Video Augmentation by Image-based Rendering

Seo, Yongduek, Kim, Seungjin, Hong, Ki Sang
MVIP Lab, Dept. EE, POSTECH

Abstract

This paper provides a method for video augmentation using image interpolation. In computer graphics or augmented reality, 3D information of a model object is necessary to generate 2D views of the model, which are then inserted into or overlayed on environmental views or real video frames. However, we do not require any three dimensional model but images of the model object at some locations to render views according to the motion of video camera which is calculated by an SFM algorithm using point matches under weak perspective (scaled-orthographic) projection model. Thus, a linear view interpolation algorithm is applied rather than a 3D ray-tracing method to get a view of the model at different viewpoints from model views. In order to get novel views in a way that agrees with the camera motion, the camera coordinate system is embedded into model coordinate system at initialization time on the basis of 3D information recovered from video images and model views, respectively. During the sequence, motion parameters from video frames are used to compute interpolation parameters, and rendered model views are overlayed on corresponding video frames. Experimental results for real video frames and model views are given. Finally, discussion on the limitations of the method and subjects for future research are provided.

1 Introduction

In this paper we are to render novel views of an object given reference images of the object and combine them with a video of real environment so that the video images look as if the camera has taken the model object in reality. Figure 1 shows an example of model views, video images and final results we intend to obtain. We address two subjects to have the goal: one is rendering views from images of a model object and the other is embedding the generated views into video images. Usually methods to render a real object from arbitrary viewpoints use a 3D model of the object, which could be a CAD model being used in computer graphics or a 3D model reconstructed by stereo vision. When using 3D model, it is easy to get a view since determination of rotation and translation parameters of camera-centered coordinate system with respect to model-coordinate system gives the view. However, it is well known that to make a 3D model by reconstruction is not easy[9, 11, 1], and we do not want to do it to get views of a *real* object.

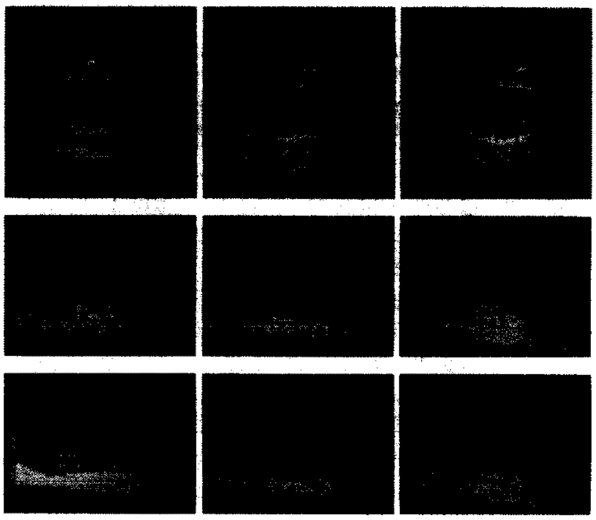


Figure 1: The first row is model views, the second video images and the third augmented video images

Researches to synthesize novel views using view-interpolation or those using algebraic functions like trilinear tensor or fundamental tensor are all based on model views and deal with how to avoid the 3D reconstruction problem. Seitz and Dyer [12, 13] proposed a method to synthesize physically-valid views by image interpolation, being linear, under orthographic camera model. They dealt with the problem of which views may be inferred from a set of model images and showed that the range of predictable views can be described for a continuous range of viewpoints by a set of discrete images under the constraint called *monotonicity*. They proposed a scan-line interpolation algorithm as a rendering method. Werner *et. al.* [17] represented an object by a set of 2D views and constructed a novel view as a combination of the model views. They showed that the combination can be linear and suggested a way to determine the visibility of constructed points.

A method using an algebraic function of views can be found in the work of Ullman and Basri [16], which showed that any view of a model object can be expressed as a linear combination of reference views. Avidan and Shashua [1] derived a tensorial operator that describes the transformation from a given tensor of three views to a novel tensor of a new configuration of three views. Being an extension of the work of [16] to perspective camera model, trilinear tensor provides a warping function to create desired virtual views from reference views, given rotation and translation with respect to the first view. Laveau and Faugeras [9] utilized the epipolar constraint for view synthesis to interpolate or extrapolate between the model images. The indirect relationship between motion parameters of virtual camera and the epipolar constraint is specified by matching points in image space.

