Image Clustering for the Exploration of Video Sequences

Vicenç Torra[1], Sergi Lanau[1], Sadaaki Miyamoto[2]
[1] Institut d'Investigació en Intel·ligència Artificial (IIIA-CSIC)
E-08193 Bellaterra, Catalonia
e-mail: vtorra@iiia.csic.es
[2] Institute of Engineering Mechanics and Systems
University of Tsukuba, Ibaraki 305-8573, Japan
e-mail: miyamoto@esys.tsukuba.ac.jp

April 11, 2005

Abstract

In this paper we present a system for the exploration of video sequences. The system, GAMBAL-EVS, segments video sequences extracting an image for each shot and then clusters such images and presents them in a visualization system. The system permits to find similarities between images and to traverse along the video sequences to find the rellevant ones.

Keywords: Information retrieval, image retrieval, clustering of video sequences, video segmentation.

1 Introduction

The amount of information currently available in internet and in proprietary databases is increasing every day. While in the past most of the stored information was textual, in present days there is more and more information that have a multimedia basis. In this work we consider a particular type of such multimedia data: sequences of video images.

Databases of video sequences are currently huge due to ubiquitous video cameras and invasive television. Nevertheless, access to individual images or selection of rellevant (or interesting) shots is still an arduous task. In fact, users require to put a lot of effort into analysing the images for obtaining good results due to the sequential nature of the video sequences and due to the current limitation of computational systems.

Current research in the multimedia field is oriented towards the application of existing data mining methods and on the development of new tools for exploration and retrieval (*e.g.* (Crestani & Pasi, 2000),(Amores & Radeva, 2005)).

This is, effort is given to systems that extract some kind of knowledge from multimedia data and to systems that help on the navigation among images. In this work we describe a system for the exploration of video sequences.

At present, several systems have been developed for exploration of image databases. See *e.g.* QBIC (QBIC, 2004) and VIPER (Müller et al., 2000) systems. Research is described in *e.g.* (Zhang & Zhong, 1995), (Sethi & Coman, 1999) and (Chen, Bouman & Dalton, 2000). (Zhang & Zhong, 1995) and (Sethi & Coman, 1999) base their approach on Hierarchical Self Organizing Maps (see (Merkl & Rauber, 2000) for a review of such maps and Kohonen (1997) for the original non-hierarchical ones). In such systems, images are organized according to a two-dimensional grid structure where each cell (neuron) in the grid contains a subset of the images. Cells located in near positions in the grid contain similar images. Traversing the hierarchical structure of HSOM users can explore the database, focusing on those images that have a larger interest to them. While in such works the structure for exploring the database is fixed (kept constant since its construction), this does not happen in the system proposed in (Chen, Bouman & Dalton 1998). In (Chen, Bouman & Dalton, 1998) authors introduced active browsing. The main idea is that the users can modify the database organization when traversing it. Nevertheless, the organization is still hierarchical and thus similar to hierarchical SOM. Nevertheless, in that work a large importance is given to computational efficiency (see (Chen, Bouman & Dalton, 2000) for details). Recently, Stan and Sethi (2003) introduced a system for image exploration based on k-means. Again, a hierarchical structure is built from images so that the user can traverse to find the desired ones.

An alternative approach has been proposed by Rodden (2002). Instead of giving listings of images at a particular level of the hierarchical structure, images are located in a space according to their visual similarity. This is, images that have a larger similarity are found in nearer positions than images that have a lower similarity. Such an arrangement permits a better understanding of the image database.

In this paper we introduce GAMBAL-EVS, a system for video sequence exploration. This system permits the user to analyze the scenes and images in a video sequence. The system, that is based on the GAMBAL system (Lanau, 2003) (a system for exploring textual information in the web) constructs a hierarchical structure based on a variation of the c-means algorithm. The system represents the images in the surface of a sphere in such a way that similar images are located in nearer positions. In this way, it is easy to apprehend the structure of the images in the video sequence. Accordingly, the approach presented here combines the advantages of systems similar to (Rodden, 2002) with the ones that build hierarchical structures.

The embedding of our exploration system in the GAMBAL environment, that includes a web crawler and is oriented to information access on the web, permits to augment the files the system can process with non-textual ones. In particular, video sequences (and images) can now be downloaded and then browsed by the user using the video extension described in this work.

