A SCALABLE INTEGRATED REGION-BASED IMAGE RETRIEVAL SYSTEM

Yanping Du *James Z. Wang*

The Pennsylvania State University, University Park, PA 16801

ABSTRACT

In this paper, we present a scalable algorithm for indexing and retrieving images based on region segmentation. The method uses statistical clustering on region features and IRM (Integrated Region Matching), a measure developed to evaluate overall similarity between images incorporates properties of all the regions in the images by a region-matching scheme. The algorithm has been implemented as a part of our experimental SIMPLIcity image retrieval system and tested on large-scale image databases of both general-purpose images and pathology slides. Experiments have demonstrated that this technique maintains the accuracy of the original system while reducing the matching time significantly.

1. INTRODUCTION

Content-based image retrieval has been widely studied. In the commercial domain, IBM QBIC [9] is one of the earliest developed systems. Recently, additional systems have been developed at IBM T.J. Watson [14], VIRAGE [3], NEC C&C Research Labs [8], Interpix (Yahoo), Excalibur, and Scour.net. In academia, MIT Photobook [10] is one of the earliest. Berkeley Blobworld [1], Columbia VisualSEEK and WebSEEK [13], CMU Informedia [15], UIUC MARS [7], UCSB NeTra [6], UCSD, Stanford WBIIS [16], and Stanford SIMPLIcity [17]) are some of the recent systems.

Region-based approach has recently become a popular research trend. Li et al. of Stanford University recently developed SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries) [17]. SIMPLIcity uses semantics type classification and an integrated region matching (IRM) scheme to provide efficient and robust region-based image matching [5]. The IRM measure is a similarity measure of images based on region representations. It incorporates the properties of all the segmented regions so that information about an image can be fully used. With IRM, region-based image-to-image matching can be performed. The overall similarity approach reduces the adverse effect of inaccurate segmentation, helps to clarify the semantics of a particular region, and enables a *simple* querying interface for region-based image retrieval systems. Experiments have shown that IRM is comparatively more effective and more robust than many existing retrieval methods. Like other region-based systems, the SIMPLIcity system is a linear matching system. To perform a query, the system compares the query image with all images in the same semantic class.

In this paper, we present an enhancement to the SIMPLIcity system for handling image libraries with million of images. The targeted applications include Web image retrieval and biomedical image retrieval. Region features of images in the same semantic class are clustered automatically using a statistical clustering method. Features in the same cluster are stored in the same file for efficient access during the matching process. IRM (Integrated Region Matching) is used in the query matching process. Tested on large-scale image databases, the system has demonstrated high accuracy and scalability.

2. THE SIMILARITY MEASURE

In this section, we describe the similarity matching process we developed. We briefly describe the segmentation process and related notations in Section 2.1. The feature space analysis process is described in Section 2.2. In Section 2.3, we give details of the matching scheme.

2.1. Region segmentation

Semantically-precise image segmentation is extremely difficult and is still an open problem in computer vision [12, 18]. We attempt to develop a robust matching metric that can reduce the adverse effect of inaccurate segmentation. The segmentation process in our system is very efficient because it is essentially a wavelet-based fast statistical clustering process on blocks of pixels.

To segment an image, we partitions the image into blocks with $t \times t$ pixels and extracts a feature vector for each block. The k-means algorithm is used to cluster the feature vectors into several classes with every class corresponding to one

The authors are with School of Information Sciences and Technology and Department of Computer Scinece and Engineering. This work was supported in part by the National Science Foundation's Digital Library II initiative under Grant No. IIS-9817511. and an endowment from the PNC Foundation. The authors wish to thank the help of Jia Li, Gio Wiederhold, and Oscar Firschein.

region in the segmented image. We dynamically determine k by starting with $k = 2$ and refine if necessary to $k = 4$, etc. The details of the segmentation process is described in [5].

Six features are used for segmentation. Three of them are the average color components in a $t \times t$ block. The other three represent energy in high frequency bands of wavelet transforms [2], that is, the square root of the second order moment of wavelet coefficients in high frequency bands. We use the well-known LUV color space, where L encodes luminance, and U and V encode color information (chrominance). The LUV color space has good perception correlation properties. We chose the block size t to be 4 to compromise between the texture detail and the computation time.

Let N denote the total number of images in the image database. For the i-th image, denoted as R_i, in the database, we obtain a set of n_i feature vectors after the region segmentation process. Each of these n_i d-dimensional feature vectors represents the dominant visual features (including color and texture) of a region, the shape of that region, the rough location in the image, and some statistics of the features obtained in that region.

