Automated Reverse Storyboarding

R.D. Dony[1], J.W. Mateer[2], J.A. Robinson[2]
[1]School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada
[2]Department of Electronics, University of York, Heslington, York YO10 5DD, United Kingdom

Abstract
Storyboarding is a standard method for visual summarization of shots in film and video preproduction. Reverse storyboarding is the generation of similar visualizations from existing footage. We identify the key attributes of preproduction storyboards then develop computational techniques that extract corresponding features from video, render them appropriately, then composite them into a single storyboard image. The result succinctly represents background composition, foreground object appearance and motion, and camera motion. For tracking shots, we show that the visual representation conveys all the essential elements of shot composition.

1 Introduction

Visual summaries play an important role in the production and analysis of media. Practitioners, researchers and archivists all demand that the information presented is accurate and described in a consistent form using common metaphors derived from industry nomenclature. The goal is to enable quick access to details of specific shots or sequences without having to view the footage itself.

In the media production industry, visual summarization is typically achieved through *storyboards*. Storyboards are drawn during preproduction then used throughout production and postproduction in tasks like set design, location lighting and image compositing. They provide for all participants a common reference to the "vision" of the piece. Shorthand descriptions of all important visual components of each shot provide clear and accurate depictions of motion sequences in static form. These include specific methods of describing camera or subject movement through the use of various drawing techniques. While the term storyboard has been applied in the context of automated media analysis to a sequence of consecutive still images extracted from a film or video programme, the representations traditionally used in the production industry are much richer. Our usage in this paper corresponds to film production storyboarding: *i.e.*, we seek to describe the temporal evolution of a shot through a single picture using rich visual cues.

Storyboards incorporate the following types of information:

1. Composition of the shot, including start, end and notable intermediate camera positions
2. General appearance (perhaps sketchy) of background and foreground elements showing salient features
3. Depiction of object movement
4. Depiction of camera movement

The techniques used for depicting object movement are:

Onion skins show multiple instances of a subject that indicate its intermediate positions between the start and end frames. Figure 1 is an example.

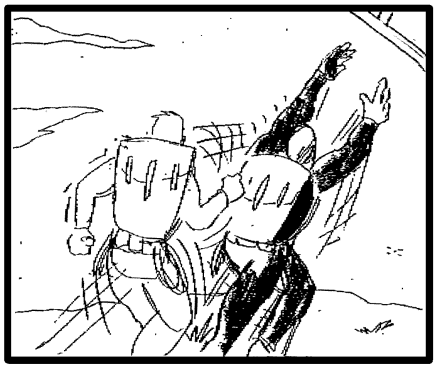

Figure 1: Onion skin effect conveying motion

Streaks are lines that show the trajectory of a moving object. The long arcs in the direction of motion in figure 2 are streaks.

Trail lines are repetitions of the trailing edge of a moving object. (Trail lines are sometimes called "ghosts": we avoid this usage because animators sometimes refer to onion skins as ghosts.) Trail lines are often used with streaks (to which they are roughly perpendicular) as in figure 2.

Arrows are sometimes used to emphasize the direction of motion

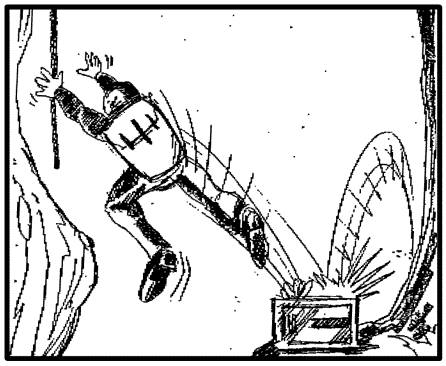

Figure 2: Motion trail lines to show speed and motion

The techniques used for depicting camera movement are:

Mosaics are storyboards that show the full panorama viewed in a pan or tilt. Again, frame outlines for start and end frames are drawn. See figure 3.

Figure 3: Mosaic storyboard showing full scope of shot

Arrows are sometimes used to emphasize the direction of camera motion.

Field cuts are frame outlines drawn on the storyboard indicating an initial or final zoom position [12], an example of which is shown in figure 4.

Given the effectiveness of storyboarding in creative development, it follows that similar motion metaphors may prove valuable in the creation of visual summaries of existing film and video sequences. The use of established visual conventions should mean that summaries created in this manner are more intuitive and more easily interpreted than those from other video summarisation techniques.

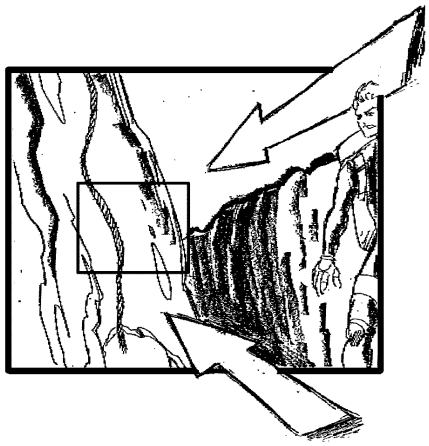

Figure 4: Field cut showing camera motion

This paper describes new work in developing image processing methods applied to footage-based storyboards that incorporate some of the same techniques used by industry storyboard artists to provide more effective descriptions that are more readily accessible to these user groups.

