TEMPORAL SEGMENTATION FOR VIDEO PROCESSING USING MATLAB

Héctor Martel, Pau Rotger, Manuel Ruiz-Sarmiento

Universitat Pompeu Fabra (Barcelona, Spain)

ABSTRACT

Temporal video segmentation is the first step towards automatic annotation of digital video for browsing and retrieval. In this article we will present and study the principal problems of the existing shot detection algorithms and propose an alternative way to perform the temporal segmentation of video files. We will first introduce the methods and give their motivation as well as details of its implementation. Subsequently, we will evaluate the results in order to determine the performance of the new method compared with the existing ones. Finally, significant conclusions will be drawn from the analyzed data and the comparison of the different methods.

Index Terms — *Shot detection, Frame Histogram Differences, Bhattacharyya Coefficient, Pixelwise Comparison, Qindex, Scene, Video Processing.*

1. INTRODUCTION

Temporal segmentation of a video, also shot detection from now, consists on dividing it into several parts according to the changes of shot that are present in the sequence. Therefore it is very important to distinguish between the concepts of scene and shot. The first is defined as a continuous sequence that is temporally and spatially coherent in the real world, but it does not necessarily mean cohesion in the projection of the film. The second concept refers to the longest continuous sequence that originates from a single camera recording with no cuts. Therefore, several shots make up a scene [1].

The temporal segmentation problem can be divided into two stages, which are scoring and thresholding. Scoring refers to performing an analysis between two consecutive frames and assigning a number to it, whereas thresholding is the decision part in which these numbers are evaluated.

It is important to take into account that temporal video segmentation is the first step towards automatic annotation of digital video for browsing and recovery. [1, 2] Despite it is a trivial task for people to tell shots apart, that is not the case for computers, so shot detection methods present several problems. In this section we will briefly explain the most important ones.

One of the main problems to consider is the lack of information. Cut detection would be a trivial task if each frame of a video was enriched with additional information about when and by which camera it was taken. However, all that metadata added to the frames would cause the video file size to increase, making storage and reproduction more expensive. For that reason, the scope of the algorithms is limited to the content of the frames themselves. [2]

The algorithms for detecting shots are based on the similarity between frames. Therefore, it turns out to be obvious that a very soft and natural change using gradual transitions, like dissolves or fades, is going to be harder to detect than abrupt cut transitions. This is because the similarity between the frames n and n+1 is higher in the first case, when the changes take a longer time involving a larger number of frames. Following the same logic, two consecutive frames will be very different if the changes are fast, making shot detection easier.

Another issue, but not less important, is that shot boundary detection is not sensitive to object and camera motion. In fact, a big object moving in the scene may cause the algorithm to trigger false positives if the changes are significant enough. This just illustrates how difficult the shot detection problem actually is. In other to evaluate the algorithms alone we are not going to consider object tracking and optical flow here, because it goes beyond the purpose of this study.

2. PROPOSED ALGORITHMS

In this section we are going to introduce several methods that can be used to perform a temporal segmentation. Some of them have been explained in class, whereas the ones from a different source are denoted with a (*). It is important to mention that some work better than the others depending on the shot and the transitions, but that there is also a tradeoff between precision and computational cost.

All the methods below cover the scoring part of the problem. This means that either fixed or adaptive thresholding must be applied afterwards to in order to make the decision. The reader can find the code in the same delivery folder with the all of the implementations for this article.

Pixelwise Comparison (PC)

This method is computed by taking the euclidean distance of the RGB components and getting the average for the frame. Let φ be a video file which is composed by RGB frames of size MxN. The index n indicates the frame number, i and j refer to a single pixel within that frame and k refers to the color component. The Pixel Comparison can be expressed as follows:

$$PC = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{\sum_{k=1}^{3}|\varphi(i,j,k,n) - \varphi(i,j,k,n+1)|^2}$$

The disadvantage is that this method is not able to distinguish between a large change in a small area and a small change in a large area. Thus, it is are very sensitive to object and camera movements and that may cause false positives.

