Video, Vision and Graphics (2003)
P. Hall, P. Willis (Editors)

Semi-Automated Logging for Professional Media Applications

J W Mateer and J A Robinson

Media Engineering, Department of Electronics, University of York, York, UK

Abstract

We report a novel method for logging and annotating video footage specifically for professional post-production and archivist end users. SALSA – Semi-Automated Logging with Semantic Annotation – is a hybrid system that utilises automated footage analysis for cut detection and camera movement classification, in conjunction with a stenographic-like keyboard input system to enable the logging of higher-level semantic information. Output is presented in both standard printed log form, with the addition of mosaic visual representations of shots, and in a fully searchable database. Results from preliminary experiments are reported.

1. Introduction

Post-production personnel and media archivists share a number of common objectives when reviewing footage. Television and film editors look to obtain technical aspects – start and end time codes, duration of shots, framing, types of camera or subject movement within shots, etc – in preparation for editing. They also need to understand specific information concerning subject-based content – locations, props, specific actors and actions, to name but a few. Archivists often require even higher-level semantic information relating to theme, style and genre. Both groups demand that the information gathered is accurate and presented in a consistent form. This often means the use of standard Hollywood nomenclature (e.g., "Medium Two Shot", "Pan Right to Close-up", etc.[19]) to characterise filmmaking attributes as well as semantic descriptions of content (e.g., "Interior Bar – Reeves drunkenly trips on a chair, spilling his drink" for narrative, "Train derailment – diesel locomotive lying on its side being examined by a salvage crew" for documentary, etc.). The goal is to enable quick access to specific shots and sequences without having to revisit the footage.

For the past several years there has been a significant amount of work undertaken to create fully automated video indexing and media analysis systems to address these needs. Although techniques have advanced and technologies have matured we are still a long way from having a computer adequately and accurately characterise the full content of a film or video. Indeed, research into the extraction of high-level semantic information is very much in its infancy. There are, however, certain footage attributes that can be obtained reliably and consistently, including shot boundary location and descriptions of camera movement. The automated extraction of this information could be of great benefit and time savings to practitioners yet such technology has barely been implemented in professional systems. Little has been written concerning how best to meet the present needs of these end-users whilst fully automated technologies evolve.

We have created *SALSA* – a *Semi-Automated Logging with Semantic Annotation* program – with the aim of determining whether some aspects of automated content analysis can be of immediate use to post-production and archivist users. *SALSA* is a hybrid system that combines an automated footage analysis system, to extract technical information from source material, with a streamlined text-based annotation input system based in part on stenographic models of interaction. Output consists of a combination of conventional log form, utilising time code and text annotation, with mosaic representations of each shot in addition to start and end frame thumbnails to provided a concise but complete print reference. A database is also automatically generated that enables direct searches by keywords, descriptions and other criteria. The objective is to enable faster, more accurate logging than is possible through current means.

2. Previous Work

There have been numerous studies into various aspects of automated media analysis techniques including shot boundary detection (such as described by Boreczky and Rowe[1] and Lienhart[2]), camera movement classification (Patel and Sethi[3] and Bouthemy et al.[4], for example) and the extraction of semantic information (Snoek and Worring[5] present a worthwhile broad review). Research into semi-automated editing tools is much more limited (Girgensohn et al.[6] is perhaps the most relevant to this paper) suggesting this is an area for further exploration. A number of commercial products, including *VideoLogger*[7], *The Executive Producer*[8] and *SceneStealer*[9], have been specifically designed for the logging of footage and include some limited automated cut detection capabilities. Few studies have been conducted into the specific needs of post-production and archivist end-users although Mateer[10] does provide a detailed description of important considerations of automated systems targeted for these groups. There has been extensive research into the generation and application of mosaics. These include significant contributions by Szeliski[11], Mann and Picard[12], Peleg and Herman[13], and Davis[14].

