The 12th Korea-Japan Joint Workshop on Frontiers of Computer Vision, 2-3, Feb., 2006, Tokushima, Japan

Panoramic 3D Reconstruction of an Indoor Scene Using A Multi-view Camera *

Sehwan Kim and Woontack Woo

GIST U-VR Lab., {skim,wwoo}@gist.ac.kr

Abstract We propose a novel method to generate a 3D surface using many 3D point clouds acquired from a multi-view camera. Until now, numerous disparity estimation algorithms have been developed with their own pros and cons. Thus, we may get various sorts of depth images. In this paper, we deal with the generation of a 3D surface using several 3D point clouds acquired from a generic multi-view camera. Firstly, a 3D point cloud is estimated based on spatio-temporal property of several 3D point clouds. Secondly, the evaluated 3D point clouds, acquired from two viewpoints, are projected onto the same image plane to find correspondences, and registration is conducted through minimizing errors. Finally, a surface is created by fine-tuning 3D coordinates of point clouds, acquired from several viewpoints. The reconstructed model can be adopted for interaction with as well as navigation in a virtual environment.

1 Introduction

Image-based reconstruction of a real environment plays a key role in providing visual realism while allowing users to navigate in and interact with a Virtual Environment (VE). The visual realism of reconstructed models encourages a user to interact with the VE proactively. Especially, off-the-shelf multi-view cameras enable generation of 3D models more conveniently. For this purpose, elaborate registration and integration are required in merging 3D point clouds for 3D model generation.

Until now, various reconstruction methods have been proposed. ICP (Iterative Closest Point) has been widely used, and Color ICP was proposed by Johnson for registration of 3D point clouds [1][2]. Especially, Park and Subbarao proposed a new method to remove inherent depth errors induced by disparity estimation [3]. Meanwhile, Pulli proposed a projective registration method employing planar perspective warping [4]. On the other hand, the volumetric methods fundamentally discretizes a 3D space, and determines the full and empty sets [5][6]. Even though an arbitrary shape can be represented, the resolution of the model is mainly determined by an initial discretization. Pixel-based PDE approaches do not depend on the discretization, and compute a continuous depth for every pixel [7]. Mesh representations can also adopt their resolutions to reconstruct detailed shapes, but have problems in dealing with self-intersections and topological changes during the search [8]. Meanwhile, there are probabilistic approaches on the basis of wide-baseline stereo techniques [9][10].

In this paper, we reconstruct a real environment using depth and color images. Firstly, a depth image is generated based on depth image refinement with the help of spatio-temporal property. Secondly, registration is accomplished by projecting 3D point clouds onto an image plane to find correspondences, and by minimizing errors. Finally, a surface is created by fine-tuning 3D coordinates of several 3D point clouds. The proposed method is carried out effectively even if the precision of 3D point cloud is relatively low by exploiting the correlation with the neighborhood. Furthermore, the proposed method is better than ICP (or Color ICP) with kd-tree with respect to the processing time. In general, many 2D images can be used for 3D reconstruction [11][12]. However, much time

* This research was supported by CTRC at GIST.

is required to generate a final 3D model.

The paper is organized as follows. In Chapter 2, high-resolution 3D scene reconstruction for an indoor scene is explained. After experimental results are shown and analyzed in Chapter 3, conclusions and future work are presented in Chapter 4.

2 Indoor Scene 3D Reconstruction

2.1 Depth Image Refinement

In general, disparity estimation results in inherent stereo mismatching errors that cause poor registration results. Thus, a depth image is refined by spatio-temporal property. Firstly, erroneous 3D points are removed by using the temporal property that the erroneous 3D points change dramatically in 3D space with time. Secondly, holes are filled by means of the spatial property that there is a spatial correlation among neighboring pixels [13]. Fig. 1 shows a flow diagram for 3D reconstruction.

