Video Summarization Using Adaptive Shot Detection and Statistical Aproach to Estimate the Motion

Wafae Sabbar[1, 2], Adil Chergui[1], Abdelkrim Bekkhoucha[1]
[1]Computer Lab. Lim@II FSTM
Mohammedia
Morocco
[2]FSJES Ain Sebaa
Casablanca. Morocco
wafae.sabbar@univh2m.ac.ma

ABSTRACT: *The video summarization is the process to present a rapid view of the important video scenes. It is a necessary step to create an efficient system of indexing and retrieve video. Mainly summarization methods apply a clustering algorithm, which use the dissimilarity matrix calculated between all video frames to extract the key frames which imply a quadratic calculation. To reduce this complexity, we propose a hierarchical approach to extract video summary based on shot segmentation and motion estimation. In the first step, we use an adaptive detection of shot transitions to segment the video in shots. In the next step, we apply a hierarchical clustering in each detected shots to extract the keyframes; the number of these keyframes is adaptive to the motion in the shot. We propose a statistical approach using a co-occurrence matrix to estimate this motion. To validate the effectiveness of the proposed approach, we present some experiment results based on real video.*

Keywords: Shot Segmentation, Key Frames, Shot Transitions, Histogram Bi-dimensional, Co-occurrence Matrix, Motion Estimation

Received: 19 June 2012, Revised 31 July 2012, Accepted 8 August 2012

1. Introduction

The use of digital videos by professionals from various fields and the general public is growing rapidly, it is important to develop efficient systems for searching, browsing and indexing videos. The video summarization is a necessary step to construct these systems; it is the process to generate a short and fast view on the important scenes and give the users an appropriate idea of the whole video in real time.

In the literature, we find two video summarization types [1] [2]: Static video summarization and dynamic video skimming. The static summarization consists to select the representative video frames, these frames called key- frames. It is similar to extract the keywords from text document; these keyframes provide the user an overview of video directly and efficiently. However, video skimming is a short video interpretation with audio information; it preserves the dynamic properties of the original video. The video skimming is further attractive by users, since it provides more information about motion and semantic. However, the time required for viewing a skimming video is not appropriate for a quick browsing in a video database [16]. In this work we interest on static summarization methods, which are classified into three families: methods based on the sampling, methods based on the segmentation in scenes and methods based on shots segmentation.

The methods are based on sampling choose keyframes uniformly or randomly at certain intervals of time, without considering the video content. The summary produced by these methods does not represent the all video parts, and may cause some redundancy of keyframes with similar contents [3] [4]. The methods based on video scene segmentation consist to extract keyframes using scenes detection, the scene include all parts with a semantic link in the video or in the same space or in the same time [5]. The disadvantage of these techniques is producing a summary, who does not take into account the temporal position of frames. In other hand, the scene segmentation is based on clustering algorithms which require the calculation of a dissimilarity matrix, the quadratic calculation of this matrix cause problem of time computation when the video has a considerable size. The methods based on shots segmentation extract adapted keyframes to video content, they extract the first image as shot keyframes [6] [7] or the first and the last frames of the shot [8]. These methods are effective for stationary shot and small content variation, but they don't provide an adequate representation for shot with strong motions. Other methods use various information such as sound, text or manual annotation [9].

In this work, we propose a video summarization based on shot segmentation. In the first time, we introduce a bi-dimensional histogram *H* 2 *D* to characterize video frames. We calculate the dissimilarity between successive frames using this histogram. This calculation produces one dimensional signal, in which we identify the shot transitions. The shot transitions are characterized by strong variation between successive frames, thus we apply an adaptive thresholding to detect these transitions. Next time, we use an agglomerative hierarchical clustering in the detected shots to extract the key frames; the number of these key frames is adaptive to the motion in the shot. Thus, we propose a new statistical approach using a co-occurrence matrix to estimate this motion.

The rest of this paper is organized as follows: in the next section we describe our video shot detection and segmentation based on bi-dimensional histogram *H* 2 *D*. In section 3, we present the new method for motion estimation by co-occurrence matrix; and in section 4, we apply a hierarchical clustering to extract the video summary. The applications and the results are presented in section 4. Finally, we present the conclusion and future works.

2. Adaptive Shot Segmentation Using a Bi-dimensional Histogramm

2.1 Shot detection

The objective of video shot segmentation is to divide the video into meaningful parts called shots, a shot represent the basic element in video; it records the frames resulting from a single and continuous running of camera, from the instant it is turned on to the instant it is turned off [16]. We can detect video shot by detecting the transition between shots; this detection is considered as the reverse process of video editing.

