From Images to Video: View Morphing of Three Images

Jiangjian Xiao and Mubarak Shah

Computer Vision Lab, School of Electrical Engineering and Computer Science
University of Central Florida, Orlando, Florida 32816, USA.
Email: {jxiao,shah}@cs.ucf.edu

Abstract

This paper presents a novel image-based approach to generate a video of a scene based on only three wide-baseline uncalibrated images without the use of a 3D model. After automatically recovering an accurate trifocal plane amongst the original cameras, we compute the correct dense disparity maps between each rectified image pairs by using our feature-based trinocular-stereo algorithm. Employing the barycentric warping scheme with the computed disparity, an arbitrary novel view located in the trifocal plane is generated.

Furthermore, after the self-calibration of these three cameras, we correctly augment 3D objects into the virtual environment synthesized by the tri-view morphing algorithm. We extend our approach to dynamic scene synthesis using three still images, where several rigid objects may move in any orientation or direction. After segmenting three reference frames into several layers, the novel views in the dynamic scene can be generated by applying our algorithm. The experiments are presented to illustrate that a series of photo-realistic virtual views can be generated to fly through a virtual environment covered by several static cameras.

1 Introduction

The techniques for rendering a novel view from a collection of sample images are called Image Based Rendering (IBR). They are broadly used to generate virtual cameras to visualize and navigate within virtual environments. Instead of using geometrical primitives as in traditional graphics, these approaches deeply discover the geometrical relationship between multiple sample images based on perspective principles. Ray-correspondences or point-correspondences are recovered to render a new virtual view by a back-tracing projection or forward warping.

Most IBR methods are based on plenoptic functions, which represent ray properties of a scene. Therefore, in order to record each ray from the scene, these approaches, such as light fields [6], lumigraphs [5], and visual hulls [10], require a large number of pictures for a static scene. Another limitation of existing approaches is that they require special markers in the images to calibrate cameras [6, 5, 10]. Some of them even need an explicit 3D model or geometric proxy to trace ray correspondences [6, 5].

Another class of approaches for interpolating two reference images is called image morphing and view morphing [19]. These approaches can transform a source view into a target view. There are three major methods in this area: shape blending [16, 1], feature-based morphing [8, 3], and view morphing [14, 20]. These approaches use very few images compared to the ray tracing methods discussed above. Compared with the last two kinds of approaches, view morphing and view interpolation are based on perspective geometry principles, which can provide more realistic results when blending the images of the *same* scene. However, all of these approaches can only be used to fill the gap between two views (i.e. navigation is limited to a straight line between two original views).

In order to overcome the limitation of the previous view blending methods, Avidan and Shashua proposed their work on tri-view synthesis by using trifocal tensor [2]. They generated an arbitrary novel view at any 3D view position based on three *small* baseline images, where the disparity can easily be determined by Lucas-Kanade optical-flow method. It will be almost impossible for Lucas-Kanade method to work for the *wide* baseline images. They only demonstrated the novel view synthesis between two images, where their third image was specified as same as the second one. Maur-

Figure 1: A typical tri-view morphing scenario. Top: three uncalibrated wide baseline reference images. Bottom: a series of synthesized virtual views.

izio Pilu et. al. determined edge correspondences and used interpolation to generate a new view over trinocular images [11]. Since they didn't recover the correct geometric relationship implied in the images, the disparity maps computed by using the conventional edge-scanline algorithm are not clean. Therefore, their results encountered a lot of artifacts due to some incorrect correspondences. Pollefeys and Van Gool combined 3D reconstruction and IBR to render a new view from a sequence of images [12]. With the help of the image sequences (small baseline), they estimated dense surface of the scene more accurately, which can efficiently improve the visual effect of their results. However, they only can reproduce one visualization trajectory, which was followed by the operator during the image capturing.

In this paper, we propose a novel technique to synthesize a virtual view in a 2D space. First, based on three wide-baseline uncalibrated images, we automatically recover corresponding feature points between each pair of images and determine the epipolar geometry for each pair. Second, we refine these correspondences and extract the trifocal plane by trifocal tensor computation. Third, for each rectified image pairs we compute the disparity maps using our feature-based trinocular-stereo algorithm. As a result, we generate an arbitrary novel view located in the trifocal plane and easily fly through the scene over a 2D space.

