VISMAP: AN INTERACTIVE IMAGE/VIDEO RETRIEVAL SYSTEM USING VISUALIZATION AND CONCEPT MAPS

William Chen and Shih-Fu Chang

{bchen, sfchang}@ee.columbia.edu
Department of Electrical Engineering
Columbia University
New York, NY 10027, USA

ABSTRACT

Images and videos can be indexed by multiple features at different levels, such as color, texture, motion, and text annotation. Organizing this information into a system so that users can query effectively is a challenging and important problem. In this paper, we present VISMap, a visual information seeking system that extends the traditional query paradigms of *query-by-example* and *query-by-sketch* and replaces the models of *relevance feedback* with principles from *information visualization* and *concept representation*. Users no longer perform lengthy "one-shot" queries or rely on hidden relevance feedback mechanisms. Instead, we provide a rich set of tools that allow users to construct personal views of the video database and directly visualize and manipulate various views and comprehend effects of individual query criteria on the final search results. The set of tools include: 1) a feature space browser for feature-based exploration and navigation, 2) a distance map for metric comparison and setting and 3) a novel concept map for query representation and creation.

1. INTRODUCTION

Currently, most content-based visual query (CBVQ) systems, such as [1,2], require the user to begin the search process by formulating a query. Such systems prompt the user to query by keyword, query by visual example, or query by sketch. The result of the search is a sorted list of images that match the keyword or are similar to the query, where similarity is measured by a predefined metric over a set of features. However, the text annotations may not be available and the similarity search results often do not satisfy the user's desired image because 1) it is difficult for the user to fully describe an image or video by low-level features, and 2) it is difficult for the user to predict the discriminative capability of different features and their similarity metrics.

Recent CBVQ systems, such as [3,4], employed relevance feedback to reduce the mismatch between the user's desired image and the return results. User's feedback in terms of relevance of the returned images is used to refine the queries, weights, or likelihood of the candidate images. While relevance feedback shows promise, there still exists a significant gap between the meaning of images and the visual features extracted from images.

Our solution is to forego the assumption that the user has a target image in mind and the system is able to match the semantic meaning of the target unambiguously. Instead, we adopt the information visualization paradigm [5] for the user to directly visualize and control the image and query spaces. We construct a system, VISMap, which combines content-based retrieval techniques with those of information visualization. VISMap now includes a rich set of tools that allow users to construct, visualize, and manipulate personal views of the video database and comprehend effects of individual query criteria on the final search results.

Previous systems [6,7,8] that combine information visualization with CBVQ have highlighted only one aspect of the problem. In [6], the authors have developed a direct manipulation interface for image query. The interface lets users place semantically similar images close together in order to create a similarity criterion based on this placement. In the Informedia project [7], visualization techniques are adopted to display complex queries and their relationships in a two-dimensional display. Finally, in [8], the authors have developed a 3D visualization scheme in which users navigate high-dimensional feature spaces in search of images. Our contribution is to provide an integrated visualization scheme that gives users greater intuition in every step of the search process.

In Section 2, we discuss the principles of information visualization and its application to CBVQ. In Section 3, we introduce VISMap and describe how VISMap handles different types of CBVQ. We present conclusions in Section 4.

2. INFORMATION VISUALIZATION

One important aspect of information visualization deals with how to display large amounts of information in ways that users can quickly understand. Many times this involves embedding the information in a metric space. The information can then be represented as feature values in an n-dimensional vector space. To display this information, it is then necessary to transform the vector space to a lower dimension while preserving some of its structural properties. Given such a space, the user can browse the data and zoom in to important regions on demand.

Another aspect of information visualization deals with how to manage complex queries. A complex query is composed of N concepts. Each concept is a separate query into a subsystem with its own similarity metric, relevance weight, and threshold setting. The concepts are combined by explicit rules to determine the final return result. Information visualization can be used to map concepts as well as their relationships graphically. Given such a mapping, the user can explore individual concepts and manipulate their relationships to find the desired return result.

