect with the retrieval region is as follows. That is, in the case of the retrieval region A, 3 page regions (left), 12 (center), and 8 (right) in Fig. 4. In the case of the retrieval region B, 12 (left), 3 (center), and 8 (right) in Fig.4. In the case of region C, 6 (left), 6

(center), and 4 (right) in Fig. 4. It can be observed that the number of pages to be accessed decreases as the shape of the given retrieval region and that of the page region become similar. Therefore, in this paper, we use this principle to improve the retrieval performance of the image information system.
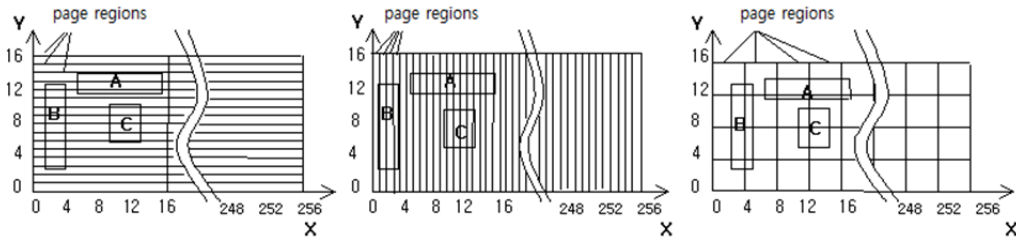

Fig. 4. Two-dimensional domain spaces partitioned into different shapes of page regions.

3.3 Page Region Configuration of Image Information Storage

We discuss the optimal condition for query processing as a correlation between the interval ratio of the retrieval region where the retrieval query is located in the transform space and that of the page region where the data page of the access structure is located. And, by using a two-dimensional MBR-MLGF for the simplicity of discussion, we propose a region bisecting method that approximates the interval ratio of the page region to the optimal interval ratio of the page region that is appropriate for the optimal condition of query processing, and extend it to the four-dimensional MBR-MLGF and apply it to the image information storage.

Multidimensional file structure has a feature that the average number of page regions crossed by a given retrieval region varies depends on the shapes of page regions constituting the access structure. Using this feature, Lee et al. [17] proposed a method of calculating the optimal interval ratio of the page regions that minimize the average number of accesses of the page regions for a given retrieval region within a multidimensional space. In this paper, we calculate the optimal interval ratio of the page regions for the image information access structure in this way.

The method of calculating the optimal interval ratio of the page regions when the data is uniformly distributed in a two-dimensional space can be summarized as follows: if the data are uniformly distributed, the size of the page regions constituting the domain space will be constant, and the optimal interval ratio of the page regions minimizing the number of page regions crossed by the given retrieval regions can be calculated as the ratio of the sum of the interval sizes for each axis of all retrieval regions. That is, for $p_i(x) \times q_i(y)$ ($i = 1, \dots, n$), the n retrieval regions given to an arbitrary position on a two-dimensional space partitioned into page regions with constant size, the optimal interval ratio ($p(x)$: $p(y)$) of page regions, which minimizes the total number of page regions crossed by the retrieval regions is $\sum_{i=1}^{n} q_i(x)$: $\sum_{i=1}^{n} q_i(y)$ [17].

The nonuniform distribution of data in the multidimensional space means that the size of the page region varies due to the different data density depending on the position in the domain space. That is, in a high density region, a large number of pages are allocated as compared with a low density region, so that the size of each page region becomes small. Therefore, in the case of the nonuniform distribution, the number of the page regions crossed by the retrieval region is proportional not only to the size of the retrieval region but also to the data density of the position where the retrieval region is given.

Thus, in order to properly estimate the size of the retrieval region, each retrieval region must be weighted with data density. We define a *normalized retrieval region* as the one weighted with the data density at the position where the retrieval region is given, and calculate the optimal interval ratio of page regions by using normalized retrieval regions. That is, let n retrieval regions of varying sizes $q_i(x) \times q_i(y)$ $(i = 1, \dots, n)$, where the data density of each region is $d_i (= n_{oi}/q_i(x) \times q_i(y))$, n_o_i is the number of object images in the retrieval region $q_i(x) \times q_i(y))$, be given in a two-dimensional space partitioned into page regions of varying sizes. Then the optimal interval ratio $(p(x) : p(y))$ of a page region is $\sum_{i=1}^{n} q_i(x)\sqrt{d_i} : \sum_{i=1}^{n} q_i(y)\sqrt{d_i}$.

