A DEPTH REFINEMENT ALGORITHM FOR MULTI-VIEW VIDEO SYNTHESIS

Hsin-Chia Shih and Hsu-Feng Hsiao

Department of Computer Science
National Chiao Tung University, Hsinchu, Taiwan
{sjshr, hillhsiao}@cs.nctu.edu.tw

ABSTRACT

With the recent progress of display, capture device, and coding technologies, multi-view video applications such as stereoscopic video, free viewpoint TV (FTV), and free viewpoint video (FVV) have been introduced to the world with growing interest. To achieve free navigation of such applications, depth information is required along with the video data. There have been many research activities in the area of depth estimation; however, it still poses us great challenge to estimate accurate depth map. In this paper, we propose a depth refinement algorithm for multi-view video synthesis. The proposed algorithm classifies the pixel-wise depth map into two categories, one is reliable and the other is unreliable, followed by the depth refinement algorithm for those pixels with unreliable depth values. Except for the depth refinement algorithm, we also propose a reliable weighted view interpolation algorithm. At last, the refined depth map is evaluated by the quality of the synthesized view.

Index Terms— FTV, multi-view, depth estimation, view synthesis

1. INTRODUCTION

Usually the traditional video applications only allow viewers to watch the scene passively. However, FTV significantly extend the sensation of classical 2D videos by enabling viewers to receive the video stream with the ability to freely change their viewpoints as if they were at the captured scene or/and the ability to provide a stereoscopic depth impression of the scene. In such scenario, the viewers can experience the free viewpoint navigation within the range covered by the cameras. To achieve free navigation functionality, depth information is required in addition to the video streams. The data representation, which is often called video-plus-depth, is shown in Figure 1.

Based on this configuration, the virtual view of arbitrary view angle can be synthesized by the video streams and the depth maps. The visual quality of the synthesized virtual view is highly related to the precision of the depth map.

Fig. 1. Video-plus-depth data representation.

Depth map can be generated by 3D depth cameras; however, such devices are not often seen and there are certain restrictions for the devices to acquire the depth map at high resolution. Alternatively, the depth map can be estimated by two or more videos captured at different angles conventionally. For the last two decades, stereo matching has been a well-known 3D depth sensing method [1]. However, occlusion area is an important issue of stereo matching [2][3]. The occlusion problem can be alleviated by using multi-view images. There are several algorithms using multi-view images as input to deal with the occlusion problem and obtain more accurate depth map. Segment-based approach is proposed in [4]. It assumes that the pixels in one object/segment shall have the same depth value. Pixel-based approaches are proposed in [5][6]. It estimates disparities first and then transforms the disparities into depth. Temporal consistency for depth estimation is also considered in [7][8][9]. The purpose of these researches is to estimate depth map at high resolution for multi-view video. However, there is still room for the improvement and the pixels with incorrect depth value will cause some artifacts in the synthesized virtual views. To get the better visual quality of virtual views, the method proposed in [10] detects the pixels with bad depth values, finds a better depth value for the pixels, and replaces the bad depth value with the better one.

The depth estimation methods in [4][5][6][7][8][9] follow the same camera setting as in [11]. This setting uses at least four cameras in order to estimate the depth map. In this paper, we use stereo camera setting to estimate the depth maps. The camera setting is illustrated in the left side of Figure 2 and one (*D_NR*) of the estimated depth maps for the views *NL* and *NR* is shown in the right side of Figure 2. This camera setting not only decreases the number of reference views available for the depth estimation but also avoids the edge-view problem of the depth estimation

algorithms with the camera setting in [11] where either the left-most or the right-most view has only one side of reference views. However, the occlusion issue still exists and the challenge becomes greater. Thus, a depth refinement algorithm is proposed in this paper to correct the unreliable depth values in the depth map, such as the occluded areas, before view synthesis is carried out. A reliable/unreliable weighting interpolation function is proposed to be included in the view synthesis algorithm [12] to further improve the visual quality of synthesized virtual views.

The rest of this paper is organized as follows. In Section 2 we describe our proposed method, followed by Section 3, the simulation results and discussions. The concluding remarks are presented in Section 4.

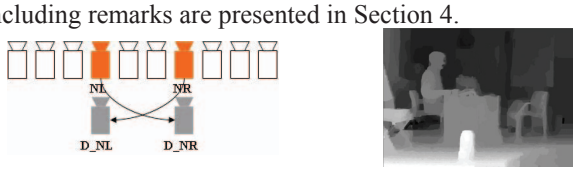

Fig. 2. Left: stereo camera setting, right: depth map estimated from stereo camera setting.

