PERGAMON

Pattern Recognition 35 (2002) 593–600

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Negotiating the semantic gap: from feature maps to semantic landscapes

Rong Zhao, W.I. Grosky*

Department of Computer Science, Wayne State University, Detroit, MI 48202, USA

Received 16 November 2000; accepted 16 November 2000

Abstract

In this paper, we present the results of a project that seeks to transform low-level features to a higher level of meaning. This project concerns a technique, latent semantic indexing (LSI), in conjunction with normalization and term weighting, which have been used for full-text retrieval for many years. In this environment, LSI determines clusters of co-occurring keywords, sometimes, called concepts, so that a query which uses a particular keyword can then retrieve documents perhaps not containing this keyword, but containing other keywords from the same cluster. In this paper, we examine the use of this technique for content-based image retrieval, using two different approaches to image feature representation. We also study the integration of visual features and textual keywords and the results show that it can help improve the retrieval performance significantly. © 2001 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Content-based; Image retrieval; Semantic gap; Feature maps; Latent semantic indexing; Anglogram

1. Introduction

Existing management systems for image collections and their users are typically at cross-purposes. While these systems normally retrieve images based on low-level features, users usually have a more abstract notion of what will satisfy them. Using low-level features to correspond to high-level abstractions is one aspect of the *semantic gap* [1] between content-based system organization and the concept-based user. Sometimes, the user has in mind a concept so abstract that he himself does not know what he wants until he sees it. At that point, he may want images similar to what he has just seen or can envision. Again, however, the notion of similarity is typically based on high-level abstractions, such as activities taking place in the image or evoked emotions. Standard definitions of similarity using low-level features generally will not produce good results.

In reality, the correspondence between user-based semantic concepts and system-based low-level features is many-to-many. That is, the same semantic concept will usually be associated with different sets of image features. Also, for the same set of image features, different users could easily find dissimilar images relevant to their needs, such as when their relevance depends directly on an evoked emotion.

In this paper, we present the results of a project that seeks to transform low-level features to a higher level of meaning. This project concerns a technique, latent semantic indexing [2], which has been used for full-text retrieval for many years. In this environment, this technique determines clusters of co-occurring keywords,

[*] Corresponding author. Tel.: +1-313-577-0722; fax: +1-313-577-6868.

E-mail addresses: roz@cs.wayne.edu (R. Zhao), grosky@cs.wayne.edu (W.I. Grosky).