Since our morphing procedure can maintain geometrical correctness of synthesized novel views, we even can correctly augment 3D objects on the blended images after the self-calibration of these three original cameras, which can provide virtual camera model to render the augmented 3D object. Moreover, we extend our method to dynamic scene synthesis using three still images, where several rigid objects may move in any orientation or direction. After segmenting three reference frames into several layers, the novel views in the dynamic scene can be generated by applying our algorithm. These two applications show that the interaction with a virtual environment constructed by IBR is achievable under certain constraint.

2 Algorithm Overview

The input to tri-view morphing is three wide-baseline uncalibrated reference images as shown in Fig. 1. A series of novel virtual views (the bottom of Fig. 1) from any arbitrary position in the trifocal plane can be synthesized to navigate through the scene based on the three original images without any knowledge of the scene. Our tri-view morphing algorithm is implemented using the following steps:

First, using a two-stage wide baseline matching algorithm method using edge corners [21], a

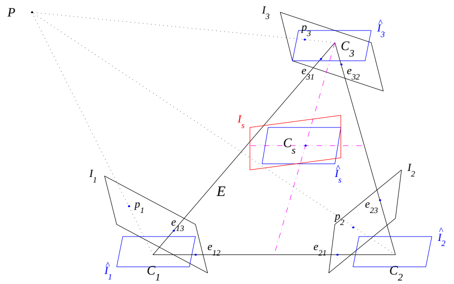

Figure 2: Tri-view morphing procedure. After automatically determining a focal plane E, which is constructed by three camera centers C_1, C_2, and C_3, three original images I_1, I_2, and I_3 are warped into parallel views $\hat{I}_1$, $\hat{I}_2$, and $\hat{I}_3$. The morphing image, $\hat{I}_s$, is blended by using the rectified images with correct disparity maps. The final image I_s at C_s is postwarped from $\hat{I}_s$.

number of corresponding points are automatically recovered to compute the fundamental matrix and epipolar geometry for each pair of reference images.

Second, a unique trifocal plane E is determined using their epipoles e_{ij} ($i, j \in \{1, 2, 3\}$ and $i \neq j$) as shown in Fig. 2. Then, the three original images I_1, I_2 and I_3 are warped into a plane parallel to the trifocal plane to obtain rectified images $\hat{I}_1$, $\hat{I}_2$, and $\hat{I}_3$.

Third, using our feature-based trinocular-stereo algorithm, the correct disparity maps between each pair of rectified images are computed.

Finally, a tri-view blending function is determined according to the viewpoint position. Following the perspective geometrical principles, the morphing image, $\hat{I}_s$, is obtained by combining the blending function with the disparity maps. Then, a 5-point postwarping scheme is used to project the morphing image to a proper final position [20].

3 Determining Corresponding Points and the Trifocal Plane

3.1 Determining Corresponding Points

The popular technique for determining correspondences over *small* baseline images is the well-known STK tracker [15], which uses an affine model to effectively compensate the motion over video frames. However, it fails to track or find correspondences if a large rotation or scaling (usually present in *wide* baseline images) between two frames is introduced. Pritchett and Zisserman have also presented an approach to estimate reliable point correspondences based on local planar homographies [13]. However, it is very difficult to determine these homographies by parallelogram structures or using motion pyramids in most cases.

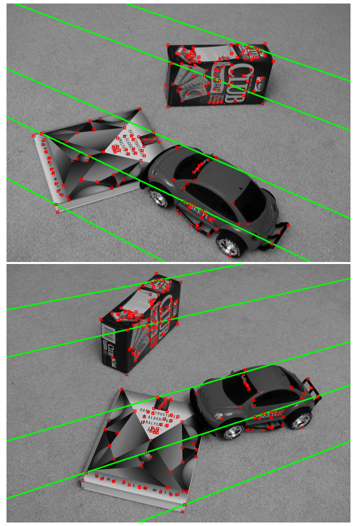

Figure 3: 182 corresponding points between two car images (Fig. 1) were found. The green lines are epipolar lines.

In this paper, we use our wide baseline matching method to determine corresponding points between each pair of images [21]. By decomposing the affine model into rotation matrix $R(\alpha)$, scaling matrix $S(\kappa)$, and stretch-shearing matrix E, the minimal image residue between two corner windows can be computed by a two-stage algorithm.

