ANALYSIS OF ANIMATION MOVIES: SEGMENTATION, ABSTRACTION AND ANNOTATION

B. IONESCU, P. LAMBERT, D. COQUIN, V. BUZULOIU[*]

In acest articol este propus un sistem de analiză a filmelor de animaţie. Prima etapă de analiză constă in detecţia tranziţiilor video. Pentru a depaşii problemele ridcate de specificitatea filmelor de animaţie, sunt propuse o serie de metode imbunătăţite de detecţie a tranziţiilor „cuts" şi „fades". De asemenea, folosind un detector de „flash" modificat, este propusă o metodă de detecţie a unui efect specific filmelor de animaţie, denumit „short color change". In final este prezentată o metodă automată de rezumare si anotare a conţinutului video. Sistemul a fost testat folosind mai multe filme de animaţie.

In this paper a system for the analysis of animation movies is proposed. First the videos are divided into shots by detecting the video transitions. Improved cuts and fades detection techniques are proposed, in order to manage the difficulties raised by the peculiarity of the animation movies. Also, we use a modified flashlight detector to detect an animation specific visual effect which is named "short color change". An automatic video abstraction and annotation are proposed. The system is tested on several animation movies.

Keywords: cuts, fades detection, shot detection, video segmentation, video annotation, video summarization, video indexing.

Introduction

During the last years the content analysis of video documents became a key element of multimedia management, as their number and availability grow dramatically. Due to "The International Animated Film Festival" ("http://www.annecy.org"), which takes place every year at Annecy, France, since 1960, a very large database of animation movies is available. Managing thousands videos is a very difficult task. An analyzing system which permits video browsing and content interpretation is required.

In this paper we present the first results in the design of such a system, built to overcome the difficulties raised by the peculiarity of the animation movies. Animation movies are different of natural ones as the events do not

[*] PhD student, The Image Processing and Analysis Laboratory, University "POLITEHNICA" of Bucharest, Romania; Associate Professor, Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance, Université de Savoie, France; Prof., The Image Processing and Analysis Laboratory, University "POLITEHNICA" of Bucharest, Romania.

follow a natural way (objects or characters emerge and vanish without respecting any physical rules, the movements are not continuous), the camera motion is very complex (usually 3D), the characters could have any shape, a lot of visual effects are used, every animation movie has its own color distribution, various animation techniques are used (3D, cartoons, etc.).

In order to analyze the movie, first of all, it is divided into its fundamental video units or shots which provide the ground for nearly all existing video abstraction and high-level video segmentation algorithms [1]. The shot detection is performed by detecting the video transitions: cuts, fades and the ”short color change” effect or SCC.

A cut is a direct concatenation of two different shots and produces a temporal visual discontinuity in the video stream. Existing cut detection algorithms differ in the features used to measure that discontinuity. Cut detection techniques can be classified as: intensity/color based, contour based and motion based [1]. A comparison of all the three main cut detection approaches is presented in [2]. Different techniques using histograms were used: color-based histogram intersection metric using Cb-Cr and r-b space [3], YUV color histograms [4], sub-window based histogram comparison [5], multi-level Hausdorff distance histogram [6]. The proposed cut detection algorithm improves the detection by using the derivatives on Euclidian distances between color histograms in order to detect the visual discontinuity.

Fades are defined as an optical effect that causes a scene to emerge gradually on the screen from a monochromatic image (fade in) or vice versa (fade out). Various approaches using the contours changes and the standard deviation of the frame pixels are presented in [1] and [2]. The proposed fade detection method uses statistical measures of the pixel intensities on the YCbCr color space and is based on the algorithm proposed in [8]. A new parameter less sensitive to scene motion and illumination changes is proposed.

The SCC effect is specific to animation movies and correspond to short time color modifications (see Fig. 3), i.e. explosions, apparition and disappearance of objects, etc. The detection is based on the classical approach for detecting flashlights [9]. It uses a similarity measure between the frames just before and after a SCC occurred.

