



column vector  $\mathbf{P}=[X \ Y \ Z \ 1]^T$  and a Euclidean image point  $(u,v)$  by  $\mathbf{p}=[u \ v \ 1]^T$ . We reserve the  $\mathbf{P}$  and  $\mathbf{p}$  for points expressed in Euclidean coordinates, i.e., whose last coordinate is 1. Scalar multiples of these vectors will be written with a tilde, as  $\tilde{\mathbf{P}}$  and  $\tilde{\mathbf{p}}$ . A camera is represented by a  $3 \times 4$  homogeneous projection matrix of the form  $\mathbf{M}=[\mathbf{H}|\mathbf{C}]$ . The vector  $\mathbf{C}$  gives the Euclidean position of the camera's optical center and the  $3 \times 3$  matrix  $\mathbf{H}$  specifies the position and orientation of its image plane  $\mathbf{I}$  with respect to the world coordinate system. The perspective projection equation is

$$\tilde{\mathbf{p}} = \mathbf{M}\mathbf{P} \quad (2)$$

The term view will henceforth refer to the tuple  $(\mathbf{I},\mathbf{M})$  comprised of an image and its associated projection matrix[4].

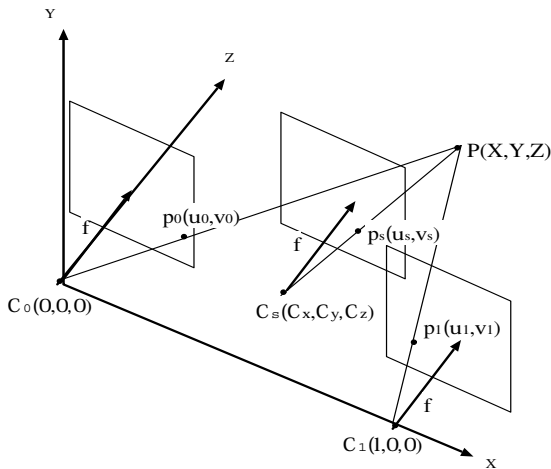
The view morphing method proposed on the base of high free degree space viewpoint in this paper can show the camera's epipolar geometry relations correctly. Suppose that the camera center  $C_i=[C_{ix},C_{iy},C_{iz}]^T$ . The projection matrices for each parallel view  $C_i$  can be represented as:

$$M_i = \begin{bmatrix} f_i & 0 & 0 & -f_i C_{ix} \\ 0 & f_i & 0 & -f_i C_{iy} \\ 0 & 0 & 1 & -C_{iz} \end{bmatrix}, \quad (3)$$

where  $f_i$  is focal length of  $C_i$ , and  $\hat{\mathbf{p}}_i = \mathbf{M}_i \mathbf{P} (i=0,1,s)$ .

$$\begin{cases} u_s = \frac{u_0 + \frac{C_x}{l}(u_0 - u_1)}{1 - \frac{C_z}{fl}(u_0 - u_1)} \\ v_s = \frac{v_0 + \frac{C_y}{l}(u_1 - u_0)}{1 - \frac{C_z}{fl}(u_0 - u_1)} \end{cases} \quad (4)$$

Therefore, in Fig. 2, we can get the relationship equation(4) from  $p_s(u_s, v_s)$  and  $p_0(u_0, v_0), p_1(u_1, v_1)$ , in which  $p_s(u_s, v_s)$  is the corresponding point of special point  $P(X,Y,Z)$  in the image of free view point



**Figure 2. Improved view morphing. Linear interpolation of corresponding pixels in parallel views  $C_0$  and  $C_1$ , representing another parallel view  $C_s$  of the same scene.**

$C_s(C_x, C_y, C_z)$ , at the same time,  $p_0(u_0, v_0)$  and  $p_1(u_1, v_1)$  are the corresponding points on images from  $C_0$  and  $C_1$  respectively. In the equation (4),  $l$  is the distance between  $C_0$  and  $C_1$ ,  $f$  is the focal length for each camera. The image with free view points is generated on this basis. The details are shown as following steps:

1. Set the enough number of corresponding points of the two input images.
2. Divide the images into triangle pieces by connecting points on each image.
3. According to the equation (5), calculate the corresponding coordinates of points and triangles by the parameters, such as  $(\alpha, \beta, \gamma)$ .

