Shot boundary detection via similarity analysis*

Matthew Cooper, Jonathan Foote, John Adcock, and Sandeep Casi
FX Palo Alto Laboratory
3400 Hillview Ave. Bldg. 4
Palo Alto, CA 94304
http://www.fxpal.com

October 31, 2003

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 similarity measure and is robust as the data is used to model itself. We present the approach and its application to scene 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 scene boundary detection and other video characterizations. 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.

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 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.

Video frames are parameterized and are then embedded in a 2-dimensional representation [1]. Figure 1 shows how the distance measure is embedded. A

[*] for additional information contact cooper@fxpal.com

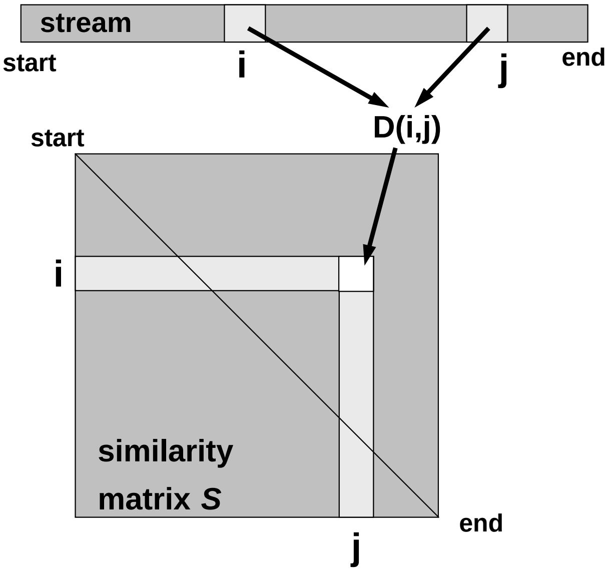

Figure 1: Diagram of the similarity matrix embedding.

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 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 processes every 10^{th} frame and 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 feature vectors are compared using the cosine distance measure and Algorithm 1 is used to compute the scene boundaries. 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 (a) is from the video "19980203_CNN.mpg" from the 2003 test set (SB03). There are cuts at frames 1496, 1567, 2374, and 2564. There are dissolves from frames 1124–1132 and 1641–1653.

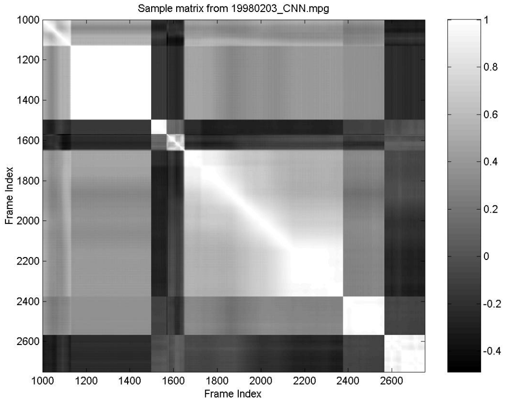

Figure 2: The figure shows a similarity matrix computed per Algorithm 1.

3.2 Scene Segmentation Via Kernel Correlation

The similarity matrix of Figure 2 exhibits the segment structure of the source video. Frames 1132–1496 show high within-segment similarity in the corresponding square region along the leading diagonal. At frame 1496 there is a cut. The frames from 1496 – 1567 comprise the next segment, also exhibiting high similarity along the main diagonal. At the same time, the rectangular region off the leading diagonal (bounded by rows 1132–1496 and columns 1496–1567) show low (inter-segment) similarity. This forms a checkerboard with crux at element (1496,1496). This 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.

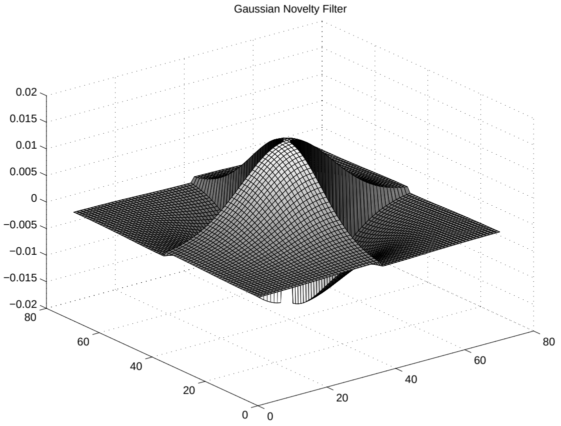

Figure 3: The figure shows a similarity matrix computed per Algorithm 1.

3.3 Varying the kernel width

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 similar to scale-space analysis [3], but includes second order information off the main diagonal of the kernel. It is also somewhat in the spirit of the multi-scale work at TREC 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. We have built a heuristic-based system to exploit these differences in the novelty features to detect cut and gradual transitions for the

TREC competition. Because of intellectual property concerns, we can not describe the system in any additional detail at present.

Algorithm 1 *Self-Similarity Based Scene Change Localization*

1. *Compute cosine similarity matrix*
    - (a) *Transform every tenth frame from RGB to intensity (greyscale) image*
    - (b) *Compute histogram of intensity image*
2. *Compute correlation along diagonal of similarity matrix with Gaussian checkerboard kernel (Figure 3(c))*
3. *Locate peaks via analysis of the first and second differences of the output signal*
4. *Label peaks as scene boundaries*

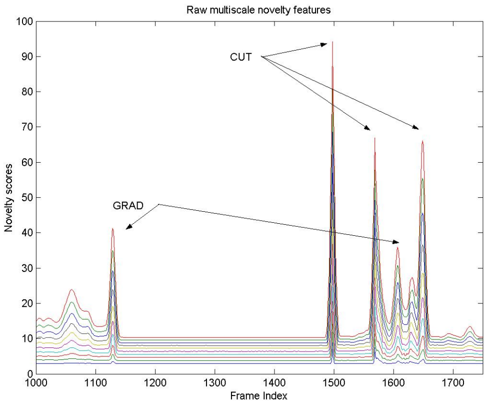

Figure 4: The figure shows novelty scores computed with varying kernel widths from the matrix of Figure 2.

4 Experiments

5 Conclusion

We have presented a novel video segmentation algorithm and demonstrated its performance. Better features based on color and shape further improved the segmentation. Though on first inspection it seems that Algorithm 1 requires

$\mathbf{O}(n^2)$ computations, this is not the case in practice. For segmentation, there is no reason to calculate similarity matrix values further from the diagonal than the extent of the kernel, which is typically a small constant. Thus the algorithm can be computed in $\mathbf{O}(n)$ 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 with seemingly simpler approaches such as histogram differences.

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. Rger. Multi-timescale Video Shot-Change Detection. TREC 2001.