Video Summarization: A Machine Learning Based Approach.

Santanu Chaudhury

Indian Institute of Technology Delhi

Jayanta Basak

Video Summarization: A Machine Learning Based Approach

Koustav Bhattacharya
Dept. of Electical Engg.
IIT Delhi, India
koustav79@hotmail.com

Santanu Chaudhury
Dept. of Electrical Engg.
IIT Delhi, India
santanuc@ee.iitd.ernet.in

Jayanta Basak
IBM India Research Lab
IBM, India
bjayanta@in.ibm.com

Abstract

In this paper we propose schemes for using learning in video analysis tasks like content based filtering and shot summarization. Shot segmentation is performed by our neuro-fuzzy framework, which extracts fuzzy rules for video segmentation from the trained neuro-fuzzy network. We explore Independent Component Analysis and extract Independent Components that act as features for describing the content of a shot. We prove our claim by showing simple results of a Content Based Filtering scheme based on this. We also propose a technique for summarizing content of a video shot. Unlike, keyframe based approaches we try to find out those "critical windows" from the shot sequence that best describes the content of a shot. Hierchical clustering of these windows provide the summarization of the shots. This scheme thus preserves the original component objects that make up the video thus characterizing the semantically essential information present in the video.

1. Introduction

Video Analysis is critical for information and entertainment appliances delivering video or video shots on user's demand. The sheer volume of video data makes analysis task difficult. Browsing tools are important requirement for the users to obtain a quick idea about the video content.

The development of browsing tools is a very active area of research[9, 2, 13, 5]. Browsers use as building blocks subsets of frames called keyframes, selected because they summarize the video content better than their neighbors[11]. Obviously, selecting one keyframe per shot may not always summarize the complex information content of long shots. Shots should be sampled by a higher or lower density of keyframes according to their activity level[3]. However, most video representation and summarization approaches that have appeared in the literature does ultimately rely on static arrangements of key frames. Specifically, a set of key frames is selected from each video shot and spatially arranged in a variety of pictorial summary forms[10]. Such compact representations of video provide viewers with a global picture of the entire video content on a single screen[14]. Key Frame based representations that are used by most video summarization schemes have the common drawback that they are not natural to non-experts and can be hard to grasp on a single screen, particularly when the underlying video is complex[6].

In contrast our technique compactly summarizes a video data by preserving its original component objects that make up the video which characterize the semantically essential information present in the video. Our scheme for generating summary is in terms of window in the frames of the shot and hence the summary generated is in terms of constituent objects that make up a video shot. Our approach is thus focussed on identifying the semantic content of a shot in terms of the objects that make up the shot. We thus provide a novel scheme for building description of a video. We also provide a textual summary of a video shot which assumes presence of labelled windows of simple objects which can be compared with the *critical windows* of a shot generated by our summarization scheme. This textual summary provides to the naive user a text based description of the video content in a shot. The scheme thus provides a novel way of generating automatic textual annotation of a video shot. We have also shown application of this technique for establishing similarity between video shots.

2. Video Segmentation: A fuzzy rule based approach

First step in our video summarization scheme is segmentation of the sequence into shots. We have used a neuro fuzzy rule based approach that we have proposed in [4]. We use features such as histogram difference and pixel difference[4] as input features to train a neuro-fuzzy network. Histogram difference typically considers global changes in a video sequence and is helpful to detect abrupt changes in a shot. On the other hand, a gradual change can be characterised by both global (fading) and local changes (wipe). Pixel difference is used to identify local changes in a sequence. Thus for detecting gradual changes we need a combination of histogram difference and pixel difference features. We use a neuro-fuzzy network for extracting fuzzy rules for video segmentation[4]. We train our system on large number of examples. We use three consecutive frame values of histogram difference and pixel difference as input features. The input neurons in the network consist of three π-set neurons[4] corresponding to fuzzy sets low,mid and high for each feature and all three of them are connected to a single hidden layer neuron. The hidden layer neurons are in turn fully connected to output neurons. We use error backpropation as our training algorithm. After completion of training, the training data is clustered and the cluster centres are presented as input to the network. The path that contributes maximally to the input neuron is sensed and fuzzy rules are extracted depending upon the confidence factor[4] of the winning node. The process is iterated till duplicate rules are generated. This process is carried out offline.