In this paper, we calculate the optimal interval ratio of the page regions by using retrieval regions expressed on multidimensional transform space by object image retrieval queries given as query patterns. And we present a region-bisecting method that allows the domain space of the image information access structure to be composed as page regions of the optimal interval ratio.

In MBR-MLGF, if the capacity of a page is exceeded due to the insertion of a new object image, the page region corresponding to the page is bisected into two equal-sized subregions. At this time, by choosing a bisecting axis that makes interval ratio of subregions is closer to the optimal interval ratio, it is possible to induce the interval ratio of all page regions to approach the optimal interval ratio at the time of successive bisection due to the continuous insertion of images.

When a retrieval region is given to an arbitrary location on a two-dimensional domain space, the following theorem 1 presents the condition that the size of the location region of the retrieval region crossed by a certain page region is minimized.

THEOREM 1. If any retrieval region $q(x) \times q(y)$ is given in a two-dimensional domain space, the region where the left-upper point of the retrieval region crossed by a certain page region $p_i(x) \times p_i(y)$ could be located is minimized when the shape of the page region is equal to that of the given retrieval region. That is, $p_i(x) : p_i(y) = q(x) : q(y)$.

Proof. See the reference [17].

Theorem 1 can be used to select the bisecting axis that makes the shape of subregions closer to the optimal interval ratio for bisecting the overflowed page region. Let the optimal interval ratio be $a:b$. In Fig. 5, LQ_x and LQ_y represent the shadowed regions where the center point of a retrieval region Q of the $a \times b$ (having optimal interval ratio $a:b$) could crossed by one of the subregions bisected from a page region $p(x) \times p(y)$. Fig. 5(a) shows LQ_x for choosing the X axis, and Fig. 5(b) shows LQ_y for choosing the Y axis. The sizes of LQ_x and LQ_y are obtained as follows:

$$\text{SIZE}(\text{LQ}_x) = (p(x)/2 + a)(p(y) + b) \tag{5}$$

$$\text{SIZE}(\text{LQ}_y) = (p(x) + a)(p(y)/2 + b) \tag{6}$$

According to the Theorem 1, the size of the LQ is minimized when the shape of the page region is equal to that of the retrieval region. Thus, the interval ratio of the page region gets closer to that of the retrieval region Q as the size of the LQ gets smaller. Therefore, by selecting the axis with the smaller LQ size, the interval ratio of the page region after bisecting can be made closer to the given optimal interval ratio.
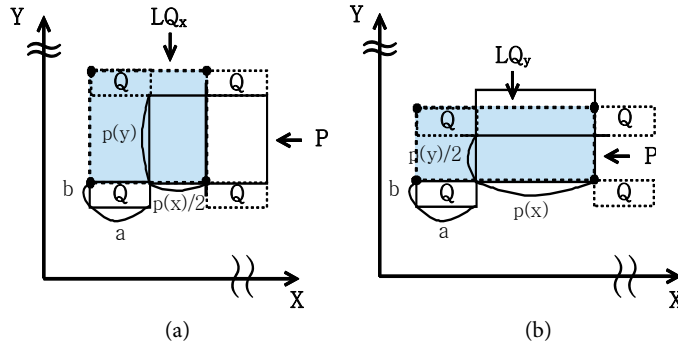
Fig. 5. The region where the center point of a retrieval region could intersect with one of the subregions bisected from a page region. (a) X axis bisect. (b) Y axis bisect.