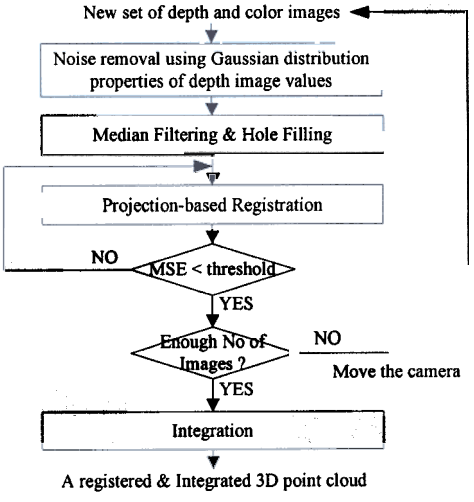


Fig. 1: Flow diagram for 3D reconstruction

2.2 Projection-based Registration

ICP algorithm is widely used for registration of 3D point clouds. However, the registration method exploiting the conventional ICP (or Color ICP) is not appropriate since it relies on the shortest distance [1][2]. Thus, a projection-based registration method is proposed by carrying out a pairing process that searches for correspondences between 3D point clouds acquired from destination and source viewpoints. We let a multi-view camera stationed around a wall, and acquire partial surfaces successively. Destination and source viewpoints mean the camera viewpoints at the previous and current positions of the camera, respectively.

In *initial registration* phase, a rigid-body transformation is applied to 3D points of corresponding features to estimate the poses of the camera [14]. We project each 3D point cloud acquired from destination and source viewpoints onto the destination viewpoint after the initial registration. It should be noted that the projection of 3D point cloud acquired from source viewpoint causes self-occlusion. This is eliminated based on the rays that originate at the camera center and pass though each pixel. However, there exist discrepancies between two projected data due to the errors in disparity estimation, camera calibration, etc.

In *fine registration phase*, corresponding features are employed. We register two partial surfaces by iteratively adjusting extrinsic parameters of source viewpoint with respect to destination viewpoint. In other words, we apply a Euclidean transformation T: $\Re^3 \rightarrow \Re^3$ to the source surface. The destination surface, S_{Dst}, is projected onto its own image plane and features, f_{Dst}, are extracted in the projected image plane. On the other hand, at each iteration, the source surface, S_{Src}, is projected onto the destination image plane and corresponding features, f_{Src}', are searched for. This is illustrated in Fig. 2.

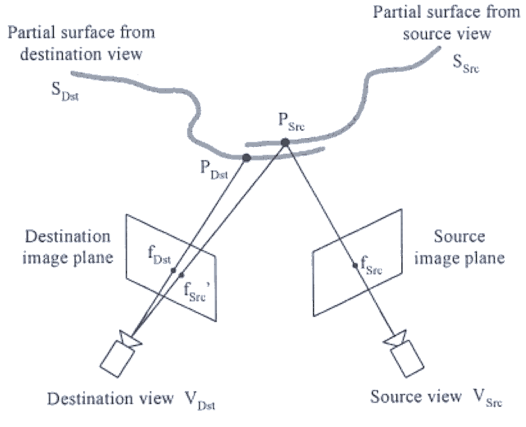


Fig. 2: Selection of corresponding features

For each feature f_{Dst} of the destination image, the corresponding feature f_{Src}' of the source image is searched for within the neighborhood of f_{Dst} using the modified KLT feature tracker [15]. P_{Dst} and P_{Src} are

3D points of f_{Dst} and f_{Src}, respectively. c_{Dst} and c_{Src} are RGB color components of f_{Dst} and f_{Src}, respectively.

Firstly, features are extracted over the overlapping area, Ω, in the destination image. The modified KLT feature tracker is adopted to extract feature corners. Then, S_{Src} is projected onto the destination image plane using the same calibration parameters that are used for the projection of S_{Dst}. Correspondences are searched for in the projected source image in sub-pixel unit. However, there may occur some mismatches that should be filtered out. That is, we should eliminate outliners and obtain only correct pairs between source and destination viewpoints.

Projecting S_{Src} onto the destination viewpoint produces an image I_{Src}'. Then, we can define a cost function measuring the mismatch between I_{Src}' and the destination image I_{Dst} as follows.

$$L = \sum_{i=1}^{N_{feat}} \kappa \left\{ \left(- \frac{\left\| f_{Dst,i} - f_{Src,i}' \right\|}{Dist_{ff}} \right) \left\| f_{Dst,i} - f_{Src,i} \right\| + \kappa_2 \left\| c_{Dst,i} - c_{Src,i}' \right\|^2 \right\} \tag{1}$$

where $\| \cdot \|$ and $Dist_{ff}$ represent norm, a value preset by considering the distances between f_{Dst} and f_{Src}', respectively. κ_2 is a weighting factor for color information, and N_{feat} denotes the number of features. κ_1 is described as follows to exclude the pair whose distance in 3D space exceeds a preset threshold Th. In other words, if the distance between P_{Dst} and P_{Src} is large, they are not included. Otherwise, the weighting is decided depending on the distance between the pair.

$$\kappa_1 = \begin{cases} \dfrac{\left\| P_{Dst} - P_{Src} \right\|}{Dist_{PP}} & if \left\| P_{Dst} - P_{Src} \right\| < Th \\ 0 & o/w \end{cases} \tag{2}$$

where $Dist_{PP}$ represents a value preset by considering the distances between P_{Dst} and P_{Src}.