3. Method: *SALSA – Semi-Automated Logging with Semantic Annotation*

The automation of logging for archiving or post-production represents the 'Holy Grail' of automated media analysis. Unfortunately such a system is still a long way off, as stated above. However robust technologies do now exist that can be used to at least streamline and speed up the process. Used in conjunction with some user input, automated parsing systems should enable great time savings with no loss in logging accuracy.

One of the most time consuming tasks in manual logging or indexing is the determination of the exact start and end frame of a shot or camera movement. This typically requires running footage back and forth through a playback system, noting the time code of the event. Whilst professional editing systems make this a relatively simple matter, often logging can only take place using less precise VCRs making this a tedious process. Clearly the automation of this process could be a valuable time saver.

SALSA combines proven automated media analysis methods with an enhanced input system to create a semi-intelligent logging tool. It consists of five basic components: an automated shot boundary and camera movement parsing system, a keyboard input based annotation system, a mosaic generation system, a log output system and a database generation system.

3.1 Parsing System

The parsing engine is based directly on *ASAP*, our automated shot analysis program[15]. It 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.

3.1.1 Camera Motion Estimator

We use a fast, high-accuracy, perspective estimator developed for image mosaicing and registration in augmented reality[16]. The estimator uses simplex minimization of a disparity function calculated over a mesh of samples taken from the picture (described in detail by Robinson[17]). In comparison tests with other perspective estimators, it performs as well as the state of the art but up to 30 times faster than its competitors. This estimator has been used for object-based video analysis and coding[18], but *SALSA* and *ASAP* only use the output of eight perspective transform parameters, along with a single measure of disparity, for input to the movement parser.

3.1.2 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.

3.1.3 Movement Parser

The movement parser clusters consistent movements over consecutive frames into tentative zooms, pans and tilts. It also detects cuts. While there are several methods for cut detection from both raw and coded video[1, 2], we are able to use the output of our motion estimator directly. 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 easily detected from translation parameters, and zooms from a combination of the scale/rotation matrix entries in the perspective transform. It is also possible to detect and quantify camera roll, though this is such an unusual movement that we do not parse it.

Having divided the stream of camera movements provisionally into zooms, pans and tilts (which may happen in parallel), the parser applies a second level of analysis. If zooms are 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.

3.1.4 Processing Speed

At maximum accuracy the global perspective estimation algorithm used processes a pair of 720x560 frames in about 1.6s on a 2 GHz Pentium IV. Through control of a speed/accuracy parameter, this can be accelerated to below 140ms per frame pair.

We are able to achieve a low average processing time by applying *ASAP* 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, *SALSA/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 the program at an average rate of 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. In the experiments reported below, ASAP was run at full accuracy (not hierarchically), so the processing time was approximately 1.6s per frame.

3.1.5 Cut Detection and Move Classification Accuracy

Media professionals require cut detection to be truly frame accurate. Straightforward measurement is therefore possible in terms of missed and erroneously flagged cuts. Any cut that is not frame accurate should be counted as two mistakes: a completely missed cut, plus an additional false cut. Overall accuracy is given by

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

We have previously compared *ASAP* against established histogram-based methods (specifically *CutDet*[21]) to directly gauge relative performance in cut detection[15]. Several trials were run using different thresholds to determine optimal settings and compare areas of strength and weakness in both systems using rigorous sample footage chosen with principled criteria[10]. Examining results obtained using the optimal settings for both systems, *ASAP*'s score of 95.9% overall compares very favourably with *CutDet*'s best result of 85.2%. Looking at the areas where the systems failed it is clear that *ASAP* is much better able to cope with occlusion, failing in only one instance. Figure 1 summarises the results.