Different techniques for shots detection are proposed: techniques based on color histograms comparison [10] or comparison pixel to pixel of the successive frames [11], others are based on the motion [12]. These methods have advantages and disadvantages, methods based on color histograms comparison are not able to make the difference between two images with similar histogram but different in content [15]. Methods based on pixel comparison are very sensitive to noises and motion in frames, but they are fast in calculation time. Although, the effectiveness of methods based on motion, the calculation time still more expensive [12]. To make a compromise between the gains and the weaknesses of these methods, we use a bi-dimensional color histogram *H2D*. It is computed in real time, allow the spatial information contained in the frames and the difference between two frames with similar classic histogram and different contents.

2.2 Bi-dimensional Histogram H2D

The color is the most visual feature widely used to represent video frames and is often described by a color histogram, the classic histogram is not able to detect the dissimilarity between two images with similar histogram but different in content (*figure* 1).

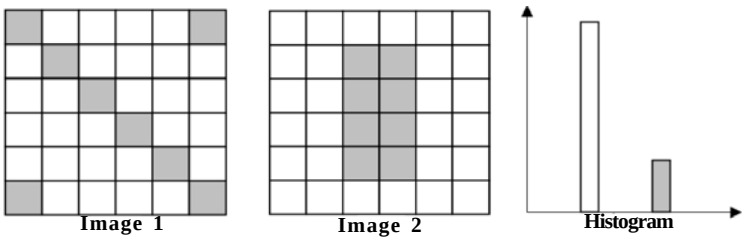

Figure 1. Two Different Images with Similar Histogram

To make a compromise between the gains and the weaknesses of the classic histogram, we introduce a bi-dimensional color histogram $H2D$. It's computed in real time, allow the spatial information contained in the frames and the difference between two different frames with similar classic histogram.

Let $I(i, j)$ the color of the pixel (i, j) and $V(i, j)$ the squared neighborhood of size: 3×3 (or 5×5, 7×7, ...), the proposed $H2D$ histogram is calculated using equation (1):

$$H2D(c, s) = \sum_{(i,j) \in image} \delta(s, \text{Sij})\, \delta(c - I(i, j)) \quad (1)$$

Where c is a given color and s is a parameter used to test if the pixel (i, j) belong to a uniform region, s depend on the size of the chosen neighborhood. The term S_{ij} is the number of pixel in $V(i, j)$ who have a color identical to the pixel (i, j), it's calculated by the following formula:

$$S_{ij} = \sum_{(k,l) \in V(i,j)} \delta(I(i, j), I(k, l)) \quad (2)$$

δ it's the Kronecker operator:

$$\begin{cases} \delta(c_1 - c_2) = 1 & si \quad c_1 = c_2 \\ \delta(c_1 - c_2) = 1 & si \quad c_1 \neq c_2 \end{cases} \quad (3)$$

To reduce the histogram size and introduce the fuzzy and imprecision that characterize the pixel values; we integrate the fuzzy logic. The basic idea is define the term S_{ij} which eliminate the abrupt transition between colors. Thus, the term S_{ij} is the number of pixels in $V(i, j)$ haven a color near to (i, j), we use the Hafner model [17] to measure the membership degree between color given by the function μ:

$$\mu_c(c') = \exp(-\sigma(\frac{d(c, c')}{d_{max}})^2) \quad (4)$$

d_{max} is the maximal distance between two element in color set and σ is the confusion degree between colors.

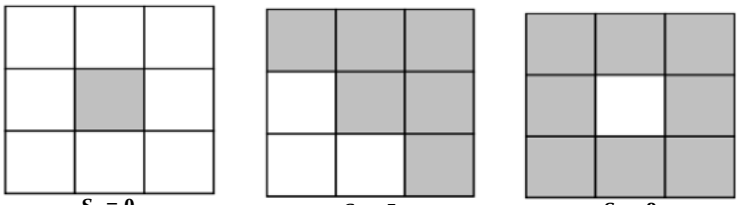

Figure 2. Example of S_{ij} calculation in neighborhood 3×3

2.3 Adaptive Shot Detection

To detect video shots, we seek the transition between consecutive frames. Therefore, we calculate the dissimilarity between successive frames i and $i+1$ using $H2D$ histogram to characterize frames. We use Manhattan distance to calculate this dissimilarity given by following equation:

$$dissmilarity(H2D_i, H2D_{i+1}) = \sum_{c=1}^{c=n} \sum_{s=1}^{s=N} |H2D_i(c, s) - H2D_{i+1}(c, s)| \quad (5)$$

n and N are respectively the number of color and the size of choosing neighborhood. This calculation produce one dimensional signal $S(i)$, in which we identify the shot transitions:

$$S(i) = dissmilarity(H2D_i, H2D_{i+1})_{0 \leq i \leq frames\_number} \quad (6)$$