New sequences are segmented using the fuzzy rules generated. We found that the scheme worked reliably on large number (more than 100) of sequences from different domains.

3. Independent Components as shot descriptors

To describe the content of any shot we need a canonical representation scheme. For this, we choose a set of windows of specified size from different frames of the video shot. Considering the ensemble of these frame windows, which we name *critical windows* of the shot, we find the set of independent components characterizing these windows. We then take only the informative independent components considering the non-Gaussianity measure[7]. Each shot of the database video, can be viewed as a combination of these informative independent components. In other words, we approximate each shot window as an additive mixture of these components. The approximation error reflects how far the database shot content matches with the content of the critical windows.

3.1. Obtaining *critical windows* in a shot

We use learning to find optimal position of those windows which are expected to capture regions of interesting shot features. Further, important regions in a shot can get associated with more weightage through placement of multiple overlapped windows over the region.

Effectiveness of this scheme depends upon the choice of critical windows. We use genetic learning for identifying critical windows. Genetic algorithms provide powerful search mechanisms that can be used in optimization problems as they possess the ability to exploit the information accumulated about an initially unknown search space in order to bias subsequent searches into useful subspaces.

Genetic algorithms provide us with a novel scheme to perform *unsupervised learning* for positioning windows in a large, complex and poorly understood search space spanned by the frames of the shot. Note that learning critical window positions in the frames of a shot using GA is done in offline mode. Next, we explain the protocols followed in our genetic algorithm based learning.

3.1.1 Representation

We encode the location of window centers from various frames in a single chromosome. For a 256×256 frame image a pixel coordinate requires 16 bits. Thus for M windows we have a chromosome of length $16M$ bits, i.e., $2M$ bytes. We also keep track of the frames to which these windows belongs to.

3.1.2 Fitness Function

For evaluating fitness of each chromosome we first divide each frame of the candidate shot into a grid such that each cell in the grid gives a image window. Thus we have same sized image windows for all frames of the shot. Now given the location of the *critical window* centres that is encoded in the chromosomes as well as the respective frames to which they belong we compute the informative Independent Components of the space spanned by these *critical windows*. The cost of each chromosome or its objective value, which denotes how well these critical windows describe the content of the shot, is simply the error in approximation in expressing all the windows in the grid of all the frames in the shot by the Independent Components produced by the *critical windows*. Thus after the GA converges the fittest chromosome in the gene pool gives us those ”critical” windows in a shot that adequately describe content of the shot that is being analysed.

Thus mathematically, we consider M *critical windows* of size $w \times w$ from various frames in the shot sequence. Each window, therefore, consists of w^2 pixels. Figure 1 demonstrates the representation that we used.

We divide each frame of a shot into grid giving n windows of the same size as the critical windows and if there are f such frames in the original shot then we have ultimately N such windows, where $N = n \times f$.

Formally, if the critical windows be represented as $\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_M}$ where each $\mathbf{x_i}$ is stream (or signal) of length w^2 and if we represent the independent components derived from critical windows as $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_M}$ where each $\mathbf{y_i}$ represents a signal of length w^2, we can choose the K informative independent components based on the non-Gaussianity measure such that we have $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_K}$ independent components. Let the windows obtained after dividing each frame into a grid for all frames in the shot be represented as $\mathbf{d_1}, \mathbf{d_2}, \cdots, \mathbf{d_N}$ where each $\mathbf{d_i}$ represents a stream of w^2 pixels. We then approximate each $\mathbf{d_i}$ as a linear combination of $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_K}$ and find out the error in approximation.