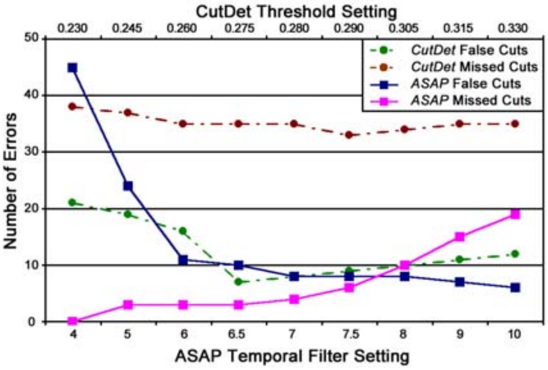


Figure 1: *Cut detection performance.*

The classification of camera movement is not typically regarded as a frame accurate measurement[10], however, for testing purposes we treat it as such. Previously we have tested *ASAP*'s camera move characterization and camera move frame accuracy using a programme that took *ASAP*'s output and compared it to an expert's hand log[15]. Overall the system correctly identified 71.3% of camera moves from an extreme case test set that included complex camerawork with multi-directional movement (e.g., a zoom in that pans left and tilts down). Results are summarized in Figure 2 using a windowed average of ± 15 shots.

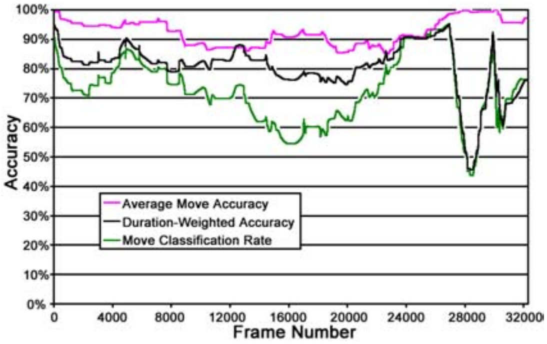


Figure 2. *Camera move classification performance.*

Clearly our approach to cut detection is effective and accurate enough for professional needs. However, although the performance of the move classifier is encouraging (given the difficulty of the trial) it could not be considered as an indication that the system is presently ready for unattended use (an area of our ongoing work). However, in the semi-automated context of *SALSA*, where user input is a component of log creation, this accuracy level is acceptable as any errors can be corrected during the entering of other shot information. Although not ideal, this level of accuracy does still indicate that substantial time savings can be made given the system is correct a significant majority of the time. Preliminary experiments (below) support this claim.

3.2 Annotation System

The description of shot framings and basic content using Hollywood nomenclature can be broken down into a few keys terms that can describe the vast majority of cases. For example, common framings are described by Katz[19] as 'close-up', 'close', 'medium' and 'wide' shots, with modifiers – such as 'extreme', 'tight', 'loose', etc. – used to further refine the description. Likewise, grouping content is often characterised in the same way though the use of 'single', 'two-shot', 'three-shot', 'group shot' and 'crowd shot'. As a result it should be possible to create a stenographic model of input so that users do not need to repeatedly type these descriptions.

SALSA uses dedicated keys to represent the most common classifications (as listed above) as well as a standard keyboard input system to enable unconstrained descriptions of higher-level semantic content. Figure 3. shows the current configuration.

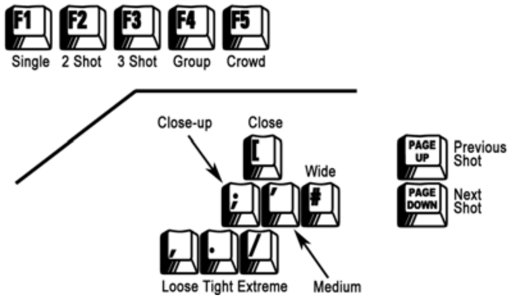


Figure 3: *Key mappings.*

These specific keys were chosen to enable an easy two-handed entry approach without the use of shift, alt or control keys, minimizing crossover and enabling easy movement for fast text entry. The optimal design for the layout has yet to be determined as usability tests are ongoing. However, it is already apparent from preliminary experiments that the stenographic model provides an appreciable time savings over straight keyboard entry alone.

