The LaTIn Median Frame Shot Boundary Detector

Thiago T. Santos
Institute of Mathematics and Statistics
University of Sao Paulo
Rua do Matao, 1010
Sao Paulo, Brazil 05508-090
thsant@ime.usp.br

Carlos H. Morimoto
Institute of Mathematics and Statistics
University of Sao Paulo
Rua do Matao, 1010
Sao Paulo, Brazil 05508-090
hitoshi@ime.usp.br

ABSTRACT

This paper provides a compact overview of LaTIn **shot boundary detection task** system and results (LaTIn-USP MD rev107) for the TRECVID 2006. The run is an **image processing** approach that has **2 steps**: an event detector that performs **absolute pixel difference** between each frame and a **median frame** followed by a shot boundary detection step based on **histogram intersection difference** between each event frame and the shot median frame.

The approach depends on a small parameters set but tuning is not trivial. So, there is a clear performance difference between American and Asian videos. However, the median frame is an interesting way to evaluate shot activity and to represent visual content in a short time range.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Time-varying imagery*

General Terms

Experimentation

1. INTRODUCTION

This work is a simple shot boundary detector based on common image processing operations. We use a *median image* to represent the shot in a time point and to judge if a frame is a possible boundary, comparing it against that reference.

The algorithm has two steps. First, shot-boundary candidates are chosen based on a fast comparison with the previous frames. The second step is a refinement that evaluate two hypothesis: if we have two shots in a time window or if a single shot is a more reasonable choice.

2. EVENT DETECTION

The first step is *event detection*. Here, we are interested in *any* type of large discontinuity or activity in the frames' pixels.

Let $V = \langle f_1, f_2, f_n \rangle$ be a sequence of frames f_i. Each frame is a gray scale image, a function $f_i : [0, X] \times [0, Y] \rightarrow [0, 255]$, where X and Y are the width and the height of video frames. The *median frame* of a time slice $[i, j]$ is given by

$$\bar{f}_{i,j}(x, y) = \text{median}(f_i(x, y), f_{i+1}(x, y), ..., f_j(x, y)).$$

The median frame will be a representation of the shot content around a time point. To estimate the discontinuity at frame f_j, we use the *absolute frame difference* between f_j and $\bar{f}_{i,j-1}$

$$d_{f_a,f_b}(x, y) = |f_a(x, y) - f_b(x, y)|,$$

obviously a way to differentiation in time. To pick f_j as an event point we consider the *pixel change rate*

$$c_{\text{rate}}(i, j) = \frac{1}{XY} \sum_{x,y} c_{f_j,\bar{f}_{i,j-1}}(x, y)$$

where

$$c_{f_a,f_b}(x, y) = \begin{cases} 1 & \text{if } d_{f_a,f_b}(x, y) > T_p \\ 0 & \text{otherwise} \end{cases}$$

If $c_{\text{rate}}(i, j) > T_c$ we will consider f_j an event point. This step has three parameters: the window size $w = j - i$, the temporal support, a pixel difference threshold T_p and a change rate threshold T_c.

Figure 2 show two events from a video of TRECVID 2006 training set. The first is a true shot-boundary. The other is a camera movement event.

3. SHOT BOUNDARY VALIDATION

The second and last step is to filter event frames, looking for shot boundaries only. Consider a $2w + 1$ size window centered at position j. If f_j is a shot-boundary, it divides the window into two shot segments $S_1 = \langle f_s, f_{s+1}, ..., f_{j-1} \rangle$ and $S_2 = \langle f_j, f_{j+1}, ..., f_t \rangle$ where $s = j - 1 - w$ and $t = j + w$. Otherwise, we has an unique segment $S = \langle f_s, ..., f_t \rangle$.

We can represent the segments by their median frames $\bar{f}_{s,j-1}$, $\bar{f}_{j,t}$ and $\bar{f}_{s,t}$. So, we can check what is the more reasonable hypothesis: do we have two shots or just one shot inside the window?

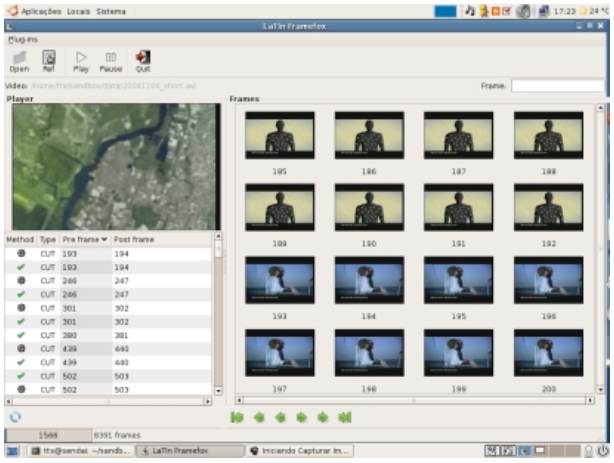

Figure 1: The visualization tool: *Framefox*.