Rendering views of a 3D model (a virtual object) and mixing them into video have been studied in the area of augmented reality and virtual reality. In [8] a lot of references to this area can be found including [2, 6, 18, 7]. Also, many augmented reality activities around the world can be found at the web site[1] of James Vallino who is an author of [8]. One problem in augmented reality is that geometric relationship among physical objects (the environment), virtual objects and the camera should be established first. That is, external and internal parameters of the video camera should be computed to embed rendered images into video frames, for which some reference points or patterns whose Euclidean geometry is known *a priori* are used[15, 5, 2, 18, 3]. In this case, matching the reference points and calibrating the camera at every frame are necessary, and a difficulty is in the fact that if we have to deal with a general video sequence we cannot find references for camera calibration. To attack this problem, Kutulakos and Vallino[8] suggested a calibration-free augmented reality system using affine object representation. Camera calibration was replaced by tracking reference points in video images and the locations of a virtual object were specified in two video images instead of establishing 3D relationship. Images of the virtual object rendered by graphics machine were overlayed on real-video frames and they looked as if the object had been taken by the video camera.

A limitation in augmented reality is that a 3D graphic model is always necessary in the sense that if we want to handle a real object such as the National Assembly building or White House, then we cannot help recovering a 3D information from the images. However, as mentioned 3D reconstruction is not an easy problem. We propose a method to overlay novel views synthesized from reference images on live video images according to the motion of the video camera. Main problems considered in this paper are:

1. *how to insert the model object into video,*
2. *how to specify parameters describing a novel view with respect to reference views according to the motion of video camera,*
3. *how to synthesize a novel view,*
4. *and how to solve self-occlusion and visibility*

In this paper two interpolation parameters determine novel views which are computed using the relationship between view vectors from video images and those from model views. Initial location of the model in video is specified in two video images using epipolar constraint. Correspondences are found using stereo matching algorithm between two images. For rendering, we use linear interpolation method proposed in [12, 13] which provides physically-valid views by image interpolation. Self-occlusion and visibility problem is not considered in this paper. However, since we compute Euclidean reconstruction for some feature points visibility can be solved as described in [9, 13]. Section 2 shows the proposed algorithm in detail and section 3 gives experimental results. Finally, concluding remarks and discussion on limitations of the method are in section 4.

2 Algorithm

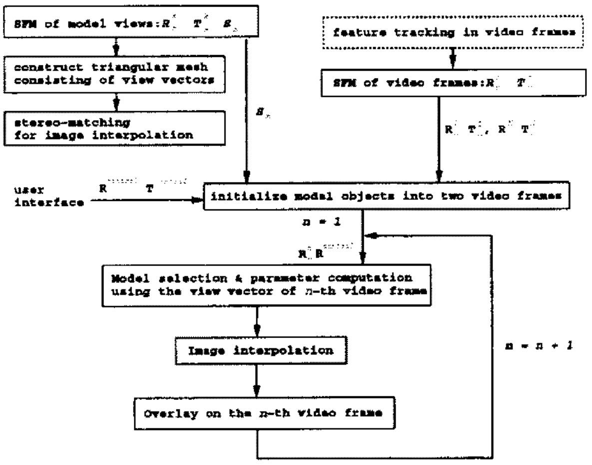


Figure 2: algorithm flow of our method

Figure 2 shows flow of proposed algorithm which can be divided into four steps. Full explanations are in the following subsections.

1. *Model construction:* Structure and motion parameters for model views are computed from point correspondences [10, 14]. View vectors of model views are represented by direction vectors in unit sphere and they are nodes of triangular meshes which are used in selecting model views. Disparity maps (image correspondences)

---

[1] http://www.cs.rochester.edu:80/u/www/u/vallino/research/AR/

between two connected model images on the meshes are computed by a stereo matching algorithm for novel view interpolation, which is explained in section 2.5. Some corner points are provided mannually in order to help stereo matching and initialization of model view.

2. *SFM for video:* Motion parameters for video images are recovered, which together with those of model images determine interpolation parameters.

Notice that even though structure recovery in the two steps are to establish 3D geometric relationship between video camera and virtual object, it does not mean full structure recovery as 3D reconstruction-projection algorithms. It is used in deterimining interpolation parameters.

3. *Initialization:* This step locates model coordinate system into video coordinate system. In other words, the locations of model views being inserted into the video images are determined. Section 2.4 explains the way to place the object in video by specifying three locations of model points in two video images.

4. *Image interpolation:* This step plays the same role as ray-tracing in computer graphics. Given motion parameters of n-th video frame, we select three reference views, interpolate them to make a new view and overlay it on the video image. In the interpolation, the precomputed stereo-matching is utilized. A method to deal with three images to render a novel view is in section 2.5.