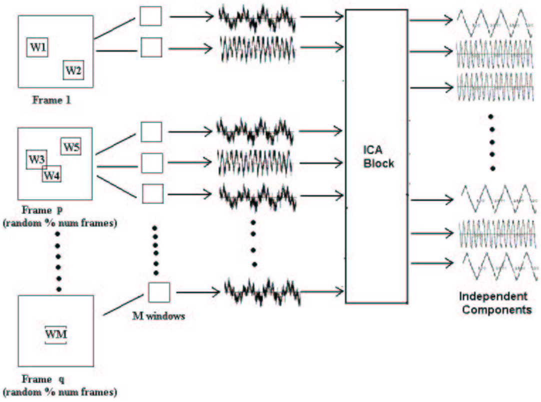

Figure 1. Representation Scheme

- Error calculation For calculating this error in approximation, we first obtain the optimal linear fit of the informative independent components to a shot window $\mathbf{d_i}$ by solving the linear regression equation given as

$$\mathbf{Y}\mathbf{a_i} = \mathbf{d_i} \tag{1}$$

where $\mathbf{Y}$ is a matrix whose column vectors are the informative independent components $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_K}$ such that $\mathbf{Y}$ is a $w^2 \times K$ matrix and $\mathbf{d_i}$ is a $w^2 \times 1$ vector. The coefficients $\mathbf{a_i}$ (which is a $K \times 1$ vector) can be solved for $K \leq w^2$. In the case $K = w^2$, we can get an exact solution provided $\mathbf{Y}$ is invertible. For $K > w^2$, this is an under-complete representation and the problem becomes ill-posed. In general, we have $K < w^2$ (overcomplete representation) which essentially gives us a constraint on the trade-off between the number of windows and the window size. An optimal solution for the overcomplete representation can be obtained by linear regression as

$$\mathbf{a_i} = (\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{d_i}) \tag{2}$$

Note that, for overcomplete representation, in general, the matrix $\mathbf{Y}^T\mathbf{Y}$ is invertible. The error in the linear regression fit can be obtained as

$$E(\mathbf{d_i}) = ||\mathbf{d_i} - \mathbf{Y}\mathbf{a_i}||^2 \tag{3}$$

Since the matrix $\mathbf{Y}^T\mathbf{Y}$ is symmetric, simple algebraic manipulation gives us

$$E(\mathbf{d_i}) = \mathbf{d_i}^T(\mathbf{I} - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T)\mathbf{d_i} \tag{4}$$

where $\mathbf{I}$ is the identity matrix.

3.1.3 Determining Fitness value of a chromosome

For every shot window we obtain an error measure $E(\mathbf{d_i})$ which reflects the error in approximating the shot window by the informative independent components obtained from the *critical windows*. Thus we obtain N such errors for N such windows in a shot. Intuitively, if $E(\mathbf{d_i}) = 0$ for any window $\mathbf{d_i}$ of a shot then it indicates that the content of that window perfectly matches with the content provided by the *critical windows* of the shot. If for all N windows the error is low then the *critical windows* encoded by the chromosome spans the largest subspace of the space provided by the all the N windows of the shot. We therefore, combine these error measures as

$$S = \sum_{i=1}^{N} V(\mathbf{d_i}) \tag{5}$$

where S denotes the score of expressivity of the critical windows encoded by a particular chromosome. We define $V(\mathbf{d_i})$ as follows:

$$V = \frac{1}{1 + \exp(m * (E(\mathbf{d_i}) - \theta))} \tag{6}$$

where m and θ being two parameters controlling the shape of function. The score function as defined above decreases slowly as the error of mismatch increases. The function can be viewed as a soft threshold function such that all chromosomes having errors of mismatch less than the threshold θ have higher scores and all other chromosomes having errors of mismatch greater than θ score very low. This score thus gives the fitness value of each chromosome.

3.1.4 Method

We initialize the chromosome population with random window centers from random frames in the shot. We used *roulette wheel* selection strategy with *elitist model* of computation where the best set of *critical windows* along with their corresponding frames are retained [8]. Mutation is performed by flipping each bit with a probability in $[0.001, 0.01]$. We performed single point cross-over with a probability of 0.75. For a shot of average length (80 frames) and average complexity we found that the Genetic Algorithm terminates after about 1300 generations when the cost of the best chromosome usually falls below a certain threshold. On completion of GA we obtain a set of critical windows and their corresponding independent components which spans best the space of the frame images represented by the video shot.

