COMPAQ

Video Editing Using Figure Tracking and Image-Based Rendering

James M. Rehg *Sing Bing Kang* *Tat-Jen Cham*

Cambridge Research Laboratory

Cambridge Research Laboratory

Technical Report Series

CRL 99/8

December 1999

Cambridge Research Laboratory

The Cambridge Research Laboratory was founded in 1987 to advance the state of the art in both core computing and human-computer interaction, and to use the knowledge so gained to support the Company's corporate objectives. We believe this is best accomplished through interconnected pursuits in technology creation, advanced systems engineering, and business development. We are actively investigating scalable computing; mobile computing; vision-based human and scene sensing; speech interaction; computer-animated synthetic persona; intelligent information appliances; and the capture, coding, storage, indexing, retrieval, decoding, and rendering of multimedia data.

Robert A. Iannucci, Ph.D.
Vice-President of Research

Video Editing Using Figure Tracking and Image-Based Rendering

James M. Rehg Sing Bing Kang Tat-Jen Cham

December 1999

Abstract

We describe a new approach to video editing based on the semi-automatic segmentation of video into multiple layers and the compositing of layers using image-based rendering. Using figure tracking and background motion estimation, we can segment a moving figure and reconstruct the background. Using geometrically-correct pixel reprojection, layers can be composited on the basis of the geometry of the underlying scene and the position of the virtual camera. We have implemented out approach in an editing system called *SpliceWorld*. We show results from editing a Fred Astaire dance sequence.

The work described in this report was conducted at the Cambridge Research Lab.
Authors email: rehg@crl.dec.com, sbk@microsoft.com, tjc@crl.dec.com

©Compaq Computer Corporation, 1999

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of the Cambridge Research Laboratory of Compaq Computer Corporation in Cambridge, Massachusetts; an acknowledgment of the authors and individual contributors to the work; and all applicable portions of the copyright notice. Copying, reproducing, or republishing for any other purpose shall require a license with payment of fee to the Cambridge Research Laboratory. All rights reserved.

CRL Technical reports are available on the CRL's web page at
http://crl.research.compaq.com.

Compaq Computer Corporation
Cambridge Research Laboratory
One Cambridge Center
Cambridge, Massachusetts 02142 USA

1 Introduction

Digital video editing and compositing is ubiquitous in film and broadcast postproduction. The recent introduction of low-cost editing systems such as Avid Cinema promises to extend this technology to home use. In these applications, video is represented as a set of layers with associated *mattes* that determine how the layers are combined during compositing. The matte for each layer specifies the contribution it will make to each pixel in each frame of the composite video.

Using the technique of chroma-keying, mattes can be generated automatically for objects that are imaged in front of a blue or green screen.[1] This approach is used, for example, to combine the image of a weather announcer with a computer-generated weather map. Unfortunately, this technique cannot be applied to existing video footage since it relies on a specialized imaging setup. The only alternative is the laborious, frame-by-frame construction of mattes using specialized editing tools. The ability to extract mattes automatically from conventional video could have a major impact on postproduction and would support the repurposing of archival video footage. It could also enable sophisticated editing of home videos by consumers.

Vision techniques for automatically segmenting a video into layers based on motion coherence [12, 11, 21] can provide a solution to matte-extraction for video editing. These techniques work well for video in which the camera motion is dominant or where there are a small number of independently-moving objects. However, they often fail on video which contains complex objects such as people undergoing nonrigid motion.

In this paper we demonstrate that a combination of figure tracking and conventional intensity-based segmentation makes matte-extraction of moving figures possible in uncalibrated, unconstrained video footage. Figure tracking is coupled with background registration to obtain a coarse segmentation of the video. This provides a starting point for a more accurate segmentation of the figure based on intensity and edge cues.

The counterpart to matte-extraction is *compositing*, in which matted video layers are combined to produce the final video output. The conventional approach to compositing uses a transparency (alpha) parameter stored with each pixel to weight the combination of colors from multiple layers [7]. Augmenting transparency information with depth information at each pixel would enable a new class of 3-D compositions. Composited pixel values would depend upon both on the relative depths between layers and the position of a virtual camera.

Image-Based Rendering [14] (IBR) can be used to compose video layers containing depth information, thereby supporting 3-D compositing and editing. In the IBR algorithm of Avidan and Shashua [1], novel images of a static scene are synthesized from a set of photos using geometrically-correct pixel reprojection. We extend this approach to the synthesis of video sequences using multiple video layers. Using our multi-layer IBR algorithm, video layers can be rendered from novel camera viewpoints. In addition, occlusion relationships between layers are maintained automatically during compositing. We believe these are the first results in applying Image-Based Rendering techniques to video editing.

We have developed a system called *SpliceWorld* which showcases the use of figure

[1] Also see a more recent variant of this idea known as Z-keying [13].

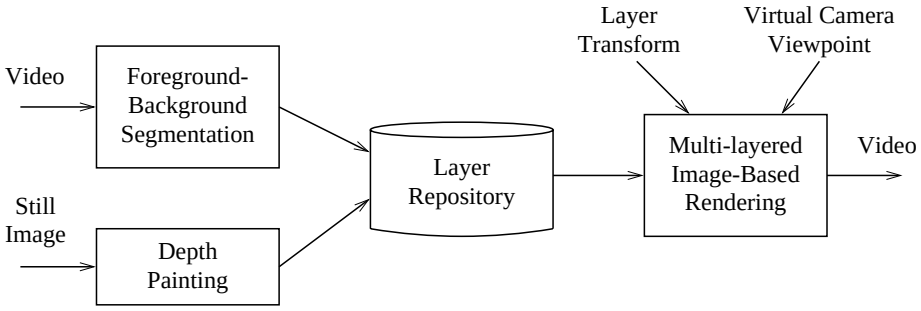

Figure 1: Architecture for the *SpliceWorld* system.

tracking and Image-Based Rendering in video editing. As we demonstrate in Section 4, our system significantly reduces the amount of labor required to produce mattes for figure motion. Using *SpliceWorld* we have segmented a dance sequence from the 1937 movie *Shall We Dance?* and composited it into a novel background.