Although this system is to be used for information access on the web, the

system needs, as most search engines do, that the web crawler first downloads the files (the video sequences). This is previous to user's navigation because the visualization system requires similarities between images to be known, otherwise they cannot be properly displayed. Accordingly, the system does not provide a dynamic access to the web but a static one.

The structure of the paper is as follows. In Section 2 we give an overview of the exploration system and focus on the process of video segmentation. In Section 3 we describe in detail the clustering process. Section 4 gives some examples. The paper finishes in Section 5 with the conclusions.

2 GAMBAL-EVS

As said in the introduction, GAMBAL-EVS is a system for video sequence exploration. Such system has been built on the top of GAMBAL (see (Lanau, 2003) for details), a system for clustering and visualization of textual documents based on clustering techniques.

GAMBAL-EVS stands for *GAMBAL for the Exploration of Video Sequences*.

The architecture of GAMBAL-EVS is given in Figure 1. On the one hand we have a video segmentation module that decomposes a video sequence into a set of shots. For each shot a representative image and a representative histogram is given. Shot detection is done on the basis of image histograms (*i.e.*, comparing histograms). This is detailed below in Section 2.1.

Once the sequence is decomposed into a set of shots, an extended version of the GAMBAL system is applied (denoted by GAMBAL* in the Figure). The extension was carried out so that images can be dealt and represented. In fact, GAMBAL clusters the images so that similar images are put together (in a hierarchical structure). At this point, image histograms are used to compute similarities between images. Then, GAMBAL* uses such hierarchical structure, the image histograms and the images themselves for presenting the results to the user. In this way, the images that define the video sequence can be explored by the user.

In this section we give the details on the video segmentation process (the Video Segmentation module in Figure 1). Then, in Section 3 our clustering approach is described in more detail.

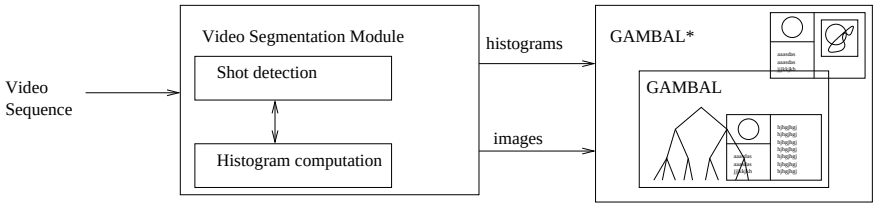

Figure 1: The GAMBAL-EVS system for Exploration of Video Sequences

2.1 Video Segmentation Module

Video segmentation and vector representation are based on color histograms. More precisely, the system builds an histogram for each image and uses such histograms as their numerical representatives. Segmentation is based on differences on the histograms.

Images are considered in terms of their RGB color representation. Then, to define the histogram of an image, each RGB color is reduced from 256 to 8 different possible values. Then, for a given image im, each pixel $p \in im$ is *counted* in the following position:

$$index(p) = indexR(p) * 64 + indexG(p) * 8 + indexB(p)$$

where

$$indexR(p) = floor(R(RGB(p))/32)$$

$$indexG(p) = floor(G(RGB(p))/32)$$

$$indexB(p) = floor(B(RGB(p))/32)$$

Therefore, the histogram of image im corresponds to:

$$h^{im}(i) = |\{p \in im | index(p) = i\}|$$

The histograms of two consecutive images (h, h') are compared using the Kolmogorov-Smirnov test. This is:

$$ks = \max_i |CDF_h(i) - CDF_{h'}(i)|$$

where CDF_h is the cumulative distribution function of h.

This test was proposed for video segmentation by Sethi and Patel in (Sethi & Patel, 1995) and compared positively in (Ford, Robson, Temple & Gerlach, 2000) against other approaches based on histograms as *e.g.* the Chi-square.

As histogram metrics produce the best results when computed for blocks rather than globally (see *e.g.* (Ford, Robson, Temple & Gerlach, 2000)), our system uses block-based histograms. The implementation is parametric, but to avoid a large computational complexity we use images of 16 blocks. Accordingly, the representative of an image im, h^{im}, is a set of histograms $\{h_j^{im}\}_j$ for $j < 16$. Note that as 8 different possible values are considered for each RGB color, this corresponds to block-histograms with a dimension equal to 512. Therefore, each image is represented by (vectors of) $512 \times 16 = 8192$ values.

