SHOT SEGMENTATION AND GROUPING FOR PTZ CAMERA VIDEOS

Andrew Pulver*, Ming-Ching Chang†, Siwei Lyu‡

* Computer Science Department
University at Albany, State University of New York
Email: apulver@albany.edu

† GE Global Research
Niskayuna, NY 12309
Email: changm@ge.com

‡ Computer Science Department
University at Albany, State University of New York
LI-67A, 1400 Washington Avenue, Albany, NY 12222
Email: slyu@albany.edu

***Abstract*—We present a method for detecting shot boundaries and grouping together shots that were taken from identical camera directions. The technique utilizes methods including spectral clustering and phase correlation to achieve fast and accurate segmentation and grouping.**

I. Introduction

The exact definition of a shot varies, but it is roughly defined as an uninterrupted sequence of frames that have been taken from one camera [10][9]. We examine videos taken from PTZ (pan, tilt, zoom) security cameras. A PTZ camera may be programmed to automatically transition between a set of directions. The resulting video would consist of many shots, each of which correspond to a period of time where the camera was not moving. We define a scene as a group of shots taken from the same camera direction and thus having the same background. Organizing these videos into scenes can be useful as a preprocessing step for video summarization techniques such as video synopsis [6], which requires a fixed background.

II. Related Works

Many methods exist for shot boundary detection. A recent survey lists a large collection of these techniques [8]. Two methods noted in the survey to perform well are phase-corrleation [9] and background tracking [5]. We combine a phase-corrlation based approach with background-tracking.

Our method uses spectral clustering to group shots together, as do some previous works on scene detection and shot grouping [4] [2]. Spectral clustering requires some way of measuring the similarity between two items. Past research on scene detection involve methods that rely, at least partially, on color similarities between scenes [2] [7]. In [2], a scene is defined as "...a series of semantically correlated shots". Similarly, in [7], they define a scene as "... a subdivision of a play in which either the setting is fixed or when it presents continuous action in one place". These definitions are more general than ours and their similarity measures are not directly applicable to this problem.

Color histograms are not a reliable measure of shot similarity in our case, since the videos we examine tend to have different scenes with a high degree of color similarity. Another point to consider is that the color content may vary greatly between different shots of the same scene, for instance, a video with moving traffic. Since our definition of a scene is more specific, we must define a different similarity measure. Temporal similarity is also taken into account in [4]. However, in our case, temporal distance is also not a good indicator of shot similarity. Our task boils down to background matching, but it is complicated by unstable lighting conditions and foreground appearance.

III. Method

Phase correlation is an established method for detecting hard cuts [10] [9]. However, applying phase correlation over the frames of a high resolution video can be computationally expensive. Instead we use phase correlation on a small portion of the image. We also propose using phase correlation as a measure of similarity for grouping shots into scenes.

IV. Shot Boundary Detection

We define a shot as an uninterrupted segment of video taken from the same camera direction. It is assumed that foreground objects may change rapidly, and that changes in the foreground do not necessarily indicate a cut. The background tracking technique in [5] makes use of only a portion of the image, which they call the "fixed background area", or FBA. This consists of a bar along the top of the image and two bars along the left and right sides of the image. These three portions are combined into one image called the "transformed background area" or the TBA. This is done by rotating the left portion clockwise and rotating the right portion counter-clockwise by ninety degrees. The process is nicely illustrated in the original paper. They also provide formulas to calculate the exact dimensions for the FBA.

Their shot detection algorithm is a multi-stage process. In our experiments, we noted that the second stage of their algorithm (which is designed to filter out false positives) tended to filter out actual cuts, likely due to the color similarity of our shots. Better results can be obtained by skipping their detection method altogether and using phase correlation on the transformed background area. This method is simple and fast, given that FFTs are only computed over a fraction of the image. A simple method for phase correlation applied to cut detection is given in [10]. The correlation surface can be computed as [10]:

$$S(f_i, f_{i+1}) = F^{-1}\left(\frac{F(f_i^*) \cdot F(f_{i+1})}{\mid F(f_i^*) \cdot F(f_{i+1}) \mid}\right) \tag{1}$$

where f_i is the ith frame after applying a window function, $*$ is the complex conjugate, and F is a 2-dimensional Fourier transform. If f_i and f_{i+1} are approximately the same image, there will be a high peak in the correlation surface.