PII: S0031-3203(01)00062-0

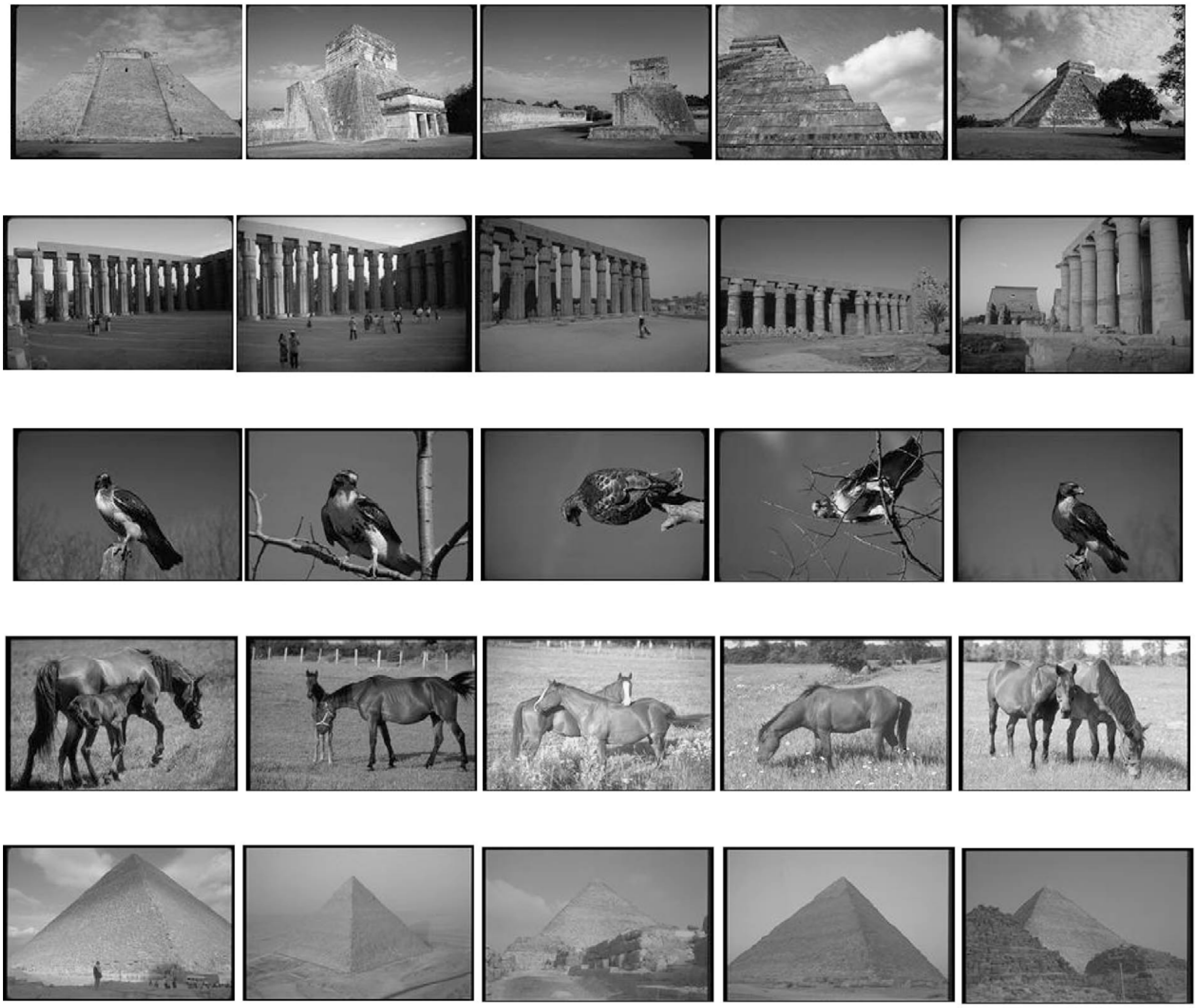


Fig. 1. (a)–(e) Ancient towers, ancient columns, birds, horses, pyramids; (f)–(j); Rhinos, sailing scenes, skiing scenes, sphinxes, sunsets.

sometimes, called *concepts,* so that a query which uses a particular keyword can then retrieve documents perhaps not containing this keyword, but containing other keywords from the same cluster. In this paper, we examine the use of this technique for content-based image retrieval.

The remainder of this paper is organized as follows. In Section 2, we present the results from some experiments using latent semantic indexing, normalization, and term weighting with global color histogram matching, while Section 3 presents similar results for color anglogram matching. In Section 4, we present some intriguing results concerning using image features with textual annotation. Finally, Section 5 presents our conclusions.

2. The effects of latent semantic indexing, normalization, and weighting for global color histograms

In this and the next section, we show the improvement that latent semantic indexing, normalization, and weighting can give to two simple and straightforward image retrieval techniques, both of which use standard color histograms. For our experiments, we use a database of 50 JPEG images, each of size 192 × 128. This image collection consists of ten semantic categories of five images each. The categories consist of: ancient towers, ancient columns, birds, horses, pyramids, rhinos, sailing scenes, skiing scenes, sphinxes, and sunsets. These images are shown in Fig. 1.

