Real time Virtual View Generation For Augmented Virtuality System

Yong Tang
Nanjing University of Aeronautics and Astronautics Nanjing, 210016, P.R.China
Email: tangyong159@gmail.com

Hongbin Gu and Lai Zhou
Nanjing University of Aeronautics and Astronautics Nanjing, 210016, P.R.China
Email: ghb@nuaa.edu.cn, zhoulai@nuaa.edu.cn

Abstract—**This paper describes a novel method of real-time virtual view generation for augmented virtuality system. The aim is to rectify the displacement between virtual eye position and video cameras which are attached on the head mounted display. It consists of three steps: Firstly, stereo calibration was used to generate cameras' internal and external parameters, and then image planes were rectified to canonical configuration state that made them alignment or corresponding points on the same scan lines. At last the virtual view images between two cameras are generated using view morphing which is a shape-preserving transition method from a source to a destination view. We can adopt this method to choose locations of two virtual cameras yielding an in-between stereoscopic camera views without reconstructing the 3D model of the object. The view of the object corresponding to a virtual camera that is located between two real cameras with same position of eyes. So we can control the feeling of depth and provide customized viewpoint to the user. It can be used in a variety of augmented virtuality applications to integrate live video objects into virtual environments. Experimental results verify the validity of the proposed approach, and we also show interlaced stereo composition results using a pair of synthesized views from input views acquired by stereo cameras.**

Index Terms— **view morphing, stereo calibration, stereo rectification, augmented virtuality**

I. Introduction

The rapid progress in computer vision, computer graphics and enables the user to enter a realistic augmented virtuality (AV) space. When the user wearing video see-thru HMD in such systems, the real image is captured two cameras that processed in a computer (augmented data) is added on the images display, subjects which the user see around them are the fusion of real object (captured by carmeras) and virtual environment (generated by computer) [1]. Such as the Poser [2], a augmented virtuality system that integrating the hand image to virtual environment to enhance the presence in HMD environments by visual body feedback. Unfortunately, the cameras have the same position as eyes is impossible because of the limitation of the HMD's physical structure in AV system. We have to set the virtual viewpoint in virtual environment consist with the cameras' position to ensure the correct registration in our system. In this case, it makes a displacement that between the real and virtual eye position, and this may cause uncoordination, uncertainty, and inaccuracy on the user’s body [3]. Although human brain is capable to adapt very fast when an immersive system is used, still there remains a problem when using the HMD after he adapted the reality. This is why the position of the cameras can not be ignored in our investigation. Currently, where to fix the camera are categorized as follow [4]:

- Placed each camera in front of each eye. It is adopted widely, but this includes that the seen images are closer and bigger as in reality, such as the poser which the paper mentioned.
- Place each camera next to each eye or above it. This has the disadvantage that it is not exactly the same view as with the eyes.
- Place the camera next to helmet and in front of the face pointing to a mirror which is with a 45 angle in front of the eyes. This give the user the same impression as viewing with his naked eyes, however the size of the mirror should be too big to place two of them and the construction gets to bulky.

In this case, it is necessary to find a way to eliminate the displacement of between the real and virtual eye position. In this paper, a virtual view generation was introduced to correspond to the user’s viewpoint in the AV space. Since a technique which was called novel view synthesis [5] that aims at generating the view of an object corresponding to a virtual camera using a set of 2D reference images, we can fixed cameras next to eyes and generate new virtual view images that correspond to install two virtual cameras which have the same position of eyes on the HMD. In novel view synthesis system, view morphing algorithmic [6] is proposed to generate novel views of an object that is observed by two fixed video cameras widely because of its simpleness and efficiency. So we can make live video frames which was calibrate and rectified as input, the realistic view corresponding to a virtual camera that fix between the

doi:10.4304/jcp.6.5.873-880

two real cameras is generated in real-time based on view morphing. Without recovering the 3D model of the object, the resulting novel view ensures a natural and a new virtual view image of the same position of eyes between the captured image pair. The proposed method can be used in various applications in AV system. For example, an object (e.g. the hand) that is captured in real-time by two video cameras in a natural environment can be integrated into a AV space (e.g. a virtual touch board), where the user is able to observe the new virtual view image in real-time during his/her interaction in the AV environment.