The shot transitions are characterized by strong variation of the signal $S(i)$, thus we apply a thresholding to detect these transitions. Many thresholding methods use a fixed threshold [18] [19] [20] [21], it's difficult to preset a fixed threshold because the videos have different nature. Other approaches define dynamic thresholds depending to frames content [24] [25] [26], other choose a machine learning algorithm to discern the transition [22] [23], but this approach requires an adequate base for learning [22] [23].

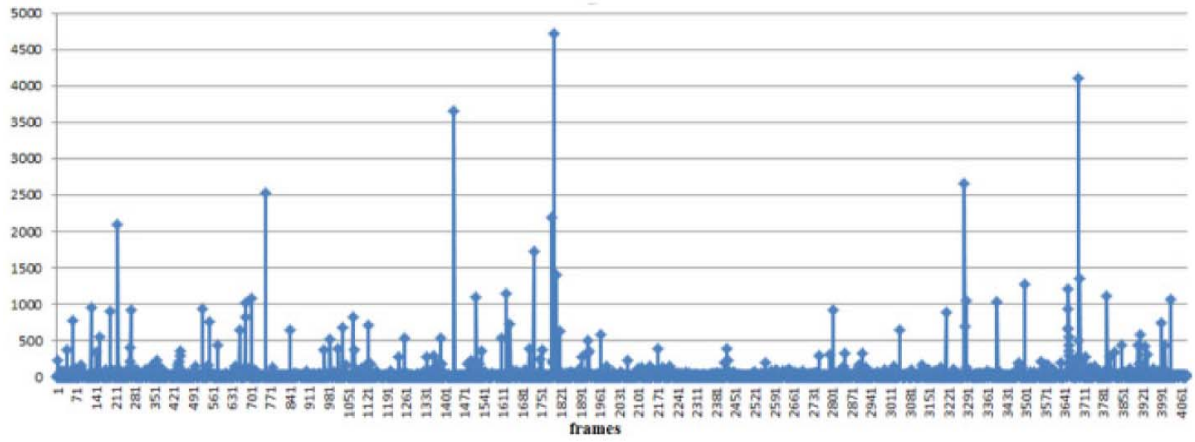

Figure 3. Example of signal S(*i*) for consecutive video frames

In this work, we introduce an adaptive threshold *T* by analyzing the evolution of signal *S*(*i*). In the first step, we calculate the average avg_1 and the absolute deviation σ_1 of the *S*(*i*):

$$avg_1 = \frac{1}{n_s}\sum_{i=1}^{n_s} S(t) \qquad \sigma_1 = \frac{1}{n_s}\sum_{i=1}^{n_s} |S(t) - avg_1| \tag{7}$$

Naturally in the video, the number of shot transitions is constantly very small to the number of similar frames, so the value ($avg_1 + \sigma_1$) are very small to threshold *S*(*i*). For that reason, in the second step we determine the local maxima *LM* (*i*) of signal *S*(*i*) defined by:

$$LM(i)_i = S(i)_i / S(i) \geq avg_1 + \sigma_1 \tag{8}$$

We calculate the average avg_2 and the absolute deviation σ_2 of *LM* (*i*), then the adaptive threshold *T* is defined by:

$$T = avg_2 + \sigma_2 \tag{9}$$

The shot transitions are detected between the two frames *i* and *i* + 1, with dissimilarity superior to the calculated threshold *T*, we use this detection to segment the video in shots.

3. Motion Estimation by Co-occurrence Matrix

The important problem posed in video summarization is: "*what type of frames should be selected as keyframes in order to capture the significant video content* ". Various works in literature suggest that the frames with minimal motion should be selected because cameras often focus on important objects or places for a relatively long period. Others regarded the frames with maximal motion as keyframes because if these frames are not captured, some information will be lost. In this section, we present a general idea of motion estimation methods and we describe the proposed motion estimation method to extract keyframes from video shots. The idea is to calculate dissimilarity between shot frames by taking into account the large motion and find de representatives frames in each shot, so we propose a new statistical approach using a co-occurrence matrix to characterize this motion.

