Superpixel-based depth map inpainting for RGB-D view synthesis

Pierre Buyssens, Maxime Daisy, David Tschumperlé, Olivier Lézoray

SUPERPIXEL-BASED DEPTH MAP INPAINTING FOR RGB-D VIEW SYNTHESIS

P. Buyssens, M. Daisy, D. Tschumperlé, O. Lézoray

GREYC CNRS (UMR 6072), UNICAEN, ENSICAEN, Image Team
6, Bd. Maréchal Juin, 14000 Caen, FRANCE

ABSTRACT

In this paper we propose an approach to inpaint holes in depth maps that appear when synthesizing virtual views from a RGB-D scenes. Based on a superpixel oversegmentation of both the original and synthesized views, the proposed approach efficiently deals with many occlusion situations where most of previous approaches fail. The use of superpixels makes the algorithm more robust to inaccurate depth maps, while giving an efficient way to model the image. Extensive comparisons to relevant state-of-the-art methods show that our approach outperforms qualitatively and quantitavely these existing approaches.

***Index Terms*—** View synthesis, Depth map disocclusion, superpixels.

1. INTRODUCTION

3DTV and the more general Free-Viewpoint Rendering (FVR) have become promising technologies in 3D research. To synthesize new virtual views from known ones, Depth Image Based Rendering (DIBR) is a key technic which consists in rendering a depth map in addition to the classical intensity image. Given this latter image and its corresponding depth map, one can synthesize a new virtual view of the scene by warping these images from a new view point [1].

A critical problem then arises with the apparition of occluded areas : background (BG) areas that are hidden (and not known) by a foreground (FG) object in the original view may have to be rendered in the synthesized view (Fig. 1). Filling these holes is known as *disocclusion* and belong to the more general problem of *inpainting*. But contrary to the general problem of removing an object from the scene, an important remark can be made here : holes are almost always surrounded by both FG and BG since their apparition are due to significant depth difference between FG and BG. Moreover, an important additional resource to fill holes is the depth map that can be used to guide the inpainting process [2].

State-of-the-art overview: several works have been proposed in the literature to tackle this disocclusion problem, almost all of them are based on inpainting frameworks such as the algorithm of Criminisi *et al.* [3] widely used for its ability to efficiently reconstruct large structure portions.

This research was supported by French national grant Action 3DS.

Based on the work of [3], [2] adds depth information to both the priority term and the patch distance computations. A similar work [4] introduces 3D-tensor for the priority term computation, but assumes that the depth map is completely known (which is impossible in practice since it contains holes). [5] first inpaints the depth map with a line-based algorithm, then uses a sprite-based algorithm to fill in the intensity image. Both [4] and [5] manipulate the filling order such that it starts from the background in priority. While [6] extrapolates the information of the missing pixels from its direct surrounding BG neighborhood, [7] simulatenously fills depth map and intensity image with variants (similar to the ones of [4]) added to the Criminisi *et al.* algorithm. Finally [8] recently proposed a smart algorithm to first inpaint the depth map, and in a second time inpaints intensity image with a variant of [3] to ensure inter-view consistency.

Despite these recent works on disocclusion, very few of them deal properly with the depth map inpainting problem. This sub-problem is in fact a not-so-easy task, and it is often tackled with trivial methods that are insufficient.

Problem statement : a major problem arises when an occlusion hole that has to be filled with background is only surrounded with foreground. Fig. 1 illustrates such a case: due to the warping process, the hole pointed by the red arrow can not be properly inpainted with its surrounding neighbor. Most of methods of the literature fail on such a case since they only consider the hole neighborhood for its inpainting. To the best of our knowledge, [8] is the only work that poses this problem correctly.

The basic idea of [8] for the depth map inpainting is to use the original image (before the warping) to infer the correct missing depth values (FG or BG). Specifically, the line wise filling method of [8] analyzes the depth distributions of the local patches around the boundary of the hole in the synthesized view and the corresponding boundary in the original view. Local maxima of depth distributions for both the synthesized and original views are then used to infer the correct depth value of the line that is then drawn. Nevertheless, this algorithm suffers from several flaws:

• since the depth inpainting relies on horizontal lines (constant depth along the line), BG planes that are not parallel to the camera plane can not be properly inpainted,

• horizontal lines may be sufficient with a horizontal transla-

Fig. 1: Illustration of one of the background recovery problems : The hole pointed by the arrow is only surrounded by foreground pixels. Top: synthesized intensity image with holes, middle: synthesized depth map with holes, and bottom: our depth map inpainted result.

tion of the point of view, but becomes clearly inadequate for general warpings.