2. RELATEDWORKS

The methods for generating arbitrary view images from real camera images have been studied since the 1990s in the field of computer vision [7]. These techniques which are called Image Based Rendering (IBR) can be categorized into two groups, Model Based Approach and Transfer Based Approach. Model Based Approach constructs a 3D shape model of an object to generate the desired view [8, 9]. Since the quality of the virtual view image depends on accuracy of the 3D model, a large number of video cameras surrounding the target object or range scanners are used for construction of an accurate model. Also strong camera calibration, which is carried out to relate 2D coordinates in images to 3D coordinates in object space, is usually required. As 3D positions of several points in the object space must be measured, this calibration becomes difficult especially in a large space. For such reasons, the object area is generally limited within a few cubic meters in this approach. On the other hand, Transfer Based Approach synthesizes arbitrary view images without an explicit 3D model [10]. Instead, image warping such as transfer of correspondence is employed for synthesizing new view images. The dense correspondence between the original images, which is required for view-synthesis, is often obtained manually or by the use of optical-flow, so almost all targets are static images or slightly varying images such as facial expression. Thus we have proposed view-synthesis method targeting dynamic events in our paper to replace the position of the camera, which is included in Transfer Based Approach [11, 12]. As for arbitrary view presentation, A related approach to our method has been proposed in [13, 14]. Naho Inamoto described a new framework for arbitrary view synthesis and presentation of sporting events for mixed reality entertainment. Virtual view image of sporting scene is generated by view interpolation among multiple videos captured at real stadium. The user can watch the superimposed images of a remote collaborator in the real world. Hua Chen described a novel method of real-time novel view synthesis for an object that is observed by two fixed video cameras in a natural environment. In above system, the object was remote captured and should use user input for post-warping. In addition, the generated in-between views are much affected by self-occlusion. One advantage of our method is based on projective geometry between two cameras instead of reconstructing 3D models.

3. OVERVIEW

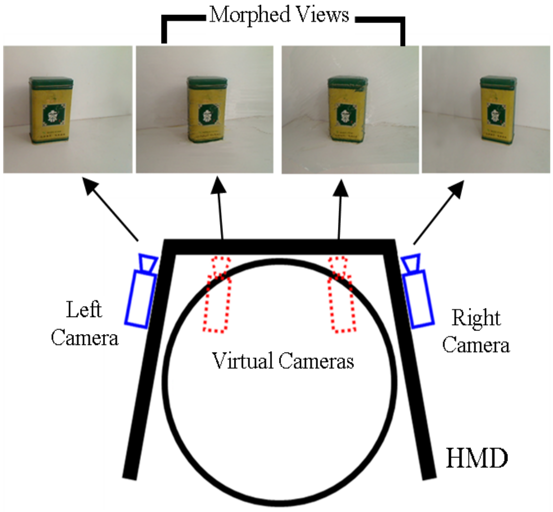

Figure 1. The overview of virtual view generation system

Figure 1 shows overview of the proposed method. A video see-through HMD is used for AV environment. An observer sees the views of two virtual cameras which are generated by the real cameras that attach next to the HMD, so the virtual viewpoint can keep the same position of eyes in AV environment.

In order to generate novel new views, three step was implemented as follow.

- Calibrate the pair of cameras.
- Rectification the of pair cameras
- Generate new virtual images between two pair cameras

Step (1) is process of computing the geometrical relationship between the two cameras in space. In contrast, Step (2) is the process of “correcting” the individual images so that they appear as if they had been taken by two cameras with row-aligned image planes. With such a rectification, the optical axes of the two cameras are parallel and so we say that they intersect at infinity. At the step (3), neighboring cameras near the viewpoint position, which are reference cameras, synthesize the virtual viewpoint image of the new view by linear interpolation. At the final stage, synthesized object scene is overlaid on virtual environment. The observer can virtually watch AV environment from favorite viewpoints in the system.

This paper is organized as follows. Section 4 describes how to calibrate the cameras and calculate the necessary parameter. Section 5 presents the solution to stereo rectification is to project the image planes of cameras so that they reside in the exact same plane. Section 6 how to generate novel new views of the object corresponding to the user’s viewpoint in an AV space real time. Experimental results are given in section 7. Finally, conclusions and some future work are summarized in section 8.

