Tour Into the Video: Image-based Navigation Scheme for Video Sequences of Dynamic Scenes

Hyung Woo Kang
Korea Advanced Institute of Science and Technology
373-1 Kusung-dong Yusung-gu
Taejon, South Korea
henry@jupiter.kaist.ac.kr

Sung Yong Shin
Korea Advanced Institute of Science and Technology
373-1 Kusung-dong Yusung-gu
Taejon, South Korea
syshin@jupiter.kaist.ac.kr

ABSTRACT

Tour Into the Picture (TIP) is a method for generating a sequence of walk-through images from a single reference image. By navigating a 3D scene model constructed from the image, TIP provides convincing 3D effects. This paper presents a comprehensive scheme for creating walk-through images from a video sequence by generalizing the idea of TIP. The purpose of this work is to let users experience the feel of navigating into a video sequence with their own interpretation and imagination about a given scene. To generate images from new viewpoints, we first extract the background and the foreground information from the video, and then exploit the notion of a vanishing circle to construct a 3D scene model. The proposed scheme covers various types of video films of dynamic scenes such as sports coverage, cartoon animation, and movie films, in which objects are continuously changing their shapes and locations. It can also be used to produce a variety of synthetic video sequences by importing and merging dynamic foreign objects with the original video.

Categories and Subject Descriptors

I.3.3 [**Computer Graphics**]: Picture/Image Generation-Viewing algorithms; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism-Color, shading, shadowing, and texture

Keywords

Video sequence, Image-based rendering, Animation

1. INTRODUCTION

1.1 Motivation

Real-time generation of photorealistic images is a recurring theme in computer graphics. Recently, a novel approach for real-time realistic image generation called image-based rendering, has been of much attention. Tour Into the Picture (TIP) proposed by Horry et al. [9] is one of the image-based methods for generating a sequence of walk-through images from a single reference image. By navigating a 3D scene model constructed from the image, TIP provides convincing 3D effects. Assuming that the image has one vanishing point, they proposed a scene modeling scheme called spidery mesh with which the users can make their imagination real as they find or feel in the 2D image.

Due to the widely available video imaging devices such as camcorders or CCTVs and the growth of the internet which abounds with digitized movie files, video clips are now emerging as familiar sources for image-based techniques. This trend motivates the need for a more general version of an image-based navigating scheme that can deal with video sequences. However, we cannot directly apply the original TIP to video sequences since TIP handles only a still image. While various image-based techniques have been proposed, most of them use a set of still photographs [4, 5, 8, 14, 16] rather than video sequences. Moreover, previous work on video sequences usually focused on automatic reconstruction of structures from static scenes in which only the camera moves around fixed objects [2, 23].

This paper presents a scheme for creating walk-through images from videos by generalizing the idea of TIP. The proposed scheme aims at helping users experience the feel of navigating into the video sequence with their own interpretation of the scene, and create new synthetic videos by importing and compositing foreign objects according to their own imagination. To generate images from new viewpoints, we first extract the background and the foreground information from the video, and then exploit the notion of a *vanishing circle* to construct a 3D scene model. Our scheme covers various types of video films of dynamic scenes such as TV broadcast, cartoon animation, and movie films, where objects are allowed to change their shapes and locations continuously.

1.2 Related work

McMillan et al. explained image-based rendering with the notion of a plenoptic function, which defines the radiant energy to an eye position through every incident direction [16]. For example, an environment map is a sample of a plenoptic function at a fixed viewpoint [4]. An image-based rendering scheme based on environment maps has a major limitation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for pro£t or commercial advantage and that copies bear this notice and the full citation on the £rst page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior speci£c permission and/or a fee.
VRST'02, November 11–13, 2002, Hong Kong.
Copyright 2002 ACM 1-58113-530-0/02/0011 ...$5.00.

that the viewpoint is fixed. One way to relax this limitation is to use a warp function that describes the relative movement of each pixel with respect to camera movement [5, 16]. Alternatives are to construct a light field from a set of plenoptic samples such as multiple reference images taken at regular grid points [8, 14].

TIP (Tour Into the Picture) proposed by Horry et al. [9] generates realistic walk-through images by constructing a simple 3D scene model from a 2D image. However, with the assumption that the image has a single vanishing point, their modeling scheme requires major modification when the image contains multiple vanishing points or no clearly-identified vanishing point. Liebowitz et al. provided how to compute plane rectification or plane orientation to reconstruct architectural models from a single image, exploiting various geometric constraints such as parallelism and orthogonality [15].

Recently, Kang et al. proposed a new modeling scheme for TIP based on a vanishing line [11], which is simpler than that of Horry et al., and yet more general to cover a broader class of input images. They also showed that their modeling scheme is naturally extended to navigation into a panoramic image, by introducing the notion of a vanishing circle. Compared to the conventional panoramic image viewers (such as QuickTimeVR^R), their method can provide the real sense of walk-through or navigation into the panoramic scene by enabling continuous camera translation as well as rotation. In this paper, their modeling scheme for a single planar or panoramic image is further extended to a video sequence.