The architecture of the *SpliceWorld* system is illustrated in Figure 1. The central data structure is the layer repository which holds the video layers that will be composited to produce the final output. Layers enter the repository through either the Figure-Background Segmentation module or the Depth Painting module. The segmentation module takes an input video clip of a moving figure and generates two layers corresponding to the figure and the background. It is described in Section 2. The depth painting module allows a user to manually construct a background layer from a single image by adding depth information. It is described briefly in Section 3.4.

Once layers have been created and placed in the repository, the multi-layer IBR module renders them to create the video output. During this process the user can reposition the layers in 3-D and change the pose of the virtual camera used for rendering. This permits a wide range of 3-D edits. This process is the subject of Section 3, which includes a more detailed description of the layer repository data structure. In Section 4 we demonstrate the result of compositing a figure layer extracted by the segmentation module with a background layer created through depth painting.

2 Figure-Background Segmentation

The automatic motion-based segmentation of video into layers has been a popular research topic in the computer vision literature. The basic approach is to fit a set of parametric motion models to video, estimating both model parameters (i.e affine or projective transforms) and a segmentation map (e.g. matte) which specifies the contribution each model makes to the observed motion of each pixel in each frame. See [21] for a detailed review.

Segmentation techniques can be divided into two classes based on whether the models interact with the pixel motions *simultaneously* or *sequentially*. In simultaneous techniques [12, 24] the Expectation-Maximization (EM) algorithm is used to automatically assign pixels to multiple layers based on the consistency of their motion. A

straight-forward application of the EM approach to articulated motion models would require each link in the model to compete for each pixel in the input video. When the number of links is large, as in the case of the figure, this can be prohibitively expensive. See [20] for details.

Sequential techniques [11, 2] assume that robust estimation of a single dominant motion is possible at each stage of a recursive algorithm. For example, when a small target is moving with respect to a fixed background the camera motion is dominant [3]. Unfortunately, videos containing moving figures often violate the dominant motion assumption. For example, the torso of the figure may remain relatively stationary while the arms or legs move, or the figure may occupy a large portion of the field of view. The result is that dominant motion analysis will often incorrectly assign figure pixels to the background model.

Figure 2 illustrates a common failure mode of the dominant motion approach in video clips with moving figures. It was obtained by stabilizing the image sequence using dominant motion analysis, and then median filtering the stabilized frames to produce a reconstructed background image [11]. Fred Astaire's torso is included in the background because it is nearly stationary for a significant number of frames in the latter part of the sequence, while his arms and legs are moving. Also significant is the blurring of the reconstruction due to errors in the camera motion computation.

Figure 2: Failure of dominant motion approach on a sequence containing figure motion is illustrated via median filtering.

While the automatic segmentation of articulated motion remains a challenging problem, there is a crucial distinction between video editing and the image coding and video indexing applications which have been the focus of previous motion segmentation work [23]: Video editing is an *interactive* process in which the user can be expected to provide an initial segmentation of the figure. In *SpliceWorld* this is done by aligning a stick figure model with the initial frame in a motion sequence. This can

be accomplished quickly and easily using a simple graphical interface.

Figure tracking techniques [18, 5] can be used to propagate the initial segmentation through the image sequence. The output of the figure tracker gives the location of the figure in each frame. Templates or bounding boxes associated with each body part can provide a coarse segmentation of the figure. Background registration can be used to recover the camera motion. Its accuracy can be greatly improved by excluding the coarsely-segmented figure region from analysis.

Once estimates of the background motion are available across the sequence, a background image can be reconstructed for each frame by filling in the pixels which are occluded by the figure. This reconstructed background can be used to refine the estimated position of the figure and generate a detailed segmentation of the figure pixels through background subtraction. It is worth noting that traditional tracking methods usually start with a good initial state and attempt to propagate it over time. We are simply extending this idea to the segmentation map.

Our complete algorithm for segmenting a moving figure from the background in the presence of unknown camera motion is shown as a block diagram in Figure 3. It consists of two modules: *Coarse Segmentation* and *Segmentation Refinement*. The coarse segmentation module couples figure tracking with background registration. Its output is a sequence of initial figure mattes and reconstructed background images, along with parameter estimates for the tracker and background motion models. In the refinement module, probabilistic foreground mattes are computed using intensity and boundary information. We now describe these two stages in more detail.

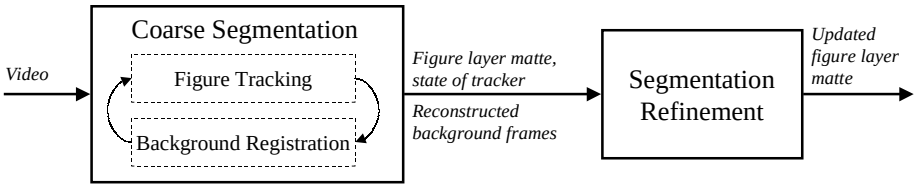

Figure 3: Figure-Background Segmentation Module

2.1 Coarse Segmentation

The coarse segmentation module implements a coupled figure tracking and background registration process. By interleaving these two steps we improve the reliability of both: Appearance-based figure tracking benefits from access to a background template, while background registration is more reliable if pixels from the figure are excluded from processing.

