Depth Map Discontinuity Correction for 3D Video Systems

Camilo Dórea and Ricardo L. de Queiroz
Departamento de Ciência da Computação
Universidade de Brasília, DF, Brasil
Email: camilo@cic.unb.br, queiroz@ieee.org

***Abstract*—Inaccurate depth map discontinuities are an important source of artifact and distortion in multiview systems and applications requiring depth-based view rendering such as 3DTV or free-viewpoint video. Such errors along depth discontinuities frequently occur during depth estimation and/or depth map coding. In this paper we present a novel method for correcting inaccuracies on the discontinuities along object borders within depth maps using color features available within the associated view. Erroneous depth map border areas are split into a set of fine, color-homogeneous regions and a region merging technique is used to form new depth map borders in conformance with the available color cues. The proposed algorithm can align depth discontinuities to object borders using only information within a single view. Results demonstrate the effectiveness of our method on multiview video-plus-depth test images.**

(a) (b)

Fig. 1. (a) The *Ballet* sequence (size 1024x768), view 2, frame 0 and (b) a corresponding gray scale depth map.

I. Introduction

Depth maps are an increasingly common component within video representation formats for 3D video and multiview systems [1], [2]. The multiview video-plus-depth (MVD) format, for example, consists of multiple video sequences, captured by synchronized cameras from different viewpoints, and the corresponding per pixel depth maps. The pixel value within the depth map represents the distance of that point to a reference camera. Objects which are closer to the camera will appear brighter on the depth map while those farther from the camera are shown darker as illustrated in Fig. 1.

Recent advances in multiview video display technology have popularized new applications such as 3DTV and free-viewpoint video (FVV). In both of these applications, virtual views must be rendered in order to provide the streoscopic effect required by 3D or allow the user to select an aribitrary viewing angle in FVV. When using the MVD format, the accuracy of the depth map is fundamental to the quality of the rendered virtual view. In particular, erroneous definition of depth discontinuities along object borders can lead to very disturbing rendering artifacts such as incomplete or disconnected objects. However, it is precisely along depth discontinuities that accurate depth acquisition becomes the most challenging. For instance, depth (or disparity) estimation via block matching techniques [3] will generally fail if the block contains a depth (or motion) boundary.

Multiview systems must also cope with a large increase in the amount of raw data which in turn imposes strong requirements on coding efficiency. Depth maps are subject to compression with current video coding standards such as the MPEG-C Part 3 and H.264 Auxiliary Picture Syntax [4] as well as other dedicated coding techniques which exploit particular depth map characteristics: generally smooth areas separated by depth discontinuities. In [5], a quadtree decomposition is used to partition the depth map into blocks of variable sizes and represent them with piecewise linear functions. In spite of coding gains, the aforementioned techniques are responsible for further alterations, particulary along depth discontinuities, which present impairments to visual quality of depth-based rendered views.

In this paper we propose the use of a color-based segmentation technique to correct potentially erroneous discontinuities within depth maps. A region-based approach is selected, allowing the splitting of depth-homogeneous regions and the subsequent re-arrangement of these in accordance to the color cues provided by the objects within the associated view. A novel region growing technique, steered by a color-homogeneity criteria, is used to form new depth discontinuities respectful of object boundaries. The algorithm can be used to recover correct depth estimates from coarse or sparse estimation procedures as well as high compression ratios. The algorithm uses only information contained in a single view without resorting to the processing of neighboring views for refinement of the depth values.

The proposed algorithm is developed in Section II. In Section III we demonstrate the effectiveness of our approach by recovering depth discontinuities compatible with object boundaries in depth maps subject to severe blocking distortions. Conclusions and directions of future work are presented in Section IV.

II. Proposed Algorithm

Region-based processing techniques are used to achieve depth map discontinuity correction. Here, a partition of the color image or depth map is defined as a set of continuous and non-overlapping regions such that the union of all such regions covers the entire image support.

Initially, a partition of color-homogenous regions is formed for the color image associated to a depth map. We assume a region of uniform depth will be composed of one or more sub-regions each of which possessing uniform color. Next, areas of relevant discontinuities within the provided depth map are detected and classified as uncertainty areas. Finally, a color-based region growing technique is used to form new depth-homogeneous regions in accordance with the boundaries of uniform color regions.

