SIFT-BASED IMPROVEMENT OF DEPTH IMAGERY

Haopeng Li and Markus Flierl

School of Electrical Engineering
KTH Royal Institute of Technology, Stockholm
{haopeng, mflierl}@kth.se

ABSTRACT

Depth Image Based Rendering (DIBR) is a widely used technique to enable free viewpoint television. It utilizes one or more reference texture images and their associated depth images to synthesize virtual camera views. The depth image plays a crucial role for DIBR. However, most of the conventional depth image estimation approaches determine the depth information from a limited set of nearby reference images. This leads to inconsistencies among multiple reference depth images, thus resulting in poor rendering quality. In this paper, we propose an approach that uses the Scale Invariant Feature Transform (SIFT) to improve depth images at virtual viewpoints. We extract SIFT features in left and right reference images, and use feature correspondences to improve the consistency between reference depth images. By doing so, the quality of rendered virtual views can be enhanced.

***Index Terms*—** Free viewpoint television, DIBR, SIFT, depth consistency.

1. INTRODUCTION

In recent years, research on free viewpoint television has developed rapidly. Envisioned services will provide a rich and compelling immersive experience by allowing viewers to place a virtual camera in a live-action scene and move it freely. However, the multiview video data captured by multiple cameras consequently increases the data volume significantly. Thus, efficient processing at lower computational complexity is needed to handle a large amount of data for high-quality real-time view synthesis.

Depth Image Based Rendering (DIBR) utilizes one or more reference texture images and their associated depth images to synthesize ”virtual” camera views [1]. Generally, DIBR projects original pixels from reference images into 3D world coordinates according to their depth values as specified by the associated depth images. Thereafter, 3D world coordinates are projected onto the image plane of the virtual camera view. DIBR is advantageous due to its rendering ability, low bandwidth requirement, and low computational complexity [2].

The depth image plays a crucial role in DIBR. The more accurate the depth map, the better is the quality of the rendered image. Many approaches have been proposed [3-5] to improve the accuracy of depth images. However, most of the conventional depth image estimation or improvement methods estimate or improve the depth information view by view [6]. This leads to inconsistencies among multiple reference depth images. Thus, when warping a pair of reference depth images to the same target position, misaligned or inconsistent depth will reduce the reliability of each single depth value and result in poor rendering quality.

In this paper, we propose an approach that uses a Scale Invariant Feature Transform (SIFT) to improve the consistency among multiple reference depth images. SIFT has been widely used in object recognition, robotic mapping, video tracking, and match moving [7]. It has also been exploited for depth recovery in textureless image pairs [8]. We extract SIFT features in left and right reference images and use feature correspondences to improve the consistency among reference depth images. The primary advantage of SIFT is its invariance under rotation, scale change and affine transformation. It performs better than conventional feature matching techniques like ordinary correlation and Harris corners which are not invariant under rotation and changes in scale [9]. Additionally, SIFT can be run in real-time with modern GPU implementations [10]. All these benefits ensure that SIFT can be used efficiently to extract feature matches among different camera views.

The remainder of this paper is organized as follows. Section 2 presents our SIFT-based depth image improvement algorithm that can enhance the consistency among reference depth images. The verification of our algorithm and the comparison to a reference algorithm are given in Section 3, followed by a short conclusion.

2. IMPROVEMENT OF MULTI-VIEW DEPTH CONSISTENCY

In this section, SIFT features will be exploited to establish feature correspondences between two reference images. A warping method will be applied to acquire two versions of the depth images at the target position, one from the left ref-