In MBR-MLGF, the interval size of each axis of the page region is maintained in the region vector of the lowest level directory entry of the directory (see Fig. 2). That is, when the length of the hash value for one axis X of the region vector is $l(x)$, the interval size of that axis is inversely proportional to $2^{l(x)}$. Thus, the interval size for the X axis is $A(x)/2^{l(x)}$, where $A(x)$ is the entire domain size of X axis. In the two-dimensional MBR-MLGF composed of X and Y axes, a region-bisecting method for approximating the interval ratio of the page region to $a{:}b$ is as follows: when substituting $A(x)/2^{l(x)}$, $A(y)/2^{l(y)}$ for $p(x)$, $p(y)$ of Eqs. (5) and (6), the X axis is bisected if $(A(y) \times 2^{l(x)} \times a < A(x) \times 2^{l(y)} \times b)$ and otherwise, the Y axis is bisected.

The region-bisecting method of the two-dimensional MBR-MLGF can be extended to the four-dimensional MBR-MLGF. That is, when bisecting an overflow page region of the four-dimensional MBR-MLGF formed by W, X, Y, and Z axes, we can select a bisecting axis that makes the bisected subregion the closest to the given optimal interval ratio. Assuming that a retrieval region R of the $a \times b \times c \times d$ (having optimal interval ratio $a{:}b{:}c{:}d$) is given at an arbitrary location on the four-dimensional space, for the page region $p(w) \times p(x) \times p(y) \times p(z)$ to be bisected, we first calculate the size of the LR_i's, where LR_i is the one where the each axis is chosen for the bisecting axis. Than, we select the axis that makes LR the smallest. For example, if we select the W axis as the bisecting axis, the size of the LR_w is $(p(w)/2+a)(p(x)+b)(p(y)+c)(p(z)+d)$.

3.4 Design Algorithm of Image Information Access Structure

The physical design algorithm of image information storage is generalized for the optimal configuration of the four-dimensional image access structure by extending the concepts developed for the two-dimensional space in Section 3.3 and consists of the following three steps.

Step 1: The *normalization of each retrieval region* in the four-dimensional transform space for a given query pattern in a two-dimensional source space. Let an arbitrary retrieval region in the four-dimensional domain space be $p(w) \times p(x) \times p(y) \times p(z)$ and the number of image object in the retrieval region be n_o. Then, we normalize the retrieval region as $p(w)d^{1/4} \times p(x)d^{1/4} \times p(y)d^{1/4} \times p(z)d^{1/4}$ multiplying the interval of each axis by weighting factor $d^{1/4}$, where the record density d is $n\_o/(p(w) \times p(x) \times p(y) \times p(z))$.

Step 2: The *determination of the optimal interval ratio $a{:}b{:}c{:}d$* of page regions of the storage. This optimal interval ratio is obtained by summing the normalized intervals of the all retrieval regions for each axis.

Step 3: The *construction of the optimal configuration for image information access structure*. We construct an image information access structure composed of page regions that are closest to the optimal interval ratio. Here, we apply the page region configuration method mentioned in Section 3.3. The region-bisecting method of the two-dimensional MBR-MLGF is extended to the four-dimensional MBR-MLGF. If overflow occurs in the data page of the MBR-MLGF due to the subsequent insertion of the object image, the page region corresponding to this data page is bisected into two subregions, and the images of the original data page are divided and stored in two data pages corresponding to the bisected page subregions. Assuming that a retrieval region R of the $a{\times}b{\times}c{\times}d$ (having optimal interval ratio $a{:}b{:}c{:}d$) is given at an arbitrary location on the four dimensional space, for the page region $p(w){\times}p(x){\times}p(y){\times}p(z)$ to be bisected, we first calculate the size of the LR_i's, *where LR_i is the one where* the each axis is selected for the bisecting axis. Than, we select the axis that makes LR the smallest. The size of the LR is $(p(w)/2+a)(p(x)+b)(p(y)+c)(p(z)+d)$ if we select the W axis as the bisecting axis, $(p(w)+a)(p(x)/2+b)(p(y)+c)(p(z)+d)$ if we select the X axis as the bisecting axis, $(p(w)+a)(p(x)+b)(p(y)/2+c)(p(z)+d)$ if we select the Y axis as the bisecting axis, $(p(w)+a)(p(x)+b)(p(y)+c)(p(z)/2+d)$ if we select the Z axis as the bisecting axis.

