A Support Vector Machine Approach for Video Shot Detection

Vasileios Chasanis, Aristidis Likas, and Nikolaos Galatsanos

Department of Computer Science, University of Ioannina,
45110 Ioannina, Greece
{vchasani,arly,galatsanos}@cs.uoi.gr

Abstract. The first step towards indexing and content based video retrieval is video shot detection. Existing methodologies for video shot detection are mostly threshold dependent. No prior knowledge about the video content makes such methods sensitive to video content. To ameliorate this shortcoming we propose a learning based methodology using a set of features that are specifically designed to capture the differences among hard cuts, gradual transitions and normal sequences of frames simultaneously. A Support Vector Machine (SVM) classifier is trained both to locate shot boundaries and characterize transition types. Numerical experiments using a variety of videos demonstrate that our method is capable of accurately detecting and discriminating shot transitions in videos with different characteristics.

Keywords: Abrupt cut detection, Dissolve detection, Support Vector Machines.

1 Introduction

In recent years there has been a significant increase in the availability of high quality digital video as a result of the expansion of broadband services and the availability of large volume digital storage devices. Consequently, there has been an increase in the need to access this huge amount of information and a great demand for techniques that will provide efficient indexing, browsing and retrieving of video data. The first step towards this direction is to segment the video into smaller "physical" units in order to proceed with indexing and browsing.

The smallest physical segment of a video is the *shot* and is defined as an unbroken sequence of frames recorded from one camera. Shot transitions can be classified into two categories. The first one which is the most common is the *abrupt cut*. An abrupt or hard cut takes place between consecutive frames due to camera switch. In other words, a different or the same camera is used to record a different aspect of the scene. The second category concerns gradual transitions such *dissolves*, *fade outs* followed by *fade ins*, *wipes* and a variety of video effects which stretch over several frames. A dissolve takes place when the initial frames of the second shot are superimposed on the last frames of the first shot.

A formal study of the shot boundary detection problem is presented in [20]. In [11] the major issues to be considered for the effective solution of the

G.A. Tsihrintzis et al. (Eds.): New Direct. in Intel. Interac. Multimedia, SCI 142, pp. 45–54, 2008.
springerlink.com

shot-boundary detection problem are identified. A comparison of existing methods is presented in ([4], [10], and [15]). There are several approaches to the shot-boundary detection task most of which involve the determination of a predefined or adaptive threshold. A simple way to declare a hard cut is pair-wise pixel comparison [22]. This method is very sensitive to object and camera motions, thus many researchers propose the use of a motion independent characteristic, which is the intensity or color, global or local histogram ([17], [22]). The use of second order statistical characteristics of frames , in a likelihood ratio test, is also suggested ([12], [22]). In [21] an algorithm is presented based on the analysis of entering and exiting edges between consecutive frames. This approach works well on abrupt changes, but fails in the detection of gradual changes. In [5] mutual information and joint-entropy between frames are used for the detection of cuts, fade-ins and fade-outs. An original approach to partitioning of a video into shots based on a foveated representation of the video is proposed in [3].

A quite interesting approach is presented in [20] where the detection of shot boundaries is based on a graph partitioning problem. Finally, support vector machines with active learning are implemented to declare boundaries and non-boundaries. A Support Vector Machine classifier with color and motion features is also employed in [7]. In [8] the authors propose as features of SVMs, wavelet coefficient vectors within sliding windows.

A variety of methods have been proposed for gradual transitions detection, but still are inadequate to solve this problem due to the complicated nature of such transitions. In [22], a twin-comparison technique is proposed for hard cuts and gradual transitions detection by applying different thresholds based on differences in color histograms between successive frames. In [18] a spatio-temporal approach was presented for the detection of a variety of transitions. There is also research specifically aimed towards the dissolve detection problem. In [16], the problem of dissolve detection is treated as a pattern recognition problem. Another direction, which is followed in ([9], [11], and [14]), is to model the transitions types by presupposing probability distributions for the feature difference metrics and perform a posteriori shot change estimation. It is worth mentioning that the organization of the TREC video shot detection task [19] provides a standard performance evaluation and comparison benchmark.

In summary, the main drawback of most previous algorithms is that they are threshold dependent. As a result, if there is no prior knowledge about the visual content of a video that we wish to segment into shots, it is rather difficult to select an appropriate threshold.

