Schematic Storyboards from Video

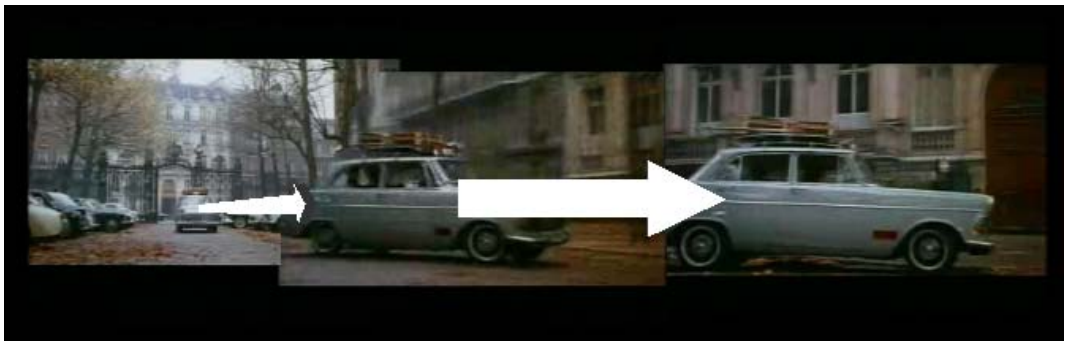

Figure 1 Schematic storyboard for the "car" sequence. The car appears in multiple locations, and the left arrow indicates a large motion toward the camera. The arrows were placed and rendered without recovering the 3D location of the car.

Abstract

We present a method for visualizing short video clips in a single static image, using the visual language of storyboards: We employ extended frame, 3D arrows, and zoom lines to convey the motion of the camera and objects within the scene. The principal advantage of this static representation is that it requires only a moment to observe and comprehend, but retains much of the detail of the source video. Potential applications include illustration of camera technique for film studies, assembly instructions, surveillance summarization, video editing, and composition of graphic novels.

1 Introduction

Film and video footage is typically divided into *shots*, where a shot is a length of frames captured in sequence at a single time. But once the footage is split into shots, how do you pick out one shot from a large collection? Video editing software usually presents each shot as a single static frame – usually the first frame. But what if many shots are similar except for the motion of the camera or the subject? How can you tell them apart? Or what if the first frame of a shot isn't representative of the rest? If you are a professional film or broadcast editor, you hire an intern to log your shots, associating text with each one; You carefully review each shot, taking notes on their contents, and you become skilled at memorizing them too.

Of course, the average home video editor doesn't have the luxury of hiring loggers, nor the time to carefully review and memorize the contents of each shot. So even though video editing software is just as readily available as photo editing software, many more people are willing to edit their digital pictures and put them online than would do the same with their home video.

But what if there were a way to summarize an entire shot – not just a single frame – in a still image? What if there were a visual representation that could communicate high-level aspects of motion of camera and subject, without requiring a moving display? A diagram that would help distinguish the different motion in two shots, even when individual frames from each shot look similar? A video schematic that you could print out on a piece of paper?

This problem has been around since the beginning of motion pictures, and filmmakers have invented a special type of diagram – the *storyboard* – to address it. Although the dictionary definition of a storyboard is just a sequence of still frames representing a moving sequence, storyboard artists have developed a distinct visual vocabulary to concisely summarize moving compositions. This iconography is not set in stone, but books on storyboarding contain some common idioms:

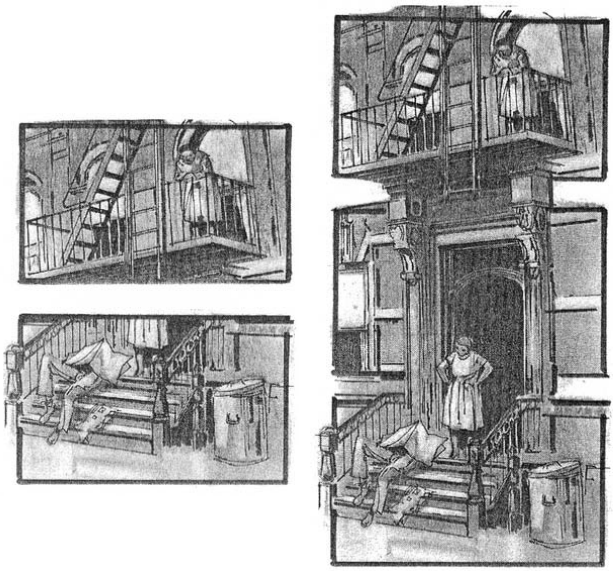

Figure 2 Extended Frame. A storyboard from *The Cotton Club* illustrating a tilt up from ground level to a second-story window. On the left, only the first and last frame of the shot are represented, omitting important visual information (the face of the woman standing on the stoop, the height of the window, etc.). Credits: © Harold Michelson, reprinted without permission.

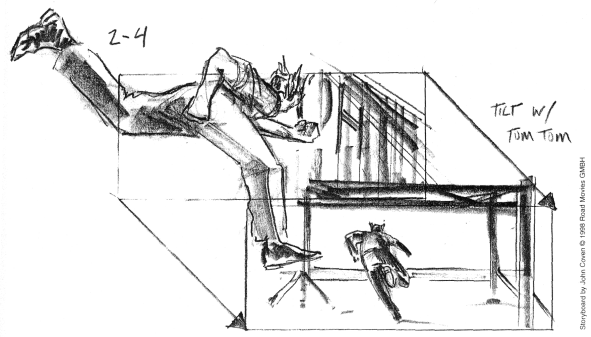

Figure 3 Another extended frame storyboard, showing the foreground character at two different moments in time. Credits: John Coven, ©Road Movies GMBH, reprinted without permission.