They partition the images into many overlapping blocks $b^{(k)}$ and take:

$$P(f_i, f_{i+1}) = -\sum_{k=1}^{B} ln(max_{y,x}(S(b_{f_i}^{(k)}, b_{f_{i+1}}^{(k)})) \tag{2}$$

For our experiments, each image was divided into four parts, overlapping by $\frac{1}{4}$ the total height/width of the image in the vertical/horizontal direction, respectively. We also took $\frac{1}{B}$ of the sum in (2), so our thresholds don't depend on the number of blocks.

Relying only on the FBA can result in false positives. A large moving object may occlude most of the FBA but not the rest of the image. To prevent this kind of false positive, the entire image can be checked with phase correlation to confirm the shot boundary. To summarize, the steps we use to detect a shot boundary between frames f_i and f_{i+1} are:

1) Compute the TBAs: t_i, t_{i+1} for frames f_i, f_{i+1}.
2) If $P(t_i, t_{i+1}) > T$ then there is a potential cut at i
3) If $P(f_i, f_{i+1}) > T$ then there is a cut at i.

Where T is a threshold.

V. Grouping

Individual shots should be grouped according to the camera direction from which they were taken. Each output video would then consist of multiple shots with the same background.

Spectral clustering can be used to group different shots together [4] [2]. However, instead of using colors to compare shots, we use phase correlation. The whole frames of each shot (as opposed to just the TBAs) are used, after being down-sampled as suggested in [9]. The mean of each shot is computed and used as a representative for that shot.

We need a measure of similarity, rather than measure of dissimilarity, so we take:

$$P_s(m_i, m_j) = -\frac{1}{B}\sum_{k=1}^{B} ln(1 - max_{y,x}(S(b_{m_i}^{(k)}, b_{m_j}^{(k)}))) \quad (3)$$

where m_i and m_j are the means of the ith and j^{th} shots.

We can also adjust this value according to the coordinates of the peak in the correlation surface. Since the camera should be still during each shot, if the peak is far from $(0, 0)$, then we do not consider these images similar. The similarity measure can be multiplied by:

$$D(m_i, m_j) = exp(\frac{-||argmax_{y,x}S(m_i, m_j)||}{C})) \quad (4)$$

where C is some constant.

Or, since we use several blocks:

$$D_b(m_i, m_j) = \frac{1}{B}\sum_{k=1}^{B} D(b_{m_i}^{k}, b_{m_j}^{k}) \quad (5)$$

Using the spectral clustering algorithm found in [3] along with our similarity measure we follow these steps to compute the clusters of shots. The number of clusters can be estimated using the method from [11]. Let $\{m_1, m_2, \ldots, m_n\}$ be the mean of each shot.

The algorithm:

1) Let $A_{i,j} = P_s(m_i, m_j) \cdot D_b(m_i, m_j)$
2) Let $L = D^{-1/2}AD^{-1/2}$ where $D_{i,i} = \sum_j A_{i,j}$
3) Find the eigenvalues and eigenvectors of L.
4) Create another matrix X using the corresponding eigenvectors as columns. Normalize the rows of X to get Y. Use k-means++ to cluster the rows of Y. [1]
5) The clusters for elements $m_j \in \{m_1, m_2, \ldots, m_n\}$ corresponds to the clusters for rows $Y_j \in Y$

VI. Experiments

We tested this method on a group of videos taken by PTZ cameras. These cameras switched between a set of directions on a schedule. For instance, a camera may rotate clockwise by ninety degrees every few seconds, producing a video with four scenes. The transitions between camera directions are nearly instantaneous, i.e. the camera rapidly changes directions rather than slowly panning to the next position. Nearly all hard cuts were detected and nearly all shots were grouped into the correct scene.

