Some tools for indexing video documents

Claire-Hélène Demarty and Serge Beucher
Centre de Morphologie Mathématique - Ecole des Mines de Paris
35, rue Saint-Honoré, 77305 Fontainebleau cedex, FRANCE
tel: 33 1 64 69 48 04 / 33 1 64 69 47 97
fax: 33 1 64 69 47 07
email: demarty@cmm.ensmp.fr / beucher@cmm.ensmp.fr

15 October 1998

Abstract

This paper proposes several tools for video indexing, all of them based on a unique local algorithm. The algorithm leads to a local comparison of two successive frames of a sequence in order to detect if changes occur. This step is then followed by a morphological filtering of the temporal evolution of the criterion. Besides its characteristic of computing a given criterion locally on color images which allows one to keep spatial information on the change, it has the properties of being fast and very simple and can be re-used for several different indexing tools. As a direct application of this algorithm, we first describe a tool for video partitioning, for which the transition detection rate reaches a high level in the case of cuts with a very small number of false detections. Furthermore, this algorithm gives access to the geometrical model of the transition. We also develop tools for inner change detection and shot relation extraction. Finally, these tools are used in a more direct application of newscaster shot detection from a TV newscast.

Keywords: content-based indexing, video analysis, image processing

1 Introduction

Due to the amount of video documents produced daily, the need for an indexing tool is of prime importance. This need appears clearly in the literature, where the number of proposed techniques is rapidly growing. Most of the techniques address the problem of indexing still images, but more and more are dedicated to video documents. For the latter type of sources, it is commonly accepted that the first step in the indexing procedure should consist of a temporal segmentation of the document into shots. Irrespective of whether this splitting is achieved directly on color [13, 10] or on grey level sequences, the splitting techniques are usually divided into two groups based on whether they are applicable to compressed or uncompressed image sequences. The first group bases its detection on the study of DCT coefficients and/or motion vectors [13, 1, 4]. The latter group can be divided into four main categories: histogram-based, motion-based, model-based and intensity or color difference techniques. Techniques based on histogram comparison

(histogram intersection or χ^2 [10]) offer good robustness to camera motions, but suffer from high computing and storage costs. Furthermore, their major drawback is the loss of spatial information contained in the image. For motion-based techniques, a robust estimation of the dominant image motion is made, as a transition actually corresponds to a temporal breaking of this motion [2]. As motion estimation is usually applied later during the indexing step, the choice of this method saves on computing time but suffers from all the problems associated with motion estimation. A third group of model-based techniques [7, 6] bases its detection on mathematical models of the transitions. For these methods, transitions are no more than particular shots, constructed according to specific equations called "spatial" for wipes, page turns, etc., and "chromatic" for those which only act on luminance (fades, dissolves). This last group has a major advantage of dealing with transitions other than cuts and trying to identify them. In the fourth group of methods, the color or intensity difference is computed point to point on two successive frames. Computing the sum of the differences on the whole image leads to a global criterion and results in a very simple method, but is sensitive to noise, object and camera motions and, once again, loses the spatial information of the transition. To reduce these drawbacks, an alternative is to compute the chosen criterion locally on small image blocks. In [10], each frame is first split into 16 rectangles before computing the χ^2 on each small rectangle. The 8 more representative values of the criterion are then kept for each frame.

Although its result is essential, the temporal segmentation of the document remains only a first step in the indexing process and therefore should be fast and simple. In order to satisfy these two conditions, we propose an algorithm of video partitioning which also works locally on images and is therefore similar to [10]. After a description of the algorithm and of its differences and improvements compared to [10] in section 2, its results when detecting transitions are discussed in section 3. Whilst extracting transitions this algorithm also presents the advantage of giving access to other syntactic information on the video document. Examples are given in section 4: firstly, one is able to detect inner changes in shots with the use of the local algorithm. Secondly, relations, this time between shots, are extracted. A more practical application in which all the shots containing the newscaster are selected from a TV newscast is then shown. All these tools (including the transition detection) give hints on the hierarchical organization of the document.

2 Description of the algorithm

Before proceeding further with the description, we emphasize that this local algorithm takes non-compressed, color video documents, acquired at a frequency of 5 Hz, as input. This frequency is a good compromise between the amount of storage required and a sufficient sampling rate so as not to lose any shots.

2.1 A local criterion

The drawback of using a global criterion on images, as mentioned in the introduction, is that this leads to the loss of spatial information, *i.e:* where exactly the transitions appear in the images. In the case of simple cuts, two shots are simply concatenated, the transition

is abrupt and affects the whole image. But a wide variety of other transitions (such as wipes, page turns, etc.) act locally on images, according to a specific geometric model. For these particular transitions, the loss of spatial information may be damaging.

