EDGE ENHANCEMENT OF DEPTH BASED RENDERED IMAGES

Muhammad Shahid Farid, Maurizio Lucenteforte, Marco Grangetto

Dipartimento di Informatica, Università degli Studi di Torino
Corso Svizzera 185, 10149 Torino, Italy

ABSTRACT

Depth image based rendering is a well-known technology for the generation of virtual views in between a limited set of views acquired by a cameras array. Intermediate views are rendered by warping image pixels based on their depth. Nonetheless, depth maps are usually imperfect as they need to be estimated through stereo matching algorithms; moreover, for representation and transmission requirements depth values are obviously quantized. Such depth representation errors translate into a warping error when generating intermediate views thus impacting on the rendered image quality. We observe that depth errors turn to be very critical when they affect the object contours since in such a case they cause significant structural distortion in the warped objects. This paper presents an algorithm to improve the visual quality of the synthesized views by enforcing the shape of the edges in presence of erroneous depth estimates. We show that it is possible to significantly improve the visual quality of the interpolated view by enforcing prior knowledge on the admissible deformations of edges under projective transformation. Both visual and objective results show that the proposed approach is very effective.

***Index Terms*—** View synthesis, depth image based rendering, 3DTV, quality enhancement

1. INTRODUCTION

Intermediate view synthesis using depth information has recently received much research efforts due the high impact that is expected to provide in the area of future 3D video compression and display technologies. Recently, Multiple View plus Depth (MVD) has been selected as the most promising 3D video data format for an efficient representation of several views. In fact, novel autostereoscopic displays require many views, up to 64, and it turns to be unfeasible to efficiently represent such large amount of information using only video data. MVD format has emerged since a limited set of views plus the corresponding depth maps can be encoded efficiently and then used on the display side to synthesize a dense set of views using Depth image based rendering (DIBR) [1]. The MPEG standardization group has started working on future 3D video compression [2] and is currently using MVD as raw 3D video format and DIBR as possible tools to improve coding efficiency.

DIBR can be used to generate novel views from a set of existing views for immersive 3D experience. The visual quality of the generated views largely depends upon the quality of the corresponding depth map in addition to the rendering method and the video coding technology. The depth of the views is usually estimated using stereo matching algorithms [3, 4] and is usually quantized to 8 bits. Moreover, depth maps quality can be further impaired by compression used in the transmission chain. Since depth information is used to shift pixels when changing the camera view point, depth errors impact on the quality of the estimated virtual view. All previous remarks point out that depth values can be inaccurate and can cause different artifacts in synthesized view e.g., structural distortion [5], ghost effect and holes.

We observed that depth maps are inaccurate especially around image edges. In fact, the depth estimation process is particularly critical when coping with sharp depth gradient that happens when moving from an object contour to the surrounding background. On the other hand, also compression artifacts are known to impair sharpness of edges in the depth map [6]. In presence of inaccurate depth maps some edge pixels are warped to wrong coordinates. This results in *structural distortion* - distortion in the shape of the objects - and causes significant loss in the perceived quality of the synthesized view since edges have a great influence on human visual perception.

Various techniques have been proposed in the literature to improve the quality of the depth map [7, 8] to avoid structural distortion in synthesized view, but success has been limited. Much less attention has been devoted to enforce constraints on visually important attributes like contours and edges in the virtual view. In [9] we investigate for the first time the possibility to improve the quality of the synthesized image by focusing on the properties of edges. In [9] a correction suited only for edges lying on the same depth level has been proposed. In this paper, we design a more general approach that aims at maintaining the structural integrity and shape of the edges of objects in the synthesized view by utilizing the prior texture information from the input views. The proposed algorithm is able to correct all kind of edges regardless of their orientation and depth they span in the image.

2. PROPOSED ALGORITHM

The proposed algorithm improves the quality of the rendered view by enhancing the rendered edges, obtained by warping selected contours inferred solely by the texture of the input views. The idea is to enforce geometrical constraints that edges must satisfy under projective geometry as detailed in the following. The change in the shape of an edge undergoing DIBR is a function of the disparity map D, the camera intrinsic parameters (focal lenght, f) and baseline b (see Eq. 1). It is worth recalling that b is usually very limited in scenario where multiple views are captured for 3D TV applications. For given intermediate view, f and b are constant and the edge shape is changed as a function of the depth map D (Eq. 1).