In summary, we search for correspondences and use them to define a total cost function within the overlapping area. By minimizing the cost function, a final pose of the source viewpoint is estimated. That is, we can estimate the pose of source viewpoint $\{R_{Src}, T_{Src}\}$, with respect to the pose of destination viewpoint $\{R_{Dst}, T_{Dst}\}$ through minimizing errors of N_{feat} corresponding features as follows.

$$\begin{aligned} &\textit{Given two sets of corresponding points,} \\ &\textit{Find } \{R_{Src}, T_{Src}\} \textit{ w.r.t } \{R_{Dst}, T_{Dst}\} \\ &\textit{such that } \arg\min_{\{R_{Src}, T_{Src}\}} L \end{aligned} \tag{3}$$

The total error is minimized through Levenberg-Marquardt algorithm.

2.3 3D Surface Generation

Even after the fine registration phase, there may exist some 3D points, which are not close in 3D space although they are very close in the real world, due to disparity estimation errors. Thus, those 3D points should be manipulated so that they may be located closely in the reconstructed space.

In general, measurements are always corrupted by noise. The uncertainty of 3D point cloud affects not only the local properties of the reconstructed entity, but also the global structure one wants to recover. Thus, we want to suppress the influence of uncertainty on the recovered structure, and integrate 3D point clouds, acquired from several viewpoints, into a single 3D point cloud. In Fig. 3, we can see one example of data acquisition from each camera viewpoint using a multi-view camera. Note that each camera represents a multi-view camera, and thus enables to capture 3D point cloud and a pair of images at the same time.

Our final goal is to find a 3D representation of a scene from a given set of image pairs as well as rough 3D point clouds with full calibration information, i.e. known intrinsic and extrinsic parameters. Thus, for every pixel in input images, we want to infer the depth of the 3D point that each pixel is seeing. These depth images are integrated into a single 3D reconstructed model. For depth estimation, we define an energy function as follows.

$$d^* = \arg\min_{d} E(d). \tag{4}$$

$$E(d) = E_{Data}(d) + \lambda_1 E_{Smoothness}(d)$$

$$\begin{aligned} E_{Data}(d^1) &= \sum_{(x,y)} \left| I_L^1(x - d^1, y) + I_R^1(x, y) \right| \\ &+ \alpha \sum_{j=2}^{N_V} \left| P_{L,d^1}^1 - P_{L,d^j}^j \right| \qquad P_{L,d^j}^j \in C_{L,d^1}^{1,0} \end{aligned} \tag{5}$$

$$E_{Smoothness}(d^1) = \sum_{\eta} \left| d^1 - d_\eta^1 \right| + \beta \sum_{i=1}^{N_C} \sum_{j=2}^{N_V} \left| \#C_{L,d^1}^{1,0} - \#C_{L,d^j}^{j,i} \right| \tag{6}$$

where N_V represents the number of views, and N_C denotes the number of neighboring cubes. P^j_{L,d^j} means color information for each 3D point. The notation j is used for the j^{th} view, and L/R is employed to indicate left or right image. The notation d^j is a disparity value with respect to the j^{th} view. Meanwhile, $\#C^{j,i}_{L,d^j}$ means the number of 3D points associated with the i^{th} cube with respect to the j^{th} view.

d^* is a final disparity value to be sought through minimizing $E(d)$. The energy function is composed of mainly two components: (i) data part E_{Data}, and (ii) smoothness part $E_{Smoothness}$. The data part measures how well a disparity value d agrees with several pairs of input images and 3D point clouds. On the other hand, the smoothness part encodes the smoothness assumptions made by the algorithm.