When rellevant differences are found between the histograms of two consecutive images, they are considered to belong to two different shots. To determine when such difference is rellevant, a threshold θ, determined in an heuristic way, has been used. This process corresponds, in fact, to the detection of a shot cut.

Therefore, a shot is detected between images im and im' when:

$$\max_i \max_{j \leq 16} |CDF_{h_j^{im}}(i) - CDF_{h_j^{im'}}(i)| > \theta \quad (1)$$

Taking into account the process just described, a sequence can be decomposed into a set of shots. The next step is to compute representatives for each shot. Such representatives take again the form of histograms. In fact, each representative is defined as the set of histograms (*i.e.*, one histogram for each of the 16 blocks) of the last image of the shot. Histograms are normalized per block so that within a block the values add to one.

Thus, the system produces a pair *(image(s),* h^s*)* for each shot s. Here, the image that represents the shot s is the last image in s, and h^s is the corresponding histogram. Naturally, such image is the one that will be displayed when the images from the video sequence are represented within the clustering system. So, while $image(s)$ is the *graphical* representation of the shot, $h(s)$ (the histogram) is the numerical one and the one used to compute similarities/distance between shots.

Therefore, and putting all together, we have that a video sequence VS defined by shots $s \in VS$ is translated into a set of pairs $\{(image(s), h^s)\}_{s \in VS}$. As all images in a shot are supposed to be similar enough, these will be the only images represented in our exploration system and the only images considered in the clustering process.

Note that among the two failures of a shot detection algorithm, false negatives (not detecting a new shot that should be detected) are more relevant than false positives (detecting a shot that should not). This is so because for false positives several representatives are computed and displayed for the same shot. Nevertheless such representatives will be probably clustered together and finally visualized in near positions in the graphical interface of GAMBAL. Instead, false negatives imply that only one representation is extracted for several shots. This causes that some shots are not visualized and they will only be detected if the user displays the whole sequence. As a consequence of this fact, the threshold θ in Equation (1) has been selected in such a way that false negatives are minimized although this causes some false positives.

2.2 Implementation issues

GAMBAL-EVS (and GAMBAL*) is fully implemented using the Java programming language. All image processing elements and the video segmentation module has also been implemented in Java. We have used for this purpose the Java Media Framework 2.1.1e API provided by Sun. This permits the system to be run in different platforms and also to consider different video formats. Nevertheless, the system was developed and tested using Linux (Redhat and Fedora Core). The files considered for this paper followed the MPEG standard.

3 Clustering and visualization

Clustering and visualization in GAMBAL-EVS relies on the GAMBAL system. Roughly speaking, the GAMBAL system builds a dendrogram (a hierarchical

structure) of the data using a clustering method and, then, such dendrogram is visualized in a graphical interface.

Here, the clustering process is applied to the images obtained from the video sequence and, thus, the dendrogram is defined with images in its leaves and clusters of images in its internal nodes. Clusters are defined putting together similar images (according to their histograms).

It has to be said that the clustering process is independent from the shot detection algorithm in what concerns to the comparison of images. As detailed in Section 3.2.1, the Euclidean distance is used in the clustering prcoess while the Kolmogorov-Smirnov is used for shot detection. Nevertheless, other methods for computing the similarities between images could be considered. From our point of view this independence is important and meaningful because one aspect is shot detection (that is application independent) and the other is the similarity between images (that is application dependent).

3.1 The graphical interface

The graphical representation of the dendrogram is done using the GAMBAL visualization system. Figure 2 gives a snapshot of the system. In such system, objects and clusters are located on the surface of a sphere. Moreover, different α-cuts of the same dendrogram (corresponding to different partitions of the elements) are represented in different concentric spheres. At user's request, the system moves from one sphere to an adjacent one, so that the user can navigate throw the hierarchy. In other words, the user can change the degree of granularity in which the objects in the hierarchy are seen. Moreover, the user can zoom or rotate the current sphere at his desire.