In order to overcome this difficulty we propose in this paper a supervised learning methodology for the shot detection problem. In other words, the herein proposed approach does not use thresholds and can actually detect shot boundaries of videos with totally different visual characteristics. Another advantage of the proposed approach is that we can detect hard cuts and gradual transitions at the same time in contrast with existing approaches. For example, in [7] the authors propose a Support Vector Machine classifier only for abrupt cut detection. In [20], features for abrupt cuts and dissolves are constructed

separately and two different SVM models are trained. In our approach, we define a set of features designed to discriminate hard cuts from gradual transitions. These features are obtained from color histograms and describe the variation between adjacent frames and the contextual information at the same time. Due to the fact that the gradual transitions spread over several frames, the frame-to-frame differences are not sufficient to characterize them. Thus, we also use the differences between non adjacent frames in the definition of the proposed features.

These features are used as inputs to a Support Vector Machine (SVM) classifier algorithm. A set of nine different videos with over 70K frames from TV series, documentaries and movies is used to train and test the SVM classifier. The resulting classifier achieves content independent correct detection rates greater than 94%.

The rest of this paper is organized as follows: In Sections 2 and 3 the features proposed in this paper are described. In Section 4 the SVM method employed for this application is briefly presented. In Section 5 we present numerical experiments and compare our method with three existing methods. Finally, in Section 6 we present our conclusions and suggestions for future research.

2 Feature Selection

2.1 Color Histogram and x^2 Value

Color histograms are the most commonly used features to detect shot boundaries. They are robust to object and camera motion, and provide a good trade-off between accuracy of detection and implementation speed. We have chosen to use normalized RGB histograms. So for each frame a normalized histogram is computed, with 256 bins for each one of the RGB component defined as H^R, H^G and H^B respectively. These three histograms are concatenated into a 768 dimension vector representing the final histogram of each frame.

$$H = [H^R \; H^G \; H^B] \; . \tag{1}$$

To define whether two shots are separated with an abrupt cut or a gradual transition we have to look for a difference measure between frames. In our approach we use a variation of the x^2 value to compare the histograms of two frames in order to enhance the difference between the two histograms. Finally the difference between two images I_i, I_j based on their color histograms H_i, H_j is given from the following equation:

$$d(I_i, I_j) = \frac{1}{3} \sum_{k=1}^{768} \frac{(H_i(k) - H_j(k))^2}{H_i(k) + H_j(k)} \; , \tag{2}$$

where k denotes the bin index.

2.2 Inter-frame Distance

The dissimilarity value given in equation (2) can be computed for any pair of frames within the video sequence. We compute the value not only between adjacent frames, but also between frames with time distance l, where l is called the inter-frame distance as suggested in ([1], [11]). We compute the dissimilarity value $d(I_i, I_{i+l})$ for three values of the inter-frame distance l:

- l=1. This is used to identify hard cuts between two consecutive frames, so the dissimilarity values are computed for l=1, *Fig. 1(a)*.
- l=2. Due to the fact that during a gradual transition two consecutive frames may be the same or very similar to each other, the dissimilarity value will tend to zero and, as a result, the sequence of the dissimilarity values could have the form shown in *Fig. 1(b)*. The computation for l=2 usually results in a smoother curve, which is more useful for our further analysis. A typical example of a sequence of dissimilarity values for l=2 is shown in *Fig. 1(b)*.
- l= 6. A gradual transition stretches along several frames, while the difference value between consecutive frames is smaller, so we are interested not only in the difference between consecutive frames, but also between frames that are a specific distance apart from each other. As the inter-frame distance increases, the curve becomes smoother as it can be observed in the example of *Fig. 1(c)*.

Of course the maximum distance between frames for which the inter-frame distance is useful is rather small. This distance should be less than the minimum length of all transitions in the video set in order to capture the form of the transition. Thus, the choice of most of the gradual transitions in our set of videos have length between 7-40 frames.

