Image Processing Techniques for Video Content Extraction

Inês Oliveira, Nuno Correia, Nuno Guimarães

INESC/IST, R. Alves Redol, 9, 6o, 1000 Lisboa
email: {Ines.Oliveira,Nuno.Correia,Nuno.Guimaraes}@inesc.pt

Abstract The main motivation for extracting the content of information is the accessibility problem. A problem that is even more relevant for dynamic multimedia data, which also have to be searched and retrieved. While content extraction techniques are reasonably developed for text, *video data* still is essentially opaque. Its richness and complexity suggests that there is a long way to go in extracting video features, and the implementation of more suitable and effective processing procedures is an important goal to be achieved.

This paper describes some of the basic image processing techniques offered by *videoCEL*, a toolkit for video content extraction, which makes available several commonly used abstractions and can be used by distinct applications.

Keywords

Content analysis, Video content extraction, Image processing, Temporal segmentation, Scene segmentation.

1. Introduction

The increase in the diversity and availability of electronic information led to additional processing requirements, in order to retrieve relevant and useful data: *the accessibility problem.* This problem is even more relevant for audiovisual information, where huge amounts of data have to be searched, indexed and processed. Most of the solutions for this type of problems point towards a common need: to extract relevant information features for a given content domain. A process which underlies two difficult tasks: deciding what is relevant and extracting it.

In fact, while content extraction techniques are reasonably developed for text, *video data* still is essentially opaque. Despite its obvious advantages as a communication medium, the lack of suitable processing and communication supporting platforms has delayed its introduction in a generalized way. This situation is changing and new video based applications are being developed. In our research group, we are currently developing tools for indexing video archives for later reuse, a system for content analysis of TV news [1], and hypervideo systems where hyperlinks are established based on content identification in different video streams. These applications greatly rely on efficient computational support, combining powerful image analysis and processing tools.

The developed toolkit prototype offers, in its processing components, all the functionality of these algorithms, hiding the implementation details and providing an uniform access methods to the different signal processing algorithms. The advantages offered by the use of libraries of specialised components have been largely debated [1, 4]: normalization, reutilization, flexibility, data abstraction and encapsulation, etc. The produced prototype results from the application of these principles to video content extraction, making available several abstractions commonly used by the related applications: a set of tools which extract relevant features of video data and can be reused by different applications. Next sections present a description of some of these tools and algorithms.

2. Toolkit overview

videoCEL is basically a library for video content extraction. Its components extract relevant features of video data and can be reused by different applications. The object model includes components for video data modelling and tools for processing and extracting video content, but currently the video processing is restricted to images.

At the data modelling level, the more significant concepts are the following:

- *Images*, for representing the frame data, a numerical matrix whose values can be colors, color map entries, etc.;
- *ColorMaps*, which map entries into a color space, allowing an additional indexation level;
- *ImageDisplayConvertes* and *ImageIOHandlers*, that convert images in the specific formats of the platforms and vice-versa.

Each of these concepts is represented by a (C++) class and integrated in a systematic hierarchy.

Tools for data processing are applied to the described data modelling classes, and also modelled as a hierarchy of classes: the *ImageOPs*, These operators represent functions which are applied to image regions and extract "single-image" or sequential content features. The implemented algorithms and procedures are described in more detaile in the next sections.

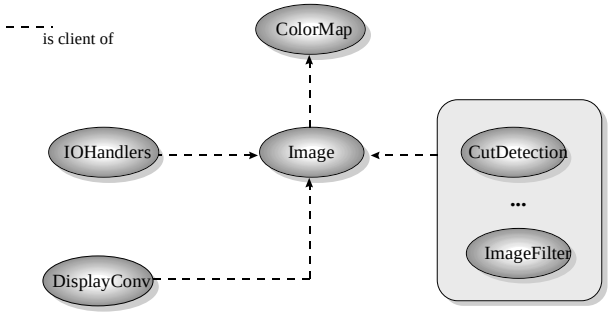

Figure 1: Object model overview.