The proposed algorithm first splits images into small rectangles (typically 20×20 pixels, but this can be user-defined) and computes the criterion locally on each pair of corresponding rectangles in two successive images. At this step, the geometry of the transition can be addressed: the criterion gives a measure of the local transition on each rectangle. This will be further studied in section 2.3. As for the choice of criterion, a mean distance in the RGB space for all the pixels in two corresponding rectangles is computed, which leads only to a complexity of $O(n)$. Color histogram differences and χ^2 were set aside because of their computing cost, although they usually lead to sharper criterion curves [10]. This choice is balanced by a further filtering step (cf. 2.2).

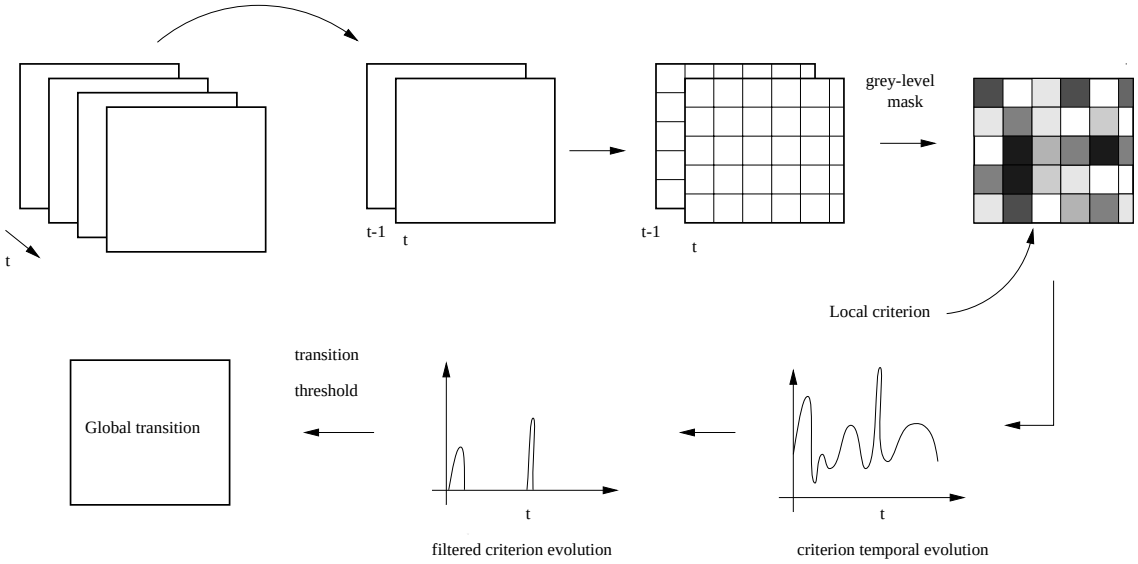


Figure 1: Description of the local algorithm.

Figure 1 summarizes the different steps of the algorithm: after the splitting of two successive images into rectangles and the computation of the criterion on each rectangle, a grey level mask of the transition is obtained, each grey level corresponding to a criterion value. We define our global criterion as the volume of this new image; a *transition* threshold is then applied directly to this criterion.

In addition to the rectangle sizes and the sampling frequency, the transition threshold is the third user-defined parameter of this algorithm. The parameters were set at 0.2 for the transition threshold, 20×20 pixels for the rectangle sizes and 5 Hz for the frequency, and these settings were used for all the test sequences. Other parameter values were also tested, but these happened to be the best choices. This leads to the conclusion that the algorithm is not really dependent on these three parameters.

As a criterion value is available for each pair of two successive images it is possible to

compute the criterion temporal evolution curve for a given sequence. An example is shown at the top left of figure 2. Each peak corresponds to a cut, but at this stage this rough curve contains too much noise or strong variations which prevents making a choice of a threshold for extracting the peaks. For this reason, an additional step of curve filtering is applied to the curve before the thresholding. The top right part of figure 2 shows another example of an even noisier evolution curve. The corresponding sequence is a kart race, in which all objects are moving fast and near the camera. Furthermore, the acquisition quality is poor and all shots are very similar to one another.

