Shot boundary detection via similarity analysis

Matthew Cooper*, Jonathan Foote, John Adcock,
and Sandeep Casi

FX Palo Alto Laboratory
Palo Alto, CA USA

http://www.fxpal.com

Abstract

In this paper, we present a framework for analyzing video using self-similarity. Video scenes are located by analyzing inter-frame similarity matrices. The approach is flexible to the choice of both feature parametrization and similarity measure and it is robust because the data is used to model itself. We present the approach and its application to shot boundary detection.

1 Introduction

Video segmentation is an increasingly important problem. Numerous video retrieval and management tasks rely on accurate segmentation of scene boundaries. In this paper, we describe a framework for video analysis based on inter-frame similarity. This approach facilitates shot boundary detection and other characterizations of media and video structure. A particular benefit of this work is that it effectively uses the signal to model itself, making minimal assumptions about the nature or genre of the target video. FX Palo Alto Laboratory participated for the first time in TRECVID in 2003 with the primary goal of benchmarking our similarity-based shot boundary detection system using the TRECVID evaluation framework. We are pleased with our initial performance and experience with TRECVID and look forward to enhancing the system in the future in view of our results.

2 Similarity analysis

We detect scene boundaries by considering the self-similarity of the video across time. For each instant in the video, the self-similarity for past and future regions is computed, as well as the cross-similarity between the past and future. A

*for additional information contact cooper@fxpal.com

significantly novel point in the video, i. e. a scene boundary, will have high self-similarity in the past and future and low cross-similarity between them.

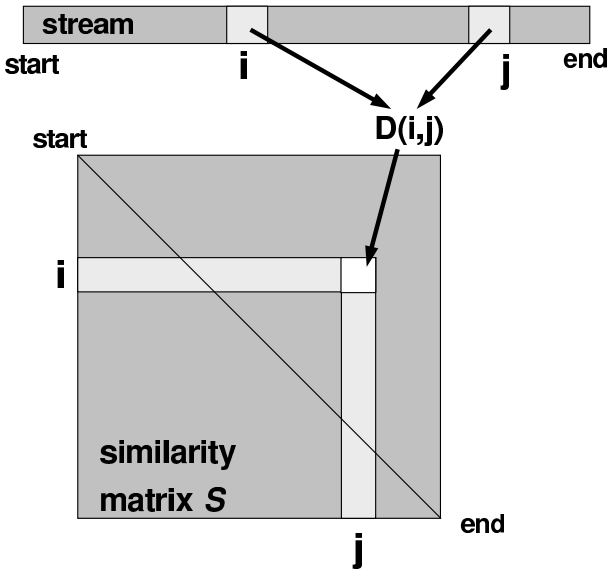

Figure 1: Diagram of the similarity matrix embedding.

Video frames are parameterized and are then embedded in a 2-dimensional representation [1]. Figure 1 shows how the distance measure is embedded. A measure D of the (dis)similarity between frame parameters $\vec{v}_i$ and $\vec{v}_j$ is calculated for every pair of video frames i and j. The matrix $\mathbf{S}$ contains the similarity measure calculated for all frame combinations i and j such that the $(i,j)^{th}$ element of $\mathbf{S}$ is $D(\vec{v}_i, \vec{v}_j)$. Time, or frame index, runs along both axes as well as the diagonal. In general, $\mathbf{S}$ will have maximum values on the leading diagonal (because every frame will be maximally similar to itself); furthermore if D is symmetric then $\mathbf{S}$ will be symmetric as well. An advantage of this approach is that the exhibited structure is derived entirely from the current video rather than from predefined models or parameterizations. There are minimal prior assumptions regarding the video content, which is an essential requirement for numerous applications.

