KMITL Sci. J. Vol. 5 No. 1 Feb. 2005

DYNAMIC DISTANCE METRIC FOR IMAGE RETRIEVAL SYSTEMS

Nualsawat Hiransakolwong[1] and Soontharee Koompairojn[2]

[1]Mathematics and Computer Science department,
King Mongkut's Institute of Technology
Ladkrabang, Bangkok 10520 Thailand
E-mail: khnualsa@kmitl.ac.th

[2]School of Computer Science, University of Central Florida
Orlando, FL 32816-2362, U.S.A.
E-mail: soonthar@cs.ucf.edu

ABSTRACT

Query-by-one-example (QBE) has been a popular query system for content-based image retrieval (CBIR) for more than a decade. However, recent research has shown that a single image is not sufficient to form its semantics or concept of the intended query. Searching concept "car," for instance, one might need many examples of car images in various colors. The color feature is then understood as a non-factor in the distance metric. This paper proposes a novel approach, which users can query using groups of query images. There are three possible groups: relevant (positive), irrelevant (negative) or neutral groups. The range for each feature within these groups of query images is defined. These ranges are used to adjust the weights of the features. As a result, some features may be cancelled out from the similarity computation. The measure then becomes a dynamic metric for image retrieval. This novel approach achieves a higher degree of precision and recall and, at the same time, significantly reduces the time complexity of matching. The proposed approach is tested against the *ImageGrouper* method. The results show that this approach is an effective and efficient technique for image retrieval systems.

Keywords: dynamic distance metric, range distance, content-based image retrieval, query-by-example

1. INTRODUCTION

Most QBE systems [9-14] employ the query-by-one-example model for multimedia query and retrieval. Recently, there emerges research (e.g., [1] and [2]) that considers query-by-multiple-examples with query semantics extracted from these images. By analyzing feature-to-semantics mapping, the authors in [2] argue that query-by-one-example approaches cannot realistically lead to a scalable and satisfactory query performance. More specifically, they cluster a small image dataset based on the images' perceptual features and show that these image clusters are not coherent to the semantic categories of the images. Though some image categories are separated from the others in the input space formed by the perceptual features, most categories are co-located in more than one cluster.

For a query concept that is mixed with others in a number of clusters, the query-by-one-example simply lacks information to clearly identify the target of the query concept and thus cannot achieve satisfactory query results.

Recently, there are a number of works supporting query-by-multiple-examples, such as *ImageGrouper* [20]. *ImageGrouper* contains a *Graphical User Interface* (GUI) that allows users to interactively form queries using multiple examples, by placing images in one of the three groups on the workspace. The user can quickly review the query results as they are displayed in another portion of the workspace. Each group is composed of a cluster of images; according to their relevancy of what the user thinks the returned images should contain. This system uses the weighted Euclidean distance of all of the features in a measure. They use the covariant between features contained within each group and between groups to adjust the weights in the formula. The covariant metric is minimized within the positive group, and concurrently, the covariant metric is maximized between the negative and positive groups. All features contribute to a different degree according to their own weights.

This paper proposes a dynamic distance computation for QBE. In this approach users can query using a single query image or multiple images arranged in groups. Users can specify three possible types of groups: positive, negative or neutral groups. For each feature within these groups of query images, the range of the feature (or range-distance) is computed and used for adjusting its weight. As a result, some features may have a zero weight and effectively be cancelled out in the distance metric. The net effect is a distance metric which is able to adapt itself automatically according to features' relevancy. The number of relevant features in a dynamic distance metric can vary anywhere from 1 to N, where N is the number of all employed features. The matching algorithm is fast, and provides a high precision and recall of relevant images. Experimental results show that this proposed method offers a significant improvement over the *ImageGrouper* method. This proposed approach is discussed in detail in this paper.