Sum of Absolute Differences (SAD)

This is the simplest and most obvious method that can be used, but it does the job. It consists on adding up the absolute values of the pixel wise difference between two consecutive frames [2]. Given that the frames of the same shot are highly correlated, the measured SAD will be low compared to the one coming from a transition.

Let φ be a video file which is composed by RGB frames of size MxN. The index n indicates the frame number, i and j refer to a single pixel within that frame and k refers to the color component. The Sum of Absolute Differences can be expressed as follows:

$$SAD = \sum_{k=1}^{3}\sum_{i=1}^{M}\sum_{j=1}^{N}|\varphi(i,j,k,n) - \varphi(i,j,k,n+1)|$$

This method is relatively easy, although it may detect a cut in the scene when actually there is just an object moving fast, or the camera is not still. So this is a simple method that is very sensitive to changes, thus, it produces a large number of a false hits.

Histogram Differences (HD)

This method is similar to the last one, but it uses statistical information of the frames instead of comparing sets of concrete pixel values. [2] The histogram of an image represents the luminance distribution of the pixels for one channel. Since we are dealing with RGB frames, the frames are first converted to grayscale for simplicity and then the luminance distribution is calculated [1]. Let φ be a video file which frames n and n+1 have histograms $H_n(x)$ and $H_{n+1}(x)$ respectively. Notice that the integral term corresponds to the Bhattacharyya coefficient.The Histogram Differences can be expressed as a distance as follows:

$$HD = \sqrt{1 - \int_{-\infty}^{\infty}\sqrt{H_n(x)H_{n+1}(x)}dx}$$

The advantage of this method respect to SAD is that HD is less sensitive to small changes, producing less false hits. However, it might be the case that two frames coming from a hard cut still have the same color distribution, so in this case HD would miss the hit.

Qindex and Qindex by blocks (*)

This is an existing method proposed in [3] that has been proved to measure the quality of an image respect to a reference taking into account statistical properties of both. As the mentioned study has shown, it works more accordingly to human perception than other objective similarity measures. It is worth trying an approach of this kind in the field of video processing since it works well for static images. Let x and y be two frames of mean $\bar{x}$ and $\bar{y}$, and σ_x^2, σ_y^2, σ_{xy} the variances and the covariance respectively. The Qindex can be expressed as:

$$Q_i = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2) + (\bar{x}^2 + \bar{y}^2)}$$

Notice that it can also be executed dividing the image into blocks of a fixed size, and then computing the Qindex for all the blocks and then getting the average. We are going to study both. Assuming the frames can be divided into M blocks:

$$Q = \frac{1}{M}\sum_{i=1}^{M} Q_i$$

Mixed Comparison (*)

This is the one that we are going to introduce as an alternative proposal for shot detection. It mixes two methods together in order to perform a multiresolution analysis of the video and obtain the best of both. It is done by using Pixel Comparison and Histogram Differences, which have been previously explained. By mixing these two methods we obtain more robustness when it comes to detecting scene changes and also soften the scenes with very fast movements.

The result of this method is extracted by doing the weighted average of the results of each frame comparison. Therefore, the expression for the n-th frame is just:

$$MC(n) = w_1 \cdot PC(n) + w_2 \cdot HD(n) \qquad \text{where} \quad w_1 + w_2 = 1$$

3. EVALUATION

In this section we will present and analyze the results that have been obtained. Three more videos have been used apart from the one provided in order to test all the methods using a wide variety of material. Each one of the videos has been selected for having different interesting features that might be challenging to evaluate depending on the method. All the videos can be found in the same folder of the delivery and are the following ones: cuentaatras.avi [5], bigbang.avi [6], timelapse.avi [7], and vipscenevideoclip.avi [8].