2 Prior Work

Because the number of still frames contained in even a short footage sequence is large, significant work has been carried out to determine more efficient means of visually describing content. Many of these studies have focused on means of determining which particular frames, or "keyframes," best convey a sequence [8] and ways of presenting those frames in more intuitive ways including the determination and subsequent larger display of dominant frames [27] and the use of the Japanese comic book-inspired Manga layouts described in [6]. Other work has looked at different forms of video abstraction and summarization [13], [10]. However, there has been only limited research on the way media practitioners create and utilise storyboards with a view toward creating systems for automated summarisation [14]. Likewise, research into the use of mosaics in an industry storyboarding context is very limited. The generation and overlaying of cartoon-style motion cues, widely used by industry storyboard artists, has been examined [4] but relies on the user identifying specific objects or areas of interest within frames.

Our interest in reverse storyboarding arose from development of a system for semi-automated footage logging for archiving or post-production [18] (Semi-Automated Logging with Semantic Annotation, or *SALSA*). *SALSA* provides two mechanisms for shot visualization: the keyframe and the mosaic. *SALSA*'s extraction of keyframes from static holds is intelligent. If the hold is interrupted by the passage of a transient occluding object, *SALSA* will avoid the frames with the object in its choice of keyframe. Similarly *SALSA* can mosaic shots

in which there is a pan, tilt or zoom, registering successfully even in the presence of moderate foreground motion. *SALSA*'s output mosaics have start and end frames outlined by bounding boxes, and the frame centres are connected by a trajectory line. They therefore summarize camera motion is a similar way to storyboards. However, they do not represent object motion, except artifactually through moving objects that may appear smeared in the mosaic.

The adequacy of keyframes plus mosaics to represent footage is content dependent. The 20 minute sequence from the 1971 feature film *Le Mans* [22] used in our previous work [18] yields the *SALSA* log, an example portion of which is shown in figure 5. The entire log of the shot analysis portion of *SALSA* for *Le Mans* can be found at [17].

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

Figure 5: SALSA example output from *Le Mans*

Figure 6 illustrates an example mosaic with the camera movement, a zoom out with minor motion, shown. In the *Le Mans* trial sequence the shots may be informally categorized as follows:

- Holds with little significant foreground motion except for transient occluding objects: 164 shots.
- Holds with significant foreground motion: 50 shots.
- Shots with a mosaic-able camera move and little other significant motion: 40 shots.
- Shots with a camera move and little other significant motion that are not immediately mosaic-able (i.e. dolly moves): 2 shots.
- Tracking shots: 17 shots.
- Tracking shots wrongly interpreted by *SALSA* as holds: 2 shots.
- Other shots involving both camera movement and significant movement of foreground objects: 11 shots.

It cannot be overemphasized that the relative number of different types of shots is a function of film language and the vision of the director. This particular sequence includes a montage that accounts for a high proportion of the 164 simple holds. It should be noted this sequence is an extreme example chosen for its complexity. The holds

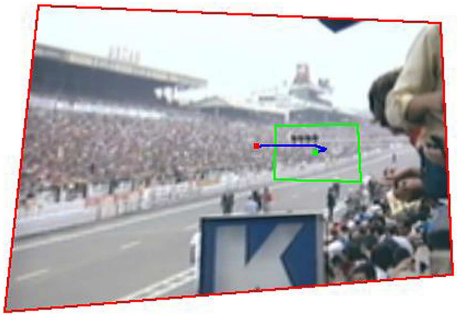

Figure 6: SALSA example output showing mosaic with start frame box, end frame box, and camera path all overlayed

and moves it contains, with significant object motion, are more complicated than may be encountered in many types of footage.

As would be expected from the figures above, *SALSA* as previously reported can efficiently summarize about two thirds of the *Le Mans* sequence with keyframes for the simple holds and mosaics for the simple moves. Of the remainder, the shots with significant foreground object motion vary greatly in complexity. Some involve the interaction of several objects translating and rotating in 3-space. Some illustrate the difficulty of automatically making good representation choices. For example in one shot, most of the measurable movement is in one part of the scene, but the story of the shot is in the movement of one person's eyes, which occupy a tiny proportion of the frame.

To extend the coverage of *SALSA*'s shot summarization facilities, and take a significant step towards a rich reverse storyboarding system, we have therefore turned to the representation of tracking shots. In these the camera follows a subject as it moves in the scene. Provided the object is not too large, the camera move is correctly detected and the frames mosaiced correctly with respect to the background. (If the object is too large in the frame, the result is a tracking shot incorrectly interpreted as a hold.) The framing of the subject in a tracking shot takes away the semantic problem of moving objects mentioned in the previous paragraph: in a tracking shot, the object being followed is the most important. Although good representation of tracking shots only adds 17 more visualizations to *SALSA*'s log of *Le Mans*, it represents the most tractable extension. The problem we are left with is how to represent the movement of the camera and the subject in a meaningful way.

3 Methods

Storyboard artists use a number of techniques as discussed in section 1 to convey the important elements of a shot including background, foreground elements, object motion, and camera motion. Our goal is to extract these elements from production video footage and convey them in a manner consistent with a traditional storyboard presentation. We must therefore identify the artists' techniques we wish to imitate for depicting these elements. Once these are identified, the next task becomes the development of, first, efficient procedures for processing the dig-

ital footage to extract the required elements and then of visualisation techniques to portray these elements consistent with the chosen artistic techniques.