3 System description

3.1 Computing the similarity matrix

Each frame is parameterized by features based on low-order discrete cosine transform (DCT) coefficients. Instead of intensity histograms, this implementation transforms the individual RGB frames into the Ohta color space according to:

$$\begin{bmatrix} o_1 \\ o_2 \\ o_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad . \tag{1}$$

In this color space, the three channels are approximately decorrelated [2]. The DCT of each channel is computed and a feature vector is formed by concatenating the resulting low frequency coefficients of the three channels. These features are compared using the (nonlinear) cosine distance measure

$$\mathbf{S}(i,j) = D(\vec{v}_i, \vec{v}_j) = \frac{< \vec{v}_i, \vec{v}_j >}{||\vec{v}_i||\,||\vec{v}_j||} \quad . \tag{2}$$

S can be visualized to let us clearly identify structure within a video. Regions of high similarity, such as a long sequence of identical frames, appear as bright squares on the leading diagonal. Repeated sequences are visible as diagonal stripes or checkerboards, offset from the main diagonal by the repetition time. For example, the similarity matrix of Figure 2 is from the video "19980203_CNN.mpg" from the 2003 test set (SB03). It is the portion of the full matrix corresponding to frames 1000–2750. There are cuts at frames 1496, 1567, 2374, and 2564. There are dissolves from frames 1124–1132 and 1641–1653.

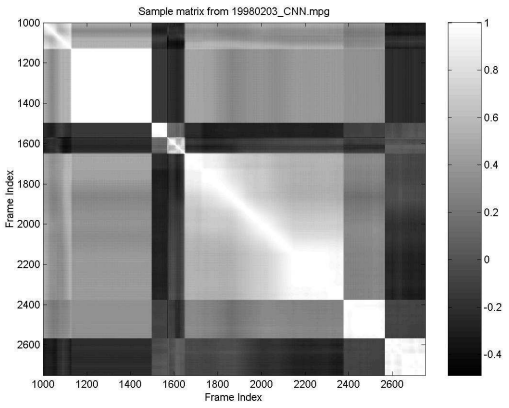

Figure 2: A similarity matrix using low frequency DCT features computed per (2).

3.2 Scene segmentation via kernel correlation

The similarity matrix of Figure 2 exhibits the segment structure of the source video. Frames 2374–2564 show high within-segment similarity in the corresponding bright square region along the leading diagonal. At frame 2564 there is a cut. The frames after 2564 comprise the next segment, also exhibiting high within-segment similarity in the corresponding square block along the main diagonal. At the same time, the rectangular region off the leading diagonal (bounded by rows 2374–2564 and columns 2564–2750) show low inter-segment similarity (i.e. high dissimilarity). This structure generates a checkerboard with crux at element (2564,2564). This general observation suggests that finding the scene boundary transitions is as simple as finding the checkerboards along the main diagonal of **S**. This can be done using a classic matched filter: correlating **S** with a kernel that itself looks like checkerboard [1]. We will call this class "checkerboard" kernels.

For automatic scene segmentation, we correlate the Gaussian checkerboard kernel of Figure 3 along the diagonal of the similarity matrix **S**. The result is a one-dimensional function of time (frame index). Intuitively, the correlation emphasizes regions with strong self-similarity while penalizing regions with significant cross-similarity.

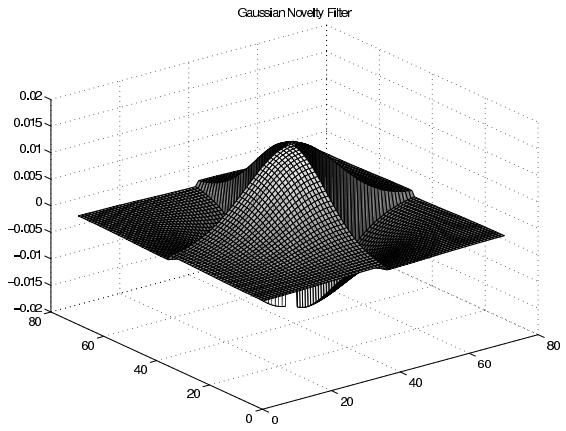

Figure 3: An example checkerboard kernel used to detect locally novel points in the video stream via correlation.

3.3 Lag domain implementation

On first inspection it seems that this segmentation algorithm requires $O(N^2)$ computations, where N is the number of frames. This is not the case in practice. For segmentation, there is no reason to calculate similarity matrix values fur-

ther from the leading diagonal than the extent of the kernel, which is typically a small constant (approximately $40 \ll N$ for TRECVID). Thus the algorithm can be computed in $O(N)$ computations, and can be computed on-the-fly as long as a small frame buffer (the size of the kernel) is available. Additionally, because both the similarity matrix and the kernel will typically be symmetric, many computations are redundant: only one half of the matrix and the kernel need be computed and correlated. Thus the algorithm is quite competitive computationally with seemingly simpler approaches such as histogram differences.

