Image Clustering for a Fuzzy Hamming Distance Based CBIR System

Mircea M. Ionescu

ECECS Department
ML 0030
University of Cincinnati
Cincinnati, OH 45221, USA
ionescmm@ececs.uc.edu

Anca L. Ralescu

ECECS Department
ML 0030
University of Cincinnati
Cincinnati, OH 45221, USA
Anca.Ralescu@uc.edu

Abstract

A linear search in a Content-Based Image Retrieval (CBIR) system is time consuming. Like any database system an indexing technique is mandatory for the system to be efficient. This paper studies the application of clustering algorithms to a Fuzzy Hamming Distance based CBIR system for building the image index. The study shows good results using complete linkage agglomerative clustering.

Introduction

Content-Based Image Retrieval (CBIR) systems are designed to find a relevant image set from a large database. As opposed to keyword-based, CBIR systems relay only on the information from the image content. This approach is more realistic for todays large data sets, where an annotation would be difficult if not impossible. A survey of the functionality of current CBIR systems can be found in (Veltkamp & Tanase 2000) and (Antani, Kasturi, & Jain 2002).

Fuzzy Hamming Distance

The Fuzzy Hamming Distance is a generalization of Hamming distance over the set of real-valued vectors. It preserves the original meaning of the Hamming distance as the number of different components between the input vectors with the added features that it uses real-valued vectors and it takes in account the amount of the difference between each component. FHD is the (fuzzy) number of different components of the input vectors. The fuzzy set shows the degree to which the input vectors are different by 0, 1,...,n, where n is the size of the vectors. In short, the Fuzzy Hamming Distance is the fuzzy cardinality of the difference fuzzy set.

The Fuzzy Hamming Distance is described in detail in (Ralescu 2003) and (Ionescu & Ralescu 2004). Here only the definition is given.

Definition 0.1 (The Fuzzy Hamming Distance) (Ralescu 2003) *Given two n dimensional real-valued vectors, x and y, for which the difference fuzzy set $D_\alpha(x, y)$, with membership function $\mu_{D_\alpha(x,y)} = 1 - e^{-\alpha(x-y)^2}$, the* **fuzzy Hamming distance** *between x and y, denoted by $FHD_\alpha(x, y)$ is the fuzzy cardinality of the difference fuzzy set, $D_\alpha(x, y)$:*

$\mu_{FHD(x,y)}(\,\cdot\,\alpha) : \{0, \ldots, n\} \rightarrow [0, 1]$ *denotes the membership function for $FHD_\alpha(x, y)$ corresponding to the parameter α. More precisely,*

$$\mu_{FHD(x,y)}(k; \alpha) = \mu_{CardD_\alpha(x,y)}(k) \qquad (1)$$

for $k \in \{0, \ldots, n\}$ where $n = |SupportD_\alpha(x, y)|$.

In words, (1) means that for a given value k, $\mu_{FHD(x,y)}(k; \alpha)$ is *the degree to which the vectors x and y are different on exactly k components (with the modulation constant α).*

The CBIR System

The CBIR system used in this study uses FHD as a similarity measure between two images. A more detailed description is given in (Ionescu & Ralescu 2004). Figure 1 shows the system architecture. It can be seen that the approach consists of three simple steps:

1. The **preprocessing module** extracts the information of interest (in this study the color histograms) from each of the images in the database and the query image. The output of this module is a collection of color histograms.
2. The **similarity assessment module** takes as input the information from the preprocessing module and computes the similarity (actually the FHD), between the query image and each image in the database. The output of this module is a collection of fuzzy sets (FHD).
3. The **ranking module** combines the outcome of similarity assessment into a score and it returns it ranked in decreasing order.

In order to include position information, in the preprocessing module each image is partitioned in $m \times n$ partitions. Then for each partition the color histogram is computed.