Figure 4 Zoom Lines. Storyboards illustrating a zoom in (left) and a zoom out (right). Note that the arrows in the left frame do not extend all the way to the corners of the frame, but are inset to increase visibility. Left credits: Mark Simon, ©EPL Productions, Inc., reprinted without permission. Right credits: © Marcie Begleiter, reprinted without permission.

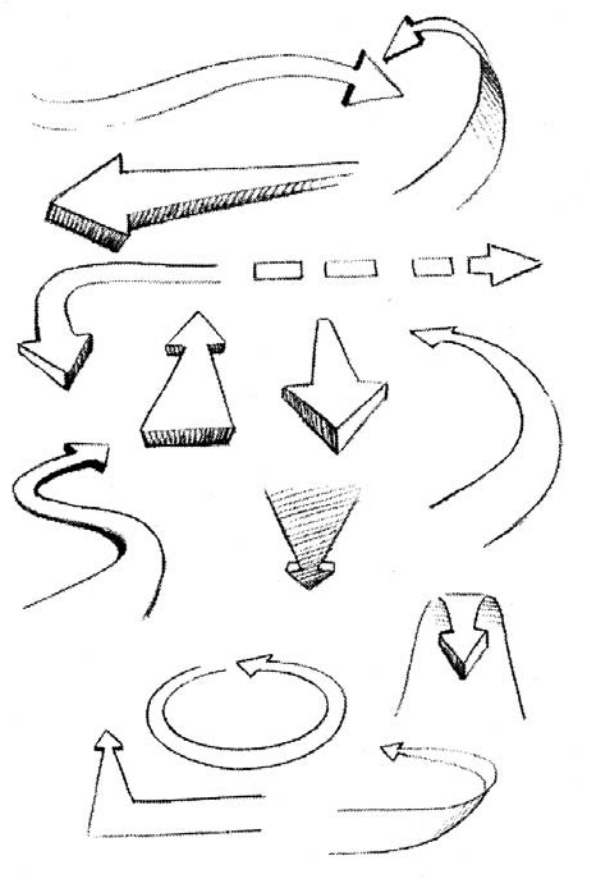

Figure 5 Arrows. A variety of arrow styles used by storyboard artists. Credits: ©Marcie Begleiter, reprinted without permission.

Extended Frame: When the camera is moving throughout a shot, multiple frames can be arranged in a single composition along the direction of motion. (See Figures 2 and 3.)

Zoom Lines: A change of focal length can be denoted by insetting an outline of the frame, with arrows between the corners of the frame indicating the direction of the zoom. (See Figure 4.)

3-D Arrows: Storyboard artists depict subject motion using many different arrow styles, taking advantage of thickness, curvature, twist, and perspective to describe elaborate motions in three-dimensional space. (See Figure 5.)

We use the term *schematic storyboard* to describe storyboards which combine both pictorial and diagrammatic elements such as these.

The rules of composition for these elements are complex and fluid, but we have identified some of the key stylistic conventions of traditional storyboards. Armed with these conventions, we have developed a computational framework and algorithms to generate storyboards from video with a small amount of user input.

2 Related Work

The Salient Stills work [Teodosio and Bender 1993; Teodosio and Bender 2005] represents one of the first works to attempt video summarization in a single image. In particular, Massey and Bender [1996] noted the applicability of Salient Stills for conveying camera and subject motion. More recently, Freeman and Zhang [2003] used range data to merge multiple frames of video into a single image as if they occupied the same space simultaneously. Agarwala *et al.* [2004] seamlessly merged multiple images using a variety of user-specified criteria, and demonstrated the application of their system to several types of time-varying scenes.

Artists and scientists employ a variety of methods to illustrate motion in static images. Cutting [2002] catalogued five distinct solutions historically employed by artists: Dynamic balance, multiple stroboscopic images, affine shear/forward lean, photographic blur, and image and action lines. Cartoonists employ these and other tools for depicting the temporal axis, including "speedlines", adjusting the shape and size of panels, and bleeding panels off the page [McCloud 1993]. Masuch *et al.* [1999] have applied speedlines to computer animation, and Kawagishi *et al.* [2003] incorporate additional techniques such as geometric deformation. Kim and Essa [2005] use these and other techniques to annotate video.

We have chosen the art and techniques of production storyboards as an ideal iconography for video visualization purposes. Storyboards have been used since the dawn of filmmaking [Hart 1999] to articulate and communicate concepts of image composition and scene blocking. Ours is not the first work to explicitly adopt this visual language: Nienhaus and Döllner [2003] previously adopted storyboard-style 3D arrows to depict dynamics in 3D scenes.

Our schematic storyboards merge multiple frames taken from different points of view into a single composite image. This is related to the ideas of multiperspective panoramas [Wood et al. 1997], multiple-center-of-projection images [Rademacher and Bishop 1998], and manifold projection [Peleg and Herman 1997], all of which create composite images with multiple viewpoints. However, these works use large numbers of viewpoints to create a near-continuous change of perspective, whereas we are using a small number of "key" frames.

Although it is not the focus of our work, our problem is related to that of video abstraction or summarization, which attempts to create a compact abstract (either still or animated) of a large collection of video. The literature in this topic is large, but Li *et al.* [2001] recently overviewed the field in brief. Irani and Anandan [1998] created a system for summarizing surveillance video which shares some common goals with our work. The PanoramaExcerpts system [Taniguchi et al. 1997] summarizes large collections of video using both single frames and panoramic mosaics. Our work attempts to extend the expressiveness of these static summaries using storyboard annotations.

We assume that our video material has already been segmented into individual shots. This can be done manually, but dozens of automatic methods have also been developed; we cite only a few recent works [Adjeroh et al. 1997; Nicolas et al. 2004; Heng and Ngan 2001; Cheong and Huo 2001; Vlachos 2000; Lee et al. 2001].

3 The Visual Language of Storyboards

We propose visualization of video in a single static storyboard diagram. This schematic storyboard will be designed to communicate

high-level motion of the observer and observed objects, abstracting away details which may be less important for understanding motion. At the same time, the storyboard should relate in an intuitive way to the original data, so that it can be used in conjunction with traditional animated or filmstrip display.