We always down-sampled images (both TBAs and whole frames) to one quarter their original size before applying phase correlation. We use the formulas in [5] to calculate the FBA dimensions, however we start with $\frac{1}{8}$ the image width instead of $\frac{1}{10}$, resulting in a slightly larger TBA. On our 640×480 resolution videos, the cut detection works in real time (using FFTW to compute the FFTs). Of course, calculating the affinity matrix becomes slow for videos with many shots. Processing long videos in shorter segments and matching the resulting scenes may be a possibility.

Figure 1 illustrates the results of performing phase correlation on TBAs and on the whole frames. The spikes in the graph indicate shot transitions. Using the TBA instead of the whole frame typically produces results that are at least as accurate. It is also interesting to look at the similarity between the mean of a shot and the means of the other shots in a video. Figure 2 shows the results of such a comparison (we used a larger portion of the video for this figure). Phase correlation and histogram correlation were used for figures 2a and 2b respectively.

We hand-labeled the cuts on six PTZ camera videos and ran the detection algorithm. The results can be seen in Figure 3. The videos were taken by cameras on a schedule and switched from one direction to the next every few seconds. The transitions between camera directions were brief, but in some cases lasted several frames. Each frame that the camera is moving or zooming was labeled as a cut. In some cases it was difficult to decide whether a frame should be a cut or not. We define a "Major False Positive" as a false positive that does not occur within one frame of an actual cut and a "Major False Negative" as a false negative that does not occur within one frame of a reported cut. In other words, the false positives and negatives not counting off-by-one errors.

VII. Conclusion

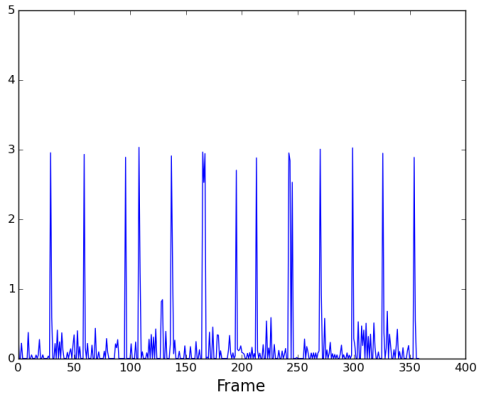
We have presented a method for dividing videos into shots and grouping shots with identical backgrounds into scenes. This method addresses a case not previously considered and performs well. Future work may focus on achieving real-time performance and possibly adapting the method to work on a more general set of videos.

References

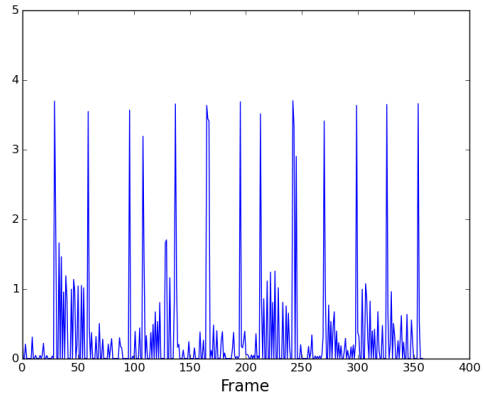
[1] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[2] Vasileios T Chasanis, CL Likas, and Nikolas P Galatsanos. Scene detection in videos using shot clustering and sequence alignment. *Multimedia, IEEE Transactions on*, 11(1):89–100, 2009.

[1] The k-means++ algorithm (see [1]) produced better results than ordinary k-means. In [2], they also use a modified version of k-means.

