Edges shape enforcement for visual enhancement of depth image based rendering

Muhammad Shahid Farid , Maurizio Lucenteforte , Marco Grangetto

Dipartimento di Informatica, Università di Torino
Corso Svizzera 185, 10149 Torino, ITALY
`lastname@di.unito.it`

Abstract—**Depth image based rendering of intermediate views with high visual quality remains a challenging goal in presence of estimated and quantized depth values. Among the other rendering artifacts we observed that edges are usually affected by significant warping errors. In particular, because of depth estimation inaccuracy around object boundaries the edges may completely loose their original shape during the warping process. Nonetheless, edges represent one of the most important cues for the human visual system. In this paper a novel technique aiming at improving the edge rendering is presented. As opposed to previous approaches, the technique exploits only texture information, thus avoiding possible errors in depth estimation. The idea is based on the enforcement of prior knowledge of the edge shape under projective transformation. The proposed algorithm works in two steps: first the damaged edges of the warped image are detected, then these latter are corrected so as to better approximate their shape in the reference view. Finally the corrected edges are rendered within the intermediate image without introducing noticeable texture artifacts. The proposed algorithm has been tested on a variety of standard video sequences exhibiting excellent results in terms of rendered image visual quality.**

Index Terms—**3D-TV, Depth image based rendering, View synthesis, Edge Enhancement**

I. INTRODUCTION

Achieving high visual quality in the synthesized view is an important goal in 3D television along with optimal encoding and fast rendering. This goal turns into a challenging task when the depths are inaccurate [1] because of estimation errors, quantization and/or lossy compression.

A number of warping techniques have been proposed in the literature to create better virtual view. Almost all of them work in two phases: first, the original views are warped and fused together to generate an intermediate view; then, in the second step missing regions in the intermediate view are recovered by various inpainting methods. Usually, the first step takes two original views with their corresponding depths and generates the intermediate view by warping them pixel by pixel. The new position of each pixel depends on its depth and camera parameters. A small error in depth may cause a large variation in the final position of the pixel. The second phase in view synthesis is known as hole filling. A number of techniques varying from pixel level [2] to texture level [3] inpainting have been proposed. Solh [4] presented a hole filling algorithm that works in a hierarchical way along an image pyramid. Another

similar approach has been proposed in [5]. A hole filling technique that maintains a background model for each left and right view is proposed in [6]. These background models are computed using Gaussian mixture model and are used to fill the holes in the synthesized view. A similar technique that uses static background model was proposed in [7] in which occlusions were copied from the background model.

Since, depth estimation from images is an ill-posed problem, even the best depth based image rendering technique generally yields improper and distorted objects in the synthesized view. Since, not all depths are exact, a number of pixels are warped incorrectly. Moreover, the intermediate view must be sampled on integer pixel coordinates, whereas the warping process yields fractional coordinates. This latter issue is generally solved using fractional pixel precision and interpolation while warping. Previous shortcomings may potentially reduce the quality of the virtual view. It is well known that some locations in the virtual view remain empty because the corresponding pixels are "miswarped" to other positions. These empty locations are termed as *holes* which are estimated in the second phase. Wrong warping can also impair the quality of image objects, in particular along their edges, where depth estimation turns to be critical. While a lot of research has been done on hole inpainting, less attention has been devoted to the enforcement of important visual constraints in the virtual images such as preserving contours of objects. This paper presents an algorithm that, after creating the virtual view, identifies the distorted edges and corrects them. The proposed algorithm can be applied on top of any depth image based rendering technique and permits to enhance significantly the visual quality.

Various techniques related to our work have been proposed aiming at computing the virtual view and limiting the artifacts introduced by the warping process. Lee and Ho [8] proposed a technique to remove the ghost that sometimes appear around the background object boundary. The missing texture in these ghost regions is replaced from texture of a reference image. A view synthesis technique that works for two videos captured with uncalibrated cameras with their depths is proposed in [9]. The geometric transformation between the scenes is automatically computed and then the scene is segmented using graph cut algorithm and finally warping is applied. In [10] depth map is divided into layers aiming at minimizing the

rendering artifacts. This permits to check that every warped pixel is determined only by pixels in the same depth layer. Many other view synthesis techniques that divide the depth into layers and then warp the pixels using these layers have been proposed [11], [12], [13].