4. Calibration Of two cameras

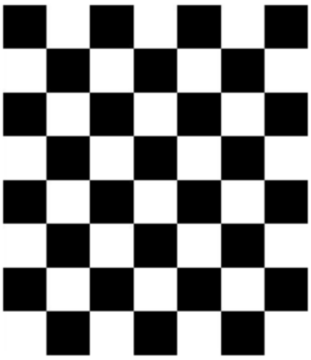

Figure 2. Planar Pattern in Zhang's Calibration Method

Stereo calibration is the process of computing the geometrical relationship between the two cameras in space, which depends on finding the internal parameter, the rotation matrix R and translation vector T between the two cameras, and then we can calculate essential matrix, fundamental matrix and so on. In this paper, we calibrated two cameras based on the Zhang Plane-based Calibration Method which proposed a flexible camera calibration technique by replacing expensive classical calibration grid with a planar pattern for single camera [15]. A print sheet (see figure 2) as the model plane, and the Euclidean coordinates of every dot on the model plane should be measured accurately. After taking a few images of the model plane at different orientations by moving either the model plane or the camera, the homographies between the model plane and its projections can be determined, and then camera ' s intrinsic parameters can be derived linearly from these homographies. Zhang's technique is flexible and low-cost, its accuracy is generally higher than self-calibration.

The basic steps of this method are as follows: print a chessboard pane picture and paste it on a plane, shoot more than 3 model images from different angels, detect the feature points from every image, calculate the plane projection matrix H for every image through the detected points, at last determine the parameters of camera.

Supposed that the model plane coincides with the plane in the world coordinate system where $Z = 0$. K indicates the internal parameter matrix of the camera, r_i is the i^{th} vector of R, for every point in the model plane we have

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = K\begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (1)$$

Let $H = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}$, so we get a homography H between the point in model plane and its corresponding image point, if known the coordinate of the point in model plane, we can compute the homography H .

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2)$$

Obviously H is a 3-by-3 matrix, where:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \quad (3)$$

So

$$\begin{cases} uXh_{31} + uYh_{32} + u = h_{11}X + h_{12}Y + h_{13} \\ vXh_{31} + vYh_{32} + v = h_{21}X + h_{22}Y + h_{23} \end{cases} \quad (4)$$

Let

$$h' = [h_{11} \quad h_{12} \quad h_{13} \quad h_{21} \quad h_{22} \quad h_{23} \quad h_{31} \quad h_{32} \quad 1]^T \quad (5)$$

We can get:

$$\begin{bmatrix} X & Y & 1 & 0 & 0 & 0 & -uX & -uY & -u \\ 0 & 0 & 0 & X & Y & 1 & -vX & -vY & -v \end{bmatrix} h' = 0 \quad (6)$$

Then we can detect some feature points in one image, then we have $Sh' = 0$, through computing these equations by using least square method, we can get h' and H . After generating the homography matrix, the next step is to compute the internal parameter.

Define every row of H as h_i , then

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda K\begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \quad (7)$$

Because r_1 and r_2 are orthonormal, there are two constraints:

$$h_1^T K^{-T} K^{-1} h_2 = 0 \quad (8)$$

$$h_1^T K^{-T} K^{-1} h_2 = h_2^T K^{-T} K^{-1} h_2 \quad (9)$$

Set $B = M^{-T} M^{-1}$

Writing this out, we have:

$$B = K^{-T} K^{-1} \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \quad (10)$$

matrix B has a general closed-form solution:

$$B = \begin{bmatrix} \frac{1}{f_x^2} & 0 & \frac{-c_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & \frac{-c_y}{f_y^2} \\ \frac{-c_x}{f_x^2} & \frac{-c_y}{f_y^2} & \frac{c_x}{f_x^2} + \frac{c_y}{f_y^2} + 1 \end{bmatrix} \quad (11)$$

Using the B-matrix, both constraints have the general form $h_i^T B h_j$ in them. Let' s multiply this out to see what the components are. Because B is symmetric, it can be written as one six-dimensional vector dot product. Arranging the necessary elements of B into the new vector b, we have:

$$h_i^T B h_j = v_{ij}^T b = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix} \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix} \tag{12}$$

Using this definition for v_{ij}^T, our two constraints may now be written as:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \tag{14}$$

If there are n images, we have such equations through

$$Vb = 0 \tag{15}$$

Where V is a 2n× 6 matrix. We can calculate b and B from (18). The internal parameters can be obtained by using Choleski decomposition.

$$f_x = \sqrt{\lambda / B_{11}}$$

$$f_y = \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)}$$

$$c_x = -B_{13} f_x^2 / \lambda$$

$$c_y = (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{12} - B_{12}^2) \tag{15}$$