A. Color-based region merging

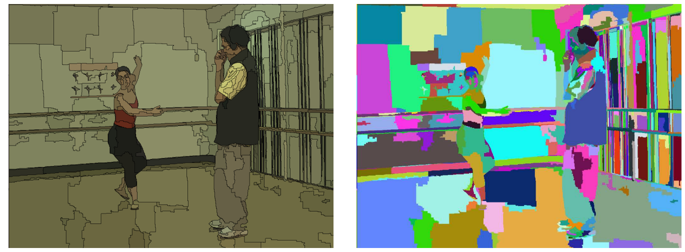
The General Merging Algorithm [6] is used to create an image partition of color-homogeneous regions. The algorithm constructs a Region Adjacency Graph (RAG) composed of a set of nodes representing the regions of an initial partition and links defining the connectivity between regions. The algorithm proceeds iteratively by removing a link from the RAG, merging the associated nodes and updating the RAG to the new neighborhood configuration. A similarity measure between regions is used to order the merging process until a termination criterion is reached.

For color-based region merging, the regions are initially assumed to be individual pixels. The color model $\mathbf{C}_R$ for a region R is a column vector containing the mean YUV color components over all pixels $p \in R$. A color-based similarity measure between regions R_i and R_j proposed in [6] is given by:

$$S(R_i, R_j) = |R_i| \left\| \mathbf{w}^{1/2} \left(\mathbf{C}_{R_i} - \mathbf{C}_{R_i \cup R_j} \right) \right\|^2 + |R_j| \left\| \mathbf{w}^{1/2} \left(\mathbf{C}_{R_j} - \mathbf{C}_{R_i \cup R_j} \right) \right\|^2 \quad (1)$$

where $\| \cdot \|$ is the Euclidean norm and the vector of weights $\mathbf{w}^{1/2} = [\sqrt{w_Y} \; \sqrt{w_U} \; \sqrt{w_V}]$ is such that $w_Y + w_U + w_V = 1$. The norms express the distance between the neighboring regions and their union. The norms are also pondered by region area $| \cdot |$ in order to account for different region sizes.

The algorithm keeps merging the regions which present the highest similarity measures, while updating the models, until a termination criterion is reached. Termination criteria are typically a total number of regions or a global color modeling error. In Fig 2 a typical color-based partition resulting from region merging is presented for the *Ballet* image of Fig. 1. A final number of regions of 500 is stipulated although, in our tests, any value as low as 50 regions suffices to guarantee that objects of interest are correctly separated from background areas. Note that alternative color segmentation techniques, such as watershed or mean shift, may also be used to create color-based partitions.



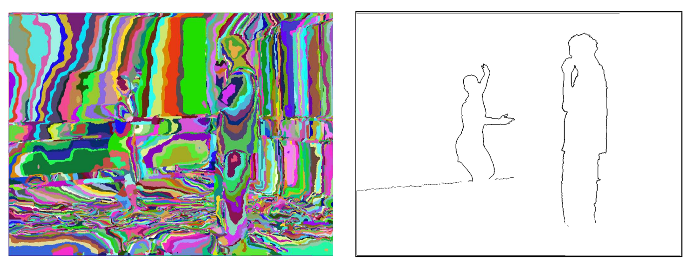
(a) (b)

Fig. 2. Partition of *Ballet* view 2 generated with color-based region merging (500 regions) (a) average YUV colors (b) pseudo color for each region.

B. Relevant depth discontinuity detection

Depth maps contain gray scale values with generally smooth areas separated by strong discontinuities as exemplified in Fig. 1(b). Partitions of such maps may be formed by labeling of flat zones, i.e., regions of constant depth value. As illustrated in Fig. 3(a), flat zone partitions of such maps will differentiate among all depth values and generally yield a large number of regions in spite of the smooth aspect of the maps.

In order to isolate the depth discontinuities of greatest relevance, which are also the ones subject to the largest inaccuracies during estimation and/or coding, a Sobel operator [7] is used to compute depth map gradients and these are thresholded against a minimum value δ. Fig. 3(b) shows a binary image resulting from a Sobel filtering and subsequent thresholding operation on the depth map of Fig. 1(a). Shown in black are the depth discontinuities deemed as most relevant which will be subject to correction using available color cues.