All the detected cuts, fades and SCC effects are fusioned into shots using a set rules. Having the shots, a static and a dynamic video abstract are proposed. Analysis of the video content requires that video content indexes are set. In our case indexes describing the positions and length of the shots, video transitions and SCC effects are generated automatically in a graphical visual annotation. As the video transitions have semantic information, i.e. the frequency of cuts is related to the video action, using the proposed video annotation one could analyze the video rhythm. An overview of the annotation techniques is presented in [10].

The paper is organized as follows: Section 1 describes the proposed algorithms for video shot detection, Section 2 presents the video abstraction and annotation, in Section 3 we present some experimental result and finally the Conclusions section contains final considerations and proposes future improvements.

1. Shot detection

The proposed shot detection is performed by detecting the video transitions: cuts, fades and the SCC special color effect. A histogram-based cut detection method is proposed and compared with a classical approach. The visual discontinuity of cuts will be transformed in representative distances between frame color histograms. Histograms were used because of their invariance to geometrical transformations. First of all, the frame's colors are reduced using an error diffusion algorithm on the XYZ color space [11] with the selection of colors in the Lab space from an a priori chosen color palette (webmaster color palette 216 [7]). After the color reduction, each frame is divided into four quadrants in order to reduce the influence of the scene entering objects.

For each quadrant a color histogram is computed. To reduce the processing time, the video document is subsampled by 2 (several tests were performed in order to see the influence of the subsampling period on the cut detection results). So, if the sequence has N frames, only $N/2$ are retained. For each frame k ($k = 1,...,N/2$), four Euclidian distances between its histograms and the correspondent ones of the next retained frame are computed. For the entire sequence four $N/2$ dimension distance vectors are obtained, $D_i()$ with $i \in \{1,2,3,4\}$.

The first method (referred as **4histograms method**) is a classical histogram based approach and uses all the four vectors in order to detect the cuts. A cut is detected whenever $\{ D_i(k) > t \}$ (color dissimilarity) and $\{ D_i(k-1) < t \}$ (color similarity), where *t* is the frame similarity threshold, i is the quadrant index and k is the current frame index. These two conditions must be verified for at least 3 of the 4 i index. This is a binary yes/no decision.

The second method (referred as **derivative method**) improves the decision (see Fig. 1) by using the second order derivatives in order to reduce the influence of motion on the cut detection (camera or object motion leads to high values of the distances). A single distance vector $D_{mean}()$ is computed as the mean of the four distance vectors, $D_i()$. A new vector $\ddot{D}_{mean}()$ is computed as the second order derivative of the $D_{mean}()$ vector. Negative values are set to 0. Scanning

$\ddot{D}_{mean}()$, a cut is detected whenever $\ddot{D}_{mean}(k) > t$ and $\ddot{D}_{mean}(k-1) < t$, where k is the current analyzed value index and *t* is the frame similarity threshold.

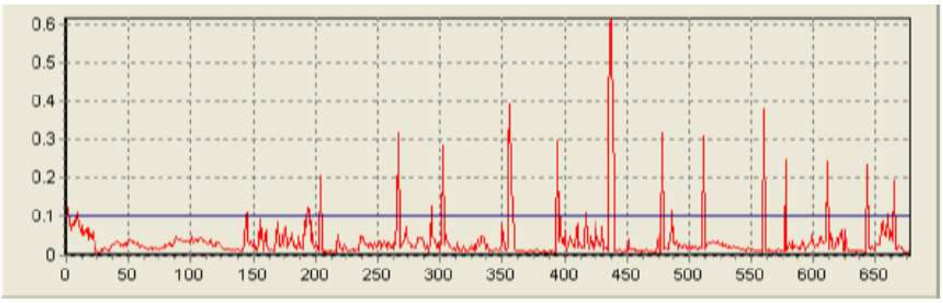

Fig.1.a. Cut detection using the 4histograms method ($D_{mean}()$ vector)