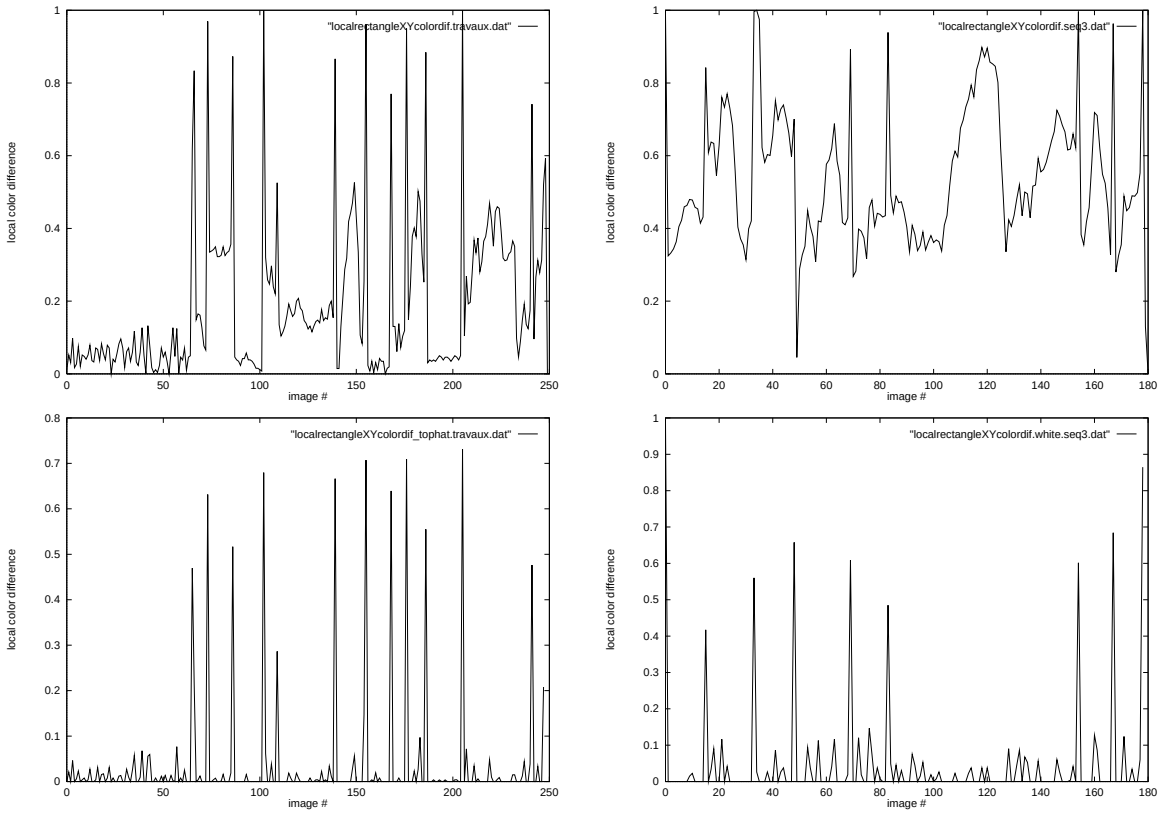


Figure 2: Examples of evolution curves of the global criterion before any filtering (top row), and filtered by a top-hat transform (bottom row). Left half: sequence *road-works*, 50 s, CIF format, frequency of 5 Hz. Right half: sequence *kart race*, 30 s, CIF format, frequency of 5 Hz.

2.2 Morphological filtering of the evolution curve

Mathematical morphology [11, 12] offers numerous efficient tools for filtering curves, images or sequences. This image processing technique is based on set theory and has already proved its efficiency in numerous image processing problems. One of the morphological

filters, called the *top-hat* transform, TH, is particulary well-designed for extracting small white details on images or peaks from one-dimensional curves. This filter has quite a simple definition. It consists of a comparison between a curve (or an image) and a structuring element, *i.e*: an object whose shape and size are chosen by the user. An example of a typical structuring element would be a line of three pixels for one-dimensional curves. The top-hat of an object X is then the subset of X, obtained by only keeping the part of X in which the structuring element cannot be included (*cf.* figure 3, part b). An alternative to the classical top-hat is the *top-hat inf*: instead of keeping the peaks in which the structuring element cannot be included exactly as on the original curve, we keep them with a height which corresponds to the difference between the maximal and the minimal values under the peak (*cf.* figure 3, part c). This leads to enhanced values of peaks.

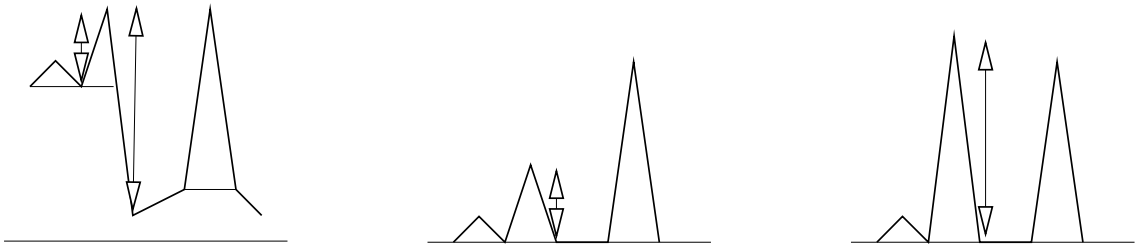


Figure 3: Top-hat and top-hat inf of an object X by a line of length 3.

In the case of the evolution curve, the structuring element is a small line of size 3. The result of the top-hat inf transform as applied to the two examples of the first row of figure 2 can be seen on the second row: only emphasized peaks remain, even for the difficult *kart race* sequence (see, especially for peak #3, the usefulness of the top-hat inf). The choice of the transition threshold is then rather simple and this is the reason why the same threshold can be applied for all of the test sequences.