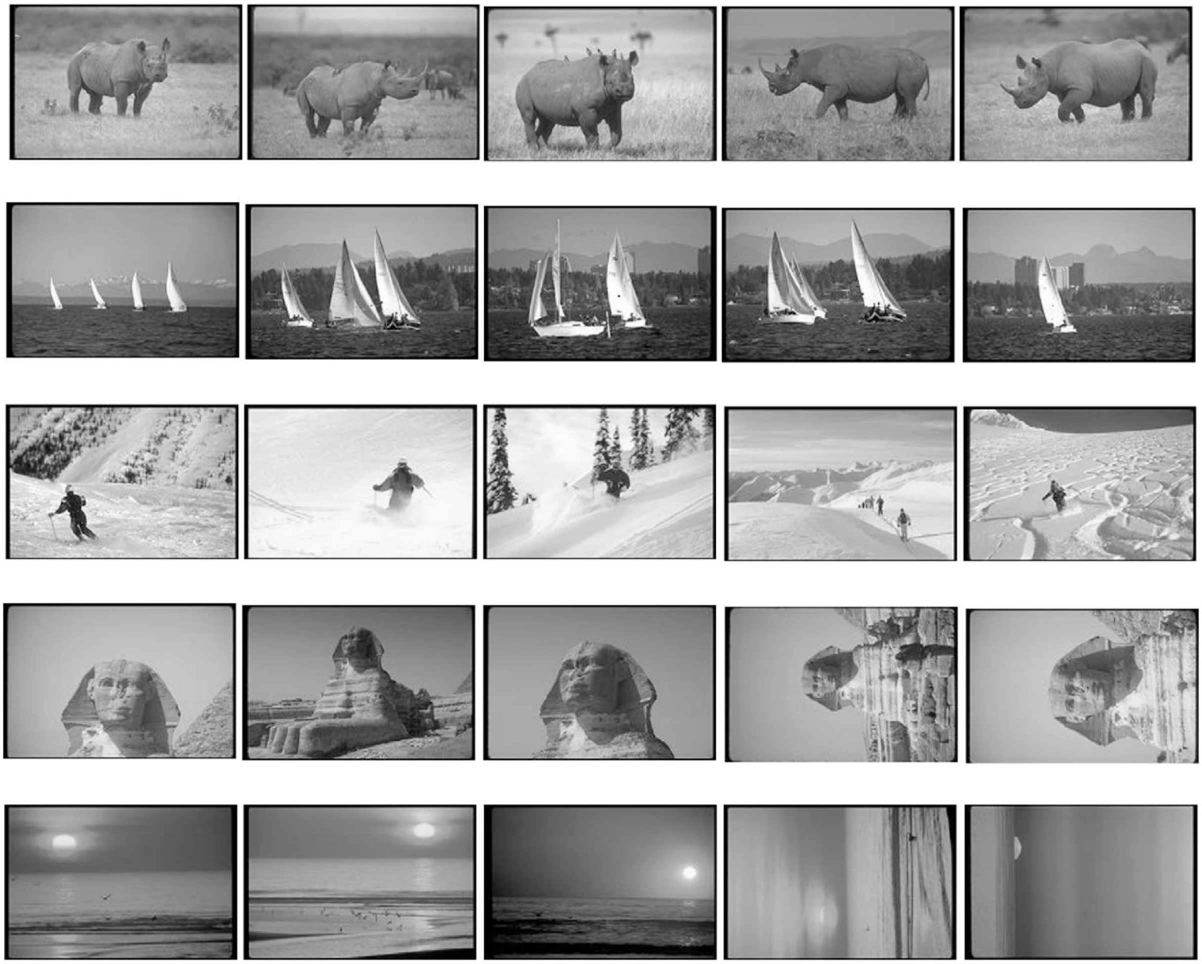


Fig. 1. (*Continued.*)

Our first approach uses global color histograms. Each image is first converted from the RGB color space to the HSV color space. For each pixel of the resulting image, hue and saturation are extracted and each quantized into a 10-bin histogram. Then the two histograms h and s are combined into one $h \times s$ histogram with 100 bins, which is the representing feature vector of each image. This is a vector of 100 elements, $\boldsymbol{V} = [f_1,\ f_2,\ f_3, \ldots, f_{100}]^{\mathrm{T}}$, where each element corresponds to one of the bins in the hue-saturation histogram.

We then generate the feature-image-matrix, $\boldsymbol{A} = [\boldsymbol{V}_1, \ldots, \boldsymbol{V}_{50}]$, which is 100×50. Each row corresponds to one of the elements in list of features and each column is the entire feature vector of the corresponding image. This matrix is written into a file so the computation is done only once. The matrix will be retrieved from the file during the query process.

A singular value decomposition (SVD) is then performed on the feature-image-matrix. The result comprises three matrices, $\boldsymbol{U}$, $\boldsymbol{S}$ and $\boldsymbol{V}$, where $\boldsymbol{A} = \boldsymbol{USV}^{\mathrm{T}}$. The dimensions of $\boldsymbol{U}$ is 100×100, $\boldsymbol{S}$ is 100×50, and $\boldsymbol{V}$ is 50×50. The rank of matrix $\boldsymbol{S}$, and thus the rank of matrix $\boldsymbol{A}$, in our case is 50. Therefore, the first 50 columns of $\boldsymbol{U}$ spans the column space of $\boldsymbol{A}$ and all the 50 rows in $\boldsymbol{V}^{\mathrm{T}}$ spans the row space of $\boldsymbol{A}$. $\boldsymbol{S}$ is a diagonal matrix of which the diagonal elements are the singular values of $\boldsymbol{A}$. To reduce the dimensionality of the transformed latent semantic space, we use a rank-k approximation, $\boldsymbol{A}_k$, of the matrix $\boldsymbol{A}$, for $k = 34$, which worked better than other values tried. This is defined by $\boldsymbol{A}_k = \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}_k^{\mathrm{T}}$. The dimension of $\boldsymbol{A}_k$ is the same as $\boldsymbol{A}$, 100×50. The dimensions of $\boldsymbol{U}_k$, $\boldsymbol{S}_k$, and $\boldsymbol{V}_k$ are 100×34, 34×34, and 50×34, respectively.