CBIR Indexing

The current system uses a linear search to find the similar images with a given query images. To speedup the retrieval process the use of clustering algorithms in building an index is explored. Four clustering algorithms are evaluated:

- K-means clustering;
- Fuzzy C-Mean clustering;

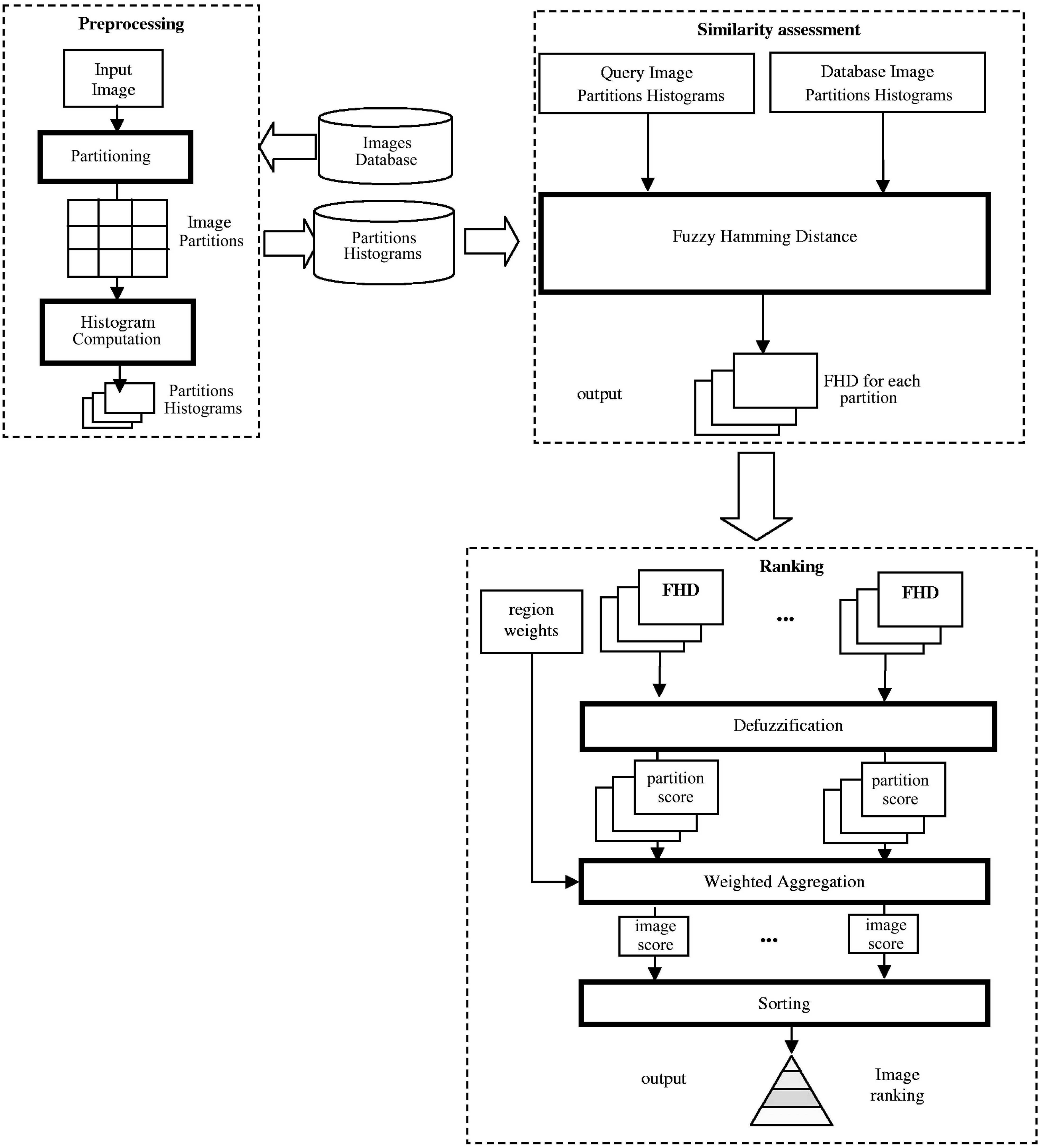

Figure 1: CBIR system architecture

- Single linkage agglomerative clustering;
- Complete linkage agglomerative clustering;

From our experiments K-means and Fuzzy C-Means clustering have a long convergence time. Actually the number of steps required to converge is undetermined and is highly sensitive to the initialization. Also they require a pre-specified number of clusters. Single linkage agglomerative algorithm creates small and long clusters. This way the center of the cluster is not representative anymore. The complete linkage agglomerative clustering (Theodoridis & Kouroumbas 2003) was selected offering a well defined number of steps and a good clustering. The algorithm used to crate M clusters is the following:

1. Start with n clusters, one for each image, $C_k = \{i_k = image, k = 1 \dots n\}$;
2. Compute the distance between clusters $D(C_i, C_j)$ as the maximum distance between all the images of the two clusters;
3. Merge the clusters with the minimum distance;
4. Repeat steps 2 and 3 until the number of clusters is M;
5. Compute the center of each cluster;

To assess the similarity between two images we used Center of Gravity defuzzification of Fuzzy Hamming Distance,FHD_{COG}.This is the same measure used by the CBIR system.