Given these properties, we argue that information visualization is a natural fit to content-based video search systems. Such systems extract features from many sources. Some features, such as color and texture, can be represented as values in a vector space. Some features such as spatial relationships among multiple objects require “join” of multiple spaces. Others such as text do not have an explicit spatial property. High-level tools will be needed to explore, manipulate, and relate such disparate information. In VISMap, we have constructed tools in which users can browse information spaces, explore metric properties and settings, and create concepts and relationships. Such visual tools 1) give personalized views of the video database and 2) show the interplay between query components, similarity metrics, and return results to reduce ambiguity in the image/video search process.

3. VISMAP

In this section, we describe the backend search engine, the feature library, and the tools and interfaces in VISMap. We also demonstrate how VISMap can be used to formulate typical content-based image/video queries.

3.1. VideoQ Database

VISMap can be used as the front end in conjunction with any visual search system that indexes images/videos by visual features and text annotations. Currently, we connect VISMap to the VideoQ video search system. The VideoQ project at Columbia University has a collection of professional stock footage that covers a broad range of topics. In previous work, we have presented a novel content-based video search and retrieval system that captures the spatio-temporal characteristics of video. Through an automatic segmentation and tracking algorithm, we extract video objects as well as global scene features.

Video objects are composed of static features, such as color, texture, and shape, and dynamic features, such as motion. Each feature, such as texture, can have multiple representations, such as Tamura or cooccurrence texture. Global features include camera motion, scene complexity, text metadata, and semantic visual templates. In VISMap, each feature is indexed separately using the spatial access method R-tree. Similarity queries are supported by efficient range searches over the R-tree index. Note that an efficient indexing scheme is critical in providing fast responses in the interactive paradigm like VISMap.

3.2. Feature, Distance, and Display Views

The VISMap GUI is divided into areas by functionality. Each area is composed of a set of tools, such as filters, classifiers, templates, browsers, canvases, etc. Users can select tools to perform range queries and adjust range threshold values. Users can also combine tools to create Boolean queries (AND,OR).

VISMap provides three important views to allow users to build personal views of the video database. The three views are *feature space browser*, *distance map*, and *concept map*. The feature space browser is created offline and shows video object features, such as color, texture, and combinations of visual features. The distance map and concept map are created dynamically with user-selected query components and threshold settings. Figure 3 shows conceptual appearances of the three views, details of which are described later.

3.2.1 Feature Space Browser

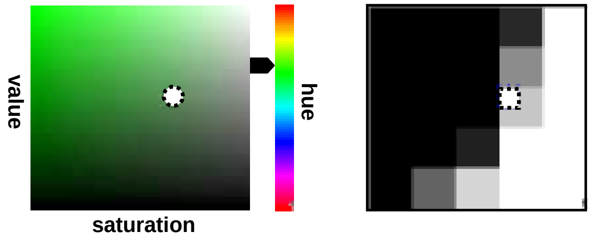

Figure 1. The color map on the left shows saturation vs value for a given hue. The density map on the right shows the probability of finding video objects for given color value.

A feature space browser visualizes feature values in a space. As shown in Figure 1, the color values of video objects are plotted in a two-dimensional display. Users can browse the feature space and select feature values <f_0,f_1,f_2> to query and retrieve similar video objects. VISMap tries to maintain intuitive dimensions, such as value vs. saturation for a given hue of color. In the case of high dimensionality (e.g., color histogram), feature space projection methods can be added, at the cost of losing the intuitive representation.

Also shown in Figure 1, we use a density map to increase the likelihood that users will encounter results in the feature space. A density map is a map of gray-scale values. The gray-scale values indicate the probability *p* of finding video objects at the corresponding selected feature value. Brighter color indicates higher probability.

3.2.2 Distance Map

The distance map visualizes the distance space. The user chooses two query components (q_i, q_j) at a time. These are used as the axes of a two-dimensional map. The return results are then plotted as points whose coordinate along each axis is the distance value to the corresponding query component. For example, a video object with a distance value of d_0=0.3 from q_0 and d_1=0.5 from q_1 would map as a single point with coordinates (d_0,d_1) in the distance map. Furthermore, the setting of threshold values becomes enclosed rectangular regions in the distance map. So that the threshold setting of $d_0 < T_0$ and $d_1 < T_1$ would return all points within the rectangle defined by (0,0) and (T_0,T_1).