The query process in this approach is to compute the distance between the transformed feature vector of the query image, $\boldsymbol{q}$, and that of each of the 50 images in the database, $\boldsymbol{d}$. This distance is defined as *dist* $(\boldsymbol{q},\ \boldsymbol{d}) = \boldsymbol{q}^{\mathrm{T}}\boldsymbol{d}/||\boldsymbol{q}||\ ||\boldsymbol{d}||$, where $||\boldsymbol{q}||$ and $||\boldsymbol{d}||$ are the norms of those vectors. The computation of $||\boldsymbol{d}||$ for each of the 50 images is done only once and then written into a file. Using each image as a query, in turn, we find the average sum of the positions of all of the five correct answers. Note that in the best case, where the five correct matches occupy the first five positions, this average sum would be 15, whereas in the worst case, where the five correct matches occupy the last five positions, this average sum would be 240. A measure that we use of how good a particular method is defined as,

$$\textit{measure-of-goodness} = \frac{48 - (\textit{average} - \textit{sum})/5}{45}.$$

We note that in the best case, this measure is equal to 1, whereas in the worst case, it is equal to 0.

This approach was then compared to one without using latent semantic indexing. We also wanted to see whether the standard techniques of normalization and term weighting from text retrieval would work in this environment.

The following *normalization* process will assign equal emphasis to each component of the feature vector. Different components within the vector may be of totally different physical quantities. Therefore, their magnitudes may vary drastically and thus bias the similarity measurement significantly. One component may overshadow the others just because its magnitude is relatively too large. For the feature image matrix $\boldsymbol{A} = [\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_{50}]$, we have $\boldsymbol{A}_{i,j}$ which is the ith component in vector $\boldsymbol{V}_j$. Assuming a Gaussian distribution, we can obtain the mean μ_i and standard deviation σ_i for the ith component of the feature vector across the whole image database. Then we normalize the original feature image matrix into the range of $[-1, 1]$ as follows:

$$A_{i,j} = \frac{A_{i,j} - \mu_i}{\sigma_i}.$$

It can easily be shown that the probability of an entry falling into the range of $[-1, 1]$ is 68%. In practice, we map all the entries into the range of $[-1, 1]$ by forcing the out-of-range values to be either -1 or 1. We then shift the entries into the range of $[0, 1]$ by using the following formula:

$$A_{i,j} = \frac{A_{i,j} + 1}{2}.$$

After this normalization process, each component of the feature image matrix is a value between 0 and 1, and thus will not bias the importance of any component in the computation of similarity.

One of the common and effective methods for improving full-text retrieval performance is to apply different weights to different components [3]. We apply these techniques to our image environment. The raw frequency in each component of the feature image matrix, with or without normalization, can be weighted in a variety of ways. Both global weight and local weight are considered in our approach. A *global weight* indicates the overall importance of that component in the feature vector across the whole image collection. Therefore, the same global weighting is applied to an entire row of the matrix. A *local weight* is applied to each element indicating the relative importance of the component within its vector. The value for any component $\boldsymbol{A}_{i,j}$ is thus $L(i,j)G(i)$, where $L(i,j)$ is the local weighting for feature component i in image j, and $G(i)$ is the global weighting for that component.

Common local weighting techniques include term frequency, binary, and log of term frequency, whereas common global weighting methods include Normal, GfIdf, Idf, and Entropy. Based on previous research it has been found that log of term frequency, log (1 + term frequency), helps to dampen effects of large differences in frequency and thus has the best performance as a local weight, whereas Entropy is the appropriate method for global weighting [3].

The entropy method is defined by having a component global weight of

$$1 + \sum_j \frac{p_{ij}\log(p_{ij})}{\log(\textit{number\_of\_images})},$$

where

$$p_{ij} = \frac{tf_{ij}}{gf_i},$$

is the probability of that component, tf_{ij} is the raw frequency of component $\boldsymbol{A}_{i,j}$, and gf_i is the global frequency, i.e., the total number of times that component i occurs in the whole collection.

The global weights give less emphasis to those components that occur frequently or in many images. Theoretically, the entropy method is the most sophisticated weighting scheme and it takes the distribution property of feature components over the image collection into account.

We conducted similar experiments for these four cases:

1. Global color histograms, no normalization, no term weighting, no latent-semantic indexing (raw data).
2. Global color histograms, normalized and term-weighted, no latent semantic indexing.
3. Global color histograms, no normalization, no term-weighting, with latent semantic indexing.
4. Global color histograms, normalized and term-weighted, with latent semantic indexing.