We address each of the four elements in turn below.

3.1 Background

The background is a sketch of the static elements of the scene upon which the moving foreground objects can be drawn. For the purposes of storyboarding, we wish to create a mosaic from the video footage to represent only the background. Therefore it is necessary to remove the moving objects as they will be added to the storyboard separately.

The generation of mosaics from moving video is an extensively researched topic [5, 9, 11, 15, 20, 26]. Most techniques involve two stages: the estimation of the projective transforms from the video sequence that map frame co-ordinates to mosaic co-ordinates, and a method of combining the frame images using the transforms.

There are numerous methods in the literature for estimating the projective transforms, *e.g.*, [9, 15]. For this work, we employ the projective estimator used in *SALSA*. It is a fast, highly-accurate, estimator previously developed for image mosaicing and registration in augmented reality [23]. The estimator uses simplex minimization of a disparity function calculated over a mesh of samples taken from the picture. This estimator has been used for object-based video analysis and coding [24, 25], but the method only uses the output of eight perspective transform parameters to calculate the correspondence between frames. We then accumulate these parameters to calculate a set of transformations, $\mathcal{P}_i$, one for each frame i, that maps the frame co-ordinate $\vec{x}_{f_i} = (x_{f_i}, y_{f_i})^T$ to the mosaic co-ordinate $\vec{x}_m = (x_m, y_m)^T$ as

$$\vec{x}_m = \mathcal{P}_i \vec{x}_{f_i} \tag{1}$$

The inverse transformation is simply $\mathcal{P}_i^{-1}$.

The problem now remains of how to combine the frame images under the set of transformations to produce an appropriate mosaic. Unfortunately, when examining previous work, most of the applications for such work has focused on tele-reality, virtual reality environments, and panoramic composites for consumer photography. Research into the use of mosaics in an industry storyboarding context is limited. Many of the approaches to mosaicing assume a static scene and therefore do not explicitly take into consideration moving objects. Others do consider motion within the scene, but the goal is to produce a "pleasing" image. For storyboarding, the proper consideration of motion is crucial to the final representation.

We can classify the various frame combination methods into two general categories: sequential and statistical. For sequential methods, the most widely investigated of the two, the frames are combined in some order of presentation. For statistical methods, the statistics of the group of frame pixels corresponding to a location in the mosaic are examined. We may be able to devise an operator using such statistics that extracts only the background pixel value for incorporation into the mosaic.

We evaluate a number of methods from both approaches below. To illustrate the differences between the background mosaic construction methods being considered, we have chosen a tracking shot from music video for *Stargazer* [19]. Stills of every 10th frame of the 200 total are shown in figure 7. The video was shot in black and white PAL widescreen format. This results in an artifact whereby 16:9 aspect ratio shots are digitally encoded into 4:3 aspect ratio video frames, thus the images appear "squeezed". The camera pans right, tracking a man and a woman walking to the right. They are initially with a couple standing still. As they walk to the right, the man exits through the door seen in the middle of the shot. The woman eventually stops and crouches down to place her handbag on the floor. The camera continues to pan right until the last frame showing the woman now crouching down and two new people just having entered the field of view - a seated woman and a man walking left.

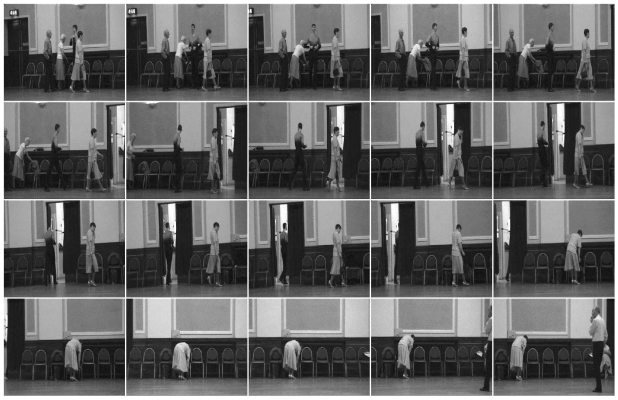

Figure 7: Tracking shot used for evaluating mosacing methods

The *Stargazer* tracking shot was chosen as an exemplar for this evaluation because it illustrates a range of features encountered in tracking shots, including the movement of multiple independent objects, the occurence of objects that appear only in one or two end frames, and the disappearance of objects (in this case because someone walks through a door). The commentary below refers directly to the *Stargazer* example, but the conclusions apply generally and have been observed consistently on a test set of a number of tracking shots of diverse kinds.

3.1.1 Sequential Mosaic Generation

Lapped Overwrite To illustrate the characteristics of sequential methods we first consider a simplistic approach where the mosaic is generated by overwriting the mosaic pixels of those of each successively new frame. This method, while being easy to implement, can introduce minor discontinuities at the edges of the frames. By modifying the algorithm to use a weighted overlap at the frame edges such edge artifacts can be reduced.

