ROBUST AUTOMATED FOOTAGE ANALYSIS FOR PROFESSIONAL MEDIA APPLICATIONS

J W Mateer and J A Robinson

University of York, UK

ABSTRACT

We report a method for automated video indexing and shot characterization that meets the specific requirements of professional post-production and archivist end users. *ASAP – Automated Shot Analysis Program* – interprets source video material in a manner consistent with industry practice and generates logs and searchable databases of cut location and camera activity. It uses projective transform estimation methods in conjunction with temporal filtering to resolve complex subject motion. Using challenging test footage and rigorous metrics we show that *ASAP* is more robust than well-established colour histogram boundary detection methods and effective at parsing complex camera movement. These results indicate that our techniques are potentially valuable for professional application.

INTRODUCTION

Shot boundary detection and camera movement classification are the backbone of any automated footage parsing system. In professional applications robustness and accuracy are vital whether for archiving historical material or streamlining and enhancing the editing process. In both contexts source footage can be of diverse quality with significant variation in visual clarity, camera and subject movement, and overall shot duration. As a result it is paramount that an automated method interpret footage accurately in a wide range of conditions. We have designed *ASAP – Automated Shot Analysis Program* – with these industry needs in mind.

Research into this area is not new. Seyler's analysis of differences between video frames (1) was the first of a host of studies into shot boundary detection (such as (2-5)), camera movement classification (6-7) and other content extraction techniques (8-10).

Several researchers have reported methods of cut detection that can yield over 95% accuracy with a false detection rate of 5% or less (typified by Lienhart (4)). The presentation of impressive results from these studies has led many to believe that this problem has effectively been solved. But all of these results are highly dependent on the footage analysed. We suggest that insufficient attention has been paid to selection of test cases that accurately reflect the range of conditions in archival and production footage (Mateer presents a detailed critique and new approach in (11)). As a result, important failure modes go unanalysed. For example approaches that combine colour histogram matching and temporal consistency often fail to accurately parse shots of very short duration (<5 frames), segments with intermittent occlusion and cuts between different but graphically similar shots (an example follows). In this paper we not only report our method and tests, but also show how appropriately chosen test footage reveals the true performance of shot analysers.

METHOD: "*ASAP – AUTOMATED SHOT ANALYSIS PROGRAM*"

ASAP consists of a frame-by-frame camera motion estimator applied both with and without temporal pre-filtering. A movement parser then connects interframe movements into strings and applies syntactic rules to distinguish different types of movement.

Camera Motion Estimator

We use a fast, high-accuracy, simplex-based projective transform estimator developed by Robinson (a detailed description can be found in (12)). The estimator uses simplex minimization of a disparity function calculated over a mesh of samples taken from the picture. In comparison tests with other perspective estimators, it performs as accurately but several times faster than its competitors. This estimator has been used for object-based video analysis and coding (13-14), but in *ASAP* we simply take the output of eight perspective transform parameters, along with a single measure of disparity, for input to the movement parser.

Temporal Filter

The motion estimator is applied directly to the raw video input and to a temporally-filtered version of the input. We use a 16-tap temporal median filter that attenuates the effect of temporary scene occlusions. This allows us to disambiguate between genuine cuts and gross image changes caused by fast-moving foreground objects.

Classifier

The classifier consists of a movement parser that also functions as a cut detector. It clusters consistent movements over consecutive frames into tentative zooms, pans and tilts. If the best perspective transform between two frames yields a significant final disparity, its parameters are examined for consistency with the temporally-filtered information, and if inconsistent, a cut is declared. Pans and tilts are detected from translation parameters, and zooms from a combination of the scale/rotation matrix entries in the projective transform. It is also possible to detect and quantify camera roll.

Having divided the stream of camera movements into *tentative* zooms, pans and tilts (which may happen in parallel), the classifier applies a second level of analysis. The zooms are examined first. If of sufficient magnitude, they are accepted as fundamental motions and subsume any other kind of movement. For pans and tilts, the parser examines the series of tentative movements in the shot, and infers that the movement is one of three types: (i) a fundamental pan or tilt, which is a consistent movement in a particular trajectory, (ii) tracking, where the camera appears to be following a moving object, (iii) jitter. The last of these is ultimately classified as part of a hold, along with any genuinely

stationary camera shots. The motion estimator is able to correct for jitter with motion stabilization if necessary.

The output of the classifier is presented in two main forms. First, a shot log with time code for in/out points, duration, a representative frame of each camera movement and a mosaic showing complex moves in a storyboard-like format, provides a quick visual reference for the footage (a web-based example without mosaics can be seen at (15)). Second, a searchable database is generated that enables easy location of cuts or movements of a particular type, duration, extent and speed. This later feature enables editors to easily find matching motions within shots enabling seamless match transitions, a highly time consuming task when done manually.