The object model of *videoCEL* is a subset of a more complete model, which also includes concepts such has shots, shot sequences and views [1, 11]. Concepts, which are modelled in a distinct toolkit that provides functionalities for indexing, browsing and playing annotated video segments.

A shot object is a discrete sequence of images with a set of temporal attributes such as frame rate and duration and represents a video segment. A shot sequence object groups several shots using some semantic criteria. Views, are used to visualize and browse shots and shot sequences.

3. Temporal segmentation tools

One of the most important tasks for video analysis is to specify a unit set, in which the video temporal sequence may be organized [7]. The different video transitions are important for video content identification and for the definition of the semantics of the video language [8], making their detection one of the primary goals to be achieved. The basic assumption of the transition detection procedures is that the video segments are spatially and temporally continuos, and thus the boundary images must suffer significant content changes. Changes, which depend on the transition type and can be measured. The original problem is reduced to the search of suitable difference quantification metrics, whose maximums identify, with great probability, the transition temporal locations.

3.1 Cut detection

The process of detecting cuts is quite simple, mainly because the changes in content are very visible and they always occur instantaneously between consecutive frames. The implemented algorithm simply uses one of the quantification metrics, and a cut is declared when the differences are above a certain threshold. Thus, its success is greatly dependent on the metric suitability.

The results obtained by applying this procedure to some of our metrics are presented next. The thresholds selection was made empirically, while trying to maximize the success of the detection (minimizing simultaneously the false and missed detections). The captured video segment belongs to an outdoors news report, so its transitions are not very "artistic" (mainly cuts).

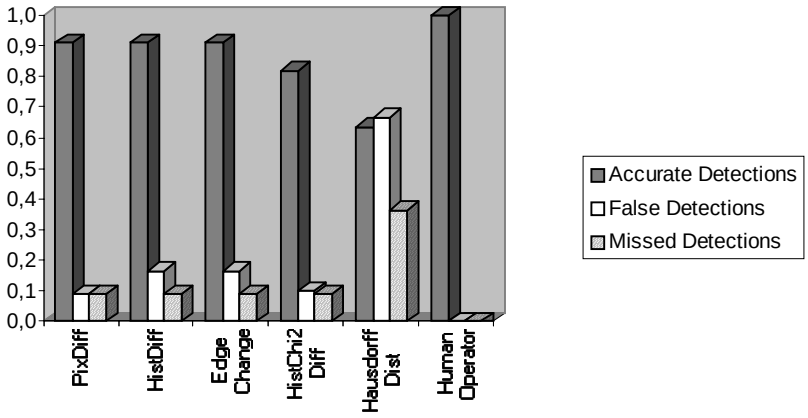

Figure 2: Cut detection results. See that almost all metrics generate a 90% accurate detection.

There are several well known strategies that usually improve this detection. For instance, the use of adaptive thresholds increases the flexibility of the thresholding, allowing the adaptation of the algorithm to diverse video content [6]. An approach that was used with some success in previous work [11], while trying to reduce some of the lacks of the metrics specific behavior, was simply to produce a weighted average of the differences obtained with two or more metrics. Pre-processing images using noise filters or lower resolution operators are also quite usual tasks, offering means for reducing image the noise and also the processing complexity. The distinctive treatment of image regions, in order to eliminate some of the more extreme values, remarkably increases the detection accuracy, specially when there are only a few objects moving on the captured scene [7].