Our contribution follows from the simple observation that depth estimates are usually noisy especially along the edges of the objects. Such errors may in turn impair the edges of the final virtual view reducing the quality of objects contours that are known to yield a significant contribution to visual perception. As a consequence, as opposed to the related literature that tries to improve visual quality working on depths, e.g. dividing it into regions or layers to enforce coherent object warping, in this paper we focus specifically on edges. Our goal is to improve the visual quality of objects in the virtual image enhancing the warped edges using only texture information independently of their depth, that may be potentially inaccurate. Instead, we exploit the prior knowledge that straight lines and edges are preserved under perspective transformation and therefore we guarantee that edge shape is not altered too much as a consequence of warping. In particular, the proposed algorithm is able to detect edges that have been warped imperfectly, to restore their shape and to blend accurately the restored edges to the surrounding texture.

The rest of the paper is organized as follows: in Sect. II depth based image rendering techniques are briefly recalled. In Sect. III the proposed algorithm is described, whereas Sect. IV presents and discusses our experimental evaluation on a number use cases. Our conclusions are drawn in Sect. V.

II. Depth image based rendering

As already mentioned, in this paper we do not focus on a particular rendering technique since the proposed algorithm is a post processing stage for improving visual quality. Without loss of generality in the following we assume to work on top of an intermediate view generation algorithm with typical experimental settings.

Let V_l and V_r be the two input views with corresponding depths D_l, D_r which are quantized into 8 bit integers. Let Z_n and Z_f be the nearest and farthest depth respectively. We assume horizontal camera setup with camera focal length f where b_l, b_r are the positions of the left and the right camera, respectively. Usually $b = b_r - b_l$ is referred to as cameras baseline. Let V_m be the intermediate virtual view that one is willing to estimate. The position of the virtual camera for V_m would be $b_m = b_l + \frac{b}{2} = b_r - \frac{b}{2}$. In order to generate V_m, the pixels of left view are warped to the target intermediate virtual camera; similarly, the right view is warped to the intermediate view. Then the two obtained virtual views are merged to get the final virtual view. The two intermediate views usually complement each other and allow to fill most occlusions. Let V_l' be the left virtual view. Because of the horizontal camera arrangement V_l' is obtained from V_l by applying horizontal shifts to pixel coordinates. Given pixel position $(u, v) \in V_l$, the warped position (u', v') in V_l' is such that $u' = u$ and $v' = v - \delta$. The column shift δ is computed as:

$$\delta = \frac{bf}{d} \tag{1}$$

where d is the depth of the original pixel with coordinates (u, v). Depths are generally provided in the form of quantized disparity (i.e. inverse depth). We use notation D_l (D_r) to identify such quantized depth maps. The actual depth d can be computed as follows:

$$d = \frac{1}{\frac{D_l}{255}\left(\frac{1}{Z_n} - \frac{1}{Z_f}\right) + Z_n} \tag{2}$$

Similarly, the right virtual view V_r' is computed (for right virtual view δ is added to the original column number to get the warped position). Finally, the two virtual views are merged together to get the intermediate view V_m and the holes are recovered by some inpainting algorithm.

In this paper we will use the *View synthesis Reference Software* [14] provided by the MPEG standardization committee as a benchmark for intermediate view synthesis performance.

III. Proposed Edge Preserving Image Based Rendering

It has been observed that the warping process often destroys the sharp boundaries of the objects, resulting in poor visual quality. An image rendering algorithm that preserves the edges of the image objects is desired. This section describes our algorithm that enforces prior knowledge of the fact that object contours are expected to exhibit approximatively the same shape in the original (left or right view) and warped views. As an example straight lines are invariant under perspective transformation and therefore one knows that lines in the left view must be re-projected as lines in the warped image. Similarly, a general edge must be warped without dramatically changing its shape.

