

# Generalizing the Virtual Camera Pose for View Synthesis

Enric X. Martín and Antonio B. Martínez

Department of System Engineering. Polytechnic University of Catalonia.  
Edifici U. C/ Pau Gargallo, 5. 08024 Barcelona, Spain.  
{emartin,abmv}@esaii.upc.es

**Abstract.** View synthesis requires the ability to estimate the image projected from a scene to a point of view where has not been placed a real camera. Many methods have been developed and three-view rectification is one of the most used, nevertheless it has some restrictions. That arises when the plane containing the focus of the three cameras involved in the process is parallel to the view direction of any of the cameras. This paper deals with the geometry of the method and gives analytically a way for dodging the singularities in the position of the virtual camera. That allows us to obtain a synthetic view from a previously forbidden point and automate the process towards fast software or hardware implementations.

## 1 Introduction

Excellent expectations in fields like remote visualisation of information, navigation through far environments, telerobotics, teleconferencing and realistic gaming, are promoting the interest of researchers in view synthesis. In the field of computer vision and computer graphics, the birth of these applications has forced the development of techniques to synthesise realistic images with physical coherence with the scene.

Since the publication of the classical Chen and Williams approach [1] in 1993, several groups are obtaining encouraging results applying different techniques. Some researchers try to previously recover the 3D structure of the scene. T. Kanade et al. are using multiple-view stereo matching techniques [2], INRIA researchers (O. Faugeras, S. Laveau) are using mathematical approaches [3] [4] and others (Seitz, Kutulakos) are creating plenoptic models from multiple images [5] [6].

Meanwhile, there is a different way to work in view synthesis, without explicitly recover the 3D structure of the scene. Authors like T. Naemura et al. [7]; A. Katayama et al. [8] are developing interpolation techniques able to synthesize new views, although dealing directly with the 2D images. In order to extend the feasible poses for the virtual camera, the three-view rectification method allows the synthesis of new views from any point of view around the existing cameras. This method extensively explained by D. Scharstein [9] in Springer Verlag's "View Synthesis using Stereo Vision", has some mathematical singularities in the virtual camera pose.

As researchers specialised in system engineering, our motivation is to automate the generation of new views in real time. So, our group have chosen the methods that

deal directly with the images, because they are easier to translate into DSP or FPGA. With this aim, we propose in this paper some modifications of the three-view rectification method to avoid their singularities. The modified method will be applied to view synthesis in industrial environments [10] and augmented reality for entertainment computing [11].

## 2 The three-view rectification method

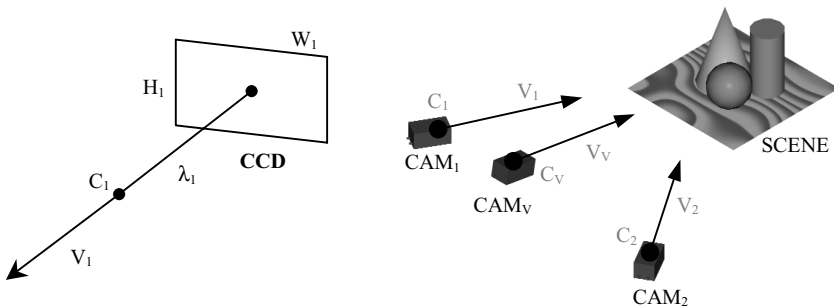
In view interpolation techniques the virtual camera is always placed along the baseline defined by two real cameras. Three-view rectification will extend the virtual camera poses around the real cameras. Obviously, to be able to synthesize some image pixels, the virtual camera must be looking at the stereo-zone defined by the two real cameras, and this stereo-zone must exist.

### 2.1 Definitions

We assume that we have built a fully calibrated system, where we know the position and orientation of two real cameras  $CAM_1$  and  $CAM_2$ . So, we know  $C_1$  and  $C_2$ , the optical centres (or focuses) of the cameras, and we know  $V_1$  and  $V_2$ , the camera vectors between each optical centre and the centre of the camera CCD. Besides, we assume that  $\lambda_1$  and  $\lambda_2$  are the focal distances for each camera. We define  $W_1$ ,  $H_1$  and  $W_2$ ,  $H_2$  as the dimensions of each camera CCD (width and height, respectively). Finally, we suppose that  $IM_1$  and  $IM_2$  are the images captured by  $CAM_1$  and  $CAM_2$ . Thus,  $IM_1$  and  $IM_2$  will be two pixel arrays of dimensions  $W_1$ ,  $H_1$  and  $W_2$ ,  $H_2$ .

