

# Augmented Reality

Nipun Kwatra

Sumit Jain

November 20, 2003

## Abstract

*We present here a novel approach to augment destination environments with objects cut from source environments (without prior information), rather than augmenting 3-d models, which allows us to create virtual environments augmented with real objects such that the user cannot tell the difference between the real world and the virtual augmentation of it.*

*The current work models the source objects as planes. A modified version of the ray tracing algorithm was developed which renders the destination environment using these planar objects and also the possible occluding planes in the destination environment. These planes in the destination environment were modelled using single view 3d reconstruction methods.*

## 1 Introduction

In the past few years, Virtual Environments (a.k.a. Virtual Reality) have attracted a great deal of attention. In Augmented Reality, the user can see the real world around him, with computer graphics superimposed or composited with the real world. The ultimate goal is to create a system such that the user can not tell the difference between the real world and the virtual augmentation of it. To the user of this ultimate system it would appear that he is looking at a

single real scene.

### 1.1 Previous Work

Previous work involves augmenting 3-d models into destination environments, using vision based position tracker. A limitation of these approaches is that they are limited to only augmenting already available 3-d models into destination environment and do not permit augmenting free objects from any source environment without any prior information.[1]

## 2 Extracting the source objects

Before augmenting the destination environment with a source object, we need to extract the object from the source environment. This is implemented by an extremely general and automatised process.

### 2.1 Background subtraction

This technique is extremely general if the background in which the object moves is known beforehand. The process involves simply subtracting from each frame the background frame, giving the source object as a result. The technique is further enhanced for better results by using continuity analysis and extracting the largest connected component.

## 2.2 Extracting the Texture

Once the background subtraction is done, we need to extract the texture of the objects from each frame. This can be done by clicking the bounding points of the source object in the original frame, and applying the homography thus obtained to the background subtracted frame. The hand-clicking needs to be done only on the first frame and a tracker can be used for the following frames.

## 3 Processing the destination environment using interactive methods

If the source object in the source environment is moving on a planar trajectory, we should be able to make it move in a similar planar trajectory in the destination environment so that the augmented environment looks as real as possible.

Thus to correctly paste the source object into the destination environment and to handle other problems like occlusion, we need to know 3-d information of the planes (note that we have used only planar modelling presently) on which we want the object to move (the corresponding ground plane of the destination environment), and those of the possible occluders.

### 3.1 Calibrating The Destination environment

The above objective is obtained by calibrating the destination environment using a simple interactive 3-D reconstruction from a single view[3]. This calibration enables determination of 3-D co-

ordinates of some points of interest by interactive methods. This information can then be used to determine the equations of the planes of interest. The camera Matrix that obtained above is used for rendering the final augmented environment using a modified version of the ray-tracing algorithm.

### 3.2 Sparse 3-D modelling of possible occluders

The augmented object may be occluded by some of the objects of the destination environment. As the no. of such occluders tends to be quite low, it is not a bad idea to model these, and later render these along with the source object, so as to handle occlusion effectively. We have modelled the occluders as planes which is a reasonable assumption as long as the trajectory of the augmented object does not go inside through these possible occluders.

The modelling involves:

1. Extracting the texture of possible occluding objects.
2. Determination of 3-D co-ordinates of points of possible occluding objects, to determine its plane equation.[3]

## 4 Registration of source and destination environments

To map the trajectory of motion of the source object from the source environment to destination environment, we need to do some interactive registration. This interactive method gives complete flexibility in choosing the plane of trajectory in the destination environment. For e.g.

registering a ground plane of the source environment with the ceiling or may be a wall in the destination environment will make an object moving in the ground plane in the source environment to move on the ceiling or on a wall in the destination environment.

#### 4.1 Registration of corresponding ground planes

This involves clicking of 4 corresponding points on the (virtual)ground planes of the 2 environments. This registration will give us a homography between the source and destination (virtual)ground planes, say **hom\_source\_dest**.