Where:

$$\lambda = B_{33} - c_y (B_{13}^2 + c_y (B_{12}B_{13} - B_{11}B_{23})) / B_{11}$$

Where V is a 2K-by-6 matrix. As before, if $K >= 2$ then this equation can be solved for $b = [B_{11} \quad B_{12} \quad B_{22} \quad B_{13} \quad B_{23} \quad B_{33}]$. The camera intrinsic parameters are then pulled directly out of our closed-form solution for the B-matrix:

$$r_1 = \lambda K^{-1} h_1$$

$$r_2 = \lambda K^{-1} h_2$$

$$r_3 = r_1 \times r_2$$

$$t = \lambda K^{-1} h_3 \tag{16}$$

Here the scaling parameter is determined from the orthonormality condition $\lambda = 1 / \| K^{-1} h_1 \|$.

5. RECTIFICATION OF TWO CAMERAS

The goal of rectification is to reproject the image planes of two cameras to ensure them reside in the exact same plane, with image rows perfectly aligned into a frontal parallel configuration. The rotation matrix and translation matrix (R, T) between the stereo images are generated in section 4, So we can make use of Bouguet's algorithm[16] for stereo rectification simply attempts to minimize the amount of change reprojection produces for each of the two images (and thereby minimize the resulting reprojection distortions) while maximizing common viewing area.

In order to minimize image reprojection distortion, the rotation matrix R that express the rotation between the right camera's image plane and the left camera's image plane is split in half between the two areas, respectively. In our method, each camera rotates half a rotation of R, so their principal rays each end up parallel to the vector sum of where their original principal rays had been pointing. Such a rotation puts the cameras into coplanar alignment but not into row alignment. To compute the matrix R_{rect} that will take the left camera's epipole to infinity and align the epipolar lines horizontally, we create a rotation matrix by starting with the direction of the epipole e_1 itself. Taking the principal point (c_x, c_y) as the left image's origin, the direction of the epipole is directly along the translation vector between the two cameras' centers of projection:

$$e_1 = \frac{T}{\|T\|} \tag{17}$$

The next vector e_2 must be orthogonal to e_1 but is otherwise unconstrained. For e_2, choosing a direction orthogonal to the principal ray is a good solution. This is accomplished by using the cross product of e_1 with the direction of the principal ray and then normalizing so that we can get another vector:

$$e_2 = \frac{[-T_y T_x 0]}{\sqrt{T_x^2 + T_y^2}} \tag{18}$$

The third vector is just orthogonal to e_1 and e_2; it can be found using the cross product: $e_3 = e_1 \times e_2$

Our matrix that takes the epipole in the left camera to infinity is then:

$$R_{rect} = \begin{bmatrix} (e_1)^T \\ (e_2)^T \\ (e_3)^T \end{bmatrix} \tag{19}$$

This matrix rotates the left camera about the center of projection so that the epipolar lines become horizontal and the epipoles are at infinity. The row alignment of the two cameras is then achieved by setting:

$$R_l = R_{rect} r_l \ , \ R_r = R_{rect} r_r \tag{20}$$

Then We can also compute the rectified left and right camera matrices M_{rect_l} and M_{rect_r} but return them combined with projection matrices p_l and p_r:

$$P_l = M_{rect\_t} P_l' = \begin{bmatrix} f_{x\_f} & a_l & c_{x\_l} \\ 0 & f_{y\_f} & c_{y\_l} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{21}$$

and

$$P_r = M_{rect\_t} P_r' = \begin{bmatrix} f_{x\_f} & a_l & c_{x\_l} \\ 0 & f_{y\_f} & c_{y\_l} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{22}$$

The projection matrices take a 3D point in homogeneous coordinates to a 2D point in homogeneous coordinates as follows:

$$P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ w \end{bmatrix} \tag{23}$$

Where the screen coordinates can be calculated as $(x/w, y/w)$. Points in two dimensions can also then be reprojected into three dimensions given their screen coordinates and the camera intrinsic matrix. The reprojection matrix is:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (c_x - c_y)/T_x \end{bmatrix} \tag{24}$$

Here the parameters are from the left image except for c_x, which is the principal point x coordinate in the right image. If the principal rays intersect at infinity, then $c_x = c'_x$ and the term in the lower right corner is 0. Given a two-dimensional homogeneous point and its associated disparity d, we can project the point into three dimensions using:

$$Q\begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \tag{25}$$

Once we have our stereo calibration parameter, we can compute left and right rectification lookup maps for the left and right camera views separately. As with any image-to-image mapping method, a forward mapping (in which we just compute where pixels go from the source image to the destination image) will not, owing to floating-point destination locations, hit all the pixel locations in the destination image. So instead we work backward: for each integer pixel location in the destination image, we look up what floating-point coordinate it came from in the source image and then interpolate from its surrounding source pixels a value to use in that integer destination location. This source lookup typically uses bilinear interpolation. The process of rectification is illustrated in Figure 3 [17].