We began our investigation by undertaking an informal survey of storyboarding and film studies references [Simon 2000; Begleiter 2001; Hart 1999; Block 2001; Katz 1991]. This survey revealed some of the specific techniques used by storyboard artists, which we attempt to formalize in the remainder of this section.

Although our system does not – and probably never will – produce storyboards of the same quality as a human storyboard artists, we find it useful and instructive to enumerate some of the considerations that these artists must take into account. However, note that only a small number of these are used as criteria in the automated system described in Section 4.

Key Frames. Storyboards typically depict several "key" moments in the time span of a shot. The depicted moments in time represent some or all of the following qualities;

- extrema of motion,
- "representative" poses,
- clarity of expression or pose, and
- dynamic balance, suggesting the motion in progress.

Also, different objects or individuals in the scene may be depicted at different moments in time in order to more fully optimize these criteria.

Extended Frame. An extended frame is an arrangement of multiple frames on the two spatial axes of a screen or page. The frames are arranged so that the background appears continuous. Typically standard planar projections are used, but different regions of the extended frame may have different perspective projection. Changes in perspective are typically hidden in featureless regions or at architectural boundaries.

In contrast to multiperspective panoramas [Wood et al. 1997], a storyboard is intended to be viewed in its entirety at a single orientation. Therefore, even if the best alignment between multiple frames includes a rotation, all frames in an extended frame are placed on the page or screen without rotation. (One rare exception is when the camera undergoes a large and rapid change of roll angle over the duration of a shot.)

Note that arranging many frames in a single extended frame may sacrifice clarity and legibility. Therefore, storyboard artists use extended frame sparingly, typically only when the camera and/or subject are moving smoothly and the image composition changes from the beginning to the end of the shot. This can be the case even if only a portion of an object is in motion, as in Figure 6. However, extended frame compositions are split into smaller segments or even individual frames if the resulting composition would sacrifice clarity. Such a confusing composition may result from:

- poor alignment between frames,
- large scale changes between frames due to changes in focal length or motion of the camera, or
- motion of the camera or subject that "backtracks," so that distant moments in time would obscure each other.

In addition, storyboard artists may break an extended frame into several segments in order to avoid wasted space on the page.

Motion Arrows. Storyboard artists often augment the subjects of the frames with 3D arrows that roughly follow the path of motion of the camera or subjects. These are usually rendered as if they were solid or semi-transparent objects in the scene itself, using different line styles or shading to distinguish the motion paths of different objects. Motion arrows provide a more definitive sense of direction of motion than speedlines, motion blur, or some of the other mechanisms previously discussed. Furthermore, they can describe additional degrees of freedom, having both thickness and "twist" which may vary over the length of the arrow. Many storyboards observe the following conventions for motion arrow depiction:

Figure 6 A storyboard in which two moments in time are represented in a single composition, even though only a small portion of the scene is in motion. Credits: © Mark Simon, reprinted without permission.

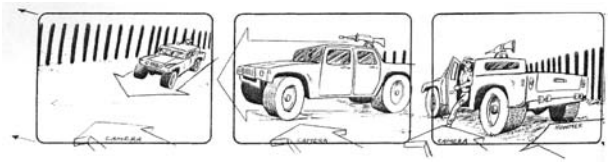

Figure 7 Labelled arrows, with different twist angles for the car vs. camera. Credits: Mark Simon, ©Universal City Studios, reprinted without permission.

- Arrows are piecewise smooth, emphasizing broad motion rather than small details.
- Arrows never obscure important objects in the scene.
- For objects rotating about their direction of translation ("rolling") – eg. a banking aircraft – the arrow twist varies over its length, and maintains alignment with the horizontal or vertical plane of the object.
- For objects that do not roll – eg. a person or animal – the arrow twist may either be aligned to the object's horizontal or vertical plane, or aligned so as to maximize the thickness of the arrow as seen from the camera.
- Arrow size is proportional to the object size, so that a change in size from front to back indicates motion along the camera axis, though the change in perspective may be exaggerated to emphasize the motion.
- Arrows are omitted if the motion is short or self-evident.
- When multiple objects move in the same general path, a single arrow may be used to represent their aggregate motion.
- If the object referred to by an arrow is ambiguous, the arrow may include a textual label. (This is often the case for arrows indicating camera motion, since the camera itself is not visible.)

Zoom Lines. Changes in focal length (zooms) are denoted by concentric sets of frame lines, using 2D arrows to indicate the direction of zoom. The frame lines are typically unbroken (even if they intersect important details of the image), but the arrow heads and tails may be offset from the corners of the frames in order to improve legibility.

Depth Ordering. The foregrouned objects depicted in extended frame are composed in depth order, with closer objects appearing in front of more distant objects, regardless of temporal ordering. Motion arrows are also rendered in depth order, unless they would be too heavily obscured by closer objects. The background of the storyboard is depicted as a continuous scene, hiding changes in perspective in featureless areas or along architectural boundaries.

Sketch Style. Storyboards are typically rendered by hand using pencil or charcoal. These are often rendered quickly without significant detail, texture, or shading. Often the dominant moving object is rendered in the most detail, with static background objects rendered more loosely.

4 System Overview

We have developed a system that partially automates the creation of a storyboard from a single shot using some of the techniques observed in Section 3. Our system is broken into the following five stages, which are performed in sequence: keyframe selection, feature tracking and labelling, extended frame layout, compositing, and annotation rendering. The algorithms for each stage are described in detail in subsections which follow.

At present, the first two stages – keyframe selection and feature tracking and labelling – are performed by hand, but we hope to further automate the process by the time of our final Siggraph submission. Also, our system does not presently render the image content in a sketch style, but we also hope to explore stylization in future work. Finally, the text which follows assumes that only two labels are used (foreground and background) for objects in the scene, but we hope to extend the work to multiple objects soon.

