Algorithms for Compressed Video Processing in Multimedia Applications *

Francesca Dardi and Giovanni L. Sicuranza
DEEI - University of Trieste, via Valerio 10, 34127 Trieste, Italy
e-mail: sicuranza@ipl.univ.trieste.it

Abstract

Recently, there have been considerable efforts for detecting scene changes and for efficient matching and clustering of video shots using compressed data. We propose here some algorithms that reduce false detections in cases of significant object motions, subtitling, picture in picture or special effects.

Introduction

With the emergence of digital library, there is an urgent need to automatically extract key informations from images and videos for indexing, fast and easy retrievals and scene analysis. However, image compression tecniques have been developed for efficient storage and it can be expected that a large part of future video material will be in compressed form. Therefore, it is thus advantageous to implement algorithms to operate directly on compressed data without having to first perform full decompression. For compressed video, a common first step is to segment the video into temporal "shots", each representing an event or continuous sequence of actions. The boundaries between video shots are commonly known as scene changes. In general, there are two types of camera shot: break and dissolve transitions. A break transition is an abrupt change between two camera shots that is completed within two consecutive frames. A dissolve is a transition between two camera shots that takes several frames. However, frames on both side of a scene change generally display a significant variation in the content. The effects of a scene change can be observed on dc_images or on motion vectors in a MPEG bitstream.

Dc_images are reduced versions of the original images and they are useful for fast and efficient video analysis operations. The original image is divided in blocks of 8*8 pixels: the (i,j) pixel of the dc_image is 8 times the average value of the (i,j) block of the original image. The change detection can be based on the measurement of successive dc-level differences: an example is the distance

$$d(X,Y) = \sum_{i,j}(|x_{i,j} - y_{i,j}|) \qquad (1)$$

where X and Y are dc_images corresponding to consecutive frames of the sequence [1]. While a break transition is usually represented by a single high pulse, a dissolve transition is usually represented by a number of consecutive medium-heighted pulses. A break transition can be detected using a threshold; for a dissolve, the difference of the current frame can be compared whit the average of the differences of the frames within a window preceeding the current frame [2].

A scene change algorithm can be based also on considerations about the motion vectors in the MPEG bitstream [3]: in fact, the occuring of a scene change causes the prevalence of determined prediction types in I, P, B-frames of the sequence.

The performance of scene change detection algorithms using only dc coefficients can be affected by factors such image manipulations or rapid object motions: these factors will increase the possibility of missed or false detection.

For example, an abrupt change as in the case of a "picture in picture" generates a difference diagram with a single high pulse; rapid object motions or subtitling cause consecutive medium-heighted pulses as a dissolve. These situations are due to the fact that previous differences do not incorpo-

*This work was partially supported by MURST and ESPRIT LTR Project 20229 Noblesse

rate motion analysis and are not able to examine the local activity. We can improve the accuracy of the parsing algorithms by developing a new approach which captures significant local variations in the content of two frames. Our method offers as advantage the use of the data available in the compressed stream, i.e. the dc coefficients for I-frames and the motion vectors; all the information is included in the distance definitions.

1 Measurement of local changes

To detect local variations, we consider each image as a matrix, the elements of which are 8×8 pixel blocks: we evaluate the sum of absolute differences between dc coefficients of corresponding blocks for each block row and each block column of two frames. We propose two difference projection vectors defined as follows:

$$P_{col}(X_k, X_{k+1})(j) = \sum_{i=1}^{N}(|\Delta DC(i,j)_k| * w(i,j)) \quad (2)$$

$$P_{row}(X_k, X_{k+1})(i) = \sum_{j=1}^{M}(|\Delta DC(i,j)_k| * w(i,j)) \quad (3)$$

$$d_1(X_k, X_{k+1}) = \sum_{i=1}^{N}\sum_{j=1}^{M}(|\Delta DC(i,j)_k| * w(i,j)) \quad (4)$$

where X_k is the current I-frame, X_{k+1} is the next I-frame, $DC(i,j)_k$ is the dc coefficient for the block (i,j) of the frame X_k, $\Delta DC(i,j)_k$ is the difference between the (i,j) dc coefficients in X_k and X_{k+1}, M and N are the total number of block rows and the total number of block columns respectively; the $w(i,j)$ weight depends on the motion vector of the block (i,j) in a B-frame close to X_{k+1}.
For video-conferences, we can assume that certain features, such as object motion and lighting, should not change significantly within a shot: so the extraction of the dc coefficients only from I-frames is sufficient to guarantee a correct analysis, i.e. $w(i,j)$ in (2), (3) and (4) is always set equal to 1.
In a shot where object and camera motions are relevant, it is more difficult to detect variations in the contents and this method is not accurate. We can solve this problem by considering that image manipulations or special effects often induce specific patterns in the field of motion vectors; so we can try to recognize an event either from luminance components or from a typical configuration of motion vectors, by setting the weight $w(i,j)$ to 1 if the motion vector for the block (i,j) corresponds to the pattern induced from the selected operation: otherwise, the weight is set to 0.
The next sections show how common manipulations as the subtitling and the "picture in picture" can be detected by the combined application of difference projection vectors and global metrics.