2. FEATURE VECTORS

As in *ImageGrouper*, the visual image features in this proposed approach consist of 37 components from three groups of features: *color*, *texture* and *edge structure*. In *ImageGrouper*, the standard deviation is used to represent the *Texture* feature. However, the standard deviation is a good representation only if there are a large number of query images. Unfortunately, users do not normally use many example images in a query. The average of query images is three to five images. On the other hand, experiments show that the average is a better representation for a small number of query images. Therefore, in the *texture* group in this paper, the average is used as a feature instead of the standard deviation. Experiments with both methods indicate that better results are achieved with the new features. For the remainder (*color* and *edge structure*), extracted features are identical to those in the *ImageGrouper*. These features are extracted from the images and indexed in the meta-data database offline, as described below.

For texture, *Wavelet-based Texture Features* [2][4][5] are used. First, the *Wavelet Filter Bank* [2][6] method is applied to each image, where the images are decomposed into ten de-correlated sub-bands (three levels), as shown in Figure 1. The upper left image in the wavelets image is the low frequency sub-band, while the lower right image is the high frequency sub-band. For each sub-band, the average of the wavelet coefficients is extracted. Thus, there are a total of ten texture features for each image.

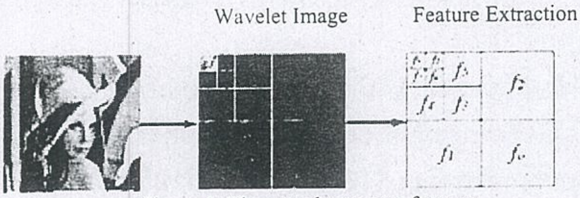


Figure 1: The wavelet texture features

For color features, the HSV color space is used. The first, second, and third moments for each of the H, S and V components are computed as features. Therefore, there are nine color features for each image.

For edge structures, the *Water-Filling Algorithm* [7][2] is applied to the edge maps of the images. The Sobel filter is applied to each image and then followed by the thinning operation [8] to generate its corresponding edge maps. From the edge map, eighteen elements are extracted. These features include the longest edge length, the histogram of the edge length, and the number of forks on the edges.

3. QUERY SETS

The query set consists of three possible groups: positive, negative, or neutral. The positive group consists of images that are similar to what the user has in mind as the results of the query. The negative group contains images that the user may want to filter from the result set. The neutral group is the group of images the user is unsure of their relevancy.

4. RANGE DISTANCE

Definition 1: For feature f_i of all images, the ***lower bound*** $(l_i) = \min\{\forall f_i\}$ and the ***upper bound*** $(u_i) = \max\{\forall f_i\}$. The ***range distance*** $(d_i) = |u_i - l_i|$.

From Section 2, there are a total of 37 extracted features for each image. For each feature in a cluster (positive, negative, or neutral groups), the corresponding lower and upper bounds are computed. The range distance is defined as the distance between the lower bound and upper bound for each feature, as illustrated in Figure 2.

In Figure 2, only two features, f_i and f_j, are shown. Assume that there are three images in the positive group: A, B and C. For feature f_i, there are $l_i = \min\{f_{iA}, f_{iB}, f_{iC}\}$, and $u_i = \max\{f_{iA}, f_{iB}, f_{iC}\}$. The range distance is $d_i = |u_i - l_i|$. Similarly, for feature f_j, there are $l_j = \min\{f_{jA}, f_{jB}, f_{jC}\}$, $u_j = \max\{f_{jA}, f_{jB}, f_{jC}\}$, and $d_j = |u_j - l_j|$.

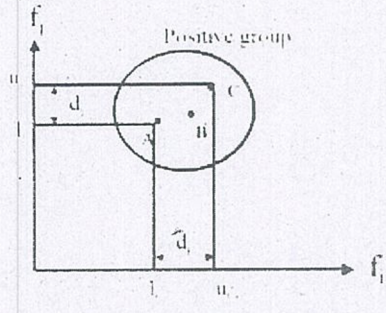


Figure 2: The range distances d_i and d_j

5. DYNAMIC DISTANCE METRIC

Definition 2: The *distance metric* $D(I,q)$ between image (I) and query image (q) is defined as $D(I,q)=\Sigma w_i|I_i-q_i|$, where w_i is a weight in range [0,1] for the feature i. I_i is the feature i of database image I. Similarly, q_i is the feature i of query image q.