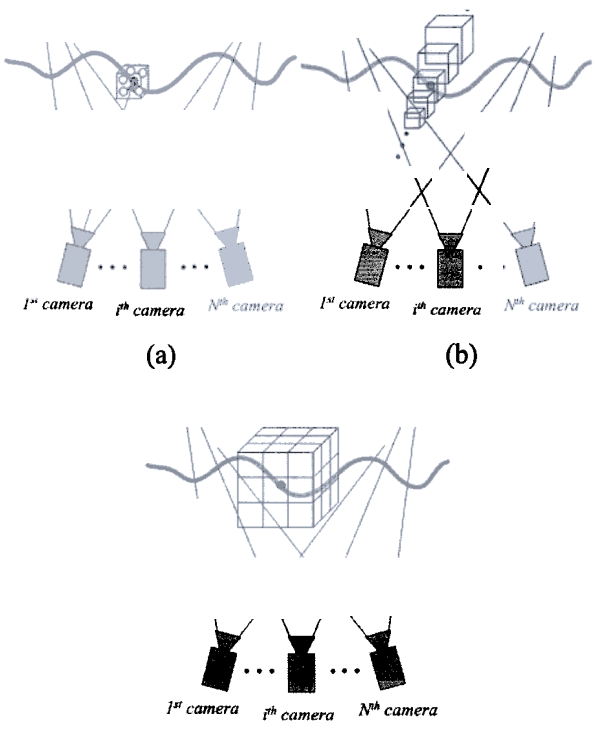


Fig. 3: Cube representation (a) a cube with respect to the current pixel of the reference camera (b) a variable cube depending on a disparity value (c) neighboring cubes (off-centered cubes) as well as a centered cube

The first terms of both parts are similar to data and smoothness constraints of the general disparity estimation algorithm except that the disparity range is determined depending on the disparity value which is already provided as an input data.

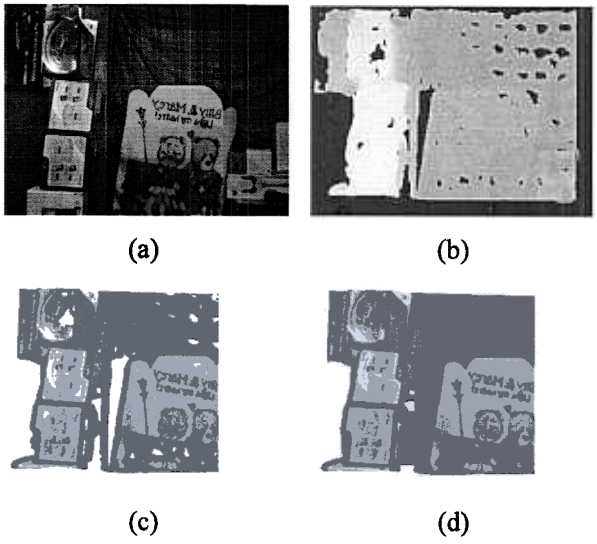
The second term of data part investigates the constellation of 3D points in 3D space so that their relative 3D positions as well as their relative colors may be compared. Fig. 3(a) shows a cube that is created with respect to a 3D point of the reference camera. The cube is employed to incorporate a spatial relationship among 3D points in 3D space. Whereas Fig. 3(a) shows a cube with respect to the current pixel of the reference camera, Fig. 3(b) depicts a variable cube depending on the changeable disparity value.

The second term of smoothness part deals with all 3D points included in neighboring cubes whose sizes change according to the distance from camera center to current 3D point of the reference viewpoint. Fig. 3(c) depicts neighboring cubes (off-centered cubes) as well as the centered cube with respect to the current pixel of the reference camera for the smoothness part.

3 Experimental Results and Analysis

The experiments were carried out under a normal illumination condition of a general indoor environment. We used Digiclops, an IEEE 1394 multi-view camera for color image and 3D point cloud acquisition [16]. It calculates 3D coordinates through a block-based disparity estimation algorithm in sub-pixel unit by exploiting three lenses, and uses ICX084AK CCD censor. Its baseline (B) is 10 cm and focal length (f) of each lens is 6 mm. Correlation error (m) and calibration error (p) are 0.08 and 0.08, respectively.

We set N_f to 30 and Th_{dd} to 0.15 [13]. Before applying the registration step, we removed invalid areas, such as object boundaries, homogeneous areas. Holes, whose depth differences are small, are also filled. Fig. 4 illustrates the results of the depth image refinement. Fig. 4(a) and Fig. 4(b) show an original image and a corresponding depth image, respectively. Corresponding 3D point cloud and the results of depth image refinement are shown in Fig. 4(c) and Fig. 4(d), respectively. In this example, we cut the right side of the original image since the error bound of the right side is very large. We can observe that invalid areas are effectively eliminated. In addition, holes, whose depth differences are small, are also filled.



