PERGAMON

Information Processing and Management 39 (2003) 267–285

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Collection fusion using Bayesian estimation of a linear regression model in image databases on the Web

Deok-Hwan Kim [a], Chin-Wan Chung [b,*]

[a] *Department of Information and Communication Engineering, Korea Advanced Institute of Science and Technology, 373-1, Kusong-dong, Yusong-gu, Taejon 305-701, South Korea*

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

Abstract

The collection fusion problem of image databases is concerned with retrieving relevant images by content based retrieval from image databases distributed on the Web. While there have been many studies about database selection and collection fusion for text databases, little research has been attempted for the case of image databases. Image databases on the Web have heterogeneous characteristics since they use different similarity measures and queries are processed depending on their own policies. Our previous study [Inf. Process. Lett. 75 (1–2) (2000) 35] provided three algorithms for this problem. In this paper, the metaserver selects image databases supporting similarity measures that are correlated with a global similarity measure, and then submits a query to them. And, we propose a new algorithm for this metaserver, which exploits a probabilistic technique using Bayesian estimation for a linear regression model. It outperforms the previous approach for diverse sizes of result sets for a query, and its improvement in effectiveness becomes especially large with small sizes of result sets. We also provide a virtual optimal algorithm to which our algorithm is compared. With extensive experiments we show the superiority of the Bayesian method over the others.

Keywords: Collection fusion; Bayesian model; Similarity search; Image database

1. Introduction

Along with the current growth of the Web environment, the access to image databases distributed on the Web has become an important research issue. This is especially so in application domains such as digital libraries, medical diagnostic systems, remote education, distributed

* Corresponding author.
E-mail addresses: dhkim@islab.kaist.ac.kr (D.-H. Kim), chungcw@islab.kaist.ac.kr (C.-W. Chung).

0306-4573/03/$ - see front matter
PII: S0306-4573(02)00051-1

publishing, and electronic commerce. The dramatically increasing number of image databases on the Web has led to a collection fusion problem.

Various approaches to collection fusion for retrieving text information have been attempted. An extensive survey of distributed information retrieval was provided by Callan (2000). Vorhees, Gupta, and Johnson-Laird (1994) proposed a learning based approach, which uses training queries to estimate the distribution of relevant objects from databases. Gravano and Garcia-Molina (1997) presented an analytic method for the first time to guarantee the retrieval of globally most similar documents from local databases. Meng et al. (1998) provided techniques that obtain the best threshold for a given local database. Fox and Shaw (1994) proposed a number of collection fusion methods using MIN, MAX and SUM operators. Lee (1997) performed experiments with Fox and Shaw's algorithms and observed that the best collection fusion was obtained when systems retrieved similar sets of relevant documents and dissimilar sets of non-relevant documents. Two models for metasearch, one based on democratic voting procedure and another based on Bayesian inference, were proposed by Aslam and Montague (2001). The Bayesian inference model requires training data.

Manmatha, Rath, and Feng (2001) showed that the score distributions for a given query could be modeled using an exponential distribution for the set of non-relevant documents and a normal distribution for the set of relevant documents. They tried to recover the relevant and non-relevant distributions to the output scores of local databases, when relevance information is not available, by fitting a mixture model. However, almost all the research for collection fusion was conducted on text databases whereas little has been done for image databases.

In the text databases, documents are generally represented by terms and frequencies of their occurrences or statistics derived from the frequencies. Various ranking algorithms and weighting methods such as *tf.idf* approaches and *cosine* function are used (Callan, 2000). Meanwhile, in the image databases, images can be represented by visual features such as color, texture, shape, etc. The distance between two feature vectors can be represented by several methods including the Euclidean distance function. Text retrieval uses more homogeneous features since the metaserver and a local database share a common feature value which is the term frequency, while image retrieval uses more heterogeneous features such as color, texture and shape. Moreover, even with the same color feature, the meaning of feature representations changes according to color spaces such as RGB and HSV. In the case of image retrieval, unlike text retrieval, the features used to compute the similarity value in the metaserver may be quite different from those in local image databases. This incurs the following important consequence.

While the collection fusion for text retrieval can be accomplished by transferring only feature values from local databases to the metaserver, that for image retrieval must transfer candidate images themselves from local databases to the metaserver. Subsequently, the metaserver extracts feature values from candidate images in terms of features it is using in order to compute the global similarity of candidate images. Since the efficiency of image transfer is much lower than that of text feature transfer, we must optimize the collection process of candidate images from local databases. Therefore, collection fusion methods for text retrieval are not applicable for image retrieval as is.

The metaserver is an agent that distributes user queries to local image databases, integrates result to fit user requirements, and also provides the illusion of a single database. To access distributed image databases, the metaserver is needed to integrate various resources and process

queries in a distributed manner. To do this, the metaserver will typically perform three main tasks: (1) decide which databases are relevant for evaluating a query (database selection), (2) send a query to selected databases using the available interfaces and query models (query translation), and (3) bring relevant images retrieved from these databases, and present them in a sorted order to the user (collection fusion). Among the above three tasks, (1) and (3) are related.

For a similarity query, an image database retrieves visual images similar to a query image using the similarity measure. Let $\mathrm{sim}_M(a,b)$ be the similarity function using a similarity measure M between two images a and b. We define 'relevant images' of a query q and a global threshold $\mathrm{GT} \in [0,1]$ as an image set $\{x|\mathrm{sim}_{\mathrm{global}}(q,x) > \mathrm{GT}\}$, where *global* is a global similarity measure. A user who issues a query through the metaserver wants to retrieve relevant images. The term 'global similarity measure' means the similarity measure of the metaserver and the term 'local similarity measure' means that of an image database.

In the database selection phase, the image database selection problem is to select relevant databases that have more images similar to the query image than others. We proposed a hybrid selectivity estimator (Kim, Lee, Lee, & Chung, 2000), estimating the result size of a query q from each image database, by using the sample images and the compressed histogram information (Lee, Kim, & Chung, 1999). The estimated result size of a query q issued to an image database db, $\mathrm{gnum}'(\mathrm{db}, q, \mathrm{GT})$, is calculated and is used for selecting relevant image databases.

In this paper, we focus on the collection fusion problem, which deals with how to retrieve relevant images for a query from distributed image databases that use different similarity measures. When a global similarity measure is completely different from a local similarity measure, for instance, the global similarity measure is using color whereas the local similarity measure is using texture, a user cannot get an appropriate result for a query. Therefore, a global similarity measure must be correlated with a local similarity measure. In this paper, we show that there exist some cases in which a degree of correlation between two similarity measures holds. And, the metaserver selects image databases supporting similarity measures that are correlated with a global similarity measure, and then submits the query to them.