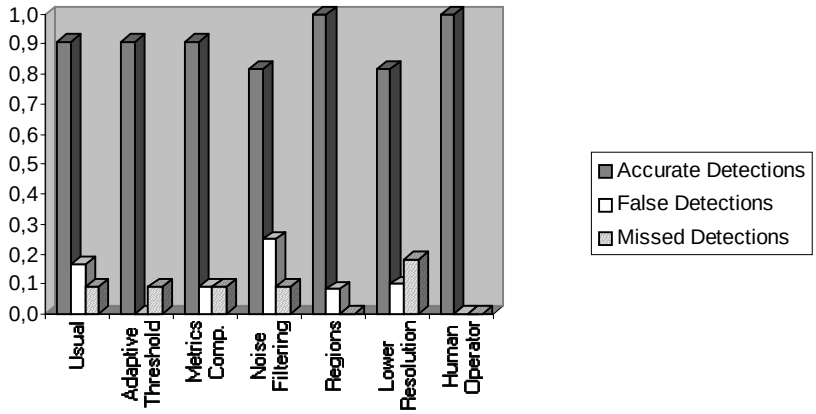

Figure 3: Cut detection results with improvements (using the *HistDiff* metric). The accuracy of the detection is clearly increased using these strategies, except in the case of noise filtering and lower resolution. One can actually explain this by defending that the images were quite clean so they were blurred with noise filtering procedure, while the use lower resolution images is essentially an approach for reducing the computation complexity.

3.2 Gradual transition detection

Gradual transitions, such as *fades*, *dissolves* and *wipes*, cause more gradual changes which evolve during several images. Although the obtained differences are less distinct from the average values, and can have similar values to the ones caused by camera operations, there are several successful procedures, which were adapted and are currently supported by the toolkit.

Twin-Comparison algorithm This algorithm [7] was developed after verifying that, in spite of the fact that the first and last transition frames are quite different, consecutive images remain very similar. Thus, as in the cuts detection, this procedure uses one of the difference metrics, but, instead of one, it has two thresholds: one higher for cuts, and another for the gradual transitions. While this algorithm just detects gradual transitions

and distinguish them from cuts, there are other approaches which also classify *fades*, *dissolves* and *wipes*, such as the Edge-Comparison presented next.

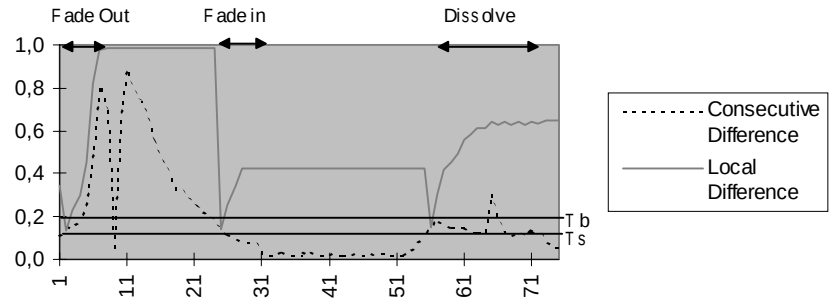

Figure 4: The *Twin-Comparison* algorithm results. When the consecutive difference is between *Tb* and *Ts*, a potential start is declared. When this happens, the local difference (the difference between the first frame of the potential segment and the current frame) starts to be computed. If consecutive frames are similar "enough" while the local difference is high, a gradual transition is declared.

Edge-Comparison algorithm This algorithm [6] analyses both edge change fractions, exiting and entering. Distinct gradual transitions generate characteristic variations of these values. For instance, a fade in always generates an increase in the entering edge fraction; conversely, a fade out causes an increase in the exiting edge fraction; a dissolve has the same effect as a fade out followed by a fade in.

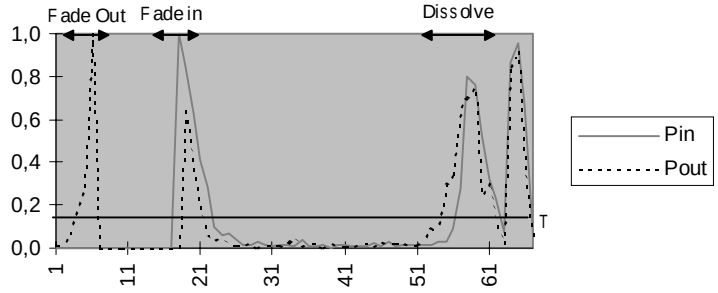

Figure 5: The Edge-Comparison algorithm results. Note that (1) in the *fade in* Pin>>Pout; (2) in the *fade out* Pout>>Pin, and (3) in the first half of the *dissolve* Pout>>Pin, and in the second half, Pin>>Pout.