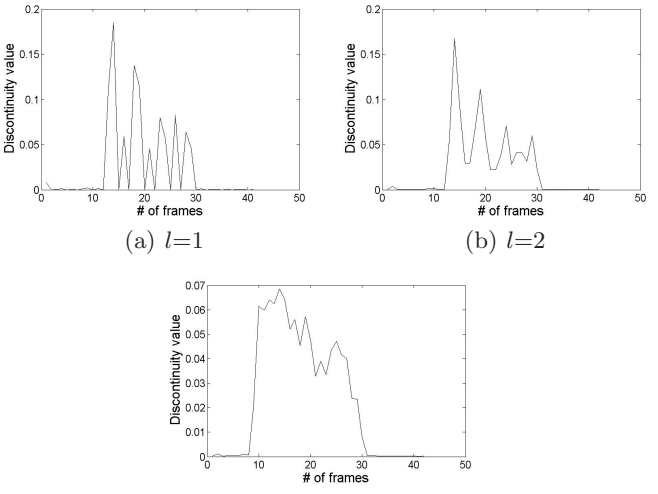
3 Feature Vector Selection for Shot-Boundary Classification

The dissimilarity values defined in Section 2 are not going to be compared with any threshold, but they will be used to form feature vectors based on which an SVM classifier will be constructed.

3.1 Definition of Feature Vectors

The feature vectors selected are the normalized dissimilarity values calculated in a temporal window centered at the frame of interest. More specifically, the dissimilarity values that are computed in section 2 form three vectors, one for each one of the three inter-frame distances l.

$$\begin{aligned} D^{l=1} &= [d(I_1, I_2), \ldots, d(I_i, I_{i+1}), \ldots, d(I_{N-1}, I_N)] \\ D^{l=2} &= [d(I_1, I_3), \ldots, d(I_i, I_{i+2}), \ldots, d(I_{N-2}, I_N)] \\ D^{l=6} &= [d(I_1, I_7), \ldots, d(I_i, I_{i+6}), \ldots, d(I_{N-6}, I_N)] \end{aligned} \quad . \tag{3}$$



(a) l=1 (b) l=2

(c) l=6

Fig. 1. Dissimilarity patterns

Moreover for each frame, we define a window of length w that is centered at this frame and contains the dissimilarity values. As a result for the i^{th} frame the following three vectors are composed:

$$\begin{array}{l} W^{l=1}(i,1:w) = [D^{l=1}(i-w/2),\ldots,D^{l=1}(i),\ldots,D^{l=1}(i+w/2-1)] \\ W^{l=2}(i,1:w) = [D^{l=2}(i-w/2),\ldots,D^{l=2}(i),\ldots,D^{l=2}(i+w/2-1)] \\ W^{l=6}(i,1:w) = [D^{l=6}(i-w/2),\ldots,D^{l=6}(i),\ldots,D^{l=6}(i+w/2-1)] \end{array} . \quad (4)$$

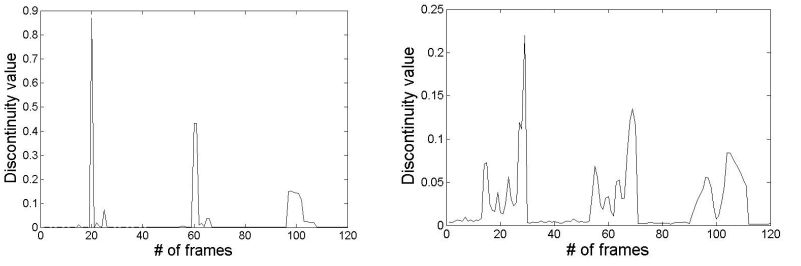
To obtain the final features we normalize the dissimilarity values in equation (4) by dividing each dissimilarity value by the sum of the values in the window. This provides the normalized "magnitude" independent features.

$$\tilde{W}^{l=k}(i,j) = \frac{W^{l=k}(i,j)}{\sum_{j=1}^{w} W^{l=k}(i,j)}, k = 1,2,6 . \quad (5)$$

The size of the window used is w=40. In our experiments we also considered windows of length 50 and 60 in order to capture longer transitions. The 120-long vector resulting from the concatenation of the normalized dissimilarities for the three windows given by

$$F(i) = [\tilde{W}^{l=1}(i)\ \tilde{W}^{l=2}(i)\ \tilde{W}^{l=6}(i)] , \quad (6)$$

is the feature vector corresponding to frame i. In what follows we show examples of the feature vectors for a hard cut and a dissolve in Fig. 2.



