Depth-Based View Synthesis Using Pixel-level Image Inpainting

Shaoping Lu, Jan Hanca, Adrian Munteanu, Peter Schelkens

Department of Electronics and Informatics,
Vrije Universiteit Brussel
Pleinlaan 2, Brussels, Belgium

Department of Future Media and Imaging,
iMinds V. Z. W.
G. Crommenlaan 8, Ghent, Belgium

Email: {splu, jhanca, acmuntea, pschelke}@etro.vub.ac.be

***Abstract*—Depth-based view synthesis can produce novel realistic images of a scene by view warping and image inpainting. This paper presents a depth-based view synthesis approach performing pixel-level image inpainting. The proposed approach provides great flexibility in pixel manipulation and prevents random effects in texture propagation. By analyzing the process generating image holes in view warping, we firstly classify such areas into simple holes and disocclusion areas. Based on depth information constraints and different strategies for random propagation, an approximate nearest-neighbor match based pixel-level inpainting is introduced to complete holes from the two classes. Experimental results demonstrate that the proposed view synthesis method can effectively produce smooth textures and reasonable structure propagation. The proposed depth-based pixel-level inpainting is well suitable to multi-view video and other higher dimensional view synthesis settings.**

***Index Terms*—depth image based rendering; view synthesis; approximate nearest-neighbor match; pixel-level inpainting**

I. Introduction

The rapid development of 3D stereo cameras, depth cameras and various cameras arrays came as response to the enormous need of 3D-oriented video applications. In this context, massive research efforts have been recently invested in the areas of three dimensional Television (3DTV) and free-view Television (FTV). In these research fields, synthesis of arbitrary views by making use of multi-view videos and depth information is of vital importance.

Depth Image based Rendering (DIBR) has been introduced as a solution to generate high quality results in virtual view synthesis. In general, DIBR techniques consist of view reprojection, corresponding to virtual view warping from known viewpoints, and content recovery (or inpainting) starting from known views, depth maps and limited geometry information (e.g. camera parameters).

Thanks to image editing techniques in computer vision and computer graphics, image inpainting has been widely studied in the past [1], [2]. However, such classical 2D image inpainting techniques neither take depth information into account nor consider special application demands as in Multi-View Plus Depth (MVD) based view synthesis and stereoscopic scene representations. The lack of geometry consistencies and higher dimensional constraints render conventional 2D inpainting techniques [1], [2] to be far from perfect. Due to the processing precision error, there are still holes left in the warped view, as well as there are many incorrectly inpainted areas. In essence, although some simple filtering (e.g. median filtering [3], [4]) can fix some small holes, disocclusion inpainting is still far from mature, in particular in neighboring areas lying in different depth layers.

Existing methods for depth-based image inpainting can be roughly divided into two categories: exemplar-based content duplication and region interpolation (or extrapolation from the camera view). The first class is mostly motivated by Criminisi's excellent work [2] in 2D image completion. To find an optimal patch-level match, a filling priority is composed of the confidence term and data term constructed by color match between patches and gradient based filling priority respectively. Such patch-based modeling and inpainting can greatly preserve the accuracy of the propagated texture structures while avoiding some local artifacts. Criminisi's work triggered a number of algorithms in depth-based image inpainting. Daribo *et al.* [5] improved the match calculation and priority by depth information. Gautier *et al.* [6] introduced a tensor based structure propagation algorithm to promote the priority of structural textures. Surely, as discussed in [5], [6], foreground/background information can be further explored and image inpainting can be improved by some reasonable constraints in depth-based view synthesis although it is still difficult to accurately classify and segment the foreground and background regions.

For the second class of depth-based inpainting, direct interpolations with filters (e.g. average or Gaussian filter) are simple and fast, but they do yield significant geometric distortions and artifacts in disocclusion areas. In 2D image inpainting, harmonic interpolation methods have been constructed to obtain smooth results (e.g. by solving a discrete Poisson equation in [1]). Ndjiki-Nya *et al.* [7] introduced a similar approach to fill small holes and keep smooth transitions between patches. Recently, Barnes *et al.* [8] introduced an approximate nearest-neighbor match based method called PatchMatch. Apart from 2D image inpainting, PatchMatch has been applied in stereo matching [9] and semantic segmentation [10].