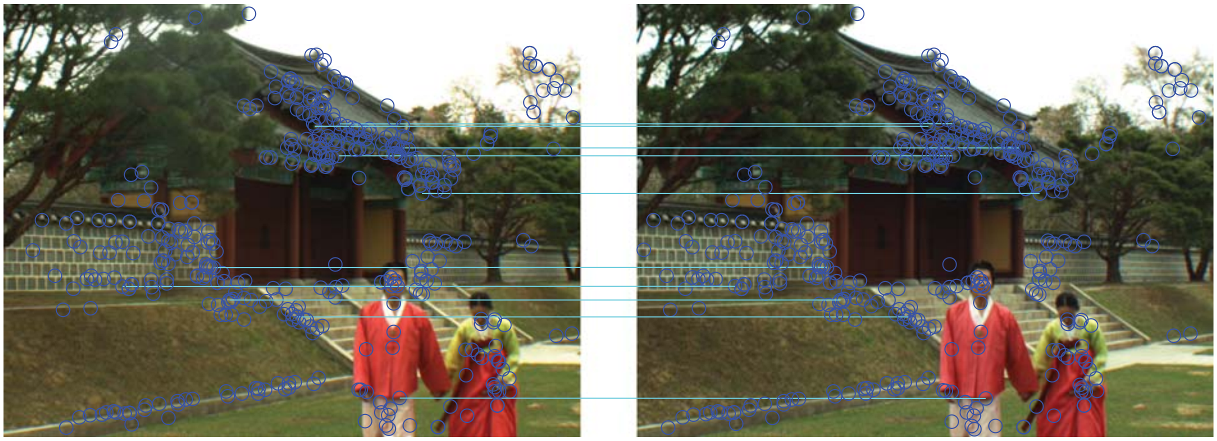

Fig. 1. SIFT features in left and right reference images.

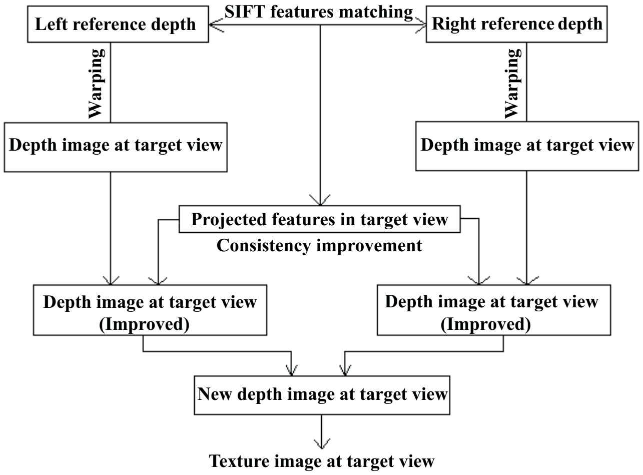
Left reference depth
SIFT features matching
Right reference depth
Warping
Warping
Depth image at target view
Depth image at target view
Projected features in target view
Consistency improvement
Depth image at target view (Improved)
Depth image at target view (Improved)
New depth image at target view
Texture image at target view

Fig. 2. SIFT-based improvement of depth.

erence d_0^l and one from the right reference depth d_0^r. The SIFT features will be used to improve the consistency between these two versions of the target position depth image. Thus, a new depth image at the target position can be generated from above two improved versions. Finally, with the newly generated depth image, the texture at the target position can be projected onto the target image plane. The block diagram of our algorithm is shown in Fig. 2.

2.1. SIFT Feature Matching and Refinement

First, we extract SIFT features in both reference images and find correct correspondences. Correct correspondences relate to the same 3D point in the scene and, hence, can be used to establish consistency in depth images. That is, a correct correspondence in two reference images leads to a pair of consistent depth values in their associated depth images.

For simplicity, we choose a rectified camera display scenario where two cameras are parallel and share the same image plane. Left I_1 and right I_2 reference images are captured by these two cameras, and we set our virtual target view at a position between them. By using the SIFT algorithm, we can extract and match the SIFT features between left I_1 and right I_2 reference images, as depicted in Fig. 1.

Let the pixel with position $q = (x, y)$ be in the left I_1 or the right I_2 image. Knowing the camera calibration parameters, we are able to check the correctness of feature matches based on the geometric constraint [1]

$$[X, Y, Z]^T = R^{-1} \cdot A^{-1} \cdot [x, y, 1]^T \cdot d - R^{-1} \cdot t, \qquad (1)$$

where $[X, Y, Z]^T$ is the 3D world coordinate, d is the depth, and where R, A and t are the camera calibration parameters which depend on the camera position. In our case, d is equal to Z since a rectified camera scenario. Thus, d is replaced by Z for simplicity.