(a) Hard cut (b) Dissolve

Fig. 2. Feature vectors for transitions

4 Support Vector Machine Classifier

After the feature definition, an appropriate classifier has to be used in order to categorize each frame in three categories: normal sequences, abrupt cuts and gradual transitions. For this purpose we selected the Support Vector Machine (SVM) classifier [6] that provides state-of-the-art performance and scales well with the dimension of the feature vector which is relatively large (equal to 120) in our problem.

The classical SVM classifier finds an optimal hyperplane which separates data points of two classes. More specifically, suppose we are given a training set of l vectors $x_i \in \mathbb{R}^n$, i=1, . . . , l and a vector $y \in \mathbb{R}^l$ with $y_i \in \{1,\text{-}1\}$ denoting the class of vector x_i. We also assume a mapping function $\phi(x)$, that maps each training vector to a higher dimensional space, and the corresponding kernel function (eq. (9)). Then the SVM classifier [6] is obtained by solving the following primal problem:

$$\begin{array}{ll} \min\limits_{w,b,\xi} & \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\ subject\ to & y_i(w^T\phi(x_i)+b) \geq 1-\xi_i \\ & \xi_i \geq 0,\ i=1,\ldots,l \end{array} \cdot \tag{7}$$

The decision function is:

$$sqn(\sum_{i=1}^{l} w_i K(x_i,x)+b),\ where\ K(x_i,x_j)=\phi^T(x_i)\phi(x_j)\ . \tag{8}$$

A notable characteristic of SVMs is that after training, usually most of the training patterns x_i have w_i=0 in eq. (8), in other words they do not contribute to the decision function. Those x_i for which $w_i \neq 0$, are retained in the SVM model and called Support Vectors (SVs). In our approach the commonly used radial basis function (RBF) kernel is employed:

$$K(x_i,x_j) = exp(-\gamma\|x_i - x_j\|^2)\ , \tag{9}$$

where γ denotes the width of the kernel. It must be noted that in order to obtain an efficient SVM classifier the parameters C (eq. (7)) and γ (eq. (9)) must be carefully selected, usually through cross-validation.

5 Experiments

5.1 Data and Performance Criteria

The video sequences used for our data set were taken from TV-series, documentaries and educational films. Nine videos (70000 frames) were used; containing 355 hard cuts and 142 dissolves, manually annotated. To evaluate the performance of our method we used the following criteria [2]:

$$Recall = \frac{N_c}{N_c + N_m}, Precision = \frac{N_c}{N_c + N_f}, F_1 = \frac{2 \times Rec \times Prec}{Rec + Prec} , \quad (10)$$

where N_c stands for the number of correct detected shot boundaries, N_m for the number of missed ones and N_f the number of false detections. During our experiments we calculate the F_1 value for the cuts (F_{1C}) and the dissolves (F_{1D}) separately. Then the final performance measure is given from the following equation:

$$F_1 = \frac{\alpha}{\alpha + b} F_{1C} + \frac{b}{\alpha + b} F_{1D} , \quad (11)$$

where α is the number of true hard cuts and b the number of true dissolves.

5.2 Results and Comparison

In our experiments, 8 videos are used for training and the 9th for testing, therefore, 9 "rounds" of testing were conducted. In order to obtain good values of the parameters γ and C (in terms of providing high F_1 values), in each "round" we applied 3-fold cross-validation using the 8 videos of the corresponding training set. A difficulty of the problem under consideration is the generation of an imbalanced training set that contains few positives examples and a huge number of negative ones. In our approach we sample negative examples uniformly, thus we reduce their number to 3% of the total number of examples. More specifically, in our training set there are 440 positive examples (transitions) and 2200 negative examples (no transitions) on average. Finally each model of the training procedure generated on average 1276 support vectors for normal transitions, 101 support vectors for gradual transitions and 152 support vectors for abrupt transitions. We also tested our method by using larger windows of width $w = 50$ and $w = 60$. In what follows in Tables 1-3 we provide the classification results using different selections of window lengths. We notice that the performance improves as the size of the window increases. False boundaries are reduced since larger windows contain more information. The use of larger windows also helps the detection of dissolves that last longer. In order to reduce the size of our feature vector, we have also consider as feature vectors used to train the SVM classifier, those obtained from the concatenation of features extracted for l=2 and l=6, only. It can be observed (Table 4) that even with the shorter feature vector the proposed algorithm gives very good results that are only slightly inferior to the ones obtained by the longer feature vector.