For each pair of images, first, we determine a large number of corners by edge-corner detector in two images respectively. The edge-corner detector can efficiently detect the corners located at the intersection of multiple edges. Then, a set of reliable corresponding points are found using the two-stage matching algorithm [21]. Fig. 3 shows the corresponding points and several epipolar lines obtained by our algorithm for a pair of images. The detailed implementation of wide baseline matching algorithm can be found in [21].

3.2 Determining the Trifocal Plane

In order to get geometrically correct morphing (Fig. 2), we first need to determine a trifocal plane E and warp the original images into parallel views. The focal plane is defined by the three camera centers. The epipole $e_{ij} = P_i C_j$, where P_i is the projection matrix of camera i, and C_j is the optical center of camera j. After computing the corresponding corners m'_1 and m'_2 between image I_1 and I_2, and corners m''_1 and m''_3 between image I_2 and I_3, we merge these two groups of corresponding points into one group and obtain new correspondences m_1, m_2, m_3, where $m_1 \in (m'_1 \cap m''_1)$.

Using Hartley and Zisserman's robust method, the outliers are eliminated and the trifocal tensor $T = [\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]$ is determined [7]. Then, the fundamental matrices F_{12}, F_{23}, and F_{31} can be extracted from the tensor. As a result, we can compute the trifocal plane by using the cross products of epipoles. For each camera, the trifocal plane normal is different. In camera C_1, the plane normal $N_{E1} = e_{12} \times e_{13}$; in camera C_2, $N_{E2} = e_{23} \times e_{21}$; and in camera C_3, $N_{E3} = e_{31} \times e_{32}$.

After the trifocal plane is determined, the three original images are warped into parallel views using the prewarping algorithm and employing the computed N_{E1}, N_{E2}, and N_{E3}. Also, the epipoles are projected into infinity. As a result of this warping, all epipolar lines in the three rectified images are pairwisely parallel. Next, for each pair of rectified images, corresponding epipolar lines are rotated to make them parallel to scanline directions.

4 Trinocular-stereo

In this section, we propose a trinocular-stereo algorithm to compute the disparity map between each rectified image pair, which is based on a pixel-to-pixel dynamic scanline algorithm [4, 17]. In our algorithm, the dissimilarity function uses three image intensity differences (SAD: Sum of Absolute Difference) of corresponding pixels in three rectified pairs of images (Eq. 1).

Fig. 4 shows that the rectified images can be rotated to make scanlines parallel to different epipolar lines. For example, image $\hat{I}_{12}$ is obtained when the x axis of $\hat{I}_1$ is rotated to make it parallel to e_{12}, and $\hat{I}_{13}$ is obtained when the x axis of $\hat{I}_1$ is rotated to make it parallel to e_{13}. Images $\hat{I}_{ij}$ and $\hat{I}_{ji}$ are called

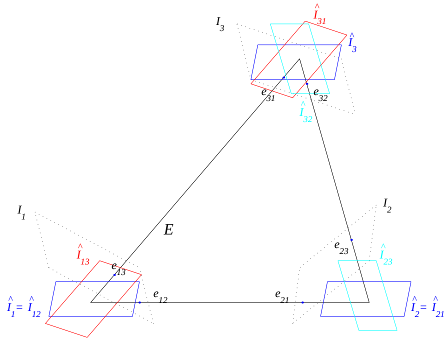

Figure 4: The x axis of rectified image $\hat{I}_i$ can be rotated to make it parallel to different epipolar directions. $\hat{I}_{12}$ and $\hat{I}_{21}$, $\hat{I}_{23}$ and $\hat{I}_{32}$, $\hat{I}_{31}$ and $\hat{I}_{13}$ are three corresponding rectified pairs.

a corresponding rectified pair, since the correspondences of these two images are always located on the same scanline.

Consider the corresponding scanlines L and R in $\hat{I}_{12}$ and $\hat{I}_{21}$, which start from L_s and R_s, and end at L_e and R_e respectively. If the two pixels x_{12} and x_{21} match, the corresponding pixel, x_3, in original image I_3 is given as the intersection of the two epipolar lines:

$$\mathrm{x}_3 = (F_{31}\mathrm{x}_1) \times (F_{32}\mathrm{x}_2),$$

where x_1 is the coordinates of the pixel x_{12} in original image I_1, and x_2 is the coordinates of the pixel x_{21} in original image I_2. Let x_{13} be the projection of x_1 on $\hat{I}_{13}$, x_{23} be the projection of x_2 on $\hat{I}_{23}$, and x_{31} and x_{13} be the projections of x_3 on $\hat{I}_{31}$ and $\hat{I}_{13}$ respectively.