The image-based rendering techniques described above are common in that they all use one or more static images to obtain 3D scene information for generating an image viewed from a new viewpoint. On the other hand, some researchers, especially in the field of computer vision, have concentrated on automatically extracting 3D information from video sequences [2, 23]. These approaches have usually been focusing on static scenes, which contain relatively simple architectural models such as buildings or houses. Thus, they cannot be applied to objects that are moving arbitrarily or changing their shapes continuously, e.g., pedestrians and soccer players. Thus, dynamic scenes of this kind have usually been the target of 2D motion tracking or segmentation, rather than that of 3D model reconstruction.

1.3 Overview

In this paper, we present an image-based navigation scheme for video sequences of dynamic scenes. Our scheme is based on the following assumptions: First, the input video is composed of a continuous sequence of images for a scene. Second, only a negligible amount of motion parallax effects appear in the video. Third, the terrain on the ground (which appears in the video) is smooth enough so that it can be modeled as a single plane. In general, most of the video sequences containing dynamic scenes satisfy these assumptions: That is, dynamic objects seldom move on a rough terrain, and it is hard to make a large amount of camera translation (which causes a strong parallax effect) while tracking the moving objects with the video camera at the same time.

Fig. 1 shows the process flow diagram of our Tour Into the Video (TIV) scheme. First, a single background image is generated from an input video sequence. The background image covers all the region viewed from the entire sequence of frames, and contains only the static entities in the scene,

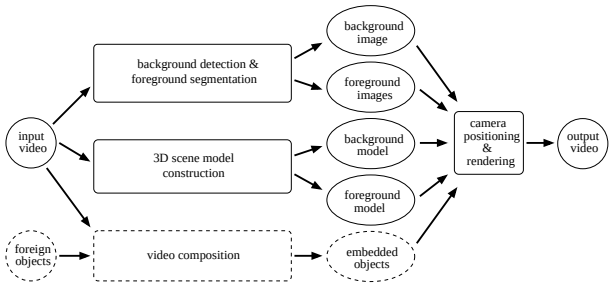


Figure 1: Schematic of TIV

that is, the background and static foreground objects. We generate the background image by employing an automatic background detection technique [7, 22] in conjunction with an image alignment (registration) algorithm based on a 4-parameter motion model to compute a camera pose for each frame [21].

For each static foreground object, its corresponding region is interactively extracted from the background image, for which we use a highly interactive image segmentation tool called enhanced lane [12]. In the case of dynamic foreground objects, their regions should be identified in each frame. With the camera poses and the background image obtained from the background detection process, we can extract the boundary information of dynamic foreground objects in each frame, by creating the gradient map of the difference image between each frame and its background image. Setting this gradient information as an initial guess, we go through a frame-by-frame iterative segmentation process based on active contour or snakes [13] to correct the given boundary.

Given the background and the foreground information thus extracted, we construct the 3D scene model which consists of a background model and foreground models. The modeling scheme for video sequences presented here evolved from that for a single image, in that it receives a single background image as an input. The background model is first constructed based on a vanishing circle detected in the background image. For the static or dynamic foreground objects, we first place 2D polygons bounding the extracted objects in the background image or the reference frame. They are then modeled as polygons in 3D space and attached to the background model after their 3D coordinates are computed. The regions inside their corresponding 2D polygons serve as their foreground texture maps (called foreground image) where only the exact portion of the objects are marked as visible.

With the constructed scene model and all the texture images prepared, the dynamic scene can be navigated by positioning the camera and successively creating images viewed from new viewpoints. Note that the polygon for each dynamic foreground object is continuously changing its shape and location on the scene model from frame to frame. It is also possible to create a new synthetic video sequence by importing foreign objects, either static or dynamic, into the scene. Since all the foreground objects are modeled as polygons with textures, even complex objects with arbitrary colors can be inserted to enrich the virtual environment. The AR (Augmented Reality) capability of this type can be used for post-production in the film industry [1].

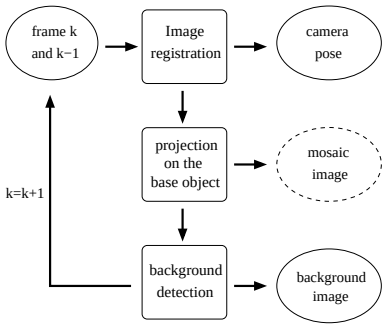


Figure 2: Block diagram for background image generation

The remainder of this paper is organized as follows. In Section 2, we present an automatic background image generation method. Section 3 discusses the semi-automatic boundary segmentation process, with which static or dynamic foreground objects can be effectively extracted from a video sequence. In Section 4, the construction scheme for the background model and the foreground object models is described in detail. Section 5 provides some experimental results with example video sequences. Finally, this paper is concluded in Section 6.

2. BACKGROUND IMAGE GENERATION

To construct a single background image from an input video sequence, successive frames in the sequence should be aligned first. The image alignment (or registration) algorithm in this work uses a 4-parameter motion model that can handle camera rotation and zoom [21]. Each registered frame is then projected on a spherical base object to generate a single image. An appropriate color value should be assigned to each pixel of the resulting image so that the pure background information remains with all the dynamic foreground objects removed. This process is called background detection, for which we adopt a pixel-based adaptive, statistical model [7, 22]. Fig. 2 shows the block diagram for our background image generation process.