2.3 Geometric model of the transition

The use of a local criterion allows one to keep track of the transition geometry. In the proposed algorithm, the spatial feature of each transition is directly accessible in what is called the *transition mask* in section 2.1. These masks can be grey level (each grey level corresponding to a certain criterion value) or binary ones. In the latter case, according to a second *local* threshold, it can be decided whether a particular rectangle should be white (a local cut took place), or black (no local cut took place).

The study of the temporal evolution of binary masks gives access to the geometry of the transition. An example of such an evolution is illustrated in figure 4, in the case of a wipe (a wipe consists of a line passing through the image, with the current shot on one side and the next shot on the other side of the line). Mask geometry characterizes this image transformation, which is even more easily identified when morphological filtering

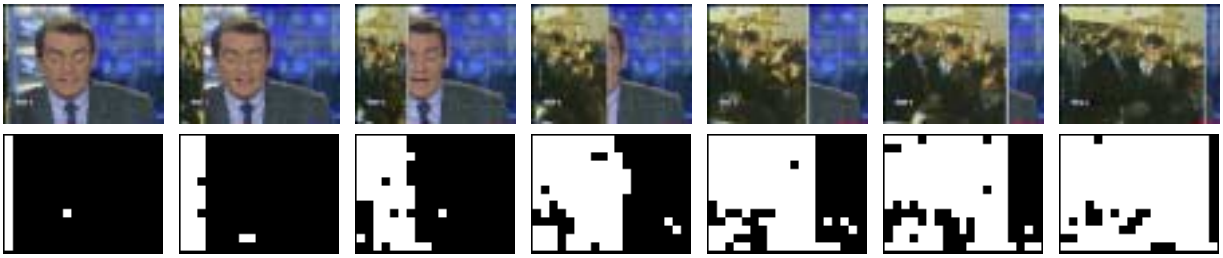


Figure 4: Temporal mask evolution in the case of a wipe.

(by opening and closing its dual operator) is applied to the mask. This suppresses erratic white rectangles and fills undesired holes as can be seen in figure 5, allowing a better recognition of the transition.

Figure 5: Examples of morphological filtering of transition masks in the case of cuts. Sequence *road-works*.

3 Results of transition detection

The algorithm was tested on 22 video documents containing cuts (274), fades and dissolves (23), wipes (3), page turns (1) and one other transition. The duration of these documents goes from 10 seconds to nearly 6 minutes. They were acquired at 5 Hz. Images are in CIF format. Mean detection rates reach 98.3% for cuts, 39.5% for fades, dissolves and the rest of the transitions, with a small rate of false detections at 1.5% on all the documents. One therefore sees that this algorithm is totally efficient for cuts but less efficient for the rest of the transitions, especially for fades and dissolves, for which the detection rate is quite low. No use was made of mask geometry in the case of spatial transitions (page turns, wipes), but this could increase the detection rate.

The false detection rate is particularly low for an algorithm without real parameters to be tuned before running on each sequence. It should be remembered that the transition

threshold was set at 0.2, the rectangle sizes at 20×20 and the frequency at 5Hz for all the sequences. Furthermore a slightly lower choice of threshold allows detection of most of the cuts lost, but with a slightly higher false detection rate. In section 4.2 it will be seen that the false detection problem can be addressed by the establishment of relations between shots. As for the fades and the dissolves, it should be stated that this algorithm does not handle this type of transition. Part of our current work addresses the problem of finding another similar criterion for fade detection. We do indeed intend to apply the same type of filtering hierarchically to evolution curves. When all transitions are considered, the detection rate is nonetheless of 90.4%, with 1.5% of false alarms. To conclude, the algorithm is fast and simple, as was expected, and works in real time (at the 5Hz frequency, on a Pentium 266 MHz).

4 Hierarchy of the shots

The detection of transitions in video documents is the first step in the elaboration of a *relation tree*. Before giving the definition of this notion, we would like to remind the reader that the shots are the first lower level of a temporal hierarchy of a video document. Several other syntactic objects, such as scenes and sequences, are used to construct this hierarchy. Shots (frames recorded contiguously, same unity of place, action and time) can be grouped into scenes (continuity of actions sharing the same unity of place and time [3]) which can in turn also be grouped into sequences (same unity of subject) [9, 8]. A practical example of such a hierarchy would be the shots, interviews and reports (scenes or sequences) of a TV newscast. In this context, a relation tree would represent all the temporal, semantic, etc., relations which exist between shots, scenes and sequences of the same video file. In this respect, the algorithm described in section 2 can be re-used in order to extract different levels of the hierarchy.

4.1 Detection of inner changes

The first example of syntactic information accessible by using the algorithm is in the detection of whether there were any changes in a particular shot. In order to answer this question, the selected keyframes of each shot are compared to one another using the proposed local algorithm.