Let us briefly describe the DIBR process. Let V_l and V_r be the left and right views with associated depth maps D_l and D_r, respectively. Typical horizontal camera setup is assumed with b_l and b_r being the positions of the left and the right camera. Let V_m be the virtual view in position $b_m = \frac{(b_r+b_l)}{2}$ to be estimated. The left view V_l is warped to b_m using DIBR computing $V_l'(u,v) = V_l(u, v-\Delta)$, where Δ is the column shift and is evaluated as:

$$\Delta = \frac{bf}{D_l(u,v)} \tag{1}$$

with baseline $b = b_m - b_l$. It must be noted that the row coordinate remains unchanged during the warping as the cameras are in horizontal setup. Similarly, the V_r' is computed by warping V_r to virtual viewpoint b_m. The two views V_l' and V_r' are merged together to obtain V_m and analogously the depth D_m is computed. The holes in the V_m are filled through inpainting [10].

The first constraint that a given edge is expected to satisfy is that its warped version, i.e. its warped pixels still form an edge. This condition can be checked by controlling that shifting an edge according to (1): i) we still obtain a connected warped edge and ii) the structure and shape of the original and warped edge remain very similar as far as b is limited. To this end we define a structural distortion metric that, by comparing the coordiantes of the pixels of a given edge before and after warping, aims at detecting those that are significantly miswarped. In the following, we will consider two different situations as shown in Fig. 1, namely edges lying approximatively on the same depth level (*constant depth edges*) and edges spanning a significant depth range (*varying depth edges*). In the first case all edge pixels undergo the same translation and therefore the warped edge must be a rigid shift of the original one. In the second case the depth varies and the edge represented under a different perspective can be obtained by affine transformation of the original. These constraints will be used to recover the correct position of edges that exhibit a large structural distortion. Finally, the corrected edges are enforced in the novel synthesized view by proper blending with the surrounding texture. The details of the proposed algorithms are are described in the following subsections.

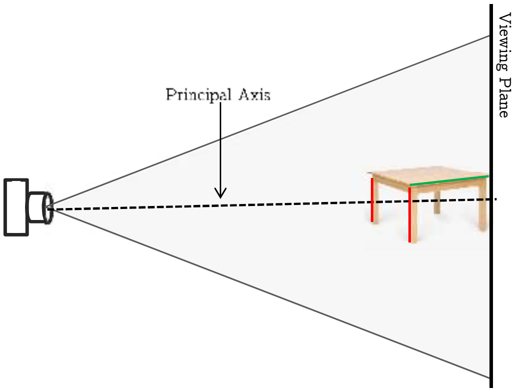


Fig. 1: Red edges are examples of 1st case (constant depth edges) and the edge marked green is an example of 2nd case (varying depth edges).

2.1. Edge detection

The first step is the detection of the dominant edges in the scene. To this end we should detect (and match) edges using both the left and the right view. Nonetheless we noted that, for limited b, effective results can be obtained at a much lower computational cost by detecting edge from a single view, e.g. V_l in our experiments. In our implementation Canny detector [11] with thresholds $[\tau_{low}, \tau_{high}]$ is used to detect the edges from V_l. Let $\mathcal{E}$ be the set of detected edges in one frame and e_j be the j-th edge in the set. We use notation $e_j(i) = (e_j^r(i), e_j^c(i))$ to refer to i-th coordinate (row, column) of j-th edge in $\mathcal{E}$. Since short edges have very little impact on the whole image quality they are pruned from $\mathcal{E}$ to limit further the computational cost. Hence, the edges with length $|E_j| \leq \gamma$ are removed from $\mathcal{E}$, γ being a predefined threshold.