Our goal will be to obtain the best image for a virtual camera,  $CAM_V$ , placed in a point  $C_V$  with orientation  $V_V$ . The virtual camera will have a focal distance  $\lambda_V$ ; CCD dimensions  $W_V$ ,  $H_V$ , and will capture an image named  $IM_V$ .

Figure 1, left, shows the geometric meaning of defined points, vectors and camera parameters. Figure 1, right, shows a scheme of the working environment. Definition 1, (next page) summarizes defined variables.



**Fig. 1.** Description of the camera parameters (left) and overview of the three cameras involved in the process (right).

$$C_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}, \quad C_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}, \quad C_V = \begin{bmatrix} x_V \\ y_V \\ z_V \end{bmatrix}, \quad V_1 = \begin{bmatrix} v_{x_1} \\ v_{y_1} \\ v_{z_1} \end{bmatrix}, \quad V_2 = \begin{bmatrix} v_{x_2} \\ v_{y_2} \\ v_{z_2} \end{bmatrix}, \quad V_V = \begin{bmatrix} v_{x_V} \\ v_{y_V} \\ v_{z_V} \end{bmatrix} \quad (\text{Def.})$$

## 2.2 How the method works

The three-view rectification method is based on the reprojection of the three images:  $IM_1$ ,  $IM_2$  and  $IM_V$  into a plane  $P_R$  parallel to the plane that contains the three camera centres:  $C_1$ ,  $C_2$  and  $C_V$ . Let  $P$  be the plane containing the three camera centres. It will be calculated as follows:

a)  $V_B$  is defined as the vector baseline between  $C_1$  and  $C_2$ , and  $V_{AUX}$  is defined as an auxiliary vector, connecting  $C_1$  and  $C_V$ .

$$V_B = C_2 - C_1 = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix}, \quad V_{AUX} = C_V - C_1 = \begin{bmatrix} x_V - x_1 \\ y_V - y_1 \\ z_V - z_1 \end{bmatrix} \quad (\text{Def. 2})$$

b)  $V_P$  is the vector perpendicular to plane  $P$  containing the three centres.

$$V_P = V_B \times V_{AUX} = \begin{vmatrix} \hat{x}_u & \hat{y}_u & \hat{z}_u \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_V - x_1 & y_V - y_1 & z_V - z_1 \end{vmatrix} = \begin{bmatrix} V_{PX} \\ V_{PY} \\ V_{PZ} \end{bmatrix} \quad (\text{Eq. 1})$$

c) Now, is possible to find the plane  $P$  having the equation of definition of a plane with the vector  $V_P$  and to solve the parameter  $K$  for any of the camera centres.

$$P: \quad X \cdot V_{PX} + Y \cdot V_{PY} + Z \cdot V_{PZ} + K = 0 \quad (\text{Eq. 2})$$

Next step, is to calculate the reprojection plane  $P_R$ . As a parallel plane to  $P$ , it will have the same equation (Eq. 2), with a different value of  $K$  that will be appointed  $K_R$ . So, equation 2 will be solved for a point placed at distance  $d$  from the plane  $P$ . This point will be named  $C_P$  and will allow the calculation of  $P_R$ .

$$C_P = C_1 + d \cdot \frac{V_P}{\|V_P\|} \rightarrow P_R: \quad X \cdot V_{PX} + Y \cdot V_{PY} + Z \cdot V_{PZ} + K_R = 0 \quad (\text{Eq. 3})$$

Now, is possible to take the images  $IM_1$ ,  $IM_2$  and  $IM_V$  and project them to the plane  $P_R$ . This can be done by intersecting the projection cones of each camera with  $P_R$ , obtaining the new images  $IM_1'$ ,  $IM_2'$  and  $IM_V'$ . Thus, these images will lay in the same plane, and all the cameras centres will have the same distance to the plane:  $d$ . If  $d$  is treated as the new focal distance for the cameras and the plane  $P_R$  is treated as a plane containing the three camera CCD, the epipolar geometry conditions will be reached. Figure 2 shows the meaning of  $P$ ,  $P_R$  and the reprojected images.