We employ the appearance-based figure tracker described in [18]. It is based on a 2-D Scaled-Prismatic model of figure motion in which links can rotate around their joint centers in the image plane and change length in response to foreshortening. Such a 2-D motion model is convenient for video editing as it captures the basic properties of articulated motion in the image plane without requiring a complex 3-D kinematic model. 3-D models are difficult to initialize and can be challenging to track using a

single video sequence. The tracker uses templates attached to each link to model the appearance of the figure. Figure 4 shows a template figure model of Fred Astaire in two different poses.

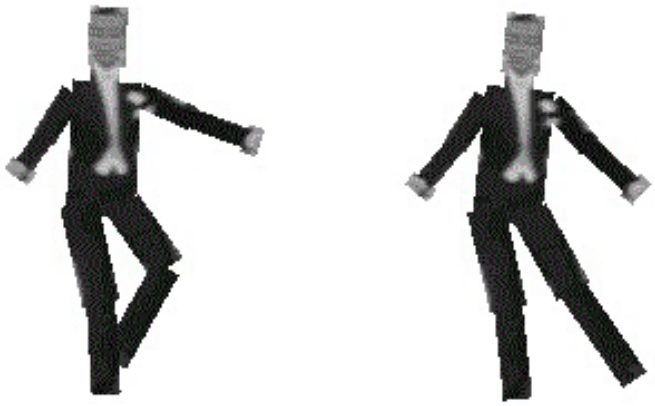

Figure 4: Scaled Prismatic Model with Fred Astaire templates shown in two configurations. Each template can rotate in the image plane and scale along the link direction.

Registration begins by predicting the state of the figure in the current frame using a dynamical model. Using the prediction as a starting point, the final estimate is produced by nonlinear least squares minimization of the pixel error between each template and the current video frame. Note that unlike conventional dominant motion algorithms, the success of the tracker does not depend upon the figure motion being independent from that of the background. In fact, the case where the figure is stationary or slow-moving with respect to the background is the easiest situation for the tracker.

However, it is difficult to construct a figure appearance model which remains reliable over long sequences. Figure motion produces dynamic appearance changes, due to out-of-plane body rotation, movement of clothing, self-occlusions, shadows and changes in illumination. Differencing with a background image can improve the robustness of the figure tracker to appearance changes. The following algorithm achieves this by coupling the background and figure registration steps:

1. In the initial frame, the state of the figure tracker is manually initialized by the user. Masking out the region covered by the figure templates gives the background image. The masks constitute the initial matte for the figure. In all subsequent frames:

2. Predict the state of the figure tracker in the current frame using the previous state estimate and a dynamic model such as constant velocity.

3. Generate an initial background frame by masking out the region underneath by the figure templates following prediction. Next register the unmasked background pixels with the previous background frame using a global motion model.

4. Fill in the holes in the current background by propagating forward pixels from previous registered background frames.

5. Using the reconstructed background and the figure appearance templates, update the figure state through nonlinear least squares minimization.

6. The projection of the figure templates under the updated state estimate gives the figure matte.

7. Repeat steps 3-6 for a few iterations. Upon convergence, step 3 gives the background motion estimate, step 4 gives the reconstructed background image, step 5 gives the figure motion estimate, and step 6 gives the figure matte. Advance to the next frame and return to step 2.

Note that although this approach significantly increases the robustness of the figure tracker to appearance variations, fully automatic tracking of long motion sequences remains an elusive goal. Fortunately the user can check for tracking failure and reinitialize the system when necessary. This task is much less burdensome then manual figure segmentation (see Section 4 for a comparison).

Figure 5 illustrates the performance of the coupled figure-background motion estimation on a 250 frame video clip from the movie *Shall We Dance?*. The top line shows the estimated figure registration for two frames in the sequence. The 2-D kinematic model is superimposed in green. Each line segment corresponds to one link in the figure model.

The mosaic at the bottom of Figure 5 illustrates the result of our background reconstruction algorithm. It was generated by transferring the segmented background pixels in each frame back into the first frame of the sequence. Pixel transfer is based on concatenating together the estimated background motions between adjacent frames. At each pixel location in frame 0, the transferred pixel values were accumulated using an exponentially weighted average. The weighting term favors pixels from frames which are closest to the frame 0. This selective weighting is possible because we have an explicit segmentation of the background in each frame.

2.2 Figure Segmentation Refinement

Since the template model assumes a simple rectangular shape for each limb, a refinement step is needed to segment the actual shapes of the limbs and the clothing. Note that the shape of the figure can vary significantly in each image frame due, for example, to the secondary motion of clothing.

The refinement step computes a probabilistic matte for the figure in which each alpha represents the probability that a particular pixel belongs to the figure layer. We write the alpha value for the ith pixel as α_i and the label variable as ω_i. Assuming that the label is either f ('figure') or b ('background'), the steps for computing the matte probabilities are:

1. Generate an initial probabilistic segmentation map based on intensity differences: $\hat{\alpha}_i = p(\omega_i = f|X, I'_i)$, where X is the previously obtained state of the