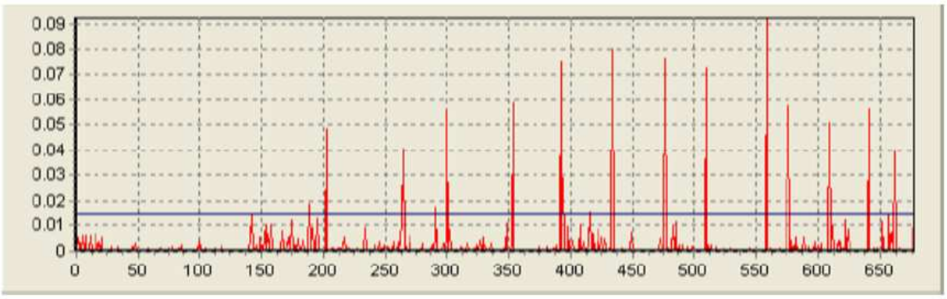

Fig.1.b. Cut detection using the derivative method ($\ddot{D}_{mean}()$ vector).
Isolated peaks correspond to cuts and the threshold is depicted with the blue line.

The cut detection threshold *t* was automatically determined. A two step method is proposed: in the first step the mean value *m* of the $D_{mean}()$ vector is computed and then the threshold is set as the mean of all the $D_{mean}()$ important peak values, greater than *m* (for the second method $\ddot{D}_{mean}()$ was used instead of $D_{mean}()$). Experimental results show a very good detection rate using the automatic threshold, as presented in Section 3. Also the use of the derivatives in the second algorithm drastically improved the detection.

The proposed fade detection algorithm is based on the algorithm proposed in [8] and uses the mean and variance of the pixels in the YCbCr color space. A ”fade in” causes a scene to emerge gradually on the screen from a monochromatic image (usually black). First, the monochromatic frames are detected by looking for frame where $\sigma^2(Y)$ is close to 0. Then the detection begins by analyzing the evolution of two statistical measures: $E\{Y\}$ and $C = |E\{Cb\} - E\{Cr\}|$. For a ”fade in” the two measures have an ascending evolution, as the global illumination grows up. The proposed parameter C is less sensitive to presence of motion (object/camera motion lead to global illumination variations) than the $E\{Y\}$ (see Fig. 2). The detection ends if one parameter remains constant or decreases for a certain period of time, or if the number of the analyzed frames exceeds the maximum fade length (set around 30 frames). The detection of ”fade out” follows the same algorithm but in the mirror (frames are analyzed in the negative direction of time). Experimental results are presented in Section 3. The thresholds used in this method are chosen experimentally.

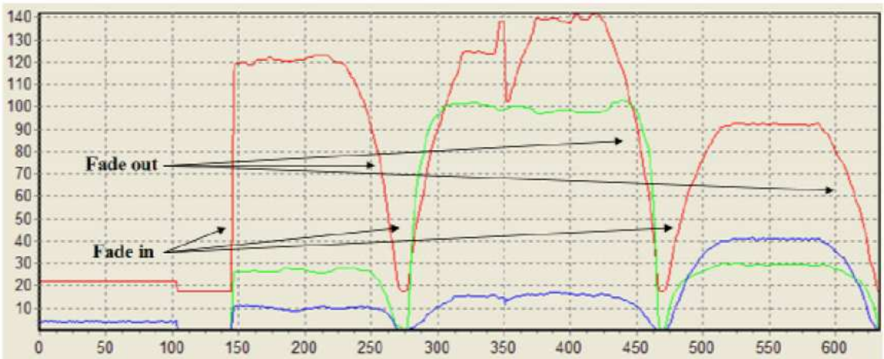

Fig.2. Fades detection: (normalized $\sigma^2(Y)$ blue line, $E\{Y\}$ red line, C green line, oX axis: frame index).

The SCC effects are specific to animation movies and are defined as a short-in-time color change of the current frame with the return to the initial state (see Fig. 3). As examples we can mention: thunders, lightening, explosions, apparition followed by the disappearance of objects, short global color effects etc. SCC effects are very short, less than 25 frames.