4. Camera operation detection

As distinct transitions give different meanings to adjacent video segments, the possible camera operations are also relevant for content identification [8]. For example, that information can be used to build salient stills [7] and select key frames or segments for video representation. All the methods which detect and classify camera operations start from the following observation: each one generates global characteristic changes in the captured objects and background [5]. For example, when a pan happens they move horizontally in the opposite direction of the camera motion; the behavior of the tilts is similar but in the vertical axis; zooms generate convergent or divergent moves.

X-ray based method This approach [12] basically produces fingerprints of the global motion flow. After extracting the edges, each image is reduced to its horizontal and vertical projections, a column and a row, that roughly represent the horizontal and vertical global motions, which are usually referred to as the *x-ray images*.

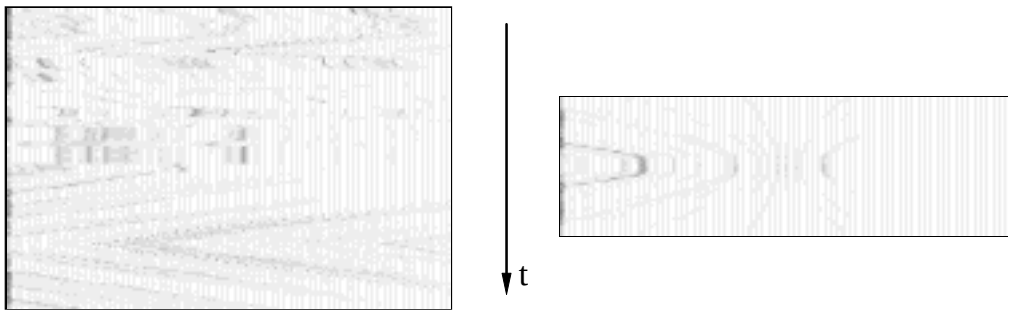

Figure 6: Horizontal x-ray images. On the left image one can see some panning operations; the right x-ray displays two zooming operations. Observing both projections it is easily perceived that (1) when the camera is still, the x-ray lines are parallel; (2) when the camera is panning or tilting, the corresponding x-ray lines slant to the opposite direction; and (3) when the lines diverge or converge, the camera is zooming.

As the above figure indicates, the behavior of the equal edge density lines, formed by the x-rays along the sequence, is characteristic of the main camera operations, giving enough information for supporting their detection. The implemented procedure basically generates the best matching percentages for each of the expected camera operations, which are then thresholded. Some of these results can be observed in the following figure, which shows all the matching percentages computed for a pan left segment.

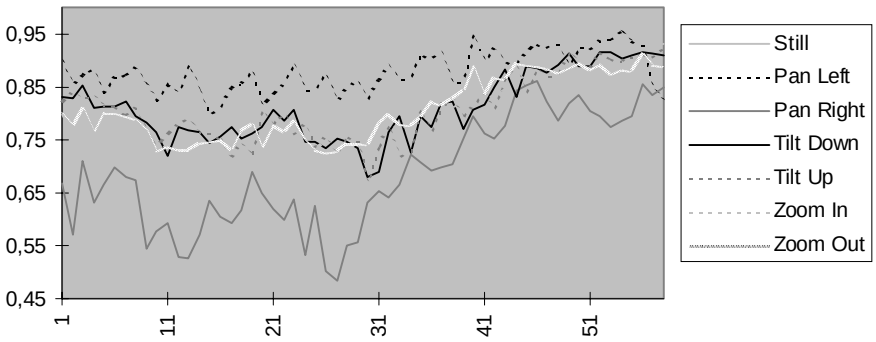

Figure 7: Pan Left results. Note that the pan left matching curve is clearly higher than the corresponding pan right results; the vertical and scaling results are also very close to each other.