2.1 Camera Model

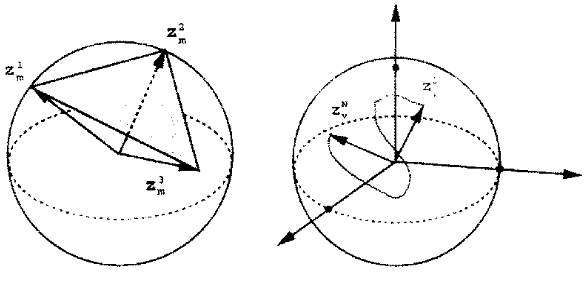
In this paper we use weak-perspective (scaled orthographic) camera model. Image location $\mathbf{p}^k = (u^k, v^k)^{\mathrm{T}}$ of a 3D point S at time k is given by the equation:

$$\mathbf{p}^k \;=\; s^k\,[I|\mathbf{0}]\,R^k\,\mathbf{S} + \mathbf{t}^k \qquad (1)$$

where R^k is a 3×3 rotation matrix, $\mathbf{t}^k$ a two dimensional translational vector and s^k a scale factor at time k. I is 2×2 identity matrix and $\mathbf{0} = (0, 0)^{\mathrm{T}}$. When $s^k = 1$ for all k, it is orthographic camera model. Notice that the third row $\mathbf{z}^{\mathrm{T}}$, a view vector, of R^k denotes the viewing direction of the camera, the optical axis.

2.2 Model Construction

When three views of an object is given, we can make views of the object at the other view points inside the triangle determined by the optical centers under the condition that there is no self-occlusion[12, 13]. That is, a triple of views is a basic unit for image-based view synthesis and we define a *triangular mesh* on the unit sphere of viewing directions. Figure 3-(a) shows an example. Each vector represents a viewing direction and corresponds to a model image. Structure and motion parameters including viewing vectors are computed using SFM algorithms: the factorization method [14] under orthographic camera model and the method of [10] under weak-perspective camera model.



(a) (b)

Figure 3: (a) Triangular mesh defined on the unit sphere of viewing vectors, (b) The locus of viewing vectors on the unit sphere: $\mathbf{z}_v^1$ corresponds to the first video image

2.3 SFM for video images

The point matches obtained by tracking in video sequence are used to compute the camera motion by the structure from motion algorithm. Notice that the computed view vectors form a locus on the unit sphere. Figure 3-(b) depicts an example of the locus.

2.4 Initialization

The location of the model object in video are determined by specifying 3D geometric relationship:

$$\mathbf{p}_m^v = s^v[I|\mathbf{0}]R^v R^{control}_{(\alpha,\beta,\gamma)}\left(\mathbf{S} + \mathbf{T}^{control}_{(X,Y,Z)}\right) + \mathbf{t}^v \qquad (2)$$

where $R^{control}_{(\alpha,\beta,\gamma)}$ and $\mathbf{T}^{control}_{(X,Y,Z)}$ denote transformation between the two coordinate systems and $\mathbf{p}_m^v$ is the image location of the model structure $\mathbf{S}$ in the v-th video image.

Initialization step corresponds to the procedure in graphics of placing an object in world coordinate system by specifying translation and rotation components. In our case two initialization methods are available: 1) locating three feature points on two video images and 2) specifying rotation and translation of model coordinate system with respect to video coordinate system. In the former case equation (2) allows us to find $R^{control}_{(\alpha,\beta,\gamma)}$ and $\mathbf{T}^{control}_{(X,Y,Z)}$ as soon as a user specifies three locations of features in two video images. We explain in section 2.4.1 and 2.4.2 how to specify the three locations. In the latter case transformation parameters are specified directly while the user looks at model views overlayed in video images.

After the transformations $R^{control}_{(\alpha,\beta,\gamma)}$ and $\mathbf{T}^{control}_{(X,Y,Z)}$ are determined, we can compute the locus of viewing vectors of video images on the unit sphere of model views. Figure 4 is an example of the unit sphere of viewing vectors after initialization. In the example, viewing vectors $\mathbf{z}_v^1, \ldots, \mathbf{z}_v^6$ are in the area of the mesh $\{\mathbf{z}_m^1, \mathbf{z}_m^2, \mathbf{z}_m^3\}$ from which novel views for this area are generated.