The dissimilarity function $d(x_{12}, x_{21})$ is computed by using three SADs over 3×3 window N.

$$\begin{aligned} d(x_{12}, x_{21}) &= \sum_N |\hat{I}_{12}(x_{12}) - \hat{I}_{21}(x_{21})| \\ &+ \sum_N |\hat{I}_{23}(x_{23}) - \hat{I}_{32}(x_{32})| \\ &+ \sum_N |\hat{I}_{31}(x_{31}) - \hat{I}_{13}(x_{13})| \quad (1) \end{aligned}$$

Next, we use a pixel-to-pixel dynamic-scanline algorithm which uses an inter-scanline penalty to compute a dense disparity map for each corresponding rectified pair. In order to obtain high quality disparity for wide baseline stereo, a small number

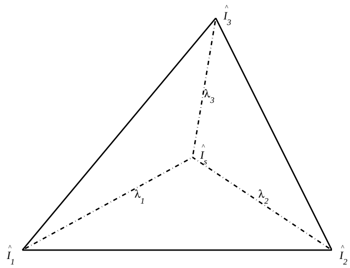

Figure 5: Blending by barycentric coefficients λ_1, λ_2, and λ_3. $\hat{I}_1$, $\hat{I}_2$ and $\hat{I}_3$ are rectified images, $\hat{I}_s$ is the desired image morphed using these three images.

of additional features are extracted for a user interface. Then, the dense disparities for wide baseline rectified pairs can be correctly computed.

5 View Blending Function

The view blending function determines the contribution of each reference image to the morphed images. In traditional image morphing, the blending function works on the original images such as I_1 and I_2. Usually, it cannot maintain geometric properties as strictly as view morphing, which interpolates the rectified images, $\hat{I}_1$ and $\hat{I}_2$, according to perspective geometry principles [14]. Following this direction, we first show that any linear combination of three parallel views satisfies the perspective geometry property.

Suppose that the trifocal plane is located at $Z = 0$, and the camera center $C_i = [C_{ix}\ C_{iy}\ 0]^T$. The projection matrices for the parallel view C_i can be represented as:

$$\Pi_i = \begin{bmatrix} f_i & 0 & 0 & -f_i C_{ix} \\ 0 & f_i & 0 & -f_i C_{iy} \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

where f_i is focal length of C_i. The new point $\hat{p}_s$ can be obtained by linearly blending the three corresponding points $\hat{p}_1, \hat{p}_2, \hat{p}_3$, which are respectively the projections of a 3D point $P = [X\ Y\ Z\ 1]^T$ on images $\hat{I}_1$, $\hat{I}_2$, and $\hat{I}_3$.

$$\begin{aligned} \hat{p}_s &= \lambda_1 \hat{p}_1 + \lambda_2 \hat{p}_2 + \lambda_3 \hat{p}_3 \\ &= \frac{1}{Z}(\lambda_1 \Pi_1 + \lambda_2 \Pi_2 + \lambda_3 \Pi_3) P \\ &= \frac{1}{Z} \Pi_s P, \end{aligned}$$

where Π_s is the linear interpolation of Π_1, Π_2, and Π_3, its focal length $f_s = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3$, and the camera center $C_s = \lambda_1 C_1 + \lambda_2 C_2 + \lambda_3 C_3$. Therefore, any linear combination of three parallel views satisfies the perspective geometry property when $\sum_{i=1}^{3} \lambda_i = 1$.