Let $p_i^r \leftrightarrow p_j^l$ be a feature correspondence, where p_i^r denotes the i-th feature point with the image coordinate (x_i^r, y_i^r) in the right reference and p_j^l the j-th feature point with the image coordinate (x_j^l, y_j^l) in the left reference. If it is a correct correspondence, p_i^r and p_j^l are originally projected from the same 3D world coordinate. Thus, by using (1) for corresponding points, the following equation holds:

$$\begin{aligned} & R_r^{-1} \cdot A_r^{-1} \cdot [x_i^r, y_i^r, 1]^T \cdot Z - R_r^{-1} \cdot t_r \\ = \; & R_l^{-1} \cdot A_l^{-1} \cdot [x_j^l, y_j^l, 1]^T \cdot Z - R_l^{-1} \cdot t_l. \end{aligned} \qquad (2)$$

As we assume to know the true camera calibration parameters, expression (2) is over-determined. For our practical application, we determine the least square error solution of the depth value Z^* according to

$$\begin{aligned} Z^* = \; & \arg\min_{Z} \| R_r^{-1} \cdot A_r^{-1} \cdot [x_i^r, y_i^r, 1]^T \cdot Z - R_r^{-1} \cdot t_r \\ & - R_l^{-1} \cdot A_l^{-1} \cdot [x_j^l, y_j^l, 1]^T \cdot Z + R_l^{-1} \cdot t_l \|_2. \end{aligned} \qquad (3)$$

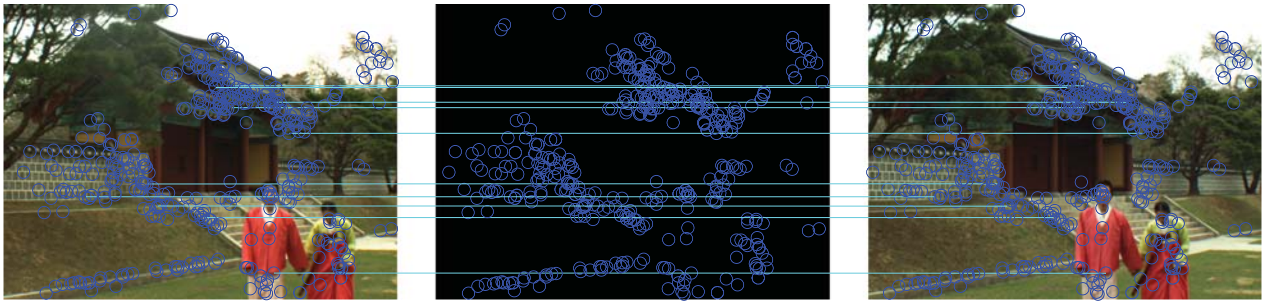

Fig. 3. SIFT features in reference and target images.

The two resulting 3D world coordinates $[X,Y,Z]_i^{rT}$ and $[X,Y,Z]_j^{lT}$ are obtained by the least square error solution and (1) with respect to p_i^r and p_j^l. However, some small misalignment caused by calibration parameters should also be considered. Thus, we use an additional criterion. If $\|[X,Y,Z]_i^{rT} - [X,Y,Z]_j^{lT}\|_2 < \delta_d$, where δ_d is a small threshold for the Euclidean distance in 3D space, the correctness of the correspondence $p_i^r \leftrightarrow p_j^l$ is sufficiently reliable.

2.2. Projection in Target View

For each correct correspondence $p_i^r \leftrightarrow p_j^l$, (1) and (3) are used to calculate a least square error solution of $[X,Y,Z]^T$, which is the 3D world coordinate of both p_i^r and p_j^l. This 3D world coordinate is then projected onto the image plane of the virtual target camera I_t, resulting in the corresponding feature p_k^t. Therefore, combining SIFT and geometric constraints, we obtain the feature correspondence $p_i^r \leftrightarrow p_j^l \leftrightarrow p_k^t$ between references and target image as depicted in Fig. 3. Thus, we define a set of correct feature matches $R = \{(p_i^r, p_j^l, p_k^t) | p_i^r \leftrightarrow p_j^l \leftrightarrow p_k^t\}$ along with the sets of chosen features for each image

$$\begin{aligned} F^r &= \{p_i^r | p_i^r \leftrightarrow p_j^l \leftrightarrow p_k^t\} , \\ F^l &= \{p_j^l | p_i^r \leftrightarrow p_j^l \leftrightarrow p_k^t\} , \\ F^t &= \{p_k^t | p_i^r \leftrightarrow p_j^l \leftrightarrow p_k^t\} . \end{aligned} \tag{4}$$