Within each of these groups of query images, the range distance of feature i is used to adjust weights w_i in the distance metric. An algorithm describing how to adjust the weight is detailed in the following section. Some weights might be set to zero and eliminated from the distance metric.

6. MATCHING ALGORITHM

The matching algorithm is proposed as follows:

1. For each positive, negative, and neutral group
   - For each feature (f_i), compute the lower bound (l_i), the upper bound (u_i), and then compute its range distance $(d_i) = u_i - l_i$

   Therefore, for each feature f_i there are three range distances, d_{ip}, d_{ine}, and d_{inu}, for the positive, negative, and neutral groups, respectively.
2. For each feature (f_i), compute $d_i = \min\{d_{ip}, d_{ine}, d_{inu}\}$

   if $d_i = d_{ine}$, then $w_i = 0$ and discard this d_i
3. For each feature (f_i)

   3.1 Compute $d_i' = d_i / \sum_i d_i$

   3.2 Compute weight $w_i = 1 - d_i'$

   if $d_i = d_{inu}$, then compute $w_i = w_i / num$

   where $num =$ the number of members in the neutral group
4. Calculate the distance metric

   For each feature i in the positive group, compute the feature average (q_i) from all group members. This forms the query features (q). For each database image I, compute the distance metric $D(I,q)$ using the following equation:

   $$D(I,q) = \Sigma |I_i-q_i| \cdot (w_i / \Sigma w_i),$$

   where $(w_i/\Sigma w_i)$ is the normalization of w_i into the range [0,1]. Finally, the results are sorted in ascending order, and the K top-ranked images are displayed.

According to the algorithm, dominant features between the positive group and the neutral group are selected in the adaptive distance metric and the weights are determined based on the range distance. The motivation for adjusting the weights comes from the fact that a small range distance signifies the user's fix on the feature and thus should receive high weight. For each feature, the shortest range distance of one group dominates those of the other groups, leading to the selection of the feature as relevant, irrelevant, or neutral. Normally, features in the negative group are scattered with different concepts. This makes d_{ine} larger than d_{ip}. From step 2, if $d_i = d_{ine}$, then $w_i = 0$. On the other hand, if feature i in the negative group dominates feature i in the positive group, it means that users do not want this feature to be included. Therefore, feature i should be ignored from the distance metric.

Steps 2 and 3 adjust the weights from the user's information. In the *ImageGrouper* approach, there is no meaning for the neutral group. In Step 3.2 of the above algorithm, the weights in the neutral group, adjusted by dividing by *num*, have lower values than the weights in the positive group. From experiments, if users provide more accurate clustering, then the proposed approach produces better precision and recall.

7. TIME COMPLEXITY

Assume there are N dimensions of features and k is the number of members in all groups. From finding weights (in step 1-3 of the algorithm) in the last section, the time complexity is $O(kN)$ and the proposed approach uses the city-block distance metric. In the *ImageGrouper approach*, covariants between features in each group are used to adjust the features' weights. The time complexity for finding weights is $O(kN^2)$ and they use the Euclidian distance metric that has higher time complexity than the proposed approach. Time requirements of both methods are reported in the experimental study section.

8. EXPERIMENTAL STUDY

8.1 DATASETS AND METRICS

A dataset of about 3,000 images was collected from the COREL image database plus some of added to test various features of the proposed technique. Query images were selected from five categories: dishes, lions, views of the sky, flowers, and cars. There are 18 images of 3-brown-dishes, 10 images of lions, 53 images of skies, 36 images of flowers, and 165 images of cars in this dataset.

In proposed approach, programs are coded in C language and GUI in Java. The *ImageGrouper* approach used the software downloaded at [3]. For each dataset image and query image, the 37 features were extracted (*color*, *texture* and *edge structure*) as described in Section 2. The experiments were run on a computer with a 2.4 GHz Pentium 4 processor.