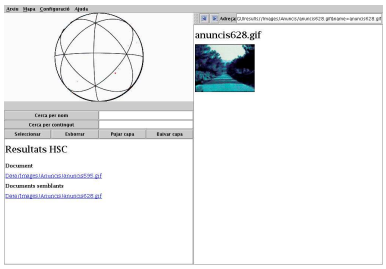

Figure 2: Snapshot of the GAMBAL visualization system

Figure 3 is to illustrate the GAMBAL system. This figure represents (on the left hand side) a dendrogram constructed by objects $\{a, b, c, d, \ldots, m\}$ and (on the right hand side) their representation on a set of concentric spheres $C0$, $C1$, $C2$, $C3$ (*circles* in this case). Each sphere (circle) represents one of the α cut in the dendrogram (in the left hand side of the figure). *E.g.* $C2$ corresponds to the α-cut defined by $\{A, B, C, D\}$ and $C1$ to the one defined by $\{E, F\}$. Although in the figure all the spheres are displayed at once, the interface only displays one

at a time. With this interface, the user can navigate through the dendrogram changing from one sphere to another adjacent one and this corresponds to change from one α-cut to another one.

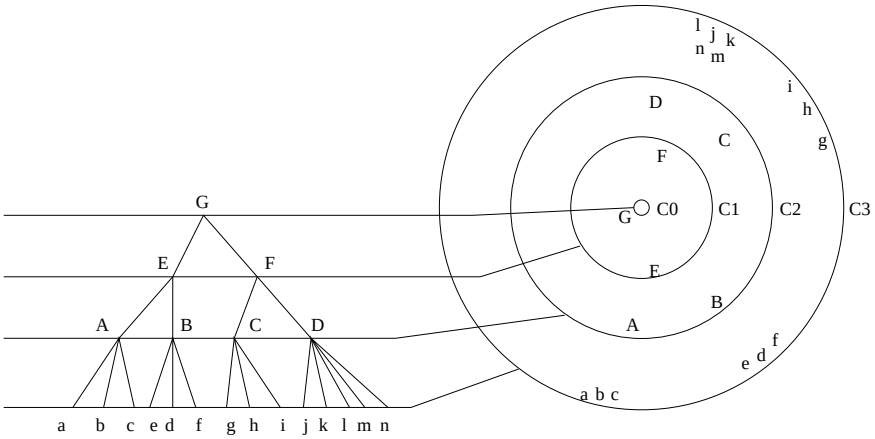

Figure 3: A conceptualization of the GAMBAL clustering and its visualization system.

At the surface of the sphere, only dots are represented. Clicking a particular dot, a window of the system displays the information (*e.g.* the file) corresponding to the clicked point as well as of the nearest objects. In the extended version of GAMBAL, the image can be displayed as well. See Figure 2.

Although we have considered here a bottom-up description of the process (from low level clusters to higher level ones), the method implemented is top-down. This is, at first, a large cluster with all images is considered, and in successive steps the sets of images are further splitted. In this way, each splitting corresponds to a further refinement of the considered cluster. The whole process is described in the next section.

3.2 The Clustering Process

As described in the previous section, the clustering process follows a top-down design. In successive steps, clusters are splitted into new clusters. To bootstrap the process, we have considered an initial partition of the images. Each partition element defines a cluster. As such initial partition is defined taking into account the localization of the objects on the surface of the first sphere (C1 in Figure 3) such process is also described.

Note that following Figure 3 a $C0$ sphere containing a single cluster with all objects can also be defined. As such sphere is not rellevant for exploring sets of documents / images we have not implemented its visualization and in the clustering process we start directly building $C1$.

3.2.1 Localization of objects on the surface (definition of $C1$)

Objects are located in the surface of the $C1$ sphere according to their similarities. Roughly speaking, objects are located in a way that similar objects are located in near positions, while dissimilar objects are located in farther positions.

In fact, this approach corresponds to a multidimensional scaling. To implement this method we have adapted the Sammon's map (a method for multidimensional scaling) so that the distance computed between pairs of objects is compared with the distance of their localization on the sphere.

To give additional details, we need some formal definitions. Let x_i denote the images, and z_i their localization on the surface of the sphere, then with $d^o(x_i, x_j)$ we denote the distance between the images x_i and x_j according to their histograms and with $d^s(z_i, z_j)$ we denote the distance according to their position on the surface (*i.e.*, the angle that define z_i and z_j). Given such definitions, to locate all data x_i on the surface is equivalent to find their position z_i so that the following expression is minimized:

$$\sum_{i>j} \frac{\left(d^o(x_i, x_j)/d^o_{max} - d^s(z_i, z_j)/d^s_{max}\right)^2}{d^o(x_i, x_j)/d^o_{max}}$$

In our application we have defined the distance between two images im and im' in terms of their (set of) histograms h and h' using the Euclidean distance as follows:

$$d^o(h, h') = \sqrt{\sum_i (h(i) - h'(i))^2}$$

At this point, other distances $d^o(h, h')$ might be considered as well. Also, it might be possible to define distance taking into account the whole image but not only the histograms. As argued in Section 3, computing similarities between images in the exploration stage is application dependent, as it depends on the kind of *similarities* that the user is interested in detecting.

