

Digital Video Automatic Segmentation Algorithms Using Edge Detection

Faten Hassan AL_Kathy*

Received on:2/6/2009

Accepted on: 16/2/2010

Abstract

With the increasing use of digital techniques in cinema, television and photography, there is a growing need for software capable of efficiently editing and compositing high quality digital images. This research captures video scenes after extracting frames from video file .Video scene change detection is a fundamental operation used in many multimedia applications and it must be performed prior to all other processes.

One goal of the work presented in this research is to develop technique for video scenes detection, by detecting the appearance of intensity edges in a frame that is at a fixed distance away from the intensity edges in the previous frame. The results of testing the program against abase line of different types of digital videos are represented graphically and analyzed to investigate how well the program can detect different filming techniques

خوارزميات التقطيع الاتوماتيكي للفيديو الرقمي باستخدام كشف الحافة

الخلاصة

مع الاستخدام المتزايد للتقنيات الرقمية في مجال السينما والتلفزيون والتصوير الفوتوغرافي أصبحت الحاجة ملحة لبرامج كفاءة عالية على تحرير وتركيب صور رقمية عالية الدقة. يهدف البحث الى دراسة الهيكلية الرقمية للفيديو الرقمي واستخلاص الصور (Frames) من ملف الفيديو الرقمي ثم تطوير تقنية لتحديد المشاهد الرقمية (Detection Scene) وذلك باستخدام خوارزمية تحديد كثافة الحافات في الإطار الحالي ومقارنتها مع كثافة الحافات في الإطار السابق وفق حساب النقاط الداخلة و الخارجة من صور الفيديو الرقمي عندها يتم تحديد موقع القطع (Cut Breaks) في الفيديو الرقمي. اثبت البحث نتائج جيدة من خلال اختباره على انواع مختلفة من الفيديو الرقمي.

Introduction

With the increasing use of digital techniques in cinema, television, photography, and the increasing use of digital images and movies in web design, entertainment, medical analysis, virtual environment creation, etc. The demand will continue to grow for fast and easy-to-use segmentation tools that accurately select, extract, measure,

and/or define objects of interest.

Recently, there has been a movement in computer vision from the processing of static images to the processing of video sequences. The requirements for efficiently accessing mass amounts of multimedia data are becoming more and more important [ZHA 94].

Video scene change detection is a fundamental operation used in many

*Department of Computer science, Kirkuk Technical Institute, Kirkuk

multimedia applications such as digital libraries and video on demand (VOD), and it must be performed prior to all other processes. Video data can be divided into different shots. A shot is a video sequence that consists of continuous video frames for one action. Scene change detection is an operation that divides video data into physical shots. Just as a text can be divided into sentences, a video sequences can be hierarchically divided into smaller components such as scenes and shot as shown in Figure (1.1) [SHA 95].

Typically the annotation of the information in the video has to be associated with shot transitions where a shot is defined as the sequences of successive frames from the moment a camera starts recording until it stop, (is a sequences of images taken from a single operation of the camera and that depicts a continuous action in time and space). Scene is a group of related shots (is defined as a collection of semantically related and temporally adjacent shots depicting and conveying a high-level concept or story) [SMO95].

While shot is the building block of video it is scene that conveys the semantic meaning of the video to the viewer. During the shot duration, both the camera and the objects in the image can move, but the sequences of frames is part of the same shot .therefore the first step in indexing digital video is the identification of shot transitions.

There are four different types of shot changes,cuts, fades, dissolves and wipes.

- A cut is an instantaneous transition from one scene to the next, and it occurs over two frames.
- A fade is a gradual transition between a scene and a constant image

(fade out) or between a constant image and a scene (fade in).

- A fade involves slowly changing the shot intensity to give the impression that a shot is appearing or disappearing.

- A dissolve is a gradual transition from one scene to another, in which the first scene fades out and the second fade in.

- A dissolve combines a fade-out and a fade-in as one shot vanishes another shot appears .In a dissolve for a short period of time both shots are super imposed.

