

3D MARS: Immersive Virtual Reality for Content-Based Image Retrieval

Munehiro Nakazato and Thomas S. Huang

Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
{nakazato, huang}@ifp.uiuc.edu

ABSTRACT

3D MARS is an interactive visualization system for Content-Based Image Retrieval (CBIR). In *3D MARS*, the user browses and queries images in an immersive 3D Virtual Reality space of CAVE. The results of the query are displayed in 3D, so that the user can see the result with respect to three different criteria such as color, texture and structure. The user can examine the result from different view angles by flying-through the space with the joystick. Based on the user's feedback, the system dynamically reorganize its visualization scheme. By giving meanings to each axis, the user can determine which features are important. In addition, the Sphere mode visualization is provided as a powerful analyzing tool for CBIR researchers.

Keywords

Content-Based Image Retrieval, Relevance Feedback, Virtual Reality, Information Visualization

1. INTRODUCTION

By the advent of inexpensive digital cameras and popularity of World-Wide Web, digital image became very common. While Content-base Image Retrieval (CBIR) systems [4][8] relieve the user of tedious tasks of organizing and searching digital images, they have a significant limitation.

In traditional Content-Based Image Retrieval systems, the query results are ordered and displayed in a line (i.e. 1D) based on the weighted sum of the distance measures. Meanwhile, the image features consist of high-dimensional vectors of different image properties such as color, texture and structure. Thus, much information is lost for visualization. This cause problems especially when the number of query examples is small. The system cannot tell which feature is the most important for a user. Consequently, the most important image may not appear in the early stage of relevance feedback. One solution to this problem is to allow the user to adjust the query parameter as often used in other image retrieval systems [8]. In this approach, the user has to specify the weights of each feature. This process, however, is very tedious and difficult for novice users. Our approach is to visualize the relationship among images using different criteria at the same time.

In this paper, we propose a new visualization system for image retrieval named *3D MARS*. In *3D MARS*, the user browses and queries images in an immersive 3D Virtual Reality space of CAVE. In general, 3D visualization has two benefits. First, more images can be displayed at the same time without occluding one another. Second, by giving the meanings to each axis, the user can examine the query result with respect to three different criteria such as color, texture and structure.

In addition to this, our system dynamically reorganizes the visualization scheme based on the user's feedback. Therefore, the users can browse images effectively. Furthermore, by giving feedback to the previous query results, the user can incrementally refine the query [4][5][6].

The outline of this paper is as follows. In the next section, we describe related work for 3D visualization of text and image retrieval. In Section 3, we present the visualization system of *3D MARS*. Then, the system architecture is described in Section 4. Finally, we provide the discussion and future work as well as conclusion.

2. RELATED WORK

While many researchers have proposed information visualization with 3D virtual reality environment, most of them were applied to text retrieval systems [1][2][3]. However, there are significant difference between text retrieval and image retrieval with regard to visualization.

First, in most text retrieval systems, only the title and minimal information can be displayed at once. Otherwise, the display would be cluttered with lots of texts. It is difficult, however, for user to judge the relevance only from the title of a document. In order to see more information such as the abstract or the contents of the documents, the user has to open up another display window. This degrades the usability of the VR system, especially for immersive VR such as HMD and CAVE. On the other hand, in image retrieval, all the user need for relevance judgement is image itself. This user judgement is instant and does not require additional display window. Hence, the system need to show only images themselves and titles.

Second, the index of the text retrieval is made of keywords in the documents. Thus, the most effective method for text retrieval is to allow the user to type in keywords. Meanwhile, the image retrieval systems index the images into numerical low-level features such as color and texture. These low-level features are not understandable to the user. Accordingly, the most common way of image query is "Query by Examples." In order to express the user's semantic concept with these low-level features, the weights of these feature components have to be adjusted automatically. *Relevance Feedback* technique for image retrieval was introduced by Rui et al. [4] for this purpose.

