Projection-Based Registration Using Color and Texture Information for Virtual Environment Generation

Sehwan Kim, Kiyoung Kim, and Woontack Woo

GIST U-VR Lab.
Gwangju 500-712, S.Korea
{skim,kkim,wwoo}@gist.ac.kr

Abstract. In this paper, we propose a registration method for 3D data which uses color and texture data acquired from a multi-view camera for virtual environment generation. In general, most registration methods depend on 3D data acquired by precise optical sensors. However, as for a multi-view camera, depth errors are relatively large and depths in homogeneous areas are not measurable. We propose a projection-based registration method to cope with these limitations. First, we perform *initial registration* by establishing relationship between multi-view cameras through inter-camera calibration. Then, by applying color and texture descriptors to projected images, *fine registration* is accomplished. Finally, by exploiting adaptive search ranges, *color selection* is attained. Even if the accuracy of 3D data is relatively low, the proposed method can effectively register 3D data. In addition, an effective color selection can be done by setting up adaptive search ranges based on depth. Through this method, we can generate a virtual environment that supports user interaction or navigation.

1 Introduction

Modeling of real environment plays a vital role in various virtual reality applications. Image-based virtual reality systems (IBVR) are gaining popularity in computer graphics as well as computer vision communities. The reason is that they provide more realism by using photo-realistic images and modeling procedure is rather simple. For generating realistic models, accurate registration of acquired 3D data is essential.

Until now, various methods for object modeling have been proposed. ICP (Iterative Closest Point) algorithm has been widely used [1]. Johnson proposed Color ICP to reconstruct indoor environment [2]. On the other hand, Levoy et al. registered 3D data of several statues, obtained from range scanners, by utilizing a volumetric approach [3]. A registration method for multiple range images using an M-estimator was also proposed [4]. Pulli proposed a projective registration method that employs planar perspective warping [5] [6]. Invariant features were used to improve ICP [7]. On the other hand, Fisher applied projective ICP to Augmented Reality (AR) applications [8]. However, most methods

K. Aizawa, Y. Nakamura, and S. Satoh (Eds.): PCM 2004, LNCS 3331, pp. 434–443, 2004.

depend on expensive equipment, and require substantial time for generating 3D models. Cameras are usually used for modeling small objects in a short distance. Furthermore, effective registration is very difficult when 3D depth data includes relatively large errors.

In this paper, to remedy the above-mentioned problems, we propose a registration method based on color and texture information acquired from a multi-view camera for virtual environment generation. First, in *initial registration step*, we get initial pose relationship between cameras by inter-camera calibration. Second, in *fine registration step*, we project 3D data acquired from each camera onto a destination camera. Then, we find an optimized transformation matrix based on color and texture information by exploiting Levenberg-Marquardt algorithm. Finally, we determine adaptive search ranges in a *color selection step* and select the most suitable color.

In general, registration methods depend on very precise range scanners. However, it is expected that off-the-shelf multi-view cameras will soon be popular. The proposed method employs multi-view cameras whose depth errors are relatively large for a middle-range distance. However, we can generate a virtual environment conveniently by moving multi-view cameras. Furthermore, adaptive search ranges enable effective color selection. Although it has a disadvantage in terms of higher computational complexity than ICP, it provides better visual quality.

The paper is organized as follows. In chapter 2, we explain the projection-based registration method for VE generation. After experimental results are analyzed in chapter 3, conclusions and future work are presented in chapter 4.

2 Projection-Based Registration for Virtual Environment

The conventional ICP, based on the shortest distance, is not appropriate for registration of 3D data acquired from a multi-view camera due to its inherent depth errors. Thus, we propose a projection-based registration to carry out a pairing process effectively. Fig. 1 shows a flowchart of the proposed method. Fig. 1(a) depicts the overall procedure, and Fig. 1(b) illustrates only the projection-based registration process.