2.2. Feature space analysis

The new integrated region matching scheme depends on the entire picture library. We must first process and analyze the characteristics of the d-dimensional feature space.

Suppose feature vectors in the d-dimensional feature space are $\{x_i : i = 1, ..., L\}$, where L is the total number of regions in the picture library. Then $L = \sum_{i=1}^{N} n_i$.

The goal of the feature clustering algorithm is to partition the features into k groups with centroids $\hat{x}_1, \hat{x}_2, ..., \hat{x}_k$ such that

$$D(k) = \sum_{i=1}^{L} \min_{1 \le j \le k} (x_i - \hat{x}_j)^2 \qquad (1)$$

is minimized. That is, the average distance between a feature vector and the group with the nearest centroid to it is minimized. Two necessary conditions for the k groups are:

1. Each feature vector is partitioned into the cluster with the nearest centroid to it.

2. The centroid of a cluster is the vector minimizing the average distance from it to any feature vector in the cluster. In the special case of the Euclidean distance, the centroid should be the mean vector of all the feature vectors in the cluster.

These requirements of our feature grouping process are the same requirements as those of the Lloyd algorithm [4] to find k cluster means.

If the Euclidean distance is used, the k-means algorithm results in hyper-planes as cluster boundaries. That is, for the feature space $\mathbb{R}^d$, the cluster boundaries are hyper-planes in the $d - 1$ dimensional space $\mathbb{R}^{d-1}$. Here, the use of the Euclidean distance is reasonable because we cluster the feature space with each feature point corresponding to a region. Our region-to-region distance is a variation of the Euclidean distance.

Both the initialization process and the stopping criterion are critical in the process. We initialize the algorithm adaptively by choosing the number of clusters k by gradually increasing k and stop when a criterion is met. We start with $k = 2$. The k-means algorithm terminates when no more feature vectors are changing classes. It can be proved that the k-means algorithm is guaranteed to terminate, based on the fact that both steps of k-means (i.e., assigning vectors to nearest centroids and computing cluster centroids) reduce the average class variance. In practice, running to completion may require a large number of iterations. The cost for each iteration is $O(kn)$, for the data size n. Our stopping criterion is to stop after the average class variance is smaller than a threshold or after the reduction of the class variance is smaller than a threshold.

2.3. Image matching

To retrieve similar images for a query image, we first locate the clusters of the feature space to which the regions of the query image belong. Let's assume that the centroids of the set of k clusters are $\{c_1, c_2, ..., c_k\}$. We assume the query image is represented by region sets $R_1 = \{r_1, r_2, ..., r_m\}$, where r_i is the descriptor of region i. For each region feature r_i, we find j such that

$$d(r_i, c_j) = \min_{1 \le l \le k} d(r_i, c_l)$$

where $d(r_1, r_2)$ is the region-to-region distance defined for the system. This distance can be a non-Euclidean distance. We create a list of clusters, denoted as $\{c_{r_1}, c_{r_2}, ..., c_{r_m}\}$. The matching algorithm will further investigate only these 'suspect' clusters to answer the query.

With the list of 'suspect' clusters, we create a list of 'suspect' images. An image in the database is a 'suspect' image to the query if the image contains at least one region feature in these 'suspect' clusters. This step can be accomplished by merging the cluster image IDs non-repeatedly.

To define the similarity measure between two sets of regions, we assume that the image R_1 and image R_2 are represented by region sets $R_1 = \{r_1, r_2, ..., r_m\}$ and $R_2 = \{r'_1, r'_2, ..., r'_n\}$, where r_i or r'_i is the descriptor of region i. Denote the distance between region r_i and r'_j as $d(r_i, r'_j)$, which is written as $d_{i,j}$ in short. To compute the similarity measure between region sets R_1 and R_2, $d(R_1, R_2)$, we first compute all pair-wise region-to-region distances in the two

images. Our matching scheme aims at building correspondence between regions that is consistent with our perception. To increase robustness against segmentation errors, we allow a region to be matched to several regions in another image. A matching between r_i and r'_j is assigned with a significance credit $s_{i,j}$, $s_{i,j} \geq 0$. The significance credit indicates the importance of the matching for determining similarity between images. The matrix $S = \{s_{i,j}\}$, $1 \leq i \leq n$, $1 \leq j \leq m$, is referred to as the significance matrix. The significant credits can be assigned to the area percentages of the regions. Other schemes can also be used.