(a) (b)

Fig. 3. (a) Flat zones of the *Ballet* depth map in pseudo color (b) detected relevant depth discontinuities with Sobel operator ($\delta = 20$).

The thickness of the detected depth discontinuities may be controlled through morphological erosion. A square structuring element of size s eroding the binary image formed by thresholding the depth map gradient is used to further define the extent of the depth discontinuity areas.

C. Region growing

Relevant depth discontinuities are used to define an uncertainty area over which regions of homogeneous color are allowed to grow, providing new depth values and thus correcting depth discontinuities within uncertainty areas. Different stages of this procedure are illustrated in Fig. 4 and discussed next.

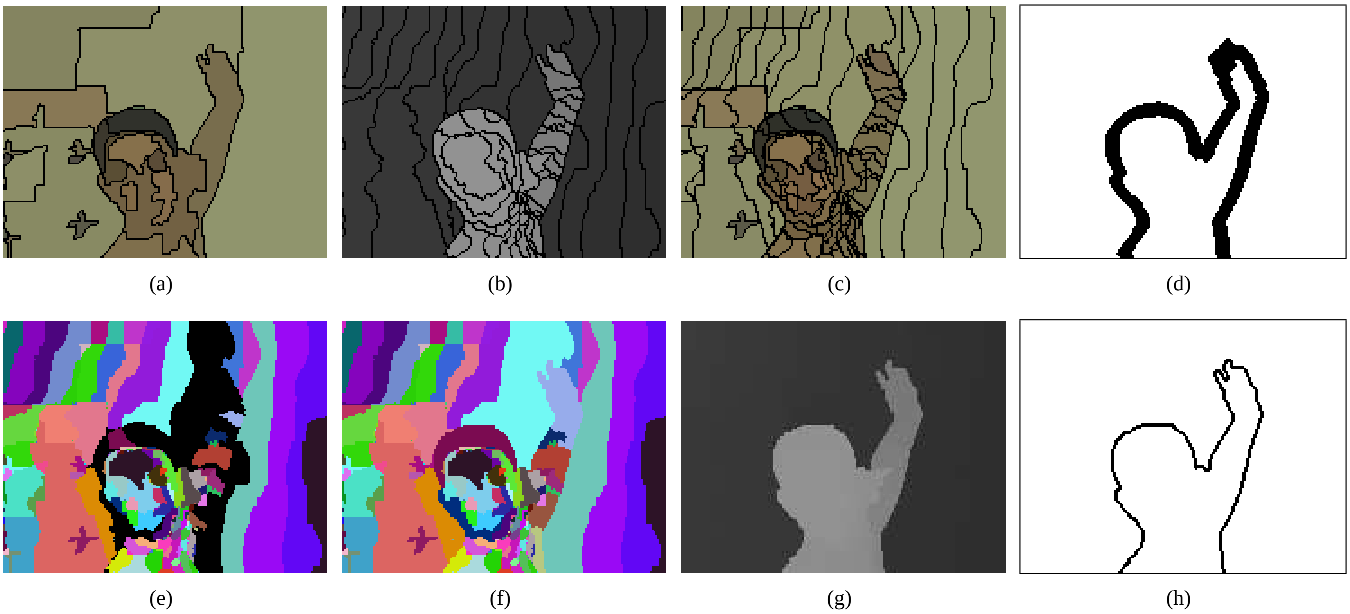


Fig. 4. Various stages within the region growing procedure. Detail crop of *Ballet* containing (a) color-homogeneous partition, (b) constant depth partition, (c) infimum partition, (d) detected relevant depth discontinuity area, (e) valid regions of the infimum partition after fitting (in pseudo color) and uncertainty area (in black), (f) partition after region growing in pseudo color, (g) corrected depth map and (h) recovered depth discontinuity.

The first step towards uncertainty area definition and region growing implementation is the creation of an infimum partition. The infimum of two partitions is the partition obtained by intersecting all the regions of both partitions. Using the previously defined color-homogeneous partition (see § II-A) and constant depth partition (see § II-B) we obtain an infimum partition consisting of color homogeneous regions with constant depth. Presented in Fig. 4(c) is an infimum partition obtained from the partitions shown in Figs. 4(a) and (b). This partition respects all of the computed color boundaries as well as all the discontinuities provided by the depth map.