8.2 COMPARATIVE STUDY

There were a total of 150 query sets. These query sets consisted of a variety of images including positive, negative and neutral groups. The same query sets were tested with both the proposed approach and the *ImageGrouper* approach. The average response time required by *ImageGrouper* was 0.456 seconds per query, while the proposed approach required 0.000667 seconds. In other words, the proposed method is more than 600 times faster than the *ImageGrouper* technique.

Some query results are shown in Figures 3-5. In Figures 3-5, (A) is the result set from the dynamic distance approach, and (B) is the result set from the *ImageGrouper* approach. The query set is on the right-hand side and the results on the left-hand side. The returned images are ordered from left to right and top to bottom. In Figure 3, the user wants images of red or pink flowers. People wearing red and black clothes were selected in the negative query group. The red car was selected as a neutral query set. In Figure 4, the user attempts to retrieve images of dishes that have a 3-dish structure, excluding images of only one plate or one cup. In Figure 5, the user tries to query images of lions (enclosed in the positive group), not a white tiger (enclosed the negative group). As seen in these examples, the proposed approach is more effective than the *ImageGrouper* approach. A comparison of the recall and the precision rates between these two techniques is shown in Table 1. The present approach consistently has higher recall and precision than the *ImageGrouper* in all tested categories.

In summary, the proposed approach has the following advantages over the *ImageGrouper* approach:

1. The matching algorithm can automatically detect dominant features and assign weights accordingly. Irrelevant features are aggressively removed from the similarity computation. This results in better retrieval performance.
2. Eliminating unwanted features helps reduce matching time. Together with a simpler, but more effective distance metric, the proposed technique is many times faster than the *ImageGrouper*.

9. Conclusions

Query-by-multiple-examples are a promising paradigm for QBE. A dynamic distance computation method is proposed in this paper. This algorithm detects significant features and automatically adjusts the features' weights based on their range distances. These weights effectively eliminate features that are not relevant in the similarity measure. The present technique achieves a high recognition rate and, at the same time, significantly reduces time complexity of matching. The results show that the proposed technique is an effective and efficient technique for QBE.

10. References

[1] Munehiro Nakazato and Thomas S. Huang, "Extending Image Retrieval with Group-Oriented Interface," In Proceedings of *IEEE ICME2002*, 2002.

[2] Thomas E. Bjoerge and Edward Y. Chang, "Why one example is not enough for an image query," In Proceedings of *IEEE ICME 2004*, 2004.

[3] http://www.ifp.uiuc.edu/~nakazato/grouper/

[4] Smith, J. R. and Chang S-F. Transform features for texture classification and discrimination in large image databases. In *Proceedings of IEEE Intl. Conf. on Image Processing,* 1994.

[5] Smith J. R. and Chang S-F. "Quad-Tree Segmentation for Texture-based Image Query." In *Proceedings of ACM 2nd International Conference on Multimedia,* 1994.

[6] Strang, G. and Nguyen, T., "Wavelet and Filter Banks," Wellesley-Cambridge Press, 1997.

[7] Zhou, X. S. and Huang, T. S., "Edge-based structural feature for content-base image retrieval," *Pattern Recognition Letters, Special issue on Image and Video Indexing*, 2000.

[8] Gonzales, R. C. and Woods, R. E., "Digital Image Processing," Addison-Wesley, 1992.

[9] A.W.M. Smeulders et al, Content-based image retrieval at the end of the early years. *IEEE Trans. On PAMI*, 22(12): 1349-1380, 2000.

[10] Ying Dai and Dawei Cai, "Imagery - based digital collection retrieval using eigen SGLD matrices," In Proceedings of *IEEE ICME 2004*, 2004.

[11] Joo-Hwee Lim and Jesse S. Jin, "Image retrieval using spatial icons," In Proceedings of *IEEE ICME 2004*, 2004.