Contributions : This paper focuses on the inpainting of occlusions that appeared in depth maps after a warping process. To this end, we propose an efficient algorithm to specifically inpaint these holes. Based on the same idea as [8], it uses the original image (before warping) to infer correctly the missing depth values. The algorithm is detailed in Sec. 2, and is compared both qualitatively and quantitatively to [8] in Sec. 3.

2. PROPOSED DEPTH MAP INPAINTING METHOD

The main idea of our proposed approach is to guess and decide between which FG and BG depth values to use to fill holes. To tackle the problem, the proposed solution is to find and extend planes in the original image with the use of superpixels.

The proposed algorithm is composed of 3 steps:
1) compute superpixel oversegmentations of the original and synthesized views, and the correspondence between both their superpixels,
2) for each pixel to inpaint, find a set of candidates superpixels candidates in the original view,
3) modelize the remaining superpixels by planes, and infer the depth value of the pixel by a linear combination of these planes.

Notations : a depth source image S is warped according to an offset map to generate a new depth view D. Pixels $p \in D$ where the depth information is unknown form a set of holes $\Omega = \{\Omega_1 \dots \Omega_n\}$. Superpixels R_i^S and R_i^D denote superpixels with the same label i belonging to S (respectively D). In the following, we adopt the convention that a pixel that is far from the camera has a low depth value, and a pixel close to the camera has a high depth value.

2.1. Superpixel rendering

Our method starts by computing an oversegmentation of the source image S into superpixels. We have considered the recent superpixel algorithm *Eikonal-based Region Growing Clustering* (ERGC) [9] that formulates the pixel clustering as the solution of an Eikonal equation. Given a set of initial seed pixels $\{s_i\}$ regularly sampled on the image, ERGC associates to each pixel p the potential $P(p) = \|S(p) - R_i\|^2 + \|p - s_i\|_2^2$, where $S(p)$ is the depth of p, R_i is the mean depth of the superpixel R_i being formed, and $\|p - s_i\|_2$ is the euclidean distance (a spatial constraint) of p to the seed pixel s_i of the superpixel R_i. The resulting label map produced by ERGC (left column of Figure 2) is then warped to the new view [1], and the superpixels of D are eroded by a circular structuring element of 1 pixel radius. This step prevents the potential bad labeling of pixels belonging to object boundaries. ERGC is then applied on D with the warped eroded labels taken as seeds, without diffusing into the holes (right column of Figure 2) Figure 2 (bottom) shows some correspondences between superpixels of S and D.

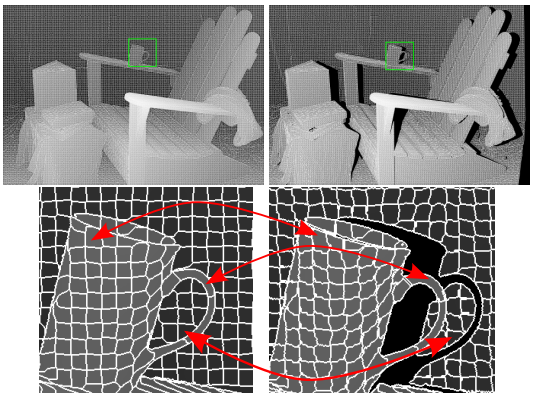


Fig. 2: Illustration of the Superpixels matching. Arrows indicate superpixels in the original image (left) and in the synthesized image (right) that share the same label.