4. Performance Evaluation

The storage structure of the existing image information retrieval system is a one-dimensional sequential file in which object images are sorted and stored by one attribute value [12]. For the retrieval of a specific range of images in this storage structure, the stored object images are first binary searched with sorted one of the retrieval attribute keys, and the remaining retrieval attribute keys are compared one by one.

The length of the record for one object image to be retrieved in the experiment is about 200 bytes, the number of information of the image information image, that is, the number of records is 400,000, and the size of the page constituting the file is 2048 bytes. In this case, the number of data pages constituting the file is about 40,000. In order to retrieve the object image information by comparing the four attribute values one by one, all 40,000 pages should be accessed. In a sequential file structure constructed by sorting records with one attribute value, a binary search can be performed for retrieval of a first image record having a value of a retrieval range of the attribute value. Therefore, In this case, 16 ($=\lceil Log_2 40{,}000 \rceil$) pages should be accessed, and then the remaining three attribute values must be compared one by one. Therefore, if the retrieval range of the sorted attribute value is about one hundredth of the total value, 400 pages should be accessed, so a total of 416 pages should be accessed.

The following shows the experimental results for the performance evaluation of the multidimensional file structure used as the object image information storage system proposed in this paper. We constructed a storage structure containing 200,000 records using MBR-MLGF as a multidimensional file structure used in this experiment. The distribution characteristics of the attribute values of the record used to construct the storage structure have a uniform distribution and a nonuniform distribution. For the nonuniform distribution, the values of each attribute taking normal distribution $N(\mu, \sigma^2)$ with a standard deviation $\sigma = 2^{31}{\times}2/5$ over the range $[-2^{31}, 2^{31}-1]$ are artificially generated and used.

In the first experiment, to simplify the experiment, we used two-dimensional MBR-MLGFs by limiting the retrieval attribute values of image information to two, and the values of each attribute are uniformly distributed over the range of $[-2^{31}, 2^{31}-1]$. The purpose of this experiment is to confirm that the optimal interval ratio of the page regions minimizing the number of page regions crossed by the given retrieval regions can be calculated as the ratio of the sum of the interval sizes for each axis of all normalized retrieval regions when the records are uniformly distributed. The retrieval regions generated by the experiment have interval ratios of 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, and 1/512. For each of these intervals, we generated retrieval regions with three different sizes: (1) large retrieval regions (1/500 of the domain space) L_1, L_2, L_4, L_8, L_16, L_32, L_64, L_128, L_256, and L512; (2) medium retrieval regions (1/5,000 of the domain space) M_1, M_2, M_4, M_8, M_16, M_32, M_64, M_128, M_256, and M_512; (3) small retrieval regions (1/50,000 of the domain space) S_1, S_2, S_4, S_8, S_16, S_64, S_128, S_256, and S_512.

We generated several two-dimensional MBR-MLGFs with uniform distribution of data with various target interval ratios of page regions. We then measured the number of page accesses for processing a query pattern with mixed types of retrieval regions for each target interval ratio. Target interval ratios of page regions used for generation the MBR-MLGFs were 1/.5, 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, and 1/1024. The query pattern used included large retrieval regions L_1, L_2, L_4, medium ones M_8, M_16, M_32, and small ones S_64, S_128, S_256. We generated 20 retrieval regions for each type of large retrieval regions, 200 for each type of medium ones, and 2000 for each type of small ones. All of these retrieval regions are uniformly distributed in the domain space.