The first one ("cuentaatras.avi" [5]) is a piece of the mythical countdown of the twentieth century filmmaking industry, going from six to two. This video has been considered an interesting case of study because it comes from a non-natural source and the scene changes are abrupt and clean, making it easy to detect a priori.

The second video ("bigbang.avi" [6]) is the counterexample. It is the intro of the famous TV show "The Big Bang Theory" well known for having many changing images in a very short period of time. This video has been chosen because it is a good way to show that despite the efficiency of shot detection algorithms, they will not always work optimally and detect all the changes properly.

The last proposed video ("timelapse.avi" [7]) refers to the technique whereby the frequency at which film frames are captured is much lower than the used to view the sequence at the end. The visual effect is that everything that has been captured moves very quickly, such as the movement of clouds, sunset, the opening of an egg, etc. The consequence of taking frames over a longer period is that the differences between two consecutive frames are much higher in general. Because of that, this video sequence can also be a representative case of study.

Now that we know what the different cases are intended to test we will show and analyze the results using the six methods for all the videos listed above.

The analysis of the first video sequence gives the results we were expecting. As it can be seen in Figure 1, the peaks that

correspond to shot changes are clean and very distinguished from the rest of the values. The magnitude of the first three peaks is very similar because the type of change is the same between the numbers six, five, four and three. However this is not the case with the fourth scene because the next frame is completely black after the number two appears, causing an abrupt change and an absolute maximum of difference.

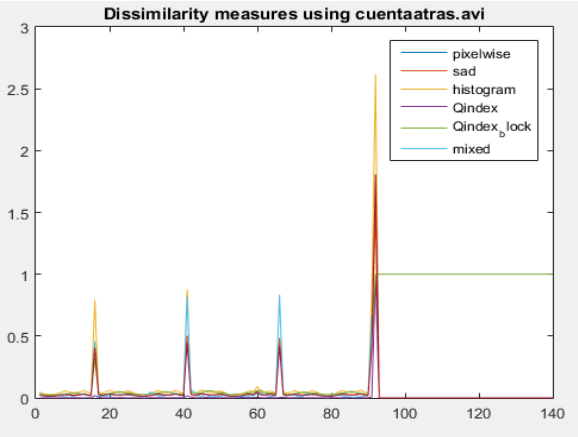

Figure 1. Analysis of "cuentaatras.avi". Pixelwise: recall of 100% and precision of 100%; SAD: recall of 100% and precision of 100%; Histogram: recall of 75% and precision of 100%; Qindex: recall of 75% and precision of 100%; Qindex_block: recall of 75% and precision of 75%; Mixed: recall of 100% and precision of 100%.

The analysis of the second video results in the totally opposite situation we had before. In this case the lines are very irregular due to the fact that the video is constantly changing, as it can be seen in Figure 2. Since there are no clear peaks outstanding from the rest of the values it is hard for the method to detect the shots well. In fact, it can be seen that the recall and precision drop to non-acceptable values for any of the methods. It is important to mention that this case of study was intended to be a very extreme one.

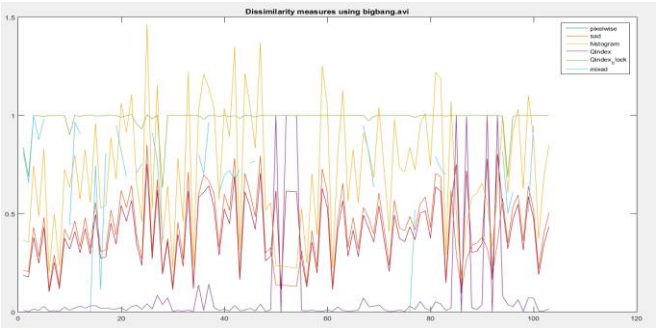

Figure 2. Analysis of "bigbang.avi". Pixelwise: recall of 3.5% and precision of 66.6%; SAD: recall of 3.5% and precision of 66.6%; Histogram: recall of 3.5% and precision of 66.6%; Qindex: recall of 1.8% and precision of 50%; Qindex_block: recall of 5.3% and precision of 100%; Mixed: recall of 5.3% and precision of 100%.