To maximize the number of relevant images and minimize the number of irrelevant images in retrieved images from selected image databases, two heuristic algorithms and a probabilistic algorithm using classical regression were provided in Lee, Kim, Lee, Chung, and Cha (2000). However, their effectiveness will decrease when the sizes of result sets for a query become small. Therefore, we propose new collection fusion algorithms to overcome this disadvantage.

Our contributions in this paper are as follows:

(1) We provide a new collection fusion algorithm using the probabilistic estimator based on Bayesian regression.
(2) We refine three algorithms proposed in Lee et al. (2000) using the estimated result size, $\mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$, in the database selection phase.
(3) We provide a virtual optimal collection fusion (OPTCF) algorithm for an absolute comparison of the proposed algorithms.

Extensive experiments show that the new collection fusion algorithm achieves on average over 40% improvement in precision and recall against previous algorithms.

2. Objective of the collection fusion problem

The collection fusion problem for image databases is how to retrieve an optimal data set with maximum recall and precision given a constraint on the size of the retrieved set. We propose metasearch algorithms to determine in advance how many instances to retrieve from each database, in order to retrieve more images among more relevant databases and maximize the ratio of relevant images among retrieved images.

With $\mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$ which is the estimated result size of a query q for the ith image database db_i and is calculated during the database selection phase, we can estimate the total number of relevant images that a user wants to retrieve, $\sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$, when the number of selected image databases is M. If the metaserver retrieves exactly $\sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$ number of images from selected image databases, the recall will be less than 1 because there will be some irrelevant images in the retrieved images. Therefore the metaserver must get more than $\sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$, that is $c\sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$, images where c is a constant larger than or equal to 1.

More specifically, the formal definition is as follows: for a distributed similarity search of a given query q, let R_q^i be the set of relevant images in the ith image database and I_q^i be the set of irrelevant images in the ith image database. Then $R_q^i \cap I_q^i = \emptyset$ and $R_q^i \cup I_q^i = \{\text{all images in the } i\text{th image database}\}$. Let W_q^i be the set of images retrieved from the ith image database and $|S|$ be the number of elements of the set S. We have the constraint that the total number of retrieved images from image databases is fixed as $\sum_{i=1}^{M} |W_q^i| = c\sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$.

Objective: The objective of the collection fusion problem with this constraint is as follows:

(1) The ratio of retrieved images among relevant images should be maximized. That is, maximize $\sum_{i=1}^{M}(|R_q^i \cap W_q^i|)/|R_q^i|$ subject to the constraint $\sum_{i=1}^{M} |W_q^i| = c\sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$.
(2) The ratio of relevant images among retrieved images should be maximized. That is, maximize $\sum_{i=1}^{M}(|R_q^i \cap W_q^i|)/|W_q^i|$ subject to the constraint $\sum_{i=1}^{M} |W_q^i| = c\sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$.

(1) is to maximize the recall and (2) is to maximize the precision. The precision, however, will be decreased, as recall increases. In practice, the objectives, (1) and (2), are conflicting with each other. Therefore, it can be an issue to adjust the tradeoff between (1) and (2). We will use $P \times R$ as the combined measure optimizing the precision and the recall.

3. Heterogeneous similarity measures

Color histograms are popular methods to represent the distribution of colors in images where each histogram bin represents a color in one of various color spaces (RGB, YCbCr, HSV, etc). For the average color features extracted from the color histogram in RGB or YCbCr space, *the Euclidean* distance function is used, while the *angular* distance function is used for HSV space (Kim et al., 2000). The distance value is converted into the similarity value using the feature normalization technique suggested in MARS (Ortega, Chakrababarti, Porkaew, & Mehrotra, 1998).

Since databases on the Web are heterogeneous, their feature extracting methods and distance functions may be different and so might be the similarity measures, although their attributes used

in the similarity search are the same. Therefore the local similarity value between a query image and an image is different from the global similarity value between them.

3.1. Relationship between similarity measures

We investigate the relationship between two similarity measures when different features and similarity measures are used in image databases. The following example illustrates this:

Example 1. The metaserver and the image database support the similarity search using the color attribute. The metaserver extracts average color features from the color histogram in the HSV color space while the image database extracts them in the RGB color space. The metaserver measures its similarity value against a query image as the image database does. Fig. 1(a) shows the scatter diagram of global similarity values (y coordinate) and local similarity values (x coordinate) for 4716 pairs of images selected from a set of 4716 images. Each of the 4716 images is selected as the first element of a pair, and the second element of the pair is selected arbitrarily among the 4716 images. In this case, the diagram shows that the shape of a graph is a straight line.

Example 2. The metaserver and the image database support the similarity search using the texture attribute. The metaserver extracts texture features from the second moment of the color histogram in the HSV color space, while the image database extracts them in the RGB color space. The metaserver measures its similarity value against a query image as the image database does. The scatter diagram of the global similarity values (y coordinate) and the local similarity values (x coordinate) for 4716 images is shown in Fig. 1(b). The diagram shows the shape of a straight line.

Example 3. In Fig. 2, the similarity values of the y coordinate are obtained using the average color from the color histogram in the HSV color space while those of the x coordinate are obtained using the texture extracted from the second moment of the color histogram in the RGB color space. Contrary to previous cases, the scatter diagram does not show any relationship between two similarity measures with different attributes.

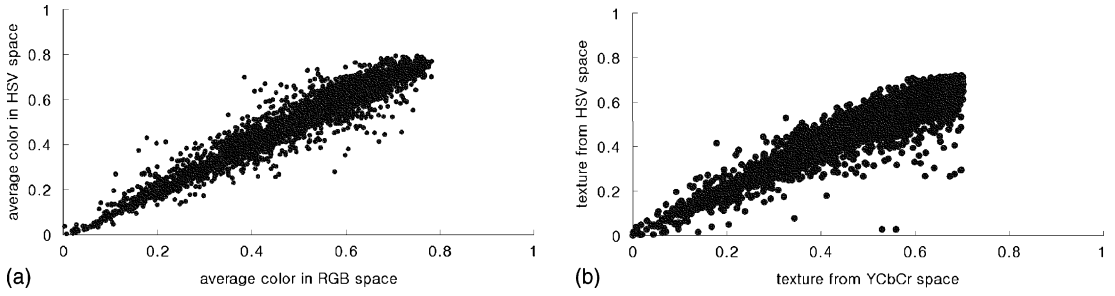

Fig. 1. Scatter-diagram of correlated similarity measures. Scatter-diagram of similarity values for different color features (a), texture features (b).