Finally, in both text and image retrieval systems, documents are indexed in a high dimensional space. Thus, in order to display documents in a 3D space, the dimensionality has to be reduced. Because the index of text retrieval is made of the occurrence and frequency of keywords, it is difficult to automatically group these components in meaningful manner. Such a organization is usually domain specific and requires human operation. On the other hand,

the feature vector of image retrieval systems can be grouped easily, for example, into color, texture and structure.

Several researchers have investigated 3D visualization for image retrieval [9][10][16]. Virgilio [10] is a non-immersive VR environment for image retrieval. Because their system is developed on VRML, however, the visualization is static and interactive query is not possible. Only system administrators can send a query to the system and the other users can only browse the resulting visualization.

Hiroike et al. [9] also developed a monitor-based VR system for image retrieval. In their system, hundreds of images in the database are displayed in 3D space at once. According to the user feedback, these images are re-organized and form clusters in the space. There are significant differences, however, between their system and 3D MARS. First, while in our system, more than one images can be selected as one set of query, their system regards each selected image as a different query. Thus, it creates one cluster for each query example. Second, our system expresses the degree of similarity of the images by the distance from the origin. In contrast, in their system, the size of the images represents the degree of similarity. Third, the users of their system need to adjust weights of images features by themselves. On the other hand, 3D MARS automatically adjusts the weights of the features. Finally, unlike our system, their system does not take advantage of the dimensionality to provide understandable meanings to the users.

3. NAVIGATION IN 3D MARS

The user of 3D MARS immerse himself into a projection-based Virtual Reality environment (Figure 1.) By wearing shutter glasses, the user can see a stereoscopic view of the world and can sense the depth.

When the system starts, it displays a number of images aligned in front of the user. As the user moves, the images rotate to face the user. These images are randomly chosen by the system. When the user touches one of images by the wand, the image is highlighted and the filename is displayed below it. By moving the wand, the image can be move to any position. The user can select an image as relevant (i.e. a query example) by pressing a wand button. More than one images can be selected. The selected images are displayed with red frames. In order to de-select an image, press the button again. The user can also specify an image as a negative example. The negative examples are displayed with blue frames. Moreover, the users can fly-through in the space by joystick. In order to prevent the user from getting lost, a *virtual compass* is provided on the floor. Three arrows of the compass always show X-axis, Y-axis, and Z-axis respectively.

When the user presses the QUERY button, the system retrieves and displays the most similar images from the image database. The locations of the images are determined by the feature distance from the query images. The X-axis, Y-axis and Z-axis represent color, texture and structure of images respectively. The more similar an image is, the closer to the origin of the space is it located. If the user finds another relevant (or irreverent) image in the result set, he can select it as an additional relevant (or irreverent) example and press the QUERY button again. By repeatedly picking up images, the query is improved incrementally

For researchers of image retrieval systems, visualizing how query vector is formed and how images are clustered in the feature space is useful for evaluation of their algorithms. For this purpose, we have implemented *Sphere Mode* in our system (Figure 2.). In this mode, all the images are represented by spheres. The positive examples are displayed as red spheres, and the negative examples are displayed as blue spheres. By flying through the space in this mode, the researcher can examine how images are clustered from different view angles

Unlike [9], where non-stereoscopic monitor-based virtual reality is used, our system is developed on a projection-based virtual reality environment of NCSA CAVE system. The space is projected on four walls (front, left, right, and floor) surrounding the user. With shutter glasses, the user can see a stereoscopic view of the world. In this space, the user can freely walk around. Therefore, it is much easier for the user to travel in the virtual space.

Although we are currently using a projection based virtual reality system, our system is not limited to this. The system can be used in other type of 3D display such as Head-Mounted Display or Monitor-based virtual reality system.