3.2.2 Initial clusters:

The surface of the sphere is divided into six triangular regions, all of them having the same size. Images are assigned to the corresponding cluster. Regions were defined as triangular because such shape permits an homogeneous recovering of all the sphere surface (*i.e.* a triangularization of the surface).

3.2.3 Cluster splitting and new cluster formation:

The process of splitting a cluster corresponds in our system to split a triangular region into three new (but smaller) triangular regions.

This process is achieved using a variation of the c-means (Duda & Hart, 1973), (Miyamoto & Umayahara, 2000) clustering algorithm. In fact, the varia-

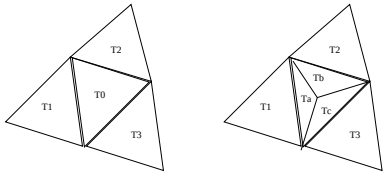

Figure 4: Splitting a triangle into 3 new triangles

tion was defined so that the new partition is consistent with the triangularization of the surface.

Accordingly, we consider an objective function based on the one of the *c*-means but that takes into account some additional elements related to the triangular shape. In particular, the objective function considers that only three new clusters are built and that the centroids of these clusters should be *near* the neighbouring triangles. So, if a triangular region $T0$ (see Figure 4) is splitted into regions Ta, Tb and Tc, the centroids of Ta, Tb and Tc (namely v_1, v_2 and v_3) should be near to the centroids of the limiting triangles $T1$, $T2$ and $T3$ (namely a_1, a_2 and a_3).

This is expressed by means of the following objective function:

$$J(U,V) = \alpha \sum_{k=1}^{n} \sum_{i=1}^{3} u_{ik} d(x_k, v_i)^2 + (1-\alpha) \sum_{i=1}^{3} d(v_i, a_i)^2$$

where x_k are the elements to cluster, v_i and a_i are as defined above and c_{ik} is a boolean value representing whether the x_k belongs to the cluster with centroid v_i. As it is assumed that a certain element can only belong to a single class, the matrix (c_{ik}) should belong to the set:

$$M = \{(u_{ik}) | u_{ik} \in \{0,1\}, \sum_{i=1}^{3} u_{ik} = 1 \text{ for all } k\}$$

Here, α is assumed to be constant, and it corresponds to a selected trade-off between the usual *c*-means not considering the neighbors (i.e., the result of minimizing the expression $\sum_{k=1}^{n} \sum_{i=1}^{3} u_{ik} d(x_k, v_i)^2$) and just putting the centers at the neighbors position (i.e., the result of minimizing $\sum_{i=1}^{3} d(v_i, a_i)^2$).

As at this point we are considering the splitting of the regions on the surface, the distance d in $J(U,V)$ corresponds to the distance on the surface.

The minimization problem stated above is solved using the general algorithm for c-means:

Step 1: Start with an initial set $\bar{V}$

Step 2: Solve $min_{U \in M} J(U, \bar{V})$
and let the optimal solution be $\bar{U}$

Step 3: Solve $min_V J(\bar{U}, V)$
and let the optimal solution be $\bar{V}$

Step 4: Repeat steps 1-3 while (U, V) is not convergent.

These steps are computed in the following way:

Step 1: $\bar{V} = (v_1, v_2, v_3)$ are defined as the average value between the center of the triangle we are splitting and the center of the corresponding neighbouring triangle.

Step 2: Elements are assigned to the nearest v_i.

Step 3: To find the $\bar{V} = (\bar{v_1}, \bar{v_2}, \bar{v_3})$ that minimizes:

$$J(v) = \alpha \sum_{k=1}^{n} \sum_{i=1}^{3} u_{ik} \psi^2_{x_k, v_i} + (1 - \alpha) \sum_{i=1}^{3} \psi^2_{a_i, v_i}$$

the gradient method is used.

Accordingly, an iterative process is applied where at the k-th step the values for v^k are computed using the ones at the $k-1$-th step (*i.e.* v^{k-1}) and the gradient ∇J using the following expression:

$$v^k = v^{k-1} - \gamma \nabla J$$

where γ is a learning rate constant.