- A wipe occurs as a line moves across the screen, with the new scene appearing behind the line.

Atypical wipe involves one shot replacing the other as a line crosses the screen. As one side of the line shows the old shot the other shows the new shot. As the line crosses the screen the old shot disappears. In a wipe, shots are not superimposed but occupy different areas of the screen.

One difficulty in automatic video segmentation stems from the variety of techniques used to create transitions between shots. The simplest of these is the Cut in which frames of one shot are immediately followed by frames of the next shot. A fade to a color (usually black)is a gradual transition of the frames of a shot that fades out to that color followed by the gradual transition of the frames of the shot that fades in form that color .In a dissolve the two effects(fade out and fade in)overlap [INT-16 02][INT-18 03].

The process of scene break detection is a fundamental component in automatic video indexing, editing and archiving .The detection and classification of scene breaks is a first step in the automatic annotation of digital video sequences. To facilitate browsing and retrieval of digital video, it is necessary to be able to detect logical scene breaks.

The edge detection algorithm is developed to detect the scene breaks in digital video sequence using the method of edge detection,

by detecting the appearance of intensity edges in a frame that is a fixed distance away from the intensity edges in the previous frame.

2. Digital Video Segmentation Algorithm

The detection of scene breaks is a first step in the automatic annotation of digital video sequences. The proposed algorithm for detection the scene breaks in video sequences depends on the detecting of the appearance of intensity edges in a frame that are a fixed distance away from the intensity edge in the previous frame this is done after segment video to frames (images) as shown in Figure (2.1). [INT-10 97]

The algorithm below describes the edge detection method developed to detect scene breaks (cuts) in digital video sequences:

Input: Video sequences file.

Output: Segment video sequences to scenes.

Step1: Open video file.

Step2: Play video file and segment video to frames (images).

Step3: Imaging process procedure contains:

Step 3.1: Removing color (gray scale images).

Step 3.2: Filtered images using Sobel filtering.

Step 3.3: Dilation images using Dilation filtering.

Step 4: Computing the edge change fraction procedure for all frames as follows:

Step 4.1: Compute exiting edge pixels (P_{out}) from frame to the next by comparing every pixel in first frame with dilated pixel in next frame.

Step 4.2: Compute entering edge pixels (P_{in}) from frame to the next by comparing every pixel in the

second frame with dilated first frame.

Step 4.3: replace the second frame with the third frame in the video sequences and the second frame is moved in the position of first, repeat step 4 until end of file.

Step 4.4: Compute the edge change fraction for every frame.

$$P = \max(p_{in}, p_{out})$$

Step 4.5: Save frame no. , max value and frame type (p_{in} , p_{out}) for every frame in a file ($max.file$).

Step 5: Detecting scene break (cuts) procedure contains:

Step 5.1: Compute difference between the edge change fractions (ecf) for every frame by subtracting the value of ecf for a frame from the last frame the result is saved in file ($diff.file$).

Step 5.2: By comparison the maximum value using $max.file$ and $diff.file$ according threshold, each of high peak represents a cut.

Step 6: Stop

3. The Method of Edge Detection

The scene breaks detection technique is based on detecting edges in two adjacent images and comparing them. By detecting the appearance of intensity edges in a frame that are far away from the intensity edges in the previous frame, it should be possible to detect scene breaks .This is achieved by taking a digital video sequence and selecting the first two frames as shown in Figure (3.1).The frames are rescaled if necessary and the color is removed, leaving two gray-scaled images as shown in Figure (3.2). These images are then filtered using the SOBEL filtering technique as shown in Figure (3.3).Other filters were tried but this one proved very efficient. Each intensity edge pixel is then dilated, by surrounding it with a diamond shape of pixels as shown in Figure (3.4) [UMB 98].

This is to allow for movement between frames when one frame is being compared against the next. An edge pixel in the second frame that appears far from an existing edge pixel in the first frame is defined as an ‘entering’ edge pixel. An edge pixel in the first frame that disappears far from an existing pixel in the second frame is defined as an ‘exiting’ edge pixel. By counting the number of entering and exiting edge pixels, the edge change fraction can be calculated. By analyzing the results of the edge change fraction for every frame, it is possible to detect cuts.