$$\begin{cases} u_s = \frac{(1-\alpha)u_0 + \alpha u_1}{1 + \gamma(u_1 - u_0)} \\ v_s = \frac{v_0 + \beta(u_1 - u_0)}{1 + \gamma(u_1 - u_0)} \end{cases} \quad (5)$$

In equation (5),  $(u_s, v_s)$  is the coordinates of free view points of image  $C_s$ , where special view point  $P$  locates, and  $(u_0, v_0), (u_1, v_1)$  are two input points of image  $C_0$  and  $C_1$  and  $v_0=v_1$ . The relationship among position parameters of free view points  $(\alpha, \beta, \gamma)$ , real coordinates of free view points  $C_s(C_x, C_y, C_z)$ , focal length of camera  $f$ , distance between input viewpoints  $l$  is shows as following:

$$\alpha = C_x/l, \beta = C_y/l, \gamma = C_z/lf. \quad (6)$$

4. Render triangles of the arbitrary viewpoint images with texture mapping method and produce the parallel viewpoint image. In this case, we use texture blending by using parameter  $\alpha$ .

## 4. Experimental Results

In order to improve the quality of the synthesized images, we point the points separately on objects and the background by. And then we use these points to generate the image of arbitrary views. As what is shown bellow, two cameras located at the input view points  $C_0, C_1$  in Fig. 4. The focal length is 400. The distance between cameras is 60cm, the height is 23.3cm and it is 150cm to



(a.) Camera  $C_0$

(b.) Camera  $C_1$

**Figure 3. Input view images.**

the object. And then, take the photos of the box (Fig. 3) as input image in the size of  $320 \times 240$ .

We point 15 pairs of points on the object and 12 pairs of points on the background. The results of the triangle dividing are shown in Fig. 5. When we do the texture mapping on the triangles, the pixel values of the arbitrary views images is depend on the rate parameter  $\alpha$  of the two corresponding points on the viewpoint input images.

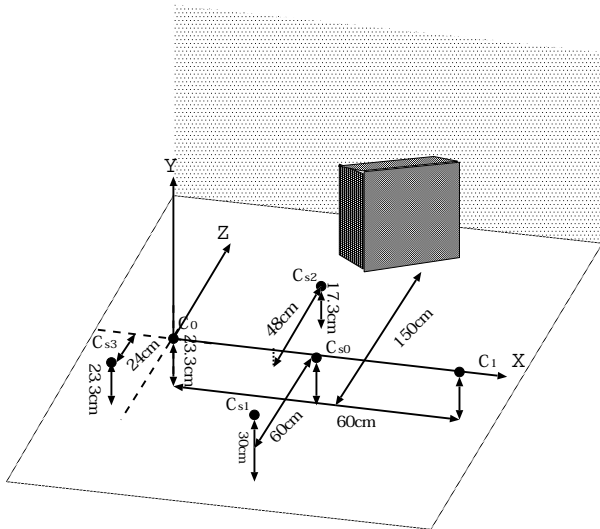


Figure 4. Experimental environment.

But if there are some points which can only be observed by one viewpoint, the result pixel value is decided by visible point directly. With the position parameters  $(\alpha, \beta, \gamma)$  of free view point, the generated image of the corresponding point of free view point  $C_{si}(i=0,1,2,3)$  is shown in Fig. 6. And under the same condition, the real corresponding point of free view point image are shown in Table 1.

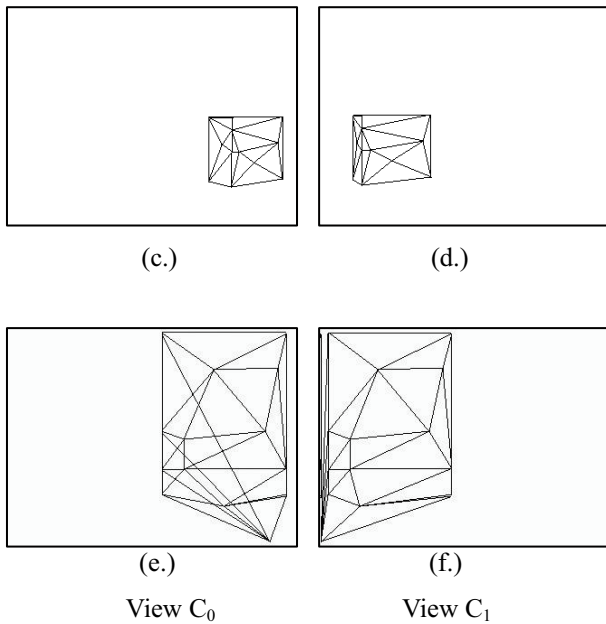


Figure 5. The triangle dividing. Top: the triangle patches of the object from viewpoints. Bottom: the triangle patches of the background from viewpoints.