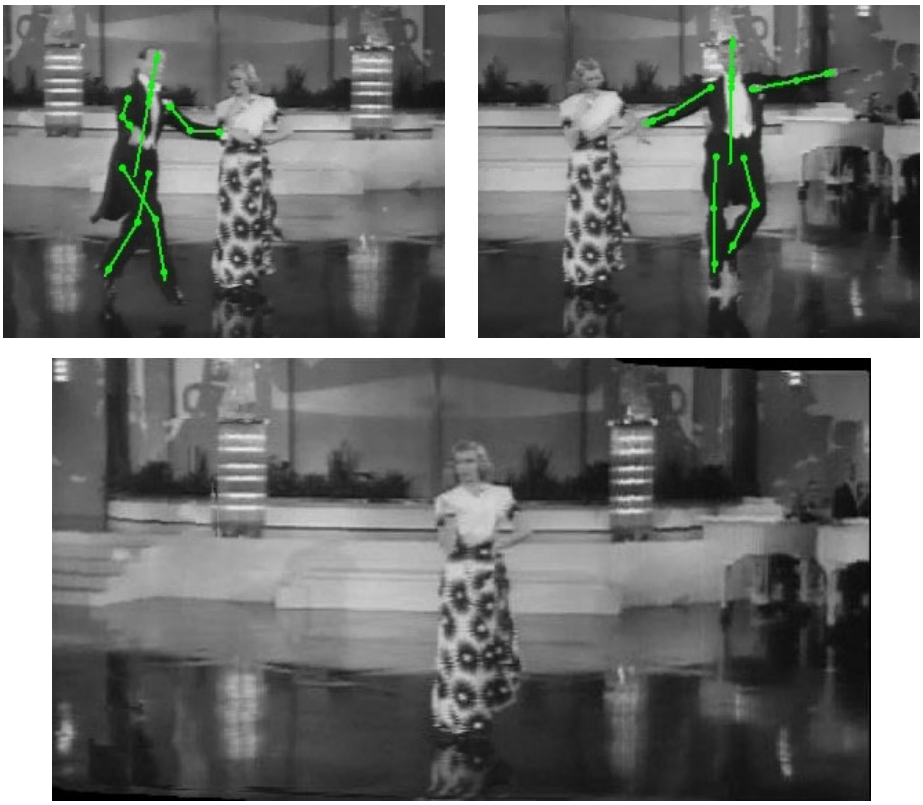

Figure 5: Top: Figure tracking results shown for two frames. Bottom: Background reconstruction and mosaicking produced by coarse segmentation stage (compare with Figure 2).

figure tracker, and I'_i is the observed intensity difference between the pixel and the corresponding pixel in the reconstructed background frame.

2. Estimate a closed boundary for the figure using a chain of cubic B-splines B. The B-splines are estimated such that the boundary lies as much as possible on the 0.5 probability points in the initial probability segmentation map.

3. Compute the final probabilistic segmentation map $\alpha_i = p(\omega_i = f|B, I'_i, X)$ by taking into account the position of the B-spline boundary.

The initial probabilistic segmentation, $\hat{\alpha}_i$, can be expressed as

$$p(\omega_i = f|X, I'_i) = \frac{p(I'_i|\omega_i = f)p(\omega_i = f|X)}{p(I'_i|X)} \tag{1}$$

The likelihood of observing the difference between the pixel intensity in an image frame and the predicted intensity from the background appearance model may be com-

puted by assuming that additive Gaussian noise has been added to the computed background pixel, i.e. $p(I'_i|\omega_i = b) \propto \exp(I'^2_i/\sigma^2)$, where the σ^2 is the pixel noise variance which can be empirically determined. For simplicity in our experiments we simply assigned a constant value to $p(I'_i|\omega_i = f)$.

The probabilities $p(\omega_i = f|X)$ depend upon estimates of the tracker state and the known template dimensions. If the template shapes and state estimates were exact, $p(\omega_i = f|X)$ would be a binary mask that assigned all pixels under the template to the figure model. Under noisy conditions, an effective alternative is to apply an isotropic Gaussian blur to the template shape, where the associated variance is set manually based on the accuracy of the tracker state.

Figure 6(a) shows a representative frame from a Fred Astaire dance sequence. Figure 6(b) shows the initial segmentation map for the figure computed using Equation 1. The probabilistic segmentation map can be directly interpreted as an alpha channel for the purpose of compositing. In compositing a segmented figure into a novel background, each pixel in a frame would contribute to the composite in direct proportion to the probability that it belongs to the foreground (figure) layer.

(a) (b) (c) (d)

Figure 6: Segmentation Refinement. (a) original figure, (b) initial segmentation map, (c) B-spline boundary estimation, (d) final segmented figure.

In some applications it may be desirable to have a binary segmentation of the figure pixels. Intensity-based segmentation is often noisy and setting a threshold on the probability to produce a binary segmentation can result in noisy boundaries. B-spline smoothing can be used to generate smooth boundaries, although there is an obvious tradeoff in preserving the high-frequency detail of the bounding contour.

The B-spline boundary may be considered to be a soft classification model which imposes a probability profile in directions normal to the curve. The profile used has a sigmodal shape of the form $p(\omega_i = f|B) = 1/(1 + \exp(d_i/s))$, where d_i is the normal displacement of the pixel from the curve in an outward direction (i.e. positive away from the figure, negative towards the figure), and s is a variable determining the sharpness of the profile. Based on this profile, the maximum-likelihood configuration of the B-spline control points is computed using standard active contour techniques[16], with the goal being to maximize

$$p(I'_i, X|B) \quad \propto \quad p(\omega_i = f|X, I'_i)p(\omega_i = f|B) +$$

$$p(\omega_i = b|X, I'_i)p(\omega_i = b|B) \quad (2)$$

The steps involved in the B-spline boundary estimation are:

1. The tracker state and original configuration of the templates are analyzed and an initial polygonal bounding curve is extracted.

2. The vertices of this polygonal bounding curve are then automatically adjusted to maximize $p(I'_i, X|B)$ in the neighborhood of each vertex. The vertices of the polygonal bounding curve are then fixed.