Figure 8 shows the result for the lapped overwrite approach on the above shot. The resulting mosaic appears quite well-formed without apparent artifacts. Since most of the pixels come from near the edges of the individual frames, objects that never leave the field of view do not ap-

pear in the mosaic. For example, we never see the walking man who exits through the centre door. However, moving objects which enter or leave the field of view are recorded in the mosaic. If an object were to move with the camera and remain at the trailing edge of the field of view during a portion of the shot, the object would appear as a long, drawn-out smudge. The only region in the mosaic without any possibility of such distortion is the last frame. It appears completely intact in the right-hand side of the mosaic, with all subjects appearing irrespective of their previous motion. The result for this approach can vary significantly depending on the order in which the frames are processed, for example in reverse order or middle outwards. Further, the method is not explicitly designed to extract only the static background. As a result, the output is not consistently suitable for our work.

Figure 8: Result of lapped overwrite mosaic generation

Optimal Boundary Davis [5] composites images by stitching a new, registered, frame into an existing mosaic along a data-determined, usually irregular, boundary. The boundary is computed by finding the absolute differences between the new frame and the mosaic at every overlapping point, then using Dijkstra's algorithm to define the minimum-total-difference path through each overlapping section. Points on one side of that boundary are taken from the old mosaic; points on the other side are taken from the new frame. There is no blending, simply a juxtaposing of frames. Davis shows that, in certain circumstances, the method can create effective mosaics of moving objects, because the objects are not blurred through blending.

Figure 9 shows the effect of applying Davis' method to the *Stargazer* sequence. Note that the method does indeed prevent blurring due to moving objects. However, as soon as the camera moves so that an object's position in the first frame goes out of shot, the object is re-rendered in the mosaic. For example, the walking man who exits through the middle door is depicted twice and so is the walking woman who ends up crouching down. Further, an interesting artifact is introduced in the left-most rendering of the walking man – his upper and lower bodies are somewhat shifted relative to each other. The final appearance of the mosaic can be altered if it is generated from last frame to first, or in some other order. Although some outputs are fortuitously similar to onion skin motion representation, others are not. As with simple overwriting and lapped joints, the Davis method, being order-sensitive, generates a background with unpredictable foreground content and is therefore also unsuitable for our purpose.

Figure 9: Result of Davis' optimal boundary mosaic generation

3.1.2 Statistical Mosaic Generation

The sequential methods are not explicitly designed to extract only the static background in the presence of moving foreground elements. It is not surprising, then, that such methods are inadequate for our work.

We wish to devise a mosaic generation method that consistently extracts only the background. To this end, we employ a statistical framework. Using the statistical properties of the set of frame pixels corresponding to a particular mosaic co-ordinate it may be possible to classify the intensities into two groups: static background and moving foreground. With such a classification the representative background intensity can be incorporated into the mosaic at each co-ordinate.

To begin, we define $\mathbf{F}_{\vec{x}_m}$ to be the set of all frame images whose frame co-ordinate resulting from the inverse transform of the mosaic co-ordinate, $\mathcal{P}_i^{-1}\vec{x}_m$, falls within the frame image, or more precisely

$$\mathbf{F}_{\vec{x}_m} = \{i \mid \mathcal{P}_i^{-1}\vec{x}_m \in \mathbf{X}_{f_i}\} \tag{2}$$

where $\mathbf{X}_{f_i}$ is the set of valid co-ordinates for frame i. Next, we define $\mathbf{I}_{\vec{x}_m}$ as the set of frame image values at the respective valid locations corresponding to the mosaic co-ordinate $\vec{x}_m$, that is,

$$\mathbf{I}_{\vec{x}_m} = \{I_{f_i}(\mathcal{P}_i^{-1}\vec{x}_m) \mid i \in \mathbf{F}_{\vec{x}_m}\} \tag{3}$$

In other works, the set $\mathbf{I}_{\vec{x}_m}$ contains the frame image intensities from locations that map onto the mosaic co-ordinate $\vec{x}_m$.

If no moving objects were to pass in front of the background at mosaic co-ordinate $\vec{x}_m$, the set $\mathbf{I}_{\vec{x}_m}$ would only contain the intensity value, I_B, of the background. If an object of intensity I_O were to occlude the background for a period of time, the set would contain both intensity values. If we were to assume that the object appears for a shorter time than the background (a not unreasonable condition since it follows from the definition of the background as being the static element of the shot that we expect some degree of permanence), then the number of background intensities in $\mathbf{I}_{\vec{x}_m}$ is larger than those of the object. So, we wish to use an operator which returns the value of the intensity that occurs most frequently in the set.

Mode Based on the above argument, the mode of the set appears to be an appropriate statistic for our use. Therefore a mode-based mosaic can be calculated as

$$I_m(\vec{x}_m) = \arg\max_{I \in \mathbf{I}_{\vec{x}_m}} N_{\mathbf{I}_{\vec{x}_m}}(I) \tag{4}$$

where $N_{\mathbf{I}_{\vec{x}_m}}(I)$ is the number of times intensity I occurs in $\mathbf{I}_{\vec{x}_m}$ also referred to as the histogram. However, when the number of samples is small, the mode operator is very sensitive to noise. To reduce the effects of noise, the histogram is Gaussian smoothed.

Figure 10 shows the result of the mode mosaic generator. It successfully renders a mosaic that is mostly free of moving foreground objects as expected. For example, the crouching woman on the right has been partially removed. As well, the walking man on the left is nowhere to be seen. However where an object appears for a longer length of time, the mosaic is quite noisy. This is quite evident for the crouching woman and man on the right. Further, there are obvious distortions in the intensity values in the image.