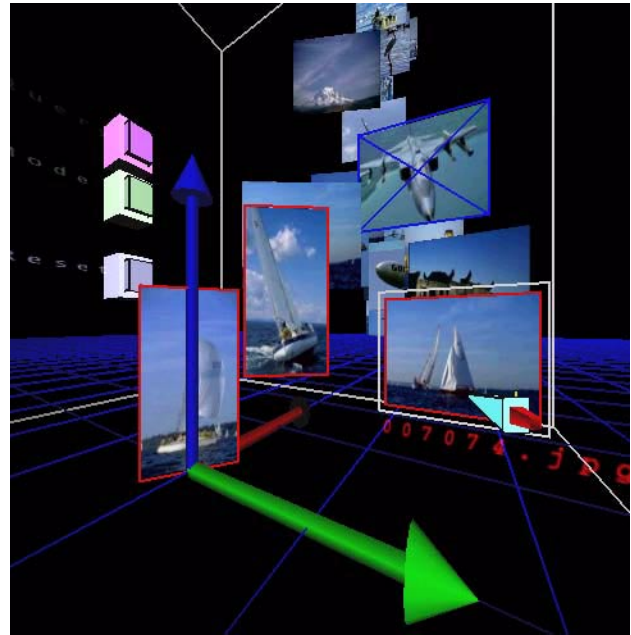


Figure 1. The interface of 3D MARS. Relevant and irrelevant images are displayed in red frames and blue frames, respectively. The three arrows in the bottom is the compass.

4. SYSTEM ARCHITECTURE

The system consists of the *Visualization Engine* (client) and the *Query Server* as shown in Figure 3. They are communicating via Hyper-Text Transfer Protocol (HTTP.)

4.1 Visualization Engine

The Visualization Engine displays a set of images on the four walls of the projection-based Virtual Reality system, called CAVE. It takes a request from the user, sends the request to the server and then receives the result from the server. When the user

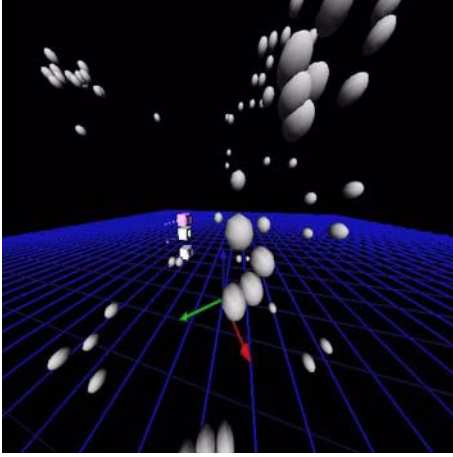


Figure 2. The *Sphere Mode*. All the images are represented by spheres. This makes easy to examine clustering of images.

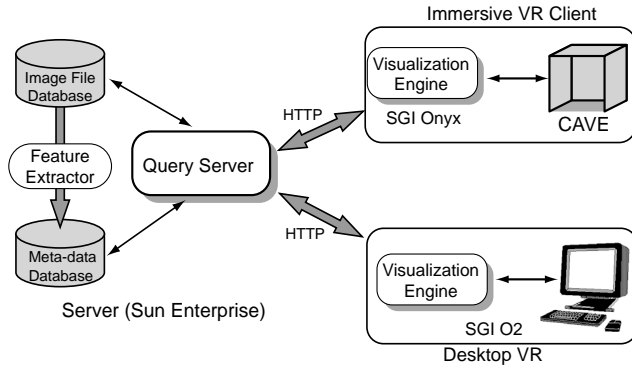


Figure 3. The System Architecture

pushes the QUERY button, it sends IDs of the selected images to the server. The requests are sent as a “GET” command of HTTP. When the reply is returned, the client receives a list of IDs and the location of the k most similar images as well as their feature vectors. Next, it downloads all the relevant image files from the image database on the server. Finally, these images are displayed on the virtual space. The location of each image is determined by its low-level features.

This component is written in C++ with OpenGL and CAVE library. The system is running on a twelve-processor Silicon Graphics Onyx 2. Each wall of the CAVE is drawn by a dedicated processor. Loaded image data are shared on a share memory and accessed from these processors.

4.2 Query Server

The Query Server maintains image files and their meta-data. When the server receives a request from a client, it computes the weights of features and compares user-selected images with images in the database. Then, the server sends back IDs of the k most similar images and their locations in 3D [4][5][6][7].