In this paper, we introduce a depth-based view synthesis approach that performs pixel-level image inpainting. Our

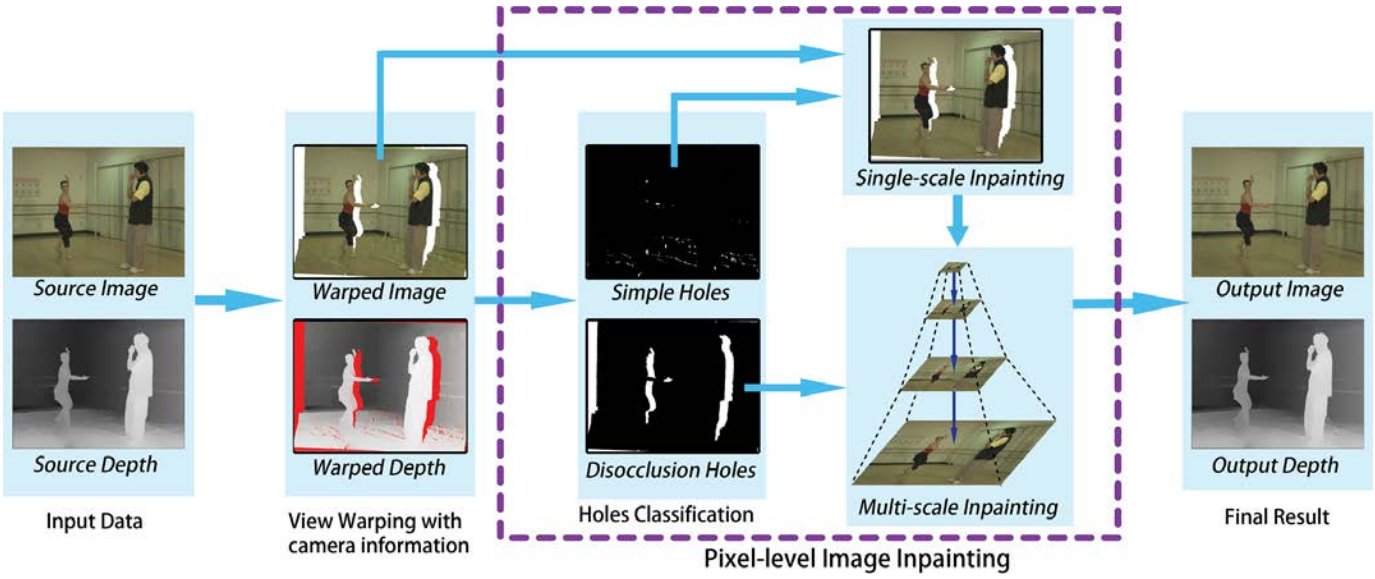


Fig. 1. The proposed pixel-level view synthesis framework.

approach falls in the category of exemplar based inpainting techniques, which means that all filled textures are plausibly extended with known image information. Our pixel-level propagation makes use of depth information and other 3D information constraints to perform inpainting. Furthermore, in contrast to existing depth-based inpainting techniques in the literature operating at single scales [5], [6], our approach performs multi-scale inpainting. By taking advantage of the randomized algorithm from [8], depth-based inpainting can efficiently implement approximate nearest-neighbor matches and propagation at pixel-level. To benefit more from such match at different scales, the proposed approach automatically classifies image holes as small simple holes or as disocclusion regions. We perform our pixel-level inpainting directly completing simple holes at a single scale followed by a depth-based multi-scale disocclusion inpainting. The proposed approach combines some special depth-based constraints, such as coarse foreground/background calculation for selecting candidates in multi-scale inpainting. In contrast to other algorithms, our approach does not need to constraint special filling orders (e.g. horizontal direction [6], [7]) and can adaptively complete holes in each direction. The inpainting results demonstrate that the proposed approach can avoid local visual artifacts by escaping from suboptimal local minimal in the match, and produces reasonable structure reconstructions by taking benefits from our depth-based multi-scale propagation. Furthermore, the flexibility of pixel-level manipulation built in our approach provides great potential in multi-view video settings [11], [7], [12] and some other higher dimensional spatial-temporal inpainting.