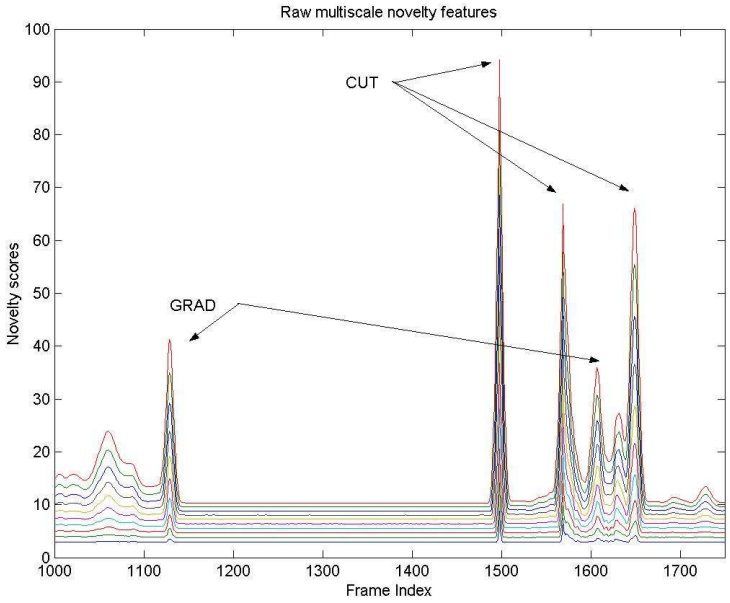

Figure 4: The figure shows novelty scores computed with varying kernel widths from a portion of the matrix of Figure 2.

3.4 Multi-scale shot boundary detection

The width of the checkerboard kernel determines the sensitivity of the novelty score to segments of different lengths. To take a multi-scale approach we compute novelty scores using a range of kernels with varying width. This is somewhat similar to scale-space analysis [3], but includes second order information off the main diagonal of the kernel. It is also loosely in the spirit of the multi-scale work at TRECVID by Pickering, *et al.* [4]. Figure 4 shows novelty scores computed using several kernels with widths between 4 and 16. Note that the cuts in the video exhibit sharp peaks at all scales while the gradual

transitions show shorter, broader peak structure.

For the competition, we built a system that processed novelty scores computed with kernels of width ranging from 4 to 32. The first step was to calculate the cumulative novelty score by summing the individual novelty scores across scale (kernel width). This cumulative score was filtered linearly for peak enhancement. To locate candidate shot boundaries, we applied a threshold to the filtered cumulative novelty score. Although the filtering clearly reduced the number of false positives, it also reduced the recall. A second thresholding step was used to separate gradual and cut boundaries. Additionally, gradual boundaries were required to remain above the secondary threshold for a minimum duration. This was based on the observation of the breadth of the peaks corresponding to gradual, relative to the sharpness of the peaks for cuts. The system was implemented in C and was computationally lightweight. There was no additional processing for brightness, camera flash, or motion estimation as is common in other systems.

Results on SB03 Test Set									
	Overall			Cuts			Graduals		
	R	P	F	R	P	F	R	P	F
Comp.	0.643	0.837	0.727	0.805	0.893	0.847	0.199	0.502	0.285
	0.781	0.803	0.792	0.900	0.895	0.897	0.462	0.523	0.491
	0.772	0.828	0.799	0.901	0.888	0.895	0.424	0.597	0.496
	0.768	0.839	0.801	0.901	0.888	0.895	0.409	0.631	0.496
	0.760	0.859	0.807	0.902	0.889	0.895	0.378	0.708	0.493
	0.779	0.810	0.794	0.891	0.904	0.898	0.476	0.533	0.503
	0.769	0.837	0.802	0.892	0.899	0.896	0.439	0.606	0.509
	0.766	0.848	0.805	0.893	0.899	0.896	0.425	0.643	0.512
	0.757	0.870	0.809	0.893	0.901	0.897	0.390	0.717	0.505

Table 1: Performance results on SB03 test set. "Comp." is one of the best of our submitted runs. The remaining columns are variants of the system used in TRECVID that have been tuned since the competition. The column headings R, P, and F, denote recall, precision, and F-score, respectively.