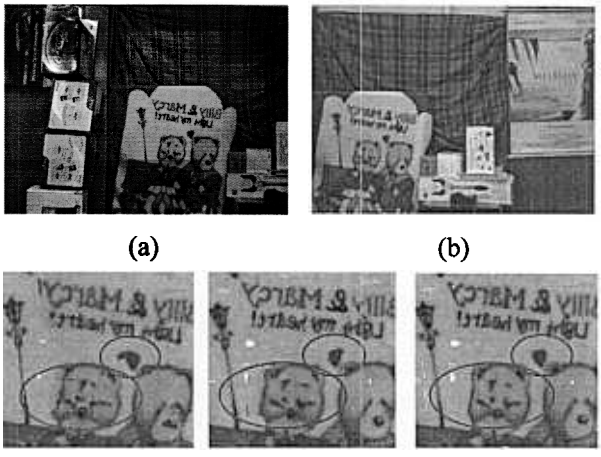
(a) (b) (c) (d)

Fig. 4. Depth image refinement results (a) original image (b) depth image (c) 3D point cloud before depth image refinement (d) 3D point cloud after depth image refinement

The registration results are shown in Fig. 5, which explain that the visual quality of the proposed method is better than that of ICP. Fig. 5(a) and Fig. 5(b) show left and right images, respectively. After initial registration, we can obtain the results as shown in Fig. 5(c). Note that heart shape, face part of bear and some letters are smeared. In Fig. 5(d) and Fig. 5(e), we can see the final registration results of ICP and the proposed method, respectively. Actually, total error of the proposed method is larger than that of 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.

Fig. 6 depicts the performance comparison with other methods. In the experiments, the numbers of 3D points acquired from the destination and source viewpoints are 145,870 and 189,341, respectively. We can see that speed as well as performance of the proposed method is better than that of ICP (or Color ICP) with kd-tree. Table 1 shows the registration accuracy and processing time.

The registration and modeling results for two walls are shown in Fig. 7. To get the results, we moved the multi-view camera around two walls and registered the acquired 3D point clouds. On the left wall, sofa, vase, table, TV, doll are observed. On the other hand, vase, sofa, bookshelf and window are on the right wall.



(a) (b) (c) (d) (e)

Fig. 5. The comparison of visual quality (a) left image (b) right image (c) initial registration (d) ICP (e) proposed method

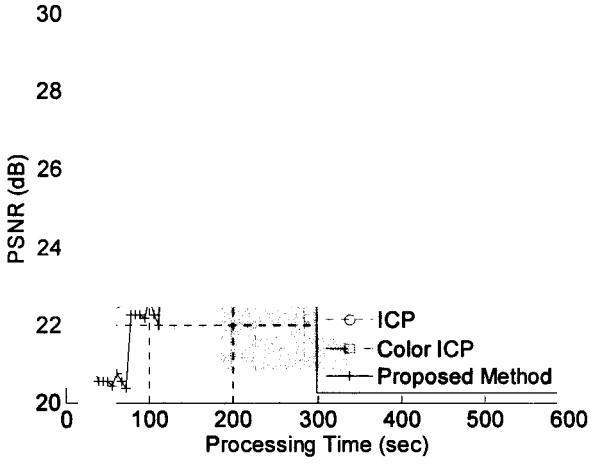


Fig. 6. Performance comparison

Table 1. Registration accuracy and processing time

Method	ICP with kd-tree	Color ICP with kd-tree	Proposed
Final PSNR (dB)	26.2556	27.2856	28.1307
Time (sec)	436.531 (10 Itr's)	511.516 (10 Itr's)	238.8590 (43 Itr's)
Time/Itr (sec)	43.6531	51.1516	5.55486