Fig.3. Short color change (examples, oX axis correspond to temporal axis)

The proposed detection algorithm is inspired from the flashlight detection in the natural movies (see [9]) and follows the effect producing process. A SCC starts with an important change in color and ends with almost the same frame as the starting one. The dissimilarity between frames is transformed in Euclidian distances between global color histograms. Instead of using an accurate but slow color reduction algorithm (as for cut detection), a fast uniform quantification of the RGB color cube into 125 colors is used (5x5x5). As for the cut detection, the video document is sub sampled by 2. The detection starts when $d_E(h_k(), h_{k+1}()) > t$, where $h_a()$ is the color histogram for the frame a, *t* the same threshold as for the cut detection and $d_E()$ is the Euclidian distance operator.

Distances between $h_k()$ and $h_{k+1+l}()$ ($l \in Z^*$, k fixed) are successively computed until the distance is lower than *t* (a frame is again similar to the one which initiated the detection) or the number of compared frames exceeds an imposed maximum effect length. The proposed algorithm achieved a detection error smaller than 12% (see Section 3).

2. Video abstraction and annotation

In order to quickly visualize the video content, a short and efficient visual representation is required. This kind of representation is called video abstraction or summary [12]. After the detection of video transitions and SCC effects, the data obtained are fusioned into shots using the following rules: all the cuts detected within the fades or SCC intervals are erased, all the shots smaller than n frames are erased (n < 5), the shots between a fade out and a fade in are erased (shots containing black frames), SCC does not determine a shot change.

Two video abstracts are proposed: a static and a dynamic one, which are automatically generated from the video shots. The static abstract is the ensemble of all the shots middle frames, as they are the most likely to be in the shot's action. It offers a fast global view of the movie content. For the dynamic abstract each shot is resumed by a sub sequence of p% frames centered in the middle of the shot. The dynamic abstract is the sequence computed as the direct concatenation of all the sub sequences with respect to time evolution. Using a percent of frames for each shot, more detail will be given for the longer shots than for the smaller ones, as they contain more information. Also, by removing unimportant frames (i.e. black frames between fades or transition frames) the obtained abstract will better represent the video content. Depending on p, the dynamic abstract summarize the movie action with minimal information loss in a more than 12 times shorter sequence.

In order to analyze the movie's rhythm, a video annotation is proposed. Video annotation is a tool for automatic or manual semantic interpretation of video documents and is indispensable to video indexing systems in order to access the video content (see [10]). The proposed video annotation consists of a visual graphic representation of the shots, video transition and SCC distribution and apparition frequency. The video transitions and SCC effects are represented by distinct shapes with the respect to their frame length (i.e. a cut is represented by a descending followed by an ascending slope, a SCC by a small peak). An annotation example is presented in Fig. 4.

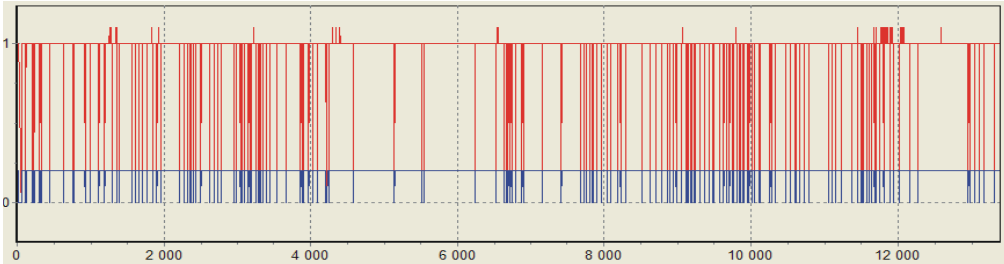

Fig.4.a. Video annotation: shots distribution (blue line), video transitions and SCC distribution (red line), the oX axis corresponds to frame index.