2 Text extraction

In images of a subtitled sequence, the text usually slides in a fixed number of rows or of columns (Figure 1): this event shows particular properties which can be identified by dc coefficient and motion vector analysis. We assume for simplicity that

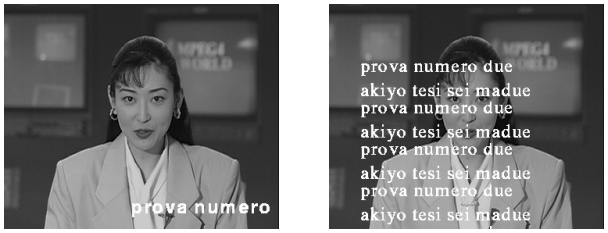

Figure 1: Sliding in rows (on the left); sliding in columns (on the right).

text pixel values are much higher than background pixel values: therefore, the dc coefficients strongly depend on the number of text pixels in each block. For a given block, this number changes from frame to frame because of the text sliding: so more significant variations in the dc coefficient values correspond to blocks belonging to the region of the text. This fact means that high values in a difference projection vector can indicate the presence (and the position) of text in a sequence.
Subtitling involves also the motion vectors; in fact, the constancy of the sliding velocity causes a specific motion configuration characterized by a most probable vector which coincides with the sliding velocity.

2.1 Sliding in rows

Let $X_k, k = 1, \ldots, N$ be the I-frames of a subtitled sequence; let the text occupy the block rows l and $l+1$; in this situation, a typical $P_{row}(X_k, X_{k+1})$ vector plot shows two adjacent peaks, located in positions l and $l+1$, which persist until the text

slides out of the image. Therefore, to automatically detect the text, we should find a sufficient number of consecutive $P_{row}(X_k, X_{k+1})$ vectors that present adjacent peaks in constant position. A value is marked as a peak in $P_{row}(X_k, X_{k+1})$ only if it exceeds the threshold $Th_{row}(k) = \mu(k) + m * \delta(k)$ where $\mu(k)$ and $\delta(k)$ are respectively the average and the standard deviation calculated for $P_{row}(X_k, X_{k+1})$; m is an experimental parameter. To locate the text exactly, it is possible to capture its beginning/end in a I-frame X_k by evaluating the vector $P_{col}(X_k, X_{k+1})$: the position of the first/last value higher than the mean value in this vector corresponds to first/last letter.

If the text is sufficiently extended in a P or B frame, we can expect that a dominant motion vector will be present in the block rows l and $l + 1$: this vector will coincide with the sliding velocity, that is needed either to recognize a subtitling or to track and reconstruct the text.

Once the subtitling has been recognized, the text region is decoded from consecutive I-frames, with a frequency which depends on the sliding velocity, to avoid information redundancy: all the extractions are composed in a single image.

To exhibit the text, we perform a threshold operation on the composed image; the threshold is selected so as most of the background pixels will be set to zero and the others to 255.The threshold value can be easily determined by the observation of the luminance histogram. In fact, a typical histogram for the composed image shows a peak corresponding to the text pixel value: to obtain a binary image, it is sufficient to assume as threshold the minimum value before the peak. For high motion shots, the weight $w(i, j)$ is set to 1 if the motion vector (i, j) in the B-frame immediately subsequent the I-frame X_{k+1} in (3) presents a null vertical component and a horizontal component value included in a suitable range; this second condition is based on the consideration that the sliding velocity has to be small to consent the reading.

2.2 Sliding in columns

It is possible to extend the previous considerations to this case. If a text is sliding in the block columns $l, \ldots, l + h$, the derived $P_{col}(X_k, X_{k+1})$ vector contains high values in these positions: however, it is necessary to add a new criterion to identify this event, because variations in such large region of the image could be confused with other object motions. This kind of subtitling can be observed also on a global difference diagram, that appears as a number of consecutive medium-heighted pulses and can be detected as a dissolve transition. For frames corresponding to the pulses, the $P_{col}(X_k, X_{k+1})$ vectors are examined to detect values exceeding the threshold $Th_{col}(k) = \mu(k)$ where $\mu(k)$ is the average calculated on $P_{col}(X_k, X_{k+1})$; if the same adjacent positions are found in at least m consecutive vectors, a suspected subtitling is declared. This criterion is needed to distinguish a subtitling from a dissolve transition. Then we proceed as shown in section 3.1. For high motion shots, the weight $w(i, j)$ is set to 1 if the motion vector (i, j) in the B-frame immediately subsequent the I-frame X_{k+1} in (2) presents a null horizontal component and a vertical component included in a suitable range. To improve the detection, we propose some treatments: a local median filter (size 3) is applied on the difference d_1 sequence to smooth the diagram and eliminate peaks due to abrupt changes; in the projection vector, the filter contributes to stabilize the marked positions; finally, it is useful to extract, in addition to the signal text zone, some neighbouring blocks to avoid information loss.