3.2. Content Based Video Filter

Using the critical windows and the ICs of a query shot we can filter out similiar video shots from a video database. Given video shots from a video database, frames of these shots are divided into a grid of same sized windows. After this we analyze the error in approximation, to express the windows of a database shot in terms of the Independent Components obtained from the learnt ”critical” windows of the query shot. For this we use the same error formulation as discussed in the previous section. Instead of thresholding and selecting/rejecting database video shots we provide a soft thresholding function as defined in the previous section and display similarity scores of the query shot to that of a database shot.

4. Automatic Textual Annotation and Hierchical Shot Summarization

In this section we discuss our technique to summarize shots and generate textual annotations to it. To summarize a shot adequately we need to cluster the selected ”critical” windows. To remove position sensitivity of objects within windows clustering should be based on *two dimensional central moment* values. We consider upto second order, two dimensional central moments which makes windows insensitive to arbitrary translations of the objects within. We note that, to make the windows insensitive to rotation or scaling clustering should be based on Hu invariant moments. Moreover, the clustering should be hierarchical so that the appearance based relationship between object components that is captured by the ”critical windows” can be made explicit. For hierarchical clustering, we use the Linkage Algorithm[12].

We then generate automatic text based annotation for a video shot by simply traversing the *dendrogram* (tree produced by Linkage Algorithm).

When two leaf levels are combined in the dendrogram they are combined by appending the keyword ”AND” in between their respective labels. When a interior node is combined they are combined by appending the keyword ”WITH” instead. Nodes generated at each level therefore can be associated with a textual annotation. Thus after combining nodes (using any of the keywords) in a dendrogram we get the final textual summary or the annotation of the shot sequence by simply reading off the label assigned to the root node.

5. Implementation and Experimental Results

In this section we analyse the results obtained for various video content analysis tasks using the techniques explored in the previous sections.

5.1. Protocol

We experimented with the system in the Matlab environment and VC++ on Windows 2000 running on Intel Xeon dual processor of 1.7 GHz and 2 GB RAM. The system was tested for about 103 test video sequences which included domains like sports, news, and commercials. For computing Independent Components we use Hyvarinen’s FastICA algorithm [1]. We provide subjective measures to demonstrate the effectiveness of our system. We gathered feedback from nine independent subjects and took the average of the rating provided by all subjects.

5.2. Results of our system as a Content Based Video Filter

VC++ was used to implement our algorithm that generates fuzzy rules from the trained neuro fuzzy network. These rules were then coded into a rule-list using Matlab’s fuzzy logic toolbox and its fuzzy inferencing engine was used to segment any test video into shots based on these fuzzy rules.

We demonstrate performance for different segmented database shots when our system was trained to the field shot of a cricket video sequence given in Figure 2. The critical windows extracted from the frames are also shown as marked squares in Figure 2. The critical windows capture essential components of the shot like scoreboard, ESPN logo, pitch, ground, gallery etc. We demonstate performance of our content based filter in Figure 3. We observe that while the second shot given in Figure 3, which is also a typical field shot from a cricket video sequence, received high scores from our system, the first shot given in Figure 3 which was from a news video sequence obtained a much lower similarity score.

5.3. Results on shot summarization

We next demonstrate results of our shot summarization algorithm. In Figure 4 we show the results of our summarization scheme for the shot sequence shown in Figure 2. We find that Clusters 1 and Cluster 3 are combined which consist of windows of ground and gallery respectively. Cluster 6 contains windows of the stand. Cluster 5 provides windows of ESPN logo. Cluster 4 provides windows of a scoreboard in the background of ”The Great Clock”(near Lords stadium in England), while Cluster 2 contains only windows with the ”Great Clock” against various background. Parsing the dendrogram by our method also generates a textual summary of the shot as ” ’The Great Clock’ WITH Scoreboard WITH ESPN WITH Stands WITH Ground and Gallery”. We found that as complexity of a shot decreases the number of clusters required to summarize the shot also decreases and hence results in a much simpler textual annotation.