The proposed technique comprises four steps. In the first step the most important boundaries (edges) of the objects are detected using a single view, e.g. the left picture. In the second step the intermediate view is created using a standard warping algorithm, e.g. VSRS. Next, the edges detected during the first step are warped to their intermediate position allowing us to detect the set of edges damaged by depth based warping. Finally, the impaired edges are corrected enforcing prior knowledge about their shape according to the original view. To improve the visual quality a proper blending operator is applied to ensure both edge sharpness and coherent texture.

A. Creating the edge list

The quality of the resultant video depends on two factors. First, how finely edges are detected and second to what extent the errors introduced in the edges by the warping have been corrected. A number of techniques exists to detect the edges in the images like simple and fast Sobel and Prewitt kernels and the more complex Canny edge detector [15]. This latter has been used in our proposal for its superior performance. Edge detection can be performed on both texture image and depth map. At a first glance the depth map seems a good

option since each object in the image can be theoretically discriminated based on depth. Nonetheless, depth map is estimated and quantized in our case and therefore cannot be trusted for locating edges. Indeed depth values are often erroneously estimated especially along objects boundaries. As a consequence it is more safe to apply edge detection to the texture image. Moreover, to save memory and computational costs edges can be extracted by a single view, e.g. the left view, without noticeable impairments of the whole algorithm.

Let vL be the left view and in particular, Canny detector with threshold $[\tau_{low}, \tau_{high}]$ is run on the left view V_l to detect the set of edges $\mathcal{E}$, where $\mathcal{E}_j$ denotes the j-th edge in the set and $e_j(i) = (e_j^r(i), e_j^c(i)) = (u, v)$ is the i-th point of the j-th edge with pixel coordinates (u, v). Since the edge detector may return very short edges, that do not have a strong impact on the overall image quality, the shorter edges are removed from $\mathcal{E}$ to reduce the computational effort of the next steps. Hence, all edges with length $|\mathcal{E}_j| \leq \gamma$, γ being a predefined threshold, are discarded.

The edge list $\mathcal{E}$ is then transformed into the warped edge set $\widetilde{\mathcal{E}}$, by applying warping to the corresponding edge pixel coordinates.

B. Edge error estimation and correction

As already discussed above, the warping process may introduce some errors and distort the smoothness of the edges. In order to counteract this issue, the proposed algorithm detects the edges damaged by warping, then locates the edge points affected by error and finally applies edge correction.

Damaged edges are detected by analyzing the gradient of edges before and after warping. Given an edge point $e_j(i)$ its gradient $\nabla_j(i)$ is defined as:

$$\nabla_j(i) = \left(\nabla_j^r(i), \nabla_j^c(i)\right) = e_j(i) - e_j(i+1)$$

Similarly, $\widetilde{\nabla}_j(i)$ is computed for the warped edge point $\widetilde{e}_j(i)$. Before warping, edges are formed by connected points by construction and therefore it must hold that $|\nabla_j^{r/c}(i)| \leq 1$, i.e. horizontal and vertical coordinates of two consecutive edge points must be in adjacent positions. We observed that, for limited values of the camera baseline b, edges are characterized by similar gradients before and after warping. On the contrary, non correct warping significantly modifies the edge gradient and some edge points are no longer connected, i.e. $|\widetilde{\nabla}_j^{r/c}(i)| > 1$. To detect anomalous edge warping we compute the gradient difference $\widetilde{\nabla}_j(i) - \nabla_j(i)$. Since warping does not change the row index u it follows that:

$$\begin{aligned} \widetilde{\nabla}_j^r(i) - \nabla_j^r(i) &= 0 \\ \widetilde{\nabla}_j^c(i) - \nabla_j^c(i) &= \widetilde{e}_j^c(i) - \widetilde{e}_j^c(i+1) - e_j^c(i) + e_j^c(i+1) \end{aligned}$$

For every edge the sum of absolute difference of the horizontal gradient before and after warping $\mu_j = \sum_i |\widetilde{\nabla}_j^c(i) - \nabla_j^c(i)|$ is used as a metric to detect a damaged edge. In particular the j-th edge undergoes the following correction procedure if $\mu_j > \lambda$, where λ is a proper threshold.