Table 1. Performance results for $w = 40$, l=1, l=2 and l=6

TRANSITION TYPE	N_c	N_m	N_f	Recall	Precision	F_1
CUTS	351	4	9	98.87%	97.50%	98.18%
DISSOLVES	127	15	33	89.44%	79.38%	84.11%
AVERAGE	-	-	-	96.18%	92.32%	94.21%

Table 2. Performance results for $w = 50$, l=1, l=2 and l=6

TRANSITION TYPE	N_c	N_m	N_f	Recall	Precision	F_1
CUTS	352	3	8	99.15%	97.78%	98.46%
DISSOLVES	130	12	25	91.55%	83.87%	87.54%
AVERAGE	-	-	-	96.98%	93.80%	95.37%

Table 3. Performance results for $w = 60$, l=1, l=2 and l=6

TRANSITION TYPE	N_c	N_m	N_f	Recall	Precision	F_1
CUTS	353	2	4	99.44%	98.88%	99.16%
DISSOLVES	127	15	25	89.44%	83.55%	86.39%
AVERAGE	-	-	-	96.58%	94.50%	95.53%

Table 4. Performance results for $w = 50$, l=2 and l=6

TRANSITION TYPE	N_c	N_m	N_f	Recall	Precision	F_1
CUTS	350	5	5	98.59%	97.49%	98.04%
DISSOLVES	129	13	21	90.85%	86.00%	88.36%
AVERAGE	-	-	-	96.38%	94.21%	95.28%

To demonstrate the effectiveness of our algorithm and its advantage over threshold depended methods, we implemented three methods that use thresholds in different ways. More specifically, we implemented pair-wise comparison of successive frames [22], likelihood ratio test ([12], [22]) and the twin-comparison method [22]. The first two methods can only detect cuts, while the third can identify both abrupt and gradual transitions. The obtained results indicate that our algorithm outperforms the other three threshold dependent methods. In Table 5 we provide the recall, precision and F_1 values for our algorithm and the three methods under consideration. For our algorithm we present the results using $w = 50$, for best values (C, γ) , using all features (l=1, l=2 and l=6) and less features (l=2 and l=6). The thresholds used in these three methods were calculated in different ways. We used adaptive thresholds in pair-wise comparison algorithm, cross validation in likelihood ratio method and finally global adaptive threshold in the twin-comparison method. Especially for the dissolve detection, our algorithm provides far better results than the twin-comparison algorithm.

Table 5. Comparative results using Recall, Precision and F_1 measures

METHOD	TRANSITION TYPE					
	CUTS			DISSOLVES		
	Recall	Precision	F_1	Recall	Precision	F_1
$w = 50$, l=1, l=2 and l=6.	99.15%	97.78%	98.46%	91.55%	83.87%	87.54%
$w = 50$, l=2 and l=6.	98.59%	97.49%	98.04%	90.85%	86.00%	88.36%
PAIR-WISE COMPARISON [22]	85.07%	84.83%	84.95%	-	-	-
LIKELIHOOD RATIO [22]	94.37%	86.12%	90.05%	-	-	-
TWIN-COMPARISON [22]	89.30%	88.05%	88.92%	70.42%	64.94%	67.57%

6 Conclusion - Future work

In this paper we have proposed a method for shot-boundary detection and discrimination between a hard cut and a dissolve. Features that describe the variation between adjacent frames and the contextual information were derived from color histograms using a temporal window. These feature vectors become inputs to a SVM classifier which categorizes transitions of the video sequence into normal transitions, hard cuts and gradual transitions. This categorization provides an effective segmentation of any video into shots and thus is a valuable aid to further analysis of the video for indexing and browsing. The main advantage of this method is that it is not threshold dependent. As a future work, we will try to improve the performance of the method by extracting other types of features from the video sequence.

Acknowledgments

This research project (PENED) is co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%).