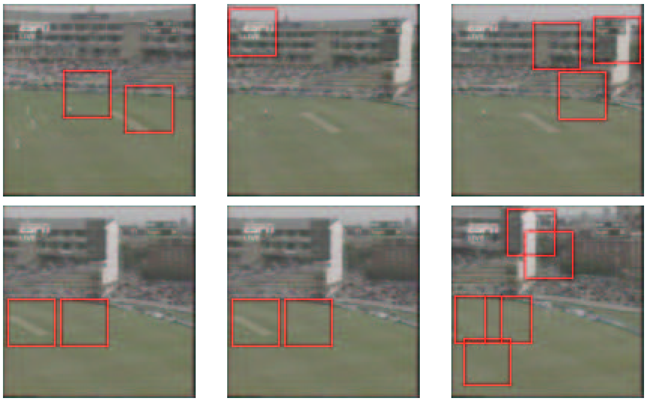

Figure 2. A training shot of cricket field sequence. The critical windows learnt by the genetic algorithm is shown by the marked squares.

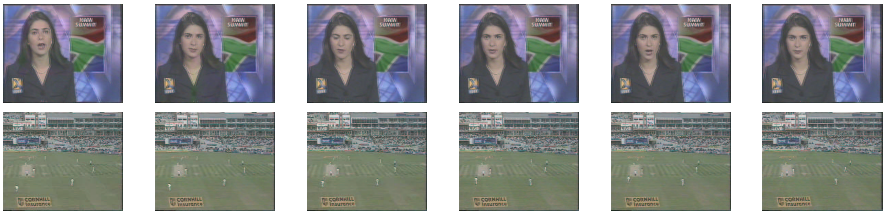

Figure 3. A subsampled news shot and a cricket shot. Our system assigned a similarity score of 0.210 to the first shot and a score of 0.893 to the second.

5.4. Subjective Rating of the Summarization Performance

We evaluate the subjective performance of our Shot Summarization system by taking subjective evaluations of 8 randomly chosen individuals and asking them to evaluate the expressivity of our summarization scheme with regard to the original shot. We found that when shot complexity was low, the summarization scheme achieved good ratings as it was easy to understand the simple textual annotation containg few predicates. For higher complexity the structure of the tree became complex which made the naive users difficult to understand the complicated structure of the textual annotation. We demonstrate the average performance of our summarization scheme using Table 1.

Domain	Num. of seq	Avg len.	Subj. Rat.(in 10)
Sports	27	75	7.76
News	21	110	8.50
Commercials	17	70	6.30

Table 1. Subjective Rating of our Summarization Scheme

6. Conclusions

In this paper we have explored the use of learning in Content Based Retrieval in the domain of videos with the Independent Component Analsysis as our basic building block. For exploring video analysis tasks like content based video filter design or shot summarization we first divide any video into shots. We then described a novel approach to the design of a content based filter based on learning the ICs of *critical windows* in a shot. We also presented the development of a shot summarization scheme based on hierchical clustering of the moments from the *critical windows* in a shot. This novel technique summarizes a video data by preserving its original component objects that make up the video which characterize the semantically essential information present in the video. We also provide a textual summary of a video shot to the naive user and thus the scheme can be looked upon as one providing automatic annotation of a shot.

References

[1] Fastica matlab implementation. *http://www.cis.hut.fi/projects/ica/fastica/code/dlcode.html*.