Another threshold, the *inner change threshold*, is set at one and a half times the value of the transition threshold for all the documents, which helps in deciding if a change occurs. For the sake of simplicity, the first and last frames of a given shot are chosen as keyframes for this application. Nevertheless the change detection could be applied on other keyframes. An illustration of the changes for an interview document is available in table 6; the selected keyframes for the document are shown in figure 7.

Five shots were detected and only the first one was described as having changed. In the case of the first shot, the change detection is due to the text at the bottom of the first keyframe which disappears during the shot and to the small movement of the woman's head. Coupling this detection with the study of the mask also gives more information on the exact location of the change. This indicator is interesting in order to run specific

Shot #	0	1	2	3	4
Change	yes	no	no	no	no

Figure 6: Inner changes for the shots of the sequence *Interview*.

Figure 7: Keyframes of the sequence *Interview* before change detection.

algorithms on the changing region afterwards (like text extraction for example). When no change is detected in a shot all keyframes, bar one, can be suppressed for the shot as they represent redundant information. The simplified keyframe mosaic for the same document *interview*, is proposed in figure 8: four keyframes out of ten have been suppressed.

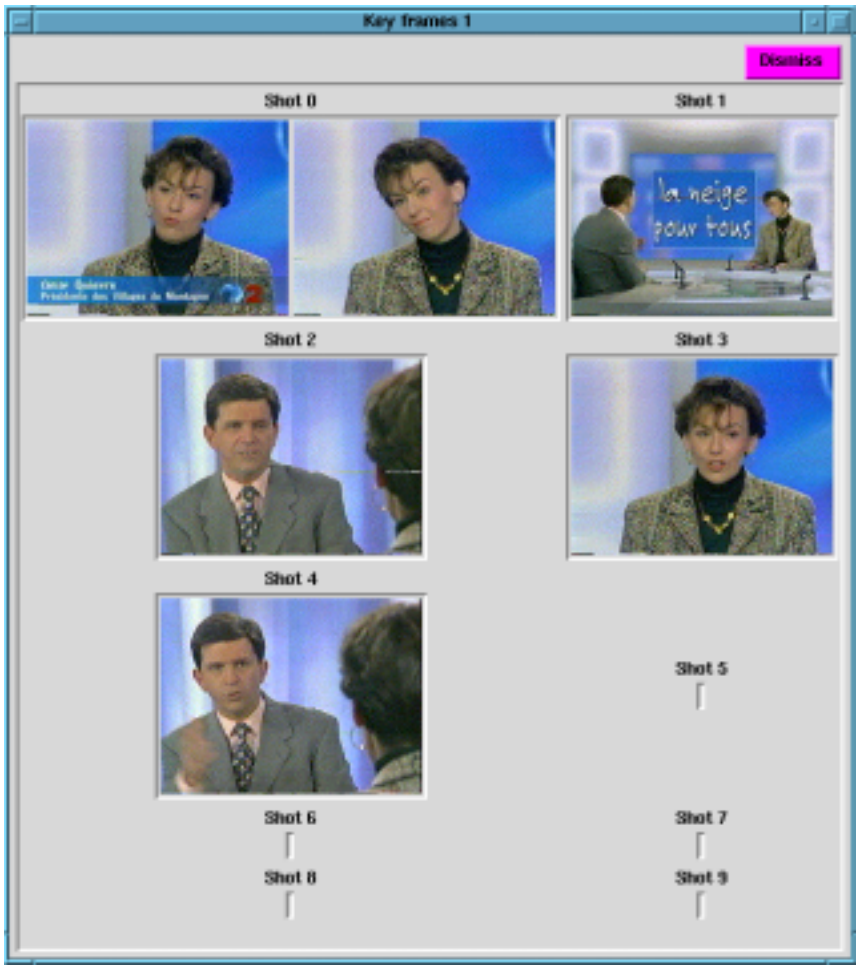


Figure 8: Keyframes of the sequence *Interview* after change detection.

As a side effect, this change detection allows confirmation of the first step of transition detection. When no change is detected, there is hardly any chance that a non-detected transition occurred in the current shot. On the contrary, a change detection could lead to more precise study of this part of the document, for example by motion estimation. In the case of fades and dissolves which are often not detected, this could be interesting. Once again, the use of the algorithm is both simple and fast as, for the case of two keyframes per shot, only one extra comparison is computed.