2.2. Edge structural distortion

The edge impairments introduced by warping with incorrect depth must be first detected and then corrected. To this end, the edges coordinates selected in the previous step are warped to their virtual point and a proper metric is defined to measure the edge structural distortion. Let $\widetilde{e}_j$ represents the warped coordinates of edge e_j. We define the structural distortion μ_j by comparing the shape of the original and warped edges, respectively. In order to match the shapes of the two edges their coordinates are expressed with respect to a reference point that is computed as the centroid of the edge. Namely, we define

$$c_j(c_j^r, c_j^c) = (\mathcal{M}(e_j^r(1), \cdots, e_j^r(n_j)), \mathcal{M}(e_j^c(1), \cdots, e_j^c(n_j))) \tag{2}$$

where n_j is the number of pixels in the edge and $\mathcal{M}$ is the median operator. The centroid $\widetilde{c}_j$ of the warped edge is computed analogously. The centroids are estimated using the median (in place of mean operator) to reduce the bias produced by the shift errors in $\widetilde{e}_j$. Then, the edge warping error is computed as

$$\delta_j(i) = (e_j(i) - c_j) - (\widetilde{e}_j(i) - \widetilde{c}_j) \tag{3}$$

Given the horizontal camera arrangement the row coordinates of the original and warped edges are the same and can be clearly omitted from previous computation. Finally, the edge structural distortion is defined as

$$\mu_j = \sum_{i=1}^{n_j} |\delta_j(i)| \tag{4}$$

Any edge that has been warped correctly is expected to have limited μ_j, whereas an edge with $\mu_j > \lambda$ is selected for possible correction, being λ a suitable constant.

Moreover, the edge warping error (3) can be used to identify the wrong edge coordinates. From our experiments it turns out that simple shareholding of δ_j is not reliable enough. Therefore, we use a criterion based on the standard deviation of δ_j,

$$\sigma_j = \sqrt{\frac{1}{n_j - 1} \sum_{i=1}^{n_j} (\delta_j(i) - \overline{\delta}_j)^2} \tag{5}$$

with $\overline{\delta}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \delta_j(i)$. The edge coordinates with warping error $\delta_j(i) \in [\overline{\delta}_j - \sigma_j, \overline{\delta}_j + \sigma_j]$ are classified as *correctly warped pixels* whereas the complementary ones are marked as *"miswarped" pixels* . In the following we use the notation C_j and M_j to represent the set of integers indexing coordinates of correct and miswarped pixels of the j-th edge, respectively.

2.3. Edge error correction

Before applying edge correction we need to discriminate between constant depth and varying depth cases. To this end, the sum of depth gradient of the correct edge coordinates is computed as follows:

$$\nabla_j = \sum_{i \in C_j} |d_j(i) - d_j(i+1)| \tag{6}$$

where $d_j(i)$ is the depth of the i-th coordinate of edge j. An edge is considered as a constant depth one if $\nabla_j \leq \Delta$. The correction strategy for constant depth edges is straightforward and amounts to subtracting the warping error $\delta_j(i)$ from the warped edge. The corrected edge $\hat{e}_j$ is computed as:

$$\hat{e}_j(i) = \begin{cases} \widetilde{e}_j(i) - \delta_j(i), & \text{if } i \in M_j \\ \widetilde{e}_j(i), & \text{otherwise} \end{cases} \tag{7}$$

In case of varying depth edge, i.e. $\nabla_j > \Delta$, the correction requires the estimation of the geometrical relationship between the original and the warped edges. To this end, we estimate the affine transformation characterizing the correctly warped coordinates of the edge by solving

$$H = \underset{H}{\operatorname{argmin}} \sum_{i \in C_j} \left| [e_j(i), 1]^\top - H \cdot [\widetilde{e}_j(i), 1]^\top \right| \tag{8}$$

where H is the 3×3 affine matrix or homography

$$H = \begin{bmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

and edge points are expressed using 3×1 homogeneous vectors. The estimated homography is then used to evaluate the edge error of the miswarped pixels $\delta'_j(i)$, computed as $[\delta'_j(i), 1]^\top = [e_j(i), 1]^\top - H \cdot [\widetilde{e}_j, 1]^\top$, with $i \in M_j$; finally, the estimate errors $\delta'_j(i)$ are canceled out from the warped edge using (7).