Let W_{ij} be the warping function between images $\hat{I}_i$ and $\hat{I}_j$, which specifies the correspondences between $\hat{I}_i$ and $\hat{I}_j$.

$$W_{ij} = \hat{p}_i + d_{ij},$$

where d_{ij} can be obtained using the dense disparity map in Section 4 when $i \neq j$, and $d_{ij} = 0$ when $i = j$. Next, we create a new warping function B_i to warp each of the images $\hat{I}_i$ to $\hat{I}_s$ and blend them together.

$$B_i = \Sigma_{i=1}^{3} \lambda_i W_{ij}, \tag{2}$$
$$\hat{I}_s = \Sigma_{i=1}^{3} \lambda_i B_i(\hat{I}_i), \tag{3}$$

where $\lambda_i B_i(\hat{I}_i)$ represents the warped image $\hat{I}_i$ into $\hat{I}_s$ with opacity value λ_i. It is easily verified that Eq. 2 and 3 have the linear blending property $\hat{p}_s = \sum_{i=1}^{3} \lambda_i \hat{p}_i$.

Based on this result, we present a scheme to blend three rectified images, which is based on image $\hat{I}_s$'s barycentric coordinates $\lambda = (\lambda_1, \lambda_2, \lambda_3)$, subject to $\lambda_i \geq 0$ and $\sum_{i=1}^{3} \lambda_i = 1$ (Fig. 5). Along each edge of the triangle, one of the three coordinates (corresponding to the opposite vertex) is zero. This property is very useful for continuous interpolation across the edges of a triangulation in multiple view morphing, which can be divided into multiple triangles by a simple triangular tessellation. After blending the rectified image pairs, we reproject the morphing images to the final position by 5-point postwarping algorithm and obtain the final novel views as shown in Fig. 1.

6 Augmenting 3D Objects

In order to augment 3D objects in the virtual morphing environment, first we self-calibrate three cameras, and obtain the intrinsic and external parameters of these cameras. Then, for any novel view, the new camera matrix can be interpolated using blending coefficient λ. Based on this camera model,

Figure 6: Augment object in tri-view morphing. Top: three original images with the augmented object ("Bunny" model from Stanford University). The 3D model is accurately augmented on the top of the book. Bottom: a series of synthesized virtual views with the augmented object can maintain geometric correctness in the morphing procedure. The shadow of model is cast on the book by assuming a light source.

the 3D objects can be correctly rendered and augmented into the scene during the scene navigation, which is a prerequisite for the interaction with the synthesized environment.

From the recovered trifocal tensor T, the projective camera matrices Π_{p_1}, Π_{p_2}, and Π_{p_3}, can be extracted as follows:

$$\begin{aligned}\Pi_{p_1} &= [\mathrm{I} \mid 0],\\ \Pi_{p_2} &= [[\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]e_{31} \mid e_{21}],\\ \Pi_{p_3} &= [(e_{31}e_{31}^T - \mathrm{I})[\mathrm{T}_1^T, \mathrm{T}_2^T, \mathrm{T}_3^T]e_{21} \mid e_{31}].\end{aligned}$$

Since we used one digital camera to capture these images, we can safely assume that only the focal length in the camera's intrinsic parameters was changed during taking the pictures. Then, we employ the self-calibration method [7] to recover a 3D homography H and obtain a metric 3D construction of the sparse corresponding points, where the metric camera matrix $\Pi_{m_i} = \Pi_{p_i}H$. The metric camera matrix Π_{m_i} also can be represented as:

$$\begin{aligned}\Pi_{m_i} &= K_i[R_i \mid R_i t_i]\\ &= K_i[R_z(\phi_i)R_y(\theta_i)R_x(\psi_i) \mid R_i t_i],\end{aligned}$$

where K_i is intrinsic parameter of camera Π_{m_i}, R_i and t_i are the rotation and translation components of the camera's external parameters. Each rotation matrix R_i can be decomposed into three components $R_z(\phi_i)$, $R_y(\theta_i)$, and $R_x(\psi_i)$.

Therefore, the parameters of new virtual camera can be easily linearly interpolated by multiplying the coefficient $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ respectively.

$$\begin{aligned}K_s &= \lambda_1 K_1 + \lambda_2 K_2 + \lambda_3 K_3,\\ t_s &= \lambda_1 t_1 + \lambda_2 t_2 + \lambda_3 t_3,\\ \alpha_s &= \lambda_1 \alpha_1 + \lambda_2 \alpha_2 + \lambda_3 \alpha_3,\end{aligned}$$

where α can be substituted by angles ϕ, θ, and ψ. And the new camera matrix is given as

$$\Pi_{m_s} = K_s[R_z(\phi_s)R_y(\theta_s)R_x(\psi_s)|R_s t_s].$$

This new camera model is guaranteed to move on the trifocal plane due to the linear interpolation of the translation components of the cameras. Based on this new camera matrix Π_{m_s}, we use OpenGL to render the 3D object and augment the object into the view generated from the same viewpoint. In Fig. 6, we augment a 3D "Bunny" model on the top of the book. In order to generate the shadow, we simply recover a plane of the book using sparse 3D points reconstructed from self-calibration method. Then, after assuming an approximate light source, the shadow is generated and projected on this plane and "Bunny" model by multiple pass rendering.