4.1 Extended Frame Layout

Although our schematic storyboard may contain multiple separate *segments* with independent coordinate systems, we first find an alignment of all frames in a single extended frame. This single extended frame will later be segmented if necessary.

Consider a single pair of frames i and j, with N background features in common, denoted $\mathbf{f}_i(x)$ and $\mathbf{f}_j(x)$. The obvious approach is to solve for the least-squares transformation between $\mathbf{f}_i(x)$ and $\mathbf{f}_j(x)$ using only uniform scale and translation, but this can produce degenerate solutions with zero or negative scales when large rotations are present.

An alternate approach is to find correspondences with rotations, then "undo" the rotations. Our approach is simply a modification of a method due to Horn [1988] and refined by Umeyama [1991], in which we have substituted the optimal rotation R with the identity matrix. Indeed, when there is no rotation between the two sets of feature points, this transformation is the optimal least-squares uniform scale and translation between the points.

First, we compute the centroids and standard deviations of the features in each frame in the standard way:

$$\bar{\mathbf{f}}_i = \sum_x \mathbf{f}_i(x)/N \tag{1}$$

$$\sigma_i = \sum_x \|\mathbf{f}_i(x) - \bar{\mathbf{f}}_i\|^2/N \tag{2}$$

Then the relative scale between frames i and j is computed as the ratio of the standard deviations of the feature positions, and the relative translation is given by the difference of scaled centroids:

$$s = \sigma_j/\sigma_i \tag{3}$$

$$\mathbf{t} = \bar{\mathbf{f}}_j - s\bar{\mathbf{f}}_i \tag{4}$$

We denote this transformation $\mathbf{M}_{i\to j}$. (Note that all features are not visible in all frames, so for each pair of frames we recompute $\bar{\mathbf{f}}$ and σ using the subset of feature points visible in both frames.)

After computing the transformations between temporally adjacent pairs of frames, we assign each frame a transformation $\mathbf{M}_i$ in a global extended frame coordinate system. The first frame is arbitrarily assigned to lie at the origin with a scale of 1, and each successive frame is transformed by appending the transformation from the previous frame:

$$\mathbf{M}_0 = \mathbf{I} \tag{5}$$

$$\mathbf{M}_i = \mathbf{M}_{(i-1)\to i} \circ \mathbf{M}_{i-1} \tag{6}$$

This placement is not globally optimal, since small errors between pairs of frames may accumulate over the length of a sequence. But we have found this acceptable as long as temporally distant frames do not overlap – that is, as long as the camera does not pass over the same background multiple times.

Note that using this layout, the spatial direction of the temporal axis aligns with the direction of dominant motion, so that a continuous motion is perceived spatially along a continuous curve on the display.

Extended Frame Segments We split the sequence of frames into multiple extended frame segments whenever the relative scale between frames in a segment becomes too large. This is done using a greedy algorithm: Frames are accumulated into continuous segments until the scale ratio between smallest and largest frame drops below a threshhold T_s, at which point the current frame becomes the first frame of a new segment. When all the segments have been computed, the segments are rescaled so that the first frame of each segment has scale $s = 1$.

In a typical storyboard, separate extended frame segments are laid out so as to minimize wasted space on a page. However, we have developed an alternate method for segment layout that adopts the spirit of extended frame layout: We place temporally adjacent segments as close as possible without overlap (and with a small amount of pad space) so that the offset vector between the last frame of the first segment to the first frame of the next segment is parallel to the direction of camera movement between the same frames. Again, the spatial layout of the temporal axis aligns with the direction of background motion, allowing the eye to travel across a layout in the same direction as the camera motion. (See Figure 9 for example.)

4.2 Compositing

At present, we composite the extended layout before adding other 3D annotations such as arrows. At some point we hope to include arrows in the compositing proces.

When assembling our storyboard composite, we must balance competing goals. Where regions of background overlap, we would like to create a seamless composite, as in a panoramic mosaic. Where regions of foreground appear, we would like to ensure that the foreground object appears in the final composite. Finally, where multiple foreground objects overlap, we would like them to appear in the proper depth priority, with closer objects occluding more distant ones.

If we knew with certainty which pixels belonged to foreground and background, we could treat these problems independently: First create a seamless background composite using all the background pixels B, then composite the foreground pixels F from back to front.

But we have found that even the best natural image matting algorithms cannot successfully matte noisy, heavily textured input images such as those seen in Figure 9a.

Instead, we will treat compositing as a probabilistic labelling problem, in which the label assigned to a pixel determines the frame from which we draw the color of that pixel. Let p_i be a pixel in frame i, and assume we have some probabilities $P(p_i \in F)$ and $P(p_i \in B)$ that p_i is in the foreground and background, respectively. Let $L^*(p)$ denote the "ideal" labeling for pixel p. Then we we write the probability that p_i is the topmost pixel in the composite as

$$P(L^*(p) = i) = P(p_i \in F) \prod_{j \in K(i)} P(p_j \in B) \tag{7}$$

where $K(i)$ is the set of frames with foreground objects closer than in frame i. That is, a pixel at layer i is likely to be topmost if it is likely to be the foreground **and** if the corresponding pixels at all closer layers j are likely to be in the background.

We presently compute the probabilities $P(p_i \in F)$ and $P(p_i \in B)$ using the GrabCut matting algorithm [Rother et al. 2004], which computes a hard matte while also estimating Gaussian mixture models (GMM) for foreground and background pixel colors.