For the timelapse video the results look more like in the first case. The lines are well defined and it can be clearly seen in Figure 3 that there are three prominent peaks. There is also some added noise due to the fact that the frame rate is lower than in a regular video causing the frames not to be as similar depending on the changes occurred over that period of time.

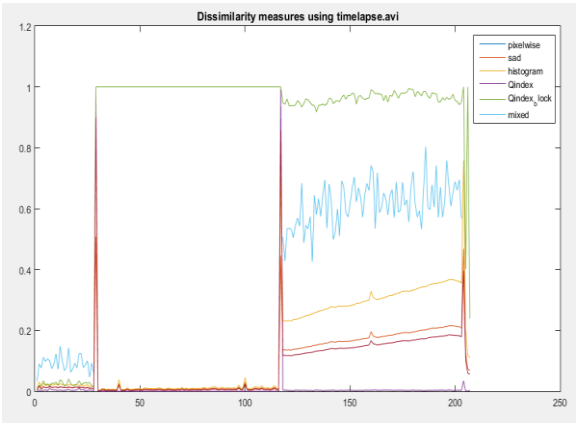

Figure 3. Analysis of "timelapse.avi". Pixelwise: recall of 100% and precision of 60%; SAD: recall of 100% and precision of 50%; Histogram: recall of 100% and precision of 100%; Qindex: recall of 33.3% and precision of 50%; Qindex_block: recall of 0% and precision of 0%; Mixed: recall of 100% and precision of 100%.

In the last case of study we are evaluating the supplied video ("vipscenevideoclip.avi" [8]) which has two changes of scene. In these cuts there are big changes in color, for instance when the snow appears. Since the color contrast is very high from one frame to the next, the peaks get higher and very well defined in the methods that take distributions into account.

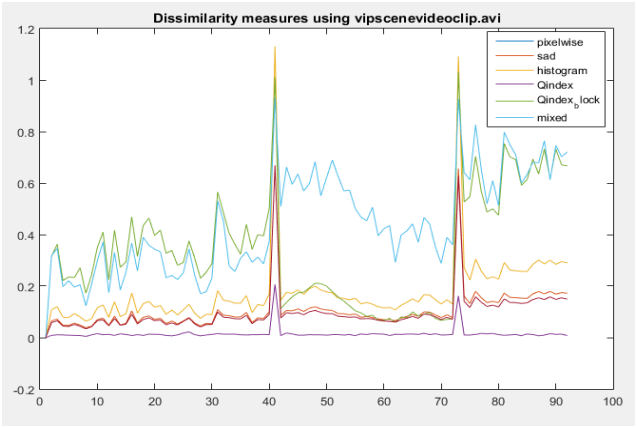

Figure 4. Analysis of "vipscenevideoclip.avi". Pixelwise: recall of 100% and precision of 66.6%; SAD: recall of 100% and precision of 66.6%; Histogram: recall of 100% and precision of 100%; Qindex: recall of 100% and precision of 66.6%; Qindex_block: recall of 100% and precision of 66.6%; Mixed: recall of 100% and precision of 66.6%.

4. CONCLUSIONS

From the results that have been obtained in the last section we are going to compare the methods and give a perspective of what has been studied and what is yet to be improved. The comparison between the methods is done taking into account the both the general purpose and the oriented to task as criteria.

On the one hand, the method that turned out to be better in terms of precision for a general purpose was Histogram Differences (HD), having an average recall of 69.6% and a precision of 91.7% over the different cases tested. This means that for an unknown type of video file this is the method that is best suitable for the analysis. It makes sense that a method that takes into account statistical properties of the frames can be used in a generic setting, since cuts are normally related with changes in the luminance, or color, distribution. This is mainly due to the fact that the camera, angle and light may change from one shot to another. However, this method can fail when the colors are distributed in the same way but in different locations of the frame, making it hard to detect the shot in situations where some of the conditions keep unchanged.