3.1 The Edge Change Fraction

By manipulating the edge change fraction for all frames in a video sequence, it is possible to detect scene changes. The edge change fraction, p , is the maximum value of edge change pixels that enter or exit a frame. P_{in} measures the fraction of entering edge pixels from one frame to the next. P_{out} measures the fraction of exiting edge pixels from one frame to the next. All frames have a value for P_{in} and P_{out} .

3.2 Computing the Edge Change Fraction

The method used to compute the edge change fraction, p , is to compare every pixel in the first undilated frame, E , against the corresponding pixels in the second dilated frame, E'_{dil} .

When comparing a pixel in the first frame against a pixel in the second frame, there are a few different scenarios that could be met:

- If a pixel is found in location (x,y) in the first frame E , and a matching pixel is found in location (x, y) or its dilated area $(x+\delta x, y+\delta y)$ in the second frame, E'_{dil} , then this implies that no change has occurred. Therefore, this is not an entering or exiting pixel.
- If a pixel is found in the first frame, E , and not in the second frame E'_{dil} , this implies that a pixel has exited from the first frame, E . A repeat of this procedure is carried out, where the second undilated frame, E' , is

compared against the first dilated frame, E_{dil} .

If a pixel is found in location (x, y) in the second frame E' , and a matching pixel is found in location (x, y) or its dilated area $(x+\delta x, y+\delta y)$ in the first frame, E_{dil} , then this implies that no change has occurred. Therefore, this is not an entering or exiting pixel.

If a pixel is found in the second frame, E' , and not in the first frame E_{dil} , this implies that a pixel has entered the second frame, E' .

By accounting for both scenarios, the number of exiting pixels, P_{out} is calculated for the first frame and, the number of entering pixels, P_{in} is calculated for the second frame.

$$P_{out} = \frac{\sum_{x,y} E[x+\delta x, y+\delta y]_{dil_E}[x, y]}{\sum_{x,y} E[x, y]} \quad ..(1)$$

$$P_{in} = \frac{\sum_{x,y} dil_E[x + \delta x, y + \delta y]_{E'}[x, y]}{\sum_{x,y} E[x + \delta x, y + \delta y]} \quad ..(2)$$

This process occurs in the same loop in which the image processing is carried out. At the end of the loop, the third frame in the video sequence is taken to replace the second frame, and the second frame is moved into the position of the first frame. The whole process of imaging and calculating the edge change fraction is repeated on these two frames. This implies that P_{out} is now calculated for the second frame and P_{in} for the third frame. The edge change fraction, p is the maximum value of P_{in} and P_{out} in each frame [LAW 01]. Scene breaks can be determined by looking for peaks in p , the edge change fraction:

$$P = \max(p_{in}, p_{out}) \quad \dots (3)$$

$$S = \sqrt{\sum_i^n (x_i - \bar{x})^2 / n} \quad \dots (4)$$

3.3 Analysis of Edge Change Fraction

A file containing an array with each frame number and frame type, whether it be exiting or entering as well as the edge change fraction value, p , for each frame, is the resultant from the first stages of the process to detect scene changes. A scene break can be detected by looking for peaks in the edge change fraction p . However, a more effective and easier method for analysis is used by calculating difference values from the array of edge change fraction values. The difference between the edge change fraction values for each frame is calculated by subtracting the value of one frame from the last and the results are written to an array in a file. This centers the edge change fraction around zero, if a cut occurs, it should have a positive and negative peak, whereas if there is extra content it will not necessarily have the same symmetric characteristic and therefore it is easier to detect using the thresholding scheme. A threshold value is initially set and if the difference value is found to cross the threshold in either the positive or negative direction, a cut is considered to have occurred.