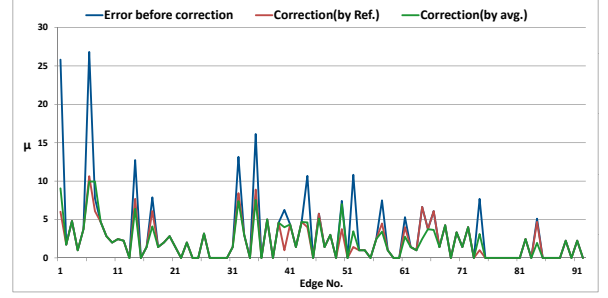
Fig. 1: Error metric μ_j evaluated on warped edges in one frame of Poznan_Hall2 sequence before and after correction (by smoothing and reference).
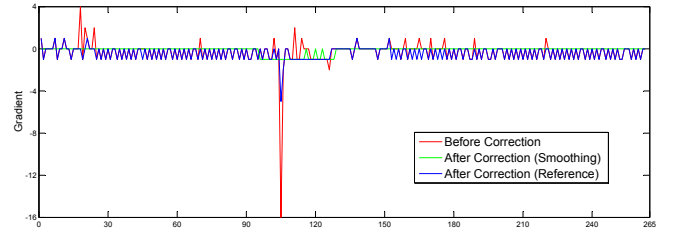

Fig. 2: Values of $|\widetilde{\nabla}_j^c(i)|$ along an edge of one frame of Poznan_Hall2 sequence before and after correction by smoothing.

As an example, in Fig. 1 the metric μ_j is reported for a frame of the Poznan_Hall2 test sequence and shows that only a few edges exhibit large values making them candidate for correction.

Once the j-th edge is classified as damaged, it is necessary to locate the points that actually require correction. Based on previous observations on the edge gradient a point of the warped edge is considered in error if the magnitude of horizontal component of the gradient of the i-th point is larger than 1, $|\widetilde{\nabla}_j^c(i)| > 1$, i.e. the warped edge looks disconnected around the i-th point. Then, two alternative correction strategies described in the following are applied to every selected point.

1) Correction by smoothing: In this case we aim at correcting the edge by smoothing the anomalous values of $\widetilde{\nabla}_j^c(i)$. This is done applying a linear running average filter of size ω to $\widetilde{\nabla}_j^c(i)$ and the obtained values are used to correct the locations of the edge points. Our experiments show that using $10 \leq \omega \leq 25$ one gets good results. Fig. 1 shows $\widetilde{\nabla}_j^c(i)$ for a certain edge before and after the correction. Clearly, the smoothing approach removes the spike corresponding to miswarped edge points.

