Joint Projection Filling method for occlusion handling in Depth-Image-Based Rendering

Vincent Jantet, Christine Guillemot, Luce Morin

Joint Projection Filling method for occlusion handling in Depth-Image-Based Rendering

Vincent Jantet

ENS Cachan, Antenne de Bretagne

Campus de Ker Lann - 35170 Bruz, France

Christine Guillemot

INRIA Rennes, Bretagne Atlantique

Campus de Beaulieu - 35042 Rennes, France

Luce Morin

IETR - INSA Rennes

20 av. Buttes de Coësmes - 35043 Rennes, France

September 5, 2011

Abstract

This paper addresses the disocclusion problem which may occur when using Depth-Image-Based Rendering (DIBR) techniques in 3DTV and Free-Viewpoint TV applications. A new DIBR technique is proposed, which combines three methods: a Joint Projection Filling (JPF) method to handle disocclusions in synthesized depth maps; a backward projection to synthesize virtual views; and a full-Z depth-aided inpainting to fill in disoccluded areas in textures. The JPF method performs the pixels warping for virtual depth map synthesis while making use of an occlusion-compatible pixel ordering strategy, to detect cracks and disocclusions, and to select the pixels to be propagated in the occlusion areas filling process. The full-Z depth-aided inpainting method fills in disocclusions with textures at the correct depth, preserving the boundaries of the objects. Ghosting artifacts, which might otherwise result from pixel projections, are here avoided by introducing a confidence measure on background pixels to be used in the JPF process.

1 Introduction

One classical problem in computer vision applications is the synthesis of virtual views from a single video sequence, accompanied by the corresponding depth map. This problem is encountered in applications such as robot navigation, object recognition, intermediate view rendering in free-viewpoint navigation, or scene visualization with stereoscopic or auto-stereoscopic displays for 3DTV. Many rendering algorithms have been developed