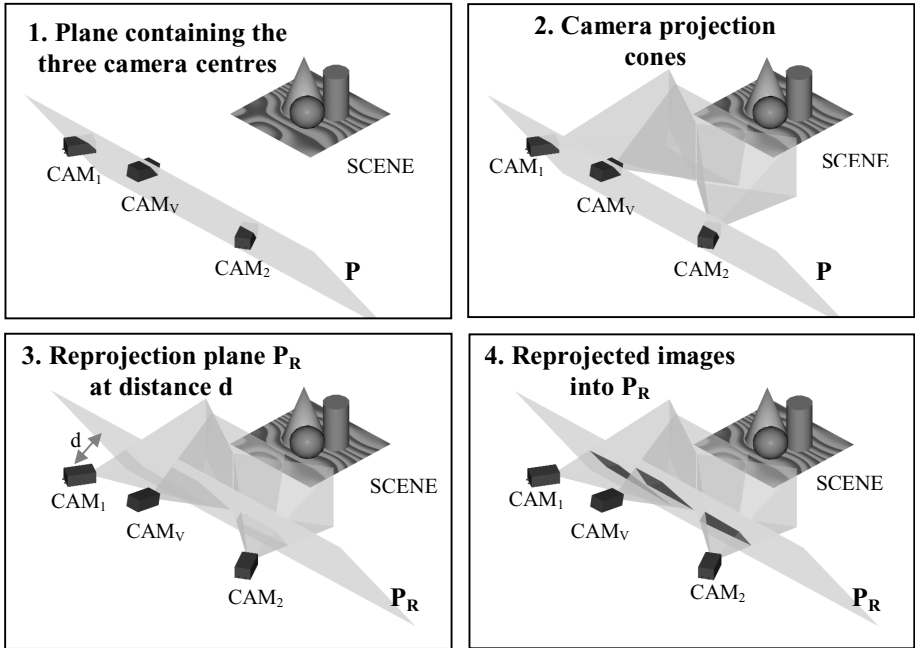


Fig. 2. Geometric representation of three-view rectification method steps.

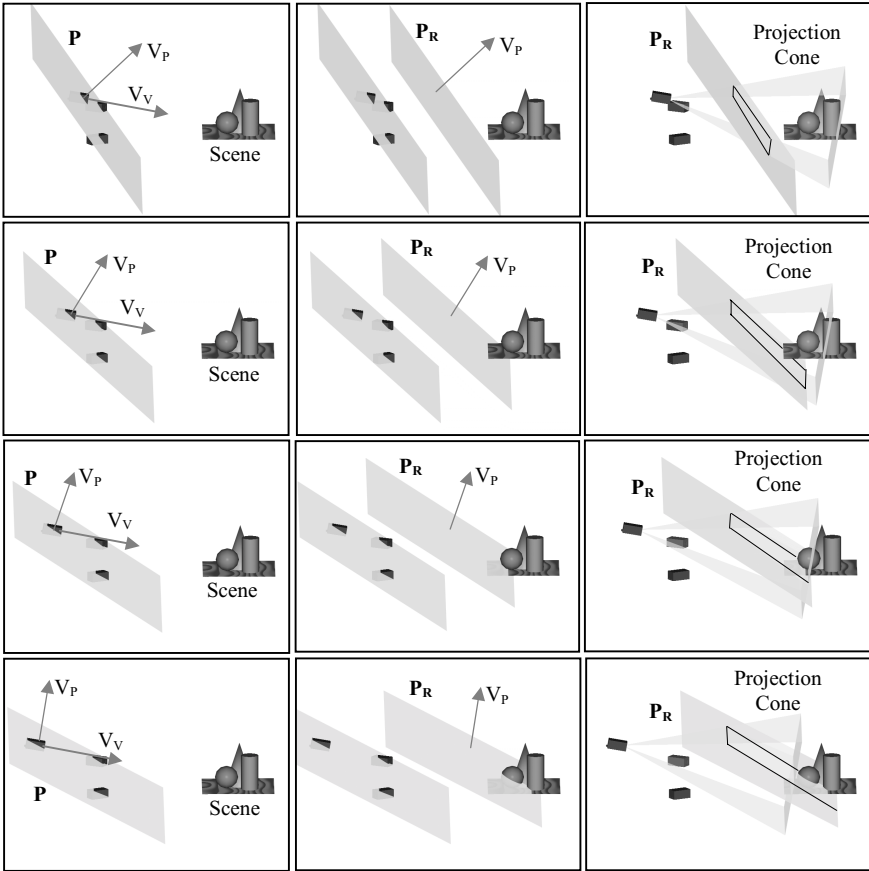
### 3 The reprojection plane problem

The presented method has a problem related to how much perpendicular are the three camera vectors to the reprojected plane. If the reprojected plane is parallel to any of the camera vectors the respective image can't be projected because the projection surface will be infinite. This problem can be originated by two reasons:

- 1) One of the cameras has a view direction quasi-perpendicular to the others.
- 2) The three camera centres lay in a plane that is parallel to the view directions of the cameras.