Once an infimum partition has been obtained, a region fitting procedure with the relevant depth discontinuity area is used to determine an uncertainty area. All regions of the infimum partition which are partially excluded from the relenvant depth discontinuity area by at least ρ are considered to be valid. Typically $\rho = 55\%$ to guarantee a majority of the valid region is beyond the depth discontinuity. All other, non-valid regions of the infimum partition form the uncertainty area. Shown in Figs. 4(d) and (e) are a detected relevant discontinuity area and the infimum regions considered valid along with the uncertainty area. Note that each valid infimum region has an associated (constant) depth value which is recorded for future reference.

Lastly, a color-based region merging procedure is responsible for growing the valid regions over the uncertainty areas. At this stage the previously described General Merging Algorithm is employed whereby regions with valid labels are not allowed to merge amongst themselves and regions with uncertainty labels inherit the labels of the valid regions to whom they may merge. In other words, valid regions grow by incorporating regions of the uncertainty area. The similarity measure of equation (1) is used to order the merging process. Merging is iterated until no regions with uncertainty labels remain. Once region growing is terminated, depth map pixels within the uncertainty areas are attributed the depth values of their valid infimum partition regions before growth. As an example, the valid regions depicted in Fig. 4(e) are grown over the uncertainty area resulting in the partition of Fig. 4(f). Corrected depth values are presented in Fig. 4(g) and the recovered depth discontinuity of in Fig. 4(h). In this example, the growth of regions within the uncertainty area allows the new depth discontinuities to accurately match the original.

III. Experimental Results

The proposed depth discontinuity correction algorithm was tested on two publicly available data sets: *Ballet* and *Breakdancers* [8]. To simulate inaccuracies along depth map discontinuities, a blocking effect was generated on the original maps by substituting depths for their median values within NxN blocks. Similar blocking effects arise, for example, when coding depth maps at lower resolutions for compression purposes.

Color images and corrupted depth maps from view 2, frame 0 were used in all tests along with color-based partitions of 500 regions; similarity measure with weights $w_Y = w_U = w_V$; erosion structuring element of size $s = 7$ for discontinuity detection; and fitting percentage of $\rho = 55\%$ for uncertainty area definition.

A depth map corrupted with median values from 16x16 blocks along with a detail crop of the detected relevant discontinuity is presented for the Ballet image in Figs. 5(a) and

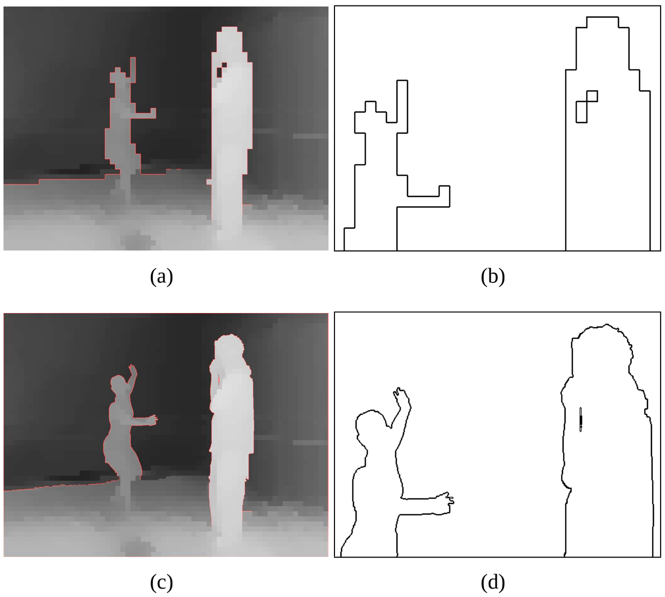


Fig. 5. (a) Corrupted depth map for *Ballet* with median value in 16x16 blocks and (b) detail crop of relevant discontinuities; (c) corrected depth map and (d) corrected detail crop . Relevant discontinuities are also highlighted in red in (a) and (c).

(b). Severe blocking effects are noticeable around the relevant discontinuities defined with gradient threshold $\delta = 20$. The results of discontinuity correction with the proposed algorithm are shown in Figs. 5(c) and (d). The corrected relevant discontinuities are greatly refined and seen to accurately conform to the contours of the objects within the scene.