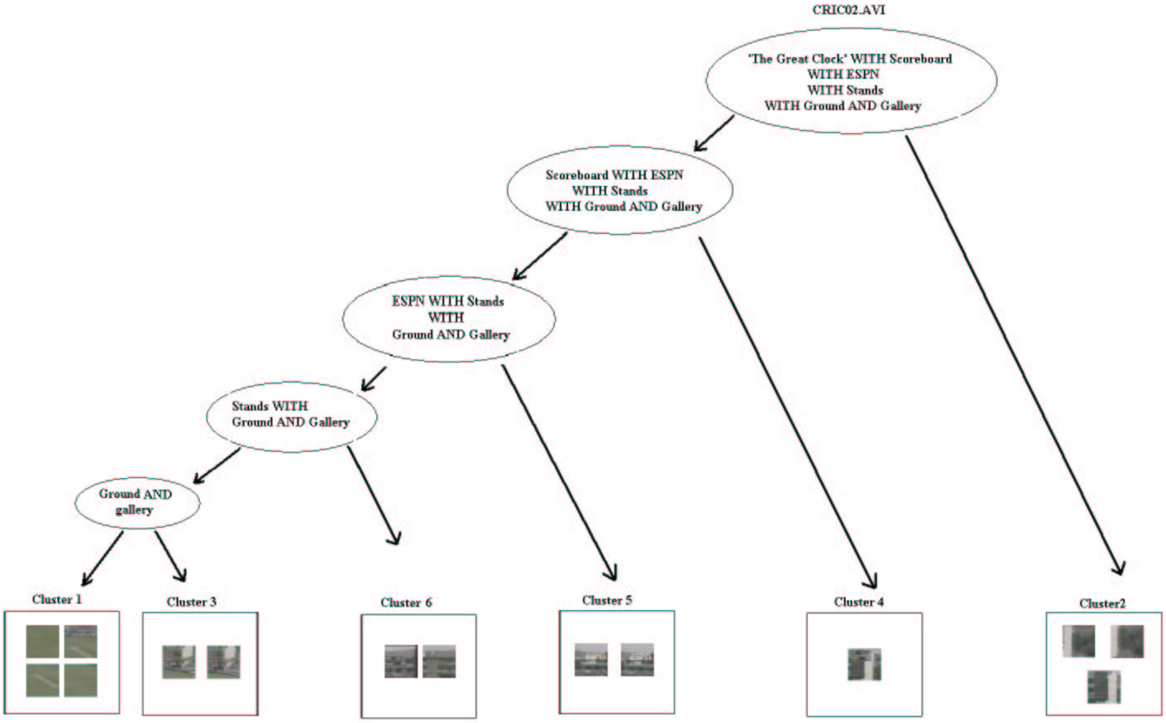
CRIC02.AVI
'The Great Clock' WITH Scoreboard WITH ESPN WITH Stands WITH Ground AND Gallery
Scoreboard WITH ESPN WITH Stands WITH Ground AND Gallery
ESPN WITH Stands WITH Ground AND Gallery
Stands WITH Ground AND Gallery
Ground AND gallery
Cluster 1
Cluster 3
Cluster 6
Cluster 5
Cluster 4
Cluster2

Figure 4. Summarizing the Video Shot of Figure 2

[2] E. Ardizzone and M. L. Cascia. Automatic video database indexing and retrieval. *Multimedia tools and applications*, 4:29 –56.

[3] Y. S. Avrithis, A. D. Doulamis, N. D. Doulamis, and S. D. Kollias. A stochastic framework for optimal key frame extraction from mpeg video databases. *Computer Vision and Image Understanding*, 75:3 – 24, 1999.

[4] K. Bhattacharya, J. Basak, and S. Chaudhury. A neuro-fuzzy technique for video analysis. *In Proceedings of The Fifth ICAPR*, pages 483 –487 , 2003.

[5] S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong. Videoq - an automatic content-based video search system using visual cues. *ACM Multimedia Conference*, 1997.

[6] W.-G. Cheng and D. Xu. Content-based video retrieval using the shot cluster tree. *Proc. of Intl. Conference on Machine Learning and Cybernetics*, 5:2901 –2906 , 2003.

[7] P. Comon. Independent component analysis - a new concept? *Signal Processing Workshop*, 36:287 –314 , 1994.

[8] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[9] F. Idris and S. Panchanathan. Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation*, 8:146 – 166, 1997.

[10] A. Komlodi and G. Marchionini. Keyframe preview techniques for video browsing. *Proc. of the 3rd ACM International Conference on Digital Libraries*, pages 118 –125 , 1998.

[11] A. Komlodi and L. Slaughter. Visual video browsing interfaces using keyframes. *Proc. of ACM Conference on Human Factors in Computing sytems*, pages 337 –338 , 1998.

[12] O. Maqbool and H. Babri. The weighted combined algorithm: A linkage algorithm for software clustering. *In Proceedings of The Eighth Euromicro Working Conference on Software Maintainence and Reengineering*, pages 15 –23 , 2004.

[13] V. Roth. *Content-based retrieval from digital video*, volume 17. Image and Vision Computing, 1999.

[14] Y. Rui, T. S. Huang, and S. Mehrotra. Constructing table-of-content for videos. *ACM Journal of Multimedia Systems*, 1998.