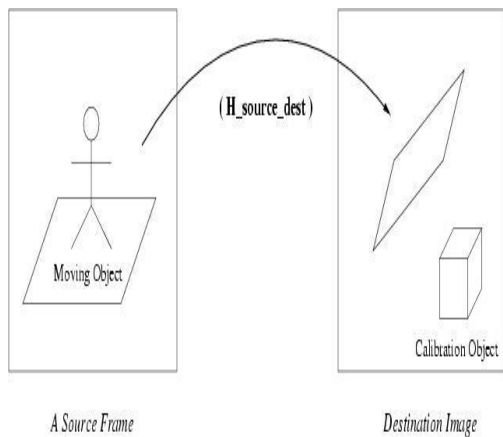


Figure 1: registration

This registration is also used to find another homography matrix, between the above destination plane and an imaginary plane with corners at  $(0,0)$ ,  $(1,0)$ ,  $(0,1)$  &  $(1,1)$ , say **hom\_implane2Gplane**. This functionality of these homographies are explained later.

#### 4.2 Registration of destination ground plane's 3-D co-ordinates

The 3-D co-ordinates of the above points of the destination ground planes can be found using the single view reconstruction method[3]. Let the 3-D co-ordinates of these points in the clicking order be  $\vec{p00}$ ,  $\vec{p10}$ ,  $\vec{p01}$ ,  $\vec{p11}$ .

#### 4.3 Determination of height of the object in the destination environment

The height of the object in the destination environment cannot be find from the above method due to the constraints of the single view reconstruction method, and has to be found through indirect method like inspection, or by keeping the aspect ratio constant, or comparing the height of an object kept in both the source and destination environments.

### 5 Determination of object plane in the destination environment

Due to constraints of the single view reconstruction method, we can determine this plane only upto 2 degrees of freedom, i.e. we cannot determine the inclination of this plane with the ground plan. We force the 3rd degree of freedom from the assumption that the object plane is perpendicular to the ground planes of the 2 environments.

### 5.1 Determination of 3-D co-ordinates of a point in destination environment from its pixel co-ordinates in the source environment

The problem is to find the 3-D co-ordinates of a point in the destination environment, given its pixel co-ordinates in the source environment.

Let the pixel co-ordinates be  $\vec{v}$ .

$$\vec{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} \quad (1)$$

The corresponding pixel co-ordinates (in homogenous co-ordinates) of the point  $\vec{v}$  in the destination image are given by the following equation:

$$\vec{v}_2 = \mathbf{H\_source\_dest} \begin{pmatrix} \vec{v} \\ 1 \end{pmatrix} \quad (2)$$

The matrix **hom\_implane2Gplane** found above in Section 4.1 can be used to find the co-ordinates of  $\vec{v}_2$  in a local co-ordinate system attached to the destination ground plane registered in Section 4.1. This point  $\vec{v}_3$  (in homogeneous co-ordinates) is given by:

$$\vec{v}_3 = \mathbf{hom\_implane2Gplane} \times \vec{v}_2 \quad (3)$$

The corresponding cartesian co-ordinates are given by;

$$\vec{v}_4 = \begin{pmatrix} v3_x/v3_z \\ v3_y/v3_z \end{pmatrix} \text{ where } \vec{v}_3 = \begin{pmatrix} v3_x \\ v3_y \\ v3_z \end{pmatrix} \quad (4)$$

Given the vectors  $\vec{p00}$ ,  $\vec{p10}$ ,  $\vec{p01}$ ,  $\vec{p11}$  evaluated in Section 4.2, the 3-D co-ordinates of the

point  $\vec{v}_5$ , corresponding to  $\vec{v}$  in the destination environment is given by:

$$\vec{v}_5 = \vec{p00} + v4_x(\vec{p10} - \vec{p00}) + v4_y(\vec{p01} - \vec{p00}) \quad (5)$$

$$\text{, where } \vec{v}_4 = \begin{pmatrix} v4_x \\ v4_y \end{pmatrix}$$

### 5.2 Determining the bounding quadrilateral

We need to find the bounding quadrilateral in the destination environment to which the texture of the source object goes. This needs to be found by finding the corresponding 3-D co-ordinates of the bounding points evaluated in Section 2.2, but since the above method allows determination only for points in the ground plane of the source image, the 3-D co-ordinates of points not in the ground plane are evaluated by adding the height determined in Section 4.3 in a direction perpendicular to the ground plane of the destination environment, to the 3-D co-ordinate of a corresponding ground plane point.