The results can be represented as shown in Fig. 2, where the number in parenthesis is the measure-of-goodness of the particular method.

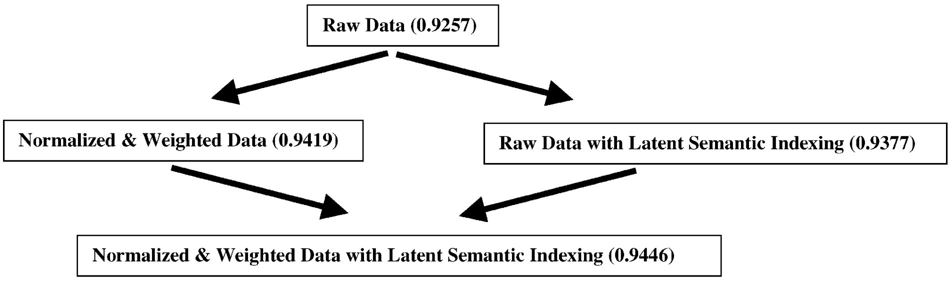

Raw Data (0.9257)
Normalized & Weighted Data (0.9419)
Raw Data with Latent Semantic Indexing (0.9377)
Normalized & Weighted Data with Latent Semantic Indexing (0.9446)

Fig. 2. Results for global color histogram representation.

Thus, for the global histogram approach, using normalized and weighted data or using latent semantic indexing with the raw data improves performance, while using both techniques is even better.

3. The effects of latent semantic indexing, normalization, and weighting for color anglograms

Our next approach performs similar experiments utilizing our previously formulated approach of color anglograms [4]. This is a novel spatial color indexing scheme based on the point feature map obtained by dividing an image evenly into a number of $M * N$ non-overlapping blocks with each individual block abstracted as a unique feature point labeled with its spatial location, dominant (average) hue, and dominant (average) saturation. An example is shown in Fig. 3. Fig. 3(a) shows a pyramid image of size 192×128. By dividing the image evenly into $16 * 16$ blocks, Fig. 3(b) and (c) show the image approximation using dominant hue and saturation values to represent each block, respectively. Fig. 3(d) shows the corresponding point feature map perceptually. Fig. 3(e) shows the resulting Delaunay triangulation of a set of feature points labeled with saturation 5, and Fig. 3(f) shows the corresponding anglogram (feature point histogram) obtained by counting only the two largest angles out of each individual Delaunay triangle. Such a feature point histogram provides a sufficient and effective way for image object discrimination.

For our experiments, we divide the images into $8 * 8$ blocks, have 10 quantized average hue value ranges and 10 quantized average saturation value ranges, count the two largest angles for each Delauney triangle, and have an anglogram bin of 5°. Our vector representation of an image thus has 720 elements: 36 hue bins for each of 10 hue ranges and 36 saturation bins for each of 10 saturation ranges. We use the same approach to querying as in the previous section.

We conducted similar experiments for these four cases:

5. Color anglograms, no normalization, no term weighting, no latent-semantic indexing (raw data).
6. Color anglograms, normalized and term-weighted, no latent semantic indexing.
7. Color anglograms, no normalization, no term-weighting, with latent semantic indexing.
8. Color anglograms, normalized and term-weighted, with latent semantic indexing.

The results can be represented as shown in Fig. 4, where the number in parenthesis is the measure-of-goodness of the particular method.

From these results, one notices that our anglogram method is better than the standard global color histogram, which is consistent with our previous results [4,5]. One also notices that latent semantic indexing improves the performance of this method. However, it seems that normalization and weighting has a negative impact on query performance. We more thoroughly examined the impact of these techniques and derived the data shown in Fig. 5.

The impact of normalization is worse than that of weighting. Normalization is a compacting process which transforms the original feature image matrix (the anglogram elements) to the range [0, 1]. Now, the feature image matrix in this case is a sparse matrix with many 0's, some small integers, and a relatively small number of large integers. We believe that these large integers represent the discriminatory power of the anglogram and that the compacting effect of normalization weakens their significance. Local log-weighting also has a compacting effect. Since both the local and global weighting factors lie between 0 and 1, the transformed matrix always has smaller values than the original one, even though no normalization is applied. Thus, normalization and weighting do not help improve the performance, but actually makes it worse.

4. Utilizing image annotations

We conducted various experiments to determine whether image annotations could improve the query results of our various techniques. The results indicate that they can.