The results are plotted in Fig. 6, where the horizontal axis represents the interval ratio indicating the type of the page regions constituting the MBR-MLGF, and the vertical axis represents the number of page accesses in processing the query pattern. The optimal interval ratio, such that the ratio of the sum of the interval sizes for each axis of all retrieval regions, was calculated as 1:62. As can be seen in Fig. 6, the best performance is achieved in the MBR-MLGF having a target interval ratio of 1:64, which is closest to this optimal interval ratio. These experimental results justify the purpose of our first experiment. Therefore, when the multidimensional file structure is used as the image information storage, the performance of the retrieval processing of the image information can be improved by constituting the page regions having such an optimal interval ratio.
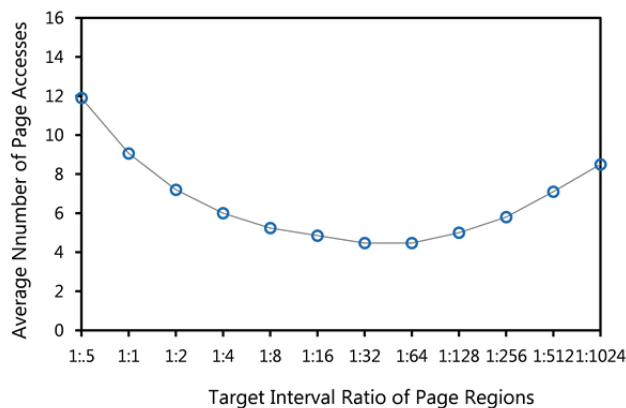

Fig. 6. Image retrieval processing cost in various two-dimensional MBR-MLGFs for data with uniform distribution.

In the second experiment, the same experiment as the first experiment is performed on the four-dimensional file structure with nonuniform distribution of data. That is, we generated several four-dimensional MBR-MLGFs having page regions of various target interval ratios with the nonuniform distribution data. We then measured the number of page accesses occurring when retrieving images belonging to various types of retrieval regions for each target interval ratio. The purpose of this experiment is to verify the correctness of the design algorithm presented in Section 3.4 for the image information storage structure composed of four-dimensional file structure. The types of retrieval regions used in the experiment were five types of small retrieval regions S1_1_1_1, S1_2_4_8, S1_4_16_64, S1_8_64_512, and S1_16_256_4096 that had interval regions of 1/1/1/1, 1/2/4/8, 1/4/16/64, 1/8/64/512, and 1/16/256/4096, respectively. In order to construct a query pattern, we generated 200 retrieval regions of each type, where they were uniformly allocated on the domain space.

The experimental results are plotted in Fig. 7. After the normalization for each retrieval region, the optimal interval ratio of page regions, such that the ratio of the sum of the interval sizes for each axis of all normalized retrieval regions, was calculated as 1:6:68:936. As can be seen in Fig. 7, the best performance is achieved in the MBR-MLGF having the page regions with the interval ratio closest to the optimal one calculated. These experimental results justify the purpose of our second experiment. That is, for the nonuniformly distributed four-dimensional MBR-MLGF, the optimal interval ratio of the page regions can be calculated as the ratio of the interval sizes of the normalized retrieval regions for each axis. In addition, the results of this experiment demonstrate that the proposed Physical design algorithm for constructing a four-dimensional image information access structure is useful in a practical way.
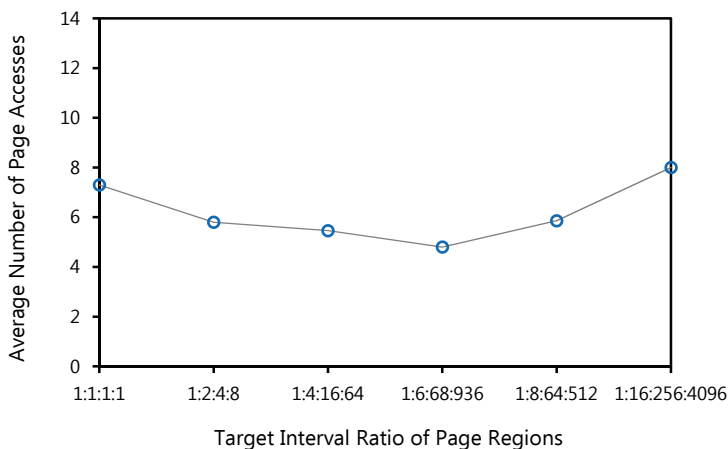

Fig. 7. Image retrieval processing cost in various four-dimensional MBR-MLGFs for data with nonuniform distribution.