3. Each straight edge in the polygonal bounding curve is then separately treated as an open cubic B-spline active contour with end-points which are fixed to the end-points of the edge. If the straight active contour does not sufficiently maximize $p(I'_i, X|B)$ in its vicinity, additional control points are automatically inserted [4] to increase curve complexity and further optimized. This insertion-optimization cycle is applied a number of iterations until $p(I'_i, X|B)$ is close to the maximum.

An example of a B-spline boundary estimation result is shown in figure 6(c). Given the estimation of the B-spline boundary, the final probabilistic segmentation map $p(\omega_i = f|B, I'_i, X)$ can be expanded into products of conditional probabilities as in Equation 1. The final segmentation map is illustrated in Figure 6(d).

3 Multi-layered Image-based Rendering

We now describe the Image-based rendering (IBR) module which is used in *SpliceWorld* to generate video output (see Figure 1). IBR has emerged as an attractive alternative to traditional 3-D model-based rendering. IBR techniques operate on an input set of images to directly produce novel output images; they include morphing, mosaicking, dense sampling and interpolation, and geometrically-valid pixel reprojection [1, 17], which is the basis for our approach. See [14] for a comparative review.

The appeal of IBR stems from its ability to create novel images by transferring and interpolating pixel values, *without* an intermediate step of 3-D reconstruction and modeling. As such, IBR is well-suited to video editing applications, where one is often interested in making small changes in camera viewpoint or in the positions of objects and it is usually not feasible to accurately reconstruct the complete 3-D geometry of the scene.

We will show that by augmenting a conventional layered description of video based on color and transparency with relative depth we can employ IBR techniques to enable a new class of 3-D edits. We describe a novel algorithm for multi-layer geometrically correct pixel reprojection and use it to resynthesize existing video footage from novel camera angles with significant occlusions. We describe a technique called *depth painting* which simplifies the specification of relative depth.

The general architecture for our proposed multi-layered image-based rendering system is depicted in the block diagram in Figure 7. The input to the system is a set of models which provide the sources for novel view generation. We have identified three types of models:

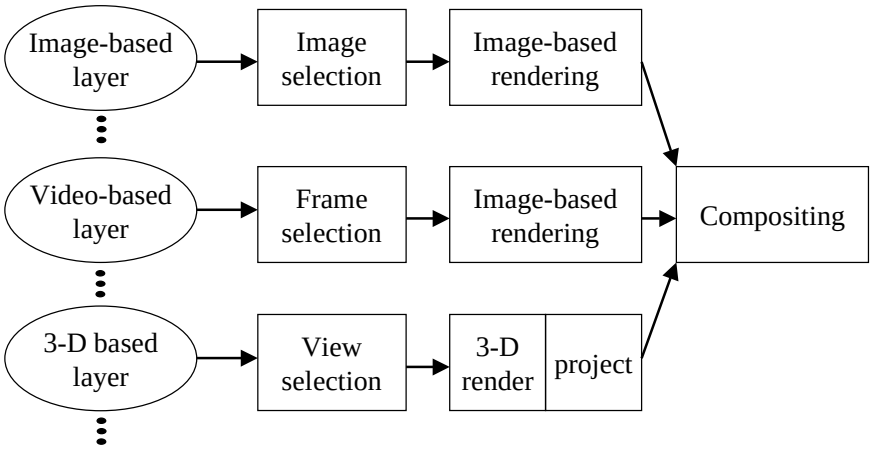

Figure 7: Architecture for multi-layered image-based rendering system.

- *Image-Based Layer*, a pixel plus depth single image or a collection of still images of a static scene,
- *Video-Based Layer*, a matted video sequence, such as a video sequence of the segmented-out figure, and
- *3-D Based Layer*, a conventional 3-D graphics model.

The relative pose and scale of each layer can be changed independently to create a new scene. Subsequently, virtual viewpoints of the new scene can then be generated. An earlier version of this module that handles only still image and 3-D model inputs is described in [15].

Image-based layers consist of either a depth-augmented image or a set of still images of a static scene. This model is often used to describe a static background, which forms the backdrop for foreground objects, such as actors and actresses on a movie set. For each collection of images, a set of correspondence maps is assumed computed. These correspondences indirectly specify the relative geometry in the image data.

The video-based layer is essentially a matted video sequence. The matte is a mask which specifies the pixels in each frame of the video that are associated to the model. Each video layer can describe a single coherent rigid body motion, or in the case of the figure, a body with complex motion. A video sequence containing multiple moving objects would produce multiple models, each one containing a different matte sequence which selects a single object. As with the still image model, each video frame has an associated correspondence map which brings its pixels into correspondence with pixels in the previous video frame.

In addition to the pixel and correspondence data, each of the still and video models also contain a description of the pose, position, and intrinsic camera parameters for the camera for each image in the model. Intrinsic camera parameters include the focal length, aspect ratio, and image skew.

The third and final type of input layer is a conventional computer graphics model, consisting of a set of explicit 3-D surfaces (whose representation may be polygons, Non-Uniform Rational B-Splines (NURBS), etc.) with texture-mapped or shaded surfaces. The 3-D model may be a volumetric model as well. There is no explicitly stored correspondence information with this model, since correspondences can be generated automatically given two viewpoints of the 3-D model.

3.1 Intra-layer indexing

Each input layer is represented either by a set of image stills, a collection of video frames, or a 3-D model. Each of these has an associated *intra-layer index* which identifies a layer component as a reference for the synthesis algorithm. An image layer, for example, consists of a number of frames, any one of which could be used as reference for synthesis. The intra-layer index specifies the reference frame. In conjunction with the desired virtual camera view, it completes the specification of the synthesis task.

