REIHE INFORMATIK
6/95

Automatic Recognition of Film Genres

Stephan Fischer, Rainer Lienhart and Wolfgang Effelsberg
Universität Mannheim
Praktische Informatik IV
L15,16
D-68131 Mannheim

Automatic Recognition of Film Genres

Stephan Fischer, Rainer Lienhart and Wolfgang Effelsberg

Praktische Informatik IV

University of Mannheim, 68131 Mannheim, Germany

{fisch, lienhart, effelsberg}@pi4.informatik.uni-mannheim.de

Abstract

Film genres in digital video can be detected automatically. In a three-step approach we analyze first the syntactic properties of digital films: color statistics, cut detection, camera motion, object motion and audio. In a second step we use these statistics to derive at a more abstract level film style attributes such as camera panning and zooming, speech and music. These are distinguishing properties for film genres, e.g. newscasts vs. sports vs. commercials. In the third and final step we map the detected style attributes to film genres. Algorithms for the three steps are presented in detail, and we report on initial experience with real videos. It is our goal to automatically classify the large body of existing video for easier access in digital video-on-demand databases.

1. Introduction

The first generation of multimedia workstations and PCs presented digital video and audio to the user without doing any content processing. Audio and video were integrated into the user interface, and the operating system extensions and application software concentrated on maintaining continuous streams. Video compression and decompression, already quite demanding and sophisticated, were typical functions executed on a stream [11] [3]. There are still many unsolved problems with stream handling, in particular in distributed multimedia systems; examples are real-time multicast [7], quality-of-service specification and mapping, forward error correction and advance reservation.

But a general-purpose computer can do more than just route continuous media streams. Only recently researchers have begun to use the computer for content recognition of digital video. Pioneer work describes cut-detection algorithms that identify scene cuts in compressed streams [1]. Other groups are working on the computation of linear transformations, such as zooms and pans, in compressed or uncompressed video [15]. Unlike that for digital videos, pattern recognition for still-images has a long history and has reached a mature level [13]. In our context, object recognition is of particular interest, and recent work shows that it is now feasible to use images to query still-image databases [20] [16].

Our CoP project (Content Processing) aims to combine existing techniques and develop additional algorithms in order to understand as much as possible about the content of films. Automatic content recognition can be used to classify and index the huge amounts of existing stored video. It can also be used to select those parts of online video relevant to an individual. In the years to come the prevalent problem will no longer be to how to get access to multimedia information, but how to automatically filter out the relevant pieces. Pioneering work in this area is also reported by the National University of Singapore [24].

We describe here content processing for the recognition of video genres, such as news casts, sports, commercials or cartoons. Our input is uncompressed digital video; each frame is an RGB pixel image. We process the video in three steps, at increasing levels of abstraction:

- The *syntactic properties* of a video are extracted. We compute color statistics, cut detection, motion vectors, simple object segmentation and audio statistics at this level.
- *Style attributes* are derived from the syntactic properties. Examples are scene lengths, camera motion (panning, zooming, parallel drive), scene transitions (i.e. cuts vs. fades vs. morphs), object motion, speech vs. music etc. Film directors use such style elements for artistic expression.

- The *style profile* of a video is compared to profiles typical of the various video genres, and an "educated guess" is made as to the genre to which a film belongs.

The algorithms used in the three steps are described in Section 2 of the paper. Section 3 reports experimental results for five genres. Section 4 concludes the paper.

2. A Three-Step Approach to Genre Recognition

The basis for our recognition process is a compressed digital video on disk. We accept either MPEG-1 or Motion JPEG streams. As the stream is read from the disk, it is decompressed on the fly, frame by frame. If the workstation is equipped with appropriate decompression hardware, this is invoked by the decompression module; otherwise decompression is done in software. The decompressed video is then run through our three steps, as shown in Figure 1. A detailed description of each step follows.

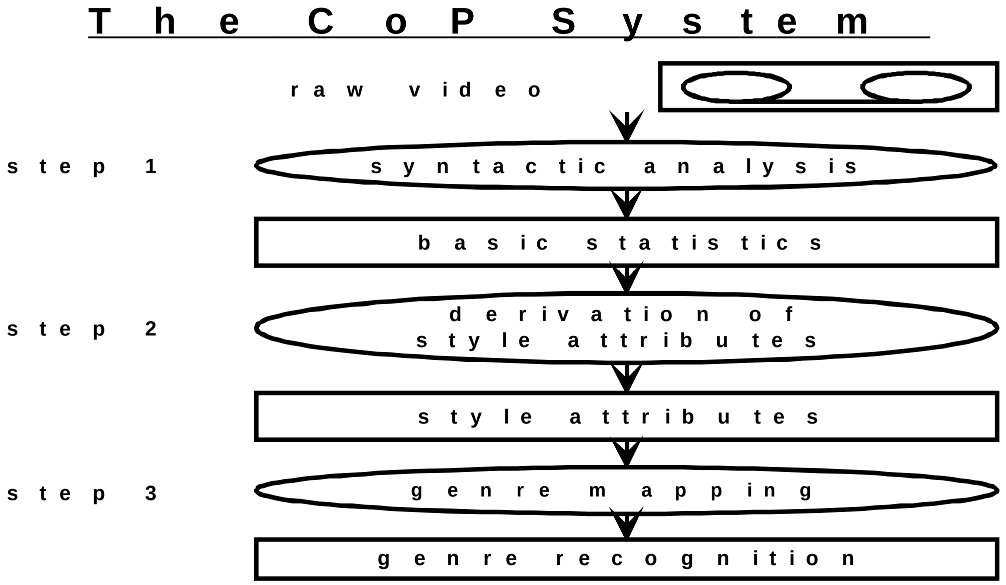

Figure 1: Genre recognition in three steps

2.1 Step 1: Syntactic Analysis of Digital Videos

In the first step, syntactic analysis, we compute simple statistics for the sequence of RGB frames.