As has been reported in several papers, we also intend to experiment some affine functions [6], which allow the determination of the occurred transformation between images. Although tome tests have been performed using the hausdorff distance, computed with a new multi-resolution algorithm [2], the obtained results still need further improvements.

5. Lighting conditions characterization

Light effects are usually mentioned in the cinema language grammar, as they contribution is essential for the overall video content meaning. The lighting conditions can be easily extracted by observing the distribution of the light intensity histogram: its mode, mean and average are valuable in characterising its distribution type and spread. These features also allow the quantification of the lighting variations, once the similarity of the images is determined.

Figure 8 presents some measures performed on an indoors scene, while varying its light conditions. As one will notice, the combination of these three basic measures let us easily perceive the light variations and roughly characterize the different lighting environments.

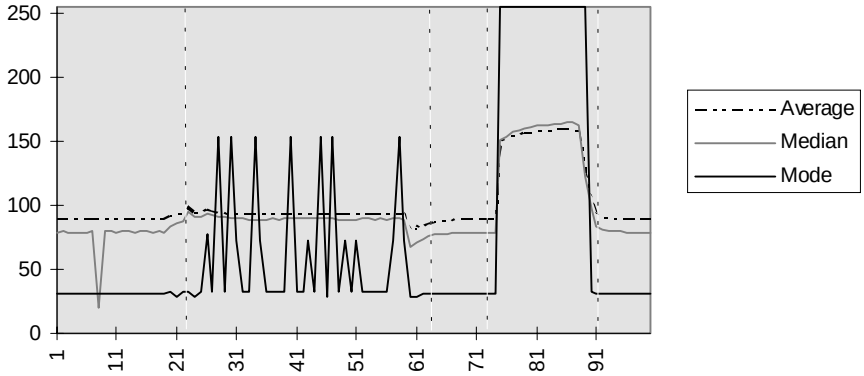

Figure 8: Luminance Statistical Measures. The first, third and fifth segments were captured in a natural light environment; the second video portion was obtain after turning on the room lights, which are fluorescent, and the fourth condition was simulated using the camera black light.

The luminance variations detection is in fact a powerful procedure, which requires further attention. There has been some trouble in distinguishing it from the changes generated by transitions. The real difficult still remains: detecting similarity when the light conditions severely change.

6. Scene segmentation

Scene segmentation refers to the image decomposition in its main components: objects, background, captions, etc. It is a first step for the identification and classification of the scene main features, and its tracking during all the sequence. The simplest implemented segmentation method is the amplitude thresholding, which is quite successful when the different regions have distinct amplitudes. It is particularly useful procedure for binarizing captions. Other methods are described below.

Region-based segmentation Region-based segmentation procedures find out various regions in an image which have similar features. One of such algorithms is the split and merge algorithm [3], that first divides the image in atomic homogeneous regions, and then merges the similar adjacent regions until they are sufficiently different. Two distinct metrics are needed: one for measuring the initial regions homogeneity (the variance, or any other difference measure), and another for quantifying the adjacent regions similarity (the average, median, mode, etc.).

Motion-based segmentation The main idea in motion-based segmentation techniques is to identify image regions with similar motion behaviors. These properties are determined by analysing the temporal evolution of the pixels. This process is carried out in the frequency image produced for all the image sequence. When more constant pixels are selected, for example, the final image is the background causing the motion removal. Once the background is extracted, the same principle can be used to extract and track motion or objects.

Figure 9: Background extraction. These images were artificially built, after determining, for each location, the sequential average, median and mode pixels values, which are shown by this order. The video sample used has about 100 frames and belongs to the initial sequence of an instructional video.

Figure 10: Object Extraction. These frames were obtained by subtracting the computed background to some image, arbitrary chosen in the sequence, that was then thresholded. The moving objects were completely extracted, specially with the median background.