2.1 Preprocessing for Noise Removal

We assume that surface of the whole scene is Lambertian. However, we observe that some parts have very large variations in depth values even in a static scene. Thus, we must exclude unstable parts. In disparity map of a static scene, the variations of disparity values are modeled as Gaussian distributions. Thus, the threshold value for pixel i is determined by function of standard deviation of each pixel for excluding the unstable areas. A pixel is excluded if the following condition is satisfied.

$$Th_i(d) > \lambda\sigma_i(d) \tag{1}$$

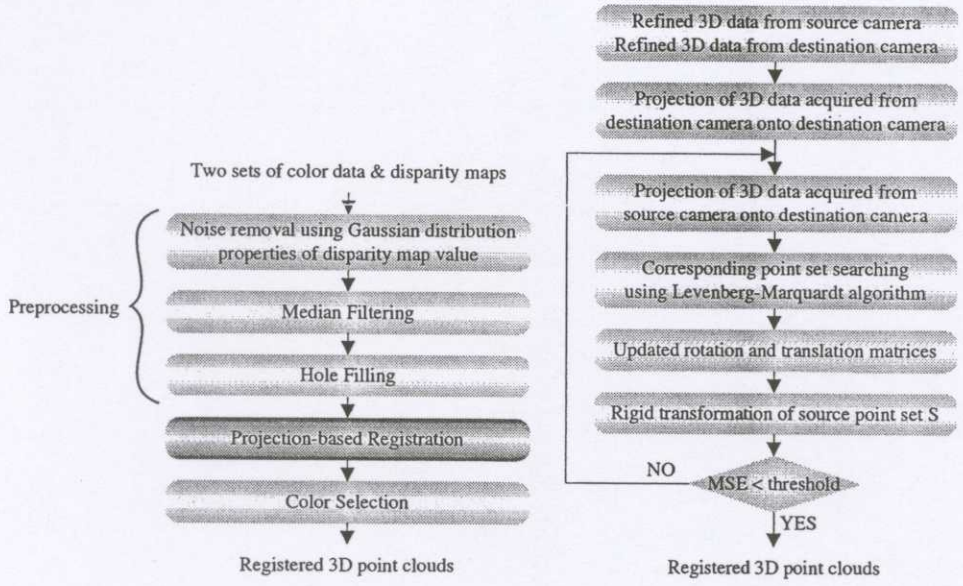


Fig. 1. Flowchart for projection-based registration (a) overall procedure for projection-based registration (b) projection-based registration part of overall procedure

where d denotes disparity and $\sigma_i(d)$ represents standard deviation. Scale factor λ must be decided empirically. $Th_i(d)$ is the threshold value for pixel i. Then, Median filter is applied to remove spot noises.

As a final step, hole filling is required on homogeneous areas or holes generated in the above step. We fill in the holes by only using valid depth values as follows.

$$\begin{aligned} x_c &= ((1-u)x_l + ux_r + (1-v)x_t + vx_b)/2 \\ y_c &= ((1-u)y_l + uy_r + (1-v)y_t + vy_b)/2 \\ z_c &= ((1-u)z_l + uz_r + (1-v)z_t + vz_b)/2 \end{aligned} \tag{2}$$

where (x_c, y_c, z_c) are 3D coordinates of the current pixel (within a hole) in an image. We can reach 4 valid points in horizontal and vertical directions starting from the current position. The corresponding 3D coordinates to these 4 points are (x_t, y_t, z_t), (x_b, y_b, z_b), (x_l, y_l, z_l) and (x_r, y_r, z_r). Here, u and v denote the ratios for horizontal and vertical directions.

However, this procedure generates errors if depth difference between adjacent pixels is large. To avoid this, we do not apply this procedure if depth discontinuity is larger than the threshold Th_{dd}, e.g at boundary of an object. After examining 3D coordinates of each of the 4 directions, we apply this procedure only to holes which are small enough to be considered a plane.

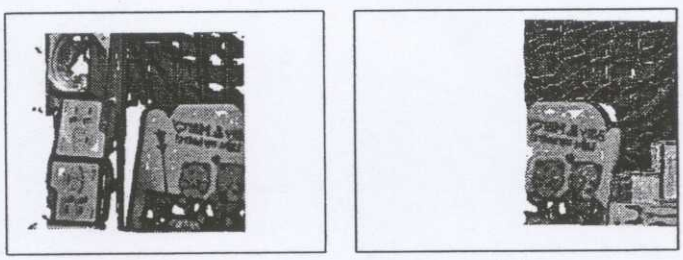
2.2 Initial Registration Using Inter-camera Calibration

In *initial registration step*, we calculate initial pose relationship between multi-view cameras by ICP-based inter-camera calibration [1] [9]. We estimate rotation and translation matrices, (R_S, T_S) and (R_D, T_D) using Tsai's algorithm [10]. S and D denote source and destination point sets, respectively. Each camera is a generalized multi-view camera with several lenses in horizontal and vertical directions.