The remainder of the paper is organized as follows. Section II describes the proposed inpainting algorithm. In Section III, experimental results and discussions are presented. Finally, concluding remarks are drawn in Section IV.

II. Proposed Inpainting Algorithm

A. Problem formulation and system architecture

According to known image color and depth information from a viewpoint, a new viewpoint scene can be regenerated by depth-based view synthesis. Usually, such viewpoint changing corresponds to a camera translation with different angles in horizontal direction. Suppose the known image color and depth maps are (I_0, D_0) in viewpoint V_0, from which we have to compute the new image and depth values (I_1, D_1) in the new viewpoint V_1; there exists a mapping relation between V_0 and V_1, given by:

$$F : (V_0, I_0, D_0) \mapsto (V_1, I_1, D_1) \tag{1}$$

where F denotes the mapping under view transformation in the 3D scene. Given the camera parameters, we can obtain the precise 3D warping expressed by F [13]. Such a mapping is a one-to-one function, but if there is some loss of information in the original scene or due to inaccurate processing, the mapping operation is affected. Unfortunately, even if employing highly accurate pixel-level texture and depth information (I_0, D_0), we cannot still compute a perfectly accurate view (I_1, D_1). The reason is that the scene projection in V_0 does not contain all 3D information in the real world. In other words, the information in (I_0, D_0) is necessary for the computation of (I_1, D_1) but not sufficient. A second observation one can make is that, the content in the planar scene is discrete, whereas the actual projection from objects in the real world to such a planar scene is continuous and may often fall at sub-pixel locations. The inherent information loss produced by such precision error will affect the mapping operation, leading to small holes in the computed view (I_1, D_1). Secondly, trivial depth gaps, occurring between objects from similar depth planes or mistakes caused be erroneous depth map computation will again affect such a mapping operation. Hence, when directly

processing the mapping for each pixel, there will be some *missing* pixels in both I_1 and D_1. Third, there are also large *disocclusion* areas produced by objects from different depth planes that are overlapping in V_0 and which are visible in V_1. Following this analysis, in the proposed framework we classify the holes in the warped views as *simple holes* caused by precision errors and *disocclusion holes* (see Fig. 1). Due to local spatial continuity of the image and temporal consistency of motion objects or camera, the pixels falling in simple hole category are usually a few. Conversely, such continuity makes disocclusion areas larger in size.

In contrast to patch-level duplication in some exemplar or diffusion based completion [1], the proposed solution introduces a pixel oriented filling for image holes which does not need to consider the distortions incurred by overlapping patches. Similar to [5], [2], [6], pixel-level inpainting for depth-based synthesis needs also to take the candidate match and texture propagation into account.

With our pixel-level inpainting algorithm, we firstly complete the simple holes directly, then the resulting filled image and its corresponding depth map are taken as reference for disocclusion inpainting. In disocclusion processing, our pixel-level inpainting is performed in a multi-scale pyramid with which the strong textures can be effectively propagated. The detailed description of the proposed multi-scale inpainting as well as some special constraints taken in our solution are presented next.

B. Pixel-level inpainting algorithm

To complete the image holes, the inpainting processing needs to solve the problem of texture similarity evaluation and structure propagation. Here we note that our method borrows concepts from similarity synthesis [14] and random correspondence based PatchMatch editing [8]. Within image I, the missing hole and its boundary are denoted as Ω and $\partial\Omega$ respectively. For exemplar-based completion, the missing holes should be filled according to the known region $\Phi = I - \Omega$. Similar to some other inpainting methods [5], [6], [2], the candidate selection is still based on patch match. For each pixel $m \in \partial\Omega$, for a patch Ψ_m centered at m, the optimal candidate is:

$$\Psi_{n'} = \arg\min_{n\in\Phi} S(\Psi_m, \Psi_n), \tag{2}$$

the similarity metric being the Sum of Squared Differences (SSD) in $CIE * Lab$ color space:

$$S(\Psi_m, \Psi_n) = \frac{1}{N_m}||\Psi_m - \Psi_n||^2, \tag{3}$$

where N_m is the number of known pixels inside the patch Ψ_m. While different from using confidence priority [2], we employ an approximate nearest-neighbor match [8] to propagate the texture. As an initialization step, each pixel in Ω is firstly filled by a randomly chosen candidate from Φ. To preserve local consistency of the texture in spatial domain, updating is processed by candidate propagation as well as a random search. For the propagation, if the spatial coordinate of m in Ω is (x, y), and its currently optimal candidate is (u, v) in Φ, potential optimal candidates of m's adjacent pixels $(x+1, y)$ and $(x, y+1)$, still in Ω, should be $(u+1, v)$ and $(u, v+1)$ respectively. After that, a random search is used to escape from the suboptimal local candidate selection in such propagation. Thus, extra neighboring pixels of (u, v) will be considered as candidates for m and they are defined as

$$T_m = \bigcup_{((i,j),(i',j'))} (u + \omega_i R_j, v + \omega_{i'} R_{j'}) \tag{4}$$

where $R = [-1, 0, 1]$ denotes the search direction and $\omega = [64, 32, 16, 8, 4, 2, 1, 0]$ represents the search radius which is set to be exponentially decreasing. Note that only valid candidates T_m belonging to Φ are considered, which means that all the propagated pixels are sampled only from the known image areas.

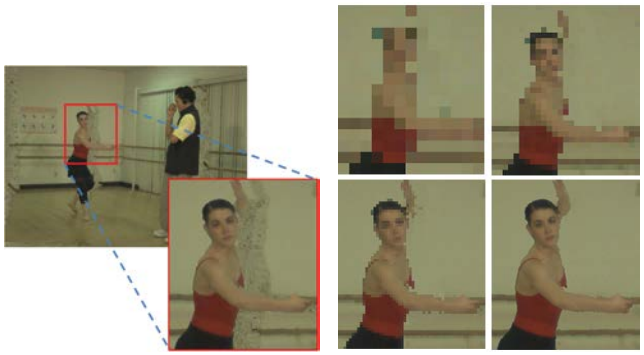
When inpainting an image hole, the search and updating are firstly finished for all the pixels of boundary $\partial\Omega$. After finding a best candidate pixel, the pixel and depth information of m are replaced by the best candidate, then its adjacent pixels belonging to the updated $\partial\Omega$ will be set into the filling order list.

For depth-based image inpainting, there are more constraints that can be further explored to generate better results than traditional 2D image completion. With this respect, our pixel-level inpainting performs a classification of holes and processes each class differently, according to depth constraints and different procedures, as detailed next.

1) Simple holes inpainting: As aforementioned, simple holes can be seen as the result of processing precision error or trivial depth gaps. To process them, we first remove small artifacts, produced by imperfect warping of the image, with a median filter. This step can also be regarded as a preprocessing step [3] in depth-based image inpainting. Still, simple holes cannot be completely filled. To separate them from disocclusions, we employ a mathematical morphology operation performing combined erosion and dilation. After that, we use the pixel-level image inpainting algorithm described in section II-B to directly fill them. The inpainting result of simple holes can be seen in Fig. 2. Actually, since such holes are much smaller than disocclusion regions, diffusion based methods (e.g. solving Laplacian equations [1]) can also fix them by performing optimal smooth interpolation requiring a solution of sparse linear equations.

(a) simple holes (green regions). (b) Simple holes filled effect.

Fig. 2. Single-scale inpainting result for simple holes.



(a) Single-scale disocclusion. (b) Multi-scale disocclusion.

Fig. 3. Comparison between single-scale and multi-scale inpainting for diocclusion areas.

2) Multi-scale disocclusion inpainting: Once simple holes are completed, the generated image and depth map are used as references for multi-scale disocclusion inpainting. If the completed simple holes are Ω_s, the remaining disocclusion regions are given as $\Omega_d = \Omega - \Omega_s$. Then a pyramid image set is constructed:

$$P = \bigcup_k (I - \Omega_d) \downarrow^k \tag{5}$$

where $k = [0, 1, ..., S-1]$ and $\downarrow$ means downsampling. The pyramid is composed of S images obtained by successively downsampling $I - \Omega_d$. Note that in these images simple holes have been completed. Firstly we complete the top image P_{S-1} in the pyramid making use of the proposed pixel-level inpainting algorithm. The completed image in the coarsest scale is denoted as $P^{'}_{S-1}$. Then we get the upsampled image $P_{S-2} = (P^{'}_{S-1}) \uparrow$ and obtain inpainted $P^{'}_{S-2}$ given P_{S-2}. The process continues recursively until reaching the finest scale, the completion image $P^{'}_0$ being the output result of the proposed system.