In the first case there is an unsolvable problem. Is impossible to generate a virtual view if the stereo-zone doesn't exists, or if the virtual camera is not looking to the stereo-zone.

But in the second case, the problem is due to a limitation of the method, as we show in figure 3. Each row presents how the algorithm calculates the projection of the images in the reprojected plane in three steps. First column shows the cameras and the plane containing the three focuses. The second column shows the reprojected plane  $P_R$  and the vector  $V_p$ . Finally, the third column shows the projection cone for the virtual camera, the reprojected plane, and their intersection. Notice how the intersection grows when the vector  $V_v$  is quasi-parallel to the tri-focal plane.



**Fig. 3.** Representation in a cartoon-like sequence, of the impossibility of calculating the intersection between the projection plane  $P_R$  and the projection cone of the virtual camera if the view direction of the camera and the projection plane are quasi-parallel.

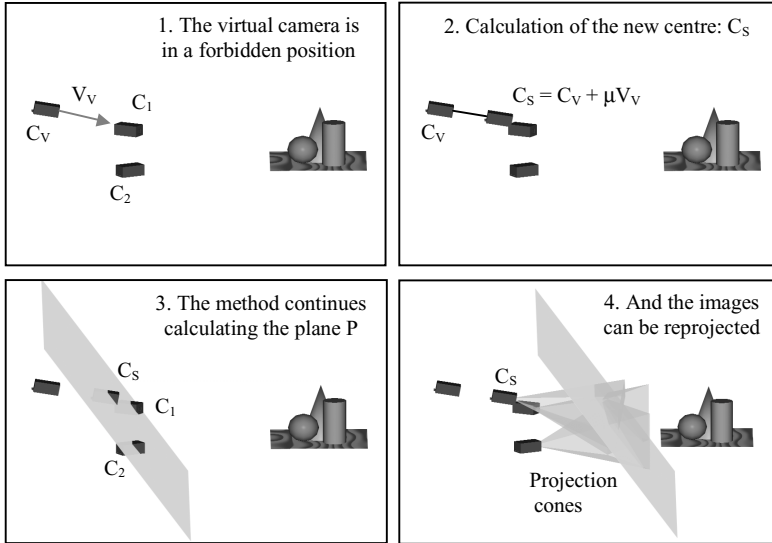
### 4 Modifications on the method

The presented problem can be solved by giving a degree of freedom to the position of the virtual camera. So, a new point  $C_S$  will be considered as the new camera centre instead of  $C_V$ . The new virtual camera centre  $C_S$  will be defined as follows:

$$C_S = C_V + \mu \cdot V_V \tag{Eq. 4}$$

A parameter  $\mu$  has been entered to represent this degree of freedom. So, the virtual camera will be moved on its view direction, forward and backward, allowing finding a reprojection plane able to contain the three projected images. Figure 4 shows how this idea can solve the problem exposed on previous chapter. The applied

modification to the method implies that the virtual image will be calculated for the new point of view  $C_S$ . The virtual view for  $C_V$  can be obtained applying a scaling matrix to the image.



**Fig. 4.** When the virtual camera is in a singular place, a new camera position is calculated, and then the three-view rectification method follows their ordinary formulation.

Now is time to find the optimum value for the parameter  $\mu$ . This value must maximize the perpendicularity between the plane  $P$  and the vector  $V_V$ . To find it, the general equation of the plane containing  $C_1$ ,  $C_2$  and  $C_S$ , must be calculated as described previously in the equations 1 and 2, as a function of the vector  $V_p$ :

$$V_B = C_2 - C_1 = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix}, \quad V_{AUX} = C_S - C_1 = \begin{bmatrix} (x_V + v_{xV} \cdot \mu) - x_1 \\ (y_V + v_{yV} \cdot \mu) - y_1 \\ (z_V + v_{zV} \cdot \mu) - z_1 \end{bmatrix}$$

$$V_P = V_B \times V_{AUX} = \begin{vmatrix} \hat{x}_u & \hat{y}_u & \hat{z}_u \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ (x_V + v_{xV} \cdot \mu) - x_1 & (y_V + v_{yV} \cdot \mu) - y_1 & (z_V + v_{zV} \cdot \mu) - z_1 \end{vmatrix} = \begin{bmatrix} V_{PX} \\ V_{PY} \\ V_{PZ} \end{bmatrix}$$