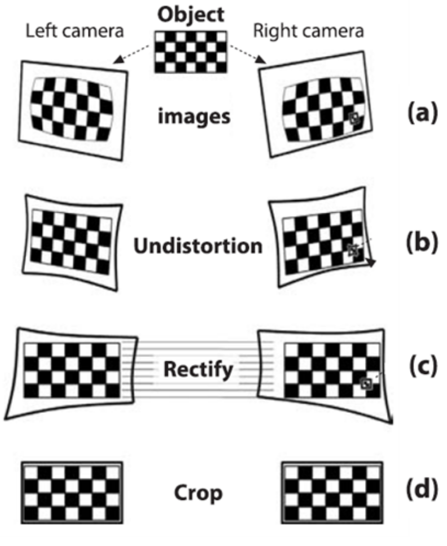

Figure 3. The process of stereo rectification

6. View Generation of In-between Views by view morphing

In this section, we explain the algorithm of in-between view synthesis for the object images. We used the view interpolation based view morphing [18] between neighboring cameras near the virtual viewpoint position generates virtual view images of object image for each frame real time.

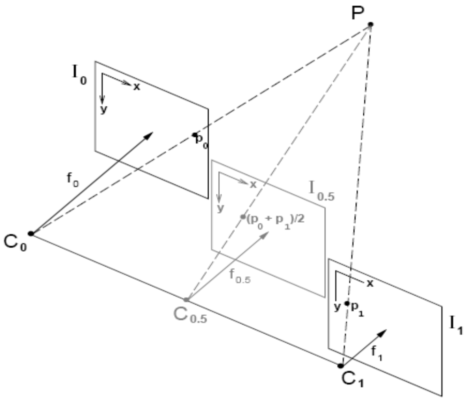

Figure 4. Morphing Parallel Views. Linear interpolation of corresponding pixels in parallel views

Suppose I_0 is a photograph of an object, and I_1 is a second picture, as shown in figure 4.

Alternatively, we could produce the same two images by moving the camera instead of the object. Linear image interpolation could produce new perspective views when the camera moves parallel to the image plane. Indeed, suppose that the camera is moved from the world origin to position $(C_X, C_Y, 0)$ and the focal length changes from f_0 to f_1. We write the respective projection matrices, $\prod_0$ and $\prod_1$, as:

$$\prod_0 = \begin{bmatrix} f_0 & 0 & 0 & 0 \\ 0 & f_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{26}$$

$$\prod_1 = \begin{bmatrix} f_1 & 0 & 0 & -f_1 C_X \\ 0 & f_1 & 0 & -f_1 C_Y \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{27}$$

Let $p_0 \in I_0$ and $p_0 \in I_1$ be projections of a scene point $P = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$. Linear interpolation of p_0 and p_1 yields:

$$(1-s)p_0 + sp_1 = (1-s)\frac{1}{Z}\prod\nolimits_0 P + s\frac{1}{Z}\prod\nolimits_0 P = \frac{1}{Z}\prod\nolimits_s P \tag{28}$$

Where

$$\prod\nolimits_s = (1-s)\prod\nolimits_0 + s\prod\nolimits_1 \tag{29}$$

Image interpolation therefore produces a new view whose projection matrix $\prod_s$, is a linear interpolation of $\prod_0$ and $\prod_1$, representing a camera with center C_s and focal length f_s given by:

$$C_s = (sC_X, sC_Y, 0)$$

$$f_s = (1-s)f_0 + sf_1 \qquad (30)$$

Consequently, interpolating images produced from parallel cameras produces the illusion of simultaneously moving the camera on the line $\overline{C_0C_1}$ between the two optical centers and zooming continuously. Because the image interpolation produces new views of the same object, it is shape-preserving.