In the case of a video input layer, the index specifies a particular reference frame from the sequence, around which new viewpoints can be synthesized. Similarly, for a collection of image stills, the user can choose a reference image. In each of these cases the virtual camera input specifies a camera motion relative to the camera configuration for the reference image. In the 3-D model case, the index is the reference pose at which the 3-D model will be rendered using conventional 3-D graphics. The rendered 3-D model can then be processed in exactly the same manner as the other still-image-based and video-based layers.

Once the intra-layer indexing is done, the system then makes all the layers compatible by transforming all of them to a global reference frame (for our case, corresponding to the "background" layer). This allows the creation of composite reference images, from which new views can be generated using a pixel transfer technique.

The implemented version of our multi-layered IBR system is a subset of the general architecture described above. Instead of recovering all the camera intrinsic parameters, we recover only the camera focal length from input images. We also assume that each frame of the video layer has depth that is known *a priori*. We assign a "bump" depth distribution to each frame, i.e., the depth as a monotonic function of the distance to the boundary.

3.2 Single Layer Rendering

The base representation for a layer is a pair of images with correspondences and camera motion. This representation is adopted because it is independent of camera intrinsic parameters such as the focal length. In addition, pixel transfer can be performed without intermediate depth computation using the trilinear tensor. This idea has been used to generate novel views from either two or three reference images [1]. Our work extends this framework slightly by describing efficient pixel transfer in the multi-layer case and in the case of video sequences. Other representations exist, such as pixel with (single) depth, or the LDI (Layered Depth Image) [8].

All layers are converted into the base representation for rendering. To see how our multi-layered IBR technique works, let us first delineate the steps for rendering a single

layer from two images:

1. Register images (using spline-based registration [22])

2. Recover epipolar geometry and camera focal length [6]

3. Compute trilinear/trifocal tensor and use it to transfer pixels for the creation of novel views

4. Fill intralayer holes, so-called because they occur within a layer, through direct pixel interpolation [9].

The key to rendering in our multi-layered IBR technique is the precomputation of reference viewpoints. These reference viewpoints usually correspond to the viewpoints of the "background" layer, which we define as the layer that dominates the output by virtue of its relative size. When multiple layers are added and manipulated, their appearance within the new scene at the reference views is computed based on depth ordering. Note that this is done only once. Subsequent viewpoints at perturbed locations are generated using the exact same pixel transfer method as applied to a single layer, with one notable exception: Two different types of holes can occur, and they must be filled differently.

3.3 Multi-Layer Pixel Transfer and Rendering

In multi-layered IBR, two types of holes can occur in generated images: *intralayer holes*, which occur within a layer, or *interlayer holes*, which occur between layers. Intralayer holes are created when projecting a collection of pixels within a small region to a larger region. They can be removed through any standard interpolation technique; we use the Elliptical Weighted Average filter described in [9].

In contrast, interlayer holes are the result of disocclusion between different layers. They should not be filled in the same manner as intralayer holes. Doing so would cause textures from different layers to be blended by interpolation, resulting in an undesirable mixing of textures. Intralayer holes are identified by the fact that they are surrounded by pixels from the same layer. While interlayer holes are surrounded by pixels from different layers.

To fill interlayer holes, we perform forward mapping from the layers to screen space. In the forward mapping process, only regions of each layer that are not part of the precomputed composite image are involved. The general idea is to compute, for each unexposed pixel in each layer, its new location corresponding to the new camera viewpoint. If a pixel is mapped to an interlayer hole, its depth is computed and stored. Once this is done for all the unexposed pixels, depth ordering is then used to determine the right pixel to expose. Note that depth computation and comparison are necessary *only* at the interlayer hole locations. We do not employ the alternative inverse mapping technique based on epipolar search due to its sensitivity to errors in correspondence and its higher computational expense.

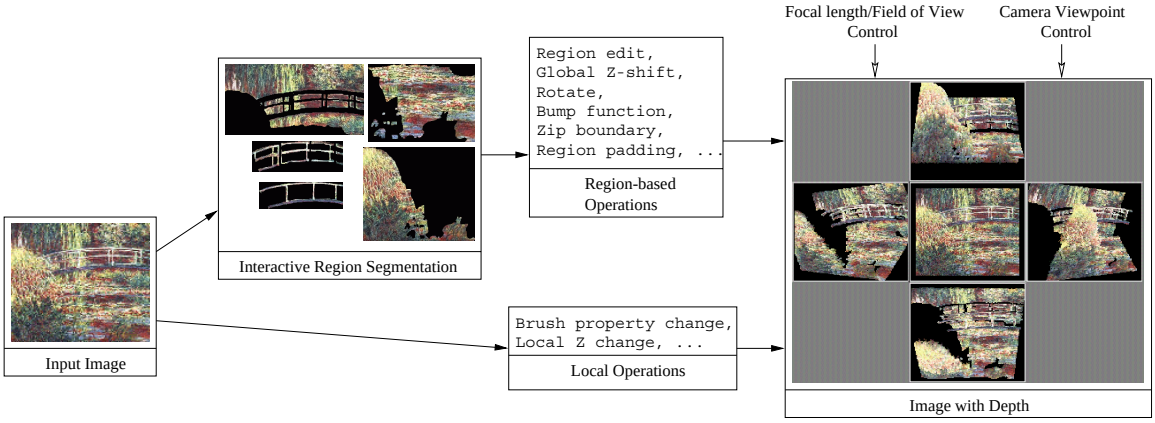

Figure 8: The depth painting system, with the interface shown on the right.