Finally, we define a sparse depth map only for the chosen features in each image. For example, for the right reference image, we obtain

$$d_f^r(p^r) = Z^*(p^r), \forall p^r \in F^r, \tag{5}$$

where the feature depth map is not defined for other pixels than the chosen features. Similarly, for the target view image we define

$$d_f^t(p^t) = Z^*(p^t), \forall p^t \in F^t. \tag{6}$$

2.3. Warping of Reference Depth Images

Widely used depth images use a finite number of depth planes, e.g., 256 planes for 8 bit depth images. We exploit the finite set of depth planes to define sets of pixels at the same depth level. The depth level set for the right reference image is the set of all pixels q that have the depth u

$$L_u^r = \{q | d_0^r(q) = u\} . \tag{7}$$

The depth level set for the left reference image is defined similarly.

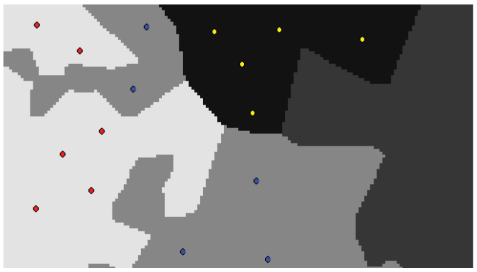

Fig. 4. An illustration of the level set L_u with correct features F and the features in a given level set W_u. The gray segments depict the level sets L_u. The points depict the set of correct features F. The collection of points with the same color indicate the set of features W_u at the depth u.

To warp reliably the depth level set toward the target view, we choose the features in a given level set

$$W_u^r = F^r \cap L_u^r, \tag{8}$$

where F^r is the set of chosen features, and L_u^r is the level set for depth u, as illustrated in Fig. 4.

As each depth level set specifies pixels that lie in the same depth plane that is parallel to the camera, the warping process can be accomplished by a disparity shift.

To calculate the correct disparity for a given depth level, we consider only the correct features in the given level set according to

$$\Delta_u^r = \frac{1}{|W_u^r|} \sum_{p_i^r \in W_u^r, p_i^r \leftrightarrow p_k^t} (p_i^r - p_k^t), \tag{9}$$

where p_i^r is correctly connected to p_k^t in the target view. The disparity for the left reference is determined similarly.

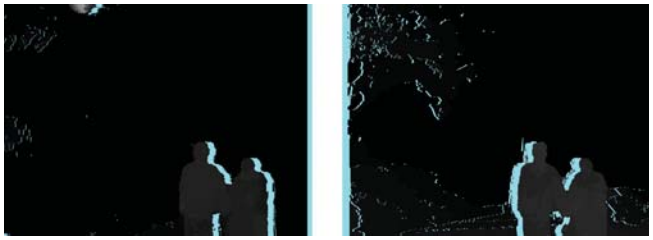

Fig. 5. Left and right versions of the target depth map from left and right references (light-blue parts are occlusions).

After we have shifted all level sets by their corresponding disparity, we observe overlap for certain areas. In that case, the foreground overrides the background, i.e., the smaller depth values are kept. Note, possible holes are kept unchanged at this stage of the algorithm. This process results in the left and right version of the target depth map, $d_1^l(q)$ and $d_1^r(q)$, respectively. An example is shown in Fig. 5.

2.4. SIFT-Based Improvement of Depth

Now we are able to improve the depth information d_1 in the left and the right version of the target view by using the correct feature correspondences in the target view. Note that the occlusion problem will not be solved in this step.

We use again the definition of depth level sets. At this stage, it is based on the target depth d_1. For the right version of the target view, we obtain the depth level set

$$Q_u^r = \{q | d_1^r(q) = u\} \ . \tag{10}$$

The level set for the left version is defined similarly.

For improving the depth d_1, features of the target view for a given level u are used. The chosen target view features for the left version are given by

$$S_u^l = F^t \cap Q_u^l, \tag{11}$$

where F^t denotes the correct features in the target view according to (4). Chosen target view features for the right version are defined similarly.