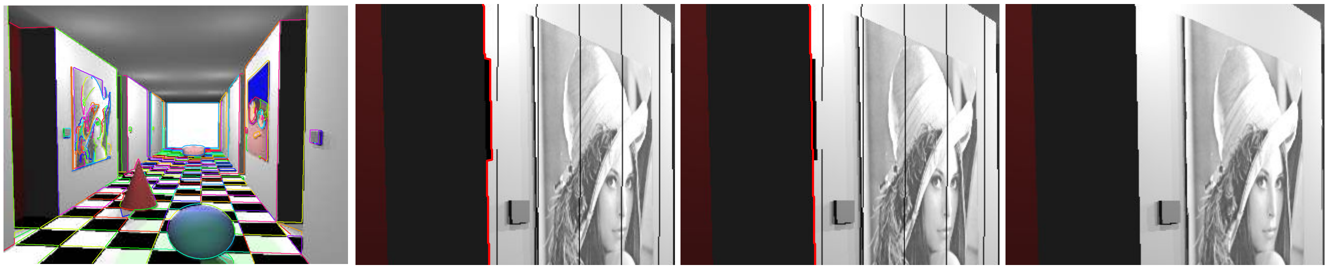
2.4. Edge enforcement in synthesized view

The final step of the algorithm is the enforcement of the corrected edge in the virtual image. This is achieved by setting the correct locations of the edge pixels according to the restored coordinates $\hat{e}_j$. Clearly, this is not enough to improve the visual quality of the virtual view; in fact, besides moving the edge pixels, proper blending of the surrounding texture must be operated to conceal the new edge position. To this end we employed the same approach proposed in [9] which is based on simple linear filtering for blending the neighboring pixels on both sides of the corrected edge.

3. EXPERIMENTS AND RESULTS

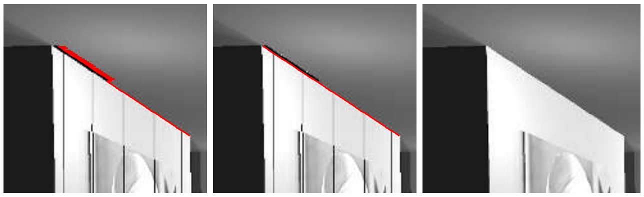
The proposed algorithm has been tested on both synthetic images with ground truth depth and using a standard 3DV dataset consisting of several multiview videos with estimated depth. The proposed algorithm can be used as a post-processing tool after any DIBR technique. In our experiments, we use the View Synthesis Reference Software (VSRS) [12] provided by MPEG to generate intermediate views for each test case. All the experiments have been worked out setting $\tau_{low} = 0.05$, $\tau_{high} = 0.15]$, $\gamma = 200$, $\lambda = 50$ and $\Delta = 3$.

In Fig. 2a the left view with edges of the synthetic image 'corridor' generated by MRTStereo [13] is shown. In this case depth errors have been introduced artificially to test the resilience of the proposed correction strategy. Fig. 2b shows a case of a miswarped edge with constant depth. Fig. 2c shows the corrected edge location whereas Fig. 2d reports the final result obtained concealing the corrected edge in the synthetized intermediate view. Fig. 3 shows an example of edge



(a) Left view with edges (b) A miswarped edge in V_l' (c) Corrected edge (d) Corrected edge enforced in V_m

Fig. 2: Correction of a constant depth edge of a synthetic scene.



(a) Miswarped edge (b) Corrected edge (c) After enforcement

Fig. 3: Correction of an edge with varying depth of a synthetic scene.

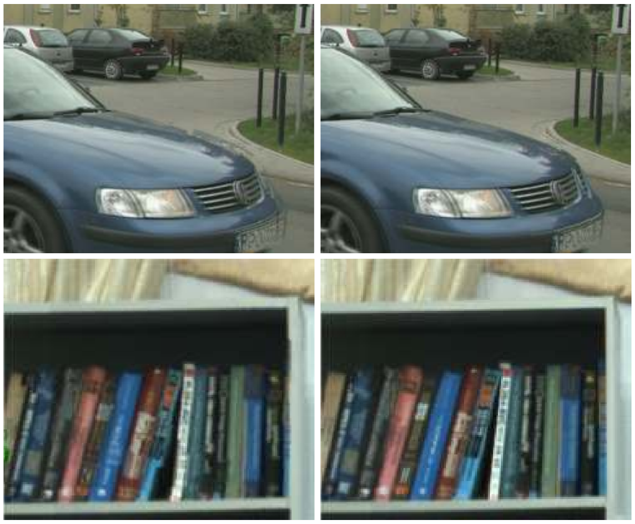
Table 1: Average structural distortion before and after correction.