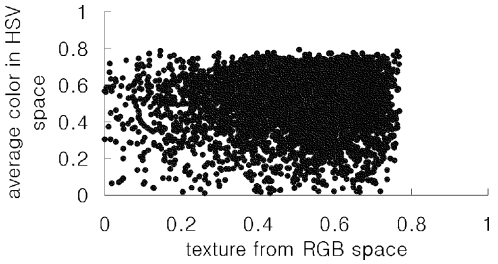

Fig. 2. Scatter-diagram of uncorrelated similarity measures.

Since we cannot prove that two different similarity measures with the same attribute have a linear relationship, we have done extensive experiments in various cases that show a linear relationship. For parsimony, please see our previous work (Kim et al., 2000).

Observation 1. Although similarity measures are different between the metaserver and image databases, the scatter diagrams of similarity values of some pairs of similarity measures show the shape of a straight line.

For any two similarity measures, if they satisfy the linear relationship, we can use that property for collection fusion.

4. Retrieval of globally most similar images

In this section, we propose metasearch algorithms to determine in advance how many images to retrieve from each database. The metasearch algorithms must result in high recall and high precision to achieve the objectives of the collection fusion problem stated in Section 2. For this purpose, we define the degree of importance and the local precision.

Definition 1. The degree of importance DI_q^i and the local precision LP_q^i of the ith image database db_i for a query q are defined as follows:

$$\mathrm{DI}_q^i = \frac{\text{number of relevant but not retrieved images for } q \text{ in } \mathrm{db}_i}{\text{total number of relevant images for } q \text{ in all databases}} \tag{1}$$

$$\mathrm{LP}_q^i = \frac{\text{number of relevant and retrieved images for } q \text{ in } \mathrm{db}_i}{\text{number of images retrieved for } q \text{ in } \mathrm{db}_i} \tag{2}$$

If the degree of importance of an image database is high, we can get more relevant images from that database than others. If the local precision of an image database is high, we can reduce the ratio of irrelevant images retrieved from that database. In a real situation, because we do not know their exact values, we calculate their values approximately using various estimators.

The OPTCF algorithm using dynamic programming is presented to compare with the proposed ones. Table 1 shows parameters to be used in proposed algorithms.

Table 1
Symbols used in the proposed algorithms

Symbol	Description
q	A query image
c	Multiplication ratio to increase recall
M	Number of selected image databases
a	Number of steps for incremental retrievals
db_i	ith image database
p_i	Number of images retrieved from db_i in one step
r	Number of images retrieved in one step from selected databases (a predetermined number)
y	Global similarity coordinate
x	Local similarity coordinate
d_y	Half size of $100(1-\delta)\%$ confidence interval on the y coordinate
GT	Global threshold given by the user
lt	Local threshold, the least of local similarity values of partially retrieved images
gt^{u}, gt^{m}, gt^{l}	Three different global thresholds corresponding to *lt*
T	Type of the global threshold being used: *upper*, *middle*, *lower* are respectively indicated by gt^{u}, gt^{m}, gt^{l}
α, β	The regression coefficients
DI_q^i	Degree of importance of db_i for a query q
LP_q^i	Local precision of db_i for a query q

4.1. Collection fusion using heuristics

Our heuristic is as follows: if an image database retrieves more images relevant to a query compared to other databases in any given step, then it will continue to retrieve more relevant images later on. Therefore, we suggest heuristic estimators that approximately calculate how many relevant images can be retrieved from each image database for every step of the algorithm.

At this point, it will be useful to discuss the heuristic algorithm and two heuristic estimators in greater detail.

Algorithm. Heuristic_Collection_Fusion$(q, c, \mathrm{GT}, a, \mathrm{db}_1, \ldots, \mathrm{db}_M)$

(1) Query image q is sent to selected databases $\mathrm{db}_1, \ldots, \mathrm{db}_M$.
(2) For each db_i, initialize $p_i = \left[(c \sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT}))/aM\right]$
(3) While $\left(\text{total number of retrieved images} < c \sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})\right)$
(4) for each db_i, get_more_objects(q, p_i, db_i)
(5) let result_i be the set of images that are retrieved from db_i.
(6) merge_results$(\mathrm{result}_1, \ldots, \mathrm{result}_M)$
(7) for each db_i, recalculate p_i using *heuristic estimator* of db_i
(8) End While

get_more_objects(q, p_i, db_i) requests db_i to get p_i more images similar to query q by using a local similarity measure of db_i as described in Seidl and Kriegel (1998). merge_results$(\mathrm{result}_1, \ldots, \mathrm{result}_M)$ merges and ranks results retrieved from selected image databases by using a global similarity measure. If all image databases have the same degree of importance, it is sufficient for the metaserver to get $p_i = [(c \sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT}))/M]$ images only once from each image

database, where [ ] is the rounding operator. However, the degree of importance is different for all db_i and cannot be known in advance. Therefore we must approximate them repeatedly. If the repetition is a, the initial value of p_i is given by $p_i = [(c\sum_{i=1}^{M} \text{gnum}'(db_i, q, GT))/aM]$. Step (7) of the above algorithm assigns a large value to p_i of the image database whose heuristic estimator is high in order to increase the recall and the precision. The strategy for assigning p_i is decided by a heuristic estimator.

4.1.1. Average ranking heuristic estimator

It is difficult to estimate the recall and precision separately using information from images retrieved from selected image databases because of insufficient information. We suggest a heuristic estimator α_i for the combined measure of recall and precision. The metaserver gets more images from an image database with a higher value of α_i and less images from one with a lower value. α_i is defined as follows: $\alpha_i = L_i / \sum_{j=1}^{L_i} \text{Rank}_{ij}$. α_i is the reciprocal of the average of merged ranks of images retrieved from the ith image database, where the merged rank is the rank based on global similarity among images retrieved from all image databases. Rank_{ij} is the merged rank of the jth image retrieved from the ith image database. L_i is the number of images retrieved in the last retrieval from the ith image database. p_i of the heuristic algorithm is given as follows:

$$p_i = \left[\frac{c\sum_{i=1}^{M} \text{gnum}'(db_i, q, GT)}{a} \frac{\alpha_i}{\alpha_1 + \cdots + \alpha_M} \right] \tag{3}$$

The average ranking heuristic estimator can be considered a good collection fusion method since by using the method, the more highly an image database is ranked, the more relevant images it may have.