2.1 Image registration

For aligning frames in the video sequence, we employ the 4-parameter motion model proposed by Shum et al. [21] which incorporates both camera rotation and zoom. Compared to the traditional 8-parameter motion model, this model provides faster and more robust convergence to the aligned position. Another benefit of this model is that it explicitly computes the camera pose for each frame, which is essential for generating a background image on a base object other than a plane. As mentioned in the previous section, we assume that the motion parallax effects are negligible in the input video. That is, the factor of camera translation can be ignored so that we can fix the camera position at a single point in 3D.

When two images are taken from the same viewpoint but in different directions, the relationship between the two images can be described by a planar homography. Thus, one image is warped into another using a 3×3 matrix $\mathbf{H}$ as $\mathbf{x}' \sim \mathbf{H}\mathbf{x}$, where $\mathbf{x} = (x, y, 1)$ and $\mathbf{x}' = (x', y', 1)$ are homogeneous coordinates, and $\sim$ indicates equality up to scale. For a camera centered at the origin, the relationship between an image point $\mathbf{x}$ and its corresponding 3D point $\mathbf{p} = (X, Y, Z)$ can be described by $\mathbf{x} \sim \mathbf{K}\mathbf{R}\mathbf{p}$, where $\mathbf{K}$ and $\mathbf{R}$ are a simplified camera calibration matrix and a 3D rotation matrix, respectively. Without loss of generality, we assume that the origin of the pixel coordinate is at the image center. The planar homography $\mathbf{H}$ between two frames k and $k-1$ is then given by

$$\mathbf{H} \sim \mathbf{K}_k \mathbf{R}_k \mathbf{R}_{k-1}^{-1} \mathbf{K}_{k-1}^{-1} \tag{1}$$

where $\mathbf{K}_k$ and $\mathbf{R}_k$ respectively denote the camera calibration matrix and the rotation matrix for frame k. The rotation matrix can be recovered by incrementally updating $\mathbf{R}_k$ based on the angular velocity $(\omega_x, \omega_y, \omega_z)$, and a focal length f_k of $\mathbf{K}_k$ can be adjusted by setting $f_k \leftarrow (1 + e_k) f_k$, where e_k is the incremental change of the focal length.

We initially place the current frame k and the previous frame $k-1$ at the same position. To align the two frames, we find $\mathbf{q} = (\omega_x, \omega_y, \omega_z, e_k)$ that minimizes the squared error metric

$$E(\mathbf{q}) = \sum_i \left[I_k(\mathbf{x}_i') - I_{k-1}(\mathbf{x}_i) \right]^2 \tag{2}$$

The least-squares problem given by Equation 2 can be solved through the standard procedure in [19]. By minimizing $E(\mathbf{q})$, we estimate the incremental rotation vector $(\omega_x, \omega_y, \omega_z)$ and the incremental change of the focal length e_k, after which $\mathbf{R}_k$ and $\mathbf{K}_k$ can be updated.

2.2 Projection on a base object

After the camera pose for the current frame is obtained by the image registration process, the frame is projected onto a base object to create a single background image. While there can be various candidates for the base object including a plane, a cylinder, a sphere, and a cube [4], we select the sphere as it is the most general type. For example, a spherical base object can deal with the entire viewing range covered by the video sequence even if it covers more than 180° in the horizontal direction, and more than 90° in the vertical direction.

The projective mapping is done by using the camera pose information (rotation matrix R_k and focal length f_k). For example, we can construct a spherical background image by first converting each pixel $\mathbf{x}''$ on this image into its corresponding 3D direction vector $\mathbf{p} = (\cos\theta\cos\phi, \sin\theta\cos\phi, \sin\phi)$, and then determining its mapping onto each frame k using $\mathbf{x} \sim \mathbf{K}_k \mathbf{R}_k \mathbf{p}$ and assign an appropriate pixel color to form an updated spherical background image. Fig. 3 shows the projective mapping of two successive frames onto the spherical base object.

2.3 Background detection

While projecting each frame on the base object, each pixel on the constructed image should be updated with an appropriate color value. To obtain a pure background image from a sequence of images, we adopt a pixel-based adaptive, statistical background detection method [7, 22]. The intensity values of a pixel over time are considered as a stochastic process, which means a time series of pixel values. For each pixel $\mathbf{x}''$ on the background image I_B, we keep track of its history

$$\{z_i : z_i = I_B(\mathbf{x}'', i), 1 \le i \le k\} \tag{3}$$

where $I_B(\mathbf{x}'', i)$ is the intensity value at $\mathbf{x}''$ at the ith frame in the sequence. The recent history of each pixel is modeled