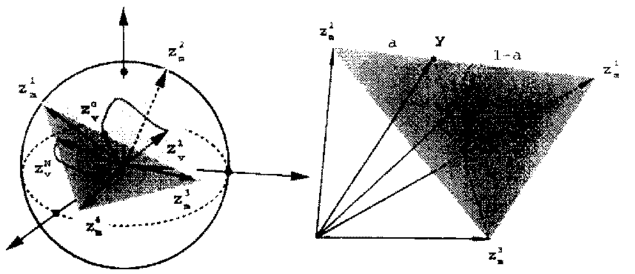


Figure 4: Left: the locus of viewing vectors from video images overlayed on the unit sphere of model views after initialization. Right: two interpolation parameters a and b are computed using viewing vectors

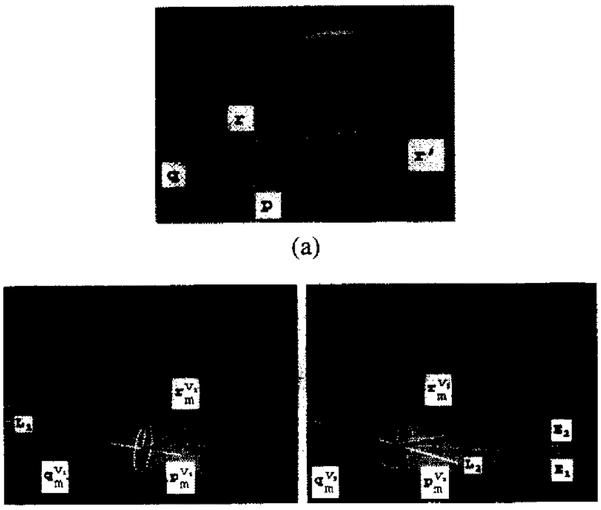
2.4.1 Initialization by specifying image locations

When we have information from model views that some feature points are on a plane or some feature points exist on a rectangular coordinate system, initialization step becomes easy. In this section we explain an initialization in the case that three feature points are on a rectangular coordinate axes. Figure 5-(a) shows three corner points for initialization and figures 5-(b) and 5-(c) will help us explain the procedure. Notice that the black strips in 5-(b) and 5-(c) are only for evaluating the results of our method. Usually they are not necessary.

1. *Draw axis-lines* $\mathbf{L}_1$ *and* $\mathbf{L}_2$ *in two video images, respectively. And specify two image locations* $\mathbf{p}_m^{\mathcal{V}_1}$ *and* $\mathbf{q}_m^{\mathcal{V}_1}$ *of two reference points* $\mathbf{S}_m^1$ *and* $\mathbf{S}_m^2$ *on* $\mathbf{L}_1$. This step computes translational parameter $\mathbf{T}_{(X,Y,Z)}^{control}$ and decides the direction of an axis of model coordinate system. Choosing a location $\mathbf{p}_m^{\mathcal{V}_1}$ in the first image $\mathcal{V}_1$ gives its epipolar line in the second image $\mathcal{V}_2$. $\mathbf{p}_m^{\mathcal{V}_2}$ is the intersection point of the two lines. As soon as $\mathbf{p}_m^{\mathcal{V}_2}$ is computed , their 3D coordinates $\mathbf{S}_{\mathcal{V}}^1$ are computed from equation (1). Then, we have $\mathbf{T}_{(X,Y,Z)}^{control} = \mathbf{S}_{\mathcal{V}}^1 - \mathbf{S}_m^1$. Notice that $\mathbf{S}_m^1$ is expressed in the model coordinate system and $\mathbf{S}_{\mathcal{V}}^1$ in the video coordinate system. In the example of figure 5, the axis-lines $\mathbf{L}_1$ and $\mathbf{L}_2$ are drawn on the basis of the contents of the images, and two reference points are selected.

2. *Compute rotation matrix* R' *and scale factor* s^v *by computing* $\mathbf{q}_m^{\mathcal{V}_2}$. $\mathbf{q}_m^{\mathcal{V}_2}$ is the intersection point of the epipolar line of $\mathbf{q}_m^{\mathcal{V}_1}$ and $\mathbf{L}_2$. The rotation R' is the one that lines up two vectors $\mathbf{S}_m^2\mathbf{S}_m^1$ and $\mathbf{S}_{\mathcal{V}}^2\mathbf{S}_{\mathcal{V}}^1$. The scale s^v is given by $s^v = \frac{\|\mathbf{S}_{\mathcal{V}}^2\mathbf{S}_{\mathcal{V}}^1\|}{\|\mathbf{S}_m^2\mathbf{S}_m^1\|}$. Notice that the model object now can just rotate about the axis $\mathbf{S}_{\mathcal{V}}^2\mathbf{S}_{\mathcal{V}}^1$. Thus, possible image locations of the third point $\mathbf{S}_m^3$ are expressed to be an ellipse in video images.

3. *Finally, choose the image location of* $\mathbf{S}_m^3$ *on the ellipse.* Then the last rotation R'' is determined, and the rotation in equation (2) is: $R_{(\alpha,\beta,\gamma)}^{control} = R''R'$.