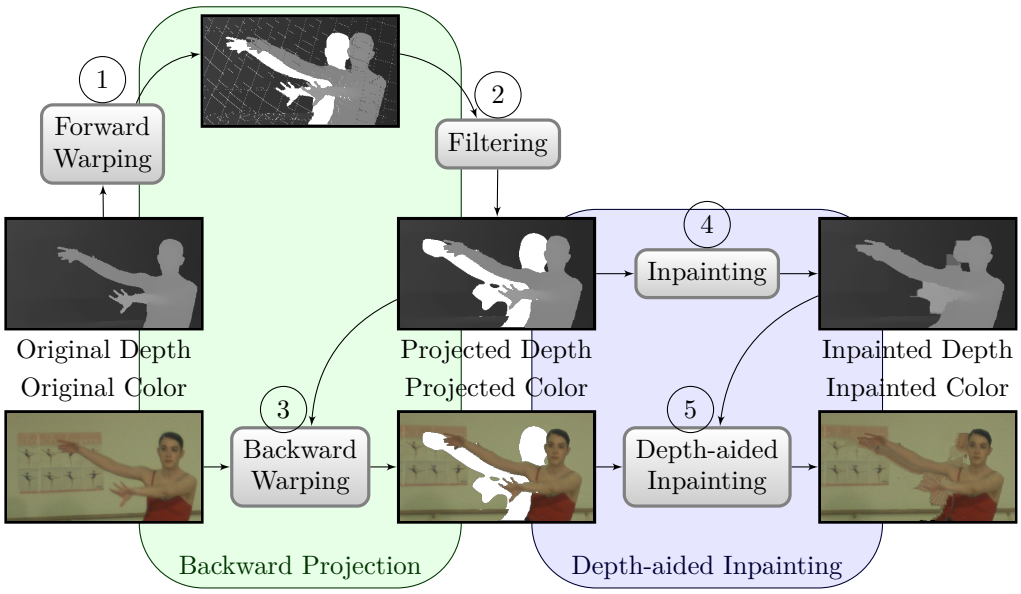


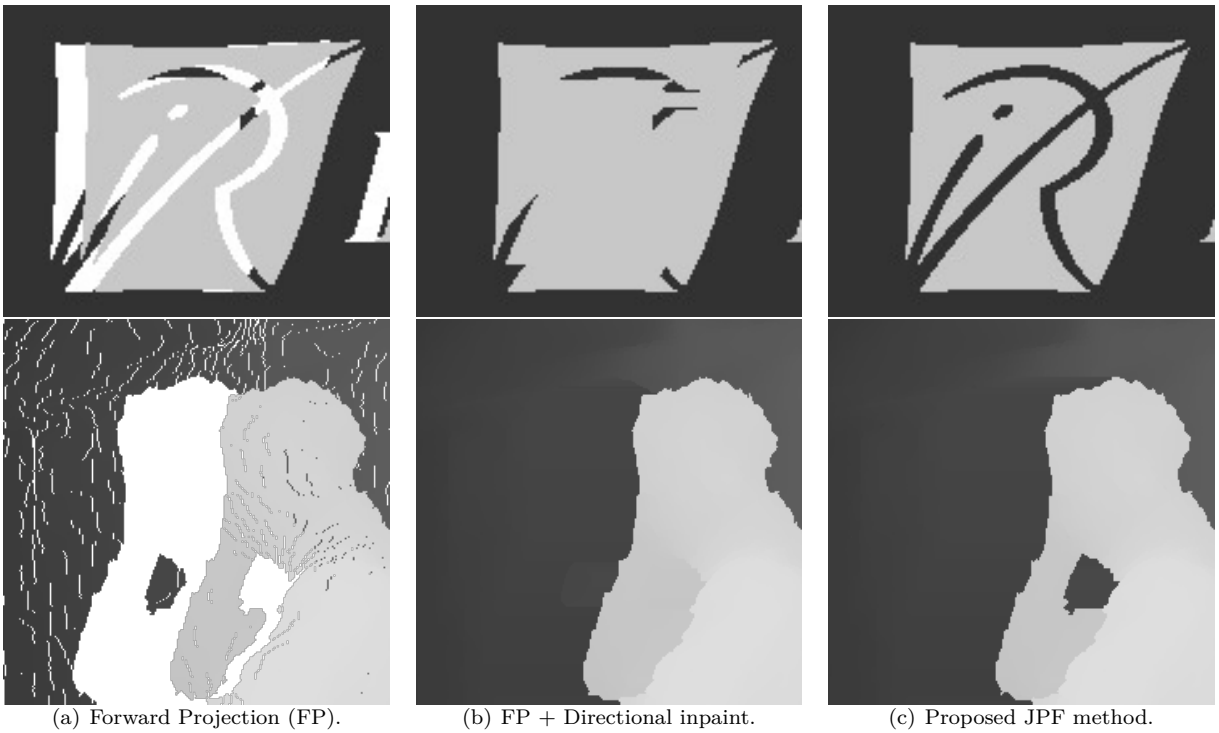
Figure 1: Classical scheme for virtual view extrapolation from a single input view plus depth video sequence. First, the input depth map is projected onto the virtual viewpoint. Second, the resulting depth map is filtered to avoid cracks and ghosting artifacts. Third, the filtered depth map is projected back onto the reference viewpoint to find the color of each pixel. Fourth, the depth map is inpainted to fill in disocclusions. Finally, the inpainted depth map is used to conduct disocclusions filling of the color map (synthesized view).

and are classified rather as Image-Based Rendering (IBR) techniques or Geometry-Based Rendering (GBR) techniques, according to the amount of 3D information they use. IBR techniques use multi-view video sequences and some limited geometric information to synthesize intermediate views. These methods allow the generation of photo-realistic virtual views at the expense of virtual camera freedom [Chan et al(2007)Chan, Shum, and Ng]. GBR techniques require detailed 3D models of the scene to synthesize arbitrary viewpoints (points of view). GBR techniques are sensitive to the accuracy of the 3D model, which is difficult to estimate from real multi-view videos. GBR techniques are thus more suitable for rendering synthetic data.