To obtain an in-between stereoscopic view we firstly choose proper positions of the virtual stereo camera between two cameras. Firstly, We set two virtual stereo cameras C_l and C_r between the origins of the first camera C_0 and the second camera C_1. The positions of the virtual cameras are defined based on the distance between eyes respectively as shown in figure 5.

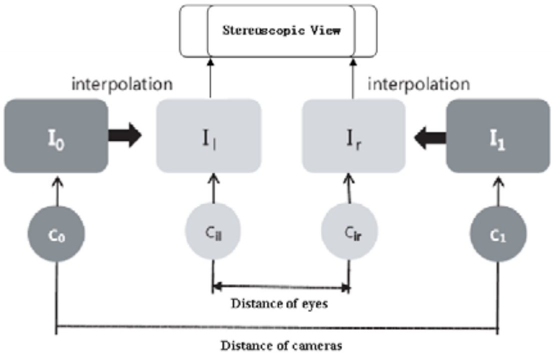

Figure 5. Synthesizing stereoscopic view based on view morphing

Once we choose the linear interpolation parameters, s_l and s_r, we can generate stereoscopic images, I_l and I_r, whose camera centers are C_l and C_r .Each coordinate of a point $p_l \in I_l$ and $p_r \in I_r$ can be defined by

$$p_{l|r} = (1-s_{l|r})p_0 + s_{l|r}p_1 \qquad (31)$$

Where p_0, p_1 are the corresponding points in I_0 and I_1, and $0 \le s_{l|r} \le 1$. For a convenience, we use '|' meaning 'or' , for example $p_{l|r}$ represents p_l or p_r .

Each pixel value of $I_{l|r}$ can be defined by forward mapping [19] using the I_0 and I_1, which yields $I_{l|r,0}$,and $I_{l|r,1}$ as

$$I_0(u,v) \to I_{l|r,0}((1-s_{l|r})u + s_{l|r}(u - d_0(u,v)), v)$$
$$I_1(u,v) \to I_{l|r,1}((1-s_{l|r})(u + d_1(u,v)) + s_{l|r}u, v) \qquad (32)$$

Where $d_{0|1}$ is a disparity map of the $I_{0|1}$.

In case of holes caused by disordered area in both I_0 and I_1 [20], linear interpolation of the neighborhood pixels can be used to define a pixel value of the area. In other cases, by refering two intermediate images $I_{l|r,0}$ and $I_{l|r}$ by (33), we can generate the stereoscopic images I_l and I_r which do not have any holes caused by disordered area from one view.

$$I_{l|r}(u,v) = \begin{cases} I_{l|r,0}(u,v) & (if\ I_{l|r,1}(u,v)\ is \text{ a hole}) \\ I_{l|r,1}(u,v) & (if\ I_{l|r,0}(u,v)\ is \text{ a hole}) \\ (1-s_{l|r})I_{l|r,0}(u,v) + s_{l|r}I_{l|r,1}(u,v) & \text{(otherwise)} \end{cases} \qquad (33)$$

7. EXPERIMENTAL RESULTS

For experimental verification, We have implemented a AV system for testing. Equipments include HMD (VR1280), two cameras, a conventional PC (with 2.2GHz CPU, 2GB RAM) were present. The users' hand that was placed in front of HMD has been integrated into an AV space. The captured videos are composed of 640*480 pixels, 24-bit-RGB color images. Our goal was to generate new virtual view of hand image which captured by cameras and then integrated them into the virtual environment to make an optimal viewpoint for user. The approach to segment the user's foreground hand from the background in the camera images was used the skin detection algorithmic [20]. The stereo hand images were showed in figure 6.

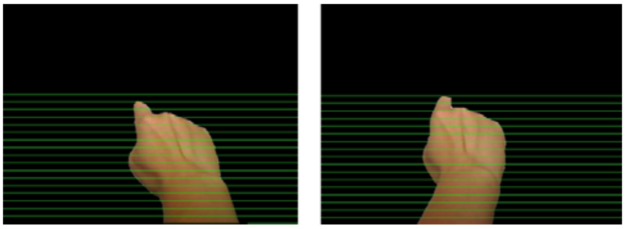

Figure 6. Hand images before calibration and rectification

Firstly, two cameras were calibrated and rectified to canonical configuration state that made the image planes alignment or corresponding points are on the same scan line. Figure 7 shows the source images and images after our process.

