

From Features to Semantics: Some Preliminary Results

Rong Zhao and W.I. Grosky
Department of Computer Science
Wayne State University
Detroit, MI 48202, USA
{roz,grosky}@cs.wayne.edu

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 analysis (LSA), which has been used for full-text retrieval for many years. In this environment, LSA 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.

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* [GuR95] between content-based system organization and the concept-based user. Sometimes, the user has in mind a concept so abstract that he himself doesn't 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 analysis [DDF90], which has been used for full-text retrieval for many years. In this environment, this technique 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.

The remainder of this paper is organized as follows. In Section 2, we present the results from some experiments using latent semantic analysis with global color histogram matching, while Section 3 presents similar results for subimage color histogram matching. In Section 4, we present some intriguing preliminary results concerning using image features with textual annotations, in the context of relevance feedback. Finally, Section 5 presents our conclusions.

2. Latent Semantic Analysis and Content-Based Image Retrieval – Global Color Histograms

In this and the next section, we show the improvement that latent semantic analysis 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, sphinxes, and sunsets. One image from each semantic category is shown below.



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, $\mathbf{V} = [f_1, f_2, f_3, \dots, f_{100}]^T$, where each element corresponds to one of the bins in the hue-saturation histogram.

We then generate the feature-image-matrix, $\mathbf{A} = [\mathbf{V}_1, \dots, \mathbf{V}_{50}]$, which is 100×50 . Each row corresponds to one of the elements in the feature vector and each column is the whole 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 is then performed on the feature-image-matrix. The result comprises three matrices, \mathbf{U} , \mathbf{S} and \mathbf{V} , where $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. The dimensions of \mathbf{U} are 100×100 , \mathbf{S} is 100×50 , and \mathbf{V} is 50×50 . The rank of matrix \mathbf{S} , and thus the rank of matrix \mathbf{A} , in our case is 50. Therefore, the first 50 columns of \mathbf{U} spans the column space of \mathbf{A} and all the 50 rows in \mathbf{V}^T spans the row space of \mathbf{A} . \mathbf{S} is a diagonal matrix of which the diagonal elements are the singular values of \mathbf{A} . To reduce the dimensionality of the transformed latent semantic space, we use a rank- k approximation, \mathbf{A}_k , of the matrix \mathbf{A} , for $k = 34$. This is defined by $\mathbf{A}_k = \mathbf{U}_k \mathbf{S}_k \mathbf{V}_k^T$. The dimension of \mathbf{A}_k is the same as \mathbf{A} , 100 by 50. The dimensions of \mathbf{U}_k , \mathbf{S}_k , and \mathbf{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, \mathbf{q} , and that of each of the 50 images in the database, \mathbf{d} . This distance is defined as $dist(\mathbf{q}, \mathbf{d}) = \mathbf{q}^T \mathbf{d} / \|\mathbf{q}\| \|\mathbf{d}\|$, where $\|\mathbf{q}\|$ and $\|\mathbf{d}\|$ are the norms of those vectors. The computation of $\|\mathbf{d}\|$ for each of the 50 images is done only once and then written into a file.

This approach was compared to one without using latent semantic analysis. The extracted features of each image results in the same hue-saturation histogram \mathbf{V} as above. However, for the similarity measure between a query image, \mathbf{q} , and a database image, \mathbf{d} , we use histogram intersection. Given the feature vectors $\mathbf{V}\mathbf{q}$ of a query image and $\mathbf{V}\mathbf{d}$ of an image in the database, this measure is defined as follows:

$$dist(q, d) = \frac{\sum_{i=1}^{100} \min(Vq_i, Vd_i)}{\sum_{i=1}^{100} Vq_i}$$

We conducted experiments in which each database image was considered as a query. Results ranged from having,



as the query image and retrieving, not using latent semantic analysis, the following images as the first, fourth, fifth, seventh, and fourteenth best matches,



while retrieving, using latent semantic analysis, the following images as the first, fourth, fifth, sixth, and tenth best matches,



to having,



as the query image and retrieving, not using latent semantic analysis, the following images as the first, second, fourth, seventh, and seventeenth best matches,

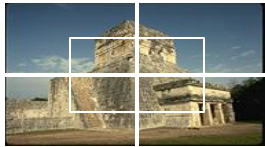


while retrieving, using latent semantic analysis, the same corresponding images as the first, third, fourth, twenty-second, and twenty-fifth best matches.