3.2.3 Concept Map

The concept map is a two-dimensional graphical representation of the query components and the return results. As shown in Figure 2, both text-based and feature-based queries components are represented graphically in the concept map. Each query component forms an anchor that is placed by the user at position a_i. Each return result is plotted as a blue square, and its position p_k is determined by both the position and similarity score to each query:

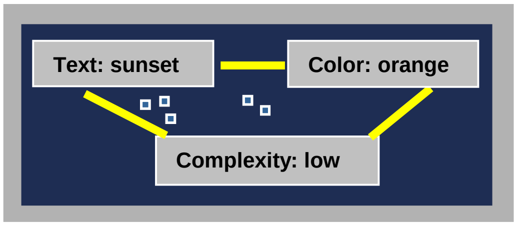

Figure 2. Concept map shows a query for an orange sunset with low complexity. Results are plotted as square dots.

$$p_k = \frac{\Sigma\, s_i * a_i}{\Sigma\, s_i}$$

where a_i is the (x,y) position of query q_i in the concept map and s_i is the similarity score (or 1-d_i) to q_i.

The concept map shows the user the influence of a query component in comparison with others. For example, a return result that is equally similar to all three query components in Figure 2 would be plotted in the center of the triangle. Movement away from the center marks the increasing similarity to one query component over others. Note that the above equation can only indicate relative not absolute similarity to each query.

3.3. Mapping Queries to Views

Typical CBVQ usually fall under three categories: single entity, multiple examples; single entity, multiple attributes; and multiple entities, multiple attributes. We consider mapping each of these queries to VISMap views.

3.3.1 Single entity, multiple examples

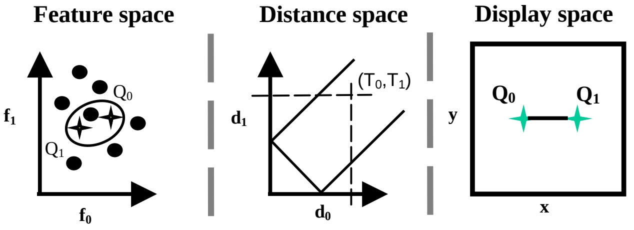

Figure 3. Conceptual appearances of three views for single entity, multiple examples query.

A single entity, multiple examples query is a query in which the user provides the system with multiple examples of the desired image/video. For instance, the user can retrieve a mountain scene similar to images I_0, I_1, .., I_k. In addition to this, certain relevance feedback schemes allow the user to refine the search with multiple positive examples from the previous return set.

As shown in Figure 3, we can conceptually visualize this type of query in the feature space as searching for images that are “close” to both query points Q_0 and Q_1. The feature space nicely shows the distribution of images around the query points. In the distance space, the images are plotted at coordinates (d_0,d_1), according to their distances from each query. Note all images will map to within the infinite half-plank. Closeness can be defined here as images that are within the threshold values (T_0,T_1) set by the user. In the display space, all images will map to points on the line joining Q_0 and Q_1. Ideal return results are at the center of the line while those off-center are more similar to one query over another.

This system of views scales as well. For high-dimensional feature spaces, creative visualizations or projection operations can be used to reduce dimensionality. For multiple queries (greater than 2), the user must select two at a time for the distance space view. The display space can handle any arbitrary number of query points.

3.3.2 Single entity, multiple attributes

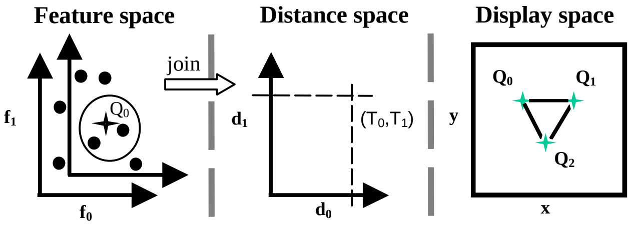

Figure 4. Conceptual appearances of three views for single entity, multiple attributes query.