Step 4: To detect convergence, the distance between centers at time k and $k-1$ is checked. When such distance is less than a certain threshold, the algorithm is stopped.

4 Examples

The system has been applied to several video sequences. The examples reported here correspond to two advertisements and one fragment from a TV entertainment program. In Figure 5 we display two images of the advertisement sequences considered (color images in the original video sequence). The image on the left corresponds to a sequence that lead to 13 different images corresponding to 13 different shots. This set is used for illustration.

Figure 6 gives a snapshot of the system when a particular image is selected. It can be observed (upper part in the left hand side of the image) the sphere with its triangularization and the dots corresponding to the images represented. In the left hand side of the image (lower part) a list of links can be observed. Such list are obtained clicking on the surface. In this case, the clicked object as well as nearest objects are listed. On the right hand side of the figure, the selected image is given.

In Figure 7 the two similar images of Figure 6 are displayed. It can be observed, that the similarities do not correspond to objects located in the same

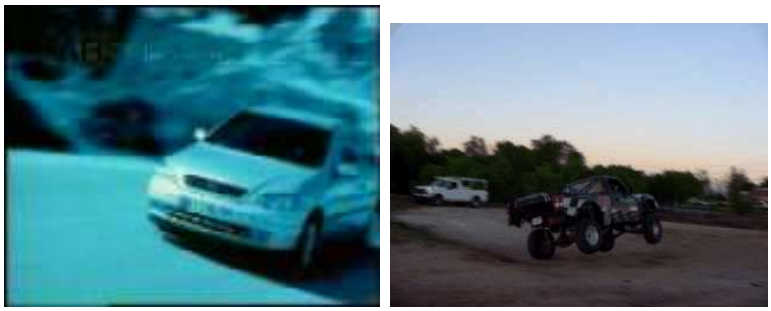

Figure 5: Two images from the video sequences considered.

position (*i.e.* a car in the center of the image) but on color and texture. Note that a significant percentage of both figures contain the same *rocky mountain*. This color and texture based similarity is due to the system approach on computing similarity based on histograms. Moreover, the segmentation process will assign to all those images from the same shot (and having objects in the same position) a single image representation. Therefore, such similar images, unless they come from different shots (as in TV news), are not duplicated.

In Figure 8, some images are displayed for the example of the entertainment program. This sequence is 4 minutes 11.12 seconds long and its segmentation took (according to linux *time* command) 4 minutes 13 seconds of real time, 3 minutes and 2.9 seconds of which were devoted to the user and the rest to the system. In fact, the segmentation takes place while the video is displayed (with sound) *in real time* on the screen. With a $\theta = 0.7$, 102 shots have been generated. Naturally, variations of θ leads to variations on the number of shots being generated.

In Figure 8 (a) and (b), we show two images that have been located in near positions on the sphere (near the center of the sphere). They correspond to two contiguous shots in the original video. Then, in Figure 8 (c), we display another image that was considered similar but that is not contiguous to the previous ones. In Figure 8 (d) we have another image that has a larger similarity with the previous ones. Figure 8 (e) and (f) we include two images that were located on the other side of the sphere and, thus, they have a larger dissimilarity with Figure 8 (a) and (b). It can be seen that the last figures are again similar to each other.

5 Conclusions and Future Work

In this work we have presented a tool for image clustering and visualization to help in the exploration of video sequences. We have described the clustering and visualization system and given an example that proves its interest.