2) Correction by reference: Another strategy to correct the location of i-th point of a damaged edge $\widetilde{\mathcal{E}}_j$ is to correct the anomalous values of the warped gradient according to the gradient of the original edge $\mathcal{E}^j$. Given the selected i-th point of the edge, its corrected location is estimated by applying the gradient of the original edge to the preceding edge point
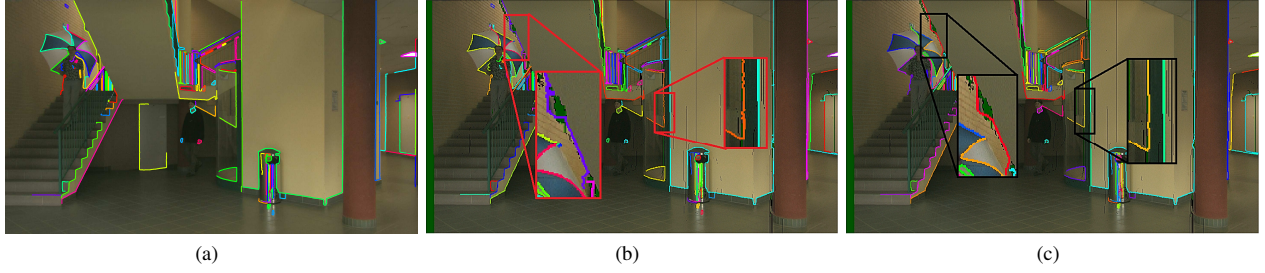
Fig. 3: Edge correction operations: edges extracted from left view (a), warped image with damaged edges (b), warped image with corrected edge locations (c).

according to the equation:

$$\hat{e}_j(i) = \widetilde{e}_j(i-1) + \nabla_j(i) \qquad (3)$$

Since usually the warping error affects a consecutive set of edge points, erroneous point identification and correction by reference are applied recursively until no more errors can be detected along the edge.

Both the proposed correction approaches are able to improve the edge quality in terms of the proposed error metric μ_j as shown in Fig. 1, where the values of the metric before and after the correction are reported for a given frame of a test sequence. In the experimental section we will show that these results are confirmed in terms of better visual quality along the contours of the warped objects.

The described algorithmic steps are represented visually in Fig. 3, where the edges $\mathcal{E}$ extracted from the left view (a), their warped counterparts $\widetilde{\mathcal{E}}$ (b) and their corrected profiles (c) are superimposed to a test image.

C. Edge enforcement in virtual view

The corrected edges must be finally rendered in the intermediate view by properly integrating them with the surrounding textures without introducing visual artifacts. Indeed previous algorithmic steps are used only to correct the edge's location without taking care of the edge's neighborhood.

Therefore, in the final stage of the proposed technique each edge is transferred into the intermediate view with its k-neighbors on both edge sides, i.e. image patches along the edge are blended so as to create the corresponding texture. In particular, the neighboring pixels are blended in the warped virtual view using weights that depends on the distance from the corrected edge. A weighting kernel w of size $(2k+1) \times (2k+1)$ is defined as:

$$w(u,v) = 1 - \frac{1}{\rho} 2\log(\sqrt{u^2 + v^2} + 1) \qquad (4)$$

where ρ is a parameter representing a lower bound on the diffusion weight ($0 < \rho \le 1$). Fig. 4 shows an example of 15×15 ($k = 7$) weighting kernel with $\rho = 0.8$.

The final correction stage takes as input the edge lists $(\mathcal{E}, \widetilde{\mathcal{E}})$, the original left view V_l and the warped intermediate view V_m and outputs an edge enhanced intermediate view V'_m. First of
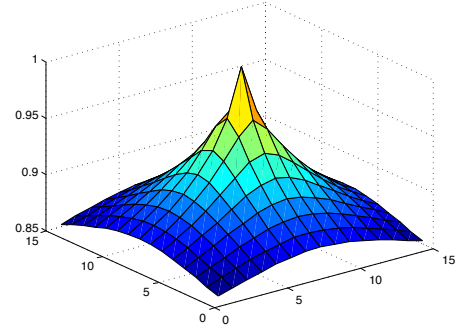

Fig. 4: Proposed weighting kernel $w(u,v)$ with $k = 7$ and $\rho = 0.8$.

all, we initialize $V'_m = V_m$, then every edge point $\widetilde{e}_j(i)$ is corrected by copying the corresponding point from V_l:

$$V'_m(\widetilde{e}_j(i)) = V_l(e_j(i))$$

Blending of the area surrounding every edge point is performed by applying the following equation:

$$V'_m(\widetilde{e}^r_j(i) + x, \widetilde{e}^c_j(i) + y) = V_l(e^r_j(i) + x, e^c_j(i) + y)w(x,y)$$
$$+ V_m(\widetilde{e}^r_j(i) + x, \widetilde{e}^c_j(i) + y)(1 - w(x,y))$$

for $-k \le x \le k$, $-k \le y \le k$, subject to $(\widetilde{e}^r_j(i) + x, \widetilde{e}^c_j(i) + y) \notin \widetilde{\mathcal{E}}$ (the last check is used to avoid smoothing previously copied edge points).