The distance between the two region sets is the summation of all the weighted matching strength, i.e.,

$$d_{IRM}(R_1, R_2) = \sum_{i,j} s_{i,j} d_{i,j} \ .$$

This distance is the integrated region matching (IRM) distance defined by Li et al. in [5].

3. EXPERIMENTS

This algorithm has been implemented and compared with the first version of our experimental SIMPLIcity image retrieval system. We tested the system on a general-purpose image database (from COREL) including about $200,000$ pictures, which are stored in JPEG format with size 384×256 or 256×384. To conduct a fair comparison, we use only picture features in the retrieval process.

3.1. Speed

On a Pentium III 800MHz PC using the Linux operating system, it requires approximately 60 hours to compute the feature vectors for the $200,000$ color images of size 384×256 in our general-purpose image database. On average, one second is needed to segment an image and to compute the features of all regions. Fast indexing has provided us with the capability of handling outside queries and sketch queries in real-time.

The feature clustering process is performed only once for each database. The Lloyd algorithm takes about 30 minutes CPU time and results in clusters with an average of 1100 images. Our image segmentation process generates an average of 4.6 regions per image. That is, on average a 'suspect' list for a query image contains at most $1100 \times 4.6 = 5060$ images.

The matching speed is fast. When the query image is in the database, it takes about 0.15 seconds of CPU time on average to retrieve a set of similar images from the 200,000-image database using our similarity measure. This is a significant speed-up over the original system which runs at 1.5 second per query. The speed-up is made possible by the statistical clustering process. If the query is not in the database, one extra second of CPU time is spent to process the query. Figure 1 shows the results of a sample query.

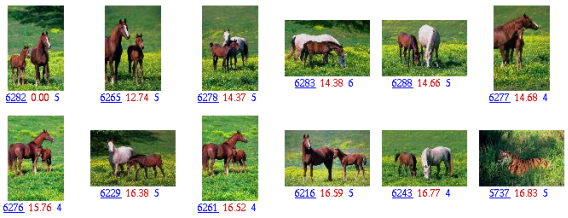

Fig. 1. Best 11 matches of a sample query. The database contains 200,000 images from the COREL image library. The upper left corner is the query image. The second image in the first row is the best match.

Category	IRM	fast IRM	EMD2	EMD 1
1. Africa	0.475	0.472	0.288	0.132
2. Beach	0.325	0.323	0.286	0.134
3. Buildings	0.330	0.307	0.233	0.160
4. Buses	0.363	0.389	0.267	0.108
5. Dinosaurs	0.981	0.635	0.914	0.143
6. Elephants	0.400	0.390	0.384	0.169
7. Flowers	0.402	0.447	0.416	0.113
8. Horses	0.719	0.669	0.386	0.096
9. Mountains	0.342	0.335	0.218	0.198
10. Food	0.340	0.340	0.207	0.114

Table 1. The average performance for each image category evaluated by average precision (p).

3.2. Accuracy on image categorization

We conducted extensive evaluation of the system. One experiment was based on a subset of the COREL database, formed by 10 image categories, each containing 100 pictures. Within this database, it is known whether any two images are of the same category. In particular, a retrieved image is considered a match if and only if it is in the same category as the query. This assumption is reasonable since the 10 categories were chosen so that each depicts a distinct semantic topic. Every image in the sub-database was tested as a query, and the retrieval ranks of all the rest images were recorded.

For each query, we computed the precision within the first 100 retrieved images. The recall within the first 100 retrieved images was not computed because it is proportional to the precision in this special case. The total number of semantically related images for each query is fixed to be 100. The average performance for each image category in terms of the average precision is listed in Table 1, where p denotes precision. For a system that ranks images randomly, the average p is about 0.1.

We carried out similar evaluation tests for color histogram match. We used LUV color space and a matching metric similar to the EMD described in [11] to extract color histogram features and match in the categorized image database. Two different color bin sizes, with an average of 13.1 and 42.6 filled color bins per image, were evaluated. We call the one with less filled color bins the Color Histogram 1 system and the other the Color Histogram 2 system. As shown in Table 1, both of the two color histogram-based matching systems perform much worse than the Lloyd-based system in almost all image categories. The performance of the Color Histogram 2 system is better than that of the Color Histogram 1 system due to more detailed color separation obtained with more filled bins. However, the Color Histogram 2 system is so slow that it is impossible to obtain matches on larger databases. The original SIMPLIcity runs at about twice the speed of the faster Color Histogram 1 system and gives much better searching accuracy than the slower Color Histogram 2 system.