It is not sufficient that one standard threshold value be used across the wide range of different video types. Clearly, using a set of thresholds would detect false cuts, also known as false positives, and miss other real cuts. It is for this reason that a statistical method is used to make the threshold adaptive. The adaptive thresholding technique used is designed by calculating the standard deviation for every frame. An analogy of a thresholding method could be of moving a sliding window across all the difference values for each of the frames in a video sequence.

The sliding window effect is implemented by calculating the standard deviation, σ , for each frame. The difference value is calculated for the mean of a frame and its five previous and following frames, \bar{x} , where x is the value of each of the individual difference values and, n , is the number of difference values in which the standard deviation is taken over.

Now that an optimum adaptive thresholding scheme is in place, if the difference value for a frame is found to cross its threshold level, a cut is considered to have occurred. Every time this happens in a video sequence, the frame number is written to a file. A cut is detected when an edge change fraction value is found to cross the upper threshold, or if the edge change fraction value increases from zero to above the lower threshold, i.e. a cut from blank screen to a new image. The standard deviation is calculated in the code as follows:

```

/* allows for first and last 5 frames
*/
for (z=0; z<5; z++)
stand_dev[z] = 0.1;
for (z= (end_frame- start_frame -
5); z<end_frame; z++)
stand_dev[z] = 0.0;
for (z=5; z< (end_frame-
start_frame-5); z++) {
around_pt [0] = diff [z-5];
:
around_pt [5] = diff [z];
:
around_pt [10] = diff [z+5];
for (i= 0; i<11; i++)
M_SUM+= around_pt[i];

```

```

Mean = M_SUM / 11;
for (i=0; i<11; i++)
SD_SUM += pow ((around_pt[i] -
mean),2);
Stand_dev[z] = sqrt (SD_SUM/11);
M_SUM=0.0;
SD_SUM=0.0 ;}

```

4. Conclusions

Scene change detection is the most commonly used method to segment image sequences into coherent units for video indexing .The proposed research developed to detect shot boundaries in digital video sequences using the method of edge detection. The basic method is to measure the pixel differences frame to frame in term of intensity or color. The number of changed pixels is counted and if the number exceeds a certain percentage a scene cut is detected. The edge detection algorithm is implemented to detect shot cuts in digital video sequence by detecting the appearance of intensity edges in a frame that are at a fixed distance away from the intensity edges in the previous frame. At running time the video is segmented to set of scenes as shown in Figure (4.1), all video scenes code are viewed on the screen, and when the cursor moves in one scene code, the frames of that scene are displayed as a montage as shown in Figure(4.2) and Figure (4.3). The detection results for the videos are very good due to the fact that most scene changes are direct scene cuts between unrelated video shots. Knowledge about scene breaks can be used to look for higher-level structures (such as a sequence of cuts between cameras) or to ensure that key frames or representative frames come from different scenes.

The results of testing the program against different videos show how well the algorithm can detect cuts in, by take-in a video sequence and detect the frame number where a shot break occurs.

References

[1]. Books and Studies

[ZHA 94] Zhang H. and Smoliar S. W., "Developing Power Tools for Video Indexing and Retrieval," in Proc. SPIE'94, Storage and Retrieval for Image and video Databases II, 1994.

[UMB 98] Umbaugh, Scott E Umbaugh,"Computer Vision Image Processing", by Prentice Hall, 1998.

[LAW 01] Lawrence S., Ziou D. and Wang S. "Motion Insensitive Detection of Cut and Gradual Transitions in Digital Video", International Conference on Multimedia Modeling, Ottawa, 2001

[SHA 95] Shahraray B., "Scene Change Detection and Content-Based Sampling of Video Sequences," in Proc. SPIE'95, Digital Video Compression: Algorithm and Technologies, 1995.

[SMO 95] Smoliar S. and Zhang I., "Content-Based Video Indexing and Retrieval ", IEEE Multimedia, Vol. 2,pp62- 72,1995.

[2]-Internet References:

[INT-10 97] "OpenDML AVI File Format Extensions ", 1997.