In Fig. 5 we show the whole algorithm steps and its integration with a standard depth based image rendering tool.

IV. VISUAL RESULTS

The proposed algorithm has been implemented on top of the MPEG VSRS and tested on several standard video plus depth sequences. The experiments show that the proposed technique is able to improve significantly the visual quality of the images rendered by VSRS.

The quality of edge enhancement depends on a number of parameters. The first design choice is related to the Canny threshold values. In our experiments we found that setting $\tau_{low} = 0.05$ and $\tau_{high} = 0.15$ performs well on all datasets. The selection of the minimum length of edges to be corrected
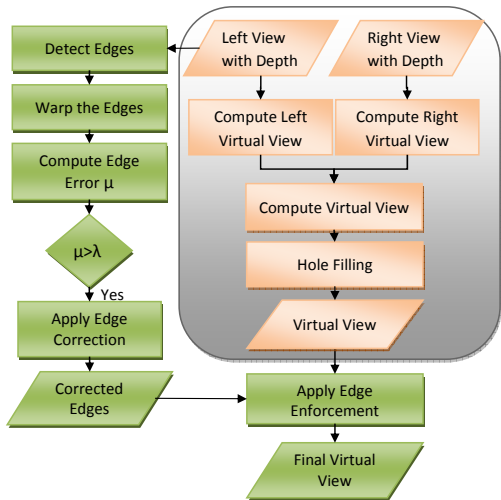
Fig. 5: Proposed technique flowchart.

TABLE I: Experimental settings: CM is used to identify the correction method by smoothing (s) and reference (r), respectively.

Experiment	Views	γ	$[\tau_{low} \ \tau_{high}]$	λ	CM	$[k \ \rho]$
Poznan_Hall2	$5,3 \rightarrow 4$	250	[0.05 0.15]	5	s	[10, 0.5]
Poznan_Street	$5,7 \rightarrow 6$	400	[0.05 0.15]	5	r	[10, 0.3]
Balloons	$5,1 \rightarrow 3$	300	[0.05 0.15]	5	s	[7, 0.3]
Newspaper	$2,6 \rightarrow 4$	350	[0.05 0.15]	5	s	[7, 0.3]

is another important parameter allowing to spend the computational efforts only on visually important features. In our case we have found that setting $\gamma \geq 250$ strikes a balance between image quality and computational cost. One of the most important choices is represented by the error threshold λ that allows to detect and correct only the visually impaired edges. In all our experiments we have set $\lambda = 5$. The size of the edge blending kernel k is also very important from the point of view of the final image quality. We have found that setting k around 10 pixels yields good results for the tested sequences but clearly the optimal setting depends on image contents and resolution. In Tab. I all the experimental settings used in our evaluation are summarized along with the video sequences and view synthesis experiment details.