Figure 10: Result of mode mosaic generation

In areas where a moving object appears as equally long as the background, for example in a region that appears only briefly in the shot, the set $\mathbf{I}_{\vec{x}_m}$ is bimodal with both intensity values being equally as likely. Under these conditions, the mode operator is unstable so in the presence of even a small degree of noise the median output randomly flips between the two different intensity values.

Median A more stable operator is the median of the set as it is more robust under noise. Where there is a clear distinction between the foreground and background, *i.e.*, there are more background than foreground intensities in the set, the median returns the most numerous intensity, the background. As the distribution becomes more bimodal, the value of the median tends toward the mean, even under a degree of noise. It therefore degrades gracefully as the distinction between foreground and background becomes unclear.

The result of the median mosaic generator is shown in figure 11. In regions where there are no moving objects or objects which appear briefly, the method produces a clean background as it was designed to. Where foreground objects dwell for a relatively longer period of time, the distortion evident in the mode image is gone. Instead, the objects may appear with some varying degree of transparency. For example there is a barely visible "ghost" of the man on the left since he dwelt there for only a short period of time. The crouching woman on the right is far more visible in the mosaic since she was relatively motionless at that location for a period of time.

Unlike the sequential methods above, both the statistical operators are designed to construct mosaics that include only the static background. Where moving objects occlude the background briefly, both are successful at rendering only the background. Under ideal conditions the

Figure 11: Result of median mosaic generation

mode operator appears to be the optimal operator for rendering a mosaic free of such moving objects. However, the median appears to be a more robust operator under real conditions since it introduces less obtrusive artifacts where the distinction between foreground and background is not clear.

Further, the statistical results have a very useful byproduct — intensities that are not part of the background are therefore from the foreground elements. In effect, we get the foreground identification for free.

3.2 Foreground Elements

We now turn to the second element in storyboards, the foreground elements.

One approach to identify moving foreground objects in a shot is to employ object tracking techniques [4]. This topic in computer vision has been extensively investigated [1, 2, 3, 7], especially in the context of human motion. The goal of such techniques typically is for measurement and modelling. Further, many require user assistance and can operate in only constrained or simplified environments. The goal of our work is to produce a visual representation of the shot in an automated manner. Therefore such techniques are not suitable for our purpose.

From the above discussion we see that the median operator which we choose for the creation of the background effectively gives us the foreground content at the same time. We use a measure of how confident we are that a frame pixel is from a foreground object by the absolute difference between its intensity value and the median calculated as

$$D_{\text{mot},f_i}(\vec{x}_{f_i}) = |I_{f_i}(\vec{x}_{f_i}) - \text{median}\ (\mathbf{I}_{\vec{x}_m})| \qquad (5)$$

where $\vec{x}_m = \mathcal{P}_i \vec{x}_{f_i}$. When the pixel location is part of the background, then $D_{\text{mot}} \approx 0$ and when it is part of a moving object, $D_{\text{mot}} > 0$. Now, if the moving object has a similar intensity at that location to the background at that same location, then D_{mot} will be small. However, since the goal is to produce a visual representation of the moving object, the small difference in intensities does not pose a problem as we will see later.

Figure 12 shows the resulting moving object images for the middle column of stills shown in figure 7. In general this method of object identification is quite successful despite its simplicity. Where there is a good contrast between a moving object and the background, the object appears quite clearly. As anticipated, there is little response where there is poor contrast. Two artifacts are of interest. In the bottom left frame, the woman appears twice – once

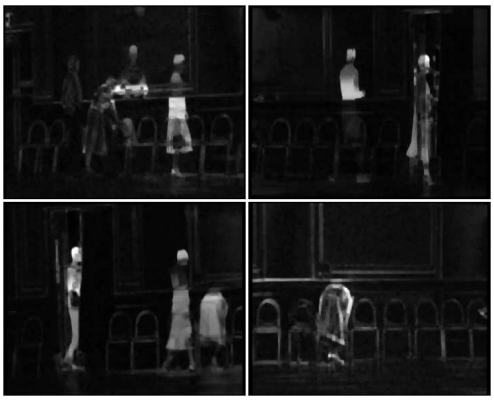

Figure 12: Moving object images for middle column of figure 7

as walking and once as crouching. Referring to the corresponding frame in figure 7 (third row, middle column), we clearly see that she appears only once as walking. The phantom image of her crouching is caused by the incorrect identification of the crouched figure as background as illustrated in figure 11. The second artifact is the presence of faint outlines of objects in the background such as chairs and wall frames. This is due to minor errors in the projective estimation algorithm.

3.3 Object Movement

Storyboard artists draw upon a number of techniques to convey motion within a shot that include, as described in section 1, onion skins, streaks, trail lines, and arrows. A good artist will tend to employ a limited number of these, typically one or two in combination, in a given board to convey succinctly and clearly the essence of the movement. The use of too many devices may introduce unnecessary clutter and confusion. For our work, then, we will focus on a subset of such cues for extraction and rendering.