Finally, in the third experiment, we compared the performance of the MBR-MLGFs constructed using the proposed design algorithm of image information storage structure with that of the one (which we define as the formal MBR-MLGF) constructed using the cyclic-bisecting method (i.e., target interval ratio = 1/1/1/1). We generated five four-dimensional MBR-MLGFs with nonuniform distribution of data having the target interval ratios of 1/1/1/1, 1/2/4/8, 1/4/16/64, 1/8/64/512, and 1/16/256/4096. For

each MBR-MLGF, we measured the retrieval cost with a query pattern consisting of 2000 small retrieval regions having the same target interval ratio. The results were compared with the retrieval cost with the same query patterns in the formal MBR-MLGF.

Fig. 8 shows the experimental results, where the horizontal axis represents the interval ratio of page and retrieval regions, and the vertical axis represents the efficiency ratio, which is the ratio of the image retrieval cost using the proposed MBR-MLGF over those using the formal MBR-MLGF. As shown in Fig. 8, the more the interval ratio of retrieval regions deviates from 1/1/1/1, the higher the efficiency ratio we obtain. When the interval ratio of the retrieval region is 1/16/256/4096, the retrieval processing performance is improved to 5.1 times. The performance should be enhanced further for the retrieval regions having biased interval ratios.
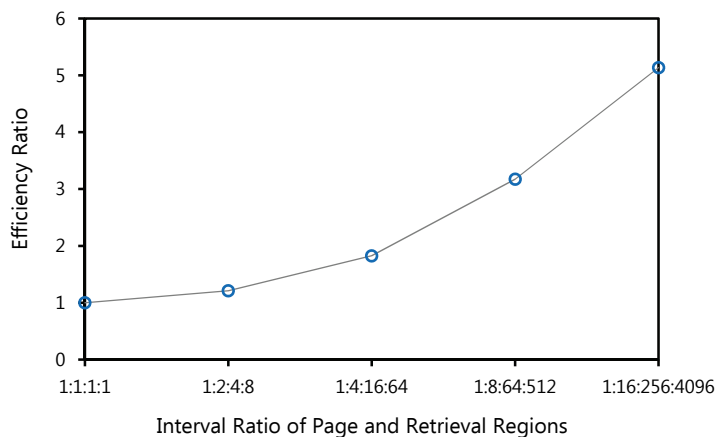

Fig. 8. The efficiency ratio of the design algorithm for image information access structure.

5. Conclusion

We have proposed a physical storage design method for optimal image retrieval processing in image information systems. We have constructed an image information storage using MBR-MLGF, one of the multidimensional file structures using corner transformation techniques, and shown that various types of image retrieval queries given in the original space can be transformed as a form of range query in the transform space. In the proposed design method, we have first performed a normalization process in which data density is weighted for each retrieval query region given as design information in order to consider data distribution property in the domain space. And then, we have determined the optimal interval ratio of the page regions constituting a storage structure as the ratio of the values obtained by summing the sizes of the intervals for each axis for all the normalized retrieval regions. Finally, we have constructed a MBR-MLGF using a region-bisecting method that makes page regions with such an optimal interval ratio as possible.

In MBR-MLGF constructed in this way, the performance could be improved several tens times as compared with the existing storage structure in which object images are sorted and stored by one attribute value using a one-dimensional sequence file, and then, for the retrieval of specific images,

stored object images are first binary searched with this sorted attribute value and the remaining attribute values are compared one by one. In addition, we have confirmed that the retrieval query performance of MBR-MLGF constructed by applying the design method proposed in this paper is improved up to five times according to the interval ratio of a given retrieval region as compared with the storage structure built with existing MBR-MLGF as it is. Also, as the retrieval region shape is biased toward a specific attribute, we predict that the image retrieval performance can be improved more than the performance confirmed in this paper.

Acknowledgement

References

[1] R. H. Guting, "An introduction to spatial database systems," *The VLDB Journal—The International Journal on Very Large Data Bases*, vol. 3, no. 4, pp. 357-399, 1994.

[2] J. H. Lee, "A design of storage structure for medicine bottle searching system," *Journal of Information Technology and Architecture*, vol. 12, no. 3, pp. 429-139, 2015.