The original depth map for the *Breakdancers* image and a set of relevant discontinuities defined with $\delta = 10$ are shown in Figs. 6(a) and (b). A corrupted depth map is formed with the median values of 16x16 blocks as depicted in Figs 6(c) and (d). Results from the proposed depth discontinuity correction algorithm are provided in Figs. 6(e) and (f). Note that the algorithm recovers the original discontinuities, respecting object contours, even under severe blocking conditions.

IV. Conclusion

We have presented a novel depth map discontinuity correction algorithm capable of aligning depth discontinuities with object boundaries using color cues from the associated view. The algorithm is based on region merging techniques and may be applied to recover accurate depth discontinuities from coarsely estimated and/or highly compressed depth maps.

Future work includes the application of our technique towards the achievement of coding gains for multiview video-plus-depth format sequences as well as the measurement of view rendering quality improvement in 3DTV and free-viewpoint video applications. Furthermore, we also intend to investigate the usage of low-resolution and compressed color images in correcting depth maps.

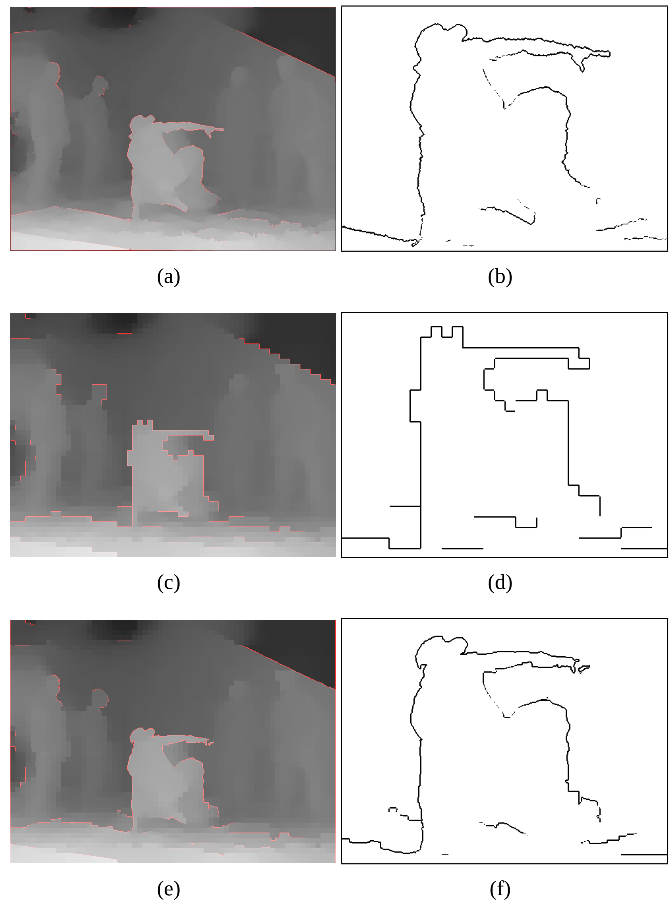


Fig. 6. (a) Original depth map for *Breakdancers* and (b) detail crop of relevant discontinuities; (c) corrupted depth map with median value in 16x16 blocks and (d) corrupted detail crop; (e) corrected depth map and (f) corrected detail crop . Relevant discontinuities are also highlighted in red in (a), (c) and (e).

References

[1] E. Martinian, A. Behrens, J. Xin, and A. Vetro, “View synthesis for multiview video compression,” in *Picture Coding Symposium (PCS)*, Beijing, China, April 2006.

[2] P. Merkle, Y. Wang, A. Smolic, and T. Wiegand, “Video plus depth compression for mobile 3d services,” in *IEEE 3DTV Conference*, Potsdam, Germany, May 2009, pp. 1–4.

[3] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International Journal of Computer Vision*, vol. 47, pp. 7–42, 2002.

[4] *ITU-T Recommendation H.264, Advnaced video coding for generic audiovisual services*, ISO/IEC, 2007.

[5] Y. Morvan, D. Farin, and P. H. N. de With, “Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images,” in *ICIP’07*, San Antonio, USA, September 2007.

[6] L. Garrido and P. Salembier, “Region based analysis of video sequences with a general merging algorithm,” in *EUSIPCO’98*, vol. 3, Rhodes, Greece, September 1998, pp. 1693–1696.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[8] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-quality video view interpolation using a layered representation,” *ACM Trans. Graph*, vol. 23, pp. 600–608, 2004.