Onion skins are an effective way of representing the object and its motion. We can make use of the above method of extracting foreground elements to create the multiple poses. In addition, artists commonly use trail lines behind the onion skin figures to further accentuate the motion. For our purposes, these two techniques will suffice. The additional inclusion of streaks is effectively redundant and will only clutter the final rendition. The use of arrows as a motion cue can be less expressive in some cases than other cues. Including arrows may be redundant and interfere with the clean portrayal of the other elements. Further, their placement sometimes requires sophisticated artistic judgment which is not easily copied by an automated system.

3.3.1 Onion Skins

The moving objects identified above in section 3.2 are overlayed on the background with varying degrees of transparency in the manner of the onion skin technique. The opacity of each pixel in the object overlay is proportional to the value found in equation 5. To differentiate these added moving objects from the background in colour images, the objects are shown in monochrome. We set the spacing between the versions as proportional to the over-all camera motion — the more the camera moves, the more versions we can include without introducing undue clutter.

3.3.2 Motion Trail Lines

To produce trailing motion lines behind the onion skin figures, we make use of the time difference between adjacent frames calculated as

$$D_{\text{dt},f_i}(\vec{x}_{f_i}) = \left| I_{f_{i-1}}(\mathcal{P}_{i-1}^{-1}\mathcal{P}_i\vec{x}_{f_i}) - I_{f_i}(\vec{x}_{f_i}) \right| \qquad (6)$$

To achieve this effect for each onion skin figure, we calculate the time difference for the preceding n frames from the figure and apply the results to the composite producing n trails following the subject. The trails are drawn in a fixed colour (black in this case) whose opacity is proportional to the motion measure as calculated in equation 6. The opacity of a trail mark further varies as a function of the frame index difference between the mark and object so that the marks appear to fade away behind the object.

3.4 Camera Movement

We now have the information for three of our four storyboard components: background, foreground elements, and motion representation. The fourth component, an indication of camera motion, can be generated from the parameters of the perspective estimation algorithm used to generate the background mosaic. As described in section 1, techniques used for depicting camera movement are field cuts, mosaics and arrows.

In previous work, *SALSA* already incorporates two of these techniques — field cuts and mosaics. It generates a mosaic of the entire shot background and draws boxes showing the fields of view of the start and end frames, the start in green and the end in red, and line segments showing the motion of the centres of the view as the camera moves. The resulting representation has been shown to be very effective in conveying the camera movement so we make use of it in this reverse storyboarding system. As a further aid to visualisation, we also add the first and last frames as opaque overlays where the two frames do not overlap instead of presenting them separately as does *SALSA*.

Since the use of drawing the field cuts on a mosaic provides sufficient motion information, we did not pursue the generation and rendering of arrows as an additional technique. As for the object motion cues, the use of arrows can be redundant and their addition to the final storyboard can unduly clutter the result.

4 Results and Discussion

The final storyboard composite for the video sequence of figure 7 is shown in figure 13. The green box on the left outlines the opening frame of the shot while the red box shows the final frame. The middle blue line with cyan marks shows the path of the camera centre through the shot. As well, the start and end frames themselves are overlayed without any motion markings to anchor the shot. Between the start and the end, three versions of