Using each image as a query, 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. Now, without using latent semantic analysis, this average sum was 30.06, while the use of latent semantic analysis brings this average sum to 29.16, an improvement of almost one position.

3. Latent Semantic Analysis and Content-Based Image Retrieval – Sub-Image Matching

Our next approach uses sub-image matching in conjunction with color histograms. Each image is first converted from the RGB color space to the HSV color space. Each image is decomposed into 5 overlapping subimages, as shown below,



Such as approach was used in [StD96], and is a step toward identifying the semcons [GFJ98] appearing in an image. For the 50 images in our case, 250 subimages will be used in the following feature extraction process. 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, $\mathbf{V} = [f_1, f_2, f_3, \dots, f_{100}]^T$, where each element

corresponds to one of the bins in the hue-saturation histogram.

We then generate the feature-subimage-matrix, $\mathbf{A} = [\mathbf{V}_1, \dots, \mathbf{V}_{250}]$, which is 100×250 . Each row corresponds to one of the elements in the feature vector and each column is the whole feature vector of the corresponding subimage. 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 is then performed on the feature-subimage-matrix. The result comprises three matrices, \mathbf{U} , \mathbf{S} and \mathbf{V} , where $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. The dimensions of \mathbf{U} are 100×100 , \mathbf{S} is 100×250 , and \mathbf{V} is 250×250 . The rank of matrix \mathbf{S} , and thus the rank of matrix \mathbf{A} , in our case is 100. Therefore, the first 100 columns of \mathbf{U} spans the column space of \mathbf{A} and all the 100 rows in \mathbf{V}^T spans the row space of \mathbf{A} . \mathbf{S} is a diagonal matrix of which the diagonal elements are the singular values of \mathbf{A} . To reduce the dimensionality of the transformed latent semantic space, we use a rank- k approximation, \mathbf{A}_k , of the matrix \mathbf{A} , for $k = 55$. This is defined by $\mathbf{A}_k = \mathbf{U}_k \mathbf{S}_k \mathbf{V}_k^T$. The dimension of \mathbf{A}_k is the same as \mathbf{A} , 100 by 250. The dimensions of \mathbf{U}_k , \mathbf{S}_k , and \mathbf{V}_k are 100×55 , 55×55 , and 250×55 , respectively.

The first step of the query process in this approach is to compute the distance between the transformed feature vector of each subimage of the query image, \mathbf{q} , and that of each of the 250 images in the database, \mathbf{d} . This distance is defined as $dist(\mathbf{q}, \mathbf{d}) = \mathbf{q}^T \mathbf{d} / \|\mathbf{q}\| \|\mathbf{d}\|$, where $\|\mathbf{q}\|$ and $\|\mathbf{d}\|$ are the norms of those vectors. The computation of $\|\mathbf{d}\|$ for each of the 250 subimages is done only once and then written into a file.