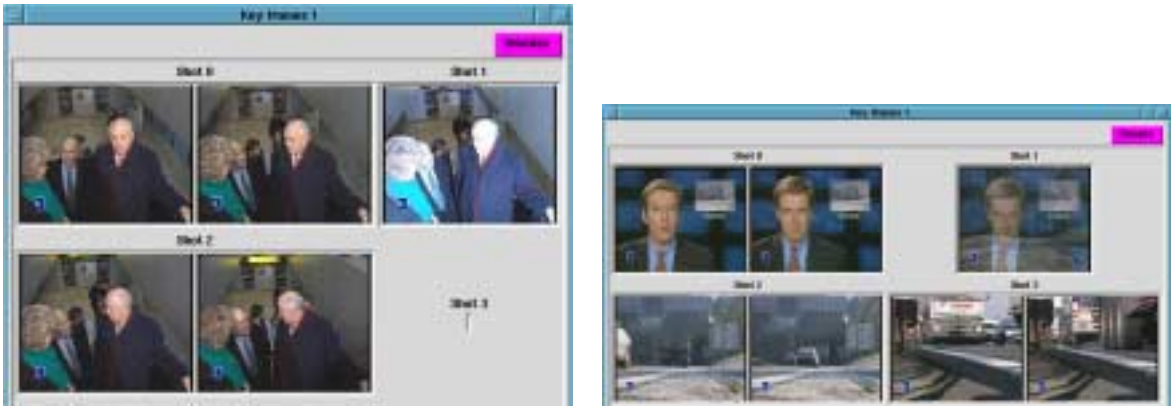
4.2 Detection of relations between shots

The logical step following the shot extraction and the inner change detection is the establishment of relations between shots. At a low level of syntactic information, this relation detecting tool aims to answer the question: are these two shots similar ? Here again, the local algorithm proves to be useful, and is applied in the same way as for the inner change detection. The difference lies in the fact that if keyframes are still compared, they are not keyframes of the same shot but keyframes coming from different shots. The same *inner change* threshold is applied, but this time it is called *relation* threshold. It still keeps the same value of one and a half times the value of the transition threshold for all the test documents. However, although during inner change two keyframes with a criterion value lower than the *inner change* threshold were considered not to have changed, we now find that two keyframes with a criterion value lower than the threshold are this time in relation. Figure 10 shows examples of detected relations for the sequence *interview*. Two relations were extracted between shot 0 and shot 3, and between shot 2 and shot 4. Apart from the results which permit construction of higher levels of the relation tree, we now have syntactic clues as to the organization of the document:

- When two groups of related shots are interlaced (as illustrated on figure 10), this is a strong indicator in favour of an interview sequence: the camera goes from the interviewer to the interviewed person.
- When a shot contains only one frame and is surrounded by two related shots, it is worth running an algorithm on this particular shot in order to determine if there was luminance flickering, which could correspond to a flash. This event is rather common when dealing with TV newscasts.
- When two successive shots are found to be related, it could mean that a false detection occurred. This can be used in the case of dissolves which, when detected, are often extracted as a whole shot, in relation to the previous shot.

Examples of these false alarms detected by way of the relations are shown in figure 9.

At this point we should note that there are two ways of building the relation groups, leading to two different notions: *strict* groups and *smooth* groups. In a strict relation group, each shot is related to every other shot in the group. In other words, two shots extracted from a strict relation group are related to one another. On the other hand, while in a smooth relation group, each shot is related to at least one other shot in the group, it will not necessary be related to all of them. Strict relation groups are usually smaller than smooth relation groups for a given sequence. These two different notions do not lead to the same classification of the shots in a sequence. In a smooth classification, each shot in the sequence appears only once in a unique group, whereas in a strict classification, a same shot can belong to several different strict groups. An example of the latter classification is given in section 4.3.



a) b)

Figure 9: Two cases of false detections which can be suppressed with the use of shot relations. a) Two false detections due to a flash (shot 1). b) One false detection due to a dissolve extracted as a whole shot (shot 1).

Figure 10: Strict groups of related shots for the sequence *Interview*.

4.3 Application to the newscaster detection

This last section proposes a final indexing tool. We emphasize that this tool was constructed to be simple: no use of high syntactic information is made.

When considering the particular document class of TV newscasts, some shots play a special role in the hierarchical organization of the rest of the document: the newscaster shots impose a structure in the newscast and separate the various topics of news. Being able to extract them from a TV newscast will enable one to go upwards in the relation tree. To achieve this goal, four criterion were elaborated and merged. Each of them leads to a probability of each shot being a newscaster shot. All of them are based on inherent properties of this kind of shot, whatever channel, land or time period they belong to:

- In these shots, there is at least one person in front of the camera of a certain size and at a certain location in the frame.
- The shots reappear regularly throughout the whole newscast.
- They are often related to a shot at the beginning and at the end of the newscast.
- In such shots, the background tends to be motionless.

The exact and precise description of these four criteria is not the topic of this section. We shall just give the features on which each of them is based. The first criterion measures the probability of having at least one connected component with the specific skin color [5], at a certain size (0.3% to 20% of the whole image) and more or less in the middle of the frame. The notion of "middle of the frame" and the sizes "0.3% to 20%" are chosen experimentally on newscaster images, and are tested according to fuzzy sets.