Linear and Hierarchical Processing

ASAP is built around a fast global projective estimation algorithm. We are able to achieve a low average processing time (<140ms per pair of 720x560 frames on a 2 GHz Pentium IV, before temporal filtering) by applying it in a hierarchical way. First we examine frames separated by four frame periods using the fastest version of the perspective analyser. When the estimate produced is sufficiently accurate, the movement parameters are scaled to per-frame values and accepted. When the estimate is poor, *ASAP* switches down through a sequence of increasingly accurate matches.

For a low-activity video sequence, it is possible to run the hierarchical version of *ASAP* at an average rate below 40ms/frame (i.e. video frame rate). For high activity, large buffers or a higher performance processor would be required in a real-time system.

EXPERIMENTS AND ANALYSIS

Heretofore many analyses of similar automated parsing systems have consisted of footage chosen arbitrarily, often based on footage at hand. Initiatives such as TRECVid (16) have attempted to provide a large-scale dataset as a representative sample of real-world conditions. However, despite covering a range of genres, film and video types and historical periods, that test set was not compiled with specific input from post-production or archivist end-users nor with any specific criteria based on expert knowledge of cinematic language or production convention. As such it is not fully indicative of the range of conditions present in these domains, particularly with regard to editing and camerawork. Fast-paced montage, jump cuts, graphic match cuts, swish pans, snap zooms and racking focus are some of the attributes found in source footage that are not represented. Our contention is that performance cannot be adequately analysed without a clearly principled basis for choosing sample sets, including a formal understanding of the cinematic style employed by the programme makers (11), if a system is to ultimately be applied in a real-world setting.

Test Footage Employed

Reviewing recognised technical and critical cinema texts (17-19) as well as drawing on professional filmmaking expertise we chose source footage from the 1970 film *Le Mans* specifically due to its directorial and editorial style. The section tested encompasses the first 290 shots (32,229 frames) after the head title sequence. It consists of a mix of location Cinema Verité hand-held footage and conventional staged narrative production. Editing builds from a slow, expository pace and to a very fast montage of shots reaching a visual climax in which the duration of some shots is very short (<4 frames, see fig. 1).

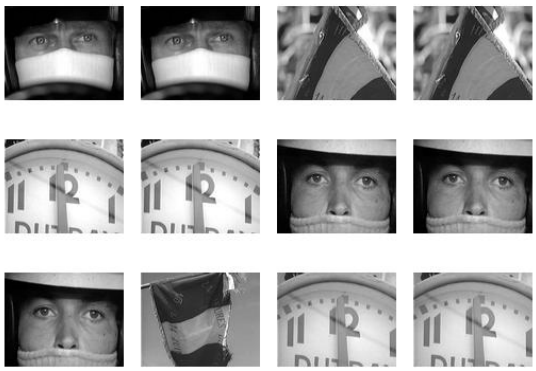

Figure 1: Consecutive frames showing fast edits

In addition, there are several instances of intentional jump and graphic match cuts. There is a wide range of shot types with many complex compositional elements, including significant subject occlusion (fig. 2), complex relative motion and fast motion of both subject and camera.

Figure 2: Consecutive frames showing occlusion

Taken as a whole, this footage represents a very significant challenge for automated analysis.

Experimental Method

An AVI file and an identical set of JPEG stills were generated from a NTSC video master of the 290 shot test sequence. A hand log of the test footage was created using industry-standard criteria to characterize start/end times, shot type and camera movements, all with frame accurate precision. This was then converted to a simple text file using abbreviations for moves (e.g., L for Pan Left, etc.) to enable automated scoring.

Cut Detection

Media professionals require shot boundary detection to be truly frame accurate. As such common measures of *Precision* and *Recall* are not best suited for this analysis. Straightforward measurement is possible in terms of missed and erroneously flagged cuts. However, any cut that is not frame accurate should be counted as two mistakes: a completely missed cut, plus an additional false cut. We measure overall accuracy as given by

$$\text{Accuracy} = 1 - N_{missed} / N_{true} - N_{false} / (N_{true} - N_{missed} + N_{false})$$

To compare *ASAP* against established methods we obtained a copy of Lienhart's *CutDet* (20) to directly gauge relative performance in cut detection using a well-studied and reportedly highly effective approach. Several trials were run using different thresholds to determine optimal settings and compare areas of strength and weakness in both systems (see fig. 3).