## 6 Rendering

Once the determination of object plane in the destination environment, and sparse modelling of possible occluders is done, we have the complete information for rendering the augmented environment. A modified version of the ray-tracing algorithm has been developed, which is very effective for rendering of planar objects. Given  $n$  planes in some world co-ordinate system & the camera  $Mx$ , in the same co-ordinate system, the problem is to render the image that the camera will view.

We present a simple approach which is a modification of the ray-tracing algorithm:

Let  $\mathbf{K}$ ,  $\mathbf{R}$ ,  $\mathbf{t}$ , be the camera parameters of the destination environment obtained by the calibration in Section 3.1

### 6.1 Determination of equation of plane

The bounding points of the object plane were determined from Section 5.2 and eq. 5, and those of the possible occluders were determined in Section 3.2.

Once we know any 4 points of the plane, the equation of the plane is easily determined. We denote the equation of the  $k^{th}$  plane by the 4-tuple  $\{a_k, b_k, c_k, d_k\}$ .

### 6.2 Determination of image to world plane Homographies

This homography for the  $i^{th}$  plane,  $\mathbf{H}_i$  is determined by the following relation.

$$\mathbf{H}_i = \mathbf{K}[\mathbf{R} - \mathbf{t}\mathbf{n}_i^t/d_i] \quad (6)$$

$$, \text{ where } [\mathbf{n}_i \ d_i] = [a_k \ b_k \ c_k \ d_k]$$

### 6.3 Determination of corresponding 3-D points of a pixel in different planes

Given a pixel and a plane, we want to determine the 3-D co-ordinate of the point, whose image the pixel would have been if no body occluded the point.

#### 6.3.1 Determination of ray vector corresponding to a pixel co-ordinate and a plane

Let  $\mathbf{X}$  be a point on some plane with the image to world plane homography  $\mathbf{H}$ . Then the pixel co-ordinate  $[u \ v \ 1]^t$  (in homogeneous co-ordinates) are given by

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{H}\mathbf{X} \quad (7)$$

As the homogenous co-ordinates are equivalent upto a scale, the inverse homography  $\mathbf{H}^{-1}$  applied on the pixel co-ordinates will give a ray in direction parallel to the  $\mathbf{X}$  vector. Let

$$\mathbf{D} = \mathbf{H}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad (8)$$

#### 6.3.2 Calculating the 3-D co-ordinates

Now given this direction vector  $\mathbf{D}$ , we can find the exact 3-D co-ordinates of the corresponding point in a given plane. Let the vector co-ordinates of 3 points in plane (in the order they were clicked), be  $\mathbf{r00}$ ,  $\mathbf{r10}$  &  $\mathbf{r01}$ . Let

$$\mathbf{x} = \mathbf{r10} - \mathbf{r00} \quad (9)$$

$$\mathbf{y} = \mathbf{r01} - \mathbf{r00} \quad (10)$$

Then

$$\mathbf{r00} + a\mathbf{x} + b\mathbf{y} = k\mathbf{D} \quad (11)$$

for some  $a$ ,  $b$ ,  $k$  where  $(a, b)$  are the co-ordinates of the points in the local co-ordinate system of the plane (assuming  $\mathbf{x}$  and  $\mathbf{y}$  to be the unit vectors), and  $k$  is the scale factor for the direction vector.

Representing these vectors as column vectors, i.e.

$$\mathbf{r00} = \begin{pmatrix} r00_x \\ r00_y \\ r00_z \end{pmatrix}, \mathbf{x} = \begin{pmatrix} x_x \\ x_y \\ x_z \end{pmatrix}, \mathbf{y} = \begin{pmatrix} y_x \\ y_y \\ y_z \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} D_x \\ D_y \\ D_z \end{pmatrix}$$

the above equation transforms to:

$$\begin{pmatrix} r00_x \\ r00_y \\ r00_z \end{pmatrix} + a \begin{pmatrix} x_x \\ x_y \\ x_z \end{pmatrix} + b \begin{pmatrix} y_x \\ y_y \\ y_z \end{pmatrix} = k \begin{pmatrix} D_x \\ D_y \\ D_z \end{pmatrix} \quad (12)$$

i.e.

$$\begin{pmatrix} x_x & y_x & -D_x & r00_x \\ x_y & y_y & -D_y & r00_y \\ x_z & y_z & -D_z & r00_z \end{pmatrix} \begin{pmatrix} a \\ b \\ k \end{pmatrix} = \mathbf{0} \quad (13)$$

Solving the above equation with normal SVD, we obtain the values of a, b, k.