The second criterion is directly derived from the establishment of relation groups in section 4.2. In a TV newscast, the group containing the biggest number of related shots has a high probability of being the newscaster group. The regularity of the occurrences of such shots during the newscast is not taken into account at this point.

Relation groups are also used for the third criterion: the group containing shots at the beginning or at the end of the document have a higher probability than others. A small drawback of this assumption is that it is necessary to process the whole newscast.

The fourth criterion computes the probability of having a static background, still using the local algorithm however applied not on rectangles, but outside a frame situated in the middle of the image and determined experimentally. We are then assuming that a good estimation of the background motion is accessible outside this frame.

All these criteria are then merged in order to get a unique probability for each group of related shots, where the maximal probability is then supposed to correspond to the newscaster group. Table 12 gives the probabilities obtained for the 13 relation groups of a sequence, *jtv1*, simulating a TV newscast of 6 minutes. These groups are constructed using the definition of strict relation groups. As mentioned in section 4.2, a shot can appear in several different strict groups (see figure 11). In this array, two groups have maximal probabilities of respectively 86.7% and 87.7%: they are shown in figure 11 and both of them correspond to newscaster groups. The newscaster detection was only tested

on the sequence *jtv1* because storing an entire TV newscast on disk is problematic. Even for this sole sequence, the results are quite encouraging. Due to the way the algorithm is implemented, any extra criterion can easily be added and its probability merged to the result. In this respect, our current work deals with adding the shape of the skin connected component and also with the fact that dissolves often play a role in newscaster shot boundaries in most TV newcasts. We plan to add these two aspects to the tool for newscaster detection.

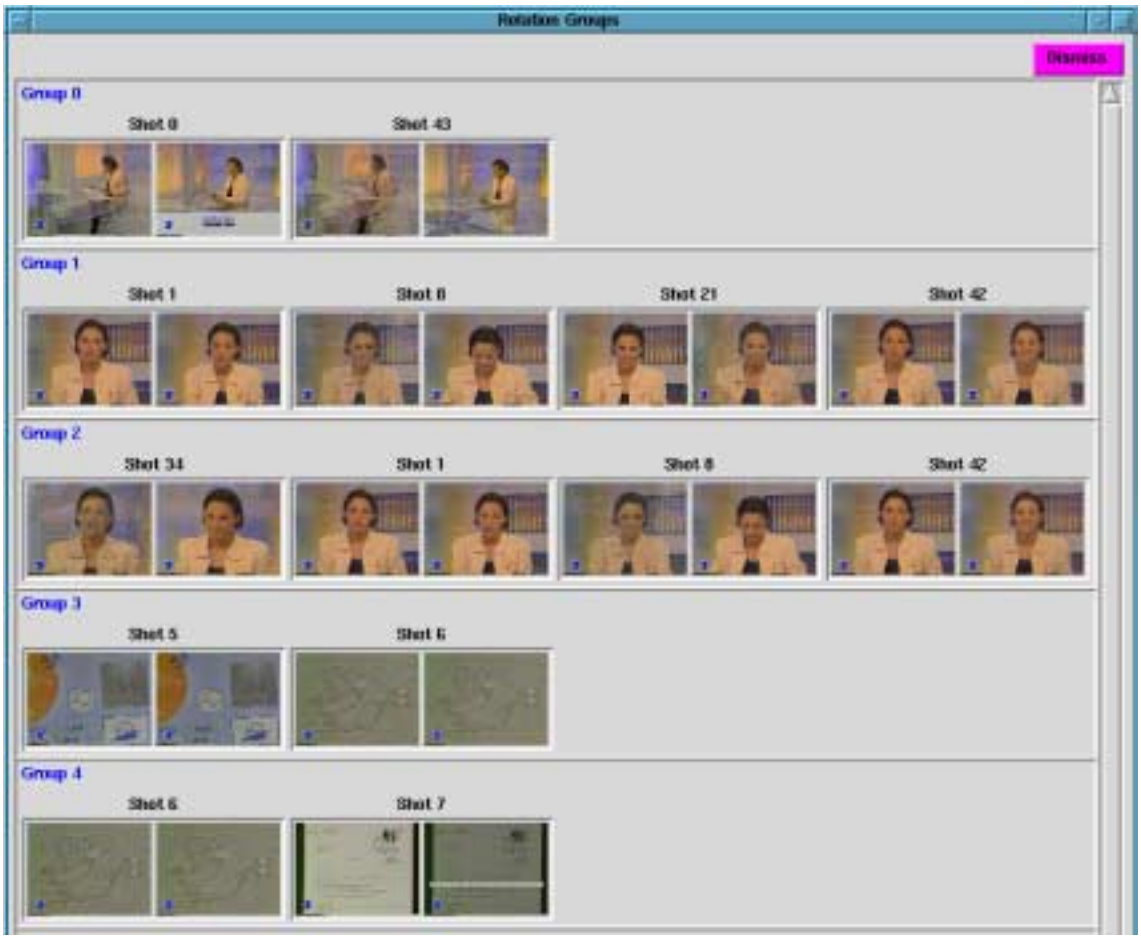


Figure 11: First five strict groups of related shots for the sequence *Jtv1*. Groups 1 and 2 reach the maximal probabilities (86.7% and 87.7% respectively) for the newscaster detection.