2.2. Source planes search

Given a pixel p to inpaint, the source plane search consists in finding a subset of superpixels that will be used to infer the depth value of p. It consists in 3 steps:

1) The closest pixel $q_1 \in D \setminus \Omega$ is found with a gradient descent performed on the distance function of pixels $p_i \in \Omega$ to pixels $p_j \in D \setminus \Omega$. The adjoining superpixel is noted R_1^D. A second pixel $q_2 \in D \setminus \Omega$ is found on the opposite side of the hole by simply following the line defined by (q_1, p). The adjoining superpixel of q_2 is noted R_2^D. Pixels q_1 and q_2 are then used to define offsets $\Delta_1^p = \|p - q_1\|$ and $\Delta_2^p = \|p - q_2\|$.
2) q_1 and q_2 are then reported onto S with the inverse warping map, *with the constraint* that they belong to R_1^S and R_2^S. From these two new points $q_3 \in S$ and $q_4 \in S$, the offsets $-\Delta_1^p$ and $-\Delta_2^p$ point to two new superpixels R_3^S and R_4^S. Note that R_3^S and R_4^S may be se same superpixel.
3) From the set of superpixels $\{R_1^D, R_2^D, R_3^S, R_4^S\}$, only those with the smallest mean depth value (BG superpixels) are retained to form the set of candidate superpixels.

Fig. 3 illustrates this process: R_1^D R_2^D are depicted in blue and red respectively (second and third rows). R_3^S and R_4^S may be two different superpixels (dark blue and dark red, third row) or may be the same superpixel (green superpixel, second row).

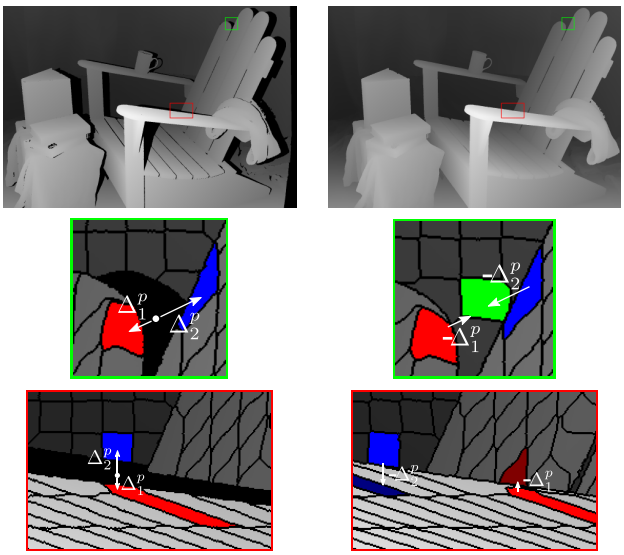


Fig. 3: Illustration of the search for the candidate superpixel. Given a pixel p to inpaint (white dot), red and blue superpixels are found (see Section 2.2) in the synthesized image (left column). The offsets Δ_1^p and Δ_2^p are then reported onto the original image (right column) from previously blue and red found superpixels. These offsets $-\Delta_1^p$ and $-\Delta_2^p$ can point to the same superpixel (green superpixel, second row), or they can point to different ones (dark blue and dark red, third row). Finally, the superpixels retained to infer the depth value of p are those with the smallest mean depth value : the green one in the first case (second row), and the blue one in the second case (third row).

2.3. Infering depth values

Each superpixel of the candidate set is represented by a unique plane via a least square regression. The depth value of p is then computed as a linear combination of these extended planes. The weights of this combination are computed according to the distance between the pixel p and the retained superpixels.

2.4. Discussion

The advantages of using superpixels in our method are twofold:
• Since the warping map may be not accurate enough, warping a single pixel may lead to errors. Using superpixels instead leads to a more robust algorithm, especially when looking for q_3 and q_4 in the source planes search step.
• ERGC superpixels adhere well to boundaries and then naturally define homogeneous parts of the depth map. The planes obtained by the least square regression for each superpixel are then sufficient to infer a good depth value of missing pixels.

Complexity: the main complexity burden of the proposed approach is the oversegmentation into superpixels of S. ERGC is based on the *Fast-Marching* algorithm that requires the sorting of pixels according to their geodesic distances to seed pixels. With an appropriate heap structure, the complexity is roughly $\mathcal{O}(n \log n)$. Despite this theoretical complexity, the proposed algorithm is very fast in practice, and is nearly linear in time. Moreover, note that $\mathcal{O}(n)$ implementation has been proposed in the literature [11].