Fig. 6 shows the results obtained on one frame of the Poznan_Hall2 test sequence. The red box identifies an area where the edge enhancement produced by the proposed technique is very noticeable. Excellent results have been obtained on other sequences as shown in Fig. 7,8,9. As noted in Tab. I we have employed edge correction by reference on the Ponznan_Street sequence whereas smoothing is used in the other cases. It is worth pointing out that both approaches yield very similar results in presence of edge artifacts of limited length. In presence of long edge errors smoothing is less effective than the other method. On the other hand, correction by reference works only if the first point is not affected by warping errors.
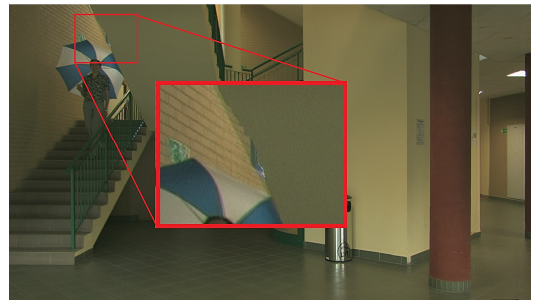
(a) Virtual View (VSRS)
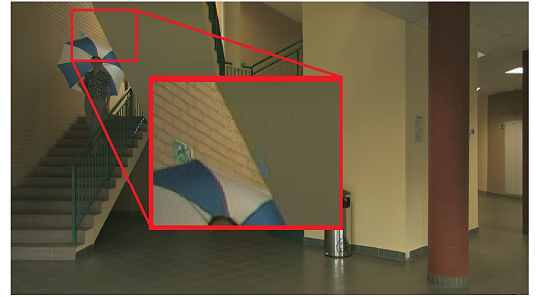
(b) Virtual view (Proposed)

Fig. 6: Performance comparison on frame 46 of Poznan_Hall2 sequence.

V. CONCLUSIONS

An algorithm to maintain the shape of the contours of the objects in synthesized view has been presented. The proposed technique identifies edges that are affected by warping errors due to synthesis or depth errors and estimates their correct shape. The edge shape is recovered using information from the texture in the reference views only, thus avoiding to use depth values that are potentially error prone especially around object edges. The edges are recovered enforcing a given level of regularity and assuming prior knowledge on the modifications admissible under projective geometry. Moreover, a proper blending strategy allowing to reshape the edges without causing visible artifacts in the surrounding texture has been designed. The proposed approach does not depend on a particular synthesis algorithm and can be used as a post-processing stage to enhance the visual quality of virtual views. Our experiments show that the proposed approach is able to enhance the warped edges with excellent visual results.

REFERENCES

[1] Emilie Bosc, Muriel Pressigout, and Luce Morini, "3d video: new techniques and challengese," in *First Sino-French Workshop on Education and Research collaborations in Information and Communication Technologies*, 2011.

[2] V. Paradiso, M. Lucenteforte, and M. Grangetto, "A novel interpolation method for 3d view synthesis," in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, Oct., pp. 1–4.

[3] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand, "Depth image-based rendering with advanced texture synthesis for 3-d video," *Multimedia, IEEE Transactions on*, vol. 13, no. 3, pp. 453–465, 2011.
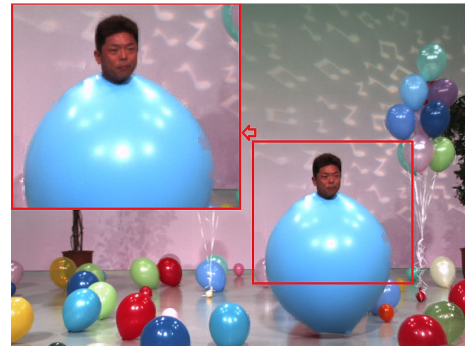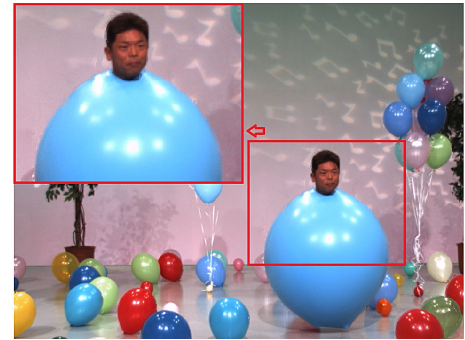
(a) Virtual View (VSRS)
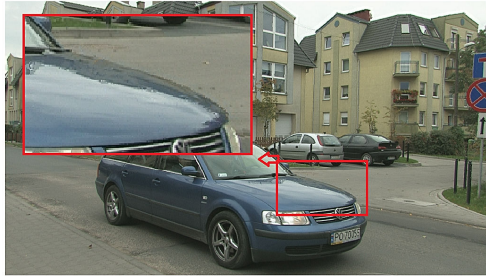
(b) Virtual view (Proposed)