3.2.1 Human-Computer Interface

Given *SALSA* is targeted to meet the needs of professional end users, we have designed the interface of the system using a layout and control methodology familiar to this group. The main entry window displays the output from the automatic parsing system in standard industry format, specifying shot number, start time, end time, duration and a breakdown any camera movements, with their respective starts, ends and durations. This log can be appended with descriptive information using the hot key system (described above) as well as with standard keyboard input for entering more detailed information. Three additional windows showing the actual start frame, end frame and the full running shot (with progress bar) are also utilised. This gives the user both quick and detailed references from which to generate higher-level semantic information. The current implementation is still not highly refined but is adequate for testing purposes. Sample screenshots can be found in Appendix B. Refinement of the approach is intended in future work.

3.3 Mosaic Generation System

The projective transform estimator used in *ASAP* and *SALSA* can be rerun on shots where the movement is simple, to generate an image mosaic. With the addition of start and end frame borders and the path of frames centres, the shot mosaic provides a closer analogue to a storyboard than a simple keyframe. A green bounding box denotes the start frame, shown in perspective in relation to the scene. A blue line indicates the direction of motion, linking centre points; a straight line would show that the movement was smooth whereas an irregular line would denote shake in the movement. A red bounding box is used to represent the end frame, also in perspective. The mosaic itself is centred on the middle frame of the sequence to minimize the overall distortion. Whether this is the best method for display is unclear and a topic for future work. Figure 4 shows an example mosaic.

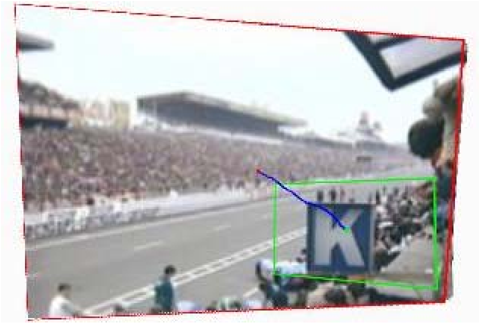


Figure 4: *Sample mosaic of zoom out, pan left.*

The use of mosaics was chosen to provide a succinct, intuitive visual description that enables users to determine all location-based attributes captured by a shot. This is particularly important if there are questions as to the viability of using a particular shot for editing. The indication of perspective does represent a departure from the typical manner of displaying footage content. Indeed, it may require some users to learn to 'see' in a way to which they are unaccustomed. However, our initial trials indicate that this is likely not a major issue and that users begin to see new benefits from the system. For example, if a studio camera tilted up very briefly, a boom microphone could appear in the mosaic even if the incursion lasted for a very short time. Likewise, in many contexts it is immediately possible to identify whether light stands, cables or other equipment are visible without having to review footage. This allows better quality control with minimal time penalty. Trials to date have been quite small so further testing is needed validate this model.

3.4 Log Generation and Output

SALSA uses output from the movement parser to present the data in several forms. These include a shot log that includes common information in standard industry format (e.g., SMPTE time code) such as start point, end point and duration of all shots and all camera movements contained within each shot. Thumbnails of start and end frames are extracted from the source material and rescaled for inclusion. If there is camera movement, these are placed on the left and right, respectively, of the generated mosaic if the predominant move is a tilt or above and beneath the mosaic, respectively, if the predominant move is a pan. A detailed example is shown in Appendix A.

3.5 Database Generation

SALSA creates database objects from both the extracted information and the user input. This enables full random access searching for any technical or semantic attribute, which can provide quick access for archivists and new creative options for editors. For example, we took two shots from the music video *Stargazer*[20] and asked *SALSA* to find the best sequence from a different roll of footage that could be spliced between them, accurately matching the speed of motion of the first shot (a tracking pan left) at its beginning, and that of the second shot (a tilt down) at its end, thus 'bridging' the two fluidly. The result (with dissolves automatically inserted between the shots) is summarized in Figure 5, which shows every fifth frame of the output sequence. As manually generated footage logs typically do not describe the precise *rate* of camera movement, creating a comparable sequence using traditional methods would be time consuming. Matching motions would not only have to be located, but visually compared to ensure the desired smooth editing flow. The ease and speed with which *SALSA* can suggest and create such sequences could prove valuable to practitioners.