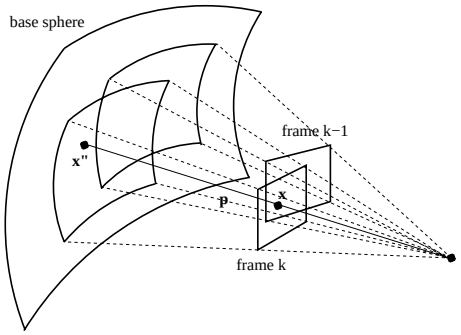


Figure 3: Projection on a spherical base object

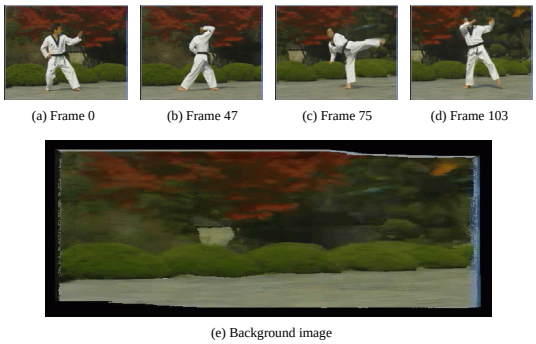


Figure 4: Background image obtained from example input video

by a mixture of N Gaussian distributions and the probability of observing the current pixel value is

$$P(z_k) = \sum_{i=1}^{N} \omega_{i,k}\eta(z_k, \mu_{i,k}, \sigma_{i,k}^2) \quad (4)$$

where N is the number of distributions, $\omega_{i,k}$ is an estimate of the weight (indicating the number of occurences of the intensity value which is accounted for by this Gaussian) of the ith Gaussian in the mixture at time k, and where η is the corresponding Gaussian probability density function with a mean value $\mu_{i,k}$ and a variance $\sigma_{i,k}^2$.

Every new pixel value z_k is checked against the existing N Gaussian distributions to find if a match occurs. A match is defined as a pixel value within the standard deviation (possibly multiplied by a positive constant) of a distribution. When two or more matches occur, only the best matched distribution is chosen by comparing the relative distance from the average value normalized by the standard deviation. If none of the N distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean, an initially high variance, and low prior weight given by users.

The weights of the ith distributions at time k, $\omega_{i,k}$, are adjusted as follows:

$$\omega_{i,k} = (1-\alpha)\omega_{i,k-1} + \alpha\xi_{i,k} \quad (5)$$

where α is the learning rate, and $\xi_{i,k}$ is one for the model which is decided as a best match and zero for the remaining models. This formulation renormalizes the weights automatically.

The μ and σ parameters for unmatched distributions remain the same. The parameters of the distribution which matches the new observation are updated as follows:

$$\mu_k = (1-\alpha)\mu_{k-1} + \alpha z_k \quad (6)$$

$$\sigma_k^2 = \min(\sigma_{min}^2, (1-\alpha)\sigma_{k-1}^2 + \alpha(z_k - \mu_{k-1})^2) \quad (7)$$

where a minimum variance σ_{min}^2 is introduced as a threshold to keep the variance from decreasing below a minimum value in case there is little change in a pixel value over a long period of time [7].

After all the parameters are updated, we determine the current background distribution as the one with the highest value of ω/σ, which means the distribution has the most supporting evidence and the least variance. Thus, the complete background image is obtained when this update procedure is done for all the pixels and for all the frames. Fig. 4 shows a background image obtained from an example input video sequence taken from a rotating camera.

3. FOREGROUND IMAGE GENERATION

The foreground (texture) image for a static foreground object is extracted just once from the background image, by placing a bounding quadrangle around it. To generate a complete foreground image, we first need the boundary information to distinguish the exact portion of the foreground object from the background portion within the foreground image. For effective extraction of the boundary information, we developed a highly interactive image segmentation tool called *enhanced lane*, a modified version of intelligent scissors [17]. Based on graph search over the localized window which follows the feature points, the enhanced lane provides both accuracy and time-efficiency in following the target boundary interactively. For more details on the enhanced lane, the readers are referred to [12]. After the boundary extraction, we assign alpha values of one inside the object boundary and zero elsewhere so that the realistic 3D effects can be produced during the navigation.

For a dynamic foreground object, its foreground image is provided at each frame as the object may change its boundary shape from frame to frame. Thus, foreground image generation for a dynamic foreground object boils down to boundary extraction at each frame in the video sequence. The boundary information of the dynamic objects can be obtained from the result of the background detection process. We first warp each frame k using its camera pose information given by the registration process, and generate the difference image I_k^D by computing $|\mathbf{x} - \mathbf{x}''|$. Then, its gradient magnitude image $|\nabla I_k^D|$ for each frame k highlights the boundaries of the dynamic objects (Fig. 5c).

However, this statistical result may not be accurate (at least partially) due to noises or ambiguities. Thus, we also provide a frame-by-frame iterative boundary tracking scheme based on active contour or snakes [13]. The enhanced lane is used to trace the initial boundary curve of the object in the first frame, which is tracked by snakes in the successive frames. As image (external) forces for attracting snakes at each frame k, we use the gradient map of the difference image $|\nabla I_k^D|$. This can be further convolved with a Gaussian smoothing filter ($G_\sigma * |\nabla I_k^D|$) to attract distant snake. Some user interaction may be necessary to post-edit the constructed boundary again with the the enhanced lane, where digression occurs. The foreground image of each object is