In the previous step (6), we have determined accurate depth values of the features in the target view $d_f^t(p^t), \forall p \in F^t$. Moreover, we have a set of chosen features S_u with accurate depth values $d_f^t(p^t)$ that are associated with the level set Q_u. If the depth values of these features do not match with the given level u, we have the opportunity to improve the depth information of the pixels in Q_u with the help of the accurate depth values of the features as illustrated in Fig. 6.

For that, we check the estimated variance of the depth values of the chosen features in Q_u. If $\mathrm{Var}_{p \in S_u}\{d_f^t(p)\} < \delta_f$, the assumption of constant depth in Q_u is accurate. Hence, we improve the depth for all pixels in Q_u according to

$$d_2(q) = \frac{1}{|S_u|} \sum_{p \in S_u} d_f^t(p), \forall q \in Q_u. \tag{12}$$

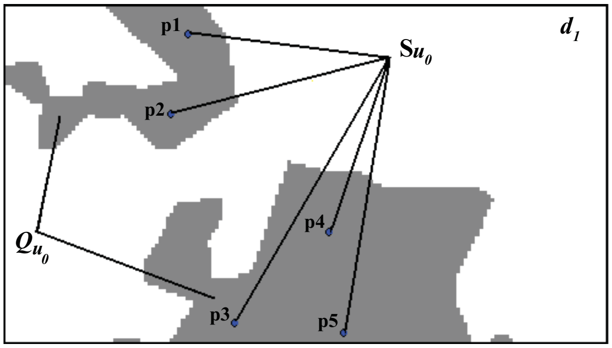

Fig. 6. Chosen features S_u at the target view for a given level set Q_u.

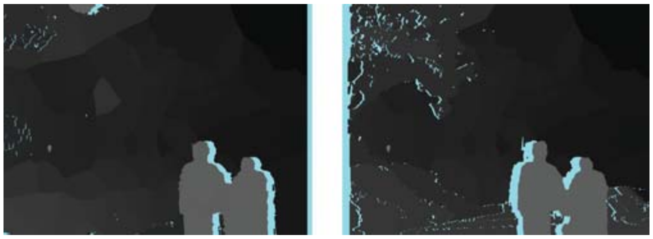

Fig. 7. Improved versions of the target depth maps d_2 from left and right reference (light-blue parts are occlusions).

However, if the estimated variance exceeds the given threshold, the assumption of constant depth in Q_u is not accurate. In this case, we find for each pixel $q \in Q_u$ its nearest chosen feature with respect to Euclidean distance, and assign the depth value of the nearest chosen feature also to pixel q

$$d_2(q) = d_f^t(p^*), \text{ with } p^* = \arg\min_{p \in S_u} \|q - p\|_2. \tag{13}$$

By doing so, the region Q_u will be fragmented into local neighborhoods as defined by the location of the features S_u. This can be observed in Fig. 7.

2.5. Fusion of Depth Images

In the previous steps, we have generated left and right versions of the depth image at the target position. Now, we are able to combine them for the final depth image at the target position.

By combining both versions, we are able to address the occlusion areas and define the final depth image at the target position.

$$d_3(q) = \begin{cases} d_2^l(q), & \text{if } \mathrm{d}_2^{\mathrm{r}}(\mathrm{q}) \text{ is not defined} \\ d_2^r(q), & \text{if } \mathrm{d}_2^{\mathrm{l}}(\mathrm{q}) \text{ is not defined} \\ \frac{1}{2} \cdot [d_2^l(q) + d_2^r(q)], & \text{if both are defined.} \end{cases} \tag{14}$$

If d_2 is not defined at position q in both versions, we will define a window around position q, calculate the histogram of

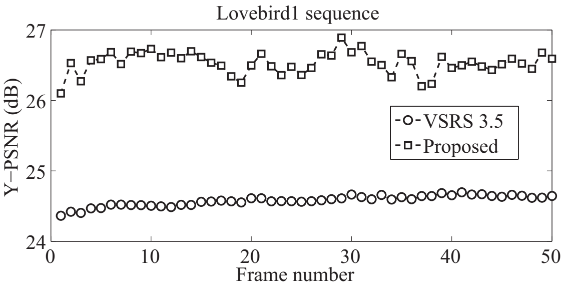

Fig. 8. Comparison of Y-PSNR of rendered images for proposed algorithm and MPEG VSRS 3.5 (wide-baseline).

adjacent depth values, select the most likely value, and assign it to $d_3(q)$.