However, back-projected 3D coordinates from each camera cannot be matched in VE due to inherent calibration errors. Thus, the inter-camera calibration is employed to find (R_S, T_S) and (R_D, T_D) by minimizing the distance between 3D grid points of a calibration pattern through an optimization process. Accurate geometric relationship between two cameras can be found by minimizing this distance. (The complete quaternion-based ICP algorithm can be found, e.g. in [1])

2.3 Fine Registration Using Color and Texture Descriptors

In *fine registration step*, we employ color and texture descriptors to obtain correct pairing. Fig. 2(a) shows the projection of 3D point cloud, acquired from a destination multi-view camera, onto 2D image plane. Fig. 2(b) is an image of 3D point cloud, from a source camera, projected onto a destination camera. Note that self-occlusion should be removed. Theoretically, Fig. 2(b) should exactly overlap with Fig. 2(a). However, discrepancies exist due to the errors in disparity estimation, camera calibration, etc. Therefore, based on the projection matrix P_S of a source camera, which minimizes errors in the overlapped area, we can register two sets of 3D point clouds.



(a) (b)

Fig. 2. Projection of 3D point cloud onto 2D image plane (a) projection of 3D point cloud of a destination camera onto its own image plane (b) projection of 3D point cloud of a source camera onto the image plane of the destination camera

We adopt color and texture information to define a cost function. That is, we split the whole image into blocks, and extract features by applying color and texture descriptors to source and destination images. Fig. 3 explains the process for extracting color and texture features from a single block.

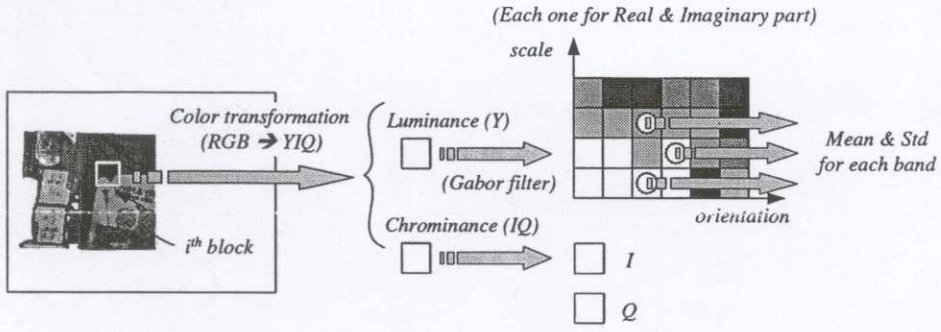


Fig. 3. Color and texture descriptors for a single block

Unlike luminance, shading does not have a significant influence on chrominance. Thus, a selection of a color space that reflects this property is essential for finding corresponding points between images. We decrease the influence of shading by separating chrominance from luminance in YIQ color space.

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3}$$

We define a cost function based on the color descriptor that takes account of only chrominance, as follows.

$$ColorDiff = \sqrt{(I_S - I_D)^2 + (Q_S - Q_D)^2} \tag{4}$$

We use a Gabor wavelet filter as a texture descriptor. Scale (frequency) and orientation tunable property of Gabor filter enables effective texture analysis [11]. Therefore, by applying Gabor filter (M scales and N orientations) to each block of an image, $M \times N$ filtered images are obtained for real and imaginary parts, respectively. We employ mean and standard deviation as features of each filtered image, and define the following cost function.

$$TextureDiff = \sqrt{(\mu_S - \mu_D)^2 + (\sigma_S - \sigma_D)^2} \tag{5}$$

where μ and σ denote mean and standard deviation, respectively, for each band of every block in source and destination images. Discrimination is very difficult in the shoulder of a bear or the upper central part in Fig. 3 because the color is similar. However, texture descriptor enables us to distinguish those blocks. The total cost function is defined as follows.

$$TotalDiff = (ColorDiff) + \alpha(TextureDiff) \tag{6}$$

where α is used as a weighting factor between color and texture information, and is determined experimentally.