Scene and object detection The process of detecting scenes or scene regions (objects) is, in certain way, the opposite process of transition detection: we want to find images regions whose differences are below a certain threshold. As a consequence this procedure uses difference quantification metrics. These functions can be determined for all the image, or a hierarchical growing resolution calculation can be performed to accelerate the process. Another tested algorithm, also hierarchical, is based in the hausdorff distance. It retrieves all the possible transformations (translation, rotation, etc.) between the edges of two images [2]. Another way of extracting objects is by representing their contours. The toolkit uses a *polygonal line approach* [3] to represent contours as a set of connected segments. The ending of a segment is detected when the relation between the current segment polygonal area and its length is beyond a certain threshold.

Caption extraction Based on an existing caption extraction method [10] a new and more effective procedure was implemented. As the captions are usually artificially added to images, the first step of this procedure is extracting high-contrast regions. This task is performed by segmenting the edge image, whose contours have been previously dilated by a certain radius. These regions are then subjected to a certain caption-characteristic size constrains, based on the x-rays (projections of edge images) properties; just the horizontal clusters remain. The resulting image is segmented and two different images are produced: one with black background for lighter text, and another with white background for darker text. The process is complete after binarizing both images and proceeding to more dimensional region constrains.

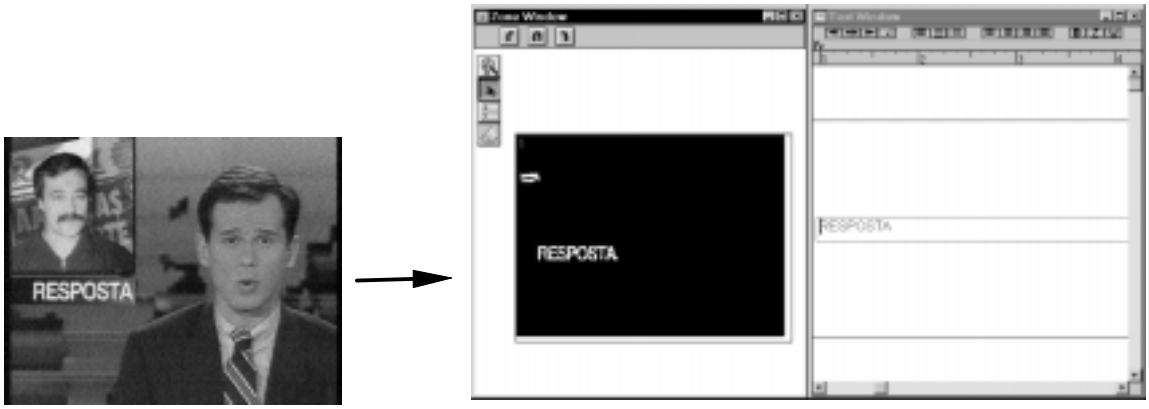

Figure 11: Caption Extraction. The right image is the result obtained after applying a commercial OCR to the frame processed by the toolkit, which is a binary image that just contains the potential caption regions.

7. Image difference quantification metrics

The accuracy of a metric is closely related to its sensitivity to changes occurred due to transitions. There are always alien factors, such as the object and camera movements, scene light changes, noise, etc., which also generates differences and may cause false detections. The metrics must be robust in these situations. The following functions were developed and tested, each one measuring the changes occurred in different features of image content:

- **Pixel differences counting** [7]**:** Counts the number of spatially correspondent pixels with different intensities, based on the principle that the transitions cause great spatial changes. It is very sensitive to

global motions and the differences introduced by the transitions are not very distinct from the average values.

- **Histogram differences sum** [7, 9]**:** Sums the differences between the histograms of both images, assuming that, unless a transition occurs, objects and background show very little color changes. These differences can be determined in several ways: χ^2, L_1, L_2, etc., with the known mathematical advantages. The pixels spatial distribution is ignored by these global measures, making them very insensitive to motion.
- **Hausdorff distance** [2]**:** Measures the maximum mismatch between two edge point sets. The edges give a preview of the image content, and are obviously affected by the transitions. This function requires high computational power and is very sensitive to noise.
- **Edge Change Rate** [6]**:** Determines the maximum of the exiting and entering edge point fractions. It is assumed that when a transition occurs, new edges appear far from the older edges, and old edges disappear far from the newer edges.