Let:

- $N =$ the number of images in the database;
- $M =$ the number of clusters C_i $i = 1 \dots M$;
- $\|C\| =$ the maximum number of images into a cluster;
- $K =$ the number of images in the result set;
- Q the query image;

Then at runtime, to retrieve the closest K images to the query image, the following algorithm is used:

1. Compute $D_i = FHD_{COG}(Q, C_i)$;
2. Select the candidate images from the cluster C_i where $i = argmin\{D_i\}$;
3. Repeat step 2 until at least K candidate images are selected;
4. Compute $d_i = FHD_{COG}(Q, I_i)$ where I_i is a candidate image;
5. Display the first K images in the increasing order of d_i;

As it can be seen from the algorithm the number of similarity measure computations for the linear and index based searched are, respectively:

$$N_{linear} = N;$$

$$N_{index} = M + \left(\left\lfloor \frac{K}{\|C\|} \right\rfloor + 1 \right) \times \|C\|;$$

Clusters #	2	6	**13**	24	28	38	51
S / **A [%]**	1.98	4.9	**9.51**	11.64	12.07	11.5	10.39
60	0.96	0.84	**0.8**	0.92	0.88	0.92	0.92
70	0.88	0.84	**0.76**	0.8	0.8	0.84	0.84
80	0.88	0.84	**0.76**	0.8	0.8	0.84	0.84
90	0.8	0.64	**0.56**	0.56	0.48	0.52	0.52
100	0.8	0.64	**0.56**	0.56	0.48	0.52	0.52

Table 1: Query agreement, A and the speedup factor S for different number of clusters and 1×1 partitioning of the images

As it can be seen, a good speedup is determined, provided that the size of the result set, K, is relatively small. In order to enforce position information the images can be divided in $m \times n$ partitions. In this case the speedup is even larger, the number of comparisons being:

$$N_{linear} = N \times m \times n$$

$$N_{index} = M + \left(\left\lfloor \frac{K}{\|C\|} \right\rfloor + 1 \right) \times \|C\| \times m \times n$$

This is because in-depth partition by partition comparisons are performed only for the images in the shortlist (from the same cluster).

Results and Conclusion

The database used (Washington-Image-Database) consist of 855 jpeg images from different categories like:

- City landscapes (Barcelona and Italy);
- Campus images;
- Park landscape;
- Sea landscape;

To evaluate the performance of the system using the index versus the system using linear search, 25 random query images were selected. For each number of clusters $M = 2 \dots 43$ the following were evaluated:

- Result set agreement (A): the percentage of images that are both in the result set using the index and using the linear search. Five agreement level are used 60%, 70%, 80%, 90% and 100%. For each level the percentage of the query images for which the agreement is grater then or equal to the agreement level is evaluated;
- Speedup (S): the average ratio between the number of comparisons required by the index and the one from linear search;

The results are shown in Table 1, Figures 2,3 and 4.

As it can be seen from the Table 1 as the number of clusters increases, the speedup factor also increases.This is because are less picture in each cluster. But in the same time the agreement factor is reduced. This is because some images from the linear search result set are in another cluster than the one where the query image is.

$M = 13$, the number of clusters, is selected which offers an average speedup $S = 9.5$ and an agreement of:

- 80% of the query images has $A \geq 60$ %;
- 76% of the query images has $A \geq 70$ %; and $A \geq 80$ %;
- 56% of the query images has $A \geq 100$ %;

In the Figures 2, 3 and 4 the query image is in the first column and for each query image the first row shows the result set return by the index and the second one is the result set of the linear search.

Even if the indexing algorithm does not provide a 100% agreement with the linear search the gain in speed is very important. This a major factor because the image databases are huge and without an indexing mechanism is imposible to create an practical CBIR system.

Clustering the images in groups add another important feature to the system: browsing. It allows to the user to browse the database in search for a group of similar images. Also several clusterizations can be employed. This is because semantically the images can be categorized differently based on classification criterion. This way semantics beyond the color similarity can be added to the query.