3.4 Depth Painting to Create Layers

In order to expand our repertoire of image-based layer inputs, we have devised an intuitive method, which we call *depth painting*, for generating depths for single images. Figure 8 shows the overall functionality of our depth painting system. In general, the system allows interactive specification of regions within the image as well as interactive change of depth. The user can define regions and objects by tracing their boundary curve using the mouse. Obviously, more sophisticated techniques for interactive boundary extraction, such as the "Intelligent Scissors," [19], can be used.

Critical to the system is the display of four auxiliary side views corresponding to left, right, up, and down side views (right of Figure 8). These views help add the third dimension to the input image. They facilitate visualization of the user's progress; any change in depth can be quickly propagated across these views.

This system allows the user to assign depth to a single image in an intuitive way. The operations used to modify the image depth can be roughly classified as either local or region-based. During local operations, which can affect any part of the image, the user uses the mouse as a "brush" that adds or subtracts depth within the footprint of the brush. Region-based operations, however, affect only the current region of interest. The user can also manipulate the viewpoint of the central image at any time in addition to changing camera parameters such as focal length or fields of view.

In contrast with Horry *et al.*'s "Tour into the picture" system [10], which adds the third dimension to the single image through polygons and planar regions, our system provides a means for more expressive depth variations that result in more realistic rendering.

4 Experiments and Results

As a proof of concept test of our editing system, we extracted a a segmented figure layer of Fred Astaire from a 170-frame dance sequence from the movie *Shall We Dance?*, and

composited this with a scene from the movie *American in Paris* using depth-painting and image-based rendering. There were two objectives in this task: (1) we wanted to compare the time and effort required to carry out figure segmentation using our figure-background decomposition system and that required using purely manual effort, and (2) we wanted to observe the effects and quality of the video generated using depth-painting and image-based rendering.

The first task involved the application of the coupled figure tracking and background registration system to the video clip, as was illustrated in Figure 3. A 2D 19-DOF articulated model was used for the tracking, while a global translation model was used for the background registration. The final result required 33 manual restarts to correct for both minor (22 slight misregistrations) and major (11 tracking failures) errors. This amounted to approximately 5 frames per restart, although the distribution of restarts across the sequence was quite uneven and depended upon the complexity of the figure motion. The total operation time was 1 hour 6 minutes, which includes computation time for both figure tracking and background registration/reconstruction.

The second task is segmentation refinement. This task took 1 hour 35 minutes but is fully automatic. A sample frame from the segmentation output may be seen in figure 9. As the segmentation still contains some gross errors, manual refinement have to be carried out on the segmentation, an example of which is also shown in figure 9.

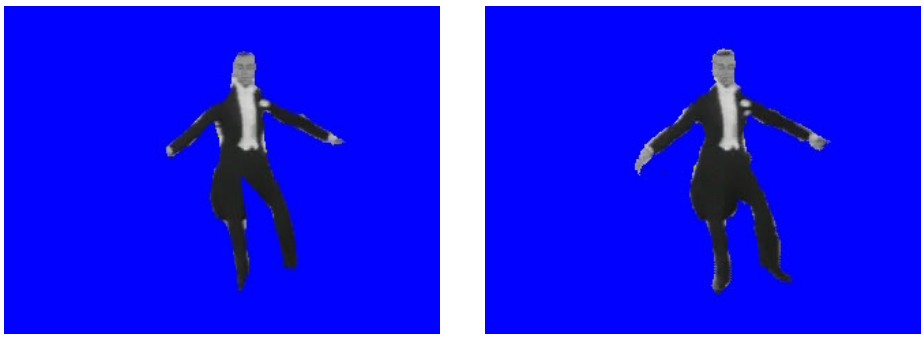

Figure 9: Left: automatic figure segmentation. Right: manually refined segmentation.

Two sets of timing tests were conducted for the manual segmentation: the first involves totally manual segmentation, the second involves refining the automatic segmentation. In both cases the users carried out the segmentation on Adobe Photoshop. Results are shown in Table 1. The time for the first frame is shown separately from successive frames because in the latter case the manual segmentation process can be bootstrapped by using the segmentation map from the previous frame because of temporal continuity. The greatest gain achieved by the refinement process is therefore in the first frame. From the results shown, it can seen that even in the case of successive frames, the refinement task achieves a 40 percent increase in efficiency compared to manual segmentation.

A further comparison can be made on total time required to complete the segmentation as shown in table 2. The results in the table show the projected total times needed for the 170-frame sequence. In the case of the system, a portion of the total time is

involved in figure tracking which requires more human attention than painstaking high precision work, and a further portion of the time is used in segmentation refinement which is totally automatic. The speedups show the efficiency gain in comparison with the high-precision manual segmentation effort.

	First Frame	Successive Frames
Fully manual	12mins 36s	5mins 18s
Refinement	3mins 29s	3mins 13s

Table 1: Comparative Timings on Manual Segmentation and Segmentation Refinement Tasks per Frame

After the segmentation was completed, the figure layer was composited with a depth-painted scene from *An American in Paris*. Figure 10(a) illustrates the disoccluded regions that become visible during view synthesis from a single background still. Figure 10(b) shows the result after depth painting is used to pad these regions.

(a) (b)

Figure 10: Frame 10 of video composited using image-based rendering: (a) unpadded scene layer with automatically segmented figure layer, (b) padded scene layer.

By specifying a variety of camera-motions, the 3D parallax effects may be observed when image-based rendering is applied. The result for several frames in the motion sequence is shown in Figure 11. An important point is that the motion of Fred Astaire is matched to that of the scene layer. This can be automatically done because the original background motion was recovered through background registration.

We informally compared the computational cost of IBR to conventional graphics pipeline rendering. *SpliceWorld* can generate depth-encoded 240×320 images through pixel reprojection at 8-10 frames/sec on a 400 MHz PC. In contrast, generating images of comparable quality using 3-D models took several seconds per frame on the same machine. We obtained this latter result by converting an image layer into a 3-D mesh with one vertex per pixel and rendering it using 3D Studio MAX.