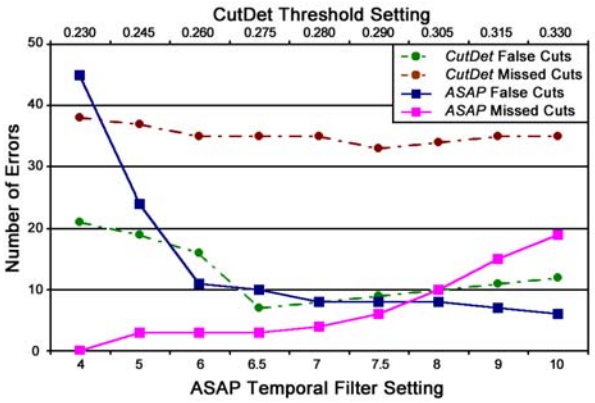

Figure 3: Cut detection performance over 290 shots

With a temporal filter setting of 5 or higher, *ASAP* correctly detects over 90% of cuts for the test footage. For this data set, the optimal setting is 7, with cut detection accuracy of 95.9% overall. This compares very favourably with *CutDet*'s best result of 85.2% at a threshold of 0.275. It is recognised that this version of *CutDet* cannot be modified to attempt the detection of shots with a duration of fewer than six frames, as occurs in shots 254-271. Discounting that section of the test set *ASAP* still outperforms *CutDet* by nearly 4%, significant in a professional end-user context. Examining the areas where the systems failed it is clear that *ASAP* is much better able to cope with occlusion, failing in only one instance. *ASAP* also correctly parsed all four graphic match cuts whereas *CutDet* was only able to detect two. Neither system was able to parse the two one-frame jump cuts. This is important as the detection of drop frames is vital to editors and thus warrants further investigation. Overall results indicate that *ASAP* is highly effective and we would welcome the opportunity for direct comparison with other approaches.

Camera Move Categorization

Locating the exact start frame of a camera move is desirable although in practice edits are rarely made using the precise start and end points of the movement. For evaluation purposes, however, it is important to judge a system based on its absolute performance.

Camera move characterization and camera move frame accuracy were analysed using a programme that took *ASAP*'s output and compared it to the expert's hand log. At present, *ASAP* cannot parse fully moving camera shots (e.g., dolly, crane, Steadicam, etc.) and so was penalised for this. The performance scores were calculated based on the following criteria:

A move was considered *correctly classified* if *ASAP* identified a move with extents that overlapped with a move of the same type in the hand log. *ASAP*'s other moves were categorized as *false*, and the hand log's other moves were categorized *missed*. The *Classification Rate* per shot is the number of correctly classified moves divided by the total of correct, false and missed moves.

A correctly classified move was assessed for frame-accuracy. The *move accuracy* was defined as the proportion of the time that the hand log and *ASAP*'s log both identified the move as happening, divided by the total extent of time from when either log identified the move starting, to when either log identified it as ending. The *average move accuracy* gives the accuracy performance of all recognized moves within a shot.

Duration-weighted accuracy measures the proportion of frames within a shot where *ASAP* and the hand log report the same movement in progress (or both report a static hold) divided by the total number of frames in the shot.

The first two metrics evaluate the parsing independent of move durations. They respectively assess the syntactical correctness of the *ASAP* log and the precision of the transitions between one move (or hold) and another. The third metric emphasizes the amount of time that *ASAP* is right (or wrong), so that long moves have more weight than short moves. Which of these metrics is more appropriate is application dependent. We therefore present results for all. Figure 4 summarizes *ASAP*'s performance using a windowed average of ± 15 shots.

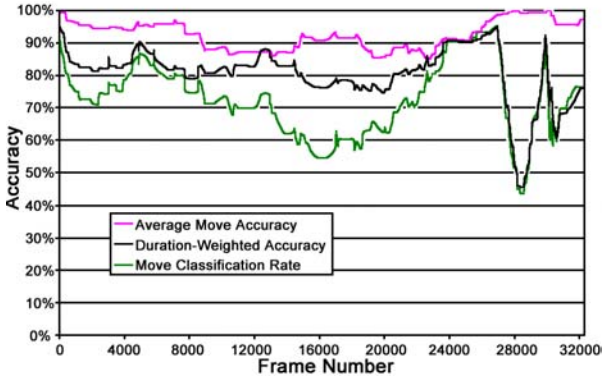

Figure 4. Move classification performance.

Overall *ASAP* correctly identified 71.3% of camera moves within the shots, including complex camerawork with multi-directional movement (e.g., a zoom in that pans left and tilts down). There are two areas where parsing is less accurate – frames 14,892-18,288 and 27,726-29,025. In the former heavy occlusion adversely affected accuracy. In the latter, the very short duration of shots coupled with the small scale of camera movement in two shots that are cut between several

times caused errors (discounting these two repeating shots alone raises overall accuracy by ≈5%).

When *ASAP* correctly classified a camera move, it detected start and end points with an overall average move accuracy of nearly 95%. As an absolute measure this is a remarkable result. However, it should be noted that this reflects *overall accuracy* and does not take into account how *beneficial* the output would be to an end-user. Developing such a metric would require a survey of professionals and industry guidance.