The overall performance of the Lloyd-based system is close to that of the original system which uses IRM and area percentages of the segmented regions as significant constraints. Both the regular IRM and the fast IRM algorithms are much more accurate than the EMD-based color histogram. Experiments on a database of 70,000 pathology slides demonstrated similar comparison results.

4. CONCLUSIONS AND FUTURE WORK

We have developed a scalable integrated region-based image retrieval system. The system uses the IRM measure and the Lloyd algorithm. The algorithm has been implemented as part of the the IRM metric in our experimental SIMPLIcity image retrieval system. Tested on a database of about 200,000 general-purpose images, the technique has demonstrated high efficiency. The clustering efficiency can be improved by using a better statistical clustering algorithm. Better statistical modeling and matching scheme is likely to improve the matching accuracy of the system.

5. REFERENCES

[1] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, J. Malik, "Blobworld: a system for region-based image indexing and retrieval," *Proc. Int. Conf. on Visual Information Systems*, Amsterdam, The Netherlands, June, 1999.

[2] I. Daubechies, *Ten Lectures on Wavelets*, Capital City Press, 1992.

[3] A. Gupta, R. Jain, "Visual information retrieval," *Communications of the ACM*, 40(5), no. 5, pp. 70-79, May 1997.

[4] J. A. Hartigan, M. A. Wong, "Algorithm AS136: a k-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100-108, 1979.

[5] J. Li, J. Z. Wang, G. Wiederhold, "IRM: Integrated region matching for image retrieval," *Proc. ACM Multimedia*, pp. 147-156, Los Angeles, ACM, October, 2000.

[6] W. Y. Ma, B. Manjunath, "NaTra: A toolbox for navigating large image databases," *Proc. IEEE Int. Conf. Image Processing*, pp. 568-71, 1997.

[7] S. Mehrotra, Y. Rui, M. Ortega-Binderberger, T.S. Huang, "Supporting content-based queries over images in MARS," *Proc. IEEE Conf. on Multimedia Computing and Systems*, pp. 632-3, Ottawa, Ont., Canada 3-6 June 1997.

[8] S. Mukherjea, K. Hirata, Y. Hara, "AMORE: a World Wide Web image retrieval engine," *World Wide Web*, vol. 2, no. 3, pp. 115-32, Baltzer, 1999.

[9] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin, "The QBIC project: querying images by content using color, texture, and shape," *Proc. SPIE*, vol. 1908, pp. 173-87, San Jose, February, 1993.

[10] R. W. Picard, T. Kabir, "Finding similar patterns in large image databases," *Proc. IEEE ICASSP*, Minneapolis, vol. V, pp. 161-64, 1993.

[11] Y. Rubner, L. J. Guibas, C. Tomasi, "The earth mover's distance, Shimulti-dimensional scaling, and color-based image retrieval," *Proc. ARPA Image Understanding Workshop*, pp. 661-668, New Orleans, LA, May 1997.

[12] J. Shi, J. Malik, "Normalized cuts and image segmentation," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 731-7, San Juan, Puerto Rico, June, 1997.

[13] J. R. Smith, S.-F. Chang, "An image and video search engine for the World-Wide Web," *Proc. SPIE*, vol. 3022, pp. 84-95, 1997.

[14] J. R. Smith, C. S. Li, "Image classification and querying using composite region templates," *Journal of Computer Vision and Image Understanding*, vol. 75, no. 1-2, pp. 165-174, Academic Press, 1999.

[15] S. Stevens, M. Christel, H. Wactlar, "Informedia: improving access to digital video," *Interactions*, vol. 1, no. 4, pp. 67-71, 1994.

[16] J. Z. Wang, G. Wiederhold, O. Firschein, X. W. Sha, "Content-based image indexing and searching using Daubechies' wavelets," *International Journal of Digital Libraries*, vol. 1, no. 4, pp. 311-328, 1998.

[17] J. Z. Wang, J. Li, D. , G. Wiederhold, "Semantics-sensitive retrieval for digital picture libraries," *D-LIB Magazine*, vol. 5, no. 11, DOI:10.10 45/november99-wang, November, 1999. http://www.dlib.org

[18] J. Z. Wang, J. Li, R. M. Gray, G. Wiederhold, "Unsupervised multiresolution segmentation for images with low depth of field," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 1, pp. 85-90, 2001.