The server is implemented as a Java Servlet on Apache Web Server. It is written in C++ and Java. The server can simultaneously communicate with other types of client such as Java Cli-

ent [15]. Currently, the server is running on a Sun Enterprise Server.

Image Features

On the server, a number of image files are stored in the image file database. In addition, the meta-data of these images are stored. These meta-data are indexed in advance. We use three features: color, texture, and edge structure. For color, the first two moments (mean and standard deviation) from each of HSV channels are extracted. Thus, the total number of color features is $3 \times 2 = 6$. For texture, the images are decomposed into 10 de-correlated sub-bands, then the standard deviation of the wavelet coefficients are extracted from each sub-bands. For structure, we used *Water-Fill* edge detector [7], which extracts eighteen elements of the edge features. Therefore, total thirty-four (34) numbers are stored for each image. Currently, 8000 images and their meta-data are stored on the database.

Feature Ranking and Total Ranking

When the server receives a request from the client, it computes the distance between the user-selected images and images in the database. Then, the server sends back ID of the k most similar images and their feature vectors. In order to take advantage of three dimensional visualization, we use two ranking strategy: *Feature Ranking* and *Total Ranking*.

The Feature Ranking is a ranking with respect to only one of the features. First, for each feature i ($i = \{\text{color, texture, structure}\}$), the system computes a query vector q_i based on the positive and negative examples specified by the user. Then, it ranks images in the database according to the distance g_{ni} of image n from the query vector. For the computation of the distance, we used Biased Discriminant Analysis (BDA.) The detail of BDA is described in [6].

After the Feature Ranking is computed, the system combines each feature distance g_{ni} into the total distance d_n . The total distance of image n is a weighted sum of each g_{ni} ,

$$d_n = \vec{u}^T \vec{g}_n \quad (1)$$

where $\vec{g}_n = [g_{n1}, \dots, g_{nI}]$. I is the total number of features. In our case, I is 3. The optimal solution of $\vec{u} = [u_1, \dots, u_I]$ is solved by Rui et al.[5] as follows,

$$u_i = \frac{1}{\sum_{j=1}^I \sqrt{\frac{f_j}{f_i}}} \quad (2)$$

where $f_i = \sum_{n=1}^N g_{ni}$, and N is the number of positive examples. This gives higher weight to that feature whose total distance is small. Which means that if the positive examples are similar with respect to a feature, this feature gets higher weight.

Finally, the Total Ranking is computed based on the total distances.

By the use of both Feature Ranking and Total Ranking, the system can return images even if only one of their feature is close to the query. These images could be ignored in the traditional CBIR systems.

5. DISCUSSION AND FUTURE WORK

There are several important issues that were not addressed in our prototype. First, in the current system, the meaning of each axis

does not change; X-axis always means the distance in the color feature, Y-axis always means the texture, and so on. Nevertheless, often some components are not meaningful for a user. Thus, an alternative approach is to dynamically change the meanings of axes by extracting important features from the data set. Several algorithms have been proposed for extracting features from a high-dimensional data [11][12]. *Multidimensional scaling* (MDS) and its variations have been used for a variety of pattern recognition problems [12] Tian and Taylor [16] applied MDS to visualize 80 color texture images in 3D. However, because MDS is computationally expensive, it is not suitable for visualization of a large number of images.

Meanwhile, Faloutsos et al.[11] proposed a faster algorithm called *FastMap*. While MDS requires $O(N^2)$ time, FastMap requires only $O(N)$. The problem of these approach is that it may be confusing for the user because the meaning of each axis is always changing. We need compare this dynamic axes approach with the static axes. Moreover, for evaluation of retrieval algorithms, it may be useful to allow the user to choose the meaning of each axis on the fly. Providing 1D or 2D view of the feature space would be helpful as well.

Second, in our prototype, the system computes one query vector at a time. As a result, the user has to select only one set of similar examples. Querying two different kinds of images at the same time is not allowed. Therefore, the system shows only one cluster for each query. For some users, however, querying more than one image classes at the same time might be desired. The important question is how to display the relationship between two different image classes in our display space. To this end, modification of the classification algorithm may be required.