4.1.2. Average global similarity heuristic estimator

This is similar to the average ranking heuristic. The rank has an integer value which has a uniform difference between adjacent ranked images. However, as previous research has shown (Lee, 1997), the rank_similarity curve is not always a straight line, that is, the similarity difference between adjacent images may not be uniform. So, the heuristic estimator β_i is defined as follows: $\beta_i = (\sum_{j=1}^{L_i} \text{Global\_Similarity}_{ij})/L_i$. β_i is the average similarity of the images retrieved from the ith image database. p_i of the heuristic algorithm is given as follows:

$$p_i = \left[\frac{c\sum_{i=1}^{M} \text{gnum}'(db_i, q, GT)}{a} \frac{\beta_i}{\beta_1 + \cdots + \beta_M} \right] \tag{4}$$

4.2. Collection fusion using ordinary least square

The proposed metasearch algorithm estimates p_i by using the ordinary least square (OLS) method and multi-step retrieval. In a real situation, since we do not know the exact values of degree of importance and local precision for each database, we use the estimated degree of importance and incremental local precision to determine the number of images p_i to be retrieved from an image database db_i in each step.

Definition 2. The estimated degree of importance EDI_q^i and the incremental local precision ILP_q^i of the ith image database db_i for a query q are defined as follows:

$$\mathrm{EDI}_q^i = \frac{\mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT}) - |R_{q,K}^i \cap W_{q,K}^i|}{\sum_{j=1}^{M} \mathrm{gnum}'(\mathrm{db}_j, q, \mathrm{GT})} \times w_i \tag{5}$$

$$\mathrm{ILP}_q^i = \frac{|R_{q,k}^i \cap W_{q,k}^i|}{|W_{q,k}^i|} \times \frac{1/d_y^i}{\sum_{j=1}^{M} 1/d_y^j} \tag{6}$$

where $R_{q,k}^i$ is the set of relevant images retrieved from db_i in the kth step and $W_{q,k}^i$ is the set of retrieved images from db_i in the kth step. $R_{q,K}^i$ and $W_{q,K}^i$ are those from the first step to the kth step. w_i is a weight obtained from all gt_i values. The range of w_i is from zero to $M-1$. Zero is assigned to w_i for lowest gt_i and $M-1$ is assigned to w_i for highest gt_i. The smaller the confidence interval, d_y^i, of the regression line of db_i, the higher the ILP_q^i value will be since less irrelevant images are retrieved by the given local threshold. In order to maximize $P \times R$, the combined measure of precision and recall, the metaserver gets more images from an image database with a high value of the combined measure, $\mathrm{EDI}_q^i \times \mathrm{ILP}_q^i$. Then the number of images to be partially retrieved from db_i is:

$$p_i = \left[r \frac{\mathrm{EDI}_q^i \times \mathrm{ILP}_q^i}{\sum_{j=1}^{M} \mathrm{EDI}_q^j \times \mathrm{ILP}_q^j} \right] \tag{7}$$

The collection fusion algorithm using the OLS is as follows:

Algorithm. OLS_Collection_Fusion$(q, c, d, \mathrm{GT}, a, \mathrm{db}_1, \ldots, \mathrm{db}_M, T)$

(1) for each db_i, $r \leftarrow [(c \sum_{j=1}^{M} \mathrm{gnum}'(\mathrm{db}_j, q, \mathrm{GT}))/a]$, $p_i \leftarrow r/M$, total_no $\leftarrow 0$, $i = 1, \ldots, M$
(2) for each db_i, get_more_objects(q, p_i, db_i) and let result_i be the set of images that are retrieved from db_i and merge_results $(\mathrm{result}_1, \mathrm{result}_2, \ldots, \mathrm{result}_M)$
(3) while (for each db_i, $i = 1, \ldots, M$, total_no $< c \sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$)
(4) analyze images from db_i using the OLS method and obtain equation $\hat{y}_i = \hat{\alpha}_i + \hat{\beta}_i x_i$ and obtain gt_i where gt_i is one of $\mathrm{gt}_i^{\mathrm{u}}, \mathrm{gt}_i^{\mathrm{m}}, \mathrm{gt}_i^{\mathrm{l}}$ according to T.
(5) calculate weight w_i by ranking all $\mathrm{gt}_i, i = 1, \ldots, M$.
(6) For each db_i, calculate $\mathrm{EDI}_q^i, \mathrm{ILP}_q^i$.
(7) if $(c \sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT}) - \mathrm{total\_no}) \geqslant r$
(8) then $p_i = [r(\mathrm{EDI}_q^i \times \mathrm{ILP}_q^i)/(\sum_{j=1}^{M} \mathrm{EDI}_q^j \times \mathrm{ILP}_q^j)]$
(9) else $p_i = [(c \sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT}) - \mathrm{total\_no})(\mathrm{EDI}_q^i \times \mathrm{ILP}_q^i)/(\sum_{j=1}^{M} \mathrm{EDI}_q^j \times \mathrm{ILP}_q^j)]$
(10) for each db_i, get_more_objects(q, p_i, db_i)
(11) merge_results$(\mathrm{result}_1, \mathrm{result}_2, \ldots, \mathrm{result}_M)$
(12) for each db_i,total_no $\leftarrow$ total_no $+ p_i$
(13) end while

The algorithm initially gets an equal number of images from each image database and obtains a regression line, $\hat{y} = \hat{\alpha} + \hat{\beta}x$, by using global similarity values and local similarity values of retrieved images and the global threshold (gt) corresponding to the local threshold. The lowest of local

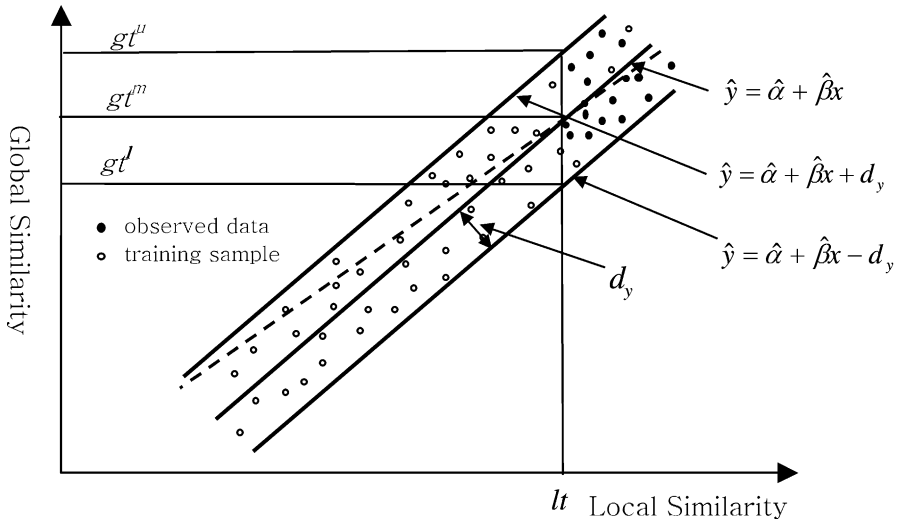

Fig. 3. The OLS (- - -) vs. BLS (—).

similarity values of partially retrieved images becomes the local threshold (lt). The mean square error (MSE) of the initial regression line is high and the gt is not accurate since the metaserver has insufficient information for each database. Therefore, it can reduce MSE and get accurate gt by progressive retrieval. As shown in Section 3.1, the size of the confidence interval of a regression line for each database may be different even though global similarity values and local similarity values are correlated.