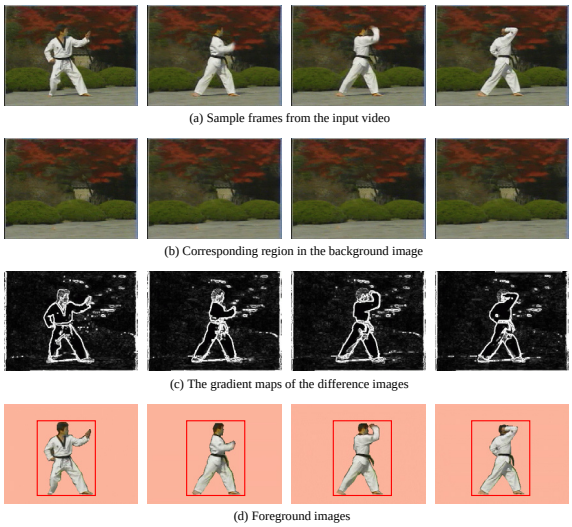


Figure 5: The foreground images from example input video

then constructed by locating around it a bounding box in which only the foreground pixels inside the boundary are marked as visible (Fig. 5d).

4. SCENE MODEL CONSTRUCTION

Given the background and the foreground information thus extracted, we construct the 3D scene model consisting of a background model and foreground object models. The background model is obtained from the background image, provided with the vanishing circle. Foreground objects are divided into two types, static or dynamic, each of which is constructed by its own modeling scheme.

4.1 Background model

Since the background image can be thought of as a spherical panoramic image, we can directly use the modeling scheme for TIPP (Tour Into the Panoramic Picture) which is based on the notion of a vanishing circle [11]. The vanishing circle, formed by a set of vanishing points on the base sphere, divides the base sphere into two disjoint hemispheres. The lower hemisphere corresponds to the ground plane in the 3D environment, and the upper hemisphere corresponds to the space above the ground plane. Thus, the vanishing circle can be thought of as the horizon that separates the earth represented by the ground plane from the sky. If we inversely project the vanishing circle back to the scene, it is mapped to a set of points at infinity on the ground plane. Each of these points is an ideal point in the direction from the viewpoint to a point on the vanishing circle.

Based on this observation, we first specify the vanishing circle interactively and then project the upper hemisphere of the base sphere on the *back hemisphere*, which has an arbitrarily large radius centered at the camera position (Fig. 6). The lower hemisphere is projected onto the ground plane. In practice, the back hemisphere is set to have some finite radius to avoid computational difficulty. This is equivalent to slightly moving down the vanishing circle from the horizon plane.

The correspondence between the points on the base sphere

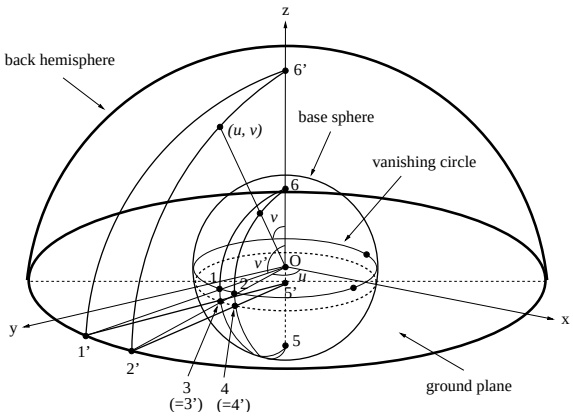


Figure 6: Base sphere and the background model

$(1 \sim 6)$ and those on the background model $(1' \sim 6')$ is shown in Fig. 6. We assume that the base sphere and the camera are centered at the origin, the initial view-up vector is towards the +z direction, the horizon plane is coincident with the x-y plane, and the height of the camera from the ground plane is h. Thus, a point (u, v) on a base sphere is projected to the point $(-h\cos u \sin v/\cos v', -h \sin u \sin v/\cos v', -h\cos v/\cos v')$ on the back hemisphere if $v < v'$; otherwise, it goes to $(h\cos u \tan v, h \sin u \tan v, -h)$ on the ground plane, where v' denotes the angle between the positive z-axis and the vector from the origin to a point on the vanishing circle [11].

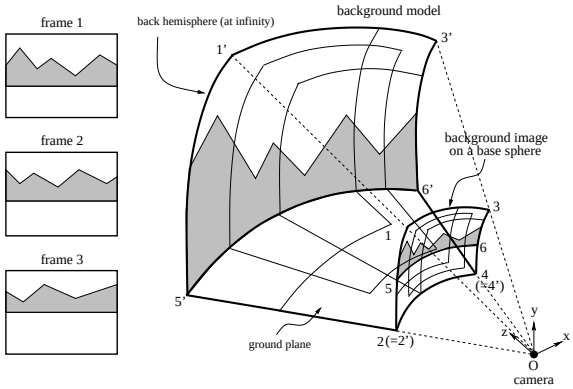


Figure 7: Background model constructed from a video sequence

Fig. 7 shows a background model constructed from an example video sequence. The background texture image is obtained by first segmenting out the static foreground objects from the background image and fill in the holes by inpainting techniques [3, 6]. We divide the image into two disjoint subimages using the line on the background image corresponding to the vanishing circle of the base sphere. The upper subimage serves as a texture map for the back hemisphere, and the lower subimage for the ground plane.