On the other hand, we have to consider what kind of video is going to be analyzed and see if there is a particular method that does a specialized task extremely well. The four examples have different features that can be summarized as the variation of the frame rate, the cut rate and the naturalness of the scene.

For videos in which the frame rate is lower, and therefore the differences between any pair of consecutive frames are higher, the best methods were Histogram Differences (HD) and the alternative Mixed method proposed in this study. Both have obtained a recall of 100% and a precision of 100% in the timelapse video. The kind of performance in which we are concerned here is that the algorithm does not trigger false positives due to the fact that frames might have changed significantly over the same shot. This is mainly why other methods that rely on the exact pixel locations cannot perform well. Pixelwise Comparison (PC) and Sum of Absolute Differences (SAD) did not get a high enough precision, which was less than 60% in both cases, despite the recall was 100%.

In the case of videos having a lot of cuts the analysis is similar to the previous one, but now what is relevant is not to miss any of the cuts. This is impossible to achieve with the adaptive threshold technique since we made the assumption that the number of cuts and the length of the video are directly related, but the fixed threshold can solve the problem if it is properly set. Anyways the results were not acceptable with any of the methods tested, since the recall was no greater than 5.3%, despite the precision was 100%. At this point two important conclusions on fast changing videos can be extracted.

First, this type of fast changing and source-varying videos are the most extreme example we can take for the testing. By source-varying we understand that the images are natural and non-natural, that come from many different cameras (the video is actually formed as a composition of pictures). How extreme this case is can be seen in Figure 2, where none of the lines is steady no matter the method, contrarily as in the rest of the graphs.

Second, the adaptive threshold is not always the best option if the video is has a high cut rate. As it was implemented, the adaptive threshold needs the number of cuts to be proportional to the length of the video, and the criterion was taken having natural scenes in mind.

Finally, there is also important to consider what happens when the video comes from an artificial source, in other words, it has been generated by a computer or any other device so that the result cannot be found in nature. This is what happens with the countdown video, which shows numbers on the screen in two relatively homogeneous colors. Moreover, there is very little movement, which is something hard to achieve in nature as well.

For this type of video, the methods based on static pixel information gave the best performance. Pixelwise Comparison (PC), Sum of Absolute Differences (SAD) and the alternative Mixed method obtained a recall of 100% and a precision of 100%. The rest though, did get a 75% in either recall or precision showing to be worse detecting non-natural changes.

That being said, there are some aspects that can be improved when it comes to the performance of the algorithms in terms of recall and precision, as it has been commented at the beginning of this document. Object tracking and optical flow can make a significant improvement in temporal segmentation of videos because they provide a more robust evaluation of the motion of the scene. If objects suddenly disappear from the scene that means there is a new shot with a very high probability, even though the statistical properties of the frames may not have changed much.

5. REFERENCES

[1] “Practical Image and Video Processing Using MATLAB” - O. Marques (Wiley_IEEE, 2011) BBS

[2] “Shot transition detection”, Wikipedia – The free encyclopedia, 20 November 2015, last visited 17 February 2016. https://en.wikipedia.org/wiki/Shot_transition_detection#Vastness_of_the_problem

[3] “A Universal Image Quality Index” by Zhou Wang and Alan C. Bovi, IEEE Signal Processing Letters, VOL. 9, NO. 3, March 2002. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=995823

[4] “Shot-Boundary Detection: Unraveled and Resolved?”, uploaded 2 February 2002 , last visited 16 February 2016.http://prb.tudelft.nl/sites/default/files/IEEETCSVT_shots.pdf

[5] “cuenta regresiva de cine y video efecto”, Downloaded from Youtube, uploaded 14 November 2011, last visited 17 February 2016. https://www.youtube.com/watch?v=yf0J_CaXDmU