This algorithm uses three different global thresholds, gt^l, gt^m, gt^u corresponding to the local threshold. In Fig. 3, d_y is half the size of the confidence interval of $\hat{y}$. gt^m is the y coordinate value of the intersection point of $\hat{y} = \hat{\alpha} + \hat{\beta}x$ and $x = lt$. gt^l is the intersection point of $\hat{y} = \hat{\alpha} + \hat{\beta}x - d_y$ and $x = lt$. gt^u is the intersection point of $\hat{y} = \hat{\alpha} + \hat{\beta}x + d_y$ and $x = lt$. T indicates the type of the global threshold, one of gt^u, gt^m, gt^l. In the case of gt^u, the recall of the result is high and the precision is low. In the case of gt^m, the recall is less than the case of gt^u while the precision is higher. In the case of gt^l, the recall is the lowest among the three cases and the precision is the highest.

Since the estimated line and gt may not be accurate because of insufficient information, the algorithm gets only p_i images and repeats again. With this repetition, it refines the regression line and finally gets the exact regression line.

4.3. Collection fusion algorithm using Bayesian least square

In this section, in order to make the retrieval more efficient, we suggest a new method using the Bayesian least square (BLS) for linear regression. For this method, we assume that similarity values between the images retrieved from an image database and a query image follow the Gaussian distribution, as shown in (Ortega et al., 1998).

The BLS method can estimate the posterior parameters (α, β) from prior parameters and observed data, if the distribution of prior parameters and the likelihood of observed data follow the

Gaussian distributions (Hamilton, 1996, Chap. 12). Using the Bayesian method, we can refine the estimated parameters more accurately as the steps progress. That is, the parameters estimated using samples are refined with newly retrieved images, and so on.

Let $(x_{01}, x_{02}, \ldots, x_{0n})$ be local similarity values between sample images and the query image, let $(y_{01}, y_{02}, \ldots, y_{0n})$ be global ones, let $(x_{t1}, x_{t2}, \ldots, x_{tp})$, $(y_{t1}, y_{t2}, \ldots, y_{tp})$, $t \geqslant 1$ be local similarity values and global ones between the query image and images retrieved at the tth step. We define X_t, Y_t as follows:

$$X_t = \begin{pmatrix} 1 & x_{t1} \\ \vdots & \vdots \\ 1 & x_{tp} \end{pmatrix}, \quad Y_t = \begin{pmatrix} y_{t1} \\ \vdots \\ y_{tp} \end{pmatrix} \tag{8}$$

Then, the parameters at the tth step are estimated using the following equation (Hamilton, 1996, Chap. 12):

$$\begin{pmatrix} \hat{\alpha}_t \\ \hat{\beta}_t \end{pmatrix} = \left(M_{t-1}^{-1} + X_t' X_t\right)^{-1} \left(M_{t-1}^{-1} \begin{pmatrix} \hat{\alpha}_{t-1} \\ \hat{\beta}_{t-1} \end{pmatrix} + X_t' Y_t\right) \tag{9}$$

$\hat{\alpha}_t$, $\hat{\beta}_t$ are the estimated values of α, β at the tth step. And the confidence of the estimate is defined as $M_0 = (X_0' X_0)^{-1}$, $M_t = (M_{t-1}^{-1} + X_t' X_t)^{-1}$, where X' is the transpose of X.

The advantages of the Bayesian method are as follows: (1) Even though the total number of retrieved images, $c \sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$, is small, it can estimate the parameters more accurately, i.e., its MSE is small, since it uses the dynamic prior. (2) It has less of a burden to estimate the parameters since it uses only newly retrieved data at each step. This algorithm estimates the linear equation using the Bayesian method and the global threshold. And, it selects the image database that has the largest global threshold and retrieves predefined p images from the selected database next time.

Let M be the number of selected image databases and a be the number of repeating step. The time complexity of the metasearch algorithm using a linear regression is $\mathrm{O}(Maz)$ when the computing time of a regression line using global similarity values and local similarity values of retrieved images in each step is z. Especially, z is obtained as $\mathrm{O}(cn^3)$, by the computing time of the matrix multiplication and the inverse matrix, where n is the number of data and c is a constant. Let $p_{i,t}$ be the number of images from the ith database in the tth step. Then the sum of the number of images retrieved from the first step to the tth step is $p_{i,1} + p_{i,2} + \cdots + p_{i,t}$. The number of images used in the algorithm using OLS is $p_{i,1} + p_{i,2} + \cdots + p_{i,t}$ while that in the algorithm using BLS is only $p_{i,t}$. Therefore, the algorithm using BLS is more efficient.

Algorithm. BLS_Collection_Fusion$(q, c, \mathrm{GT}, p, \mathrm{db}_1, \ldots, \mathrm{db}_M, T)$

/* we assume that α_0, β_0 and M_0 for each image database and a query image have already been calculated using sample images at the database selection phase.*/

(1) For each db_i, $i = 1, \ldots, M$,
   - get_more_objects(q, p, db_i).
   - Calculate $\hat{\alpha}_1$, $\hat{\beta}_1$ using Eq. (9) and obtain gt corresponding to the least value of local similarity values of retrieved images where gt is one of gt^{u}, gt^{m}, gt^{l} according to T.

(2) select the db_{l}, which has the largest gt among all image databases and let its gt be gt_{l}.

(3) if (the total number of retrieved images is greater than gt_l) $\geqslant \sum_{i=1}^{M} \text{gnum}'(db_i, q, GT)$ or (the total number of retrieved images) $\geqslant c \sum_{i=1}^{M} \text{gnum}'(db_i, q, GT)$ then stop.
(4) merge_results and rank them according to the global similarity values.
(5) get_more_objects(q, p, db_l).
(6) For db_l, calculate $\hat{\alpha}_t$, $\hat{\beta}_t$ using Equation (9) and obtain gt where gt is one of gt^u, gt^m, gt^l according to T.
(7) go to step (2).

4.4. Optimal collection fusion algorithm

For designing a collection fusion algorithm, we need a criterion in order to characterize the algorithm as efficient or inefficient. As a base-line, we will define an OPTCF algorithm even though it cannot be implemented in the real world. Past research has shown that a distributed search is less effective than a centralized search (Xu & Callan, 1998). Therefore, the idea is to make it be as effective as a centralized search and to guarantee the best possible performance in a distributed environment.