References

1. Bescós, J., Cisneros, G., Martínez, J.M., Menéndez, J.M., Cabrera, J.: A Unified Model for Techniques on Video-Shot Transition Detection. IEEE Trans. Multimedia 7(2), 293–307 (2005)
2. Bimbo, A.D.: Visual Information Retrieval. Morgan Kaufmann Publishers, Inc., San Francisco (1999)
3. Boccignone, G., Chianese, A., Moscato, V., Picariello, A.: Foveated Shot Detection for Video Segmentation. IEEE Trans. Circuits and Systems for Video Technology 15(3), 365–377 (2005)
4. Boreczky, J.S., Rowe, L.A.: Comparison of Video Shot Boundary Detection Techniques. In: Proc. SPIE Storage and Retrieval for Image and Video Databases, vol. 2664, pp. 170–179 (1996)
5. Cernekova, Z., Pitas, I., Nikou, C.: Information Theory-Based Shot Cut/Fade Detection and Video Summarization. IEEE Trans. Circuits and Systems for Video Technology 16(1), 82–91 (2006)

6. Cortes, C., Vapnik, V.: Support-vector network. Machine Learning 20(3), 273–297 (1995)
7. Dalatsi, C., Krinidis, S., Tsekeridou, S., Pitas, I.: Use of Support Vector Machines based on Color and Motion Features for Shot Boundary Detection. In: International Symposium on Telecommunications (2001)
8. Feng, H., Fang, W., Liu, S., Fang, Y.: A new general framework for shot boundary detection and key-frame extraction. In: Proc. 7th ACM SIGMM Int. Workshop Multimedia Inf. Retrieval, pp. 121–126 (2005)
9. Fernando, W.A.C., Canagarajah, C.N., Bull, D.R.: Fade and dissolve detection in uncompressed and compressed video sequences. In: Proc. IEEE Int. Conf. Image Processing, vol. 3, pp. 299–303 (1999)
10. Gargi, U., Kasturi, R., Strayer, S.H.: Performance characterization of video-shot-detection methods. IEEE Trans. Circuits and Systems for Video Technology 10(1), 1–13 (2000)
11. Hanjalic, A.: Shot-boundary detection: Unraveled and resolved? IEEE Trans. Circuits and Systems for Video Technology 12(2), 90–105 (2002)
12. Kasturi, R., Lain, R.: Dynamic Vision. In: Kasturi, R., Lain, R. (eds.) Computer Vision: Principles, pp. 469–480. IEEE Computer Society Press, Washington (1991)
13. Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer learning revisited: a stepwise procedure for building and training a neural network. In: Fogelman, J. (ed.) Neurocomputing Algorithms, Architectures and Applications. Springer, Heidelberg (1990)
14. Lelescu, D., Schonfeld, D.: Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream. IEEE Trans. Multimedia 5(1), 106–117 (2003)
15. Lienhart, R.: Comparison of automatic shot boundary detection algorithms. In: Proc. SPIE Storage and Retrieval for Image and Video Databases VII, San Jose, CA, vol. 3656, pp. 290–301 (1999)
16. Lienhart, R.: Reliable dissolve detection. In: Proc. SPIE Storage and Retrieval for Media Databases 2001, vol. 4315, pp. 219–230 (2001)
17. Nagasaka, A., Tanaka, Y.: Automatic video indexing and full-video search for object appearances. In: Knuth, E., Wegner, L.M. (eds.) Visual Database Systems II, pp. 113–127. Elsevier, Amsterdam (1995)
18. Ngo, C.W., Pong, T.C., Chin, R.T.: Video partitioning by temporal slice coherence. IEEE Trans. Circuits and Systems for Video Technology 11(8), 941–953 (2001)
19. NIST, Homepage of Trecvid Evaluation. [Online]. `http://www-nlpir.nist.gov/projects/trecvid/`
20. Yuan, J., Wang, H., Xiao, L., Zheng, W., Li, J., Lin, F., Zhang, B.: A Formal Study of Shot Boundary Detection. IEEE Trans. Circuits and Systems for Video Technology 17(2), 168–186 (2007)
21. Zabih, R., Miller, J., Mai, K.: Feature-Based Algorithms for Detecting and Classifying Production Effects. Multimedia Systems 7(2), 119–128 (1999)
22. Zhang, H.J., Kankanhalli, A., Smoliar, S.W.: Automatic partitioning of full-motion video. Multimedia Systems 1(1), 10–28 (1993)