Borrowing the terminology and notation of Agarwala *et al.* [2004], we define the cost function C of a pixel labeling L as the sum of two terms: a data penalty C_d over all pixels p and an interaction penalty C_i over all pairs of neighboring pixels p, q:

$$C(L) = \sum_p C_d(p, L(p)) + \sum_{p,q} C_i(p, q, L(p), L(q)) \tag{8}$$

In our application, $C_d(p, L(p)) = -\log P(L^*(p) = L(p))$, so that the data penalty is the negative log likelihood of the current labelling over all pixels. We adopt the interaction penalty $X + Y$ from Agarwala *et al.* [2004]:

$$X = \|S_{L(p)}(p) - S_{L(q)}(p)\| + \|S_{L(p)}(q) - S_{L(q)}(q)\| \tag{9}$$

$$Y = \|\nabla S_{L(p)}(p) - \nabla S_{L(q)}(p)\| + \|\nabla S_{L(p)}(q) - \nabla S_{L(q)}(q)\| \tag{10}$$

The resulting cost function is approximately minimized using *alpha expansion* [Boykov et al. 2001].

Note that in regions where only likely background pixels $p_i \in B$ overlap, all of the probabilities $P(L^*(p) = i)$ are low. This is in effect a "don't care" situation, where the optimal cut will be principally determined the interaction penalty. However, in areas where likely foreground pixels appear, the proper depth ordering will be maintained.

4.3 Motion Arrows

In order to quickly explore a variety of placement algorithms, we currently render only straight arrows with no curvature or varying twist. Even in this restricted setting, each arrow has ten geometric degrees of freedom: three each for starting and ending position, two for breadth and thickness, and one for twist angle. In addition, each arrow could be rendered using a different amount of foreshortening, determined by the camera's focal length. (See Figure 8.)

To simplify the situation, in our current system we choose a single coordinate system for all motion arrows, with an arbitrary user-selected focal length.

An arrow is placed between each pair of successive frames in an extended frame in which the foreground object is visible. The 3D locations of the arrow endpoints are determined using the locations of the features labelled as foreground. However, we may not have enough information about the scene to compute true three-dimensional locations of the features. Furthermore, our key frames

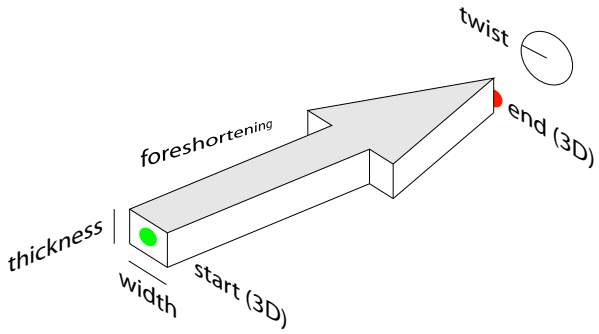

Figure 8 The ten degrees of freedom of a simple 3D arrow.

do not share a common 3D coordinate system. Therefore we employ a pseudo-3D estimation using the 2D distributions of the foreground features in the extended frame coordinate system.

First, the 2D centroids $\bar{\mathbf{f}}$ and standard deviations σ of the foreground features in each frame are computed in the global extended frame coordinate system. We assume that the object does not change size over time, so the standard deviation of the foreground feature points is roughly proportional to the size of the foreground object in each frame. If we know the distance to the object in one frame, we can estimate its distance in any other frame using similar triangles: $d_j/d_i = \sigma_i/\sigma_j$. Therefore, we only need to provide the distance to the foreground object in a single frame. We assume the object is at its closest in the frame with the largest standard deviation, $i_{min} = \text{argmax}_i \sigma_i$. The distance $d_{i_{min}}$ of the foreground object at this closest frame is specified as a constant. The foreground object centers are approximated as the point along the ray from the camera through the feature centroid $\bar{\mathbf{f}}_i$ at distance d_i. Finally the arrow endpoints are offset from the foreground positions along the arrow axis by a small amount to improve visibility of the foreground objects.

The width and thickness of the arrows (i.e. the dimension along which the arrowhead flares, and the perpendicular dimension) are set to be linearly proportional to the standard deviation of the features at the closest frame:

$$w = \alpha \sigma_{i_{min}} \tag{11}$$

$$t = \beta \sigma_{i_{min}} \tag{12}$$

4.4 Intersegment Motion Annotations

As described above, a shot may be split into multiple extended frame segments because of widely varying scales across the shot. This scale change can occur either because of motion into or out of the image plane (known to cinematographers and camera operators as *dollying*), or because of changes of focal length (known as a *zoom*).

We annotate scale changes due to dollying using 3-D arrows from the end of one segment to the beginning of the next, but for these intersegment arrows we choose nearby points on the perimeter of the segment endframes as the 2-D endpoints of the arrows. At present we do not detect the difference between these camera operations, but allow the user to choose the appropriate annotations for each scale change.

Scale changes due to zoom are annoted using zoom lines. Consider adjacent frames i and $i+1$ which have been placed in successive segments A and B. To represent a zoom-in between A and B, we draw an outline around frame i, as well as the outline of frame

Figure 10 Schematic storyboard for the telephone sequence.

$i+1$ in the coordinate frame of i. The transformation between the two outlines is simply $\mathbf{M}_{i\to(i+1)}$. For a zoom-out, the procedure is similar, but this time we draw the outlines atop frame $i+1$, using the transformation $\mathbf{M}_{(i+1)\to i} = \mathbf{M}^{-1}_{i\to(i+1)}$. Finally, the corresponding corners of the frame outlines are connected using 2D arrows.

Figure 9 contains only dollying (no zooming), but we apply both mechanisms for annotating scale changes for illustration purposes.

5 Results

We have evaluated our system on three example videos. For all three results we used $T_s = .7$ as the segment-splitting threshhold, and $d_{i_{min}} = 500$ pixels as the minimum distance to the foreground object. The width and thickness scale factors were set to $\alpha = \frac{1}{2}, \beta = \frac{1}{8}$.