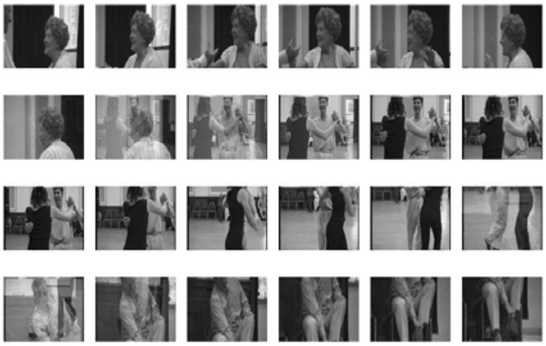


Figure 5. *Example of automated shot bridging.*

4. Preliminary Results

To date, we have conducted one trial to test the viability and the basic effectiveness of our *SALSA* approach. This test was by no means exhaustive and simply serves as a means to prove the concept; further testing is clearly needed.

In this trial, an expert editor was asked to log the first 75 shots (~7 minutes) of the film *Le Mans*, twice – first using the system and second using a non-linear editing system with a word processor. This order was chosen to minimize any advantage that might be gained through familiarity with the material; the bias in this trial favours manual entry. Each task was timed to the nearest minute. The subject was asked to characterise each shot fully – including noting principal characters, locations, objects, movements, etc. – as if he were preparing a logging for editing a narrative piece. Note we are not presently including the processing time of the footage as it is an unattended event and this trial was simply to get a sense of the impact on user time requirements.

Using *SALSA* the expert logged the test sequence in 46 minutes. This compares very favourably with the 95 minutes he took when doing the task manually. As predicted, the main time savings appear to come from the automation of logging cuts and camera moves. The stenographic approach to framing classification appears to also have an impact although it was not quantifiable given the limited scope and simple design of this experiment. Although this trial is far from conclusive, we believe the significant time savings obtained does indicate that this approach is highly effective. Larger, more rigorous trials are planned.

5. Conclusions

In this paper we have described a novel method for the semi-automated logging of video and film footage. The application and interrelation of the automated media analysis system, keyboard based annotation system, mosaic generation system, log output system and database generation system were presented. The rationale behind the use of mosaic imagery was described as well as the results of a preliminary trial. Our approach appears to have significant potential although it is recognised that more rigorous tests are needed and that the subsystems need to be further refined. These are areas for future work.

Acknowledgements

We would like to thank Zhan Peng for his assistance in recoding the mosaic system to enable the clear labelling of start and end frames as well as facilitating their inclusion into the output system.

References

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

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

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

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

5. C. G. M. Snoek and M. Worring, "Multimodal Video Indexing: A Review of the State-of-the-Art", *Multimedia Tools and Applications* (in press) 2003.

6. A. Girgensohn, J. Boreczky et al., "A Semi-automatic Approach to Home Video Editing", *Proc. of UIST'2000: The 13th annual ACM symposium on on User interface software and technology*. 2000. San Diego, CA: ACM. pp. 81-89, 2000.

7. *VideoLogger*, Virage, Inc., http://www.virage.com/

8. *The Executive Producer* (*TEPX*), ImageMine, Inc. http://www.imagineproducts.com/TEPX.htm

9. *SceneStealer*, Dubner, Inc., http://www.dubner.com

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

11. R. Szeliski, "Image Mosaicing for Tele-Reality Applications", Digital Equipment Corporation Cambridge Research Laboratory, Technical Report CRL 94/2, May 1994.

12. S. Mann, R. Picard, "Video Orbits of the Projective Group: A simple approach to featureless estimation of parameters", IEEE Trans Image Proc, Vol 6, No 9, 1997, pp 1281-1295.