Future work includes:

- Exploring other clustering algorithms;
- Cluster images using $m \times n$ partition;
- Add browsing capability to the CBIR system;

Acknowledgments

This work was partially supported by the Ohio Board of Regents, Grant N000140310706 from the Department of the Navy, and the Rindsberg Graduate Fellowship of the College of Engineering, University of Cincinnati.

References

Antani, S.; Kasturi, R.; and Jain, R. 2002. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition* 35(4):945–965.

Ionescu, M., and Ralescu, A. 2004. Fuzzy hamming distance in a content-based image retrieval system. IEEE-FUZZ-2004.

Ralescu, A. 2003. Generalization of the hamming distance using fuzzy sets. Research Report, JSPS Senior Research Fellowship, Laboratory for Mathematical Neuroscience, The Brain Science Institute, RIKEN, Japan.

Theodoridis, S., and Kouroumbas, K. 2003. Pattern recognition.

Veltkamp, R., and Tanase, M. 2000. Content-based image retrieval systems: a survey. Technical report, UU-CS-2000-34, Utrecht University.

Washington-Image-Database. Department of computer science and engineering, university of washington. http://www.cs.washington.edu/research/imagedatabase/groundtruth.

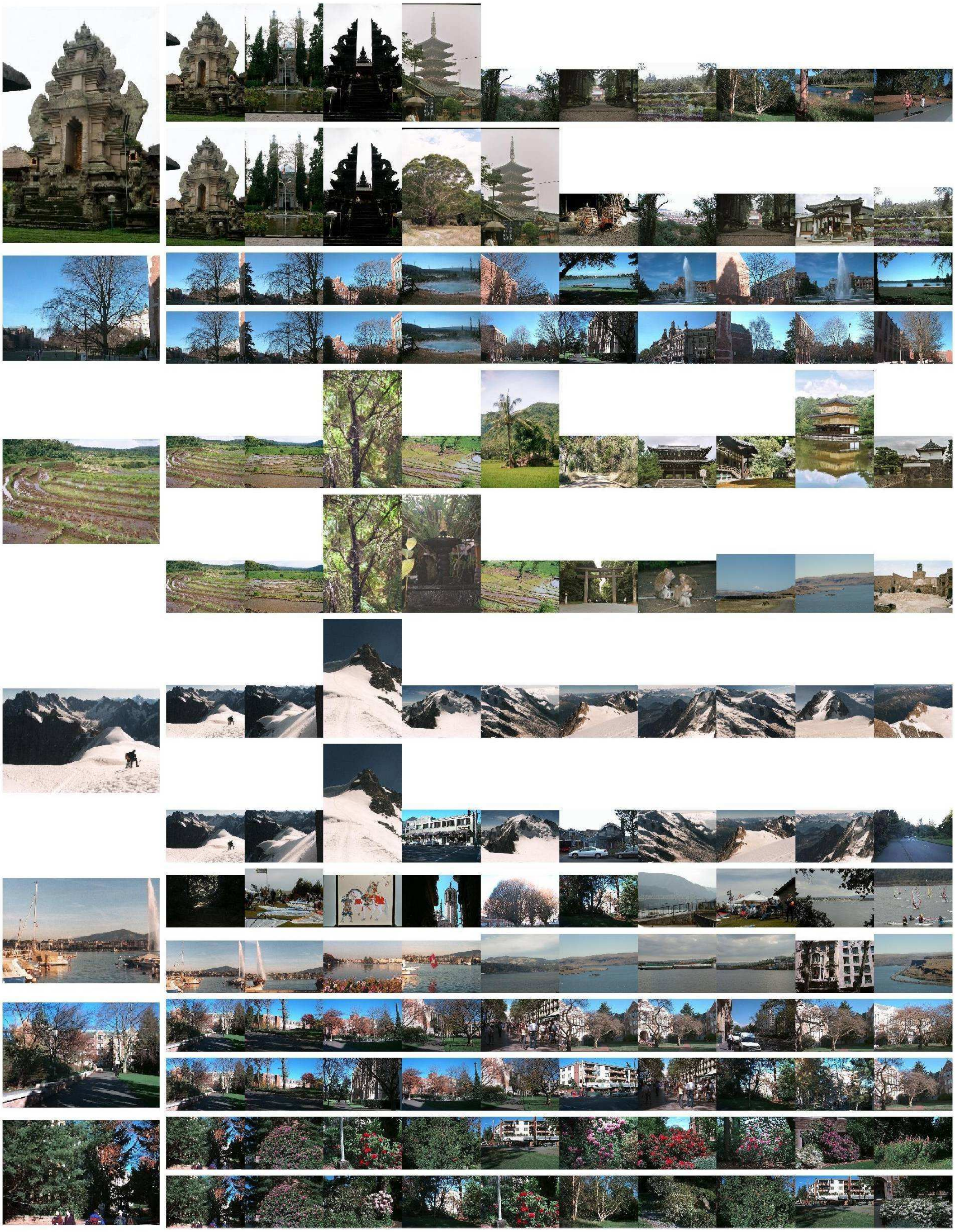

Figure 2: Result set returned by the system using the index and using linear search. First column shows the query image. For each query image in the first row is the indexing result set and second row the linear search result set.

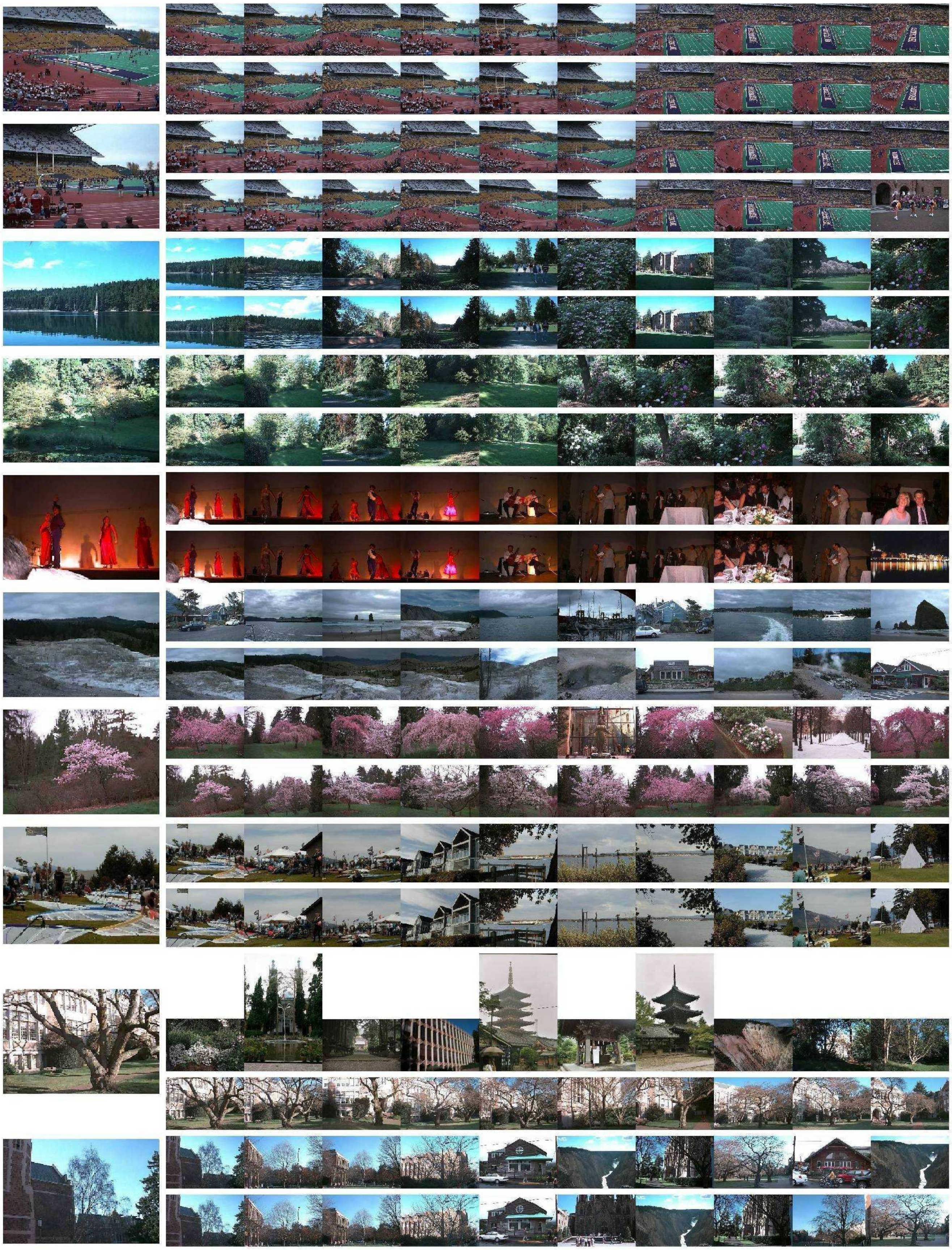

Figure 3: Result set returned by the system using the index and using linear search. First column shows the query image. For each query image in the first row is the indexing result set and second row the linear search result set.