Santini et al. [13] proposed an interesting topological user interface. In their system, the user specifies the relevance of images by moving the images in a 2D display space. If the user believes two images are similar, he moves these images close to each other. If the user believes an image is not relevant, he moves it to a distant location from the relevant images. Then, the system compute the feature weighting so that the resulting total distance reflects the user intention. In the system, however, the orientation in the display space is not considered. We are going to investigate how 3D information (both distance and orientation) can be used for query specification by the user.

Finally, we plan to evaluate the usability on human subjects.

6. CONCLUSION

In this paper, we proposed a new visualization system for Content-Based Image Retrieval, named 3D MARS. In 3D MARS, the user browse and query images in an immersive virtual environment. The result of query is displayed in 3D, so that the user can see the result with respect to three different criteria. The sphere mode help the researcher of image retrieval system to analyze their algorithms. The detail of our prototype was described on this paper.

Although we have implemented the system on a projection based virtual reality system, our approach is applicable to the other type of 3D display such as Head-Mounted Display or Monitor-based virtual reality system.

7. ACKNOWLEDGEMENT

This work was supported in part by National Science Foundation Grant CDA 96-24396.

8. REFERENCES

- [1] Card, S. K., Macinlay, J. D. and Shneiderman, B., "Readings in Information Visualization - Using Vision to Think," Morgan Kaufmann, 1999.
- [2] Wise, J.A. et al., "Visualizing the non-visual: Spatial analysis and interaction with information from text documents," In *Proc. of the Information Visualization Symposium 95*, pages 51-58. IEEE Computer Society Press, 1995.
- [3] Hearst, M. A. and Karadi, C., "Cat-a-cone: An interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proceeding of 20th Annual International ACM SIGIR Conference*, Philadelphia, PA, 1997.
- [4] Rui, Y., Huang, T. S., Ortega, M. and Mehrotra, M., "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," In *IEEE Transaction on Circuits and Video Technology*, Vol. 8, No. 5, Sept. 1998
- [5] Rui, Y. and Huang, T. S., "Optimizing Learning in Image Retrieval," In *Proc. of IEEE CVPR*, 2000.
- [6] Zhou, X. and Huang, T. S., "A Generalized Relevance Feedback Scheme for Image Retrieval," In *Proc. of SPIE Vol. 4210: Internet Multimedia Management Systems*, 6-7 November 2000, Boston, MA, USA.
- [7] Zhou, X. S. and Huang, T. S., "Edge-based structural feature for content-base image retrieval," *Pattern Recognition Letters, Special issue on Image and Video Indexing*, 2000.
- [8] Flickner, M. et al., "Query by image and video content: The QBIC system," *IEEE Computers*, 1995.
- [9] Hiroike, A. and Musha, Y., "Visualization for Similarity-Based Image Retrieval Systems," *IEEE Symposium on Visual Languages*, 1999.
- [10] Massari, et al., "Virgilio: a Non-Immersive VR System to Browse Multimedia Databases," In *Proc. of IEEE ICMCS 97*, 1997.
- [11] Faloutsos, C. and Ling, K., "FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets," In *Proc. of ACM SIGMOD95*, pages 163-174, May 1995.
- [12] Kruskal J.B., and Wish, M., "Multidimensional Scaling," SAGE publications, Beverly Hills, 1978.
- [13] Santini, S. and Jain, R., "Integrated Browsing and Querying for Image Database," *IEEE Multimedia*, Vol. 7, No. 3, 2000, page 26-39.
- [14] Brown, M.H. and Hershberger, J., "Color and Sound in Algorithm Animation," *IEEE Computer*, Vol. 25, No. 12, December 1992.
- [15] Nakazato, M. et al., UIUC Image Retrieval System for JAVA, available at <http://chopin.ifp.uiuc.edu:8080>.
- [16] Tian, G.Y. and Taylor, D., "Colour Image Retrieval Using Virtual Reality," In *Proc. of IEEE International Conference on Information Visualization (IV'00)*, 2000.