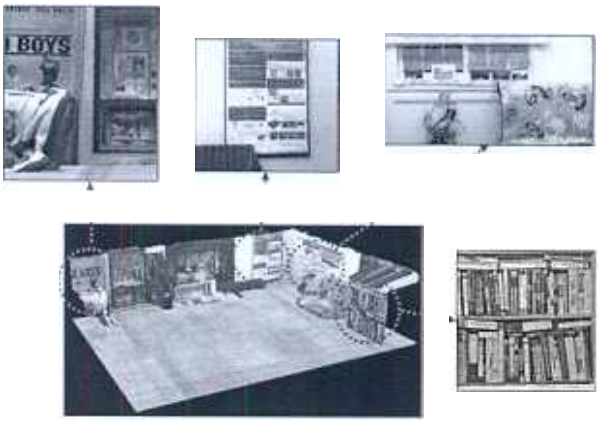


Fig. 7. Indoor scene reconstruction and virtual object augmentation results

4 Conclusions and Future Works

We proposed a novel 3D scene reconstruction method that exploits partial 3D point clouds acquired from a multi-view camera for an indoor environment. We showed that even though the error of depth information is relatively large compared to that of laser-scanned data, 3D point clouds are effectively registered between two viewpoints. Furthermore, the time required for registration is less compared to ICP (or Color ICP) with kd-tree. We also showed that an effective reconstruction is possible using 3D point clouds combined with 2D image pairs. There are still several remaining challenges. First, global registration should be optimized for 3D reconstruction of the entire indoor environment. Natural augmentation of virtual objects into the reconstructed room environment requires light source estimation and analysis to match illumination conditions of the VE. Finally, dense disparity estimation is required to obtain better results.

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, Digital Equipment Corp., Cambridge Research Lab, Oct., 1996.

[3] S. Park and M. Subbarao, "A Range Image Refinement Technique for Multi-view 3D Model Reconstruction," *3-DIM*, pp. 147-154, 2003.

[4] K. Pulli, *Surface Reconstruction and Display from Range and Color Data*, Ph.D. dissertation, UW, 1997.

[5] K. Kutulakos and S. Seitz, "A Theory of shape by space carving," *IJCV*, vol. 38(3), pp. 197-216, 2000.

[6] O. Faugeras and R. Keriven, "Complete dense stereovision using level set methods," *ECCV*, 1998.

[7] C. Strecha, T. Tuytelaars and L. Van Gool, "Dense matching of multiple wide-baseline views," *ICCV*, pp. 1194-1201, 2003.

[8] G. Vogiatzis, P. Torr and R. Cipolla, "Bayesian stochastic mesh optimization for 3d reconstruction," *BMVC*, 2003.

[9] P. Gargallo, P. Sturm, "Bayesian 3D Modeling from Images Using Multiple Depth Maps," *CVPR'05*, vol. 2, pp. 885-891, 2005.

[10] C. Strecha, R. Fransens and L. Van Gool, "Wide-baseline Stereo from Multiple Views: a Probabilistic Account," *CVPR'04*, vol. 2, pp. 552-559, 2004.

[11] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, *Visual modeling with a hand-held camera*, International Journal of Computer Vision 59(3), 207-232, 2004.

[12] T. Sato, M. Kanbara, N. Yokoya and H. Takemura, "Dense 3-D reconstruction from a monocular image sequence by estimating camera motion parameters," *CVPR'01*, 2001.

[13] S. Kim and W. Woo, "Projection-based Registration using Multi-view camera for Indoor Scene Reconstruction," *3-DIM*, pp. 484-491, 2005.

[14] K. Kim and W. Woo, "3D Camera Tracking from Disparity Images," *VCIP*, pp. 1381-1388, 2005.

[15] KLT: Kanade-Lucas-Tomasi Feature Tracker, http://www.ces.clemson.edu/~stb/klt/, 2005

[16] Point Grey Research Inc., http://www.ptgrey.com, 2002.

Sehwan Kim: received his B.S. degree in Electronics Engineering from University of Seoul (UOS), Seoul, Korea, in 1998 and M.S. degree in Dept. of Info. & Comm. from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 2000, respectively. Now, he is a Ph.D. candidate in Dept. of Info. & Comm. at GIST since 2000. Research Interest: Virtual/Mixed Reality, 3D computer vision and its applications including attentive AR and mediated reality, HCI.

Woontack Woo: received his B.S. degree in EE from Kyungpook National University, Daegu, Korea, in 1989 and M.S. degree in EE from POSTECH, Pohang, Korea, in 1991. He received his Ph. D. degree in EE-Systems from University of Southern California, Los Angeles, USA. During 1999-2001, as an invited researcher, he worked for ATR, Kyoto, Japan. In 2001, as an Assistant Professor, he joined Gwangju Institute of Science and Technology (GIST), Gwangju, Korea and now at GIST he is leading U-VR Lab. Research Interest: 3D computer vision and its applications including attentive AR and mediated reality, HCI, affective sensing and context-aware for ubiquitos computing, etc.