The "naterun" sequence is a 20 second home video taken using a handheld digital camera at 320×240, 15 frames per second, and compressed with MPEG-4. Figure 9a shows the 8 user-selected key frames, and Figure 9b shows the mattes produced automatically using GrabCut. About a dozen feature points were manually selected in each of the key frames and labelled as foreground or background. We initialized GrabCut using a padded bounding box surrounding the feature points labelled as foreground in each frame, so no additional user input was required to produce the mattes. The mattes themselves are low-quality, often cutting out large sections of the subject's body, but the Gaussian mixture models they produce can still be used to guide the graph-cut composite shown in Figure 9c. The final schematic storyboard, with 3D arrows and zoom lines overlaid on the composite, is shown in Figure 9d.

Two additional examples are shown in Figure 1 and Figure 10. The "car" and "telephone" shots, about 5 seconds each, were extracted from the film *Charade*, digitized at 320×240 and 30 frames per second, and compressed with MPEG-4. They exhibit some undesirable artifacts, such as misalignments (due to poor manual feature correspondence) and odd cuts (due to a weak foreground/background model). Nonetheless, we believe they show significant promise for automatic generation of schematic storyboards.

6 Discussion and Applications

Typical visualizations of video usually take one of two forms: Either the data is presented in the obvious way, as an animation, with the temporal axis mapped linearly to the display time, or it is displayed as a sequence of still images arranged in a line or grid, much like a filmstrip or proofsheet. However, both the animated and filmstrip representations have serious drawbacks for user understanding.

Schematic storyboards, on the other hand, are alternate visualizations of video that are static – like a filmstrip – but organized and annotated to convey continuity and directionality – like an animation.

A key advantage of schematic storyboards is that a significant time interval of the video can be observed instantaneously. In contrast, the simple act of observing an animated display takes a certain length of time: A ten-minute shot generally takes ten minutes of a user's time to observe in its entirety. Of course, one can always fast-forward through a video (in effect scaling the data along the temporal axis) but as playback speed increases, it becomes more difficult to observe details of the motion.

In addition, schematic storyboards are well-suited for applications in which several datasets must be observed and mentally processed in parallel, such as video editing. An animated display is awkward to use in such applications, since the human visual system can be quickly overwhelmed by even small numbers of video streams playing simultaneously: A rapid motion in one video may distract the observer's attention from small but important motions in another video playing simultaneously.

Although a static filmstrip representation does not share the same drawbacks as animated displays, it imposes a geometric pattern on the output (the layout of frames) that is spatially unrelated to the input data and therefore visually distracts from it. In a schematic storyboard, on the other hand, the data is placed in a pattern which emphasizes the spatial relations of multiple frames, and graphical annotations are used to suggest the data which has been omitted.

A wide range of tasks involving video will benefit from the concise summary of motion afforded by schematic storyboards. In a few instances, automated storyboard generation could replace illustrations presently drawn by hand. For instance, textbooks on film studies already use production storyboards to illustrate film techniques [Katz 1991], and where the original storyboards are unavailable, they may be reproduced by hand [Hart 1999]. Similarly, instruction manuals typically contain illustrations of a physical process that must clearly represent the assembly of an object or the operation of a device. In other applications, such as surveillance, an automated storyboard would act as a high-level preview of video material for pre-screening of unusual activity [Irani and Anandan 1998].

Furthermore, because storyboards are more abstract than filmstrips or animations, they can be especially useful for tasks in which multiple videos must be viewed, compared, and selected. For example, stock footage galleries can be logged and indexed using storyboards for easy reference and access. Storyboards may also be useful in the initial "assembly" stage of editing documentary footage, in which large volumes of data must be screened and culled to select shots that will be used in the first cut [Sangster 2005].

7 Conclusion

We have presented a system for transforming video clips into static visualizations using the visual language of storyboards. Our contributions include: Introducing the iconography of storyboards as a visualization tool for video, a novel framework for multiperspective layout, and representation of visual motion using 3D arrows without requiring true 3D reconstruction.

We believe that schematic storyboards can provide effective visualizations for presentation in both inherently static print media and dynamic digital media. In contrast with animated or filmstrip displays, a storyboard depiction allows rapid visual comprehension for