With respect to the query image and each of the 50 database images, we now have the distances between each pair of subimages by the previous step. These distance values $dist(\mathbf{q}, \mathbf{d})$ are then combined into one distance value between these two images in an approach similar to the computation of Euclidean distance. Given a query image \mathbf{q} , with corresponding subimages q_1, \dots, q_5 , and a candidate database image \mathbf{d} , with corresponding subimages d_1, \dots, d_5 , we define,

$$dist(q, d) = \frac{1}{5} \sqrt{\sum_{i=1}^5 [dist(q_i, d_i)]^2}$$

This approach was again compared to one without using latent semantic analysis. Each image is decomposed into five subimages which are then represented by their hue-saturation histograms \mathbf{V} . Histogram intersection is computed and used as the similarity metric. Given the feature vector \mathbf{V}_q of a subimage, \mathbf{q} , of a query image, \mathbf{q} , and \mathbf{V}_d of the corresponding subimage, \mathbf{d} , of a database image, \mathbf{d} , this measure is defined as follows,

$$sim(q_i, d_i) = \frac{\sum_{j=1}^{100} \min(vq_{i,j}, vd_{i,j})}{\sum_{j=1}^{100} vq_{i,j}}$$

We thus have the distance between the query image and each of the 50 database images. These similarity values are then combined into one similarity measure between these two images. Given a query image \mathbf{q} and a candidate image \mathbf{d} in the database, we define,

$$dist(q, d) = \frac{1}{5} \sum_{i=1}^5 sim(q_i, d_i)$$

Using each image as a query, we again find the average sum of the positions of all of the five correct answers. Now, without using latent semantic analysis, this average sum was 27.32, while the use of latent semantic analysis brings this average sum to 26.2, an improvement of more than one position.

We also did a similar experiment where $dist(\mathbf{q}, \mathbf{d})$ weighted the center subimage twice as much as the peripheral subimages. The results of these experiments, were 26.82 for the average sum without using latent semantic analysis and 26.14 for the average sum using latent semantic analysis.

4. Utilizing Image Annotations

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

Using global color histograms, recall that each image was represented as a 100 element vector. We appended an extra 10 elements to each of these vectors (called *category bits*), one for each semantic category. Images in the j^{th} semantic category had a 1 in the $(100+j)^{\text{th}}$ bit position and a 0 in all other bit positions between 101 and 110. This is a very simple model for incorporating annotation keywords.

We then used each image as a query, filling bits 101 through 110 with 0's in this query image. Thus, for the querying, we did not use any annotation information.

Using each image as a query, we again find the average sum of the positions of all of the five correct answers. Through the use of latent semantic analysis on these expanded vectors, our average sum becomes 28.88, an improvement, albeit small, over the previous result of 29.16.

We then utilized relevance feedback, asking the user to choose the top two matches. If these two images had the same category bit set (positions 101-110), we then changed the query image by also setting the appropriate category bit. In our examples, since at least two images shown to the user always come from the same category as

the query image, the category bit was always set in the query image for next round of relevance feedback.

Using the transformed query, we then repeated the experiment. The result was that we always found the five correct matches in the first five positions, producing an average sum of 15. This is a very intriguing result.

5. Conclusions

Clearly, while LSA 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 fact that subimage matching is improved more than whole image matching, seems to indicate that more powerful semcon decomposition techniques will show even greater improvement using LSA.

The results presented in Section 4 are quite interesting and are certainly worthy of further study. Our hope is that latent semantic analysis will find that different image features co-occur with similar annotation keywords, and consequently lead to improved techniques of semantic image retrieval.

References

- [DDF90] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, 'Indexing by Latent Semantic Analysis,' *Journal of the American Society for Information Science*, Volume 41, Number 6 (1990), pp. 391-407.
- [GFJ98] W.I. Grosky, F. Fotouhi, and Z. Jiang, 'Using Metadata for the Intelligent Browsing of Structured Media Objects,' In *Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data*, A. Sheth and W. Klas (Eds.), McGraw Hill Publishing Company, New York, 1998, pp. 123-148.
- [GuR95] V. Gudivada and V.V. Raghavan, 'Content-Based Image Retrieval Systems,' *IEEE Computer*, Volume 28, Number 9 (September 1995), pp. 18-22.
- [StD96] M. Stricker and A. Dimai, 'Color Indexing with Weak Spatial Constraints,' *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, Volume 2670, February 1996, pp. 29-39.