5 Conclusion

Four different tools: transition detection, shot change detection, shot relation detection and TV newscaster detection were presented and discussed in this paper. All of them

Group	Skin detection	Background motionless	Number of relations	Begin-end relations	% total
0	94.3	74.6	40.0	100.0	77.2
1	**83.1**	**87.9**	**80.0**	**100.0**	**87.7**
2	**79.0**	**87.9**	**80.0**	**100.0**	**86.7**
3	27.1	96.1	40.0	0.0	40.8
4	25.4	80.6	40.0	0.0	36.5
5	56.8	88.7	100.0	0.0	61.4
6	32.4	90.0	40.0	0.0	40.6
7	46.3	88.2	40.0	50.0	56.1
8	79.1	87.3	60.0	0.0	56.6
9	62.6	86.1	60.0	0.0	52.2
10	62.4	86.4	60.0	0.0	52.3
11	71.1	92.8	80.0	0.0	61.0
12	61.9	87.5	60.0	0.0	52.4
13	62.4	92.1	60.0	0.0	53.6

Figure 12: Values of the four criteria for the relation groups of the sequence *jtv1*.

provide good results, as proved by the tests on the 22 sequences, whilst remaining both simple and fast. Their simplicity mostly lies in the fact that they are based on the same local algorithm of computing a difference criterion on images or on part of images. This algorithm is the source of several syntactic hints, useful for the indexing of video documents. We have access to quite high level information using only a simple comparison between images, which is encouraging for the indexing problem. This allows one to start the building of a hierarchical representation of each video document, which could be further used in a video data indexing and retrieval system. Although the results are good, they still need to be improved, particularly for transition detection. For this reason, our current work deals with a new tool for extracting fades and dissolves based on their mathematical definition, given that the transition detection tool does not perform well with this type of transition.

6 Ackowledgments

Parts of this study are supported by the CNET-CCETT (France Télécom). Other parts are supported by the KODAK-PATHE company. The original images are copyrighted by the CCETT and by the French television channels FR3 and A2.

References

[1] Farshid Arman, Arding Hsu, and Ming-Yee Chiu. Image processing on compressed data for large video databases. *Proceedings of the ACM MultiMedia, California, USA.*

Association of Computing Machinery, pages 267–272, june 1993.

[2] P. Bouthemy and F. Ganansia. Video partitioning and camera motion characterization for content-based video indexing. In The Institute of Electrical and Inc. Electronics Engineers, editors, *Proceedings ICIP'96, IEEE International conference on image processing*, volume 1, pages 905–908, Lausanne, Switzerland, september, 16-19 1996.

[3] G. Davenport, T. G. Aguierre Smith, and N. Pincevert. Cinematic primitives for multimedia. *IEEE Computer Graphics and Applications*, pages 67–74, july 1991.

[4] Jian Feng, Kwok-Tung Lo, and Hassan Mehrpour. Scene change detection algorithm for mpeg video sequence. In The Institute of Electrical and Inc. Electronics Engineers, editors, *Proceedings ICIP'96, IEEE International conference on image processing*, volume 2, pages 821–824, Lausanne, Switzerland, september, 16-19 1996.

[5] David A. Forsyth and Margaret M. Fleck. Identifying nude pictures. In *3rd IEEE Workshop on applications of computer vision*, Sarasota, Florida, USA, december, 2-4 1996.

[6] Arun Hampapur, Ramesh Jain, and Terry E. Weymouth. Digital video segmentation. In *Proceedings Second Annual ACM MultiMedia Conference and Exposition. Association of Computing Machinery*, pages 357–364, october 1994.

[7] Arun Hampapur, Ramesh Jain, and Terry E. Weymouth. Production model based digital video segmentation. *Journal of Multimedia Tools and Applications*, 1(1):9–46, march 1995.

[8] Rune Hjelsvold. Video information contents and architecture. In *Proceedings of the 4th International Conference on Extending Database Technology*, Cambridge, UK, March 28-31 1994.

[9] Rune Hjelsvold and Roger Midtstraum. Modelling and querying video data. *Proceedings of the 20th VLDB conference*, 1994.

[10] Akio Nagasaka and Yuzuru Tanaka. Automatic video indexing and full-video search for object appearances. In *2nd Working Conference on Visual Database Systems, IFIP WG 2.6.*, pages 119–133, Budapest, Hungary, october 1991.

[11] M. Schmitt and J. Mattioli. *Morphologie Mathmatique*. Masson, Paris, 1994.

[12] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

[13] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems 1(1)*, pages 10–28, 1993.