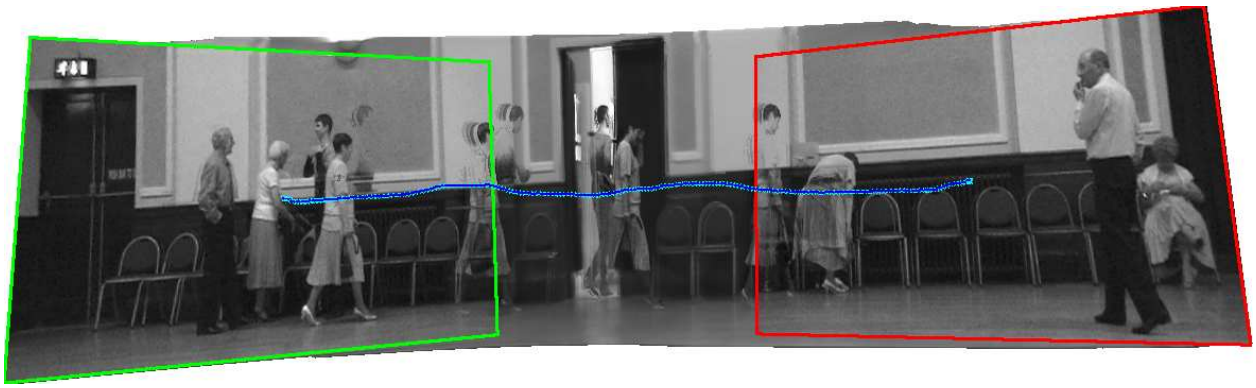

Figure 13: Storyboard automatically generated from video sequence in figure 7

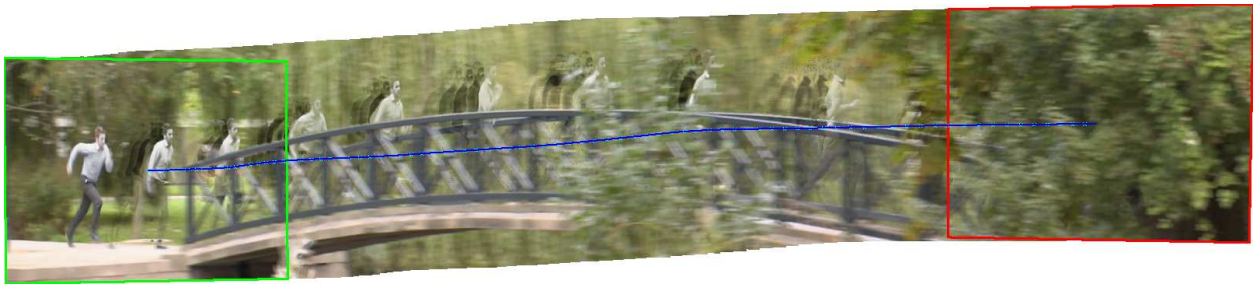

Figure 14: Storyboard automatically generated from *Tick Tock* tracking pan sequence (L-R)

the moving objects were added along with trailing black marks indicating motion.

The storyboard clearly conveys a number of key features of the shot. It is a pan right shot as indicated by the start and end frame boxes. The start frame overlay shows the initial four subjects. During the shot, the motion of the man walking to the right and exiting though the middle door is shown. As well, the motion of the woman walking to the right and eventually crouching down is conveyed. The final frame shows the crouched woman, a third woman sitting on a chair, and a third man walking to the left.

When examining the results of the individual components in sections 3.1-3.3 above, a number of artifacts were identified. However, when all the components are assembled in the storyboard these artifacts are not apparent. For example, the misclassification of the crouching woman figure as background or the figure's absence as a moving object are not visible. On the contrary, the semi-transparent rendering of her helps conveys the sequence of actions. As well, the action of the walking man entering the field of view on the right, while appearing distorted in both the background mosaic and the moving object identifier, is well conveyed by the final mosaic. Further, the discontinuities in the object identification and motion detectors due to low contrast between the object and background are not at all visible.

As another test, a tracking shot of a single person running from the movie short *Tick Tock* [16] was processed. Unlike the *Stargazer* sequence, this is in colour. The method was simply modified to use the colour median operator [21] and the L_1 distance in RGB space to measure differences for motion and object detection. The result is shown in figure 14. The result shows that the method generalises well to colour data. In this example, the elements of the shot are well represented. The onion skin of the versions of the figures and their motion tails convey their movement. Further, the blur due to the fast camera motion also conveys the sense of speed.

As a further test of the system, we processed a number of shots from the film *LeMans*. Figure 15 is from a pan shot of a crowd with a flag being waved. In the shot used for figure 16, the camera follows the man with glasses in the yellow shirt as he stands up. For figure 17, the camera does not move appreciably, but the man is motioning with his right hand. In the final shot, producing figure 18, again there is no camera motion, but subjects are moving.

Even for moderately complex shots, the system is relatively successful in conveying the composition of the shot. Figure 15 clearly shows the pan and the waving flag. However, the complex background in figure 16 interferes with the clear rendition of the moving figure producing a somewhat less than intelligible result. While our goal was initially to focus on tracking pan shots, we have included two holds with motion. The results in figures 17 and 18 show that the system can still convey the sense of motion and gives some indication of the nature of such motion.

5 Conclusions

Storyboarding is a production-industry standard visualisation tool for film and video. A storyboard effectively conveys a visual summary of shot elements such as back-

Figure 15: Storyboard generated from tracking pan (R-L-R)

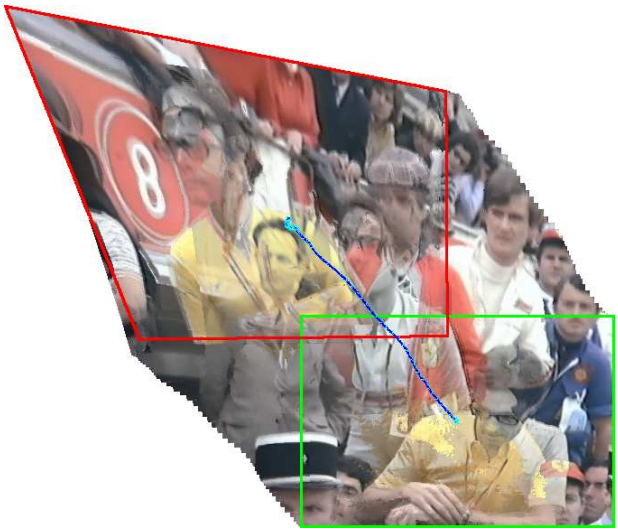

Figure 16: Storyboard generated from tracking tilt (L-Up)

Figure 17: Storyboard generated from handheld static shot (1)

ground, subject motion and camera framing and motion. Our goal has been to develop an automated system which takes raw video footage of a shot and, without the need for operator input, produces a visual representation of the shot in manner analogous to a hand-drawn story-

Figure 18: Storyboard generated from handheld static shot (2)

board frame. Such a system would be a valuable post-production tool. We build on a previously developed tool, *SALSA*, which provided some preliminary visualisation aids. Methods of image mosaicing based on sequential processing of frames do not adequately remove moving objects. We derive a statistical mosaicing method based on the median to produce the storyboard background. Foreground objects are then those that have intensity values which differ from the median. To convey a sense of motion, the difference between adjacent frames is used. The final image is a composite of the background, the moving subjects with motion cues in the manner of the onion skin method in storyboarding, boxes outlining the start and end frames with an overlay of the frames themselves, and a track of the intervening camera movement.