Depth-Image-Based Rendering (DIBR) techniques [Shum and Kang(2000), Zhang and Chen(2004)] include hybrid rendering methods between IBR and GBR techniques. DIBR methods are based on warping equations, which project a reference view onto a virtual viewpoint. Each input view is defined by a "color" (or "texture") map and a "depth" map, which associate a depth value to each image pixel. These depth maps are assumed to be known, or can be estimated from multi-video sequences by using a disparity estimation algorithm [Hartley and Zisserman(2004), Sourimant(2010)].

This paper describes a novel DIBR technique, which is designed to handle disocclusions in virtual view synthesis, from one or many inputs view plus depth video sequences.

The classical DIBR scheme for virtual view extrapolation from single input view plus depth video sequences is shown in figure 1. The process is divided in several distinct steps, each one designed to solve a specific problem. First, the input depth map is warped onto the virtual viewpoint. The obtained warped depth map contains disocclusions, cracks and ghosting artifacts (these artifacts are detailed in section 2). Second, this virtual depth map is filtered a first time with a median filter, in order to remove the cracks, then a second time to dilate disocclusion areas on the background side, in order to avoid ghosting artifacts during view synthesis. Third, the



(a) Forward Projection (FP). (b) FP + Directional inpaint. (c) Proposed JPF method.

Figure 2: Virtual depth map synthesized by three forward projection methods. The point-based projection method generates cracks and disocclusions 2(a). Median filtering and directional inpainting [Nguyen et al(2009)Nguyen, Do, and Patel] fills some holes with foreground depth 2(b). The proposed JPF method fills cracks and disocclusions with realistic background 2(c).

filtered depth map is involved in a backward warping to compute the color of each pixel of the virtual view. Fourth, this resulting depth map is inpainted, to fill in disocclusion areas. Finally, this complete depth map is used by a depth-aided inpainting algorithm to fill in disocclusions in the color map.

All these steps are inter-dependent, and errors introduced by each one are amplified by the following one. Connectivity information is lost during the first projection step, as shown in figure 2. Without this connectivity information, every inpainting method fails to fill in background disocclusions if the disoccluded area is surrounded by foreground objects. This case may happen each time a foreground object is not convex, and contains holes, as shown in figure 2(a). As a result, depth-aided inpainting uses wrong foreground patches to fill in background disocclusions, producing annoying artifacts, as shown in figure 2(b).

This paper describes two DIBR techniques, both based on a novel forward projection technique, called the Joint Projection Filling (JPF) method. The JPF method performs forward projection, using connectivity information to fill in disocclusions in a single step, as shown in figure 2(c).

The first proposed DIBR method, shown in figure 3, is designed to extrapolate virtual views from a single input view plus depth video sequence. The method differs from the classical scheme, presented in figure 1, by two points: the virtual depth map is synthesized by the JPF method, avoiding the use of dedicated filtering and inpainting processes; the depth-aided inpainting method is revised to take into account the high quality of the synthesized depth map.

The second proposed DIBR method is designed to interpolate intermediate views from multiple input view plus depth sequences. The method uses the Floating Texture approach to register multiple inputs view plus