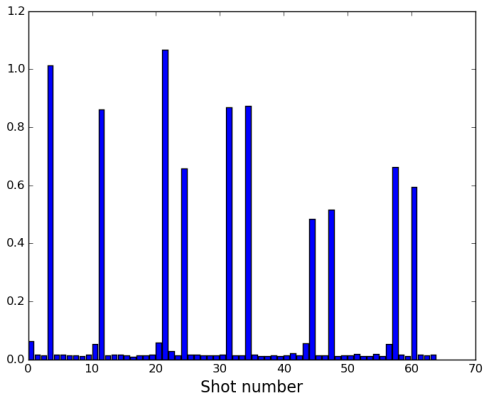


(a) TBAs

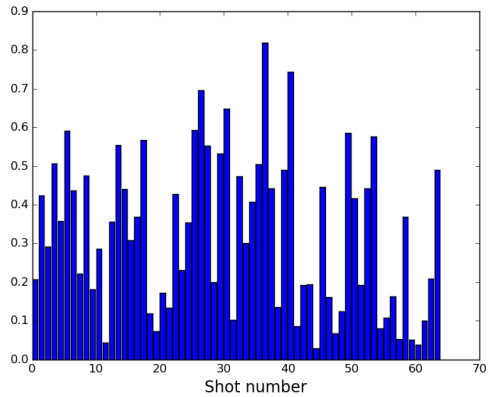


(b) Whole Frames

Fig. 1. $P(f_i, f_{i+1})$ for one of our test videos



(a) Phase Correlation



(b) Histogram Correlation

Fig. 2. $P(f_i, f_{i+1})$ for one of our test videos

Video	1	2	3	4	5	6
Frames	1801	144	2399	1357	899	1370
Actual Cuts	99	21	62	86	124	129
Reported Cuts	103	21	68	90	122	128
False Positives	5	1	7	9	6	12
False Negatives	1	1	1	5	8	14
Precision	95.1%	95.2%	89.7%	90.0%	95.1%	90.6%
Recall	99.0%	95.2%	98.4%	94.2%	93.5%	89.1%
Major False Positives	0	0	4	5	0	6
Major False Negatives	0	0	0	0	0	2
Adjusted Precision	100.0%	100%	93.8%	94.2%	100%	95.0%
Adjusted Recall	100.0%	100%	100.0%	100.0%	100%	98.3%

Fig. 3. Cut Detection Results

[3] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[4] Jean-Marc Odobez, Daniel Gatica-Perez, and Mael Guillemot. Spectral structuring of home videos. In *Image and Video Retrieval*, pages 310–320. Springer, 2003.

[5] JungHwan Oh, Kien A Hua, and Ning Liang. Content-

based scene change detection and classification technique using background tracking. In *Electronic Imaging*, pages 254–265. International Society for Optics and Photonics, 1999.

[6] Yael Pritch, Alex Rav-Acha, and Shmuel Peleg. Nonchronological video synopsis and indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1971–1984, 2008.

[7] Zeeshan Rasheed and Mubarak Shah. Detection and representation of scenes in videos. *Multimedia, IEEE Transactions on*, 7(6):1097–1105, 2005.

[8] Raahat Devender Singh and Naveen Aggarwal. Novel research in the field of shot boundary detection–a survey. In *Advances in Intelligent Informatics*, pages 457–469. Springer, 2015.

[9] Oguzhan Urhan, M Kemal Gullu, and Sarp Erturk. Modified phase-correlation based robust hard-cut detection with application to archive film. *Circuits and Systems for Video Technology, IEEE Transactions on*, 16(6):753–770, 2006.

[10] Theodore Vlachos. Cut detection in video sequences using phase correlation. *Signal Processing Letters, IEEE*, 7(7):173–175, 2000.

[11] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601–1608, 2004.

Siwei Lyu Siwei Lyu received his B.S. degree (Information Science) in 1997 and his M.S. degree (Computer Science) in 2000, both from Peking University, China. He received his Ph.D. degree in Computer Science from Dartmouth College in 2005. From 2000 to 2001, he worked at Microsoft Research Asia (then Microsoft Research China) as an Assistant Researcher. From 2005 to 2008, he was a Post-Doctoral Research Associate at the Howard Hughes Medical Institute and the Center for Neural Science of New York University. Starting in 2008, he was Assistant Professor at the Computer Science Department of University at Albany, State University of New York, and was promoted to Associate Professor in 2014. Dr. Lyu is the recipient of the Alumni Thesis Award of Dartmouth College in 2005, IEEE Signal Processing Society Best Paper Award in 2010, and the NSF CAREER Award in 2010. He has authored one book, and held two U.S. and one E.U. patents. He has published more than 50 conference and journal papers in the research fields of natural image statistics, digital image forensics, machine learning and computer vision.

Andrew Pulver Andrew Pulver is a second year PhD student at the University at Albany. His current areas of interest include computer vision and deep learning.