As future work we plan to extend the clustering system with fuzzy clustering

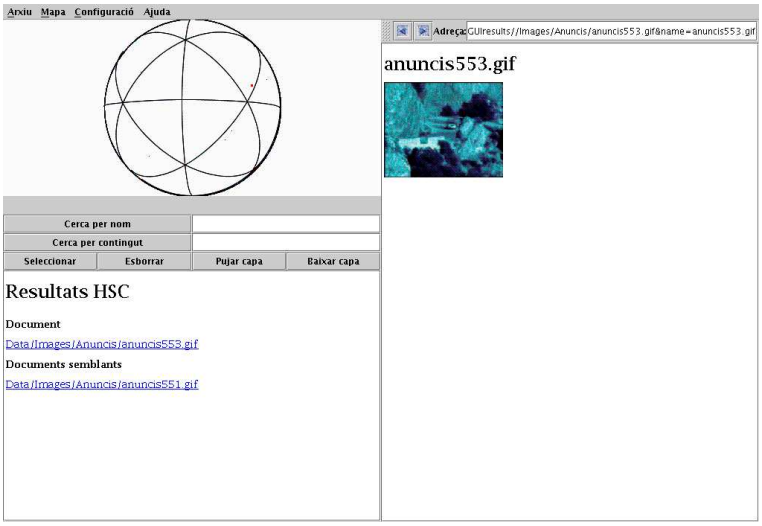

Figure 6: Example of the clustering system

techniques (following *e.g.* (Kraft, Bordogna & Pasi, 1999), (Torra, Miyamoto & Lanau, 2005), (Miyamoto, 2003)) and then to extend the approach to explore and visualize video sequences. This corresponds to find similarities between the sequences. See *e.g.* (Adjeroh, Lee & King, 1999) for an example of a similarity function on sequences.

Additionally, we plan to combine GAMBAL-EVS with filtering systems (Herrera-Viedma & Peis, 2003) so that GAMBAL-EVS permits the exploration of the results of a query or multiple query in large databases.

Acknowledgements

This work was partly funded by the Generalitat de Catalunya (AGAUR, 2004XT 00004) and the MICYT/MEC (projects TIC2001-0633-C03-02 and SEG2004-04352-C04-02).

references

Adjeroh, D. A., Lee, M. C. & King, I., (1999). A distance measure for video sequences, Computer Vision and Image Understanding, 75 25-45.

Amores, J. & Radeva, P., (2005). Retrieval of IVUS images using contextual information and elastic matching, Int. J. of Intel. Systems 20 541-559.

Chen, J.-Y., Bouman, C. A. & Dalton, J. C., (2000). Hierarchical Browsing and

Figure 7: Images located in near positions by the GAMBAL-EVS system

Search of Large Image Databases, IEEE Transactions on Image Processing 9:3 442-455.

Chen, J.-Y., Bouman, C. A. & Dalton, J. C., (1998). Similar Pyramids for Browsing and Organization of Large Image Database, Proc. of SPIE/IS&T Conf. on *Human Vision and Electronic Imaging III*, Vol. 3299, pp. 563-575, San Jose, CA.

Crestani, F. & Pasi, G., (Eds.), (2000). Soft Computing in Information Retrieval, Physica Verlag and Co, ISBN:3790812994, pp. 102 - 121, Germany.

Duda, R. & Hart, P., (1973). Pattern classification and scene analysis, Wiley, New York.

Ford, R. M., Robson, C., Temple, D. & Gerlach, M., (2000). Metrics for shot boundary detection in digital video sequences, Multimedia Systems, 8 37-46.

Herrera-Viedma, E. & Peis, E., (2003). Evaluating the informative quality of documents in SGML format from judgements by means of fuzzy linguistic techniques based on computing with words, Information Processing and Management, 39 233-249.

Kohonen, T., (1997). Self-Organizing maps, 2nd edition, Springer-Verlag, Germany.

Kraft, D. H., Bordogna, G. & Pasi, G., (1999). Fuzzy set techniques in information retrieval in Bezdek, J.C., Didier, D., Prade, H., (eds.), Fuzzy Sets in Approximate Reasoning and Information Systems, vol. 3., The Handbook of Fuzzy Sets Series, Norwell, MA: Kluwer Academic Publishers.

Lanau, S., (2003). Clasificación y visualización de datos complejos, Ms. Thesis, Universitat Autònoma de Barcelona, Catalonia, Spain.

Merkl, D. & Rauber, A., (2000). Document Classification with Unsupervised Neural Networks, in F. Crestani, G. Pasi (Eds.), Soft Computing in Information Retrieval, Physica Verlag and Co, ISBN:3790812994, pp. 102 - 121, Germany.

Miyamoto, S., (2003). Information clustering based on fuzzy multisets, Information Processing and Management, 39 195-213.

Miyamoto, S. & Umayahara, K., (2000). Methods in Hard and Fuzzy Clustering, pp 85–129 in Z.-Q. Liu, S. Miyamoto (Eds.), Soft Computing and Human-Centered Machines, Springer-Tokyo.