3.1 Motion estimation

The motion estimation in video sequence is one of fundamental problems in image processing; it is a process which studying the objects movies in video, and it seeks the correlation between two successive frames for predicting the position change of the content. Generally, this motion is represented by a motion vector describing the transformation between two frames. Jeannin [27] indicate that the motion is a pertinent information and can be extracted from camera motion, object's motion, or from both combinations. In general, all the works which adopt this approach describe the motion in video sequence by pixel differences in

frames [28] or by computing the optical flow [29] [30] [31] or by block matching [32]: the methods based on pixel differences are very sensible to noise and characterized by important calculation time.

The methods using the optical flow are based on the spatial and temporal gradients of the pixels luminance intensity. The principle of these methods is founded on the strong hypothesis: the conservation of the pixel luminance intensity along the motion track, this conservation hypothesis can be written as:

$$\frac{d}{dt}[I(x_1, x_2, t)] = 0 \tag{10}$$

Where x_1 and x_2 are the spatial variables and $I(x_1, x_2, t)$ is the intensity of the pixel (x_1, x_2) in the frame acquired at the time t, the first order Taylor's expansion of this equation provides the following formula:

$$\frac{\partial I(x_1, x_2, t)}{\partial x_1} dx_1 + \frac{\partial I(x_1, x_2, t)}{\partial x_2} dx_2 + \frac{\partial I(x_1, x_2, t)}{\partial t} dt = 0 \tag{11}$$

Where dx_1 and dx_2 are the shifting of current pixel to be estimated and dt is the temporal period for acquiring frames.

If we divide the equation (11) by dt, then we obtain the constraint equation of motion, also called optical flow equation [32]. It is therefore one equation with two unknowns, to solve this equation different methods introduce additional constraints [32] [33] [34] [35].

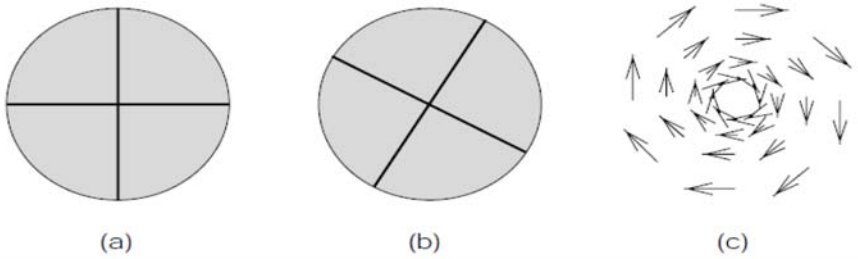

Figure 4. Illustration of optical flow: (a) sphere at time t, (b) sphere at time $t + 1$, (c) optical flow

The block-matching is a method used to estimate the motion objects in a video, it is performed between two frames, and used in video compression standards such as H.261 [36], MPEG-1, MPEG-2 or MPEG-4. [37]. A frame is partitioned into equal size blocks; the purpose of the block matching is to search in the second frame the blocks correspond to the blocks of the first frames. There are different methods proposed of block-matching depending to the computation time, the estimate quality or a compromise between the two [38] [39] [40] [41].

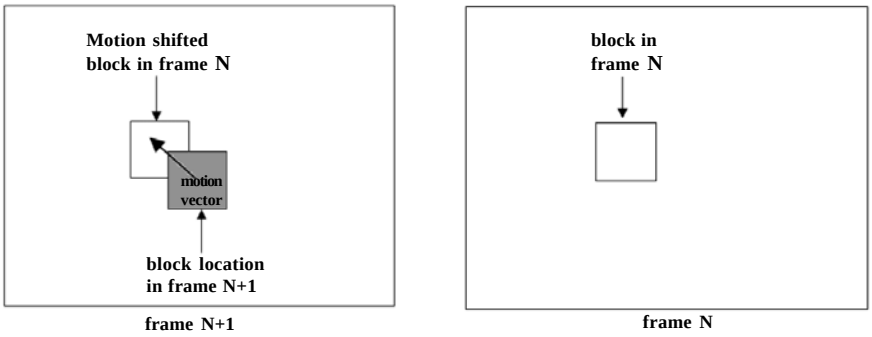

Figure 5. Illustration of the block matching for two consecutive frames

3.2 Statistical Motion Estimation by Co-occurrence Matrix

In this section, we propose a new statistical approach using a co-occurrence matrix to estimate the motion in video shots. Methods working at the pixel level are very sensible to noise and are often characterized by important calculation time. Thus, we reduce the representation space and we use a low resolution frames. We divide each frame into blocks of $n \times n$ pixels; a block will be represented by the dominant color (*Figure* 6). All pixels in a small block have the identical motion; this is due to the