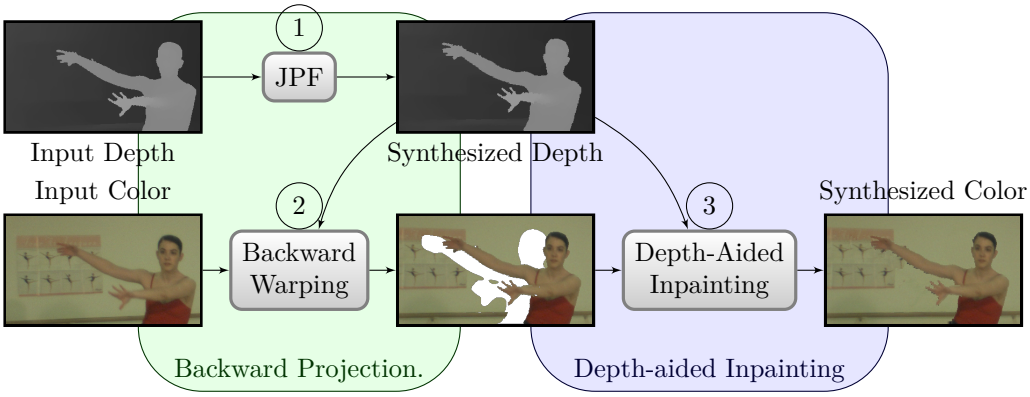


Figure 3: Proposed scheme for virtual view extrapolation from single input view plus depth sequence. First, the Joint Projection Filling (JPF) method handles cracks and disocclusions during the depth map warping. Then, the backward projection method synthesizes the virtual view. Finally, the depth-aided inpainting takes into account the high quality of the computed depth map to fill disoccluded areas in texture.

depth sequences before blending.

The JPF method fills in disocclusion areas during the projection, to ensure that geometrical structures are well preserved. The method uses the occlusion-compatible ordering presented by McMillan in [McMillan(1995)], which uses epipolar geometry to select a pixel scanning order. The algorithm was initially introduced to perform the painter's algorithm during the projection without the need of a Z-buffer. Here, not using a Z-buffer is not our purpose (by the way, the constructed depth map is a Z-buffer). The occlusion-compatible ordering is instead used to handle disocclusions gracefully. Cracks are filled in by interpolation of neighboring pixels, whereas disocclusions are only filled in by background pixels. This technique can be used with non-rectified views, avoiding prior creation of parallax maps as done in [Kauff et al(2007)Kauff, Atzpadin, Fehn, Müller, Schreer, Smolic, and Tanger].

In summary, the technique described here improves upon state-of-the-art DIBR methods as described in [Müller et al(2008b)Müller, Smolic, Dix, Merkle, Kauff, and Wiegand], by introducing the following key contributions:

- A novel forward projection method for DIBR, using occlusion compatible ordering [McMillan(1995)] for detecting cracks and disocclusions, for which the unknown depth values are estimated while performing the warping. The resulting projection method thus allows us to handle both depth maps warping and disocclusion filling simultaneously. Small cracks and large disocclusions are handled gracefully, with similar computational cost as simple forward projection, avoiding the use of the filtering step as done in the classical approach.

- A ghost removal method to avoid ghosting artifacts in the rendered views, relying on a depth-based pixel confidence measure.

- A depth-aided inpainting method which takes into account all information given by the depth map to fill in disocclusions with textures at the correct depth.

- A method to handle inaccuracies of cameras calibration and depth map estimation by the use of the Floating Texture approach.

The rest of the paper is organized as follows. Section 2 details some state-of-the-art solutions for each one of the three types of artifacts generated by the warping process, which are ghosting, cracks and disocclusions. Section 3 introduces the Joint Projection Filling (JPF) method, which simultaneously handles projection and disocclusion filling. Section 4 describes the two DIBR techniques, designed for synthesizing virtual views from one or many input views.

2 Background work

DIBR methods are based on warping techniques which project a reference view onto a virtual viewpoint. Directly applying warping equations may cause some visual artifacts in the synthesized view, like disocclusions, cracks and ghosting artifacts. Disocclusions are areas occluded in the reference viewpoint and which become visible in the virtual viewpoint, due to parallax effect. Cracks are small disocclusions, mostly due to texture re-sampling. Ghosts are artifacts due to projection of pixels that have background depth and mixed foreground/background color. Various methods have been proposed in the literature to avoid these artifacts. This section presents state-of-the-art solutions to avoid each one of these three usual artifacts.