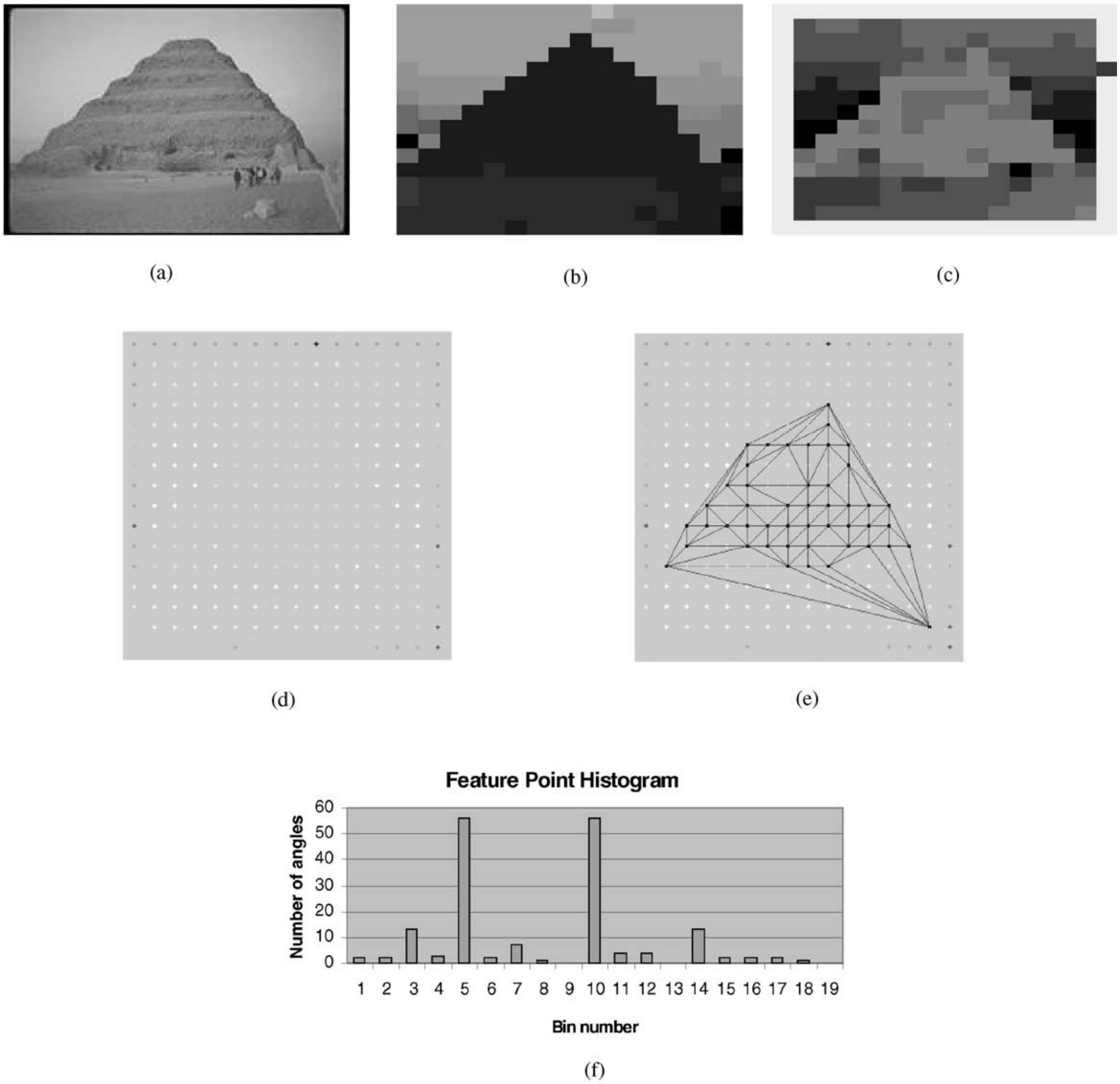
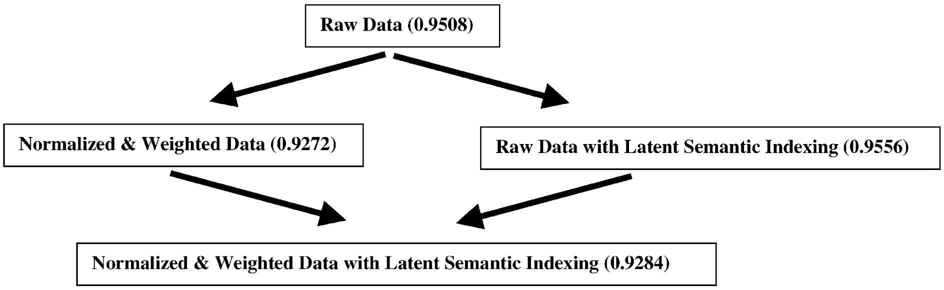


Fig. 3. (a) A pyramid image. (b) Hue component. (c) Saturation component. (d) Point feature map. (e) Delaunay triangulation of saturation 5. (f) Resulting anglogram of saturation 5.