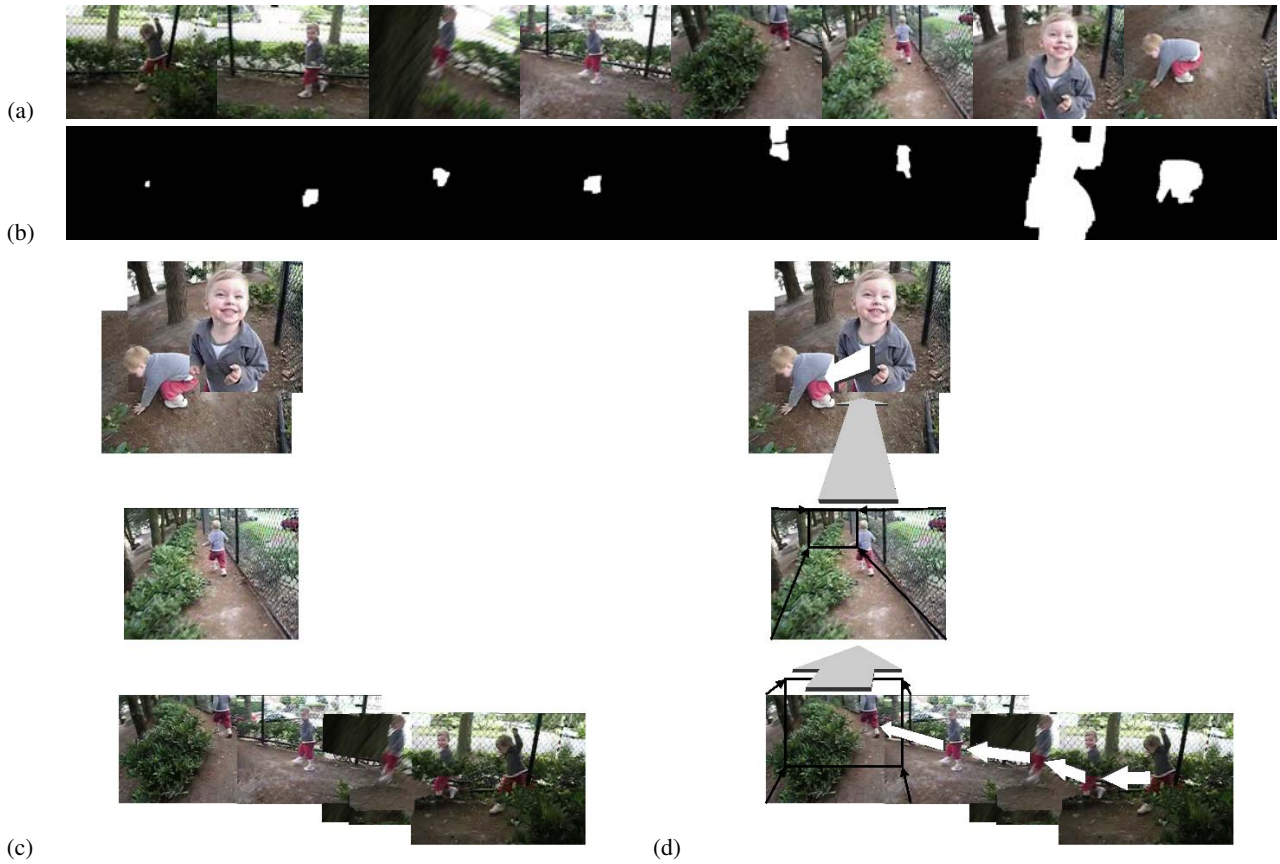

Figure 9 Source frames (a) and GrabCut mattes (b) for the naterun sequence. The mattes are not very accurate, but we can still use the Gaussian mixture models from GrabCut to produce a composite (c). The complete schematic storyboard (d) includes zoom lines and 3D arrows.

a variety of tasks involving selection, comparison, and manipulation of motion data. We look forward to exploring the use of storyboards for motion visualization.

8 Future Work

8.1 Short Term (possibly for Siggraph)

We hope to improve the quality of our results and bring them closer to the appearance of hand-drawn storyboards, for example indicating the motion of multiple foreground objects using curved arrows.

We also intend to refine the compositing results. Both the foreground/background probability model and the interaction penalty could be improved. The GMM color models provided by GrabCut without user interaction are unsatisfactory, and in some cases degenerate. However, we believe that the labelled features could be used to improve the probability model, for example by incorporating a geometric term (i.e. distance to closest feature). We would also like to encourage seams along likely foreground/background boundaries.

The amount of user interaction in our system is already fairly low, but the feature correspondence and labelling problem is likely to be partially automatable. We expect that SIFT features can be used for feature tracking, and we hope that the feature labelling task can be partially automated. For example, if the labelling is performed manually on a single frame, it may be possible to automatically propagate labels to other frames.

8.2 Long Term (probably after Siggraph)

One problem inherent in multi-perspective mosaics such as those shown here is that graph cuts often fail to hide changes of perspective or parallax. One possible solution is to encourage seams along architectural boundaries [Zelnik-Manor et al. 2005]. Another is to apply small amounts of distortion to the images in order to improve their alignment [Jia and Tang 2005].

Our schematic storyboards are presently rendered in a literal photographic style, unlike the stylized sketches of production storyboards. Although this allows us to present visual details not typically seen in actual storyboards, it also contributes to visual clutter for applications in which the motion is more important than the details. Stylization of schematic storyboards is an interesting challenge, as we can take advantage of motion cues to render differently moving regions in different styles.

Schematic storyboards can be augmented as adaptive entities that change their appearance depending on context or size. For example, when surveying a large number of storyboards for assembly or editing purposes, the storyboards may take on an abstract appearance emphasizing one or two frames with simplified color schemes and 2D motion arrows. Storyboards may also be useful as an interface into spatiotemporal data: The user might click and drag along lines of motion to index into different frames of the source video. Both of these extensions may prove especially useful for depicting motion data without a unique viewpoint, such as motion capture data. In this setting, the choice of viewpoint is an important free variable.

In this work, we have emphasized some of the principal techniques employed by storyboard artists to depict motion, but many other mechanisms remain open to exploration. We close with an unusual example in which the camera motion is represented by the solid volume swept out by the camera plane (see Figure 11).

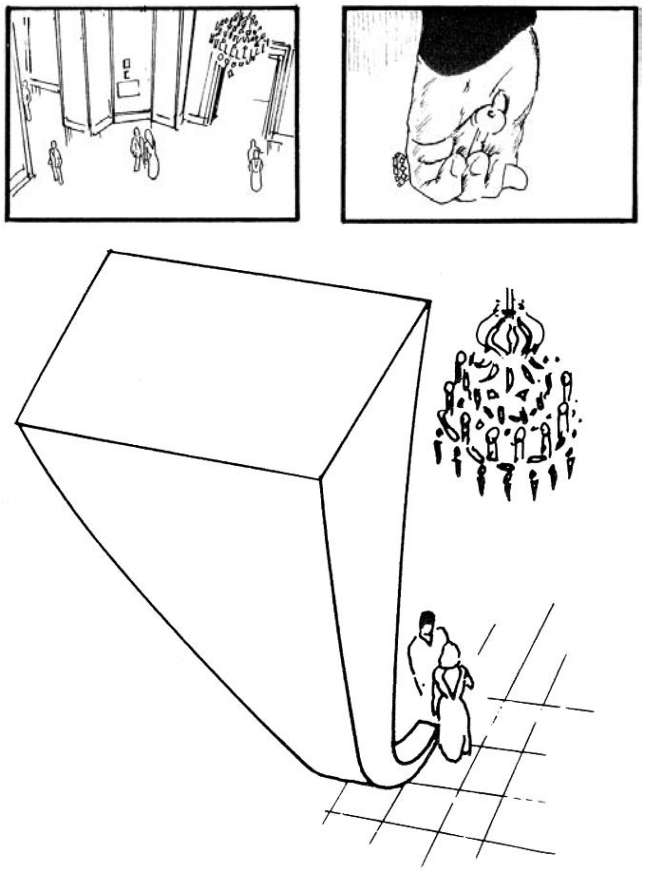

Figure 11 Top: The beginning and ending of a long crane shot from Hitchcock's *Notorious*. Bottom: An unusual depiction of the associated camera move as the solid volume swept out by the camera plane. ©Steven D. Katz and Frank Bolle, reprinted without permission.