ASAP is most accurate in charting start and end times of moves where there were low levels of subject motion or highly controlled movements (i.e., camera on a tripod in controlled conditions). Instances of handheld shots, multiple subject motion and particularly occlusion are more difficult for the system although it is quite robust, able to detect severe moves such as the snap zoom in shot 281 (where cars start coming around a turn).

One notable finding is that the 'feathering' of camera moves (i.e., the tapering of the start and end of the move to create a smooth, fluid motion) can cause frame accuracy errors as can shots with a low rate of movement (e.g., slow pans). This suggests adaptive variation of detection thresholds and is thus another area for future work.

In examining other sequences that posed problems, we identified several conditions that likely require a system to have a more formal model of visual perception. Camera moves that keep the subject static within frame as the subject physically moves can fail if the background does not have a clear pattern or texture (e.g., the clear sky in shot 10 or the unmarked tarmac in shot 56, where cars are being tracked as they slowly move). Likewise *ASAP* can have trouble distinguishing the direction of camera movement in shots where the dominant movement is not objectively clear. We believe that such errors are not unique to our perspective estimation approach but apply to other non-intelligent methods as well. Alternative camera motion classification systems were not available for direct comparison. We hope to include these in future work.

CONCLUSIONS

ASAP is a film and TV industry oriented video shot analysis and documentation tool that quickly and robustly creates logs and searchable databases of footage based on camera activity. We have shown that its cut detection is more robust than other current approaches and that it can parse complex camera movements from complex source footage. Future work will include incorporating motion segmentation capabilities to interpret object movement, the parsing of full camera movement (e.g., dolly moves) and developing *ASAP* as a plug-in for existing post-production tools.

REFERENCES

1. Seyler A, 1965, "Probability Distributions of Television Frame Differences", Proc. IEEE, 53, 355-366

2. Yeo B-L and Liu B, 1995, "Rapid Scene Analysis on Compressed Video", IEEE Trans Circ. and Sys. for Vid. Tech., v.5 no.6, 533-544

3. Boreczky S and Rowe L A, 1996, "Comparison of video shot boundary detection techniques", Storage & Retrieval for Image and Video Databases IV, SPIE 2670, 170-179

4. Lienhart R, 1999, "Comparison of Automatic Shot Boundary Detection Algorithms", Proc. Image and Video Processing VII, SPIE 3656-29.

5. Butz T and Thiran J-P, 2001, "Shot Boundary Detection with Mutual Information", Proc. ICIP 2001, 422-425

6. Patel N V and Sethi I K, 1997, "Video Shot Detection and Characterization for Video Databases", Pattern Recognition", Spec. Issue Multimedia, v.30 n.4, 583-592

7. Bouthemy P, Gelgon M and Ganansia F, 1999, "A Unified Approach to Shot Change Detection and Camera Motion Characterization", IEEE Trans Circ. and Sys. for Vid. Tech.., 9(7), 1030-1044

8. Syeda-Mahmood T and Srinivasan S, 2000, "Detecting Topical Events in Digital Video", Proc. ACM Multimedia 2000, 85-94

9. Rui Y, Gupta A and Acero A, 2000, "Automatically Extracting Highlights for TV Baseball Programs", Proc. ACM Multimedia 2000, 105-115

10. Ardebilian M, Chen L and Tu X, 2000, "Robust 3D Clue-Based Video Segmentation for Video Indexing", Jour. Vis. Comm.and Image Rep., v.11 no. 1, 58-79

11. Mateer J, 2003 (in press), "Developing Effective Test Sets and Metrics for Evaluating Automated Media Analysis Systems", Proc. ICME 2003

12. Robinson J A, 2003 (in press), "A Simplex Based Projective Transform Estimator", Proc. VIE 2003

13. Shamim M A and Robinson, J A, 2003 (in press), "Object-Based Video Coding by Global-to-Local Motion Segmentation", IEEE Trans Circ. and Sys. for Vid. Tech.

14. Shamim M A and Robinson J A, 2000, "Modified Binary Tree for Contour Coding and Its Performance Analysis", 3rd Sym. Wirel. Pers. Mult. Comms, 603-608

15. *ASAP* Classifier Output Example: http://www.elec.york.ac.uk/visual/asap/30fps7.html

16. Text Retrieval Conference Video Retrieval Evaluation (TRECVid): http://www-nlpir.nist.gov/projects/t01v/

17. Richards, R, 1992, "A Director's Method for Film and Television", Focal Press, Boston, USA

18. Katz S D, 1991, "Film Directing Shot by Shot", Michael Wiese Prods/Focal Press, Stoneham, USA

19. Mast G and Cohen M, 1992, "Film Theory and Criticism, 4th Ed.", Oxford Univ. Press, New York, USA

20. *CutDet*: http://www.informatik.uni-mannheim.de/informatik/pi4/projects/MoCA/downloads.html