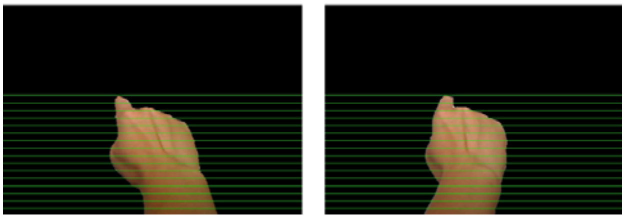

Figure 7. Hand images after calibration and rectification.

With the parallel images, arbitrary views between them could be generated as show in figure 8. There were six images of different view with the s from 0-1(The morphing time was about 20ms for each view).

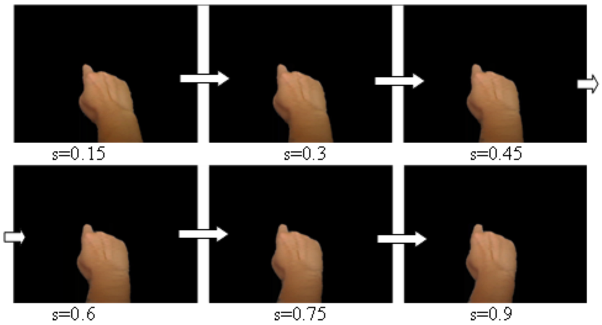

Figure 8. Arbitrary views between the pair cameras

In our AV system, the distance between cameras is about 160mm (the width of HMD), and the distance between people's eyes are about 65mm. We can compute the distance weights ($s_l = 0.3$, $s_r = 0.7$). So the left image and right image the user saw were as Fig 9 shows.

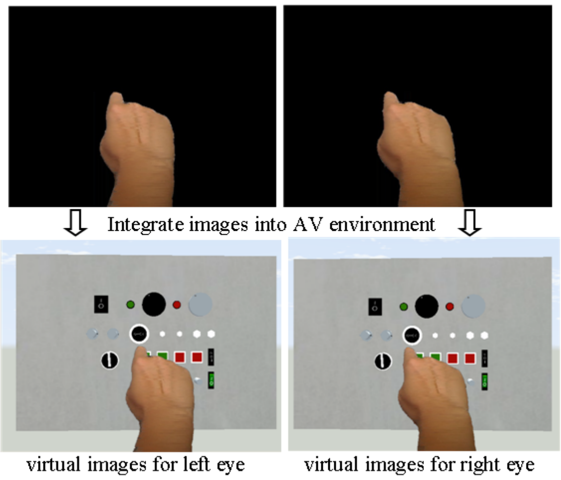
Integrate images into AV environment
virtual images for left eye
virtual images for right eye

Figure 9. Images that the user saw in HDM

Compared to the source image in figure 6, the results show that we can control the viewpoint of the stereoscopic view but also the screen parallax by choosing different eye separation(distance between eyes), which gives us different levels of depth perception.

8. CONCLUSIONS

This paper presents a novel method to generate novel views of an object that is placed in an AV environment real-time. Without 3D knowledge of the object, the novel view is generated by transforming live video frames captured by two fixed real cameras. The proposed method results in realistic novel views of the object corresponding to a virtual camera that have same position of user's eyes between two real cameras. Besides, the proposed method enables to provide any viewpoint between input views and the different level of depth perception by adjusting the distance between two virtual cameras.

Proposed method also has limitations. Firstly, since it relies on the dense correspondence between two input views, results tend to be sensitive to noisy. The other limitation is that information of the occluded area from both input images cannot be recovered resulting in holes. Therefore, appropriate hole filling algorithm for both disparity maps and input images should be supported.

8. ACKNOWLEDGMENT

The authors wish to thank anonymous reviewers for giving helpful suggestions. This work is supported by the National High Technology Research and Development Program of China under Grant No. 2007AA01Z306, and the National Nature Science Foundation of China under Grant No. 60776812.

REFERENCES

[1] A. M. HAO, B.HE, Q. P. ZHAO, "Review of augmented virtuality technology in virtual reality," Journal of Beijing University of Aeronautics and Astronautics, vol.29, no. 10, pp. 909-913, 2003

[2] B. Gerd; S. Frank; R. Kai; H. Klaus, "Enhancing Presence in Head-mounted Display Environments by Visual Body Feedback Using Head-mounted Cameras," Proc of 2009 International Conference on CyberWorlds, pp. 43-50 IEEE Computer Society, September, 2009.

