"AUTOMATIC VIDEO SHOT BOUNDARY DETECTION AND KEY FRAME EXTRACTION USING GABOR MOMENTS"

PROJECT REFERENCE NO. : 37S0388

COLLEGE : VIVEKANANDA COLLEGE OF ENGINEERING & TECHNOLOGY, PUTTUR
BRANCH : COMPUTER SCIENCE AND ENGINEERING
GUIDE : HARIVINOD N
STUDENTS : ALLAN MERWIN MENEZES
ASHLESH M S
KAUSHIK
SHARAN RAI A

Keywords: Shot boundary detection, Video segmentation, Gabor moments

Introduction:

A video consists of sequences of images and often being called as frames, which can be played consecutively at the speed of around 20 to 30 frames per second in order to view smooth motion. Shot boundary detection is the first step towards automatic annotation of digital video sequence and video summarization. Its goal is to divide the video stream into a set of meaningful and manageable segments called shots. The shots are smaller meaningful units and are got from video based on temporal discontinuities in the video sequence. More precisely, shots are defined as an unbroken sequence of frames captured by one camera in a single continuous action in time and space. Normally, it is a group of frames that have constant visual attributes, such as color, texture, and motion. In the next level, each shot is then represented by selecting key frames and indexed by extracting spatial and temporal features. So, the video shot detection and key frame extraction provides a basis for video segmentation and abstraction methods.
Objectives:

Given a video consisting of number of shots, the objective is to find the cut transition i.e. to identify the end of the each shot and the beginning of the next shot automatically without manual intervention. This problem is also known as ‘video shot boundary detection’ or ‘transition detection’. To address this issue, proper features have to be extracted from given video which forms a feature vector. This feature vector has to be computed for each frame. By comparing the feature vectors of adjacent frames using the Euclidean distance, cut is detected and shot boundary is identified. Finally a shot is represented by selecting appropriate key frames.

Methodology:

The proposed approach uses video shot boundary detection method based on Gabor Moments. The block diagram of the overall system is given in Figure 1. The major modules are shot boundary detection and key frame extraction. Further, the sub-modules under shot boundary detection are computation of Gabor moments, discontinuity computation, shot boundary detection from discontinuity measure.

![Figure 1: Block diagram of the overall system](image)

Computation of Gabor Moment

The general form of an even symmetric Gabor filter in the spatial domain is given by the following expression:

$$G_{\theta,f} = \exp\left(-\frac{1}{2} \left[\frac{p^2}{\sigma_p^2} - \frac{q^2}{\sigma_q^2}\right]\right) \cos(2\pi fp_\theta)$$
where \( p_\theta = x \sin \theta + y \cos \theta \), \( q_\theta = x \cos \theta - y \sin \theta \),

Here \( f \) is the frequency of the sinusoidal plane wave at an angle \( \theta \) (six different orientation values are considered for \( \theta \)) with the x-axis, \( \sigma_p \) and \( \sigma_\theta \) represent the standard deviations of the Gaussian envelope along the x-axis and y-axis respectively which gives the width of the Gabor filter.

The Gabor Moment is computed as follows:

- Obtain the Gabor filtered images, i.e. compute \( Q_{\theta_i,f}(I) \) where \( i = 1 \) to 6
- For every \( Q_{\theta_i,f}(I) \), compute the first and second moment which represent the mean and standard deviation of the generated filter image. i.e. calculate \( Z_{i,1} \) and \( Z_{i,2} \) defined by,

\[
Z_{i,1} = \frac{1}{N} \sum_{j=1}^{N} p_{i,j}, \quad Z_{i,2} = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (p_{i,j} - Z_{i,j})^2}
\]

Here \( Z_{i,1} \) and \( Z_{i,2} \) are the first and second moment for the \( i^{th} \) component (corresponding to orientation \( i \)), \( N \) is the total number of pixels in the frame, and \( p_{i,j} \) represents the intensity value of the pixel at location \( j \) for the filtered image with orientation \( i \). After performing the above two steps, we obtain 12 dimensional feature vector for every frame. So the feature vector for the frame \( n \) is represented as: \( V_n = \{ Z_{i,1}^n, Z_{i,2}^n \} \), where \( i = 1 \) to 6.