Due to the overlapping effect between foreground and background in disocclusion regions, pixels in such regions should share the same (or similar) depth information as the background layer. In another words, disocclusion pixels should have a higher probability to match the candidates with larger depth values. To accommodate this reasoning, we refine the search space as:

$$\Omega^{'}_k = \bigcup_{m \in P_k} (D_k(m) < \xi) \tag{6}$$

where $D_k = D \downarrow^k$. The parameter ξ is a threshold to discriminate between the foreground and background regions and we choose it according to the histogram of the depth map D_0. In our system, we find that the background generally holding between 0.65 and 0.75, which is the range where the ξ parameter is set. Note that the depth is normalized between 0 and 255, and a higher value in D means a closer pixel to camera.

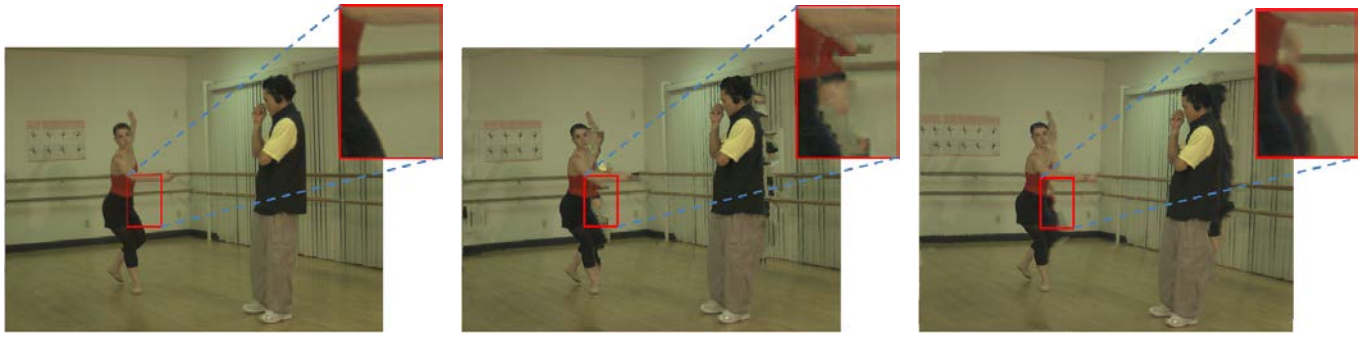
In Fig. 3, we can see the comparison between single-scale and multi-scale disocclusion inpainting. Due to the pixel-level random search and evaluation metric, using only a single scale produces random artifacts in which suboptimal candidates are selected. The reason to introduce such multi-scale inpainting is thus to preserve the correspondence of texture at different processing scales. When using a fixed patch size in a single level, the best matching gets often trapped into local suboptimal positions yielding visual artifacts even if local textures look real (see an example in Fig. 3(a)). In contrast, as one notices from Fig. 3(b), multi-scale disocclusion filling can seamlessly extend textures around $\partial\Omega$ in different spatial distances. It is also obvious that with multi-scale disocclusion recovery, structured textures can be reasonably propagated.