2.1.1 Color Statistics and Cut Detection

The color contents of the frames and color change over time are the most important basic data. We compute the color histogram for each frame as a basis for cut detection and motion energy. The cut detection used is based on histogram comparison, similar to [22].

A general problem with color histograms is the precision-performance trade-off: If every pixel is sampled in every frame, precision is high, but performance is low. If sub-sampling is used, there might no longer be enough detail in the histograms to precisely identify cuts or camera motion. Therefore we combine color histograms with the results of motion detection which has to be computed in step 1 anyway (see below). We increase histogram detail only where needed: Our cut detection normally runs with a low spatial and temporal resolution, sampling only every 4th pixel. The color histogram module buffers a small number of frames. We re-process these frames at full resolution when vector flux indicates a cut; this is the case when we observe vectors at all sizes in all directions. This combines high resolution, where needed, with good overall performance.

Another color statistic on a per-frame basis is the estimated standard deviation of the color values. Calculated as the sample mean of the estimated standard deviation of the red, green and blue component of the whole frame, it is used here to recognize monochrome frames in a video stream: they have a minimum standard deviation. Consecutive monochrome frames are grouped into a monochrome frame block. An example of the occurrence of monochrome frame blocks in videos are separators in commercials.

2.1.2 Motion Detection

Color statistics can be used not only for cut detection, but also for motion analysis. We compute block-wise difference histograms between subsequent frames, similar to [4]. In a very simple experiment we only compute to total amount of motion in a scene, and we call it *motion energy*. It contains both camera motion and object motion. We can then classify scenes as high-, medium- or low- motion scenes. We compute the motion energy by simple image subtraction applied to consecutive images, smoothed with a spatial Gaussian filter. The sum of the absolute differences serves as the motion energy indicator. Examples of motion energy statistics are shown in Figure 6.

The motion energy computed from color statistics does not always deliver enough information on object and camera motion; thus we also compute motion vector fields using the optical flow technique. Having initially used the algorithm proposed by Horn and Schunck, originally designed for motion in fluids [9], we are also testing other algorithms, e.g. from [14].

These algorithms allow us to distinguish camera motion, such as panning or zooming, from object motion. When the camera moves, all blocks are linearly transformed in the same way, whereas object motion only affects some of the blocks within a frame. In many cases, camera motion and object motion occur simultaneously, but we can computationally subtract the camera motion out of the movie. The amount of camera motion and object motion in a film is an important style attribute. The motion detected in a clip is stored in accumulated form.

2.1.3 Pattern Analysis and Object Segmentation

In principle, object recognition would also be very helpful to identify a film genre. However, it is a most difficult and computationally intensive task.

In a newly developed algorithm for object segmentation we take advantage of the fact that we often have moving objects in a video. A moving object can be segmented based on the fact that all its pixels are moving at the same speed in the same direction, and that they are the only pixels moving in this manner. We use the motion vector fields computed in step 1, as described above. Once we have computed the camera movements (panning, zooming, tilting) and fading, we are able to correct the motion vector field by subtracting the camera motion, leaving only a vector field of pure object motion. As moving objects have parallel motion vectors, this new vector “image” is rather easy to segment using the Watershed algorithm proposed by Vincent and Soille [17]. In this way we obtain object boundaries of moving objects. This approach works well for moving objects and is also much faster than traditional still-image algorithms.

Object segmentation already allows us to "cut" objects "out" of a movie, but we do not yet have a database of predefined objects other than logos with which to compare them.

2.1.4 Audio Statistics

In the multimedia literature, researchers often concentrate on the analysis of the *image* components of films while ignoring audio. In step 1 we also record basic audio frequency and amplitude statistics. These help us to determine phases of speech, music, silence and noise in step 2. Again, those are important style attributes. Examples of audio statistics are shown in Figures 7 and 8.

2.2 Step 2: Derivation of Style Attributes

After completing step 1, we have a number of basic statistics on color, motion, content patterns and audio for each scene in the video clip. In step 2 we now try to assign *semantics* to the scenes. We start with a small number of *style attributes* and explain how we derive them from the basic statistics.

2.2.1 Scene Length and Scene Transitions

Cut detection is used to decompose the video into scenes, and the scene length is stored together with each scene. Our algorithm is similar to the one described in [22]. The main difference is that we do not apply cut detection several times recursively, but use sudden changes in motion as an additional indicator, as described above. Our algorithm is quite reliable, detecting more than 95 % of all cuts.

In our current implementation, only hard cuts are accepted as scene separators. There might be other scene transitions in films, e.g. fades or morphs. These are often used as an artistic style element. Hard cuts and fades are typical for feature films while wipes, blocks, band slides, etc. are more frequent in sportscasts, news and music videos. But in fact, most scene transitions *are* hard cuts, and this is currently our basis. It turns out that the scene length as such is already an important style attribute.

All other style attributes are then computed on a per-scene basis.

2.2.2 Camera Motion and Object Motion

Using the motion statistics described in 2.1.2, we are able to calculate camera motion: panning, tilting or zooming. We first identify the motion vector direction with the highest frequency. As more than one maximum can occur in a vector histogram, we calculate the normal distribution with the lowest error approximating the vector distribution. In our experience, using this normal distribution reduces the classification error in finding the correct camera panning direction to less than 10%.