**Discontinuity computation:** The frame difference is calculated by computing the absolute difference between two successive frames using the following equation.

\[
D(V_n, V_{n-1}) = \sum_{k=1}^{2} \sum_{i=1}^{6} |Z_{i,k}^n - Z_{i,k}^{n-1}|
\]

It shall be observed that the dimension of the feature vector is 12, which is independent of the image size and hence matching can be performed faster.

**Shot boundary detection from discontinuity measure:** The process of shot boundary detection will detect the regions that contain different data. The threshold is considered one of the most important methods used for detection. It has two objectives. The first is to stop false shot boundaries and the second is to find gradual transitions. Here we use local adaptive threshold to detect the shot. If the calculation of value of threshold employs statistical data that are retrieved from the input data along with a small segment, it will be considered as adaptive data. Thus value of the local threshold is used to cover a short segment of an input sequence. The local threshold starts its process with the smaller data. A number of
experiments have shown that the size of the window can reflect the accuracy of the detection performance.

**Key Frame Extraction:** There are great redundancies among the frames in the same shot; therefore, certain frames that best reflect the shot contents are selected as key frames to succinctly represent the shot. The extracted key frames should contain as much salient content of the shot as possible and avoid as much redundancy as possible. One of the approaches for key frame extraction is to represent a shot by selecting the middle frame of the shot as key frame. So if there are $n$ shots in the video then there will $n$ key frames using this approach.

**Results and Conclusions:**

The experiments are conducted on TRECVID video database. Here the ground truths are available, which gives the actual cuts that can be used for computing performance measures. Figure 2 shows the adjacent frames where the shout boundary is detected.

**Figure 2:** Pair of consecutive frames of cuts for the Lecture Series (Senses111) video

The comparison between an algorithm’s output and the ground truth is computed based on the numbers of missed detections and false alarms, which is expressed as recall and precision. The detection rate is often tested by recall (R), precision (P) and F1-measure (F1). A compromise between recall and precision is obtained by F1 which combines the recall and precision. In shot boundary detection, recall rate is defined as the percentage of desired boundaries that are detected, and precision rate is defined as the percentage of detected boundaries that are desired boundaries.

$$Recall = \frac{No. \text{ of correct cuts detected}}{No. \text{ of actual cuts}}$$
\[
\text{precision} = \frac{\text{No. of correct cuts detected}}{\text{No. of correct cuts detected} + \text{No. of false cuts detected}}
\]

\[
F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{precision}}
\]

Table 1 shows the results of the experiments. We also compare results of the proposed method with existing methods. We see that the F1 measure is comparatively good in the proposed method.

<table>
<thead>
<tr>
<th>Name of Video Segment</th>
<th>Metrics</th>
<th>Proposed Model</th>
<th>Histogram based Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animation (BOR05)</td>
<td>Precision</td>
<td>0.944</td>
<td>0.933</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.887</td>
<td>0.848</td>
</tr>
<tr>
<td></td>
<td>F1 Measure</td>
<td>0.915</td>
<td>0.888</td>
</tr>
<tr>
<td>Central Valley Project (BOR08)</td>
<td>Precision</td>
<td>0.959</td>
<td>0.937</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.926</td>
<td>0.942</td>
</tr>
<tr>
<td></td>
<td>F1 Measure</td>
<td>0.942</td>
<td>0.939</td>
</tr>
<tr>
<td>Lecture Series (Senses111)</td>
<td>Precision</td>
<td>0.857</td>
<td>0.881</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.993</td>
<td>0.996</td>
</tr>
<tr>
<td></td>
<td>F1 Measure</td>
<td>0.920</td>
<td>0.935</td>
</tr>
</tbody>
</table>

**Scope for future work:**

In the future this work might be extended to detect gradual transition which is comparatively difficult. The gradual transition is more difficult to detect than abrupt transition owing to the different types of gradual transition such as fade, dissolve, and wipe where the changes can be stretched over more than one frame. For hard cut detection dissimilarity measures are calculated using Euclidean distance, hard cut is detected when dissimilarity measures has peak values and it is compared against a threshold, but in case of gradual transition this
process goes through number of frames. It can be achieved by using two threshold values. In a gradual transition the intersection of two consecutive frames increases gradually while the difference decreases steadily. Further work includes indexing and retrieval of the video through the key frames.