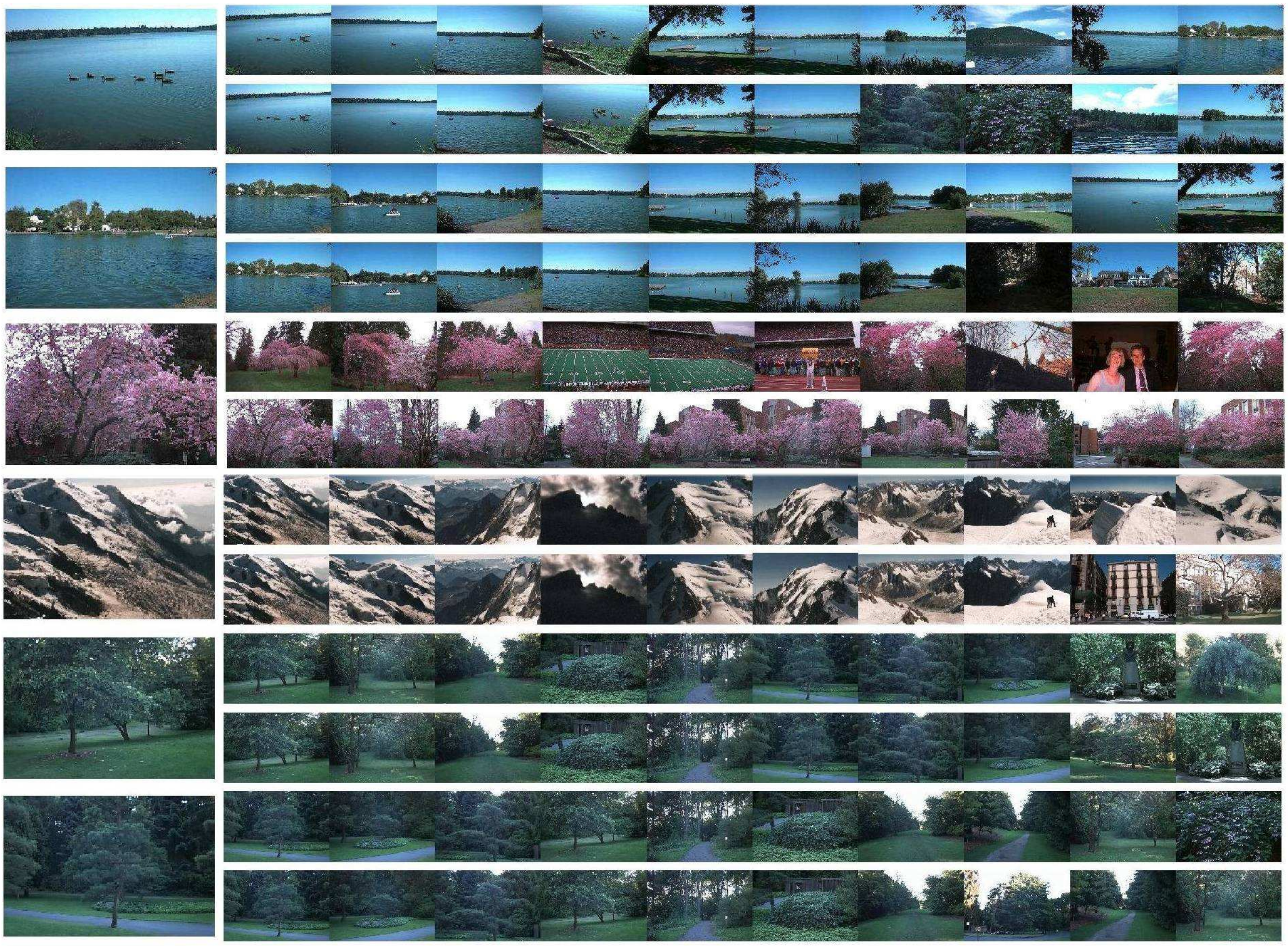

Figure 4: Result set returned by the system using the index and using linear search. First column shows the query image. For each query image in the first row is the indexing result set and second row the linear search result set.