In the distributed computing environment, the transmission cost is the most expensive item. When the total number of images to be retrieved from several image databases is fixed, we can define the cost of the algorithm to be the ratio of the number of irrelevant images to the total number of retrieved images. Now, we will define the cost function.

Definition 3 (*Cost function*). The cost function to retrieve k images from db_i for a query image q, $f_i(k, q)$, is defined as the number of the irrelevant images among k ones retrieved from db_i.

The cost function for db_i is illustrated in Fig. 4. The x coordinate represents k, i.e., the number of images retrieved from db_i for query q and the y coordinate represents $f_i(k, q)$. (1) shows the ideal case in which there are no irrelevant images. In fact, it is unrealistic except for the case when the global similarity measure is the same as the local similarity measure. (2) and (3) are real cases. Values of y are always smaller than those of x, i.e., it always lays below the line $y = x$.

If we assume that we already know the cost functions of all image databases for query q, we can define an OPTCF algorithm using a dynamic programming recurrence relation.

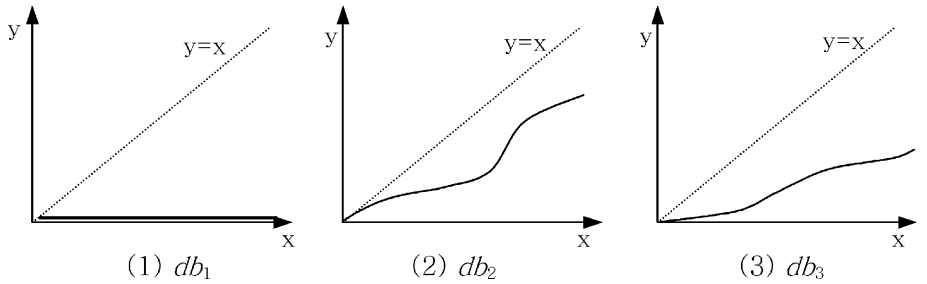

Fig. 4. Cost functions to retrieve k images from image databases.

Definition 4 (*Minimum cost function*). When we retrieve n number of images from image databases $\mathrm{db}_i, \ldots, \mathrm{db}_j$, for a query q, the minimum cost function is defined as follows:

$$F_{i,j}(n,q) = \begin{cases} f_i(n,q), & \text{when } i = j \\ \min_{0 \leqslant k \leqslant n}(f_i(k,q) + F_{i+1,j}(n-k,q)), & \text{when } i < j \end{cases} \tag{10}$$

Algorithm. Optimal collection fusion algorithm

- (step 1) For each db_i, find n_i the number of images to be retrieved from it using the minimum cost function $F_{1,M}(c\sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT}), q)$.
- (step 2) For each db_i, retrieve n_i number of images for query q.

5. Experiments and performance evaluation

5.1. Test collection

The test data consists of 14,624 images, which have been used in QBIC and WALRUS systems. We constructed three image databases using the clustering method. For the clustered allocation, the average color features in the RGB color space are used. Similar images are likely to be allocated to the same image database. Clusters are generated with centers randomly distributed. Each image database contains 4–5 clusters. About 60% of data are allocated to clusters, while the rest are distributed randomly.

Each database uses a different feature extraction method and distance function. To acquire visual features that characterize images, we extracted the average color from various color spaces using a color histogram method (Kim et al., 2000). For each image, the average color (μ_1, μ_2, μ_3) is used to represent the average intensity of each color component. In our experiments, we use the average color for 2×1 sub-images in HSV color space as the features of the metaserver. Table 2 shows feature extraction methods and database sizes for all image databases.

5.2. Experiments and results

The goal of the experiment is to evaluate the retrieval effectiveness of the proposed algorithms. For each test, we issue 10 queries using randomly selected images, and take an average of their results. In order to show the preciseness of the regression line of partially retrieved images, we present experimental results in Table 3. These indicate that the partial results approach the final

Table 2
Test collections

Collection	Feature description	Size
1	Average color for 2×2 sub-images in HSV color space	5037
2	Average color for 2×2 sub-images in RGB color space	5045
3	Average color for 2×2 sub-images in YCbCr color space	4550

Table 3
The preciseness of the regression line of partly retrieved images

# of retrieved images	MSE	r^2	α	β
OLS				
12	1.980838	0.539438	−1.26618	4.512518
24	1.526845	0.550958	−1.15815	4.181785
36	1.032742	0.570337	−1.00611	3.741204
48	0.277638	0.651683	−0.60839	2.655687
60	0.00223	0.866689	−0.056	1.069319
BLS				
12	0.00496	0.838488	0.04342	0.953364
24	0.00484	0.800292	0.93711	0.894193
36	0.00352	0.853566	−0.01633	1.023008
48	0.00151	0.905731	−0.08777	1.106338
60	0.00092	0.914253	0.004104	1.013262
All	0.00039	0.98571	0.009186	1.037139

MSE is (residual sum of squares)/(number of retrieved images).

results gradually and the preciseness of BLS is better than that of OLS. For the BLS method, we used 30 training sample images per each image database.

The Spearman rank correlation coefficient (Moroney, 1951) represents the rank difference between the global similarity values and the local similarity values for a query image and database images. Two rankings are identical when the correlation coefficient is 1 and they are uncorrelated when the coefficient is 0. Fig. 5 shows the number of retrieved images at which the ranking of images using the local similarity measure begins to match the ranking of the same images using the global similarity measure. If an image database uses a more heterogeneous similarity measure than that of the metaserver, the coefficient converges more slowly. Then the effectiveness of the

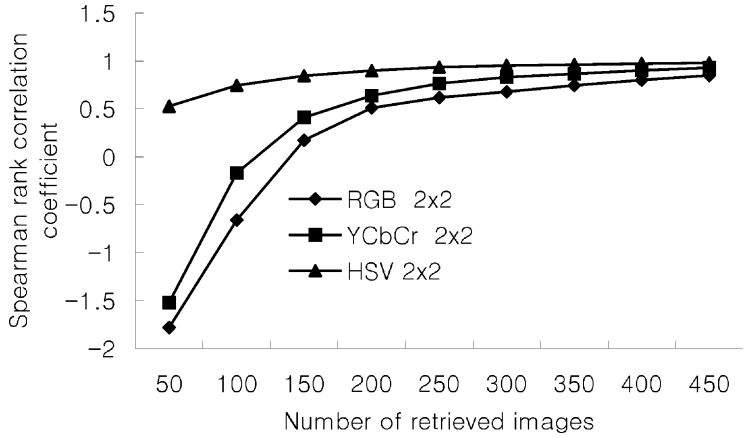

Fig. 5. Spearman rank correlation coefficient between local similarity values and global similarity values.