[3] J. P. Roland, F. A. Biocca, T. Barlow, and A. Kancherla, "Quantification of adaptation to virtual-eye location in see-thru head-mounted displays," Proc of the Virtual Reality Annual International Symposium, pp. 56-66, IEEE Computer Society, 1995.

[4] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent Advances in Augmented Reality," IEEE Computer Graphics and Application, vol. 21, no. 6, pp. 34-47, Nov. 2001

[5] C. Eddie1, K, Peter, S, Thomas, "Multi-view synthesis: A novel view creation approach for free viewpoint video," Signal Processing: Image Communication, vol. 21, no. 6, pp.476-492, July, 2006

[6] S. M. Seitz and C. R. Dye, "View morphing," IEEE Transactions on Image Processing, vol. 6, no. 4, pp.584 - 598, April, 1997.

[7] H. Saito, S. Baba, and T. Kanade, "Appearance-basedvirtual view generation from multicamera videoscaptured in the 3-d room". IEEE Trans. onMultimedia, vol.5, pp.303–316, September 2003.

[8] S. M. Seitz and C. R. Dyer. "Photorealistic scenereconstruction by voxel coloring," Proc of. ComputerVision and Pattern Recognition, pp.1067–1073, January 1998.

[9] M. D. Wheeler, Y. Sato, and K. Ikeuchi. "Consensus surfaces for modeling 3d objects from multiple range images". DARPA Image Understanding Workshop, pp.917-923, 1997.

[10] S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. IEEE Trans. On Visualization and Computer Graphics, vol.4, pp.293–306, April, 1998.

[11] N. Inamoto and H. Saito. "Fly through view videogeneration of soccer scene". International Workshop on Entertainment Computing Workshop Note, pp.94–101, May 2002.

[12] N. Inamoto and H. Saito, "Intermediate view generation of soccer scene from multiple videos". Proc of International Conference on Pattern Recognition, pp.713–716, August 2002.

[13] I. Naho1, S. Hideo, "Free Viewpoint Video Synthesis and Presentation of Sporting Events for Mixed Reality

Entertainment," Proc of ACM International Conference Series, vol.74, pp.42-50, ACM Press, 2004.

[14] H. Chen, P. F. Elzer, "Novel View Generation for A Realtime Captured Video Object," Proceedings of the ACM symposium on Virtual reality software and technology, pp.83-86, 2006.

[15] Z.Y.Zhang. "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations," Proc of IEEE International Conference on Computer Vision, pp. 666-673, IEEE Computer Society Press, 1999.

[16] J.Y. Bouguet, “Camera calibration toolbox for Matlab,”

[17] G. Bradski and A. Kaehler, "Learning OpenCV" O'Reilly Media press, Sebastopol, 2008

[18] S. M. Seitz and Charles R. Dye, "View morphing," Proc of SIGGRAPH. pp.21-30, 1996.

[19] A. Criminis, A. Blake, C. Rother, J.Shotton: “Efficient dense stereowith occlusions for new view-synthesis by four-state dynamic programming”. International Journal of Computer Vision vol.71, pp. 89–110, 2007.

[20] S.E. Chen, L. William: View interpolation for image synthesis. In: Proc of 20th Annual Conference on Computer Graphics and Interactive Techniques, pp. 279–288. ACM Press, New York 1993.

[21] C. Tsai ,A. K. Katsaggelos, "Dense disparity estimation with a divide-andconquer disparity space image technique," IEEE Transactions on Multimedia, vol.1, no.1, pp.18-29, 1999

[22] Y. TANG, H. B GU, L. ZHOU, "Real-time Hand Segmentation and Fingertip Detection for Interaction," Opto-Electronic Engineering, vol.37 no.7, pp.145-150. 2010.

Yong Tang was born in Yiyang, Hunan province China in 1983. He is currently a Ph.D. candidate at Nanjing University of Aeronautics and Astronautics, Nanjing, China, and his research interests include virtual reality, image processing, etc.

HongBin Gu was born in Changzhou, Jiangshu province China in 1955 and received the doctor's degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China in 2000. He is currently an associate professor at Nanjing University of Aeronautics and Astronautics. His main research interest is include flight simulation, virtual reality, image processing, etc.

Lai Zhou was born in Jinhua, Zhejiang province China in 1983, and he is currently a Ph.D. candidate at Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include virtual reality, human computer interaction, etc.