2. THE PROPOSED METHOD

The flow chart of the proposed depth refinement process and also the reliable weighted view synthesis algorithm are shown in Figure 3.

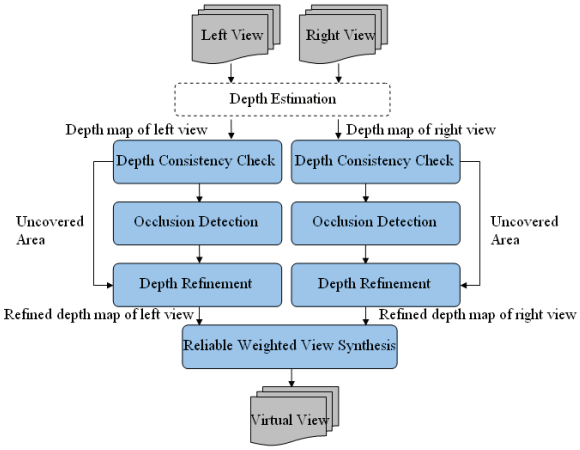

Fig. 3. Flow chart of the proposed algorithms.

The "Depth Estimation" in dotted line is the depth estimation algorithm proposed in [5]. Since there is only one reference view in the stereo camera setting, the error function *E(x,y,d)* used in [5] has to be modified as follows:

If the reference view is at left side, the error function is:

$$E(x,y,d) = E_L(x,y,d) \quad (1)$$

Otherwise, if the reference view is at right side:

$$E(x,y,d) = E_R(x,y,d) \quad (2)$$

where $E_L(x,y,d)$ is the intensity difference between the current view and left view, and $E_R(x,y,d)$ is the intensity difference between the current view and right view.

The details of each step in the depth refinement process are given below.

2.1 Depth consistency check

In this step, cross-checking is used to check whether the depth value of a pixel is reliable or unreliable. Cross-checking for the left view computes the matchness of the pixel position from the left view to the right view and then from the right view back to the left view based on the corresponding depth map before refinement. A pixel is marked as unreliable if it maps to a pixel that does not map back to it. The cross-checking for the right review is similar.

The result of the depth consistency check is shown as the depth consistency map in the right side of Figure 4. The pixels classified into unreliable are marked as white and the pixels classified into reliable are marked as black. The pixels marked as gray color are the pixels that can not find the corresponding pixels in the other view. These gray pixels are defined as "uncovered depth pixels".

2.2 Occlusion detection

By observing the depth consistency map and the original video data, some areas that have the similar texture could also be classified into unreliable. The areas marked by red ellipses in Figure 4 are examples of those areas with similar texture.

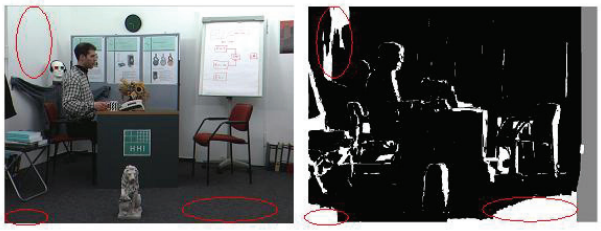

Fig. 4. The depth consistency map and areas with similar texture.

The refinement method for these similar texture areas is different from the method for the occluded areas in the proposed algorithm. Thus, the unreliable pixels need to be classified into occluded pixels and non-occluded pixels further.

We assume that if a pixel is occluded as illustrated in Figure 5, its depth value must be unreliable. If this depth value is used to find the corresponding pixel in the other view, there is a good chance that the depth value of the corresponding pixel is reliable. Based on this assumption and observation, the occlusion detection determines a pixel to be in the occluded area if this pixel is unreliable from the results of the depth consistency check and its corresponding pixel mapped by the unreliable depth should be viewable in the other view and thus should be reliable according to the cross-checking of the depth consistency.

Fig. 5. The occluded area in the left image can not be seen in the right image.

2.3 Depth refinement

The main idea for the refinement step is finding the closest pixels with reliable depth value and using the depth values of these pixels to interpolate the depth value of the unreliable pixel.

Four-neighbor interpolation of the depth value is proposed here. Not only the inverse proportion of distance but also the reliability of the found pixels with reliable depth is considered to calculate the interpolation weighting factors.

The consideration of reliability can be illustrated in Figure 6 where "a" and "c" indicate the reliable depth pixels that map to the unreliable depth pixels in the other view; "b" and "d" indicate the reliable depth pixels that map to the reliable pixels. Thus, reliability of the top and left reliable depth pixels should be lower than the right and bottom reliable depth pixels in this example.