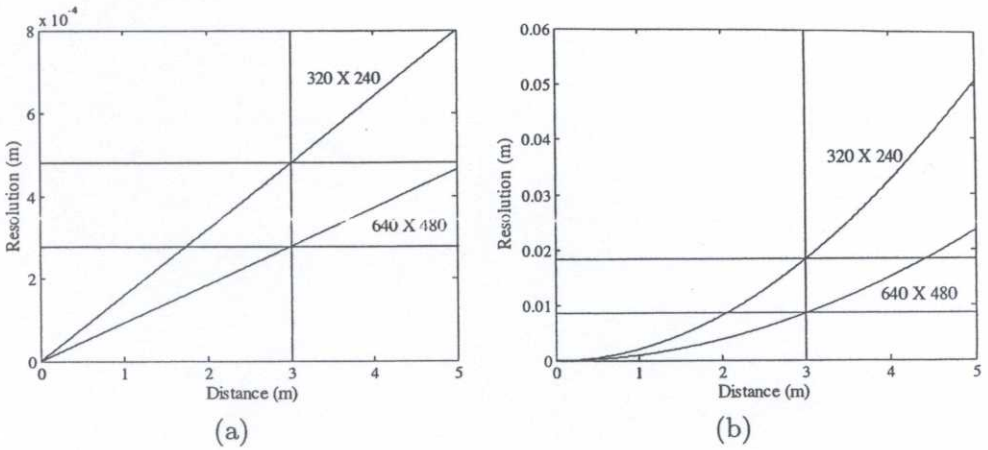


Fig. 4. Distance from camera vs. Resolution (a) Δx or Δy (b) Δz

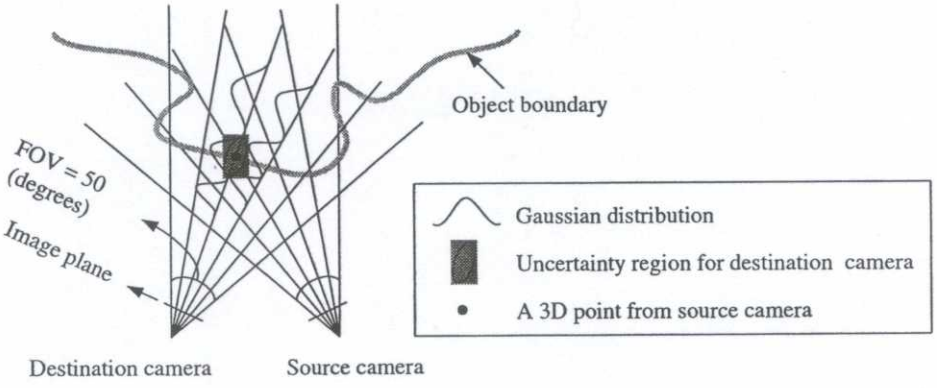


Fig. 5. An adaptive search range. When a point from a source camera is given, the corresponding point for a destination camera occupies an oval-shaped uncertainty region in 3D space

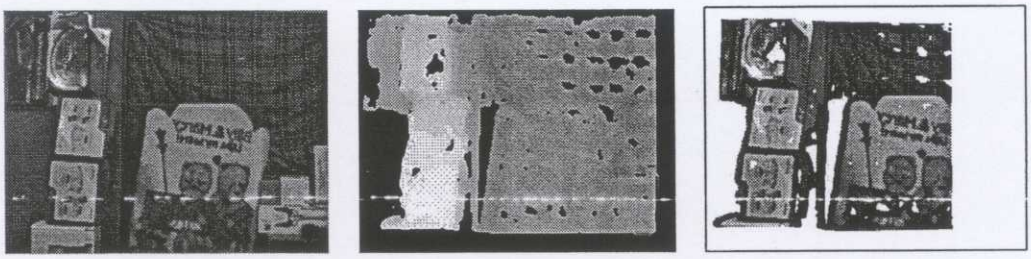
2.4 Color Selection Using Adaptive Search Ranges

After registration, for color composition between corresponding points in 3D space, we define adaptive search ranges that change with the distance between camera and object. That is, we use oval-shaped adaptive search ranges for color selection in overlapped areas. The multi-view camera has a correlation error m in disparity estimation and a calibration error p. Using 3D coordinates x, y and z, we can determine tolerance for each axis as follows.

$$\Delta x = \frac{pz}{f}; \Delta y = \frac{pz}{f}; \Delta z = \frac{fB}{d-m} - \frac{fB}{d} \tag{7}$$

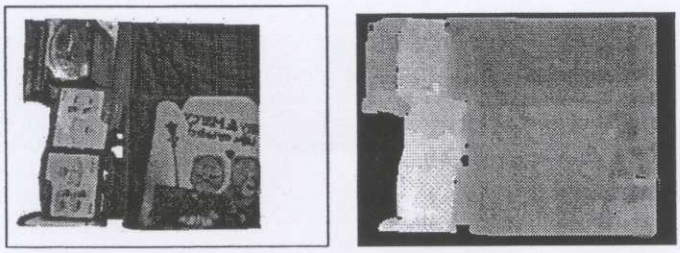
where d denotes disparity. B represents baseline and f means focal length.