Müller, W., Müller, H., Marchand-Maillet, S., Pun, R., Squire, D. M., Pecenovic, Z., Giess, C. & de Vries, A. P., (2000). MRML: An Extensible Communication Protocol for Interoperability and Benchmarking of Multimedia Information Retrieval Systems. In SPIE Photonics East - Voice, Video, and Data Communications, Boston, MA, USA, November 5–8, 2000.

Nagasaka, A. & Tanaka, Y., (1992). Automatic Video Indexing and Full Video Search for Object Appearances, Proc. of IFIP TC2/WG2.6 Second Working Conference on Visual Database Systems (Visual Database System II), 113-127.

QBIC(TM) (2004). IBM's Query By Image Content. See http://wwwqbic.almaden.ibm.com/ and its application at http://www.hermitagemuseum.org/html_En/07/hm7_41_1.html

Rodden, K., (2002). Evaluating similarity-based visualizations as interfaces for image browsing, University of Cambridge, Computer Laboratory, Technical report 543, UCAM-CL-TR-543, ISSN 1476-2986.

Sethi, I. K. & Coman, I., (1999). Image retrieval using hierarchical self-organizing feature maps, Pattern Recognition Letters, 20 1337-1345.

Sethi, I. K. & Patel, N., (1995). A Statistical Approach to Scene Change Detection, Proc. SPIE (Storage and Retrieval for Image and Video Databases III) 2420: 329-338.

Stan, D. & Sethi, I.K., (2003). eID: a system for exploration of image databases, Information Processing and Management 39 335-361.

Torra, V. & Miyamoto, S., (2002). Hierarchical Spherical Clustering, Intl. J. of Unc., Fuzz. and Knowledge-Based Systems, 10:2 157-172.

Torra, V., Miyamoto, S. & Lanau, S., (2005). Exploration of textual databases using a fuzzy hierarchical clustering algorithm in the GAMBAL system, Information Processing and Management, 41:3 587-598.

Zhang, H. & Zhong, D., (1995). A scheme for visual feature based image indexing, Proc. of SPIE/IS&T conference on storage and retrieval for image and video databases III, Vol. 2420, 36-46.

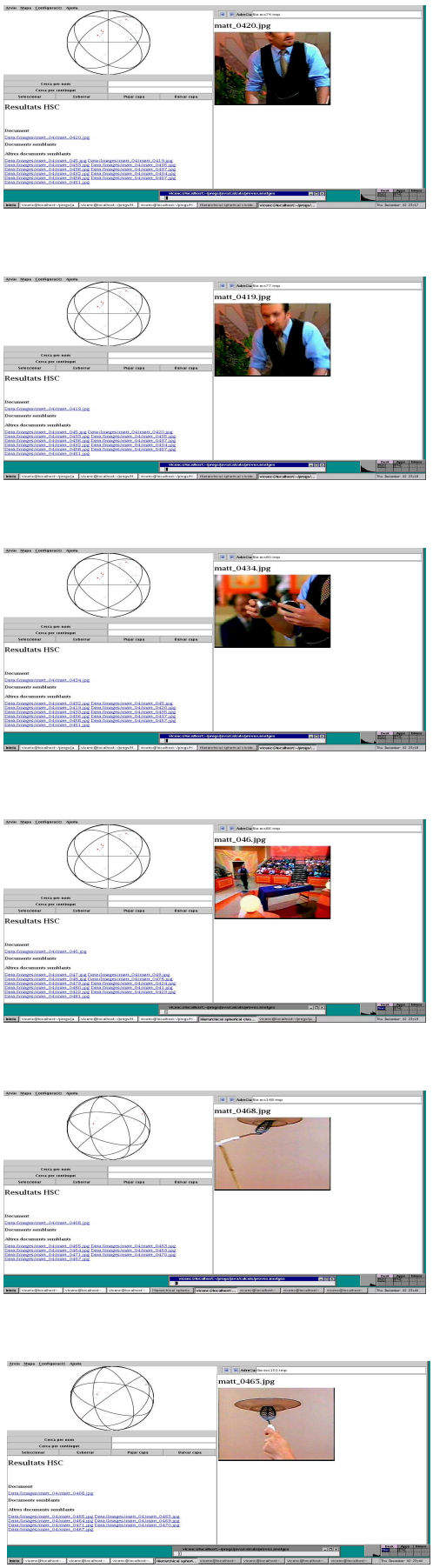

Figure 8: The example of the entertainment program: Images for shot representatives