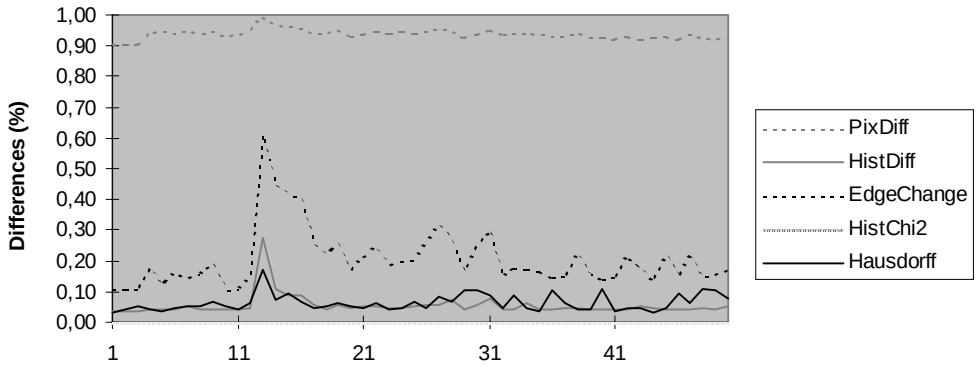

Figure 12: Differences Metrics Results. Note that all metrics have a maximum near frame 13, which clearly indicates an accentuated content change, a cut.

8. Edge detection

Two distinct procedures for edge detection [3] were implemented: (1) *gradient module thresholding*, where the image vectors are obtained using the Sobel operator; (2) the *canny filter*, considered the optimum detector, which analyses the representativity of gradient module maximums, and thus producing thinner contours. As the differential operators amplify high frequency zones, it is common practice to pre-process the images using noise filters, a functionality also supported by the toolkit in the form of several smoothing operators: the median filter, the average filter, and a gaussian filter.

9. Applications

In this section we outline the main characteristics of some applications built with the components and techniques offered in *videoCEL*.

Video browser This application [11] is used to visualise video streams. The browser can load a stream and split it in its shot segments using cut detection algorithms. Each shot is then represented in the browser main window by an icon, that is a reduced form of its first frame. The shots can be played using several view objects.

WeatherDigest The WeatherDigest application [13] generates HTML documents from TV weather forecasts. The temporal sequence of maps, presented on the TV, is mapped to a sequence of images in the HTML page. This application illustrates the importance of information models.

News analysis We developed a set of applications [1] to be used by social scientists in content analysis of TV news. The analysis was centred in filling forms including news items duration, subjects, etc., which our system attempts to automate. The system generates HTML pages with the images and CSV (Comma Separated

Values) tables suitable for use in spreadsheets such as Excel. Additionally, these HTML pages can be also used for news browsing, and there also is a Java based tool for accessing this information.