(a)

(b) (c)

Figure 5: (a): $\mathbf{p}$, $\mathbf{q}$ and $\mathbf{r}$ are used in the initialization of section 2.4.1. Since $\mathbf{p}$, $\mathbf{q}$ and $\mathbf{r}'$ are on a plane (b) and (c): $\mathbf{L}_1$ and $\mathbf{L}_2$ are the axis-lines on which two points $\mathbf{p}_m^{\mathcal{V}_1}$ and $\mathbf{q}_m^{\mathcal{V}_1}$ are selected. On the corresponding epipolar lines $\mathbf{E}_1$ and $\mathbf{E}_2$, $\mathbf{p}_m^{\mathcal{V}_2}$ and $\mathbf{q}_m^{\mathcal{V}_2}$ is computed.

2.4.2 Initialization using plane information

In the figure 5-(a), $\mathbf{p}$, $\mathbf{q}$ and $\mathbf{r}'$ are on the bottom plane of the object. If we have matches of three coplanar points in the two video images, we can restrict the location of the model object on the plane defined by the matches using epipolar geometry and a homography computed using four coplanar points.

1. *Determine the location of* $\mathbf{p}$ *and* $\mathbf{q}$ *in* $\mathcal{V}_1$. Through the *point-plane procedure*[4] (See Appendix), their locations $\mathbf{p}_m^{\mathcal{V}_2}$ and $\mathbf{q}_m^{\mathcal{V}_2}$ are computed, and then $\mathbf{T}_{(X,Y,Z)}^{control}$, R' and s^v is determined.

2. *Find* R'' *using plane homography.* Since we have five coplanar points, three from video and two from the above step, we can compute a plane homography H, a 3 × 3 matrix which transform a plane to another in projective plane. We know that the possible locus of the third point is an ellipse $\mathbf{C}_1$ and $\mathbf{C}_2$, in $\mathcal{V}_1$ and $\mathcal{V}_2$, respectively. Two candidate locations in $\mathcal{V}_2$ are found to be the intersection of $\mathbf{C}_2$ and the transformed version of $\mathbf{C}_1$ by H. Selecting one of them determines R'' and $R_{(\alpha,\beta,\gamma)}^{control} = R''R'$.

2.5 View Generation

Making a new view consists of two steps: model selection and image interpolation. Model selection is finding the mesh that contains the viewing vector of a video image given. In the example of figure 4, the mesh $\{\mathbf{z}_m^1, \mathbf{z}_m^2, \mathbf{z}_m^3\}$ provides three reference views for rendering novel views at video images $I_v^1, \ldots, I_v^c$.

2.5.1 Model selection

Let $\{\mathbf{z}_m^1, \mathbf{z}_m^2, \mathbf{z}_m^3\}$ be a mesh among the reference meshes. Then, testing whether the mesh contains the view vector $\mathbf{z}_v$ is done by checking signs of the quantities $[\alpha_1, \alpha_2, \alpha_3]^{\mathrm{T}} = [\mathbf{z}_m^1, \mathbf{z}_m^2, \mathbf{z}_m^3]^{-1}\mathbf{z}_v$ where $[\mathbf{z}_m^1, \mathbf{z}_m^2, \mathbf{z}_m^3]$ is the 3×3 matrix whose columns are three viewing vectors. That is, if $\alpha_i > 0$ for all $i = 1, 2, 3$ then the view vector $\mathbf{z}_v$ is inside the mesh and corresponding novel view is generated from the three reference views.

2.5.2 View rendering

After model selection, new model views are generated using the three model images through view interpolation. Two interpolation parameters a and b are computed first, and then new views of the model object are synthesized. Figure 4 shows two interpolation parameters a and b which are two scalar values that satisfy the equation:

$$\mathbf{z}_v^k = \frac{(a\mathbf{z}_m^1 + (1-a)\mathbf{z}_m^2)b + (1-b)\mathbf{z}_m^3}{\|(a\mathbf{z}_m^1 + (1-a)\mathbf{z}_m^2)b + (1-b)\mathbf{z}_m^3\|} \qquad (3)$$

In order to compute a and b, $\mathbf{y} = \epsilon(\mathbf{z}_m^1 \times \mathbf{z}_m^2) \times (\mathbf{z}_v^k \times \mathbf{z}_m^3)$ is first computed, where ϵ is a sign so that $\mathbf{y} \cdot \mathbf{z}_m^i$ is positive. Then, we have

$$a = \frac{1 - \mathbf{y} \cdot \mathbf{z}_m^2}{1 - \mathbf{z}_m^1 \cdot \mathbf{z}_m^2} \quad \text{and} \quad b = \frac{1 - \mathbf{z}_v^k \cdot \mathbf{z}_m^3}{1 - \mathbf{y} \cdot \mathbf{z}_m^3}. \qquad (4)$$