Now we will evaluate the quality of the free viewpoint image.  $C_{si}$  in Fig. 6 are free viewpoint images. After calculating the position parameter  $(\alpha, \beta, \gamma)$  of  $C_{si}(i=0,1,2,3)$  in Fig. 6, we takes real image by the same parameter camera, as shown in Fig 7. Then we try to get the difference image between the generated image and the real image. Which has same camera position as the arbitrary view position. We calculate average difference and

Table 1. Synthesis arbitrary view point position.

Synthesis arbitrary views	$(C_x, C_y, C_z)$ [cm]	$(\alpha, \beta, \gamma)$
Cs0	(30,0,0)	(0.5,0,0)
Cs1	(30,7,-60)	(0.5,0.1167,0.0025)
Cs2	(18,-6,48)	(0.3,-0.1,0.002)
Cs3	(-6,0,-24)	(-0.1,0,-0.001)

standard deviation for histogram of difference images. Because of the error from the human being will happen, we should modify the result. When the average difference and standard deviation are the least, the corresponding point parameters are  $(\alpha', \beta', \gamma')$ , and the  $(C'_x, C'_y, C'_z)$  can be known. Tab.2 the exam, Fig. 7(m.) shows the difference image, Tab.2 shows the the average difference and standard deviation and the Fig.8 shows difference image histogram. Table 3 shows  $(C'_x, C'_y, C'_z)$  of  $(\alpha', \beta', \gamma')$ . Through compare, the difference between  $(C_x, C_y, C_z)$  and  $(C'_x, C'_y, C'_z)$ ,  $(\Delta C_x, \Delta C_y, \Delta C_z)$ , is (0.84 cm, 0.42 cm, 0.96 cm). The distance between synthesized image and the real image is less than 1cm, and the distance between the cameras is 60cm, so the error can be ignored.

In order to interpret the well result of this method more objectively, differences of 30 images are analyzed, those images are taken with the same NTSC cameras at different time. From Tab. 2 and the result shown in Fig 8, we can see that the the average difference is about 20 and standard deviation is less than 30. The free view point image generated by this method and the real image are almost the same in 256 gray scale levels, which shown as the grave histogram image of differences and the table of average difference and the standard deviation.

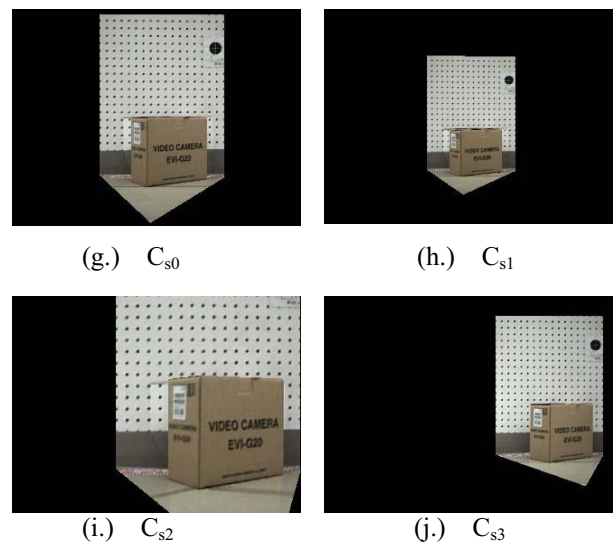
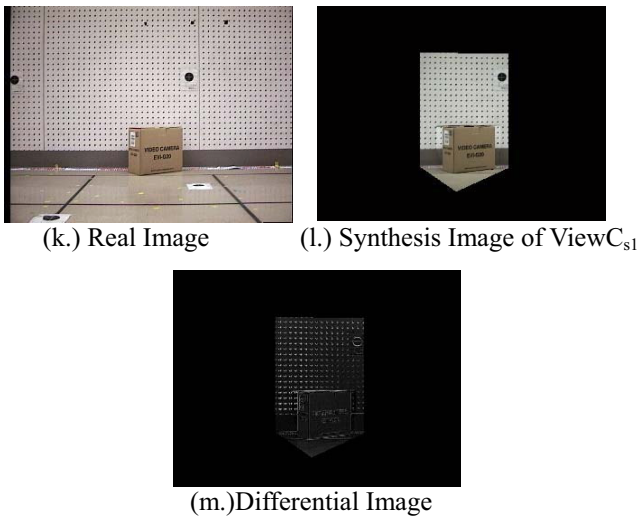


Figure 6. Synthesis image for arbitrary view points  $C_{si}$  in Fig 4. The view position parameters are in Tab 3.