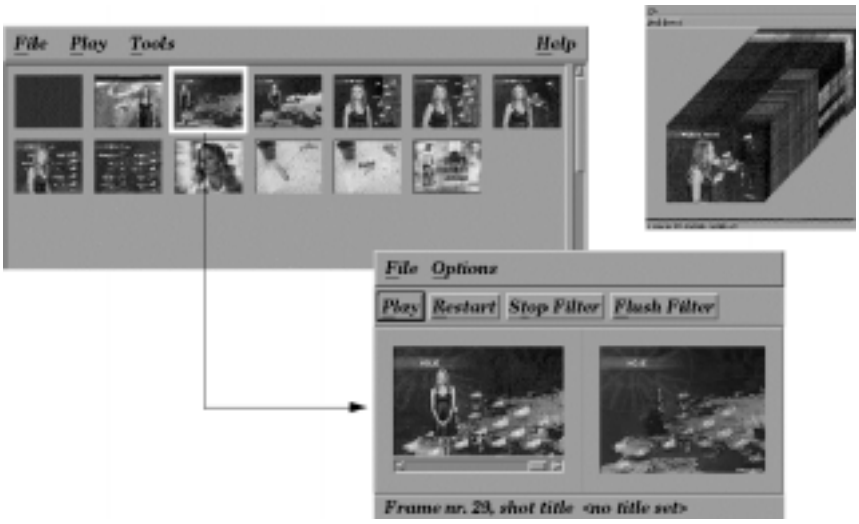

Figure 13: Video browser. The main window, and the cubic and movement filter views.

10. Conclusions and future work

The toolkit approach is a good solution when one is interested in building suitable support for extracting information content, specially because it can be reused and easily extended. While there are several efficient and normalized systems for extracting content from text and images, video related systems still remain very domain-dependent.

In this context, the components of *videoCEL* include a wide range of image processing techniques, that support the extraction of several video content features. Some of these procedures were developed specifically for video, in related works, with reported successful results. But we also have implemented several basic, but useful, image processing routines. These operations are part of the image content extraction know-how, or were simply implement to support some of the more complex operations, or the extraction of video features also considered relevant in social sciences or content analysis literature.

As future extensions, new tools will soon be added to *videoCEL* to extract additional content features. In fact, we are specially interested in including audio processing. Audio streams contain extremely valuable data, whose content is also very rich and diverse. The combination of audio content extraction tools, with image techniques, will definitely generate interesting results, and very likely improve the quality of the present analysis.

11. References

[1] Nuno Guimarães, Nuno Correia, Inês Oliveira, João Martins. “Designing Computer Support for Content Analysis: a situated use of video parsing and analysis techniques”. *Multimedia Tools and Applications Journal* (to be published during 1997).

[2] Daniel P. Huttenlocher, William J. Rucklidge. "A Multi-Resolution Technique for Comparing Images Using the Hausdorff Distance". Departement of computer Science, Cornell University, 1994.

[3] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[4] J. J. Putress, N. M. Guimarães. ”The Toolkit Approach to Hypermedia”. *Echt’90*, 1990.

[5] Y. Tonomura, A. Akutsu, Y. Taniguchi, G. Suzuki. "Structured Video Computing". NTT Human Interface Laboratories, *IEEE MultiMedia*, 1994.

[6] Ramin Zabih, Justin Miller, Kevin Mai. "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks", Cornell University, 1995. (*ftp://www.cs.cornell.edu/ home/rdz/mm95.ps.gz*)

[7] Hongjiang Zhang. “Video Content Analysis and Retrieval”. *Handbook on Pattern Recognition and Computer Vision*, World Scientific Publishing Company, 1997.

[8] Arthur Asa Berger. *Media Analysis Techniques*. Sage Publications, 1991.

[9] Arun Hampapur, Ramesh Jain, Terry Weymouth. “Production Model Based Digital Video Segmentation”. *Multimedia Tools and Applications*, vol. 1, 9-46, 1995.

[10] Rainer Lienhart, Frank Stuber. “Automatic text Recognition in Digital Videos”. University of Manhein, Departement of Computer Science, *technical report TR-95-006*, 1995. (*http://www.informatik.uni.manhein.de/~lienhart/ papers/tr-95-006.gz*)

[11] Inês Oliveira, João Pedro Martins. “TV Multimedia Processing”. *Final Project Report, IST*, 1995.

[12]Y. Tonomura, A. Akutsu, K. Otsuji, T. Sadakata. “VideoMap and VideoSpaceIcon: Tools for Anatomizing Video Content”. Proceedings of *INTERCHI’93*, 1993

[13]N. Correia, I. Oliveira, J. Martins, N. Guimarães. “WeatherDigest: an experimental on media conversion”. *Integration Issues in Large Commercial Media Delivery Systems, SPIE-Photonics East’ 95, Philadelphia, USA*, vol. SPIE 2615, 50-61, 1995.