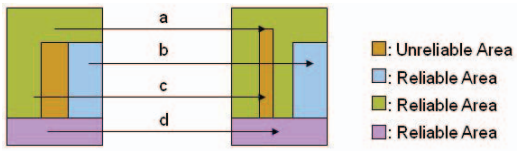

Fig. 6. The consideration of reliability.

Let WD_{Top}, WD_{Bottom}, WD_{Left}, and WD_{Right} be the weighting factors calculated by the distance from current unreliable depth pixel to the nearest reliable depth pixel in each direction. W_{HighR} and W_{LowR} are the weighting value of the reliable depth pixel with high reliability and the reliable depth pixel with low reliability, respectively. The calculation of the proposed interpolation weighting factor for each direction is described as follows:

If the reliable depth pixel is found to have high reliability:

$$WD_{Direction}' = WD_{Direction} \cdot W_{HighR} \tag{3}$$

Else

$$WD_{Direction}' = WD_{Direction} \cdot W_{LowR} \tag{4}$$

where "Direction" in lower index can be either Top, Bottom, Left, or Right.

After that, the interpolation weighting factor is normalized as $WD_{Direction}''$. The refined depth value of the unreliable depth pixel is interpolated by the following equation:

$$D_{Unreliable} = \sum_{Direction} D_{Direction} \cdot WD_{Direction}'', \tag{5}$$

where $D_{Unreliable}$ and $D_{Direction}$ are depth value of unreliable depth pixel and four-neighbor reliable depth pixels, respectively.

The calculation of the interpolation weighting factors described above serves as the basis of the proposed depth refinement algorithm with some modification for each of the uncovered area and occluded area, respectively. For the non-occluded area, Equation (3), (4), and (5) are used directly.

For the occluded area, the smallest depth value of the four-neighbor reliable depth pixels is chosen as a reference to filter out the neighbor in the foreground. The difference between this reference depth value and the depth value of neighbor in the other directions is then calculated. If the difference of any direction is larger than a threshold, the reliable depth pixel in this direction is considered as foreground and discarded.

For the uncovered pixel in the left view, the depth value of the reliable depth pixel on the right-hand side is regarded as its depth value, while the reliable depth pixel on the left-hand side is regarded as its depth value of the uncovered pixel in the right view.

The result of depth refinement is shown in Figure 7.

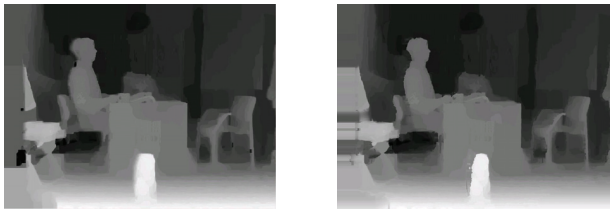

Fig. 7. Result of the depth refinement. Left: depth map before refinement, right: depth map after refinement.

2.4 Reliable weighted view synthesis algorithm

The reliable weighted interpolation function to refine the depth is described in the earlier section. A similar idea can be used in the view synthesis process.

If one of the mapped pixels is from the reliable depth pixel and the other is from the refined unreliable depth pixel, the weighting factors WD_{Left} and WD_{Right} of the view synthesis algorithm [12] are then modified by the same W_{HighR} and W_{LowR} described earlier and followed by the normalization of these factors before the view synthesis.

3. EXPERIMENT RESULTS

The quality of the refined depth map will be evaluated by the results of view synthesis. If the refined depth map has better resolution, the synthesized virtual views shall also have better quality.

The test sequences are from [13]. The reference views used to perform the depth estimation and the synthesized views for the evaluation are listed in Table 1.

The "DERS+VSRS" in the figures stands for the results of the modified depth estimation algorithm in DERS [14] and the view synthesis tool VSRS [12], while "[10]" means the results from the original depth map and the method described in [10]. These algorithms are proposed in the recent MPEG meetings. "Proposed" is for the results of the proposed algorithm with W_{HighR} =0.75 and W_{LowR}=0.25. The PSNR of the synthesized view is shown in Figure 8 for each of the test sequences.

The average PSNRs of the views are listed in Table 2. The improvement of the reliable weighted view synthesis can be seen in Table 3 where the performance of the proposed depth refinement algorithm with and without the proposed reliable weighted view synthesis algorithm is shown. To see the effect of the reliable weighted view synthesis better, only the pixels affected by the algorithm are considered to calculate the average PSNR in Table 3.

From the simulation results, the overall improvement brought by the proposed algorithms is quite substantial.