[12] S. Berretti, G.D. Amico and A. Del Bimbo, "Shape representation by spatial partitioning for content based retrieval applications," In Proceedings of *IEEE ICME 2004*, 2004.

[13] B. Ko, H. S. Lee, and H. Byun, "Region-based image retrieval system using efficient feature description," In Proceedings of *IEEE Int. Conf. on Pattern Recognition*, Barcelona, Spain, 2000, pp. 283-286.

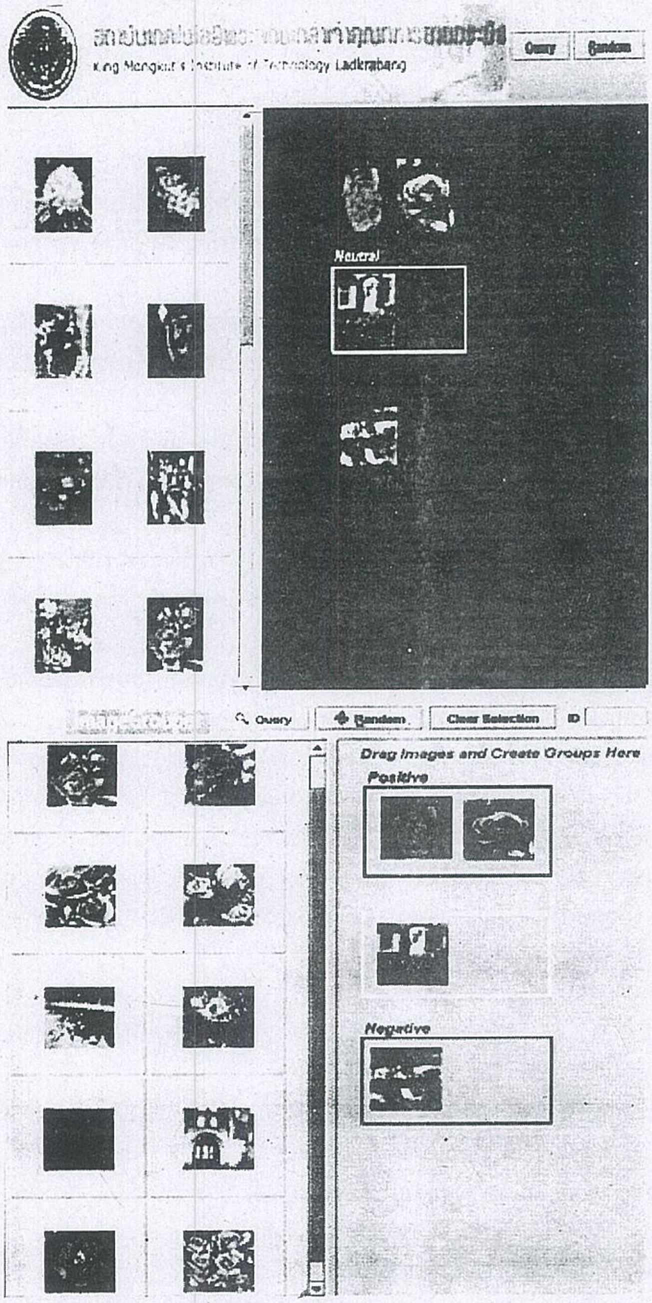
[14] F, Liu, X. Xiong, and K. L. Chan, "Natural image retrieval based on features of homogeneous color regions," In Proceedings of *IEEE on Image Analysis and Interpretation*, Austin, Texas, Apr. 2000, pp. 73-77.

[15] Xiangyu Jin and James C. French, "Content-based image retrieval for multimedia databases: Improving image retrieval effectiveness via multiple queries," *Proceedings of the first ACM international workshop on Multimedia databases*, Nov. 2003, pp.86-93.