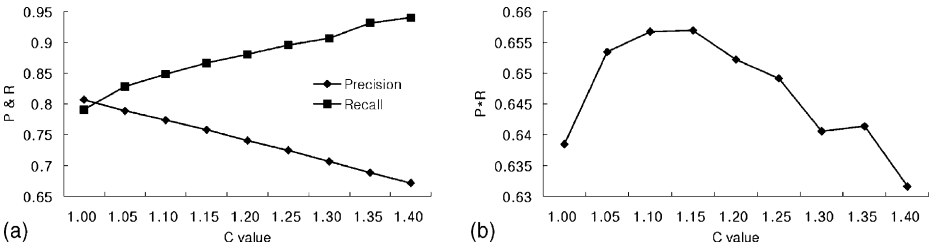

Fig. 6. Precision and recall with c values in the clustered distribution over image databases. (a) P & R, (b) $P \times R$.

collection fusion algorithm may decrease since the rank difference becomes large when the number of retrieved images is small.

We also observe the influence of the c value on the number of images fetched from image databases. Since the proposed algorithms have the constraint that the total number of images retrieved from selected databases is fixed as $c\mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$, it is important to choose an appropriate value for c. We use a $P \times R$ graph (see Fig. 6) to determine an appropriate c value. The recall has a tradeoff relation to the precision. The precision will be decreased, as recall increases, and vice versa. Therefore, we use $P \times R$ as a combined measure optimizing the precision and the recall. When the c value varies from 1.00 to 1.40—where $c = 1.00$ means just $\sum_{i=1}^{M} \mathrm{gnum}'(\mathrm{db}_i, q, \mathrm{GT})$ images are fetched from image databases, while $1.40 \sum_{i=1}^{M} \mathrm{gnum}' \times (\mathrm{db}_i, q, \mathrm{GT})$ images are fetched for $c = 1.40$—the corresponding precision and recall is changed as shown in Fig. 6. A point close to $c = 1.15$ indicates the highest $P \times R$ value, the combination measure of precision and recall, when the collection fusion algorithm using the BLS is used. We can observe that the precision decreases as the c value increases and the recall moves in the opposite direction.

To justify the assumption suggested in Section 4.3, we perform an extra experiment. As shown in Example 1, we use 4716 images as a test set. We compute their similarity values between the images and a query image by using color feature and texture feature and show their similarity distributions in Fig. 7. It shows a Gaussian distribution.

Our experiments about *collection fusion* include the comparison of five proposed algorithms: the average ranking heuristic algorithm (*Alpha*), the average global similarity heuristic algorithm (*Beta*), the algorithm using the OLS, the algorithm using the BLS, and the optimal algorithm

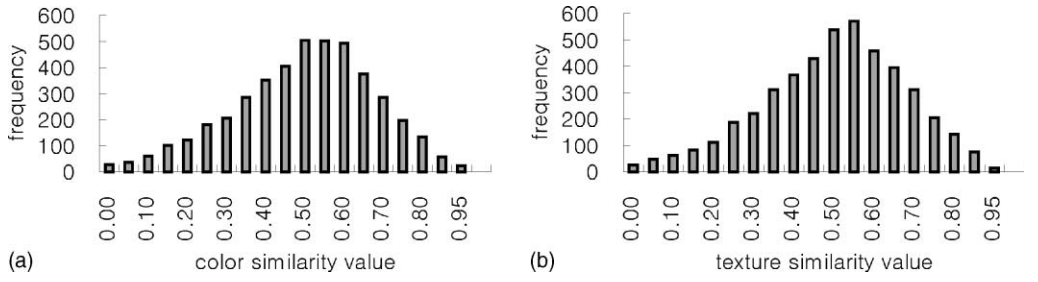

Fig. 7. Similarity distributions using color and texture features: (a) color similarity distribution, (b) texture similarity distribution.

(*Optimal*). The graphs of the precision and the recall for the proposed algorithms are summarized in Fig. 8(a)–(f).

When the number of retrieved images is large, the effectiveness of BLS and OLS algorithms are better than those of the heuristic algorithms (Alpha, Beta) since the algorithms using linear regression reflect the clustering effect of data distribution well. However, the effectiveness of OLS

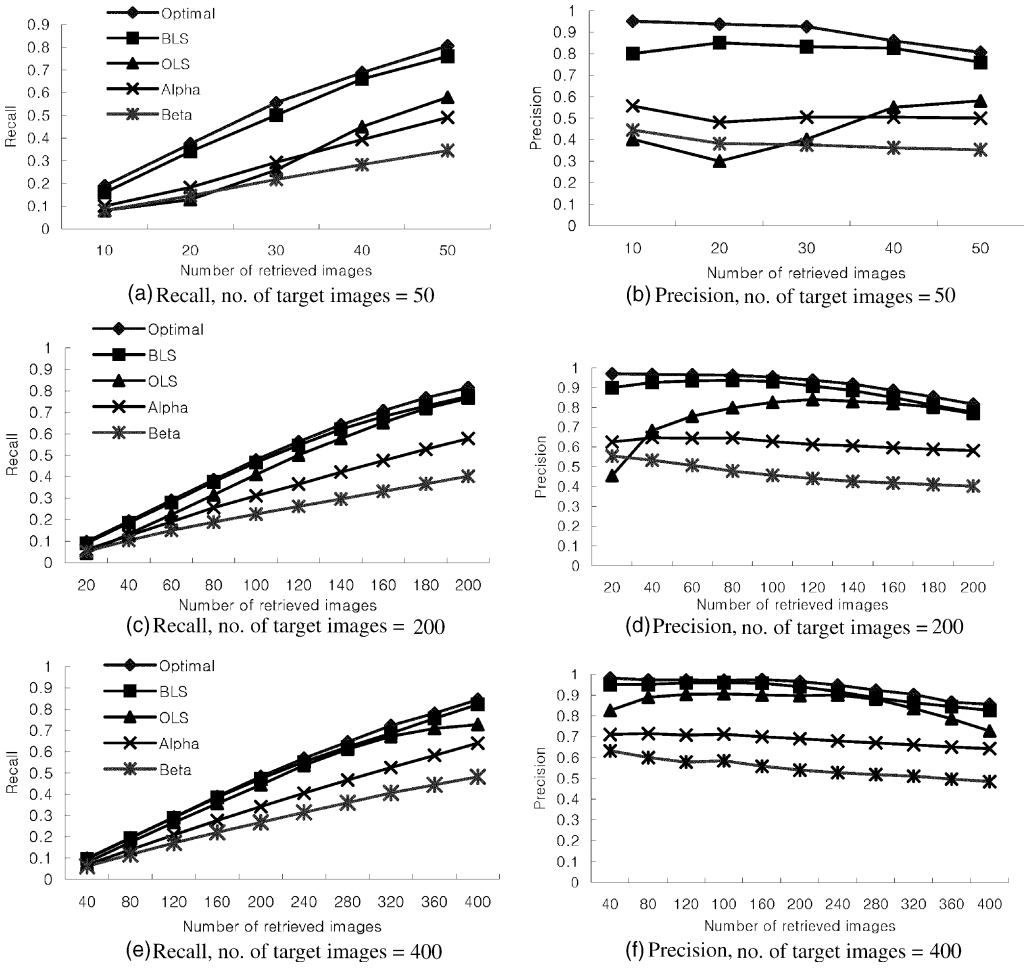

Fig. 8. Precision and recall.