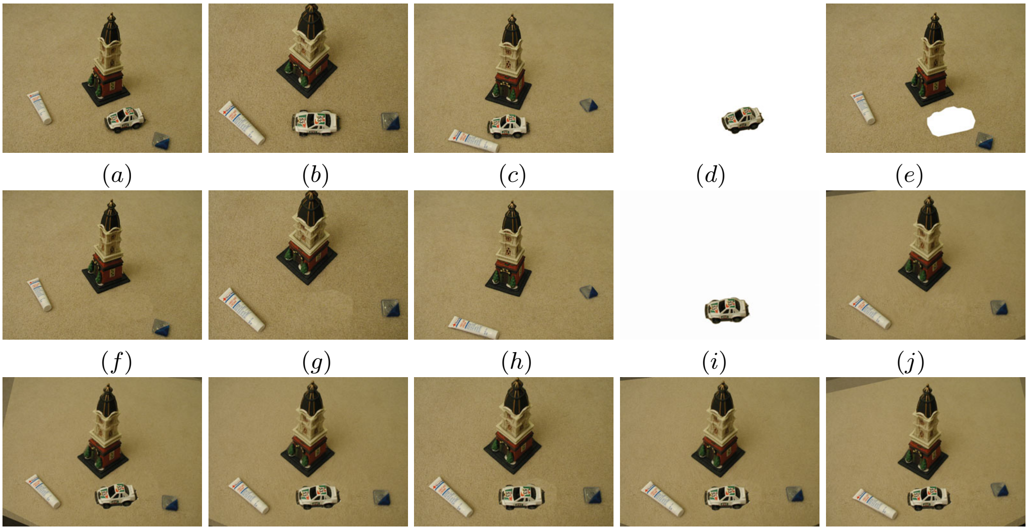

Figure 7: A typical tri-view morphing scenario. (a), (b) and (c) are three uncalibrated wide-baseline reference images. (d) and (e) are segmentation results of the first frame. (f-h) are the background images corresponding to the three original views after filling in the gaps occupied by the car. (i) and (j) are morphing results of the car and background layers respectively. The last row shows the virtual views obtained after compositing the car and background layers.

7 Dynamic Tri-view Morphing

Our dynamic tri-view morphing is focused on a dynamic scene containing only rigid moving objects, which may rotate or translate. Xiao at el. [20] have shown that novel views can be generated for such dynamic scenes by using a separate fundamental matrix for each rigid object (layer) based on the relative motion between two views. In this paper, we extend this idea to dynamic tri-view morphing. Our algorithm consists of three main steps: segmentation, tri-view morphing for each layer and multiple layer blending.

Fig. 7 shows a dynamic scene with a moving car. Three reference images were taken at different times and at different positions and orientations while the car was moving. We put an eye-drops tube in the scene to introduce a transparent material. Since the moving object (car) and the background (the rest of the image) have different epipolar geometries, we manually segment the scene into two layers: background (Fig. 7.e) and car (Fig. 7.d). Next, we compute a homography to fill in the hole in Fig. 7.e using the pixels from the other two images, in order to obtain the background image (Fig. 7.f) for the first frame. Then, the background (Fig. 7.i) and car (Fig. 7.j) are blended separately by the tri-view morphing algorithm. Finally, these two images are composited together and the final results are obtained as shown in the last row of Fig. 7. As a result, the car is moving along its own trajectory during the scene navigation.

Note: The video clips of the above experiments and more other results are available on our web site: http://www.cs.ucf.edu/∼vision/projects/triview/.