Categories	**The proposed approach**		***ImageGrouper* approach**	
	Recall	**Precision**	**Recall**	**Precision**
Dishes	1.0	0.8	0.56	0.45
Lions	0.94	0.9	0.75	0.75
Sky	0.87	0.97	0.72	0.83
Flowers	0.83	0.94	0.72	0.86
Cars	0.84	0.88	0.54	0.65

Table 1: Summary of the recall and precision rates in example categories

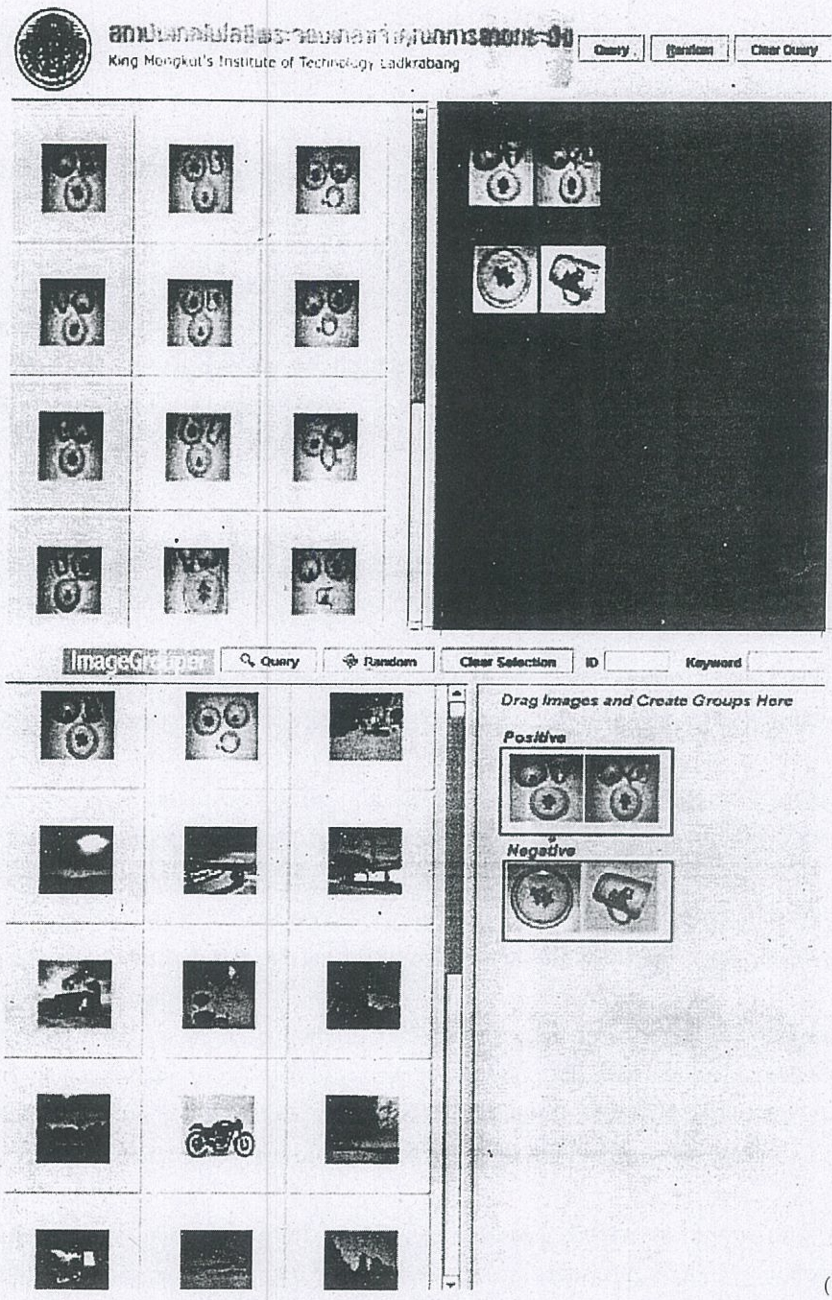
Experimental result sets



(A)

(B)