Sequence	Frame #	$\overline{\mu}$	
		Before	After
Newspaper	220	1.229	0.212
Newspaper	222	1.279	0.152
Poznan_Hall2	37	1.032	0.291
Poznan_Hall2	124	0.589	0.076
Poznan_Street	248	1.175	0.212
Poznan_Street	249	1.595	0.624

error detection and recovery when the depth of the edge is monotonically increasing. The edge error (Fig. 3a) is successfully recovered (Fig. 3b) and enforced in synthesized view (Fig. 3c) also in this case applying the affine correction.

In Fig. 4 more examples of edge corrections applied to standard 3DV sequences are reported. Significant improvements can be noted on Poznan_Street sequence (see profile of the car hood) when warping views 5 and 3 to intermediate view 2 (top) and Newspaper sequence (see boundaries of books) when warping views 2 and 6 to view 4.

We can also quantify the achieved improvements using our defined structural distortion metric. To this end we compute the overall structural distortion $\overline{\mu}$ obtained averaging (4) on all the edges of a frame. The numerical results obtained on some sample frames of standard video sequences are reported in Tab. 1 and clearly show that the proposed approach is able to significantly reduce the structural distortion of the edges in the virtual image.



(a) Before (b) After correction

Fig. 4: Edge enhancement details on Poznan_Street (top) and Newspaper (bottom) sequences.

4. CONCLUSIONS

Depth image based rendering techniques generate novel views based on multi-view video plus depth (MVD) format. The edges of the objects are usually distorted due to imperfect depth maps that lower the visual quality of the synthesized view. Since, the human visual system is very sensitive to the edges; as a result, the objects associated to those distorted edges are noted immediately as erroneous. The proposed algorithm improves the visual quality of the rendered images by identifying the corrupted edges and correcting them in the final synthesized views. The experiments demonstrate the effectiveness of the proposed method.

5. REFERENCES

[1] Christoph Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3d-tv," in *Proc. SPIE*, 2004, vol. 5291, pp. 93–104.

[2] ISO/IEC JTC1/SC29/WG11 (MPEG), "Call for proposals on 3D video coding technology @ftp: //ftp.merl.com/pub/avetro/3dv-cfp/," Mar. 2011.

[3] Daniel Scharstein and Richard Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[4] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, vol. 1, pp. I–195–I–202 vol.1.

[5] Guan-Ming Su, Yu-Chi Lai, Andres Kwasinski, and Haohong Wang, "3D video communications: Challenges and opportunities," *Int. J. Commun. Syst.*, vol. 24, no. 10, pp. 1261–1281, Oct. 2011.

[6] K. Muller, P. Merkle, G. Tech, and T. Wiegand, "3D video formats and coding methods," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, 2010, pp. 2389–2392.

[7] O.P. Gangwal and R.-P. Berretty, "Depth map post-processing for 3D-TV," in *Consumer Electronics, 2009. ICCE '09. Digest of Technical Papers International Conference on*, 2009, pp. 1–2.

[8] Xuyuan Xu, Lai-Man Po, Ka-Ho Ng, Litong Feng, Kwok-Wai Cheung, Chun-Ho Cheung, and Chi-Wang Ting, "Depth map misalignment correction and dilation for DIBR view synthesis," *Signal Processing: Image Communication*, 2013.

[9] Muhammad Shahid Farid, Maurizio Lucenteforte, and Marco Grangetto, "Edges shape enforcement for visual enhancement of depth image based rendering," in *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*. IEEE, 2013, pp. 406–411.

[10] Alexandru Telea, "An image inpainting technique based on the fast marching method," *Journal of graphics tools*, vol. 9, no. 1, pp. 23–34, 2004.

[11] John Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 6, pp. 679–698, 1986.

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

[13] Dieter W Fellner and Stephan Schäfer, "MRT++ design issues and brief reference," 1998.