Fig. 4 depicts the tolerances of Eq. (7). These values have Gaussian distributions for each coordinate axis, and correspond to standard deviations. We can apply the above description onto a pair of multi-view cameras as shown in Fig. 5.



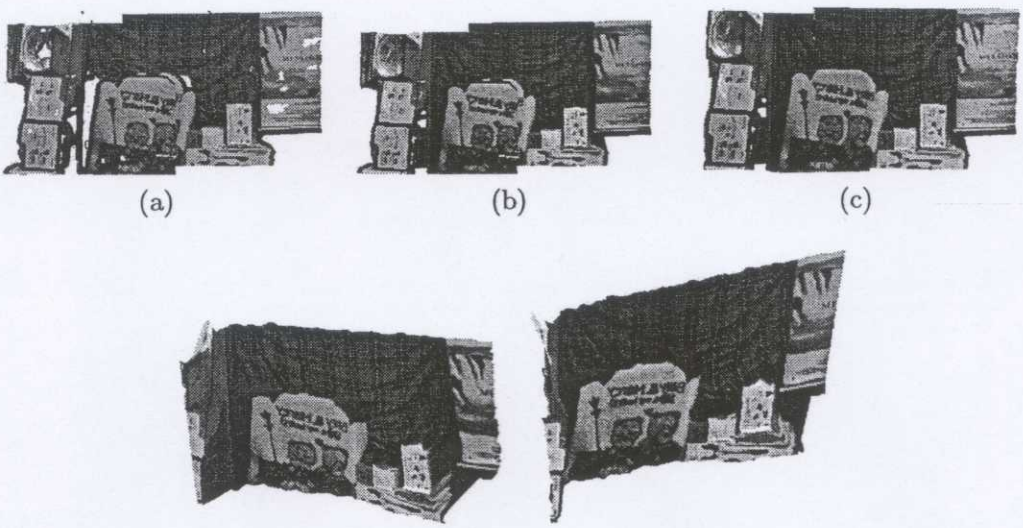
(a) (b) (c)

Fig. 6. Data from a source camera (a) original image (b) corresponding disparity map (c) 3D point cloud



(a) (b)

Fig. 7. Preprocessing results (a) 3D point cloud (b) corresponding disparity map



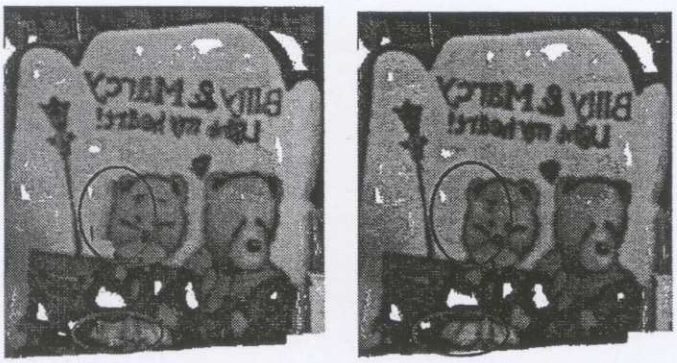
(a) (b) (c)

(d) (e)

Fig. 8. Registration results (a) combined 3D point cloud (b) combined 3D point cloud after preprocessing (c) registered 3D point cloud (d) reconstructed mesh model (view 1) (e) reconstructed mesh model (view 2)

Table 1. Performance comparison among methods

Methods	Conventional ICP	Color ICP (YIQ)	Proposed method		
			$\alpha = 6.0$	$\alpha = 7.0$	$\alpha = 8.0$
PSNR(dB)	27.33097	27.36363	28.24907	28.25096	27.10863



(a) (b)

Fig. 9. The comparison of visual quality (a) conventional ICP algorithm (b) proposed method (α=7.0)

That is, we can get Gaussian distribution linearly increasing with the distance along x and y axes; and Gaussian distribution increasing with distance along z axis. Therefore, we should consider oval-shaped adaptive search ranges, which change with the distance, to find corresponding points in D for 3D points in S.