2.6. Texture Warping

The texture of the target view will be warped from left and right reference textures. With the improved depth image d_3 at the target view, we use 3D warping to generate the left and the right version of the texture at the target view. To render the texture at the target view, we choose in general the warped texture of a reference view that is closest to the target view. However, if parts are occluded in the closest reference, we choose the other reference for rendering.

3. EXPERIMENTAL RESULTS

We compare the performance of our algorithm to a MPEG reference algorithm. For that, we use the the Peak Signal To Noise Ratio (PSNR) of the luminance component to assess the quality of the target view image with respect to an actual camera image at the same view position. The subjective quality is also discussed.

We use the multi-view video test set *Lovebird1* which is provided by ETRI [11] and the test set *Newspaper* which is provided by GIST [12]. We compare the performance of our proposed algorithm to that of the View Synthesis Reference Software 3.5 (VSRS 3.5) [13] which is used for MPEG 3DV/FTV exploration experiments. VSRS 3.5 uses also a DIBR approach which synthesizes the target view by referencing left and right texture images and their associated depth images. The average Y-PSNR between rendered view and actual camera view at the same position will be used to evaluate the performance of each algorithm.

For our experiments, we choose one wide-baseline configuration and one narrow-baseline configuration. The wide-baseline configuration is challenging for both algorithms. We use 50 successive frames from the test sequences.

Our experiments show that our SIFT-based algorithm for depth image improvement outperforms MPEG's reference algorithm. The average Y-PSNR of the rendered images improves by about 2dB for the wide-baseline setting and about 1dB for the narrow-baseline setting, as depicted in Fig. 8 and Fig. 9. Fig. 12 shows that our approach can improve the accuracy of the depth images at the target position by using a sparse set of image features. The improved depth images enhance the rendering quality of both foreground objects (Fig. 10) and background objects (Fig. 11). Our approach is particularly beneficial for wide-baseline settings.

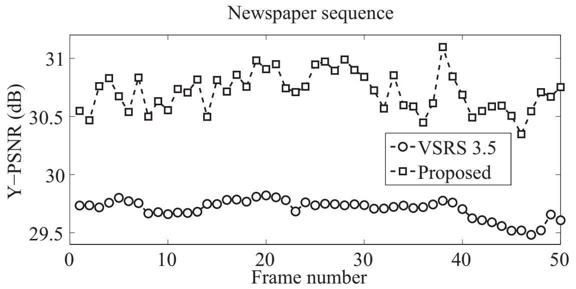

Fig. 9. Comparison of Y-PSNR of rendered images for proposed algorithm and MPEG VSRS 3.5 (narrow-baseline).

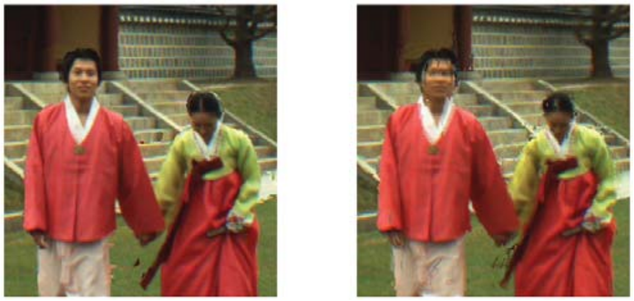

Fig. 10. Comparison of foreground objects (left: proposed, right: VSRS 3.5).

4. CONCLUSIONS

We discussed an algorithm to improve depth images for given target view positions that is based on a sparse set of accurate SIFT features. For that, a subset of reliable and accurate correspondences is defined among the SIFT features. Further, sets of pixels with the same depth level are defined. We use the accurate information of the SIFT features to update these sets of pixels. Both, the shape of the pixel regions and their depth values are enhanced. The experimental results show that our proposed approach outperforms MPEG's reference algorithm VSRS 3.5 in terms of objective and subjective image quality.