**Table 2. Verification result of difference between synthesis arbitrary view and real image.**

Synthesis Views	Average Difference	Standard Deviation
$C_{s0}$	22.62	20.37
$C_{s1}$	14.31	25.95
$C_{s2}$	29.12	26.47
$C_{s3}$	26.27	23.53
NTSC	3.23	2.74



**Figure 7. Differential image between synthesis image and real image. Top: the right is synthesis arbitrary image of the view  $C_{s1}$ , the left is real image of corresponding view point. Bottom: the differential image between synthesis image and real image.**

**Table 3. Revised Synthesis Arbitrary Position.**

Real Views	$(\alpha', \beta', \gamma')$	$(C'_x, C'_y, C'_z)$ [cm]
$C'_{s0}$	(0.514, -0.007, 0.0)	(30.84, -0.42, 0.96)
$C'_{s1}$	(0.514, 0.1097, 0.00246)	(30.84, 6.582, -59.04)
$C'_{s2}$	(0.314, -0.1069, 0.00204)	(18.84, -6.414, 48.96)
$C'_{s3}$	(-0.086, -0.0072, -0.00096)	(-5.16, -0.432, -23.14)

## 5. Conclusions

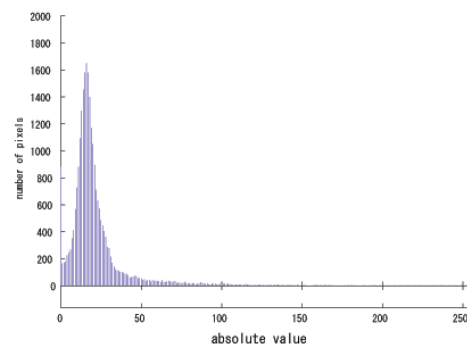
In this paper, the View Morphing method is been expanded, the method of synthesizing arbitrary views from

images with two parallel view points has been proposed. With this method, the quality of free view point image can be maintained and the scale of free view point of generating method can be expanded to 3D space with the less information of camera.

Though the error will occur because of the illumination condition, the point of the corresponding points and the triangle dividing, through the experiments, it shows that the method of morphing a free viewpoint image in the high free degree space based on two parallel viewpoint images is effective. It allows more flexible viewpoint setting from the line between two viewpoints into larger space. The experiment result confirms effectiveness of our method. In order to enlarge the area of the view point, we should do the research about the unparallel inputting viewpoint images. Meanwhile, we also will improve the method of the triangle dividing to improve the quality of the result.

## References

- [1] Steven.M.Seitz and Charles R. Dyer, "View Morphing", Proceedings of SIGGRAPH'96, pp.21-31, 1996.
- [2] Jiangjian Xiao and Mubarak Shah, "From Images to Video: View Morphing of Three Images", Conference on Vision, Modeling and Visualization, November 19-21, Munich, Germany, 2003.13.
- [3] Makoto Kimura and Hideo Saito, "View Morphing Based on Projective Geometry Three Cameras", THE JOURNAL OF THE INSTITUTE OF IMAGE INFORMATION AND TELEVISION ENGINEERS, Vol.57, No.10, pp.1342-1346 (2003).
- [4] Koichiro Deguchi, "Fundamental of Robot Vision", CORONA PUBLISHING CO.,LTD. pp.12-55, 2000.
- [5] S. Avidan, and A. Shashua. "Novel View Synthesis by Cascading Trilinear Tensors", IEEE Transactions on Visualization and Computer Graphics, 293-306, 1998.
- [6] R. Hartley, and A. Zisserman. "Multiple View Geometry in Computer Vision", Cambridge University Press, 2000.
- [7] CHEN, S. E., AND WILLIAMS, L. View interpolation for image synthesis. Proc. SIGGRAPH 93. In *Computer Graphics* (1993), pp. 279-288.



**Figure 8. Differential image histogram of valid region in Fig7.(m).**