3 Picture in picture

A "picture in picture" can be achieved by inserting images from a first sequence in images of a second sequence: each sequence evolves independently from the other. This event appears as a "local scene change", i.e. the same effects of a scene change are present in a fixed image submatrix. Our algorithm detects suspected "picture in picture" cases on I, P, B-frames separately.

3.1 Algorithm for I-frames

The fist step provides the detection of peaks in the difference d_1 diagram; $w(i, j)$ is set to 1 if the block (i, j) in the B-frame before the I-frame X_{k+1} in (4) is only forward predicted. In fact, if a "picture in picture" happens on a I-frame, the previous B-frame would present a submatrix with most of the motion vectors pointing forward. The diagram peaks can be found as in parsing algorithms. However, a peak could correspond to a scene change: to distinguish the two events, we evaluate the difference projection vectors with $w(i, j)$ values estabilished as shown before. Row and column projec-

tion vectors P_{row} and P_{col}, in a case of "picture in picture", are useful to detect an image submatrix, which contains the greatest variations. This submatrix is indicated by elements that exceed the average value in each of the two vectors.
Analogously, it is possible to detect a "picture in picture" inserted in B or P-frames, as shown in [4].

4 Experimental results

The test sequence is a low motion shot called "Akiyo" with image size of 352×288 pixels. We impose as parameter values $m = 2$ and $n = 3$. At the frame 100 of "Akiyo", a text begins to slide with a velocity of four pixels per frame in the block rows $(33, 34)$. The frame 100 is a B-frame between the I-frames X_9 and X_{10}. $P_{row}(X_k, X_{k+1})$, $k = 9, \ldots, 24$, contains elements exceeding the threshold $Th_{row}(k)$ in positions $(33, 34)$ (Figure 2). The subtitling is declared: therefore, we decode the region corresponding to the text in X_{11}. To find the sliding velocity, we observe motion vectors in the region of the text in the B-frame after X_{12}: all the blocks use the forward prediction, the 64.29% of the forward vectors has only the horizontal component which coincides with the searched velocity. Once known the velocity, we can determine the next I-frame which will relevantly contain new information: since the horizontal image size is 352 and the sliding velocity is 4 pixels per frame, we will avoid redundancy if we decode the text region in X_{17}. For the same reason, the last extraction is executed on X_{23}. The luminance histogram for the composed image is shown in Figure 3: the peak in the histogram corresponds to the value 255, i.e. the text pixel value. If we perform a threshold operation with respect to the value 230, we obtain the result in Figure 3: the text has been completely reconstructed.

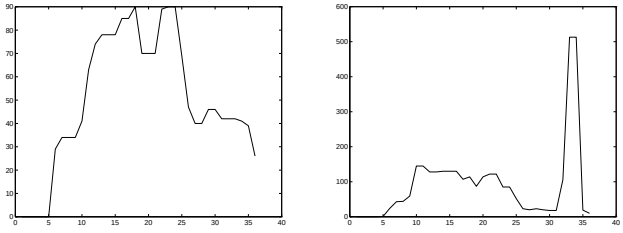

Figure 2: Example of P_{row} vector for images with no text (on the left); example of P_{row} vector for images with text (on the right).

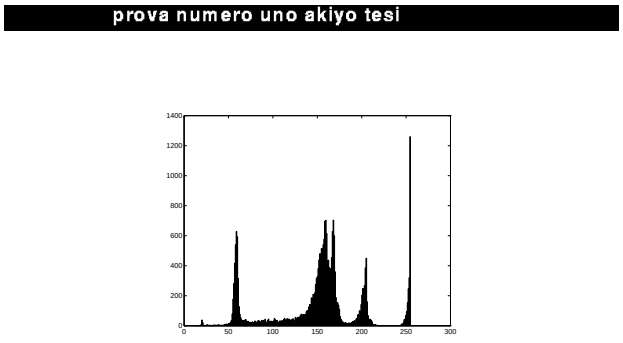

Figure 3: Luminance histogram and the threshold operation result.

5 Conclusions

The metrics defined in this paper can be applied to the identification of a wide range of effects, such as the abrupt scene changes or appareance/disappareance of a sufficiently large caption in a frame, tracking an object regularly moving on the background, etc. In addition, special effects as zooming, panning, titling and camera motions could be detected by evaluating the differences d_1; for each operation, the weight values have to be selected in accordance with the specific pattern induced in the field of motion vectors. The proposed approach can improve video segmentation, key-frames extraction and clustering processes.

References

[1] B.L. Yeo and B. Liu, *"Rapid scene analysis on compressed video"*, IEEE Transaction on Circuits and Systems for Video Technology, vol.5, no.6, Dec 1995.

[2] K. Shen and E.J. Delp, *"A fast algorithm for video parsing using MPEG compressed video sequence"*, ICIP 1995.

[3] J. Meng, Y. Juan and S.F. Chang, *"Scene change detection in a MPEG compressed video sequence"*, Digital Video Compression: Algorithms and Technology, vol. SPIE-2419, Feb. 1995.

[4] F. Dardi, Technical Report, Project 20229 Noblesse, Sept. 1997.