We store each detected camera motion with the scene. Examples of motion intensity are shown in Figure 5. The intensity of camera motion turns out to be a distinguishing film style attribute, as we will see below.

2.2.3 Object Recognition

Decades of research in machine vision and computer-based image analysis now enable us to recognize simple objects or patterns in well-defined environments [8]. Very interesting results on face recognition have been reported recently [18]. We might soon be able to recognize the face of an actor in a movie.

TV channels often use typical patterns in their productions, e.g. a logo for news casts. Therefore we examine the video, trying to recognize predefined patterns stored in a database. Our experience shows that because of their fixed size many patterns can be identified on the basis of a color histogram. The logo of a TV channel typically always has the same size and color components (within a certain range, due to quantization and noise). An example of the color content of the "Tagesschau" logo is shown in Figure 2.

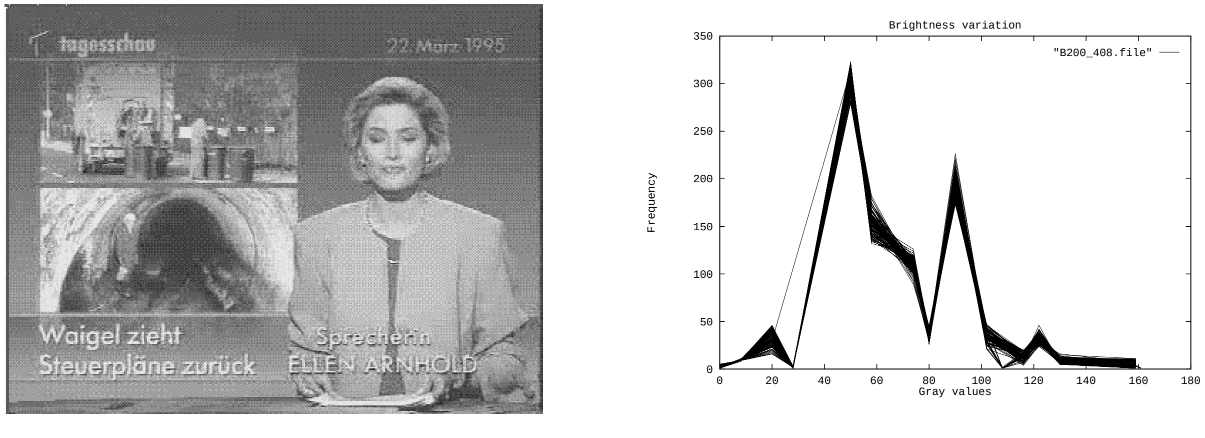

Figure 2: Pattern Recognition: (a) "Tagesschau" logo (b) Variance in color statistics in different occurrences

So far, we have implemented the logo recognition algorithm. We look for a logo in every 4th frame; if we find one, we store it with the scene statistics.

2.2.4 Semantics of Audio

Using the amplitude and frequency statistics of a scene derived in step 1, we are able to distinguish speech phases from music, noise or silence: speech has a characteristic frequency spectrum, music has a beat, noise has none of these but an amplitude above a certain threshold, and silence has amplitudes staying below the threshold. It turns out that amplitudes alone ("loudness") are not a very helpful criterion; however, the analysis of frequency data in combination with amplitudes is indeed a powerful discriminator. For example, there are always pure speech phases in a newscast, but almost never in sports.

To distinguish between speech, music and noise, we use the Fourier-transformed signal of the digital audio. Figure 8 shows the well-known fact that human speech has a very limited frequency spectrum; in contrast, the noise of a car race shows a much more homogeneous spectrum. Currently we are able to distinguish between speakers and noise with a probability better than 95 percent. For the distinction between noise and music, we are not quite as adept. Silence detection is well know from telephony [2] [19] and works very reliably in our context as well.

2.3 Step 3: Mapping Style Attributes to Film Genres

Human beings can recognize the genre of most TV broadcasts in a split second. This ability is based on finger-print-like style attributes of each genre. Thus, we should be able to define *style profiles* which are characteristic of specific film genres. These profiles are extracted from large amounts of analyzed video.

A video is classified in step 3 of our methodology in two rounds. In the first round we run as many classification modules as we have style attributes: Each module processes one film style attribute and, according to its estimation function, outputs for each genre its estimated likelihood that the style attribute indicate that genre. The likelihood values range from unlikely over indifferent to likely. A classification module is comparable to a human expert who is asked about his/her evaluation. Thus the first round collects the estimates of different "experts" which we call classification modules, one for scene lengths and transition styles, one for camera motion and object motion, one for the occurrence of recognized objects in the film, and one for audio. In the second round the estimates are combined into a final guess.

We now describe the algorithm used in the classification modules in more detail. We have currently implemented five genres which we will use throughout our examples: news cast, car race, tennis, animated cartoon, and commercials.

Each classification module i works up the film style attributes to a Fuzzy Set in G with G = {news cast, car race, tennis, commercials, animated cartoon} [21]. A membership function value of 0.5 for a genre indicates that a style attribute does not indicate any genre, while a membership function value of 1.0 indicates a very high likelihood of a particular genre, and a member function value of 0 suggests a very high likelihood that the film does not belong to that genre. As an example we present a classification module in pseudo-code in Figure 3.

There are several reasons for developing different and independent classification modules:

- Reduction of the complexity within a module.
- Addition of new knowledge to the matching process without having to change existing classification modules.
- Easy combination of different techniques developed by different researchers.