$$\begin{aligned} V_{PX} &= (z_1 v_y + z_2 y_1 + y_2 z_v - y_1 z_v - z_2 y_v - y_2 z_1) + \mu \cdot (y_2 v_{zV} + z_1 v_{yV} - y_1 v_{zV} - z_2 v_{yV}) \\ V_{PY} &= (x_1 z_v + x_2 z_1 + z_2 x_v - z_1 x_v - x_2 z_v - z_2 x_1) + \mu \cdot (z_2 v_{xV} + x_1 v_{zV} - z_1 v_{xV} - x_2 v_{zV}) \\ V_{PZ} &= (y_1 x_v + y_2 x_1 + x_2 y_v - x_1 y_v - y_2 x_v - x_2 y_1) + \mu \cdot (x_2 v_{yV} + y_1 v_{xV} - x_1 v_{yV} - y_2 v_{xV}) \end{aligned} \quad (\text{Eq. 5})$$

Thus, vector  $V_p$  is parameterised by  $\mu$ ; a function  $V_p(\mu)$  has been obtained. To obtain a good reprojection plane, is needed that the vectors  $V_1$ ,  $V_2$  and  $V_V$  are the most similar to  $V_p(\mu)$ . So, the angle between  $V_p(\mu)$  and these vectors can be used as a measure of similarity between them. The unitary vectors  $V_{1u}$ ,  $V_{2u}$ ,  $V_{Vu}$  and  $V_{pu}(\mu)$  are

obtained from  $V_1$ ,  $V_2$ ,  $V_V$  and  $V_P(\mu)$  respectively. Multiplying the camera vectors by  $V_P(\mu)$  the results obtained are:

$$V_{1u} \cdot V_{Pu}(\mu) = \cos(\alpha_1) \quad ; \quad V_{2u} \cdot V_{Pu}(\mu) = \cos(\alpha_2) \quad ; \quad V_{Vu} \cdot V_{Pu}(\mu) = \cos(\alpha_V) \quad (\text{Eq. 6})$$

Where  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_V$  are the angles between vectors  $V_1$ ,  $V_2$ ,  $V_V$  and the vector perpendicular to the reprojection plane,  $V_P(\mu)$ . As these angles must be near to zero, that implies that their cosine must be near to one. So, the function  $F$  of  $\mu$  defined as:

$$F := \cos(\alpha_1) + \cos(\alpha_2) + \cos(\alpha_3) \quad (\text{Eq. 7})$$

Must be maximized. Calculating the derivative of  $F$  by the variable  $\mu$ , a single maximum is obtained with the value  $\mu_{MAX}$ :

$$\mu_{MAX} = \frac{NUM}{DEN}; \quad \text{where } NUM \text{ and } DEN \text{ are :} \quad (\text{Result 1})$$

$$\begin{aligned} NUM &= C_0^2(A_1A_2 + B_1B_2) + B_0^2(A_1A_2 + C_1C_2) + A_0^2(B_1B_2 + C_1C_2) - \\ &A_0C_0(C_1A_2 + A_1C_2) - B_0C_0(B_1C_2 + C_1B_2) - A_0B_0(B_1A_2 + A_1B_2) \\ DEN &= C_1^2(A_0A_2 + B_0B_2) + B_1^2(A_0A_2 + C_0C_2) + A_1^2(B_0B_2 + C_0C_2) - \\ &A_0A_1(B_1B_2 + C_1C_2) - B_0B_1(A_1A_2 + C_1C_2) - C_0C_1(A_1A_2 + B_1B_2) \end{aligned}$$

$$\begin{aligned} A_0 &= (z_1y_V + z_2y_1 + y_2z_V - y_1z_V - z_2y_V - y_2z_1), & A_1 &= (y_2v_{zV} + z_1v_{yV} - y_1v_{zV} - z_2v_{yV}), \\ B_0 &= (x_1z_V + x_2z_1 + z_2x_V - z_1x_V - x_2z_V - z_2x_1), & B_1 &= (z_2v_{xV} + x_1v_{zV} - z_1v_{xV} - x_2v_{zV}), \\ C_0 &= (y_1x_V + y_2x_1 + x_2y_V - x_1y_V - y_2x_V - x_2y_1), & C_1 &= (x_2v_{yV} + y_1v_{xV} - x_1v_{yV} - y_2v_{xV}) \end{aligned}$$