13. S. Peleg, J. Herman, "Panoramic Mosaics with VideoBrush, "DARPA Image Understanding Workshop, May 1997, pp 261-264.

14. J. Davis, "Mosaics of Scenes with Moving Objects", Proc CVPR'98, June 1998.

15. J. W. Mateer and J. A. Robinson "Robust Automated Footage Analysis for Professional Media Applications," *Proc. VIE 2003*, (in press), 2003.

16. M. A. Shamim and J. A. Robinson, "Object-Based Video Coding by Global-to-Local Motion Segmentation", *IEEE Trans Circ. and Sys. for Vid. Tech.*, Vol 12, No 12, pp 1106-1116, December 2002.

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

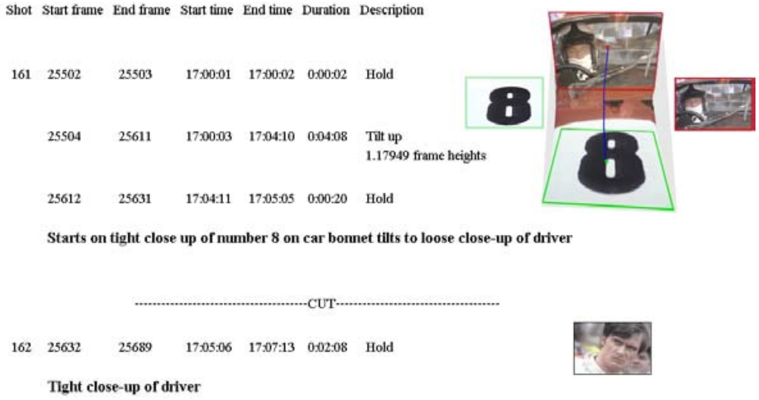
18. M. A. Shamim and J. A. Robinson, "Modified Binary Tree for Contour Coding and Its Performance Analysis", *3rd Sym. Wireless Pers. Mult. Comms*, 603-608, 2000.

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

20. J. W. Mateer, The Zephyrs, "Stargazer", broadcast music video, Southpaw Records/Play It Again Sam Music, London, 2000.

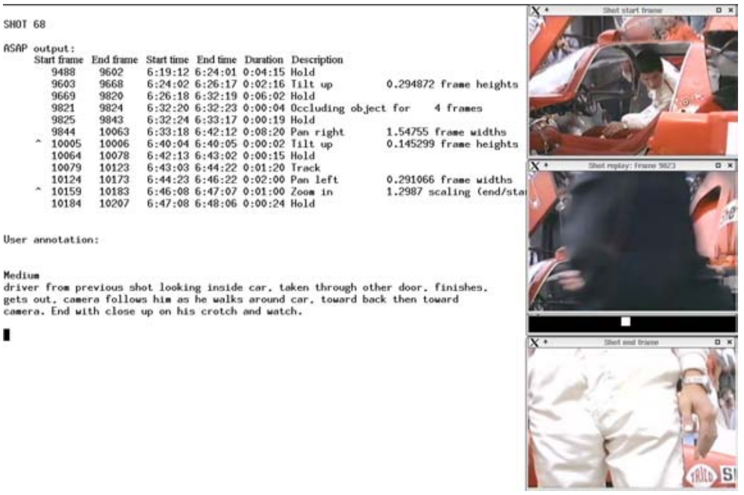
21. Lienhart's *CutDet*: http://www.informatik.uni-mannheim.de/informatik/pi4/projects/MoCA/downloads.html