The oversegmentation into superpixels of D is much more faster since the diffusion is processed on far fewer pixels (eroded parts of superpixels of S). The rest of the algorithm is linear in time and can easily be parallelized since each pixel is inpainted independently.

3. RESULTS

3.1. Comparisons on a synthetic image

In this section, we show inpainting results on a synthetic image composed of two objects : a disc perforated with two circular holes in front of a plane that is not parallel to the camera plane (see Fig. 4a). Changing the point of view reveals holes to inpaint in the synthesized view, and in particular one initial circular hole that is only surrounded by depth values of the foreground object (see Fig. 4b and 4c). While [5] fails at filling the circular hole with background values (Fig. 4d), [8] fails to correctly reconstruct the depth values of the background plane (Fig. 4e). Missing depth values are well recovered with our method (Fig. 4f).

3.2. Comparisons on real data

In this section we compare our method with the one of [8] on the recent Middlebury-2014 stereo dataset [12]. It is com-

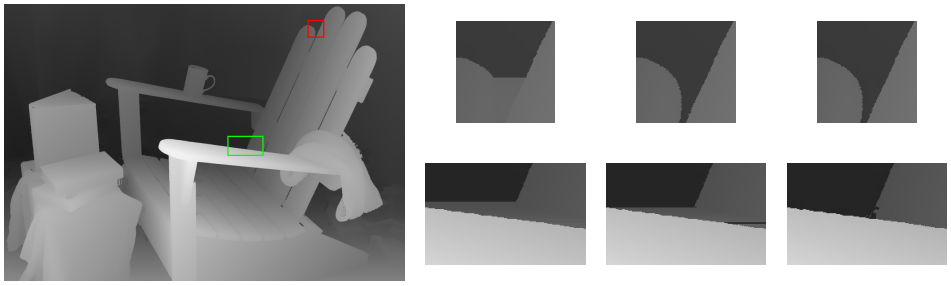
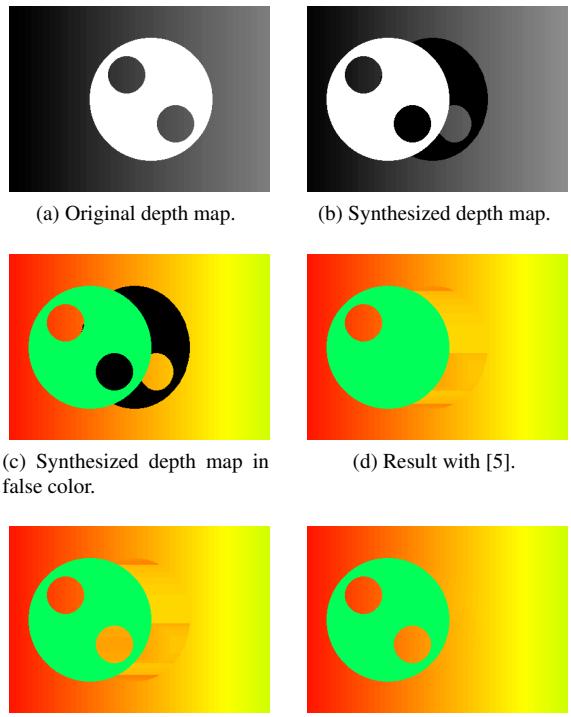


Fig. 6: First column: Inpainted depth map with our method. Second to last row: inpainting result magnifications with [5], [8], and our method respectively.



(a) Original depth map. (b) Synthesized depth map.

(c) Synthesized depth map in false color. (d) Result with [5].

(e) Result with [8]. (f) Our result.

Fig. 4: Depth map inpainting comparison on a synthetic image rendered in false color for visualization purposes.