<http://www.matrox.com/video/press/paper/odmiff2.pdf>

[INT-18 03] "Video Analysis of Human Dynamics—a survey", Department of Computer Science, University of Exeter, 2003.

<http://dsc.ex.ac.uk/research/pann/pdf/pann-ss-088.pdf>

[INT-16 02] "Digital Editing and Montage:The Vanishing Celluloid and Beyond", 2002.

<http://erudite.org/revue/cine/2002/v13/n1/007957ar.html>

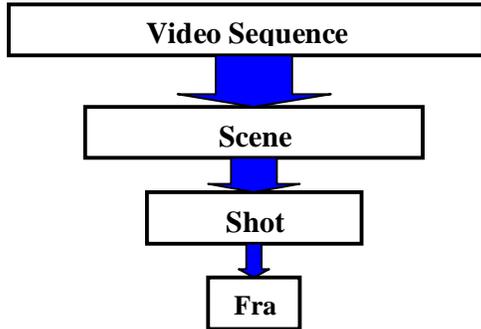


Figure (1.1) Structural view of a video sequences



Figure (2.1) segment video to frames



Figure (3.2) Gray-scaled images of two consecutive frames



Figure (3.1) Images of two consecutive frames

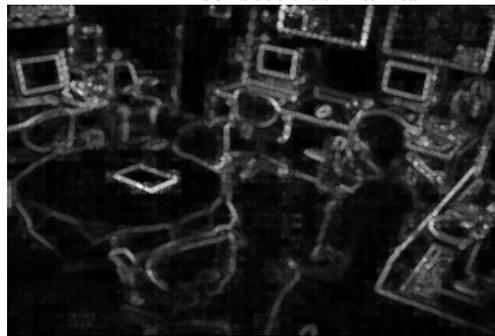


Figure (3.3) Filtered Gray-scaled images, resulting in intensity edges

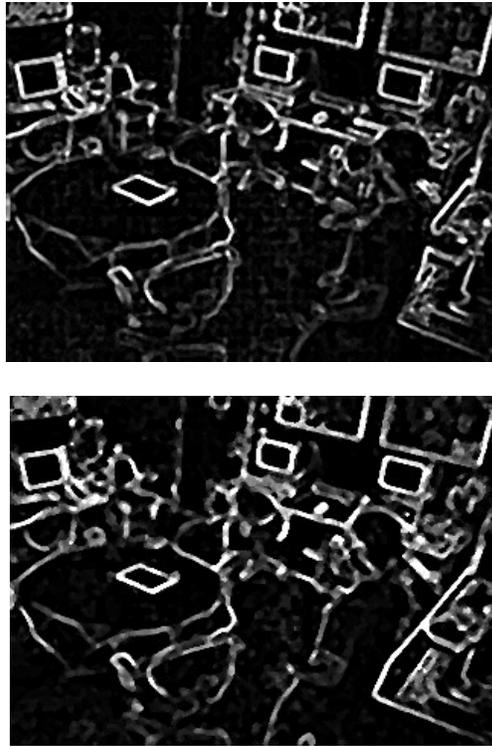


Figure (3.4) Dilated intensity edges in two consecutive frame



Figure (4.2) Extract frames interface

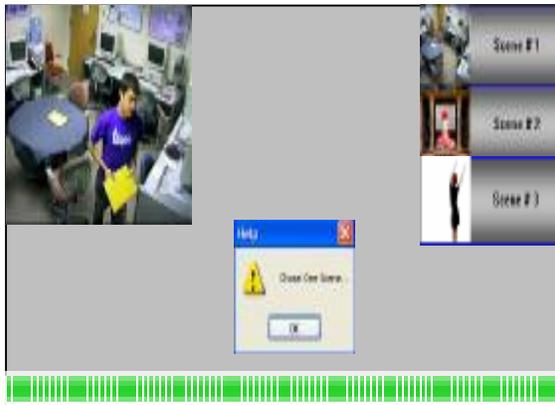


Figure (4.1) Extract scenes from the video



Figure (4.3) Frames selection interface for two Scenes