III. Experiments and Discussions

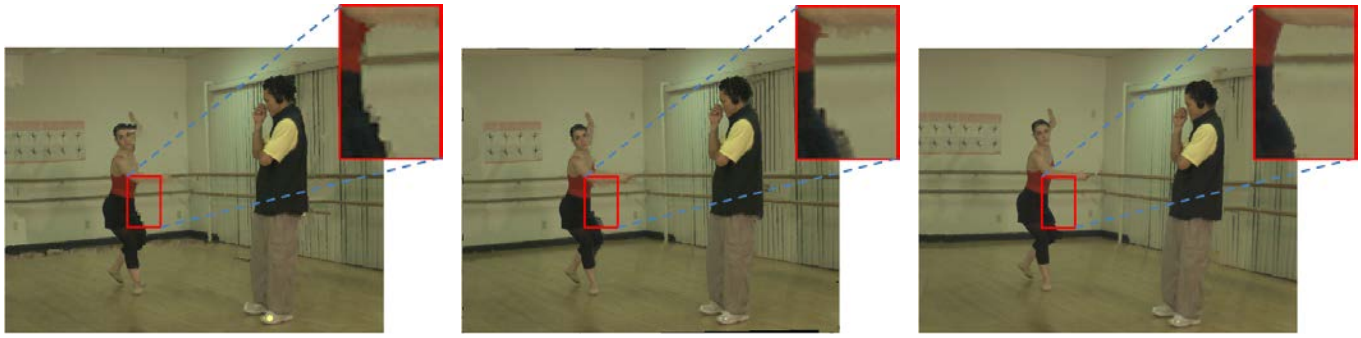
In order to evaluate the effectiveness of the proposed approach, we carried out experiments on multi-view Video-plus-Depth sequences "Ballet" and "Breakdancers" with camera parameters and estimated depth as given by Zitnick *et al.* [15]. Experiments are performed to analyze both objects disocclusion inpainting on same/different depth layers as well as synthesis of different viewpoints (view transformation from camera 5 to both cameras 4 and 2).

Besides of some aforementioned intermediate results, the final output synthesized by the proposed approach and some other view synthesis methods can be seen in Fig. 4 and Fig. 5. Each first sub-figure in these figures is the ground truth from the real camera while the others are generated by several synthesis methods. It should be noted that in the first two methods [2], [8], only 2D color information is used while the other techniques take additional depth information into account. In both test sequences we observe that although the two former methods [2], [8] can effectively complete simple holes, which also demonstrates their effectiveness in 2D image inpainting, recovered textures in disocclusion regions are unacceptably poor as they do not take depth information into account. As shown in Fig. 4, thanks to approximate nearest-neighbor matching at multiple-scales performed by our method, the bar and some strong textures are effectively propagated to disoccluded regions. Furthermore, in contrast to Fig. 4(d) and Fig. 4(e), our inpainting method can preserve the clear edge and avoid artifacts around dancing lady's hand. A similar effect can also be seen around the skirt. These results illustrate the flexibility of our inpainting technique at pixel-level which does prevent the overlapping effect generated by region-level duplication.

We also observe that the proposed algorithm can adaptively extend textures even if we did not take any gradient or directional information into account. For long distant view transformation and synthesis in Fig. 5, the proposed method can reasonably match the plausible background pixels and complete disoccluded areas even if such regions are interspersed by some foreground objects. There are also several negative effects in our result. For example, artifacts composed of discontinuous textures are still generated due to several local suboptimal match in our pixel-level inpainting approach. Also, some textures of curtains and bars behind the dancer cannot be



(a) Reference image (camera 4). (b) Criminisi's [2] result from camera 5. (c) Barnes's [8] result from camera 5.



(d) Daribo's [5] result from camera 5. (e) Gautier's [6] result from camera 5. (f) Our result from camera 5.

Fig. 4. Comparison of different methods for view synthesis from camera 5 to camera 4 on the Ballet sequence.

(a) Reference image (camera 2). (b) Criminisi's [2] result from camera 5. (c) Barnes's [8] result from camera 5.

(d) Daribo's [5] result from camera 5. (e) Gautier's [6] result from camera 5. (f) Our result from camera 5.

Fig. 5. Comparison of various methods for view synthesis from camera 5 to camera 2 on the Breakdancers sequence.

perfectly propagated because of the lack of structure priority in structural texture reconstruction.

The total computation time of the proposed depth-based view synthesis depends on the image resolution and the size of disoccluded regions, which is normally positively correlated with the transformed camera angle/distance. In general, the

computation times for our current implementation are between 5 and 20 seconds for a $1024 * 768$ resolution image for a high-end portable computer with 2.3GHz Quad-Core Intel-i7 CPU and 8GB memory. Although performing random propagation processing in approximate nearest-neighbor field, the candidate matching and inpainting at pixel-level still bear heavy computational costs.