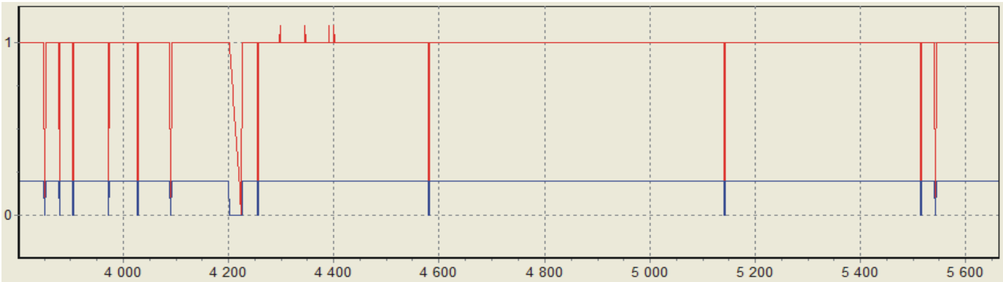

Fig.4.b. Video annotation detail, (each video transition is represented by a particular symbol)

The proposed video annotation is useful for the semantic analysis of the video document. Analyzing the shot densities one can detect regions of the movie with intense action or other type of action (i.e. in the Fig. 4 we can distinguish time intervals with a lot of shot changes, the dense vertical lines), or to detect character dialogs (camera alternatively focus on each character). Also the presence of certain transitions or SCC effects have different meanings, for example cuts increase the video's dynamism, a fades out followed by a fade in offer a time break on the action. The proposed annotation can be seen as a reverse engineering of the animation movies creation process, where shots are put together by the artist by following curtains rules and by using certain video transitions.

3. Experimental results

The proposed techniques were implemented on a dedicated software, called "Animation Movies Analysis Tool", which serve the purpose of the analysis of the animation movie database. The proposed cut detection algorithms have been tested on two different long animation movies (movie1 84min46s, movie2 73min18s, 3166 cuts). Results are presented in table 1. The proposed fades and SCC detection methods were tested on 14 different short animation movies (total time 101min47s, 37 ”fade in”, 56 ”fade out” and 120 SCC).

Results are presented in table 2. The following notations are adopted: GD - good detections, FD - false detections, “1” - 4histograms method, “2” - derivative method, E_D detection error, E_{FD} false detection error. Also the precision and recall are computed as:

$$precision = \frac{GD}{GD + FD}, recall = \frac{GD}{No.transitions} \quad (1)$$

Table 1.

Results of cut detection for the 2 test movies.

Movie	No.cuts	GD^1	FD^1	GD^2	FD^2
1	1597	**1391**	**89**	**1456**	**64**
2	1569	**1415**	**80**	**1475**	**63**

In order to estimate the detection errors all the transitions and SCC effects have been manually labeled using the implemented software. The undetectable video transitions (ex. mattes, dissolve) were treated as cuts. E_D is computed as the percentage of non detected transitions/effects (E_D = 1−recall) and E_{FD} is computed as the percentage of detected non existing transitions/effects.

The obtained global cut detection errors and the precision and the recall for the two test movies using the two proposed methods, are: E_D^1 =**11.4**%, E_{FD}^1 =**5.3**%, precision^1=**94.3**% and recall^1=**88.6**% and respectively E_D^2 =**7.4**% and E_{FD}^2 = **4**%, precision^2= **95.8**% and recall^2=**92.6**%.

Table 2.

Results of fades and SCC detection for the 14 test movies (fi=fade in, fo=fade out)

Fade in	GD	FD	Fade out	GD	FD	SCC	GD	FD
7	**7**	**0**	6	**6**	**1**	1	**1**	**0**
4	**4**	**1**	4	**4**	**1**	0	**0**	**0**
2	**2**	**1**	4	**4**	**0**	39	**38**	**2**
7	**7**	**1**	7	**7**	**0**	7	**6**	**0**
0	**0**	**0**	0	**0**	**0**	0	**0**	**0**
1	**1**	**0**	1	**0**	**0**	7	**5**	**0**
5	**5**	**0**	5	**5**	**0**	0	**0**	**1**
1	**0**	**1**	3	**3**	**0**	0	**0**	**0**
0	**0**	**0**	0	**0**	**0**	4	**4**	**0**
0	**0**	**1**	7	**6**	**0**	2	**1**	**0**
5	**5**	**3**	6	**6**	**3**	45	**40**	**0**
0	**0**	**0**	5	**5**	**0**	3	**2**	**0**
3	**3**	**0**	5	**5**	**0**	12	**9**	**5**
2	**2**	**0**	3	**2**	**0**	0	**0**	**0**

Using the derivative method an improvement of the detection error of 4% and of the false detection error of 1.3% was achieved. Also, the precision and recall are improved.