The second round integrates the output of the classification modules, finally deciding the genre of the film. The decision is based on weighted averages. Similar to human experts, classification modules may differ widely in their range and depth of knowledge. Experience teaches that genres are not homogeneous. For instance, sports casts have very few attributes in common: the style attributes of car racing are quite different from those of tennis or soccer. While the sheer breadth of the field of "sports" precludes its recognition as such, sports genres themselves, such as tennis, are easily recognized.

```
If (monochrome frame block of length 2 to 5) {
    set number_of_commercials = 0
    $f_C(g) = 0 \quad \forall g \in G$
    while (another monochrome frame block of length 2 to 5 follows within 100 s ) {
        if (time between to monochrome frame blocks > 6 sec.) {
            commercial block identified
            $f_C$(commercial) = 1
            number_of_commercials += 1
            if (still motion frames are at the end of a commercial spot) {
                    use pattern recognition tool from 2.1.3 to extract text
            }
            save commercial characteristics
        }
    }
}
output Fuzzy Set C
```

Figure 3: Pseudo-code for a classification module

3 Experiments

Having explained our theoretic framework and our algorithms in Section 2 we now present a complex, real-world example.

3.1 Video Clips

Obviously there exist many different film genres, with many subgenres. Examples are news casts, sports, talk shows, feature films, commercials, soap operas, music videos, educational and scientific broadcasts and many more. Each of them can have subgenres. So far, we have tested our algorithms on five genres:

- news cast
- sports: car race
- sports: tennis
- commercials (a block of twelve), and
- animated cartoon.

From each genre we picked out two typical sample videos of about four minutes' length each. The videos were recorded from German television in S-VHS format and then captured at a rate of 15 fps (=15 * 4 * 60 = 3600 frames) with a frame size of 384x288 pixels, on a Sun workstation with a Parallax video board and a DEC Alpha with a J300 board. The compressed format was Motion JPEG. The videos were analyzed with the algorithms presented above.

3.2 Experimental Results

3.2.1 News Cast

As far as news cast recognition is concerned, we observe a characteristic pattern of speaker and non-speaker scenes. The appearance of the channel's newscast logo is also typical. The motion-energy indicator reveals that traditional newscasts always consist of alternating low-motion speaker scenes and high-motion video inserts; Figure 6 clearly shows the low-motion and high-motion phases. As we currently use motion estimators only, without object recognition, a news speaker scene cannot be distinguished from a similar speaker scene in a block of commercials. However, a distinguishing property is that the same speaker returns after a video insert in a news show. Therefore we compare the histograms of three subsequent scenes of low motion. We divide a single frame into *n* pixel blocks and compare the color histograms block-wise. This also takes care of a possible difference in the background. Our experience shows that it suffices to have *n*=9 out of 25 pixel blocks out of a frame; if nine blocks are identical, we assume that we are seeing the same speaker as before.

3.2.2 Car Race

As far as the recognition of sports is concerned we realized very early on in our experiments that there is almost nothing in common between videos of different kinds of sports. For example, if we compare a car race with a tennis match, scene lengths are much shorter in the car race, there is much more camera motion, there is more object motion, and the audio is mostly noise at high amplitudes. Thus it is much easier to distinguish a car race from tennis than it is to distinguish tennis from some scenes of another genre (see Figures 6 and 7). We decided to give up on sports as a homogeneous genre, and to look at subgenres of sports only.

Car racing has a unique combination of style attributes, making it easy to identify. We could distinguish car races from tennis and soccer without problems.

3.2.3 Tennis

Tennis is a very good example of the discriminating power of audio: the bouncing of the ball can be clearly identified. If we look at the tennis audio in Figure 7, it starts with a noise-only phase. In the wave form we can clearly distinguish the bouncing of the ball as singular peaks. A speaker phase follows; in the Fourier transformation of that phase, the typical frequency pattern of speech can be observed (see Figure 7). Such alternating bouncing-ball and speaker phases characterize the audio of tennis.

3.2.4 Commercials

In many countries the commercials within a commercial block are separated by up to 5 monochrome frames, typically in black. We identify these monochrome frames by a standard color deviation below 10, using the algorithm described in Section 2.1.1. Thus, if our monochrome frame detection tool finds a sequence of several monochrome frame blocks of up to 5 frames over a distance of 8 to 60 sec. we can conclude with high probability that we are within an commercial block. Table 1 shows the small color variance in the monochrome frames separating the commercials.

block	first frame	last frame	block length	color variance	finished spot
1	481	484	4	7.9 to 8.2	1
2	570	572	3	5.6 to 9.7	
3	596	600	5	5.4 to 7.0	
4	787	788	2	5.4 to 9.8	
5	822	823	2	5.5 to 7.3	
6	873	874	2	7.3 to 8.6	
7	944	947	4	8.1 to 8.4	2
8	1096	1097	2	8.1 to 8.2	3
9	1328	1330	3	8.0 to 8.2	4
10	1634	1637	4	8.0 to 8.2	5
11	1944	1947	4	6.2 to 8.1	6
12	2241	2244	4	7.9 to 8.3	7
13	2704	2706	3	7.8 to 8.2	8
14	3312	3314	3	8.0 to 8.3	9
15	3764	3767	4	8.0 to 8.2	10
16	4075	4077	3	7.8 to 8.1	11
17	4449	4452	4	7.9 to 8.3	12

Table 1: Monochrome frames as separators between commercials

In rare cases monochrome frame blocks appear within a commercial scene or other genres due to the use of “fade from black” or “fade to black” in scene transitions. For instance, the second commercial spot (frame 485 to 943) has such a transition. However, the fades can be identified as such because the linear transformations on the color statistics are detected, as described in Section 2.1.2.