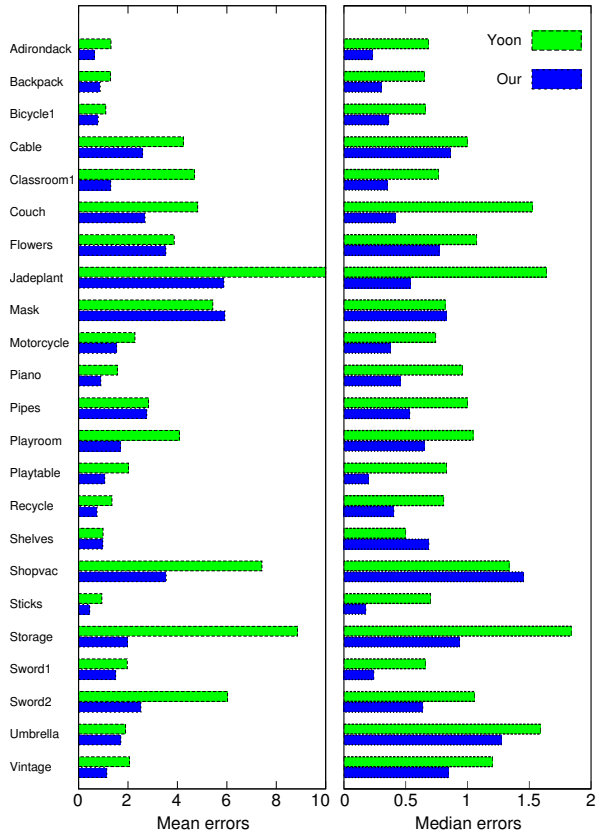


Fig. 5: Mean and median error for each image of the Middlebury-2014 dataset.

posed of 23 pair of 2880×1988 images for whose depth maps are known, and synthesis is performed from view_0 to view_1.

Figure 6 compares quality of inpainting results of [5] and [8] with our proposed method. Figure 5 plots and compares for each image the mean errors of the reconstruction of the method of [8] and ours. We also plot the median errors of each algorithm to better appreciate these results, and to better reflect the overall performances of the inpainting. These experimental results show a clear improvement of our proposed method over [8].

4. CONCLUSION

This paper proposes a novel method that specifically deals with depth map inpainting for view synthesis. Based on superpixels, the proposed approach outperforms qualitatively and quantitatively existing dedicated approaches of the state-of-the-art. Based on these robust inpainted depth map, further work consists in inpainting synthesized intensity images.

5. REFERENCES

[1] Christoph Fehn, "Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv," in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 93–104.

[2] Ismaël Daribo and Béatrice Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," in *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*. IEEE, 2010, pp. 167–170.

[3] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama, "Region filling and object removal by exemplar-based image inpainting," *Image Processing, IEEE Transactions on*, vol. 13, no. 9, pp. 1200–1212, 2004.

[4] Josselin Gautier, Olivier Le Meur, and Christine Guillemot, "Depth-based image completion for view synthesis," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*. IEEE, 2011, pp. 1–4.

[5] Patrick Ndjiki-Nya, Martin Koppel, Dimitar Doshkov, Haricharan Lakshman, Philipp Merkle, K Muller, and Thomas 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.

[6] Lai-Man Po, Shihang Zhang, Xuyuan Xu, and Yuesheng Zhu, "A new multidirectional extrapolation hole-filling method for depth-image-based rendering," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 2589–2592.

[7] Ilkoo Ahn and Changick Kim, "Depth-based disocclusion filling for virtual view synthesis," in *Multimedia and Expo (ICME), 2012 IEEE International Conference on*. IEEE, 2012, pp. 109–114.

[8] Soo Sung Yoon, Hosik Sohn, and Yong Man Ro, "Inter–view consistent hole filling in view extrapolation for multi–view image generation," in *International Conference on Image Processing*, pp. 2883–2887. 2014.

[9] Pierre Buyssens, Isabelle Gardin, Su Ruan, and Abderrahim Elmoataz, "Eikonal-Based region growing for efficient clustering," *Image and Vision Computing*, vol. 32, no. 12, pp. 1045–1054, 2014.

[10] Kwan-Jung Oh, Sehoon Yea, and Yo-Sung Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video," in *Picture Coding Symposium, 2009. PCS 2009*. IEEE, 2009, pp. 1–4.

[11] Liron Yatziv, Alberto Bartesaghi, and Guillermo Sapiro, "$\mathcal{O}(n)$ implementation of the fast marching algorithm," *Journal of computational physics*, vol. 212, no. 2, pp. 393–399, 2006.

[12] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Pattern Recognition*, pp. 31–42. Springer, 2014.