Rendering a new view $\mathcal{I}_v^k$ divides into two stages:

1. Making an intermediate view $\mathcal{I}_m'$ using two model views $\mathcal{I}_m^1$ and $\mathcal{I}_m^2$. The interpolation parameter for this stage is a.
2. Rendering final view $\mathcal{I}_v^k$ using the intermediate view $\mathcal{I}_m'$ and the rest model view $\mathcal{I}_m^3$. Now, b is the interpolation parameter.

At each stage, we use the view interpolation method proposed in [12, 13] so as to generate *physically-valid* in-between views. Namely, given two reference images $\mathcal{I}^1$ and $\mathcal{I}^2$ and the parameter λ:

1. *Rectification:* two reference images are rotated and scaled so that corresponding epipolar lines of rectified images $\hat{\mathcal{I}}_1$ and $\hat{\mathcal{I}}_2$ are parallel.
2. *Interpolation:* linear interpolation of locations and intensities yields a rectified novel view $\hat{\mathcal{I}}'$:

$$\begin{aligned} \mathbf{p}' &= \lambda\mathbf{p}^1 + (1-\lambda)\mathbf{p}^2 & (5) \\ \hat{\mathcal{I}}'(\mathbf{p}') &= \lambda\,\hat{\mathcal{I}}^1(\mathbf{p}^1) + (1-\lambda)\,\hat{\mathcal{I}}^2(\mathbf{p}^2) & (6) \end{aligned}$$

3. *De-rectification:* the rectified view $\hat{\mathcal{I}}'$ is de-rectified to produce a physically-valid in-between view $\mathcal{I}'$.

Since the rectification gives parallel scan-lines, correspondences are found between uniform intervals in conjugate scan-lines in the two reference images.

3 Experiments

We implemented the proposed method and tested it on real-video images. A total of 26 model views were obtained so that the interior angles of any two neighboring view vectors were about 15 to 30 degrees. This may seem to be a large number of views. However, there is a tradeoff between the number of views and the quality of synthesized view. Using largely separated model views, it is difficult to have a novel view of good quality. Figure 6 shows on the first column some video images in which the black lines on the plane are to evaluate our method qualitatively. Second column is the augmented video images. 'Movie 1' is the corresponding video clip. Notice that the synthesized model views follow the motion of video and the model object keeps its pose during the motion.

We have found four major defects in the augmented video images. They are 1) misalignment between model and video, 2) smoothing in view generation 3) black borders by imperfect segmentation and 4) trembling effects of inserted object in movie. At the initialization step of 2.4, we tried to align two bottom edges of the model object with the inside edges of the black lines of the video. However, some discrepancies have occurred as shown in figure 8 by solid lines. Errors in structure recovery and perspectiveness in the real images could be the source of the discrepancies. for view synthesis, model views should be as close as possible. he rotation Because the interpolation itself is a smoothing procedure and the resolution of reference views is not highe rotation smoothed as one can see. In addition, two-stage interpolation makes the result worse. Black borders are due to imperfect extraction of the model object from reference images, which is shown in figure 8 by dotted lines. Finally, when the video images are seen continuously, the overlayed part shows a motion like non-rigid body, which requires filtering of parameters from SFM and more appropriate image systhesis method.

Figure 7 shows another example of video augmentation, in which only 3 model views are used to get the novel views among 17 model views.