Currently we are developing an OCR tool which automatically grabs the text of the last five frames of a commercial. In most cases the words found there include the company name and/or the product name, so that a commercial can be identified by text-pattern matching.

3.2.5 Cartoons

In the cartoon we observe scene lengths that are longer than in other genres (see Figure 4). Also there is much less camera motion (see Figure 5).

Audio is also an interesting attribute of cartoons. Figure 7 shows that there are periods of zero amplitude (absolute silence) between noise or music or speech periods. The reason could be that audio for cartoons is typically produced in a studio, where there is no background noise. All shots taken in the real world have background noise even in phases of (relative) silence.

3.3 Interpretation of the Experiments

From the discussion above and from the Figures, it becomes clear that no single attribute is sufficient to uniquely identify a genre. However, a *style profile* based on all the attributes can be defined that allows a much more reliable classification of video.

Whereas object recognition in still images is known to prove quite difficult, it is surprising how far we can get in motion pictures using brute-force statistics.

4 Conclusions and Outlook

We have presented the CoP system and its three-step methodology for automatically detecting film genres in digital video. We have implemented the proposed algorithms, and initial experience with genre recognition is very promising. But there is much more work to be done, new algorithms for additional style attributes are currently being implemented and tested.

Fundamental to our approach is the use of a *combination of many different style attributes* of a video for content recognition. Only experience can show what the most significant attributes are, and what the style profiles of all major video genres are in terms of those attributes.

In the area of object recognition, we have only implemented simple pattern matching and object segmentation. More sophisticated techniques for object identification will be integrated into the CoP system in the future.

We have only presented a small number of examples here. It will be very interesting to add feature films, music videos etc., and subgenres for each of them. We are now in the process of establishing a much broader database. Of course, we do not expect to ever reach a precision of 100% with our genre recognition system; more experience will show how close we can get.

Currently we are far removed from processing videos in real-time; our algorithms are quite demanding in terms of resources. We intend to implement them on a parallel processor (KSR/2 under OSF/1) for better performance.

Having discussed here video content analysis with the purpose of genre recognition, it is our ambitious goal to use similar techniques for *content understanding*, e.g. the automatic detection of violence in movies.

Important remark for the referees: If this paper is accepted for the conference, we will present an accompanying video with the talk, showing the sample clips and their graphs. A videotape has been submitted to the demonstrations chairman of ACM Multimedia 95 (Tom Little).

References

[1] Farshid Arman, Arding Hsu, and Ming-Yee Chiu: Image Processing On Compressed Data For Large Video Databases. Proc. ACM Multimedia 1993, ACM, New York, 1993, pp. 267-272.

[2] P.T. Brady: A Technique for Investigating On-Off Patterns of Speech, Bell Systems Technical Journal, Vol. 44, pp. 1-22 (1965)

[3] A. Cramer, M. Farber, B. McKellar, and R. Steinmetz, "Experiences with the Heidelberg multimedia communication system: multicast, rate enforcement and performance," in Proc. 4th IFIP Conference on High Performance Networking, Liège, Belgium, pp. D4-1 - D4-20, IFIP, Dec. 1992.

[4] Nevenka Dimitrova and Forouzan Golshani. R for Semantic Video Database Retrieval. Proc. ACM Multimedia 1994, San Francisco, October 1994, pp. 219-226

[5] George R. Doddington. Speaker Recognition - Identifying People by their Voices. Proceedings of the IEEE, Vol. 73, pp. 1651-1664, Nov. 1985.

[6] D. Dubois and H. Prade. Fuzzy Sets and Systems: Theory and Applications. New York, London, Toronto, 1980.

[7] D. Ferrari, A. Banerjea, and H. Zhang: Network support for multimedia - A discussion of the Tenet approach, Computer Networks and ISDN Systems, pp.1267--1280, July 1994.

[8] Rafael C. Gonzales and Richard E. Woods: Digital Image Processing, Addison Wesley Publishing Company, Reading, Massachussetts, 1993.

[9] B. K. P. Horn and B. G. Schunck. Determining optical flow. Artificial Intelligence, 17, pp. 185-204 (1981)

[10] Christopher J. Lindblad, David J. Wetherall, and David L. Tennenhouse: The VuSystem: A Programming System for Visual Processing of Digital Video. Proc. ACM Multimedia 94, San Francisco, CA, Oct. 1994, pp. 307-314.

[11] Ketan Patel, Brian C. Smith, and Lawrence A. Rowe. Performance of a Software MPEG Video Decoder. Proceedings of ACM Multimedia 93, pp. 75-82, Anaheim, CA, USA, August 1993.

[12] H. Rommelfanger. Decision under uncertainty (in German). Springer Verlag Berlin, Heidelberg, New York, 1988.

[13] Azriel Rosenfeld and Avinash .C. Kak: Digital Picture Processing, Academic Press, London (1982).

[14] A. Singh: An estimation-theoretic framework for image-flow computation, ICCV, pp. 168-177 (1990).

[15] Brian C. Smith. Fast Software Processing of Motion JPEG Video. Proc. ACM Multimedia 94, pp. 77-88, San Francisco, CA, USA, October 15-20, 1994

[16] D. Swanberg, C. F. Shu, and R. Jain, "Architecture of a multimedia information system for content-based retrieval," in Third International Workshop on network and operating system support for digital audio and video, (San Diego, California), pp. 345--350, IEEE Computer and Communications Societies, Nov. 1992.

[17] L. Vincent and P. Soille. Watershed in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 6, June 1991.