Figure 11: Composition of segmented figure into novel multi-layer background. Note change in viewpoint and occlusion by foreground objects.

	Time	Speedup
Fully manual Total	15.1 hrs	1.0
System Total	11.8 hrs	1.28
human attention	10.2	1.48
human high precision	9.1 hrs	1.66

Table 2: Comparison of Total times for Segmentation

5 Summary and Conclusions

Video editing systems can benefit substantially from two lines of computer vision research: automatic segmentation of video into multiple layers, and image-based rendering (IBR). Automatic segmentation greatly reduces the effort involved in constructing layered video representations, while IBR enables a new class of 3-D edits in which the camera position and the spatial relationships between objects can be manipulated directly.

We have demonstrated the use of figure tracking to segment complex articulated motion which is not amenable to current motion-based segmentation methods. Coupling figure tracking with background registration improves both the quality of the background reconstruction and the tracker's performance (see Figure 5).

IBR is well-matched to the video editing problem, as it is very effective when the desired changes in camera viewpoint are small. While editing is based on multiple layers, however, conventional IBR systems have worked with only a single group of photos [1]. Our multi-layer IBR algorithm addresses the problems of interlayer hole-filling and efficient rendering through the use of precomputed reference views. We have shown that techniques developed for still photos can be easily applied to video sequences.

We have implemented a prototype video editing system, called *SpliceWorld*, which incorporates video segmentation and IBR. Using our system, we have demonstrated a 40% decrease in the time required to segment a 170-frame video sequence.

References

[1] S. Avidan and A. Shashua. Novel view synthesis in tensor space. In *Conference on Computer Vision and Pattern Recognition*, pages 1034–1040, San Juan, Puerto Rico, June 1997.

[2] S. Ayer, P. Schroeter, and J. Bigün. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *Proceedings of Third European Conference on Computer Vision*, volume II, pages 316–327, Stockholm, Sweden, May 1994.

[3] Peter J. Burt, Rajesh Hingorani, and Raymond Kolczynski. Mechanisms for isolating component patterns in the sequential analysis of multiple motion. In *Proc.*

IEEE Workshop on Visual Motion, pages 187–193, Princeton, NJ, October 7-9 1991.

[4] Tat-Jen Cham and Roberto Cipolla. B-spline curve representation with MDL-based active contours. In *Proc. 7th British Machine Vision Conference, Edinburgh (Scotland)*, pages 363–372, 1996.

[5] Tat-Jen Cham and James M. Rehg. A multiple hypothesis approach to figure tracking. In *Proc. Computer Vision and Pattern Recognition*, pages 239–245, Ft. Collins, CO, June 1999.

[6] Olivier Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.

[7] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 2nd edition, 1990.

[8] S. J. Gortler, L.-W. He, and M. F. Cohen. Rendering layered depth images. Technical Report MSTR-TR-97-09, Microsoft Research, Microsoft Corp., March 1997.

[9] N. Greene and P. Heckbert. Creating raster Omnimax images from multiple perspective views using the Elliptical Weighted Average filter. *IEEE Computer Graphics and Applications*, pages 21–27, June 1986.

[10] Y. Horry, K. Anjyo, and K. Arai. Tour into the picture. *Computer Graphics (SIGGRAPH'97)*, pages 225–232, 1997.

[11] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, February 1994.

[12] Allan Jepson and Michael Black. Mixture models for optical flow computation. In *Proc. Computer Vision and Pattern Recognition*, pages 760–761, New York City, NY, June 1993.

[13] Takeo Kanade, Kazuo Oda, Atsushi Yoshida, Masaya Tanaka, and Hiroshi Kano. Video-rate Z keying: A new method for merging images. Technical Report CMU-RI-TR-95-38, Carnegie Mellon, 1995.

[14] Sing Bing Kang. A survey of image-based rendering techniques. In *Videometrics VI (SPIE International Symposium on Electronic Imaging: Science and Technology)*, volume 3641, pages 2–16, San Jose, CA, January 1999.

[15] Sing Bing Kang and Huong Quynh Dinh. Multi-layered image-based rendering. In *Graphics Interface*, pages 98–106, Kingston, Ontario, Canada, June 1999.

[16] Michael Kass, Andy Witkin, and Demitri Terzopoulos. SNAKES: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[17] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. *Computer Graphics (SIGGRAPH'95)*, pages 39–46, August 1995.

[18] Daniel D. Morris and James M. Rehg. Singularity analysis for articulated object tracking. In *Proc. Computer Vision and Pattern Recognition*, pages 289–296, Santa Barbara, CA, June 1998.

[19] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. *Computer Graphics (SIGGRAPH'95)*, pages 191–198, August 1995.

[20] Henry A. Rowley and James M. Rehg. Analyzing articulated motion using expectation-maximization. In *Proc. Computer Vision and Pattern Recognition*, pages 935–941, Puerto Rico, June 1997.

[21] H. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):814–830, August 1996.

[22] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 194–201, Seattle, Washington, June 1994. IEEE Computer Society.

[23] J. Wang and E. Adelson. Layered representation for motion analysis. In *Proc. IEEE Conf. Comput. Vis. and Pattern Rec.*, pages 361–366, 1993.

[24] Yair Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proc. Computer Vision and Pattern Recognition*, pages 520–526, San Juan, Puerto Rico, June 17-19 1997.

COMPAQ
CRL 99/8
December 1999

Video Editing Using Figure Tracking and Image-Based Rendering

James M. Rehg Sing Bing Kang Tat-Jen Cham