$$\begin{aligned} A_V &= \frac{x_1}{\sqrt{x_1^2 + y_1^2 + z_1^2}} + \frac{x_2}{\sqrt{x_2^2 + y_2^2 + z_2^2}} + \frac{x_V}{\sqrt{x_V^2 + y_V^2 + z_V^2}}, \\ B_V &= \frac{y_1}{\sqrt{x_1^2 + y_1^2 + z_1^2}} + \frac{y_2}{\sqrt{x_2^2 + y_2^2 + z_2^2}} + \frac{y_V}{\sqrt{x_V^2 + y_V^2 + z_V^2}}, \\ C_V &= \frac{z_1}{\sqrt{x_1^2 + y_1^2 + z_1^2}} + \frac{z_2}{\sqrt{x_2^2 + y_2^2 + z_2^2}} + \frac{z_V}{\sqrt{x_V^2 + y_V^2 + z_V^2}} \end{aligned}$$

The virtual camera shifting, by  $\mu_{MAX}$  works properly and solves the method singularities. But, when the virtual camera vector belongs to the three-focal plane  $P$ , shifting is not enough to avoid the singularity, and our applied solution is to directly calculate  $P_R$  as a perpendicular plane to  $V_P$ . In all the other configurations the shifting solution is better, because it maximizes the projection of the cameras on  $P_R$ .

## 5 Conclusions and further work

We have proposed a modification of the three-view rectification method. First, we have showed how the images of the real cameras and the desired image of the virtual

camera must be reprojected on a plane to achieve the epipolar geometry conditions. Also, we have explained that this method has some limitations in the position of the virtual cameras, and have presented a solution that slightly modifies the position of the virtual camera to dodge these forbidden positions. The reached solution can be integrated in the ordinary implementation of the algorithm. As an analytical expression have been found, it will make easy the translation of the whole algorithm into hardware through DSP or FPGA components.

In the future, we will work in the analysis of the correctness of the synthesised images, and how variations in the values of other parameters in the implementation of the method can increase the amount of virtual image pixels that we are able to synthesize. Lastly, we will study the design of specified hardware architecture towards a real time implementation.

## References

1. Chen S.E. and Williams, L. View interpolation for image synthesis. In *Computer Graphics (SIGGRAPH'93)* pages 279-288. 1993.
2. Takeo Kanade, Research Group in Computer Science, Carnegie Mellon Univ., Virtualized Reality Home Page: <http://www.cs.cmu.edu/~virtualized-reality/>
3. Stéphane Laveau, Olivier Faugeras, Oriented projective geometry for Computer Vision, Tech. Report INRIA Sophia-Antipolis, 1997.
4. Faugeras O. and Keriven R. Variational principles, surface evolution, pde's, level set methods and the stereo problem. *IEEE Trans. Image Processing* 7, pages 336-344. 1998.
5. Seitz S. and Kutulakos, K. Plenoptic Image Editing. *International Journal on Computer Vision* 48, pages 115-119, Kluwer academic publishers, 2002.
6. Kutulakos K. and Seitz S. A theory of Shape by Space Carving. *International Journal on Computer Vision* 38, pages 199-218. Kluwer academic publishers, 2000.
7. Naemura T., Harashima H.. *The Ray-Based Approach to Augmented Spatial Communication and Mixed Reality. Mixed Reality, merging Real and Virtual Worlds.* Edited by Y. Ohta, H. Tamura, Ed. Ohmsha. Tokyo 1999.
8. Katayama K., Tanaka K., Oshino T., Tamura H.; Media Technology Laboratory, Canon Inc. Kawasaki, Japan. A viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images. *Proc. SPIE. Stereoscopic Displays and Virtual Reality Systems II.* Vol. 2049, pp. 11-20,1995.
9. Scharstein, Daniel. *View Synthesis Using Stereo Vision.* Ed. Springer-Verlag, Berlin 1999. ISSN 0302-9743.
10. Martín E.X. and Martínez A.B.. Generation of synthetic views for teleoperation in industrial processes. *Proceedings of the IEEE International Conference on Factory Automation. ETFA'01.* Sophia-Antipolis, October 2001.
11. Martínez A. B., Arboleda J.P., Martín E.X.. *Mixed Reality in traffic scenes.* Entertainment Computing, Technologies and Applications. Chapter 5. Kluwer Academic Publishers. 2003.