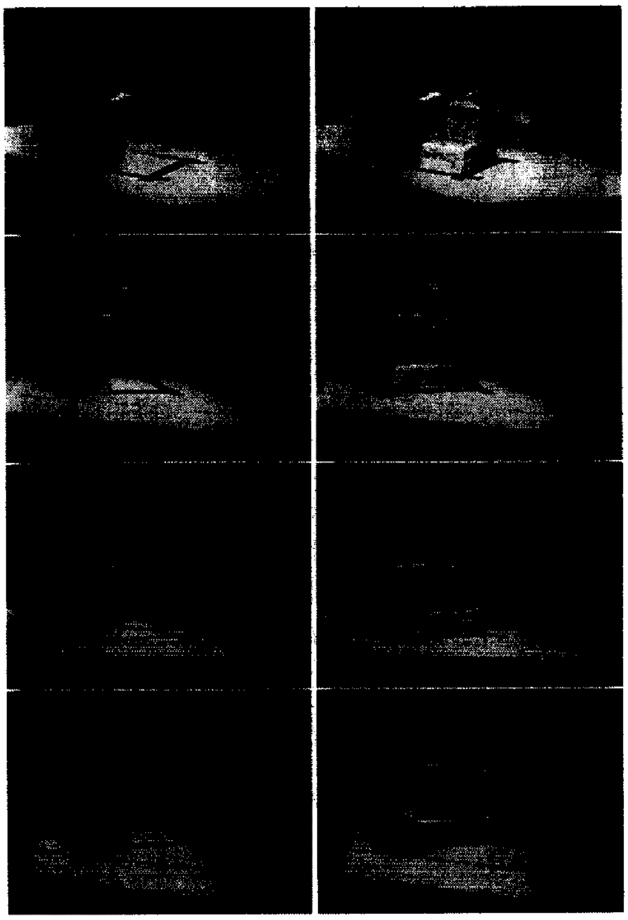


Figure 6: Left: video images, right: augmented version of the left

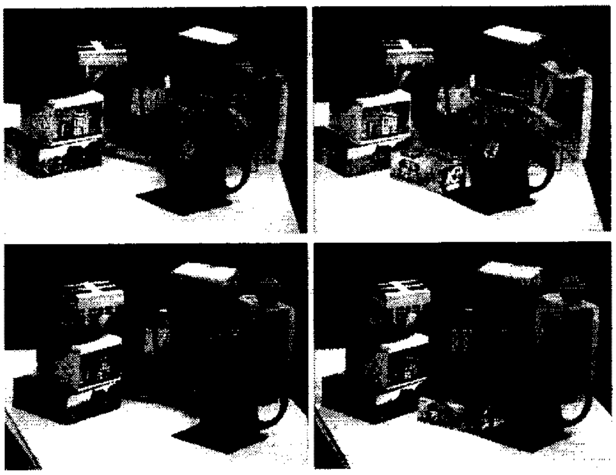


Figure 7: Another result of video augmentation.

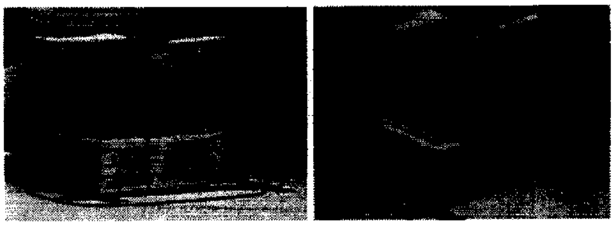


Figure 8: Magnified view of the results. Some discrepancies in matching the edge lines are shown (box of solid line). Parts of model object is imperfectly extracted from reference images as shown by the black borders (box of dotted line).

4 Conclusion

We proposed a method to augment a video using image-based rendering. Motion from video images computed using SFM algorithm and structure information from reference images are used to determine interpolation parameters. The locations of model objects are specified in two video images using epipolar geometry and structure information recovered. This plays the same role as locating a model object in camera coordinate system in 3D graphics. According to the interpolation parameters, novel views are generated and overlayed on video images using view interpolation technique. In order to make a novel view at arbitrary camera position, three model views are selected using the relation of view vectors and two stage interpolation method is applied.

In the experiments, real images are used for both reference images and video images. Novel views overlayed on video images are looked as if the model object was on the place. Four main problems which require more study are:

- trembling effects in continuous movie.
- smoothing effect due to interpolation,
- pose misalignment due to errors in SFM and feature tracking,
- and black borders by incomplete image segmentation.

In addition, a fast interpolation algorithm is required on account of the fact that we have to do scan-line matching which takes a lot of computation time at the second stage of view rendering although we have reference images and know scan-line correspondences between any two images of a mesh. One solution is to find full pointwise correspondences and use the algebraic function of [16] like the work of [1].

In spite of these defects, we think that there is some promise that this study can be used *practically* to insert a real object into real video without making a 3D model of the object. Now we are developing a method to augment a video by image-based rendering under *perspective* camera model using the warping function of [1].

Appendix

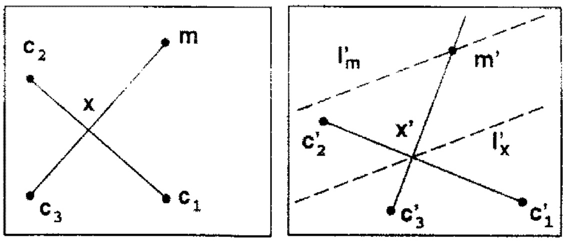


Figure 9: Point-plane procedure determines the image $\mathbf{m}'$ of $\mathbf{m}$ which is on the plane defined by the three points.