Raw Data (0.9508)

Normalized & Weighted Data (0.9272)

Raw Data with Latent Semantic Indexing (0.9556)

Normalized & Weighted Data with Latent Semantic Indexing (0.9284)

Fig. 4. Results for color anglogram representation.

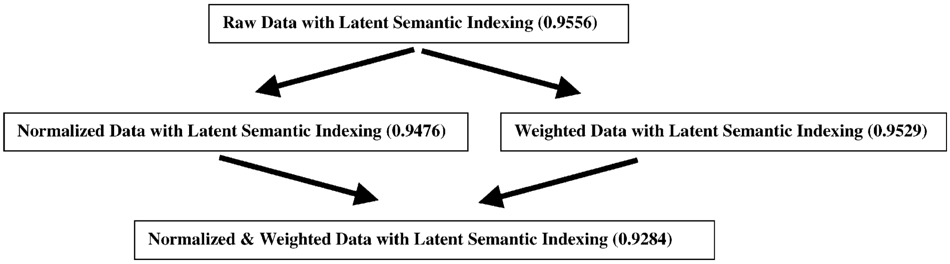

Raw Data with Latent Semantic Indexing (0.9556)
Normalized Data with Latent Semantic Indexing (0.9476)
Weighted Data with Latent Semantic Indexing (0.9529)
Normalized & Weighted Data with Latent Semantic Indexing (0.9284)

Fig. 5. A more detailed look at normalization and weighting.

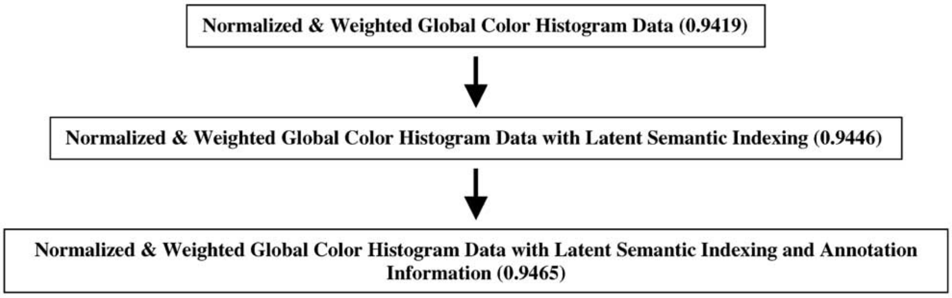

Normalized & Weighted Global Color Histogram Data (0.9419)
Normalized & Weighted Global Color Histogram Data with Latent Semantic Indexing (0.9446)
Normalized & Weighted Global Color Histogram Data with Latent Semantic Indexing and Annotation Information (0.9465)

Fig. 6. Global color histograms with annotation information.

For both the global color histogram and color anglogram representation, we appended an extra 15 elements to each of these vectors (called *category bits*) to accommodate the following 15 keywords associated with these images: *sky, sun, land, water, boat, grass, horse, rhino, bird, human, pyramid, column, tower, sphinx, and snow*. Thus, the feature vector for the global histogram representation now has 115 elements (100 visual elements and 15 textual elements), while the feature vector for the color anglogram representation now has 735 elements (720 visual elements and 15 textual elements). Each image is annotated with appropriate keywords and the area coverage of each of these keywords. For instance, one of the images is annotated with sky(0.55), sun(0.15), and water(0.30). This is a very simple model for incorporating annotation keywords.

One of the strengths of the LSI technique is that it is a vector-based method that helps us to integrate easily different features into one feature vector and to treat them just as similar components. Hence, ostensibly, we can apply the normalization and weighting mechanisms introduced in the previous sections to the expanded feature image matrix without any concern.

For the global color histogram representation, we start with an image feature matrix of size 115 × 50. Then, using the SVD, we again compute the rank 34 approximation to this matrix, which is also 115 × 50. For each query image, we fill bits 101 through 115 with 0's. We also fill the last 15 rows of the transformed image feature matrix with all 0's. Thus, for the querying, *we do not use any annotation information*. We also note, that as before, we apply normalization and weighting, as this improves the results, which are shown in Fig. 6. The first two results are from Fig. 2, while the last result shows how our technique of incorporating annotation information improves the querying process.