References

ADJEROH, D., LEE, M., AND ORJI, C. 1997. Techniques for fast partitioning of compressed and uncompressed video. *Multimedia Tools and Applications 4*, 2 (March), 225–243.

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. *ACM Transactions on Graphics (Proc. SIGGRAPH) 23*, 4, 294–301.

BEGLEITER, M. 2001. *From word to image: storyboarding and the filmmaking process*. Michael Wiese Productions.

BLOCK, B. A. 2001. *The Visual Story: Seeing the Structure of Film, TV, and New Media*. Focal Press.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11, 1222–1239.

CHEONG, L., AND HUO, H. 2001. Shot change detection using scene-based constraint. *Multimedia Tools and Applications 14*, 2 (June), 175–186.

CUTTING, J. E. 2002. Representing motion in a static image: constraints and parallels in art, science, and popular culture. *Perception 31*, 1165–1193.

FREEMAN, W. T., AND ZHANG, H. 2003. Shape-time photography. In *Proc. Computer Vision and Pattern Recognition*, 151–157.

HART, J. 1999. *The art of the storyboard: storyboarding for film, TV and animation*. Focal Press.

HENG, W., AND NGAN, K. 2001. An object-based shot boundary detection using edge tracing and tracking. *Journal of Visual Communication and Image Representation 12*, 3 (September), 217–239.

HORN, B., HILDEN, H., AND NEGAHDARIPOUR, S. 1988. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A 5*, 7, 1127–1135.

IRANI, M., AND ANANDAN, P. 1998. Video indexing based on mosaic representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 86*, 5 (May), 905–921.

JIA, J., AND TANG, C.-K. 2005. Eliminating structure and intensity misalignment in image stitching. In *Proc. International Conference on Computer Vision*.

KATZ, S. D. 1991. *Film directing shot by shot: visualizing from concept to screen*. Michael Wiese Productions.

KAWAGISHI, Y., HATSUYAMA, K., AND KONDO, K. 2003. Cartoon blur: nonphotorealistic motion blur. In *Proc. Computer Graphics International*, 276–281.

KIM, B., AND ESSA, I. 2005. Video-based nonphotorealistic and expressive illustration of motion. In *Proc. Computer Graphics International*, 32–35.

LEE, M., YANG, Y., AND LEE, S. 2001. Automatic video parsing using shot boundary detection and camera operation analysis. *Pattern Recognition 34*, 3 (March), 711–719.

LI, Y., ZHANG, T., AND TRETTER, D. 2001. An overview of video abstraction techniques. Tech. Rep. HPL-2001-191, HP Laboratories.

MASSEY, M., AND BENDER, W. 1996. Salient stills: process and practice. *IBM Systems Journal 35*, 3,4, 557–574.

MASUCH, M., SCHLECHTWEG, S., AND SCHULTZ, R. 1999. Speedlines: depicting motion in motionless pictures. In *ACM SIGGRAPH 99 Conference abstracts and applications*, 277.

MCCLOUD, S. 1993. *Understanding Comics: The Invisible Art*. HarperCollins.

NICOLAS, H., MANAURY, A., BENOIS-PINEAU, J., DUPUY, W., AND BARBA, D. 2004. Grouping video shots into scenes based on 1d mosaic descriptors. In *Proc. International Conference on Image Processing*, I: 637–640.

NIENHAUS, M., AND DÖLLNER, J. 2003. Dynamic glyphs – depicting dynamics in images of 3D scenes. In *Third International Symposium on Smart Graphics*, 102–111.

PELEG, S., AND HERMAN, J. 1997. Panoramic mosaics by manifold projection. In *Proc. Computer Vision and Pattern Recognition*, 338.

RADEMACHER, P., AND BISHOP, G. 1998. Multiple-center-of-projection images. In *Proc. SIGGRAPH*, 199–206.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. "GrabCut" – interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (Proc. SIGGRAPH) 23*, 3, 309–314.

SANGSTER, C., 2005. Personal Communication.

SIMON, M. 2000. *Storyboards: Motion in Art*. Focal Press.

TANIGUCHI, Y., AKUTSU, A., AND TONOMURA, Y. 1997. PanoramaExcerpts: extracting and packing panoramas for video browsing. In *Proc. ACM International Conference on Multimedia*, 427–436.

TEODOSIO, L., AND BENDER, W. 1993. Salient video stills: Content and context preserved. In *Proc. ACM International Conference on Multimedia*, 39–46.

TEODOSIO, L., AND BENDER, W. 2005. Salient stills. *ACM Transactions on Multimedia Computing, Communications, and Applications 1*, 1 (February), 16–36.

UMEYAMA, S. 1991. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 4, 376–380.

VLACHOS, T. 2000. Cut detection in video sequences using phase correlation. *IEEE Signal Processing Letters 7*, 7 (July), 173–175.

WOOD, D. N., FINKELSTEIN, A., HUGHES, J. F., THAYER, C. E., AND SALESIN, D. H. 1997. Multiperspective panoramas for cel animation. In *Proc. SIGGRAPH*, 243–250.

ZELNIK-MANOR, L., PETERS, G., AND PERONA, P. 2005. Squaring the circle in panoramas. In *Proc. International Conference on Computer Vision*, to appear.