[6] “The Big Bang Theory Intro Full Version”, Downloaded from Youtube, uploaded 1 November 2010, last visited 17 February 2016. https://www.youtube.com/watch?v=x6H7k3XBlk4

[7] “VisitGoldCoast.com – Australia's Gold Coast in Timelapse (30 seconds)”, Downloaded from Youtube, uploaded 10 August 2014, last visited 17 February 2016. https://www.youtube.com/watch?v=3xBSom2Awrg

[8] “vipscenevideoclip.avi”, Materials provided with “Lab 1 – Introduction to video processing”, Video Processing (UPF).

6. APPENDIX

In addition to what has been comented in the main sections of this document we will answer the questions of the rest of the assignment providing details and examples in each of them. We will not put or explain any implementation of the code because it is self explained in the coments. However, the reader can find the code in the same folder with the rest of the implementations for this article.

First, we were asked to compute the Frame Difference (FD) of a frame n and frame n-1. This distance can be computed as follows:

$$FD(x,y,n) = I(x,y,n) - I(x,y,n-1)$$

Now we will show you the results that give the FD with some of the videos provided in aula global, since they have to be in "yuv" format.

Watching the videos, we clearly see that have been done the FD. As we are subtracting one frame with its previous one it is clear that both frames will be very similar, which is why the video is almost black except in certain frames. This is because at that particular moment there is a greater difference between these, either by further movement or changes of colors. For example, in Figure 5 we are watching a news programme where the anchors, which are barely moving, are showing a video of a women dancing. Following the previous reasoning, this is why we can not see anything except the outline of a ballet dancer. Moreover, it is possible to see a mix of colors because in the last moment there is a big change of color in the background of the dancer from blue to brown. In Figure 6 we can observe a much different result. In this case we are watching a video with movement of the objects in the scene that are the boat as well as the camera. For this reason, exists more difference between frames and consequently more edges can be found in the result of the FD.

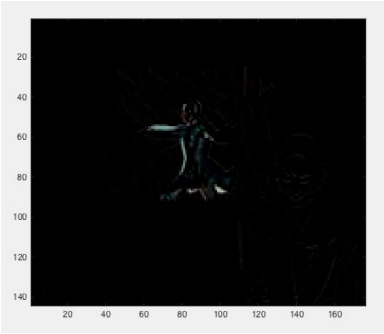

Figure 5. Interframe difference in the dancer video.

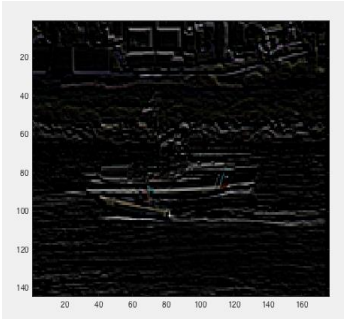

Figure 6. Interframe difference in the boat video.

In second place, we created a interlaced version (IV) of a progressive digital video. The result will have half of the number of frames of the original video. Therefore, before visually compare the two videos (the original and the one obtained), we can confirm that the interlaced video will be twice as fast the original. Furthermore, the video quality is also worse. In order to show that, we will use the video of the news programme:

Figure 7. Original frame

Figure 8. Same frame in the interlaced version.

From simple visual inspection it is easy to identify that Figure 7 is the original video and the Figure 8 the IV. Observing the two frames we can see a big difference in the dancer while the presenters are not very affected. As we mentioned before, the presenters are barely moving and the women spins very fast, creating a lot of movement. For this reason, we can see a distorsion in the women (autocorrelation between frames). Also we see how the video obtained is much faster than the original, proving our initial thoughts.

We would also highlight that by interweaving followed frames (n, n+1) makes the resulting video almost the same as the original, that is, if we interlace the frame n to frame $n + n_0$ the difference would have been greater. Thus, the higher the value of n greater the difference will be to the original video.