A single entity, multiple attributes query is a query in which the user specifies multiple characteristics of a single image, video, or video object. For example, the user can retrieve a sunset scene by giving the label ‘sunset’ and the color orange.

As shown in Figure 4, each attribute that is required is submitted as a query into a separate subsystem. For a query with three attributes, the query points Q_0, Q_1, and Q_2 are mapped to separate feature spaces. In each feature space, results close to the query point are retrieved. A join operation then follows. For an image or video, the join operation returns the images/videos close to all query points. For video objects, the join operation returns the video objects close to all query points. In the distance space, results are mapped in the full plane according to their distances to each query. Once again, thresholds are set by the user to determine closeness. In the display space, the results are now mapped within a triangle with points in the center of the triangle being equally similar to all query points.

3.3.3 Multiple entities, multiple attributes

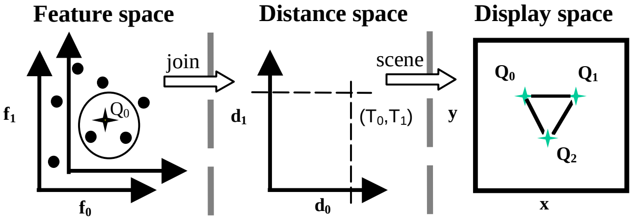

Figure 5. Conceptual appearances of three views for multiple entities, multiple attributes query.

A multiple entities, multiple attributes query is a query in which the user specifies multiple regions and multiple characteristics for each region. For example, the user can retrieve a sunset scene by sketching the sky and the sun. The sky is given color and texture attributes, and the sun is given color and shape attributes.

As shown in Figure 5, each specified region is now a query point. For a query with three regions, the query points Q_0, Q_1, and Q_2 are mapped to separate feature spaces. Similar to the previous section, in each feature space, results close to the query point are retrieved. The join operation differs. In this case, the join operation returns images/videos that contain all three regions, one from each of the feature spaces. In the distance space, images/videos are mapped in the full plane according to distance of the matched region to each query. Once again, thresholds are set by the user to determine closeness. A scene operation then follows. Spatial relationships and global scene filters must be satisfied. In the display space, the results are now mapped within a triangle with points in the center of the triangle being equally similar to all query points.

4. CONCLUSIONS

VISMap visualizes the key steps in the search process in order to give the user greater intuition and control in how to formulate queries and explore the video database. We have completed the prototype implementation of VISMap. We are currently in the process of testing VISMap on a number of query scenarios and conducting preliminary user studies for effectiveness.

5. REFERENCES

[1] S.F. Chang, W. Chen, J. Meng, H. Sundaram and D. Zhong, "VideoQ: An Automated Content-Based Video Search System Using Visual Cues," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 5, September 1998.

[2] M. Flickner, *et al*, "Query by image and video content: The QBIC system," *IEEE Computer Magazine*, Vol. 28, pp. 23-32, September 1995.

[3] I.J. Cox, M.L. Miller, S.M. Omohundro, P.N. Yianilos, “PicHunter: Bayesian Relevance Feedback for Image Retrieval,” *International Conference on Pattern Recognition*, Vienna, Austria, 1996.

[4] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," *IEEE Transactions on Circuits and Video Technology*, 8(5):644-655, September 1998.

[5] Ahlberg, C. and Shneiderman, B., Visual Information Seeking: Tight coupling of dynamic query filters with starfield displays , Proc. of ACM CHI94 Conference, (April 1994), 313-317.

[6] S. Santini and R. Jain, "Beyond Query by Example," *ACM Multimedia '98*, Bristol, UK.

[7] M. Christel and D. Martin, "Information Visualization Within a Digital Video Library,” *Journal of Intelligent Information Systems* 11(3); 235-257 (1998).

[8] A. Hiroike, et al “Visualization of the information space to retrieve and browse image data base,” *Third International Conference on Visual Systems*, Amsterdam, Netherlands, June 1999.