[18] Wu, Jian Kang; Narasimhalu, Arcot Desai: Identifying faces using multiple retrievals. IEEE MultiMedia, Vol. 1, No. 2, p.27-38 (1994).

[19] Y. Yatzuzuka: Highly Sensitive Speech Detector and High-Speed Voiceband Data Discriminator in DSI-ADPCM, IEEE Trans. Communications, Vol. 30, No. 4, pp. 739-750 (1982)

[20] Yew-Hock, Ang; Narasimhalu, Arcot Desai; Al-Hawamdeh, Suliman: Image information retrieval systems, in: Handbook of pattern recognition and computer vision. Editor(s): Chen, C. H.; Pau, L. F.; Wang, P. S. P. River Edge, NJ: World Scientific Publishing Co., Inc. 1993. p. 719-739.

[21] L. A. Zadeh. Fuzzy Sets as a Basis for a Theory of Probability. Fuzzy Sets and Systems, 1978.

[22] HongJiang Zhang, Atreyi Kankanhalli, and Stephen W. Smoliar. Automatic partitioning of full-motion video. Multimedia Systems, Vol. 1, No. 1, pp. 10-28, 1993.

[23] HongJiang Zhang and Yihong Gong and Stephen W. Smoliar and Shuang Yeo Tan: Automatic Parsing of News Video, Proc. IEEE Conf. on Multimedia Computing and Systems, 1994

[24] HongJiang Zhang and Stephen W. Smoliar: Developing power tools for video indexing and retrieval, Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases, San Jose, CA, 1994