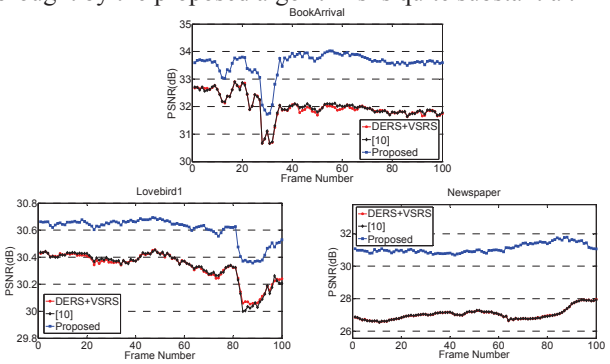

Fig. 8. PSNR performance of *BookArrival*, *Lovebird1*, and *Newspaper*.

4. CONCLUSION

Based on the results of simulation, it shows that our proposed depth refinement algorithm enhances the depth resolution well. The objective quality of the virtual views is improved significantly. Except for the depth map refinement, we also propose a view synthesis algorithm that results in noticeable improvement of quality by considering the reliable weighting factors.

5. REFERENCES

[1] D. Sharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", Proc. of IEEE Workshop on Stereo and Multi-Baseline Vision, pp. 131-140, December 2001.

[2] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions via graph cuts", ICCV, pages II: 508-515, 2001.

[3] J. Sun, Y. Li, S. Kang, and H. Shum, "Symmetric stereo matching for occlusion handling", CVPR, pages II: 399-406, 2005.

[4] S. Lee, K. Oh, and Y. Ho, "Segment-based Multi-view Depth Map Estimation Using Belief Propagation from Dense Multi-view Video", Proc. of 3DTV Conference, May 2008.

[5] M. Tanimoto, T. Fujii and K. Suzuki, "Multi-view depth map of Rena and Akko & Kayo", ISO/IEC JTC1/SC29/WG11, M14888, October 2007.

[6] M. Tanimoto, T. Fujii and K. Suzuki, "Improvement of Depth Map Estimation and View Synthesis", ISO/IEC JTC1/SC29/WG11, M15090, January 2008.

[7] S. Lee and Y. Ho, "Multi-view Depth Map Estimation Enhancing Temporal Consistency", ITC-CSCC, pp. 29-32, 2008.

[8] H. Yuan, Y. Chang, H. Yang, X. Liu, S. Lin, and L. Xiong, "Depth Estimation Improvement for Depth Discontinuity Areas and Temporal Consistency Preserving", ISO/IEC JTC1/SC29/WG11, M16048, January 2009.

[9] G. Bang, J. Lee, N. Hur, and J. Kim, "The consideration of the improved depth estimation algorithm: The depth estimation algorithm for temporal consistency enhancement in non-moving background", ISO/IEC JTC1/SC29/WG11, M16070, January 2009.

[10] J. Sung, Y. Jeon, J. Lim, and B. Jeon, "Improving view synthesis results based on depth quality", ISO/IEC JTC1/SC29/WG11, M16417, April 2009.

[11] C. Lee and Y. Ho, "Results of Exploration Experiment on View Synthesis", ISO/IEC JTC1/SC29/WG11, M15595, July 2008.

[12] M. Tanimoto, T. Fujii and K. Suzuki, "View Synthesis Algorithm in View Synthesis Reference Software 2.0 (VSRS2.0)", ISO/IEC JTC1/SC29/WG11, M16090, February 2009.

[13] "Call for 3D Test Material: Depth Maps & Supplementary Information", ISO/IEC JTC1/SC29/WG11, N10359, February 2009.

[14] M. Tanimoto, T. Fujii, and K. Suzuki, "Reference Software of Depth Estimation and View Synthesis for FTV/3DV", ISO/IEC JTC1/SC29/WG11, M15836, October 2008.

Table 1. Test sequences and virtual views to be synthesized.

Sequence Name	Left Reference	Right Reference	Virtual View
BookArrival	View 10	View 7	View 8
Lovebird1	View 5	View 8	View 6
Newspaper	View 4	View 6	View 5

Table 2. Average PSNR of each sequence.

Sequence Name	DERS+VSRS	[10]	Proposed
BookArrival	31.99	32.01	33.57
Lovebird1	30.33	30.33	30.60
Newspaper	27.04	27.05	31.10

Table 3. Improvement of reliable weighted view synthesis algorithm.

Sequence Name		BookArrival	Lovebird1	Newspaper
Avg. PSNR	w/o reliable weighting	31.161846	22.559836	26.400312
	with reliable weighting	31.696880	22.833418	26.602564