5. ACKNOWLEDGMENTS

This work was supported in part by the European Commission in the context of the project ICT-FP7-248020 "FINE – Free-

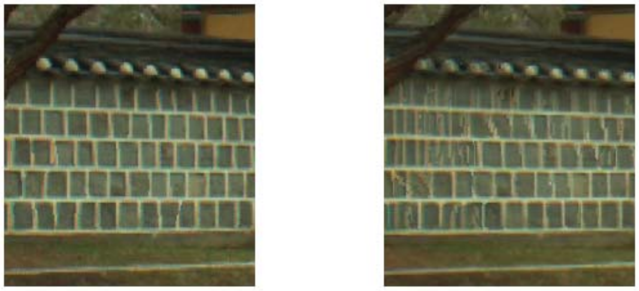

Fig. 11. Comparison of background objects (left: proposed, right: VSRS 3.5).

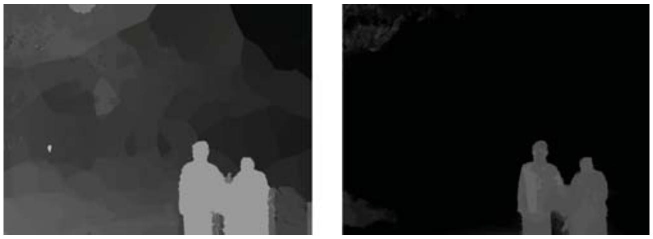

Fig. 12. Comparison of depth images at target view position (left: proposed, right: VSRS 3.5).

Viewpoint Immersive Networked Experience”.

6. REFERENCES

[1] Y. Mori, N. Fukushima, T. Fujii and M. Tanimoto, “View Generation with 3D Warping Using Depth Information for FTV,” *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pp.229-232, May. 2008

[2] C. Fehn, “A 3D-TV system based on video plus depth information,” *Signals, Systems and Computers, 2003*, vol.2, pp. 1529-1533, Nov. 2003

[3] N. Zefeng, D. Tian, S. Bhagavathy, J. Llach and B. S. Manjunath, “Improving The Quality of Depth Image Based Rendering for 3D Video Systems,” *IEEE International Conference on Image Processing*, pp. 513–516, Egypt, Nov. 2009.

[4] L. Pei-Jun and Effendi, “Adaptive Edge-Oriented Depth Image Smoothing Approach for Depth Image Based Rendering,” *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, China, Mar. 2010.

[5] L. Do, S. Zinger and Y. Morvan, “Quality improving techniques in DIBR for free-viewpoint video,” *3DTV-Conference*, Germany, 2009.

[6] P. Kumar and M. Flierl, “Depth consistency testing for improved view interpolation,” *IEEE International Workshop on Multimedia Signal Processing*, France, Oct. 2010.

[7] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60(2), pp. 91–110, 2004.

[8] S. Kumar, M. Kumar, B. Raman, N. Sukavanam, and R. Bhargava, “Depth Recovery of Complex Surfaces from Texture-less Pair of Stereo Images“, *Electronic Letters on Computer Vision and Image Analysis*, vol.8(1), pp. 44-56, 2009.

[9] M. Brown and D. Lowe, “Automatic Panoramic Image Stitching using Invariant Features,” *International Journal of Computer Vision*, vol. 74(1), pp. 59-73, 2007.

[10] M. Lalonde, D. Byrns, L. Gagnon, N. Teasdale and D. Laurendeau, “Real-time eye blink detection with GPU-based SIFT tracking,” *Fourth Canadian Conference on Computer and Robot Vision*, pp. 481-487, 2007.

[11] G. M. Um, G. Bang, N. Hur, J. Kim, and Y. S. Ho, “3d video test material of outdoor scene,” *ISO/IEC JTC1/SC29/WG11*, Rep. M15371, France, Apr. 2008.

[12] Y. S. Ho, E. K. Lee, and C. Lee, “Multiview video test sequence and camera parameters,” *ISO/IEC JTC1/SC29/WG11*,Rep. M15419, France, Apr. 2008.

[13] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, and Y. Mori, “Reference softwares for depth estimation and view synthesis,” *ISO/IEC JTC1/SC29/WG11*, Rep. M15377, France, Apr. 2008.