When we use a hardware texture mapping, we have to make sure all the texture maps are for linear (line-to-line) perspective mapping. Since the spherical image contains

a line-to-arc map, we cannot directly use this image as a texture map for the ground plane. Thus, we convert the lower subimage into a linear map using a projective mapping from the base sphere onto the ground plane.

4.2 Foreground model

4.2.1 Static foreground model

A static foreground object extracted from the background image is modeled as a polygon (usually a quadrangle bounding the extracted object) standing on the ground plane. We first compute the coordinates for the two vertices of the polygon on the ground plane, which form the bottom edge. Note that these two vertices on the ground may be assigned different depth values, which means the foreground polygon does not necessarily be parallel to the view plane. Assuming that the foreground polygon stands vertically to the ground plane, the coordinates of the remaining vertices are also computed automatically. As shown in Fig. 8, a static foreground object in the video sequence is modeled as a polygon standing on a ground plane.

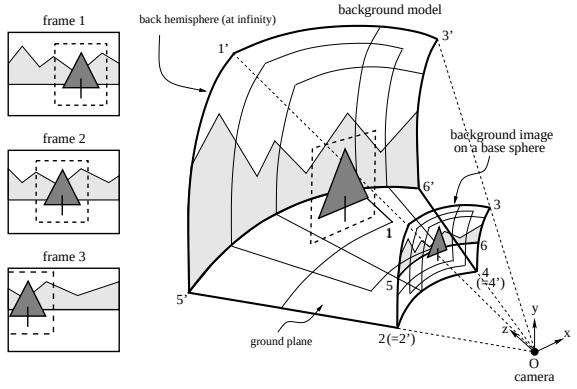


Figure 8: Static foreground model for a video sequence

The foreground image for each object corresponds to the region inside the foreground polygon from the background image. Since the spherical background image is a non-linear texture map, we convert it into a linear map to correct the foreground image. To do this, we first set a bounding rectangle of the foreground polygon on a plane containing it and then project onto this rectangle the portion of the background image which is visible from the camera through the rectangle. The resulting image on the rectangle provides a correct linear map to be used as a foreground image of the object. The exact portion of the foreground object is determined by assigning appropriate alpha values.

As proposed in [9], a foreground object can have a hierarchical structure in a more complex environment. Also, if a foreground object has a curved structure at the lower boundary on the ground plane, it is approximated by a group of piece-wise planar models. This type of model is especially useful for an object that spans a wide range in a horizontal direction. When there are foreground objects occluding others in the image, the occluded portions of the foreground objects should be restored to generate an image from a new viewpoint. Thus, multiple foreground images are required

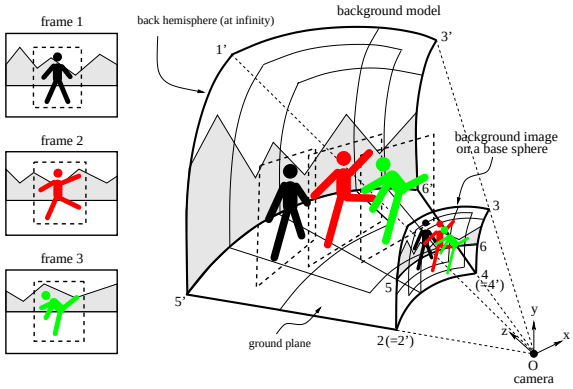


Figure 9: Dynamic foreground models for a video sequence

in this case so that each occluded object can be associated with its corresponding foreground image.

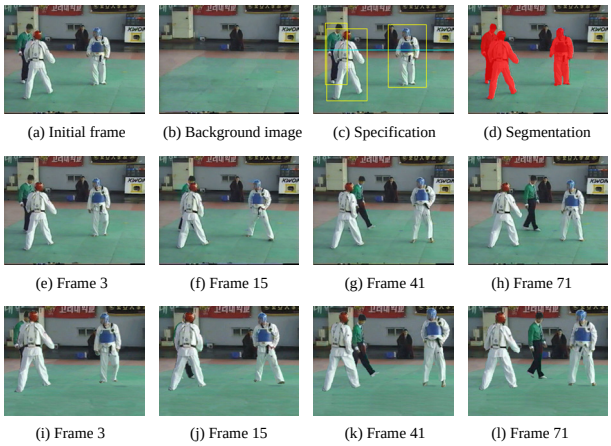
4.2.2 Dynamic foreground model

The dynamic foreground objects are also modeled as polygons and attached to the constructed background model. Since they do not appear in the background image and change their shapes and locations from frame to frame, an independent foreground model should be constructed at each frame. The foreground polygons in each frame are first mapped onto the the base object using the transformations obtained by the registration process. Then their vertex coordinates in the scene model are computed similarly as given in Section 4.2.1. After all the frames are processed, we obtain a sequence of foreground polygons which have independent shapes and locations for each frame. During the rendering process for an output video production, all the dynamic foreground objects are played back in the scene, that is, they are placed on the scene model and rendered one frame at a time, as shown in Fig. 9.