For the color anglogram representation, we start with an image feature matrix of size 735 × 50. Then, using the SVD, we again compute the rank 34 approximation to this matrix, which is also 735 × 50. For each query image, we fill bits 721 through 735 with 0's. We also fill the last 15 rows of the transformed image feature matrix with all 0's. Thus, for the querying, *we do not use any annotation information*. We also note, that as before, we do not apply normalization and weighting, as this improves the results, which are shown in Fig. 7. The first two results are from Fig. 4, while the last result shows how our technique of incorporating annotation information improves the querying process.

Note that annotations improve the query process for color anglograms, even though we do not normalize the various vector components, nor weight them. This is quite surprising, given that the feature image vector consists of 720 visual elements, some of which are relatively large

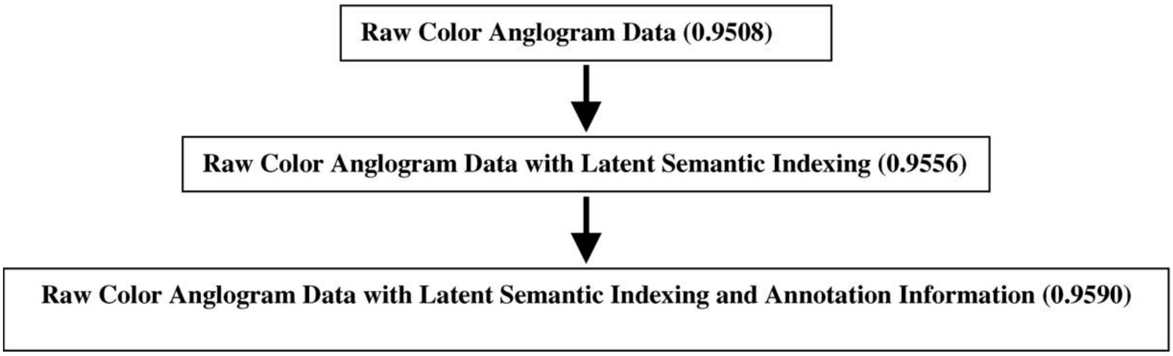

Raw Color Anglogram Data (0.9508)
Raw Color Anglogram Data with Latent Semantic Indexing (0.9556)
Raw Color Anglogram Data with Latent Semantic Indexing and Annotation Information (0.9590)

Fig. 7. Color anglograms with annotation information.

integers, and only 15 annotation elements, which are in the range [0, 1].

5. Conclusions

Clearly, while LSI seems to improve the results of our content-based retrieval experiments, this improvement is not great, perhaps due to the small size of our image collection. The results presented in Section 4 are quite interesting and are certainly worthy of further study. Our hope is that latent semantic indexing will find that different image features co-occur with similar annotation keywords, and consequently lead to improved techniques of semantic image retrieval. We are currently experimenting with various clustering techniques for images where we use the cluster identifier in place of annotation information.

References

[1] V. Gudivada, V.V. Raghavan, Content-based image retrieval systems, IEEE Comput. 28 (9) (1995) 18–22.

[2] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inform. Sci. 41 (6) (1990) 391–407.

[3] S. Dumais, Improving the retrieval of information from external sources, Behav. Res. Methods Instrum. Comput. 23 (2) (1991) 229–236.

[4] Y. Tao, W.I. Grosky, Spatial color indexing using rotation, translation, and scale invariant anglograms, Multimedia Tools Appl., in press.

[5] Y. Tao, W.I. Grosky, Spatial color indexing: a novel approach for content-based image retrieval. Proceedings of the IEEE Sixth International Conference on Multimedia Computing and Systems (ICMCS'99), Florence, Italy, June 7–11, 1999, pp. 530–535.

About the Author—WILLIAM I. GROSKY is currently professor and chair of the Computer Science Department at Wayne State University in Detroit, Michigan. Before joining Wayne State in 1976, he was an assistant professor of Information and Computer Science at Georgia Institute of Technology in Atlanta. His current research interests are in multimedia information systems, hypermedia, databases, and web technology. Grosky received his B.S. in mathematics from MIT in 1965, his M.S. in Applied Mathematics from Brown University in 1968, and his Ph.D. from Yale University in 1971. He has given many short courses in the area of database management for local industries and has been invited to lecture on multimedia information systems world-wide. Serving also on many database and multimedia conference program committees, he is currently the Editor-in-Cheif of IEEE Multimedia, and on the editorial boards of the Journal of Database Management and Pattern Recognition.

About the Author—RONG ZHAO received his Bachelor of Engineering degree in Computer Science and Technology from Tsinghua University, P. R. China, in 1996. He is currently a Ph.D. candidate in the Computer Science Department at Wayne State University. His research interests include multimedia information retrieval, digital library, data mining, and web document prefetching.