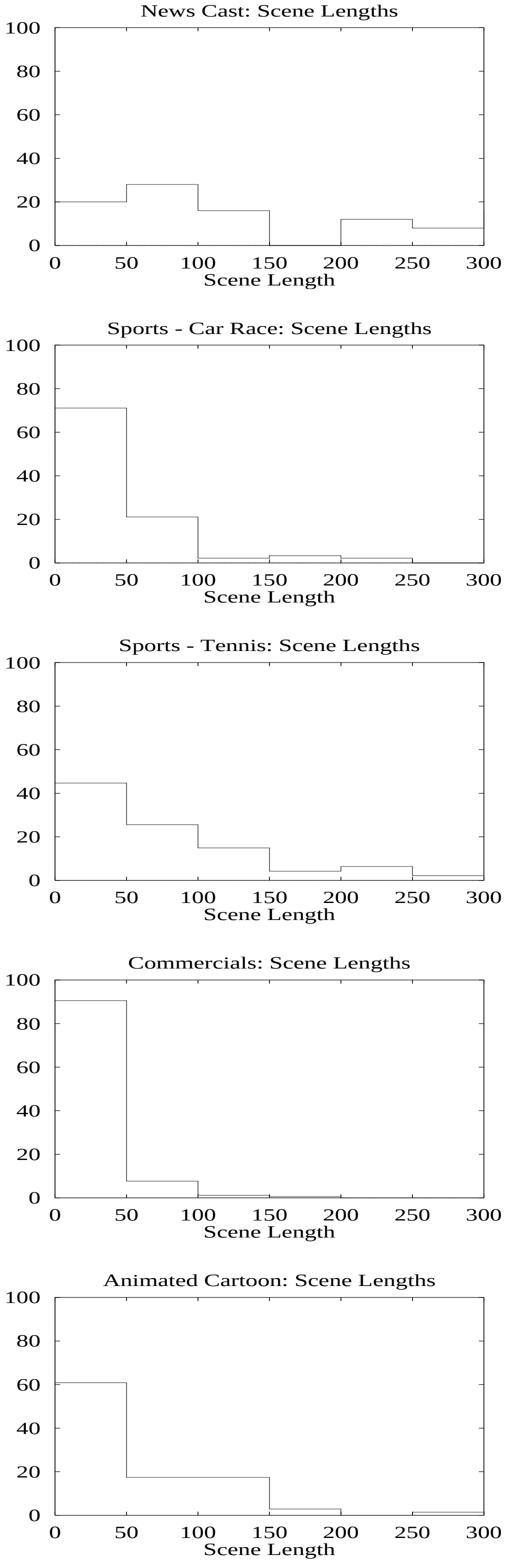

Figure 4: Scene lengths derived from color statistics

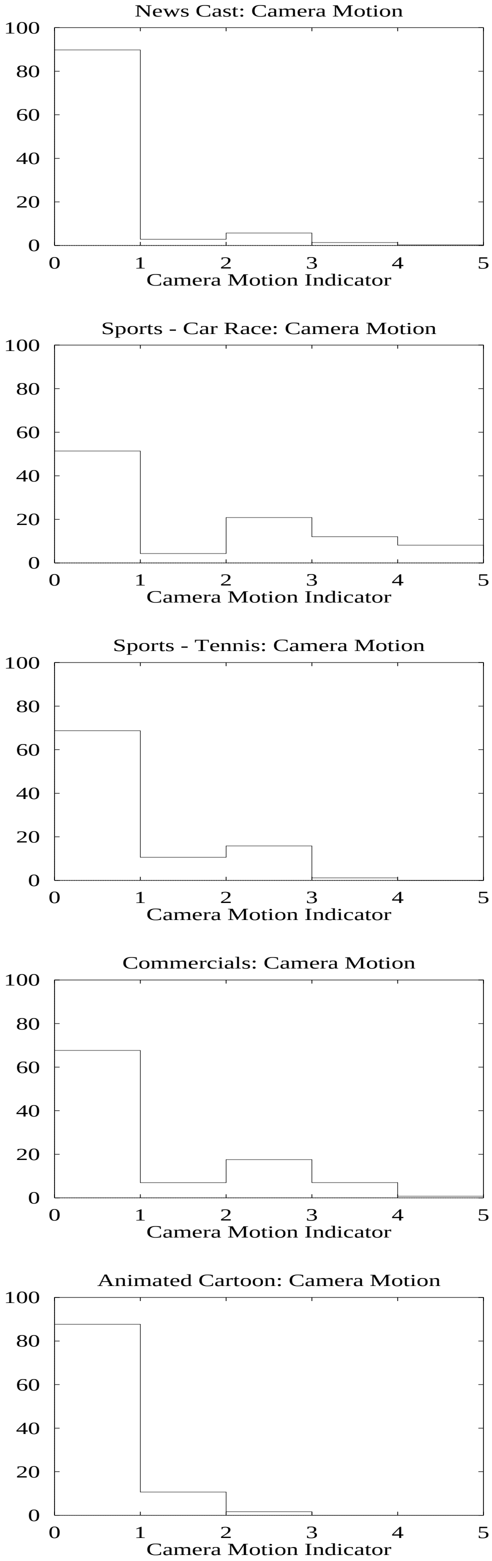

Figure 5: Camera motion derived from motion vectors