5. EXPERIMENTAL RESULTS

Fig. 10 shows the result that our TIV scheme produced with sample video. The input video contains dynamic motion of foreground objects with a fixed camera. Fig. 10a through Fig. 10d show the initial frame, the background image, the specification for the vanishing circle and the foreground polygons, and the segmented results of the foreground objects. The background model is constructed by specifying the vanishing circle in the background image. The boundaries of foreground objects are extracted for each frame as described in Section 3. Once the scene model is constructed, the user can interactively navigate the scene by controlling the camera position and orientation. Fig. 10e through Fig. 10h show samples from the original video sequence and Fig. 10i through Fig. 10l show the corresponding walk-through images seen from another viewpoint. Note the 3D effects are achieved from camera navigation due to the depth information assigned to foreground objects and the difference in viewpoints.

Fig. 11 illustrates another result of TIV with a sample video containing a camera motion. Fig. 11a through Fig. 11d show some of the initial frames. The background image



(a) Initial frame (b) Background image (c) Specification (d) Segmentation

(e) Frame 3 (f) Frame 15 (g) Frame 41 (h) Frame 71

(i) Frame 3 (j) Frame 15 (k) Frame 41 (l) Frame 71

Figure 10: Tour into the video with a fixed camera

(a) Frame 0 (b) Frame 47 (c) Frame 75 (d) Frame 103

(e) Frame 0 (f) Frame 47 (g) Frame 75 (h) Frame 103

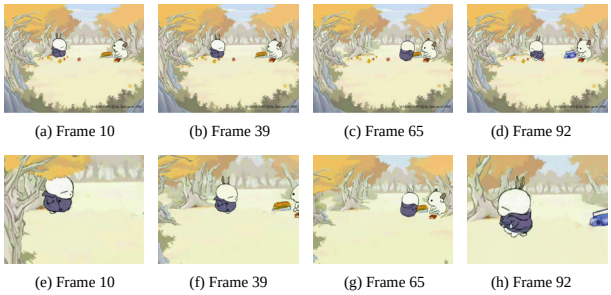
(i) Frame 0 (j) Frame 47 (k) Frame 75 (l) Frame 103

(m) Frame 0 (n) Frame 47 (o) Frame 75 (p) Frame 103

Figure 11: Tour into the video with a moving camera

is first generated after frame-by-frame image registration and automatic background detection, and the scene model is then constructed given the background image. Fig. 11e through Fig. 11h shows walk-through images seen from another viewpoint at each corresponding frame. Fig. 11i through Fig. 11l show an example synthetic video sequence obtained by inserting virtual dynamic object, and Fig. 11m through Fig. 11p show another synthetic video sequence after inserting one more virtual dynamic object imported from other video source.

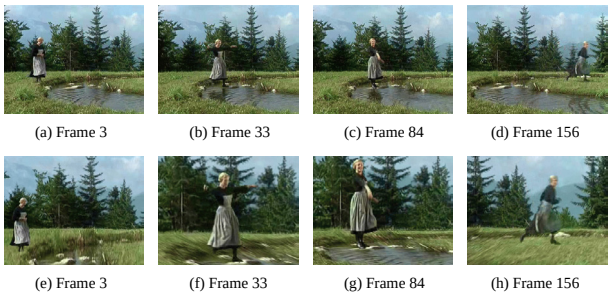
The input video in Fig. 12 is a cartoon animation. Cartoon animation can be thought of as the most suitable input for TIV since it usually consists of a set of conspicuous 2D dynamic objects with a relatively static background. Thus, we can achieve both efficiency and robustness in going through each of the three processes including the background detection, scene model construction, and boundary segmentation. In Fig. 13, TIV is applied to a movie film. This is an example where the background is also dynamic (because of the flow of water). To model the dynamic background, we use a video texture rather than a static texture, that is, a minimum number of frames are extracted and used to cover the periodic movement of the water. Thus, the output video sequence can have all three types of movements simultaneously, including that of the foreground, the background, and the camera.



(a) Frame 10 (b) Frame 39 (c) Frame 65 (d) Frame 92

(e) Frame 10 (f) Frame 39 (g) Frame 65 (h) Frame 92

Figure 12: Tour into the cartoon animation



(a) Frame 3 (b) Frame 33 (c) Frame 84 (d) Frame 156

(e) Frame 3 (f) Frame 33 (g) Frame 84 (h) Frame 156

Figure 13: Tour into the movie

The rendering speed is dependent on the number of foreground objects in the image and the image size. The entire scheme is implemented in C++ with OpenGL library on Intel Pentium^R PC (PIII 800 MHz processor and 512 MB memory) equipped with nVIDIA GeForce2GTS^R graphics processor. On average, the output sequence of images with 640×480 pixels is generated at an interactive rate (over 100 frames/sec).

6. CONCLUSIONS AND FUTURE WORK

In this paper, a novel scheme has been introduced that facilitates production of walk-through images from video sequences of dynamic scene. The scheme generalizes the idea of TIP and incorporates a new modeling scheme based on a vanishing circle. Assuming that the input video contains no strong parallax effects, an automatic background detection algorithm was employed together with a 4-parameter motion model covering camera rotation and zoom. For efficient extraction of the dynamic foreground objects in the scene, we exploited the foreground information extracted from the background detection process, which may be further corrected by frame-by-frame iterative boundary tracking based on active contour. The scheme also facilitates an augmented video production that allows dynamic foreign objects to be imported into the scene.