Table 4
The average improvement (%) of recall and precision for *Optimal* and *BLS* algorithms against *OLS* algorithm

No of target images	50		100		200		400	
	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Optimal	107.03	115.26	53.38	57.61	35.72	32.16	10.30	9.19
BLS	87.07	94.54	46.16	46.82	28.37	25.19	7.65	6.08

(a) 14 result images for a green color query image

(b) 14 result images for a red color query image

Fig. 9. Query results of the algorithm using the Bayesian model.

algorithm is below those of the heuristic algorithms when the number of retrieved images is less than 50. The effectiveness of BLS algorithm is close to that of the optimal algorithm regardless of the number of retrieved images. The effectiveness of OLS algorithm is little different from that of the previous algorithm using classical regression, which was proposed in one of our earlier works (Lee et al., 2000). Therefore, we use the OLS algorithm to measure the improvement of effectiveness of the proposed BLS and optimal algorithms. Table 4 shows that the proposed BLS algorithm achieves about 42.31% overall improvement in recall and 43.16% overall improvement in precision against the OLS algorithm. It also shows 87.07% average improvement of recall and 94.54% average improvement of precision, that is the peak improvement against the OLS algorithm, when the number of target images is 50. That is, the proposed BLS algorithm keeps high precision and recall even though the Spearman rank correlation coefficient is low. Therefore, the algorithm using the BLS has greater potential for practical usage.

Fig. 9 presents 14 query results using the BLS algorithm for a green color query image and those for a red color query image, respectively. The numeric values represent a global similarity value and a local similarity value for a given query.

6. Conclusion

In this paper, we proposed a new collection fusion algorithm using BLS. This algorithm has an advantage as it can accurately estimate the regression line and the global threshold using the Bayesian model. Extensive experiments with a large number of real image data show that the algorithm using the Bayesian model has better effectiveness than the others, especially with small sizes of result sets for a query. The overhead of this algorithm is that it uses the sample images as prior information. However, the metaserver has little added burden since it fetches the sample images in the preprocessing phase and the sample size is small. Since the image retrieval applications on the Web generally require small sizes of the result sets for a query, the proposed new algorithm can be put to much greater practical usage.

In addition, an OPTCF algorithm, which cannot be implemented in the real world, is presented to be used for comparison with other algorithms. The experiments also show that the effectiveness of the algorithm using the Bayesian model is close to that of the optimal algorithm.

As future work, we plan to apply user relevance feedback to the collection fusion.

Acknowledgements

This work was supported by Korea Research Foundation Grant (KRF-2000-041-E00262). We would like to thank Dr. Gabriella Pasi, an editor, for helpful instructions and anonymous reviewers for valuable comments. We also wish to thank Dr. Ju-Hong Lee for useful discussions.

References

Aslam, J., & Montague, M. (2001). Models for metasearch. In *Proceedings of the 24th ACM SIGIR conference on research and development in information retrieval. New Orleans, Louisiana* (pp. 276–284).

Callan, J. (2000). Distributed information retrieval, chapter. In W. B. Croft (Ed.), *Advances in information retrieval* (pp. 127–150). Dordrecht: Kluwer Academic Publishers.

Fox, E., & Shaw, J. (1994). Combination of multiple searches. In *Proceedings of the 2nd text retrieval conference (TREC-2). Gaithersburg, Maryland* (pp. 243–252).

Gravano, L., & Garcia-Molina, H. (1997). Merging ranks from heterogeneous internet sources. In *Proceedings of the international conference on very large data bases. Athens, Greece* (pp. 14–25).

Hamilton, J. D. (1996). Bayesian analysis. In *Time series analysis* (pp. 354–358). Princeton, New Jersey: Princeton University Press.

Kim, D. H., Lee, J. H., Lee, S. L., & Chung, C. W. (2000). Heterogeneous multimedia database selection on the Web. Technical reports, CS/TR-2000-147, Korea Advanced Institute of Science and Technology. Available: http://cs.kaist.ac.kr/library/tr.

Lee, J. H. (1997). Analyses of multiple evidence combination. In *Proceedings of the 20th ACM SIGIR conference on research and development in information retrieval. Philadelphia, Pennsylvania* (pp. 267–276).

Lee, J. H., Kim, D. H., & Chung, C. W. (1999). Multi-dimensional selectivity estimation using compressed histogram information. In *Proceedings of the ACM SIGMOD international conference on management of data. Philadelphia, Pennsylvania* (pp. 205–214).

Lee, J. H., Kim, D. H., Lee, S. Y., Chung, C. W., & Cha, G. H (2000). Distributed similarity search algorithm in distributed heterogeneous multimedia databases. *Information Processing Letters, 75*(1–2), 35–42.

Manmatha, R., Rath, T., & Feng, F. (2001). Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th ACM SIGIR conference on research and development in information retrieval. New Orleans, Louisiana* (pp. 267–275).

Meng, W., Liu, K. L., Yu, C., Wang, X., Chang, Y., & Rishe, N. (1998). Determining text databases to search in the internet. In *Proceedings of the international conference on very large data bases. New York City, New York* (pp. 14–25).

Moroney, M. J. (Ed.). (1951). *Facts from figures*. Baltimore: Penguin.

Ortega, M., Chakrababarti, K., Porkaew, K., & Mehrotra, S. (1998). Supporting ranked boolean similarity queries in MARS. *IEEE Transactions on Knowledge and Data Engineering, 10*(6), 905–925.

Seidl, T., & Kriegel, H. (1998). Optimal multi-step *k*-nearest neighbor search. In *Proceedings of the ACM SIGMOD international conference on management of data. Seattle, Washington* (pp. 154–165).

Vorhees, E., Gupta, N., & Johnson-Laird, B. (1994). The collection fusion problem. In *Proceedings of the third text retrieval conference (TREC-3). Gaithersbirg, Maryland* (pp. 95–104).

Xu, J., & Callan, J. P. (1998). Effective retrieval with distributed collections. In *Proceedings of the 21st international ACM SIGIR conference on research and department in information retrieval. Melbourne, Australia* (pp. 112–120).