Assume that we know the epipolar geometry between two views $\mathcal{V}_1$ and $\mathcal{V}_2$. Let $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\}$ and $\{\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3\}$ are coplanar and not colinear matching points in $\mathcal{V}_1$ and $\mathcal{V}_2$, respectively. If $\mathbf{m}$ is a point on the plane in $\mathcal{V}_1$, The point-plane procedure constructs $\mathbf{m}'$, its correspondence, in $\mathcal{V}_2$[4]. First, the intersection point $\mathbf{x}$ of the line $\mathbf{c}_1\mathbf{c}_2$ and $\mathbf{c}_3\mathbf{m}$ in $\mathcal{V}_1$ is computed. Then, $\mathbf{x}'$ is the intersection of $\mathbf{c}'_1\mathbf{c}'_2$ and $\mathbf{l}'_\mathbf{x}$, the epipolar line of $\mathbf{x}$ in $\mathcal{V}_2$. Now we can draw the line $\mathbf{c}'_3\mathbf{x}'$. Finally, $\mathbf{m}'$ is the intersection of $\mathbf{c}'_3\mathbf{x}'$ and $\mathbf{l}'_\mathbf{m}$, the epipolar line of $\mathbf{m}$ in $\mathcal{V}_2$.

References

[1] Shai Avidan and Amnon Shashua. Novel view synthesis in tensor space. In *Proceedings of CVPR'97*, 1997.

[2] Michael Bajura and Ulrich Neumann. Dynamic registration correction in video-based augmented reality systems. *Computer Graphics and Applications*, pages 52–60, 1995.

[3] M.-O. Berger, G. Simon, S. Petitjean, and B. Wrobel-Dautcourt. Mixing synthesis and video images of outdoor environments: Application to the bridges of paris. In *Proceedings of 13-th International Conference on Pattern Recogniton*, pages 90–94, 1996.

[4] O. Faugeras. Stratification of three-dimensional vision: Projective, affine, and metric representations. *J. Opt. Soc. Am. A*, 12(3):465–484, Mar. 1995.

[5] O.D. Faugeras. *Three Dimensional Computer Vision: A Geometric Approach.* MIT Press, 1993.

[6] Simon Gibbs, Costas Arapis, and Christian Breiteneder. Virtual studios: The state of the art. In *Proc. Eurographics'96 State of the Art Reports*, Feb. 1996.

[7] Pierre Jancène, Fabrice Neyret, Xavier Provot, Jean-Philippe Tarel, Jean-Marc Vézien, Christophe Meilhac, and Anne Verroust. Res: computing the interactions between real and virtual objects in video sequences. In *Proceedings of Networked Realities Conference*, 1995.

[8] Kiriakos N. Kutulakos and James R Vallino. Affine object representations for calibration-free augmented reality. *Proceedings of 1996 IEEE Virtual Reality Annual International Symposium*, pages 25–36, 1996.

[9] Stephane Laveau and Olivier Faugeras. 3-d scene representation as a collection of images and fundamental matrices. In *Proceedings of the 12th International Conference on Pattern Recogntion*, pages 689–691, Oct. 1994.

[10] C.H. Lee and T. Huang. Finding point correspondences and determining motion of a rigid object from two weak perspective views. *Computer Vision, Graphics, and Image Processing*, 52:309–327, 1990.

[11] D. Scharstein. Stereo vision for view synthesis. In *Proceedings of CVPR'96*, pages 852–858, 1996.

[12] Steven M. Seitz and Charles R. Dyer. Physically-valid view synthesis by image interpolation. In *Proceedings of IEEE Workshop on Representations of Visual Scenes*, pages 18–25, 1995.

[13] Steven M. Seitz and Charles R. Dyer. Toward image-based scene representation using view morphing. In *Proceedings of the 13th International Conference on Pattern Recogntion*, pages 84–89, Oct. 1996.

[14] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

[15] Roger Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE journal of Robotics and Automation*, 3(4):323–344, 1987.

[16] Shimon Ulman and Ronen Basri. Recognition by linear combinations of models. *PAMI*, 992-1005 1991.

[17] T. Werner, R.D. Hersch, and V. Hlavac. Rendering real-world objects using view interpolation. In *Proceedings of the International Conference on Computer Vision*, pages 957–962, 1995.

[18] Peter Wisskirchen, Klaus Kansy, and Güther Schmitgen. Integrating graphics into video image-based camera tracking and filtering. In *Proc. Eurographics'96 Workshop on Virtual Environments*, Feb. 1996.