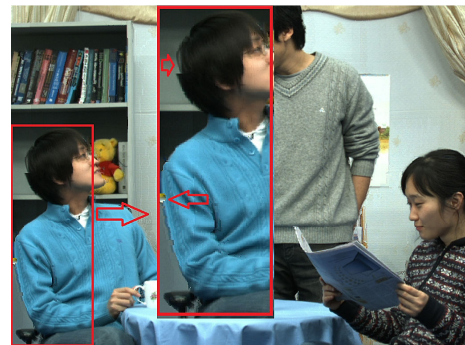Fig. 7: Performance comparison on frame 241 of Poznan_Street sequence.
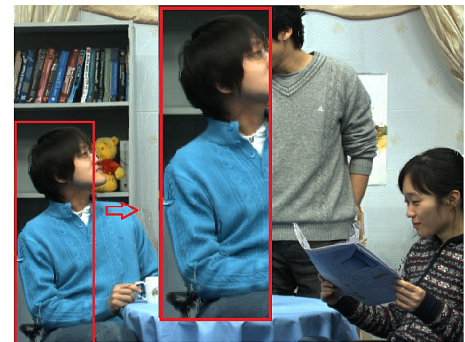

(a) Virtual View (VSRS)

(b) Virtual view (Proposed)

Fig. 8: Performance comparison on frame 296 of Balloons sequence.

(a) Virtual View (VSRS)

(b) Virtual view (Peoposed)

Fig. 9: Performance comparison on frame 220 of Newspaper sequence.

[4] M. Solh and G. AlRegib, "Hierarchical hole-filling for depth-based view synthesis in ftv and 3d video," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 6, no. 5, pp. 495–504, 2012.

[5] M.S. Farid, H. Khan, and A. Mahmood, "Image inpainting based on pyramids," in *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, Oct., pp. 711–715.

[6] Chao Yao, Yao Zhao, and Huihui Bai, "View synthesis based on background update with gaussian mixture model," in *Advances in Multimedia Information Processing PCM 2012*, Weisi Lin, Dong Xu, Anthony Ho, Jianxin Wu, Ying He, Jianfei Cai, Mohan Kankanhalli, and Ming-Ting Sun, Eds., vol. 7674 of *Lecture Notes in Computer Science*, pp. 651–660. Springer Berlin Heidelberg, 2012.

[7] M. Schmeing and Xiaoyi Jiang, "Depth image based rendering: A faithful approach for the disocclusion problem," in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*, 2010, pp. 1–4.

[8] Cheon Lee and Yo-Sung Ho, "View synthesis using depth map for 3d video," in *Proceedings of 2009 APSIPA Annual Summit and Conference*, 2009, pp. 4–9.

[9] Jiangjian Xiao, *Image Based View Synthesis*, Ph.D. thesis, University of Central Florida Orlando, Florida, 2004.

[10] Ying Chen, Weixing Wan, M.M. Hannuksela, Jun Zhang, Houqiang Li, and M. Gabbouj, "Depth-level-adaptive view synthesis for 3d video," in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, 2010, pp. 1724–1729.

[11] T.S. Huang and Chao Zhang, "A layered method of visibility resolving in depth image-based rendering," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.

[12] Zefeng Ni, Dong Tian, S. Bhagavathy, J. Llach, and B.S. Manjunath, "Improving the quality of depth image based rendering for 3d video systems," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 2009, pp. 513–516.

[13] N.A. Manap and J.J. Soraghan, "Novel view synthesis based on depth map layers representation," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, 2011, pp. 1–4.

[14] ISO/IEC JTC1/SC29/WG11 (MPEG), "View synthesis reference software (vsrs) 3.5," Mar. 2010.

[15] J Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, June 1986.