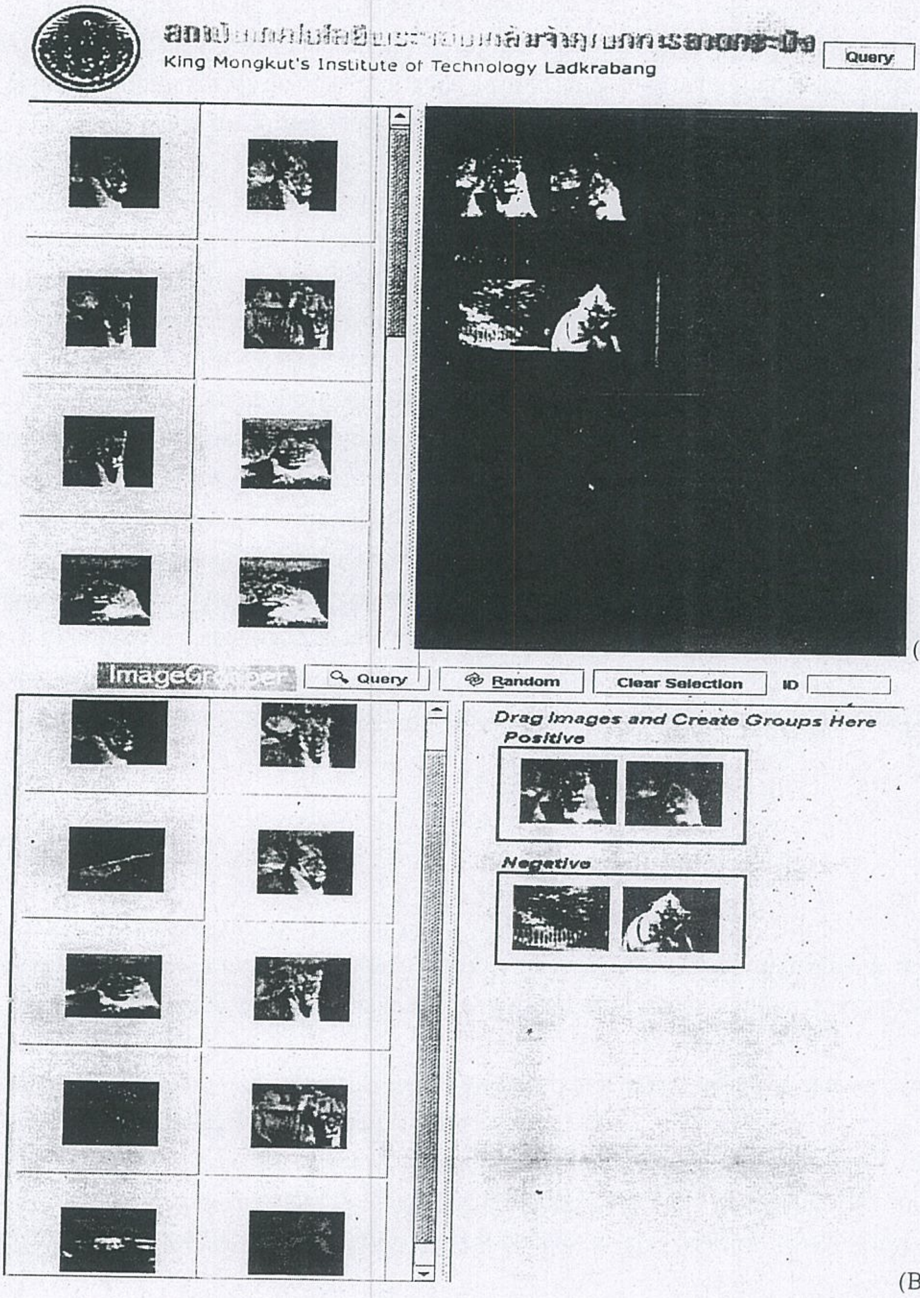
Figure 3: Result set 1 (the proposed approach & the *ImageGrouper* approach)



(A)

(B)

Figure 4: Result set 2 (the proposed approach & the *ImageGrouper* approach)



(A)

(B)

Figure 5: Result set 3 (the proposed approach & the *ImageGrouper* approach)