Let $p_{k,l}$ be

$$p_{k,l} = \prod_{i=k}^{l} z(f_i, \bar{f}_{k,l}),$$

where $z(f_a, f_b)$ is a *similarity measure* with values in range $[0, 1]$ (similar frames present values closer to 1). In this work, we used the histogram intersection [2] as similarity measure.

A event at f_j will be considered a shot boundary if

$$p_{s,t} \leq c \cdot p_{s,j-1} \cdot p_{j,t}$$

where c is a positive constant, empirically chosen ($c < 1$).

4. IMPLEMENTATION

A C program implementation was developed, using the GStreamer [3] framework to perform video frame access and the Open Computer Vision Library (OpenCV) [1] to image processing routines. A video browser tool was built using the GTK+ toolkit (Figure 1) to visualization[1]. For a faster processing, we scale each frame by a $\frac{1}{8}$ factor before the first step.

5. RESULTS

Table 1 shows the results to the TRECVID 2006 shot boundary test set. The algorithm was designed for cut detection but we tested its performance to gradual transitions too. However, the gradual transition detection performance is low and should be dropped.

The tested parameters setting was $w = 3$, $T_p = 8$, $T_c = 0.40$. We used $c = 4$ for cut detection and $c = 1.5$ for gradual detection. These are the parameters with best performance on the training phase.

The algorithm performance is very different to American and Asian videos. The means of cut recall and precision on American videos are 0.834 and 0.710 respectively against 0.446 and 0.472 on Asian ones. We believe our algorithm is not able to work with the large amount of computer graphics and special effects used by the Asian videos.

[1]The shot boundary procedure is implemented in a dynamic library. The visualization tool loads the procedure at run time, as a plug-in.

6. CONCLUSIONS

We presented an algorithm based in median frames as a representation of visual content around a time point. The approach shows regular performance to cut detection in American videos but its low performance processing Asian videos has to be investigated. Adaptive parameter selection could be a way to get best results.

7. REFERENCES

[1] Intel Corporation. Open Source Computer Vision Library. http://www.intel.com/technology/computing/opencv/index.htm.

[2] M. J. Sawin and D. H. Ballard. Indexing via color histograms. In *Proceedings of Third International Conference on Computer Vision*, pages 390–393, Osaka, Japan, December 1990.

[3] W. Taymans, S. Baker, A. Wingo, R. S. Bultje, and S. Kost. GStreamer Open Source Multimedia Framework. http://gstreamer.freedesktop.org/.

$\bar{f}_{190,192}$ f_{193} $d_{f_193,\bar{f}_{190,192}}$

$\bar{f}_{1707,1709}$ f_{1710} $d_{f_1710,\bar{f}_{1707,1709}}$

Figure 2: Examples of median frames and absolute difference in event frames from 20041106 110000 MSNBC MSNBCNEWS11 ENG. The first case is a shot boundary event. The second is a camera movement event.

	Cut		Gradual	
Test Video	Recall	Precision	Recall	Precision
20051101 142800 LBC NAHAR ARB	0.324	0.301	0.203	**0.600**
20051114 091300 NTDTV FOCUSINT CHN	0.754	0.698	0.028	0.025
20051115 192800 NTDTV ECONFRNT CHN	0.440	0.507	**0.000**	**0.000**
20051129 102900 HURRA NEWS ARB	**0.290**	**0.292**	**0.352**	0.214
20051205 185800 PHOENIX GOODMORNCN CHN	0.422	0.545	0.100	0.142
20051208 125800 CNN LIVEFROM ENG	0.851	0.696	0.216	0.338
20051208 145800 CCTV DAILY CHN	0.496	0.486	0.207	0.150
20051208 182800 NBC NIGHTLYNEWS ENG	0.801	0.717	0.210	0.253
20051209 125800 CNN LIVEFROM ENG	0.779	0.686	0.171	0.240
20051213 185800 PHOENIX GOODMORNCN CHN	0.398	0.476	0.057	0.070
20051227 105800 MSNBC NEWSLIVE ENG	0.869	0.745	0.187	0.315
20051227 125800 CNN LIVEFROM ENG	0.818	**0.765**	0.148	0.266
20051231 182800 NBC NIGHTLYNEWS ENG	**0.884**	0.649	0.176	0.510

Table 1: Results of LaTIn-USP MD rev107 system. The algorithm was developed to detected cut transitions but was tested for gradual transition too (best and worst results for recall and precision in bold).