3 Experimental Results and Analysis

The experiments were carried out under a normal illumination condition of general indoor environment. We used Digiclops which is a multi-view camera for image acquisition, and a Xeon 2.8 GHz CPU computer [12]. We employed a planar pattern with 7×5 grid points for initial registration. Distance between two consecutive points is 10.6 cm.

Fig. 6 shows original image, disparity map and corresponding 3D point cloud acquired from a source camera. Fig. 7 demonstrates preprocessing results. We can see that holes are filled only in small areas whose disparity difference is very small.

On the other hand, Fig. 8 illustrates registration results. Fig. 8(a) combines two 3D data. Fig. 8(b) and Fig. 8(c) show the results after preprocessing and registration. Fig. 8(d) and Fig. 8(e) show reconstructed mesh models adopting our algorithm.

In Table 1, performances of several methods are compared after 64 iterations. From the table, we can see that performance of the proposed method changes depending on the weighting factor α. Nevertheless, Fig. 9 explains that the visual

quality of the proposed method is better than that of ICP algorithm. Fig. 9(a) and Fig. 9(b) show the results of conventional ICP and the proposed method, respectively. Actually, total error is larger than the conventional ICP in terms of the closest distance. However, we observed that the visual quality of the proposed method is much better than that of the conventional ICP. The reason is that the conventional ICP only considers the closest distance instead of data themselves. For performance comparison, we used *PSNR* as follows.

$$PSNR = 20log_{10}\frac{255}{\sqrt{\frac{1}{N}\sum_{n=0}^{N-1}(Y_{S,n} - Y_{D,n})^2}}(dB) \tag{8}$$

where N is the number of points which are valid for both images.

4 Conclusions and Future Work

We proposed a novel registration method that employs multi-view cameras for image-based virtual environment generation using color and texture information. The proposed method can be used for modeling an indoor environment even when we cannot get accurate depth information. Furthermore, it enables a user to navigate through the generated VE by wearing a stereoscopic HMD. The proposed method not only lessens the real-time rendering burden but also provides the user with more realism and immersion as compared to model-based VE. There are still several remaining challenges. Global registration should be optimized and the time required for registration should be reduced. A natural composition between virtual objects and VE requires light source estimation and analysis to match illumination conditions of the VE.

Acknowledgements. This work was supported by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research Center at GIST.

References

1. P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," IEEE Trans. on PAMI, vol. 14, no. 2, pp. 239-256, 1992.
2. A. Johnson and S. Kang, "Registration and Integration of Textured 3-D Data," Tech. report CRL96/4, Cambridge Research Lab, 1996.
3. M. Levoy, K. Pulli, B.Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital Michelangelo project: 3D scanning of large statues. SIGGRAPH'00, pp. 131-144, July 2000.
4. K. Nishino and K. Ikeuchi, "Robust Simultaneous Registration of Multiple Range Images Comprising A Large Number of Points," ACCV2002, 2002.
5. Kari Pulli, Surface Reconstruction and Display from Range and Color Data, Ph.D. dissertation, University of Washington, 1997.
6. R. Szeliski and H.-Y. Shum. "Creating full view panoramic image mosaics and environment maps," SIGGRAPH '97, pp. 251-258, 1997.

7. G. C. Sharp, S. W. Lee and D. K. Wehe, "Invariant Features and the Registration of Rigid Bodies," IEEE Int'l Conf., on Robotics and Automation, pp. 932-937, 1999.
8. R. Fisher, "Projective ICP and Stabilizing Architectural Augmented Reality Overlays," Int. Symp. on Virtual and Augmented Architecture (VAA01), pp 69-80, 2001.
9. S. Kim, E. Chang, C. Ahn and W. Woo, "Image-based Panoramic 3D Virtual Environment using Rotating Two Multi-view Cameras," IEEE Proc. ICIP2003, vol. 1, pp. 917-920, 2003.
10. Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," Proc. of the Seventh IEEE Int'l Conf., vol. 1, pp. 666-673, 1999H.
11. B. S. Manjunath and W. Y. Ma. "Texture features for browsing and retrieval of large image data" IEEE Trans on PAMI, Vol. 18 (8), pp. 837-842, 1996.
12. Point Grey Research Inc., http://www.ptgrey.com, 2002.