Appendix A: Sample *SALSA* Log

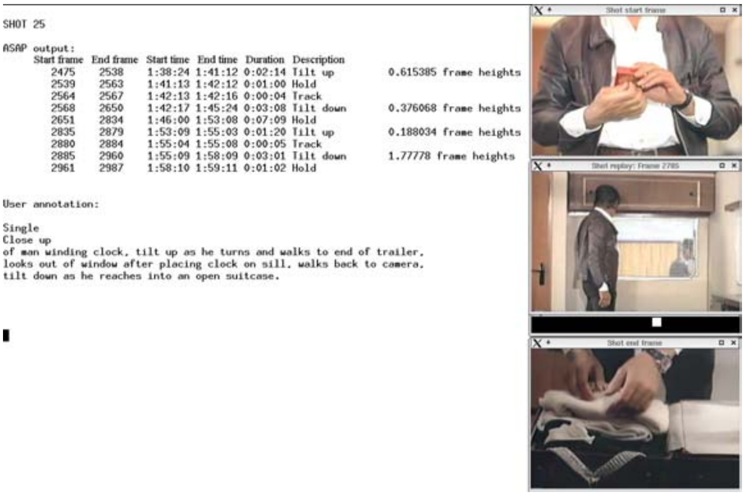


```
Shot  Start frame  End frame  Start time  End time  Duration  Description

161   25502        25503      17:00:01    17:00:02  0:00:02   Hold

      25504        25611      17:00:03    17:04:10  0:04:08   Tilt up
                                                              1.17949 frame heights

      25612        25631      17:04:11    17:05:05  0:00:20   Hold

      Starts on tight close up of number 8 on car bonnet tilts to loose close-up of driver


                   ---------------------------------------CUT------------------------------------

162   25632        25689      17:05:06    17:07:13  0:02:08   Hold

      Tight close-up of driver
```

Appendix B: Sample *SALSA* Screen Shots



```
SHOT 68

ASAP output:
     Start frame  End frame  Start time End time Duration Description
        9488      9602      6:19:12 6:24:01 0:04:15 Hold
        9603      9668      6:24:02 6:26:17 0:02:16 Tilt up        0.294872 frame heights
        9669      9820      6:26:18 6:32:19 0:06:02 Hold
        9821      9824      6:32:20 6:32:23 0:00:04 Occluding object for    4 frames
        9825      9843      6:32:24 6:33:17 0:00:19 Hold
        9844      10063     6:33:18 6:42:12 0:08:20 Pan right      1.54755 frame widths
     ^  10005     10006     6:40:04 6:40:05 0:00:02 Tilt up        0.145299 frame heights
        10064     10078     6:42:13 6:43:02 0:00:15 Hold
        10079     10123     6:43:03 6:44:22 0:01:20 Track
        10124     10173     6:44:23 6:46:22 0:02:00 Pan left       0.291066 frame widths
     ^  10159     10183     6:46:08 6:47:07 0:01:00 Zoom in        1.2987 scaling (end/sta
        10184     10207     6:47:08 6:48:06 0:00:24 Hold

User annotation:


Medium
driver from previous shot looking inside car, taken through other door, finishes,
gets out, camera follows him as he walks around car, toward back then toward
camera. End with close up on his crotch and watch.
```



```
SHOT 25

ASAP output:
     Start frame  End frame  Start time End time Duration Description
        2475      2538      1:38:24 1:41:12 0:02:14 Tilt up        0.615385 frame heights
        2539      2563      1:41:13 1:42:12 0:01:00 Hold
        2564      2567      1:42:13 1:42:16 0:00:04 Track
        2568      2650      1:42:17 1:45:24 0:03:08 Tilt down      0.376068 frame heights
        2651      2834      1:46:00 1:53:08 0:07:09 Hold
        2835      2879      1:53:09 1:55:03 0:01:20 Tilt up        0.188034 frame heights
        2880      2884      1:55:04 1:55:08 0:00:05 Track
        2885      2960      1:55:09 1:58:09 0:03:01 Tilt down      1.77778 frame heights
        2961      2987      1:58:10 1:59:11 0:01:02 Hold

User annotation:

Single
Close up
of man winding clock, tilt up as he turns and walks to end of trailer,
looks out of window after placing clock on sill, walks back to camera,
tilt down as he reaches into an open suitcase.
```