[3] J. T. Robinson, "The KDB-tree: a search structure for large multidimensional dynamic indexes," in *Proceedings of the 1981 ACM SIGMOD International Conference on Management of Data*, Ann Arbor, MI, pp. 10-18.

[4] K. Y. Whang and R. Krishnamurthy, *Multilevel Grid File*. Armonk, NY: IBM, 1985.

[5] B. C. Ooi, R. Sacks-Davis, and K. J. McDonell, "Spatial indexing in binary decomposition and spatial bounding," *Information Systems*, vol. 16, no. 2, pp. 211-237, 1991.

[6] H. Kriegel and B. Seeger, "Techniques for design and implementation of efficient spatial access methods," in *Proceedings of the 14th VLDB Conference*, Long Beach, CA, 1988, pp. 360-370.

[7] T. Sellis, N. Roussopoulos, and C. Faloutsos, "The R+-Tree: a dynamic index for multi-dimensional objects," in *Proceedings of 13th International Conference on Very Large Data Bases*, Brighton, England, 1987, pp. 507-518.

[8] N. S. Kim, S. A. Lee, S. H. Jo, and J. H. Kim, "Multi-dimensional keyword search and analysis of hotel review data using multi-dimensional text cubes," *Journal of Information Technology and Architecture*, vol. 11, no. 1, pp. 63-73, 2014.

[9] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 2nd ed. Amsterdam: Benjamin/Cummings Publishing, 1994.

[10] S. Finkelstein, M. Schkolnick, and P. Tiberio, "Physical database design for relational databases," *ACM Transactions on Database Systems (TODS)*, vol. 13, no. 1, pp. 91-128, 1988.

[11] J. W. Song, K. Y. Whang, Y. K. Lee, and M. J. Lee, "Spatial join processing using corner transformation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 4, pp. 688-695, 1999.

[12] T. H. Kim, G. S. Kim, Y. C. Song, G. S. Ryu, B. J. Choi, and K. H. Park, "A color-based medicine bottle classification method robust to illumination variations," *Journal of Korean Institute of Intelligent Systems*, vol. 23, no. 1, pp. 57-64, 2013.

[13] K. P. Fishkin, "Applying color science to computer graphics," M.S. thesis, University of California, Berkeley, CA, 1982.

[14] H. Levkowitz and G. T. Herman, "GIHS: a generalized color model and its use for the representation of multiparameter medical images," in *Mathematics and Computer Science in Medical Imaging*. Heidelberg: Springer, 1988, pp. 389-399.

[15] D. H. Yeo and J. H. Lee, "Two-dimensional grouping index for efficient processing of XML filtering queries," *Journal of Information Technology and Architecture*, vol. 10, no. 1, pp. 123-135, 2013.

[16] H. P. Kriegel, "Query processing in spatial database systems," in *New Results and New Trends in Computer Science*. Heidelberg: Springer, 1991, pp. 172-191.

[17] J. H. Lee, Y. K. Lee, K. Y. Whang, and I. Y. Song, "A region splitting strategy for physical database design of multidimensional file organizations," in *Proceedings of the 23rd International Conference on Very Large Data Bases*, Athens, Greece, 1997, pp. 25-29.

Jung-A Lee https://orcid.org/0000-0001-8112-5232

She received M.S. degrees in School of Computer Science and Engineering from Daegu Catholic University in 2002. Currently, she is with the School of Information Technolodge Engineering from Daegu Catholic University as a PhD candidate. Her current research interests include query processing and big data.

Jong-Hak Lee https://orcid.org/0000-0001-8077-3262

He received B.S. degree in Computer Science from Kyungpook National University in 1982. He received M.S. and Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST) in 1984 and 1997, respectively. From March 1, 1984 to May 9, 1987, he was a researcher in the Research Institute of Gold Star Communications Co., Ltd. From May 10, 1987 to February 28, 1998, he was a senior researcher in the Research and Development Division, Korea Telecom. Since March 2, 1998, he has been a Professor in the School of Information Technology Engineering, Daegu Catholic University. His research interests include multidimensional file structure, database design, XML database, data warehouse, big data, database security, etc.