For the 14 test animation movies the obtained global fades and SCC detection errors, precision and the recall are:

- ”fade in”: E_D = **2.7%**, E_{FD} = **24.6%**, precision = **81.9%**, recall = **97.2%**
- ”fade out”: E_D = **5.7%**, E_{FD} = **9.4%**, precision = **91.4%**, recall = **94.6%**
- SCC: E_D = **11.7%**, E_{FD} = **6.7%**, precision = **93%**, recall = **88.3%**

Conclusions

In this paper an animation movies analysis system is proposed. First, the videos are divided into shots using improved cut and fade detection algorithms. A detection algorithm for a specific effect called SCC is also proposed. Using the video shots an automatic video abstraction and annotation are generated. They allow to briefly visualize the video content and to analyze the distribution of the video effects. Despite of the difficulties raised by the peculiarity of animation movies, the proposed video transitions/effects detection algorithms achieved a very low detection error. Futures improvements of the video transitions detection include the objects/camera movement analysis which leaded to almost all the false detections.

Acknowledgments

We thanks to CICA (Centre International du Cinéma d’Animation, “http://www.annecy.org”) and “Folimage” for providing us with the animation movies and for their support.

R E F E R E N C E S

1. *R. Lienhart*: Reliable Transition Detection in Videos: A Survey and Practitioner’s Guide. International Journal of Image and Graphics **1(3)**, 2001, pp 469–486.
2. *G. Lupatini, C. Saraceno, R. Leonardi*: Scene Change Detection: A Comparison. Research Issues in Data Engineering, Workshop on Continuous Media Databases and Applications, 1998, pp 34–41.
3. *M.S. Drew, Z.N. Li, X. Zhong*: Video dissolve and wipe detection via spatio-temporal images of chromatic histogram differences. Proc. IEEE International Conference on Image Processing **3**, 2000, pp 929–932.
4. *S.H. Kim, R.H. Park*: Robust video indexing for video sequences with complex brightness variation. Proc. IASTED International Conference on Signal and Image Processing, 2002, pp 410–414.

5. *A. Nagasaka, Y. Tanaka*: Automatic video indexing and full-video search for object appearances. Proc. of IFIP TC2/WG2.6 Second Working Conference on Visual Database Systems, 1991, pp 113–127.
6. *B. Shen*: HDH based compressed video cut detection. Proc. of Visual97 San Diego CA, 1997, pp 149–156.
7. VisiBone: "http://www.visibone.com/colorlab".
8. *W.A.C. Fernando, C.N. Canagarajah, D.R. Bull*: Fade and dissolve detection in uncompressed and compressed video sequences. IEEE International Comference on Image Processing, **3**, 1999, pp 299–303.
9. *W.J. Heng, K.N. Ngan*: Post shot boundary detection technique: flashlight scene determination. Proc. of the Fifth International Symposium on Signal Processing and Its Applications, **1**, 1999, pp 447–450.
10. *A.D. Bimbo*: Visual information retrieval. Morgan Kaufmann Publishers Inc. San Francisco, 1999, pp 202–259.
11. *N. D. Venkata, B.L. Evans, V. Monga*: Color Error Diffusion Halftoning. IEEE Signal Processing Magazine, **20(4)**, 2003, pp 51–58.
12. *Y. Wang, Z. Liu, J.C. Huang*: Multimedia content analysis using both audio and visual clues. IEEE Signal Processing Magazine, 2000, pp 12–36.