The system was run on a number of video shots. For tracking shots the system produced visual representations that succinctly conveyed the composition. The median-based mosaic generator successfully produced the storyboard background upon which the motion cues were applied. The motion cues produced are similar to the onion skin technique of standard storyboard art. These were generated using simple processing methods requiring no user input. Even for more complex shots, the output con-

veyed much of the composition. When the system processed shots with little camera movement, it was successful in capturing the essence of the motion. In some cases, however, with complex background and/or motion, the resulting composition is cluttered and can be difficult to interpret.

In all, the results demonstrate that the system fulfills our goal of producing a storyboard using the same devices used by industry storyboard artists in a completely automated manner. It can successfully be used on a number of important types of shots including holds with little foreground motion, shots with a camera movement and little other motion, and tracking shots. It therefore can be a valuable tool for the creation of visual summaries of existing film and video sequences for production, post-production, and archiving applications.

References

[1] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[2] D. Bullock and J. Zelek. Real-time tracking for visual interface applications in cluttered and occluding scenarios. In *Proceedings of the 15th IEEE International Conference on Vision Interface (ICVI) 2002*, page 751, Vancouver, BC, July 9–12 2002.

[3] J. C. Clarke and A. Zisserman. Detection and tracking of independent motion. In *Image Vision and Computing*, volume 14, pages 565–572, 1996.

[4] J. P. Collomosse, D. Rowntree, and P. M. Hall. Cartoon-style rendering of motion from video. In *Vision, Video and Graphics*, pages 117–124, July 2003.

[5] J. Davis. Mosaics of scenes with moving objects. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 354–360, 1998.

[6] A Girgensohn. A fast layout algorithm for visual video summaries. In *Proceedings of IEEE International Conference on Multimedia and Expo 2003*, pages 77–80, Baltimore MD, 2003.

[7] Richard D. Green and Ling Guan. Tracking human movement patterns using particle filtering. In *Proceedings 2003 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 25–28, Hong Kong, Apr 6–10 2003.

[8] Riad Hammoud and Roger Mohr. A probabilistic framework of selecting effective key frames for video browsing and indexing. In *International workshop on Real-Time Image Sequence Analysis*, Oulu, Finland, September 2000.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[10] Liwei He, Elizabeth Sanocki, Anoop Gupta, and Jonathan Grudin. Auto-summarization of audio-video presentations. In *ACM Multimedia (1)*, pages 489–498, 1999.

[11] Ryan C. Jones, Daniel DeMenthon, and David S. Doermann. Building mosaics from video using MPEG motion vectors. In *ACM Multimedia (2)*, pages 29–32, 1999.

[12] Steven D. Katz. *Shot By Shot: Visualizing From Concept to Screen*. Michael Wiese Productions, Studio City, 1991.

[13] Rainer Lienhart, Silvia Pfeiffer, and Wolfgang Effelsberg. Video abstracting. *Communications of the ACM*, 40(12):54–62, 1997.

[14] Wendy E. Mackay and Daniele Pagani. Video mosaic: Laying out time in a physical space. In *ACM Multimedia*, pages 165–172, 1994.

[15] S. Mann and R. W. Picard. Video orbits of the projective group: A simple approach to featureless estimation of parameters. *IEEE Transactions on Image Processing*, 6(9):1281–1295, Sept 1997.

[16] J. W. Mateer. *Tick Tock*. University of York, 2003.

[17] J. W. Mateer and J. A. Robinson. ASAP output log for *LeMans*. `http://www.amp.york.ac.uk/external/visual/asap/30fps6.8.html`.

[18] J. W. Mateer and J. A. Robinson. Semi-automated logging for professional media applications. In *Proceedings of Video, Vision and Graphics 2003*, pages 25–31, Bath UK, 2003.

[19] J. W. Mateer and The Zephyrs. *Stargazer*. Southpaw Records/Play It Again Sam Music, 2000.

[20] S. Pelg and J. Herman. Panoramic mosaics by manifold projection. In *IEEE CVPR Proceedings (1997).*, pages 338–343, Washington, DC, June 1997.

[21] K. N. Plataniotis and A. N. Venetsanopoulos. *Color image processing and applications*. Springer-Verlag, 2000.

[22] Solar Productions. *Le Mans*. Paramount Studios, 1971.

[23] J. A. Robinson. A simplex based projective transform estimator. In *Proceedings Visual Information Engineering (VIE) 2003*, Guildford, July 2003.

[24] M. A. Shamim and J. A. Robinson. Object-based video coding by global-to-local motion segmentation. *IEEE Trans Circuits and Systems for Video Technology*, 12(12):1106–1116, 2002.

[25] M. A. Shamim and J. A. Robinson. Asymmetric binary tree coding for contour images. *Image and Vision Computing*, 21(9):797–807, 2003.

[26] R. Szeliski. Image mosaicing for tele-reality applications. In *IEEE Workshop on Applications of Computer Vision (WACV'94)*, pages 44–53, 1994.

[27] B-L Yeung, M.M andYeo. Video visualization of compact presentation and fast browsing of pictorial content. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(5):771–785, 1997.