4 Results

4.1 SB03

For the competition, we presented results of a system trained largely on the SB02 test set collection. This decision was made to base the system on manually, rather than machine, segmented training data. As a result, our results were skewed towards high precision and low recall, as demonstrated in the row of Table 1 denoted "Comp." in the SYSTEM column. The figures of merit for evaluation are recall (R), precision (P), and F-score (F) which are common in information retrieval and are defined in the TRECVID overview [5]. The 2002

and 2003 data sets were substantially different, and our performance, recall in particular, suffered somewhat as a result. Also, our performance detecting gradual transitions was poor relative to our cut detection performance. Our original system was deigned primarily to detect cuts, thus, we have adjusted the system parameters for high precision and low recall gradual boundary detection. Since the competition, we have further adjusted the system for improved performance on the 2003 data set. The remaining columns of Table 1 demonstrate a small range of tradeoffs in precision and recall.

4.2 Combining SB02 and SB03

Our principal objective is to build a lightweight similarity-based video segmentation system with broad applicability. For this reason, we are presently continuing system development using the combined SB02 (2002) and SB03 test data. This joint data consists of approximately 11 hours of video with 5734 total shot boundaries: 4132 cut and 1602 gradual. The boundaries are manually determined and the cumulative data represents a variety of video genres. Table 2 shows results for two variants of our system on the two test sets individually and jointly.

Results on SB02 Test Set									
	Overall			Cuts			Graduals		
	R	P	F	R	P	F	R	P	F
I	0.777	0.787	0.782	0.914	0.848	0.880	0.451	0.583	0.509
II	0.758	0.843	0.798	0.923	0.923	0.923	0.361	0.549	0.436
Results on SB03 Test Set									
	Overall			Cuts			Graduals		
	R	P	F	R	P	F	R	P	F
I	0.785	0.779	0.782	0.914	0.872	0.892	0.439	0.488	0.462
II	0.765	0.851	0.806	0.902	0.888	0.895	0.398	0.682	0.503
Results on Joint SB02 & SB03 Test Set									
	Overall			Cuts			Graduals		
	R	P	F	R	P	F	R	P	F
I	0.782	0.782	0.782	0.914	0.864	0.888	0.443	0.520	0.479
II	0.763	0.848	0.803	0.909	0.900	0.904	0.385	0.629	0.477

Table 2: Performance results for the on combined SB02 and SB03 test sets. The column headings R, P, and F, denote recall, precision, and F-score, respectively.

5 Conclusion

We have presented a novel video segmentation framework and demonstrated its performance on two year's worth of TRECVID shot boundary test data. Our system's initial performance, based on relatively simple features and lightweight processing, has been good. In view of our results, we envision several enhancements to the current system. Many groups report improved recall performance

using block-based features. Integrating block and global features is one future goal. We also hope to improve our gradual boundary detection performance with the addition of block-based features or motion estimation. Also, we likely need to look more closely the individual novelty scores, rather than relying primarily on the cumulative novelty. Finally, we hope to employ statistical classification techniques in lieu of threshold based methods for identifying shot boundaries as local maxima in the novelty scores.

References

[1] M. Cooper, and J. Foote. Scene Boundary Detection Via Video Self-Similarity Analysis. *Proc. IEEE Intl. Conf. on Image Processing*, 2001.

[2] Y-I Ohta, T. Kanade, and T. Sakai. Color Information for Region Segmentation. *Comp. Graphics & Image Processing*, **13**:222-241, 1980.

[3] A. Witkin. Scale-space Filtering: A New Approach to Multi-scale Description. *Proc. IEEE ICASSP*, 1984.

[4] Marcus J. Pickering, Stefan M. Rüger. Multi-timescale Video Shot-Change Detection. TRECVID 2001.
http://trec.nist.gov/pubs/trec10/papers/video-pickering-rueger.pdf

[5] A. F. Smeaton, W. Kraaij, P. Over. TRECVID-An Introduction. TRECVID 2003.
http://www-nlpir.nist.gov/projects/tvpubs/papers/tv3intro.paper.pdf