A. Discussions

Benefiting from the approximate nearest-neighbor match algorithm and multi-scale propagation under depth constraints, the proposed texture inpainting does generate visually pleasant results, outperforming competing algorithms from the literature. In addition of local smooth texture generation, multi-scale inpainting enables structure propagation in disocclusion processing. One of the special advantages of our approach is the manipulation flexibility at pixel-level instead of patch-level. With patch-level duplication or manipulation it is difficult to keep the balance between perfect content propagation and side-effects resulting from overlapping between edited patches. By reasonable extensions, this flexibility will enable our approach to further improve the depth-based video inpainting results by further refining the employed models and constraints in the spatial-temporal dimensions.

It should be pointed out that our inpainting technique did not consider any gradient information to steer the inpainting process. This results in some cases in unpleasant effects when propagating strong structures. To alleviate these problems, a potential improvement of our technique is to introduce structure priority [6]. Also, more robust parameters selection in the proposed approach are another potential improvements. For example, to obtain the reference background pixels as reference regions for disocclusion, currently we empirically choose a threshold based on the depth histogram of the global image after completing simple holes. Precise analysis of global foreground/background distribution and study of the local topology relation between occlusion holes [16] and their neighboring known regions would be a promising direction to be explored. Finally, the computational performance of the proposed system can also be improved by hardware-level speedups, e.g. by following a GPU implementation, as the proposed pixel-level inpainting technique is suitable for deployment on parallel architectures.

IV. Conclusion

In this paper, we have proposed a new pixel-level image inpainting approach for depth-based view synthesis. Our method first classifies image holes as small simple holes and disocclusion areas. We then introduce a depth-based pixel-level image inpainting algorithm based on approximate nearest-neighbor match and complete such holes using two different strategies. The completed image after simple holes completion is used in disocclusion inpainting as reference. Disocclusion inpainting is achieved under depth map constraints and multi-scale random propagation. Experimental results demonstrate that the proposed view synthesis method can effectively preserve consistent textures and reasonably perform structure propagation.

Because of the great flexibility offered by pixel-level modeling and processing, the proposed algorithm can be effectively applied in other higher dimensional view synthesis applications. We leave these promising, yet challenging extensions as part of future research.

Acknowledgments

The authors would like to thank J. Gautier and I. Daribo for their results used in our experimental comparisons. This work was supported by IMinds vzw and IWT in the context of the ASPRO+ project.

References

[1] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 313–318, Jul. 2003.

[2] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions Image Processing*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[3] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video," in *Picture Coding Symposium*, may 2009, pp. 1 –4.

[4] V. Jantet, C. Guillemot, and L. Morin, "Joint projection filling method for occlusion handling in depth-image-based rendering," *3D Research*, vol. 2, no. 4, pp. 1–13, 2011.

[5] I. Daribo and B. Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," in *IEEE International Workshop on Multimedia Signal Processing*, Oct. 2010, pp. 167 –170.

[6] J. Gautier, O. L. Meur, and C. Guillemot, "Depth based image completion for view synthesis," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, May 2011, pp. 1 –4.

[7] 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," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 453–465, 2011.

[8] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 1–10, Aug. 2009.

[9] C. R. Michael Bleyer and C. Rother, "Patchmatch stereo - stereo matching with slanted support windows," in *Proceedings of 22nd British Machine Vision Conference*, 2011, pp. 14.1–14.11.

[10] S. Gould and Y. Zhang, "Patchmatchgraph: Building a graph of dense patch correspondences for label transfer," in *Proceedings of European Conference on Computer Vision*, vol. 7576, 2012, pp. 439–452.

[11] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[12] M. Tanimoto, T. Fujii, and K. Suzuki, *View synthesis algorithm in view synthesis reference software 2.0*. Lausanne, Switcherland: ISO/IEC JTC1/SC29/WG11 M16090, Feb. 2008.

[13] Z. Tauber, Z. Li, and M. Drew, "Review and preview: Disocclusion by inpainting for image-based rendering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 4, pp. 527–540, 2007.

[14] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[15] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 600–608, 2004.

[16] C.-H. Ling, C.-W. Lin, C.-W. Su, Y.-S. Chen, and H.-Y. Liao, "Virtual contour guided video object inpainting using posture mapping and retrieval," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 292 –302, april 2011.