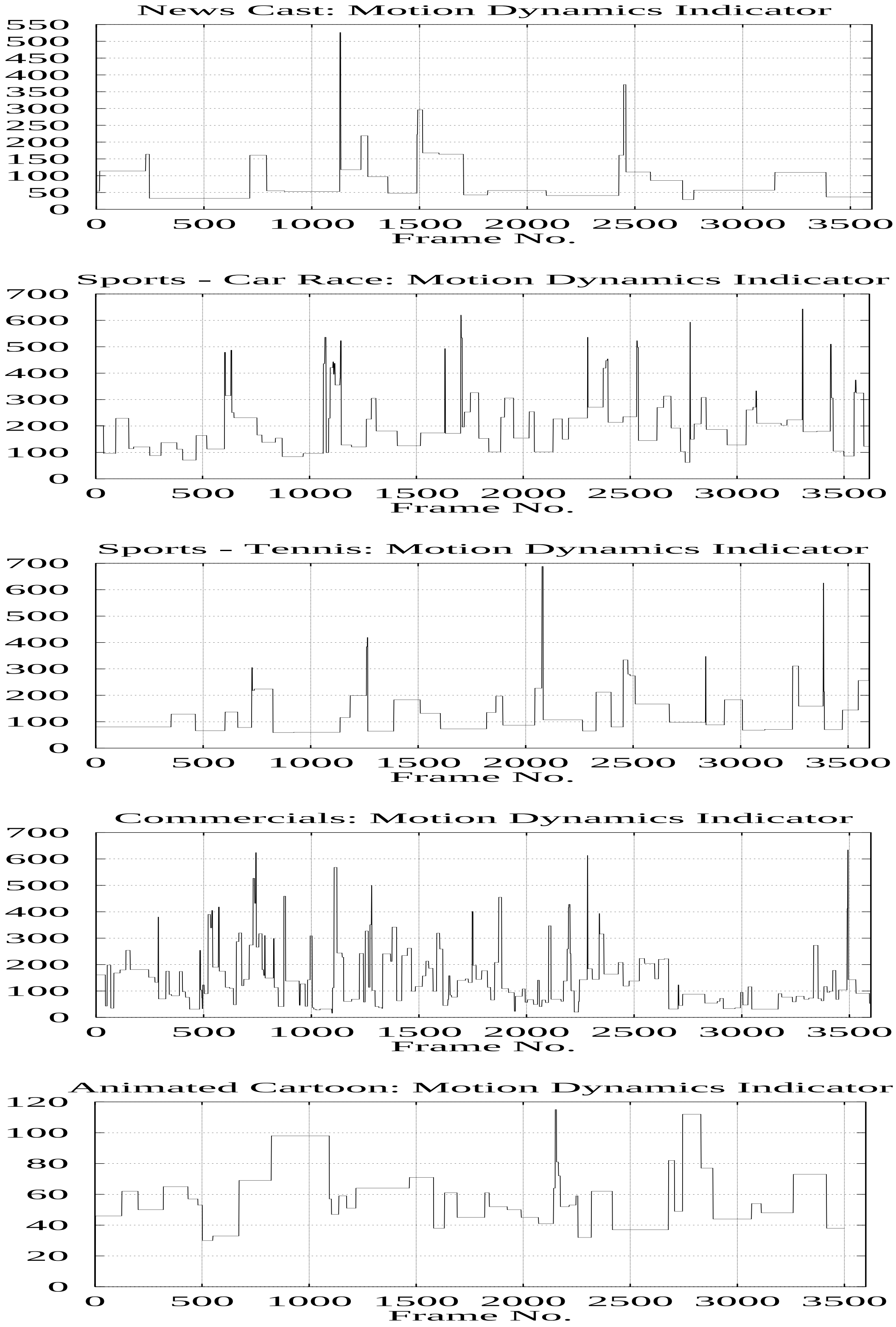

Figure 6: Motion energy

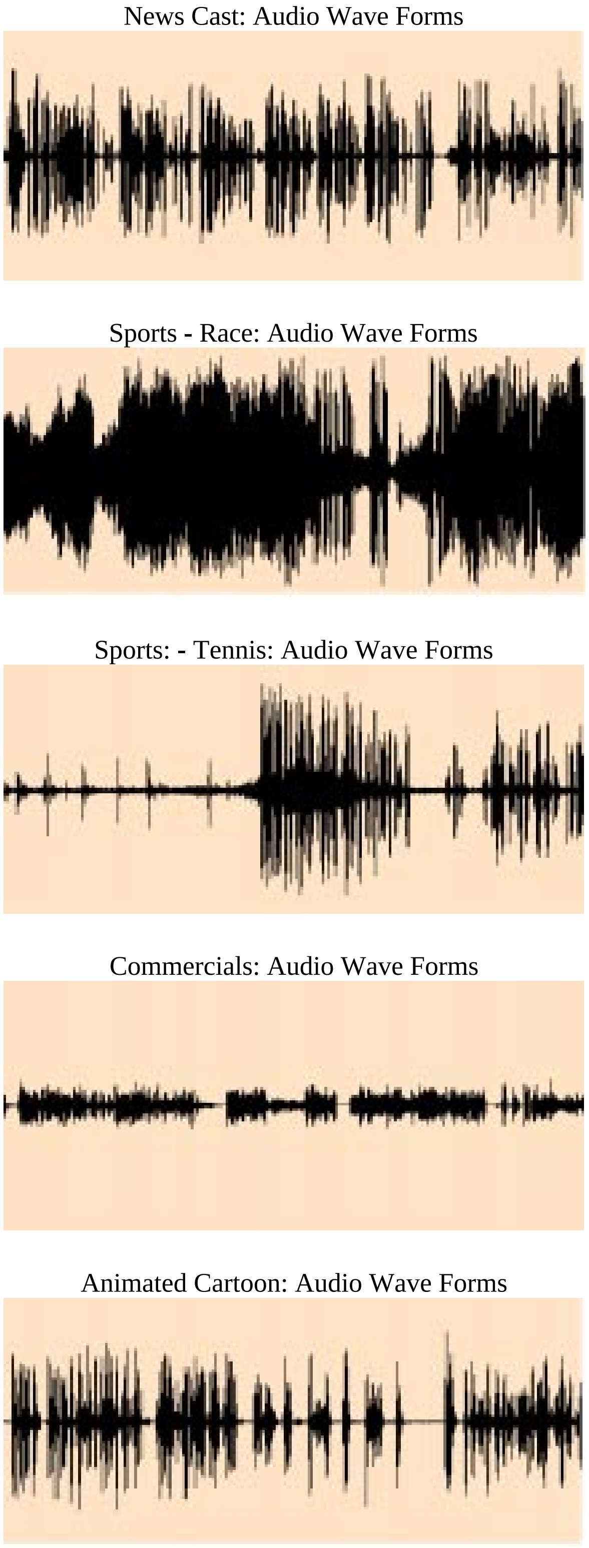

Figure 7: Audio wave forms

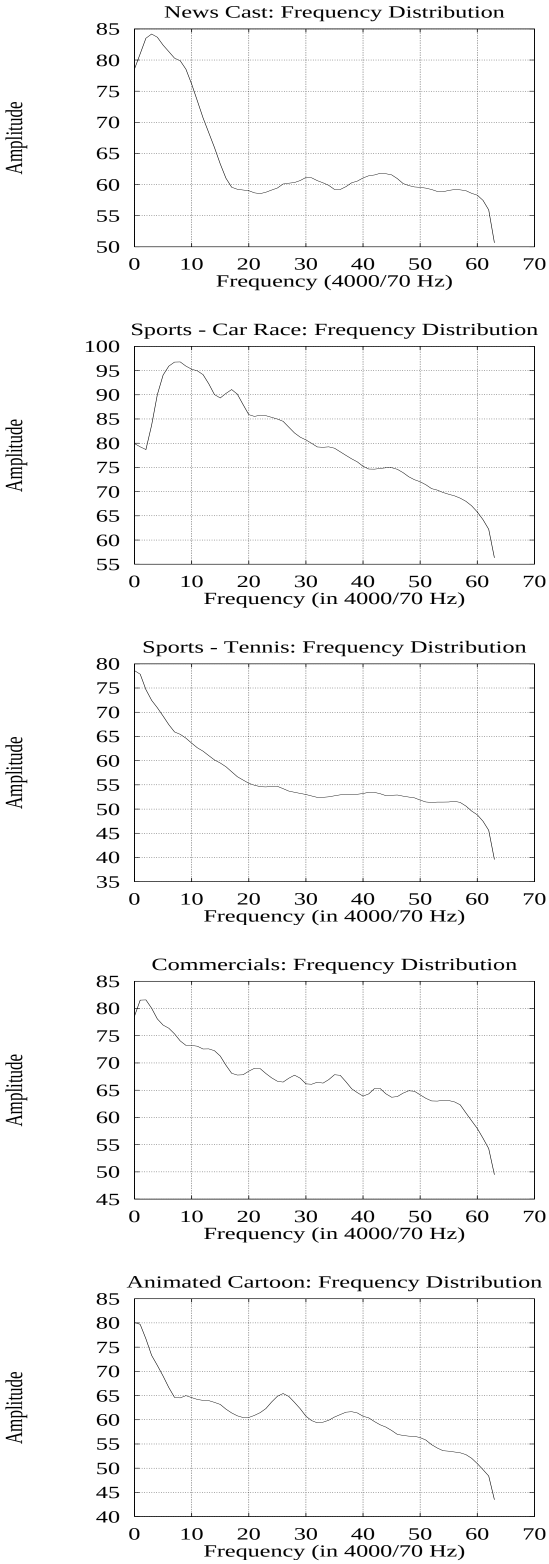

Figure 8: Audio frequency spectrum