In practical point of view, the goal of our scheme is to let users experience the feel of navigating into a given video se-

quence and also create new synthetic video by compositing virtual objects with the original scene. While the scheme employs various automatic techniques to minimize most of the tedious jobs required for handling an excessive number of frames in the sequence, it is also designed to give users high interactivity so that diverse results can be obtained with a single input video sequence reflecting their own interpretation and imagination. The proposed scheme covers various types of video films of dynamic scenes such as sports coverage, cartoon animation, and movie films, in which objects are changing their shapes and locations continuously in a relatively static background.

There are several aspects for further improvements which may lead to the development of a more powerful scheme. To handle the input video with significant parallax effects, we need to extend our motion model to incorporate the camera displacement. The resulting mosaic representation includes the intensity image plus a corresponding depth map for a scene [10]. This type of representation is also useful for constructing more sophisticated foreground models [18]. The background image in the scene model can be rendered alive by employing video texture [20], that is, a texture map consisting of a sequence of time-coherent images. This is especially useful for describing a scene with a dynamic background, restoring shadows, and creating an infinite stream of images with either random play or video loops. Finally, when a part of the scene is occluded by a foreground object, the occluded portion (or 'hole') could be restored by employing automatic hole-filling techniques such as texture synthesis and image inpainting [3, 6]. The texture synthesis techniques can also be used to extrapolate the information outside the background region covered by the footage, which may provide more immersive effects during camera navigation [6].

7. REFERENCES

[1] R. T. Azuma. A Survey of Augmented Reality. *Teleoperators and Virtual Environments*, 6(4):355–385, August 1997.

[2] P. Beardsley, P. Torr, and A. Zisserman. 3D Model Aquisition from Extended Images Sequences. In *Proc. 4th European Conference on Computer Vision*, pages 683–695, 1996.

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image Inpainting. In *ACM SIGGRAPH 2000 Conference Proceedings*, pages 417–424, 2000.

[4] S. E. Chen. QuickTime VR - An Image-Based Approach to Virtual Environment Navigation. In *ACM SIGGRAPH 95 Conference Proceedings*, pages 29–38, August 1995.

[5] S. E. Chen and L. Williams. View Interpolation for Image Synthesis. In *ACM SIGGRAPH 93 Conference Proceedings*, pages 279–288, August 1993.

[6] A. A. Efros and T. K. Leung. Texture Synthesis by Non-parametric Sampling. In *Proceedings of IEEE International Conference on Computer Vision*, 1999.

[7] A. R. Francois and G. G. Medioni. Adaptive Color Background Modeling for Real-Time Segmentation of Video Streams. In *Proceedings of the International Conference on Imaging Science, Systems, and Technology*, pages 227–232, 1999.

[8] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The Lumigraph. In *ACM SIGGRAPH 96 Conference Proceedings*, pages 43–54, August 1996.

[9] Y. Horry, K. Anjyo, and K. Arai. Tour Into the Picture: Using a Spidery Mesh Interface to Make Animation from a Single Image. In *ACM SIGGRAPH 97 Conference Proceedings*, pages 225–232, August 1997.

[10] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Mosaic representations of video sequences and their applications. In *Proc. IEEE International Conference on Computer Vision*, pages 605–611, 1995.

[11] H. Kang, S. H. Pyo, K. Anjyo, and S. Y. Shin. Tour Into the Picture using a Vanishing Line and its Extension to Panoramic Images. In *EuroGraphics 2001 Conference Proceedings*, pages 132–141, September 2001.

[12] H. W. Kang and S. Y. Shin. Enhanced Lane: Interactive Image Segmentation by Incremental Path Map Construction. Technical Report CS-TR-2001-169, KAIST, 2001.

[13] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. In *Proceedings of the First International Conference on Computer Vision*, pages 259–268, June 1987.

[14] M. Levoy and P. Hanrahan. Light Field Rendering. In *ACM SIGGRAPH 96 Conference Proceedings*, pages 31–42, August 1996.

[15] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating Architectural Models from Images. In *EuroGraphics 99 Conference Proceedings*, pages 39–50, September 1999.

[16] L. McMillan and G. Bishop. Plenoptic Modeling: An Image-Based Rendering System. In *ACM SIGGRAPH 95 Conference Proceedings*, pages 39–46, August 1995.

[17] E. N. Mortensen and W. A. Barrett. Intelligent Scissors for Image Composition. In *ACM SIGGRAPH 95 Conference Proceedings*, pages 191–198, 1995.

[18] B. M. Oh, M. Chen, J. Dorsey, and F. Durand. Image-Based Modeling and Photo Editing. In *ACM SIGGRAPH 2001 Conference Proceedings*, pages 433–442, August 2001.

[19] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, second edition, 1992.

[20] A. Schodl, R. Szeliski, D. H. Salesin, and I. Essa. Video Textures. In *ACM SIGGRAPH 2000 Conference Proceedings*, pages 489–498, 2000.

[21] H.-Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Sixth International Conference on Computer Vision (ICCV'98)*, pages 953–958, January 1998.

[22] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE Computer Vision and Patter Recognition*, volume II, pages 246–252, 1999.

[23] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, pages 137–154, 1992.