Thus the corresponding points  $\mathbf{X}$  of a pixel (u,v) in a plane is given by:

$$\mathbf{X} = k\mathbf{D} \quad (14)$$

#### 6.4 Determination of closest point

To render the final image, we want to determine for each pixel, the plane from which the point actually came.

The camera centre is given by

$$\mathbf{C} = -\mathbf{R}^t \mathbf{t}; \quad (15)$$

The distance,  $d_i$  of the 3-D co-ordinate  $\mathbf{X}_i$  corresponding to the  $i^{th}$  plane (obtained in equation-14) from the camera centre then follows;

$$\mathbf{d}_i = \mathbf{X}_i - \mathbf{C} \quad (16)$$

Thus

$$d_i = \|\mathbf{d}_i\| \quad (17)$$

Clearly the plane with the minimum  $d_i$  will finally appear on the image. The 3-D co-ordinates of the pixel corresponding to this plane gives the point that is actually seen in the image. Note: Before using the above procedure it is also checked whether the point lies in the bounding quadrilateral of a plane and only then it is considered for the minimisation. Thus there may not exist any modelled plane to which the pixel corresponds, in which case the original texture is kept.

#### 6.5 Pasting the texture information

Once the 3-D point and plane corresponding to a pixel is known, we need to bring in the texture of that point and paste it onto this pixel.

Let (a,b) be the co-ordinates of the point in the local co-ordinate system of the plane obtained from equation-13. This information can be used to determine the texture value from the textures of the planes that were cut before.

The above procedure when applied to each pixel of the destination environment gives the final augmented image.

## 7 Experimental Results

We have implemented the approach described in this paper and applied it to create virtual environments from wide variety of target images and a source video. Only three of these results are presented in this section but higher resolution images and videos can be found online [3]. We encourage the reader to peruse these results online to better gauge the quality of the our work.

We experimented with a video of a person moving in a laboratory scene (shown below). Objects which can be modelled using a plane are easy to augment using our approach. In the experiment shown in fig 8, instead of having single destination image we had a video with still camera but a movie being played on a desktop. We can support this unless the moving objects in the destination do not interfere with new object added otherwise we have to model the moving objects in the destination video also using same technique as used for original object. The first image of fig 8 shows a single frame of the environment where the person is to be augmented. The second image shows a frame of augmented environment. In experiment 1 (fig 8) 4 planes were modelled in the destination image for occlusion handling. The rest of the procedure was similar to experiment 2. The person was extracted and augmented on a per frame basis. We worked on 110 frames (of source video) and required 25 - 100 seconds per frame depending on number of planes modelled. The overall manual hand clicking of points required 2 minutes. In experiment 3 we made the person walk on the ceiling of the lab. This was done trivially by registering the ceiling with the ground plane. The rest of the procedure was similar to rest of the experiments.

## 8 Conclusion

We have presented a simple interactive method for augmenting real moving objects in a still scene using planar assumption for object as well as occluders. In this paper, it was argued that a reasonable amount of user interaction is sufficient to create high-quality virtual environments from a destination image and a source video

with planar assumptions on the occluders and the moving objects.

There are a number of interesting avenues for future research in this area. An important extension would be to devise a scheme to generate 3D models of objects from some set of images and make the object move the way user specifies instead of modelling them as planes.

## References

- [1] Simon, G., Fitzgibbon, A. and Zisserman, A., "Markerless Tracking using Planar Structures in the Scene", *In Proc. International Symposium on Augmented Reality, October, 2000*
- [2] Li Zhang, Guillaume Dugas-Phocion, Jean-Sebastien Samson, Steven M. Seitz, "Single View Modelling from Free-Form Scenes", *CVPR 2001*
- [3] Akash M Kushal, Vikas Bansal, Subhashis Banerjee, "A simple method for interactive 3D reconstruction and camera calibration from a single view", *ICVGIP 2002*



Figure 2: A frame of the source movie

We extracted the person from the video as explained in Section 3. Then it was augmented in various destination environments. The results(only single frame of the augmented video shown here) are shown below alongwith the corresponding destination image.



Figure 3: Experiment 1



Figure 4: Experiment 2



Figure 5: Experiment 3