8 Conclusion and Discussion

In this paper, we proposed a image-based approach to synthesize a video of a scene based on only three wide-baseline uncalibrated images without the use of a 3D model. Employing the trifocal plane automatically extracted from the trifocal tensor, our morphing procedure can maintain geometrical correctness of synthesized novel views, which also can be correctly augmented with 3D objects. An arbitrary novel view located in the trifocal plane is blended to obtain a photo-realistic image, which can be used to interactively navigate through the virtual environment. Moreover, the augmented tri-view morphing and dynamic tri-view morphing provide another new direction to obtain an interaction with the virtual environment built on this image-based approach.

One limitation of our work is that our navigation currently is restricted in a 2D space constructed by three images. Even though an arbitrary novel view at any position can be synthesized from two or three images by the extrapolation of three viewpoints, it usually cannot preserve the good-quality rendering results due to the missing texture from the new viewpoint. Hence, the novel view along the baseline of two images or along the tri-focal plane of three images can avoid this problem and provide the higher quality results. In order to navigate a 3D volume and obtain a realistic visual effect, four images captured by the non-coplanar cameras are the minimum requirements. In the future, we will investigate this area and extend our method to 3D volume navigation.

9 Acknowledgement

The research reported in this paper was partially supported by STRICOM under Contract No. N61339-02-C-0082.

References

[1] M. Alexa, D. Cohen-Or, and D. Levin. "As-Rigid-as-Possible Shape Interpolation", *SIGGRAPH 2000*, 157–164, 2000.

[2] S. Avidan, and A. Shashua. "Novel View Synthesis by Cascading Trilinear Tensors", *IEEE Transactions on Visualization and Computer Graphics*, 293–306, 1998.

[3] T. Beier, and S. Neely. "Feature-Based Image Matamorphosis", *SIGGRAPH 1992*, 1992.

[4] S. Birchfield, and C. Tomasi. "Depth Discontinuities by Pixel-to-Pixel Stereo", *IJCV*, (35), 269–293, 1999.

[5] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. "Unstructured Lumigraph Rendering", *SIGGRAPH 2001*, 425–432, 2001.

[6] W. Chen, J. Bouguet, M. Chu, and R. Grzeszczuk. "Light Field Mapping: Efficient Representation and Hardware Rendering of Surface Light Fields", *SIGGRAPH 2002*, 447–456, 2002.

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

[8] S. Lee, G. Wolberg, and S. Shin. "Polymorph: Morphing Among Mutltiple Images", *IEEE Computer Graphics and Applications*, 1998.

[9] R. Manning, and C. Dyer. "Interpolating View and Scene Motion by Dynamic View Morphing", *CVPR*, 388–394, 1999.

[10] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan. "Image-Based 3D Photography using Opacity Hulls", *SIGGRAPH 2002*, 427–437, 2002.

[11] S. Pollard, M. Pilu, S. Hayes, and A. Lorusso. "View Synthesis by Edge Matching and Transfer", *IEEE Workhop on Applications of Computer Vision*, 1998.

[12] M. Pollefeys, and L. Van Gool. "Visual Modeling: from Images to Images", *The Journal of Visualization and Computer Animation*, 13, 199-209, 2002.

[13] P. Pritchett, and A. Zisserman. "Wide Baseline Stereo Matching", *ICCV*, 754-760, 1998.

[14] S. Seitz, and C. Dyer. "View Morphing", *SIGGRAPH 1996*, 21-30, 1996.

[15] J. Shi, and C. Tomasi. "Good Features to Track", *CVPR*, 1994.

[16] A. Tal, and G. Elber. "Image Morphing with Feature Preserving Texture", *EUROGRAPHICS'99*, 339-348, 1999.

[17] G. Van Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool. "A Hierarchical Symmetric Stereo Algorithm Using Dynamic Programming", *IJCV*, 47, 275-285, 2002.

[18] S. Vedula, S. Baker, T. Kanade. "Spatio-Temporal View Interpolation", *Eurographics Workshop on Rendering*, 2002.

[19] G. Wolberg. "Image Morphing: a Survey", *The Visual Computer*, 360-372, 1998.

[20] J. Xiao, C. Rao, and M. Shah. "View Interpolation for Dynamic Scenes", *EUROGRAPHICS'02*, 153-162, 2002.

[21] J. Xiao, and M. Shah. "Two-Frame Wide Baseline Matching", *ICCV'03*, 2003.

[22] Z. Zhang. "Determining the Epipolar Geometry and its Uncertainty: A Review", *IJCV*, 27, 161-195, 1998.