Ghosting artifacts are often avoided by detecting depth discontinuities on the depth map, in order to separate the boundary layer (containing pixels near a boundary) from the main layer (containing pixels far from a boundary) [Zitnick et al(2004)Zitnick, Kang, Uyttendaele, Winder, and Szeliski]. The main layer is first projected into the virtual viewpoint, then the boundary layer is added everywhere it is visible (i.e. where its depth value is smaller than the main layer's one). In [Müller et al(2008b)Müller, Smolic, Dix, Merkle, Kauff, and Wiegand], the authors propose to split again the boundary layer into foreground and background boundary layers. The main layer is first projected, the foreground boundaries layer is then added everywhere it is visible, and the background boundaries layer is finally used to fill in remaining holes. Ghosting artifacts can be further avoided by estimating the background and foreground contributions in the rendered view with the help of advanced matting techniques [Hasinoff et al(2006)Hasinoff, Kang, and Szeliski, Sarim et al(2009)Sarim, Hilton, and Guillemaut, Wang and Cohen(2007)].

Cracks and other sampling artifacts are frequently avoided by performing a backward projection [Mori et al(2009)Mori, Fukushima, Yendo, Fujii, and Tanimoto], which works in three steps. At first, the depth map is warped with a forward projection, resulting in some cracks and disocclusions. Then, this virtual depth map is median filtered to fill cracks, and bilateral filtered to smoothen the depth map while preserving edges. Finally, the filtered depth map is warped back into the reference viewpoint to find the color of the synthesized views. In [Do et al(2009)Do, Zinger, Morvan, and de With], the authors propose to reduce the complexity by performing backward projection only for pixels labeled as cracks, i.e. pixels whose depth values are significantly modified by the filtering step. In [Nguyen et al(2009)Nguyen, Do, and Patel], the authors propose an improved occlusion removal algorithm, followed by a depth-color bilateral filtering, in order to handle disocclusions on the depth map. Other improved rendering methods based on surface splatting have been proposed for avoiding cracks and texture re-sampling artifacts [Rusinkiewicz and Levoy(2000), Pfister et al(2000)Pfister, Zwicker, van Baar, and Gross,

Zwicker et al(2002)Zwicker, Pfister, van Baar, and Gross].

Disocclusions are often filled in with information from some extra views, when they are available. The classical scheme is to synthesize the virtual view from each input view independently, then to blend the resulting synthesized views. In [Eisemann et al(2008)Eisemann, De Decker, Magnor, Bekaert, de Aguiar, Ahmed, Theobalt, and Sellent], the authors propose to compute an optical flow on intermediate rendered views, and then, with the help of the optical flow, to perform a registration step before the blending step, in order to avoid blurring in the final view due to blending mis-registered views. Note that specific representations such as Layered Depth Videos (LDV) can also be helpful for addressing the problem of occlusion handling since they allow storing texture information seen by other cameras [Shade et al(1998)Shade, Gortler, He, and Szeliski, Yoon et al(2007)Yoon, Lee, Kim, and Ho, Müller et al(2008a)Müller, Smolic, Dix, Kauff, and Wiegand, Jantet et al(2009)Jantet, Morin, and Guillemot].

When extra views are not available, the frequent solution for disocclusion handling is image interpolation with inpainting techniques. Unfortunately, most inpainting techniques use neighboring pixels solely based upon colorimetric distance, while a disocclusion hole should be filled in with background pixels, rather than foreground ones [Bertalmío et al(2001)Bertalmío, Bertozzi, and Sapiro, Telea(2004), Criminisi et al(2003)Criminisi, Pérez, and Toyama]. A good review on the use of inpainting for image-based rendering can be found in [Tauber et al(2007)Tauber, Li, and Drew]. In [Do et al(2009)Do, Zinger, Morvan, and de With], the authors estimate each pixel value inside a disocclusion area from nearest known pixels along the eight cardinal directions, after nullifying the weight of foreground pixels. In [Oh et al(2009)Oh, Yea, and Ho], the authors temporarily replace foreground textures by background texture before inpainting, so that disocclusions are filled in only with background texture.

Advanced depth-aided inpainting methods assume that the depth map of the virtual viewpoint to be rendered is available. In [Daribo and Pesquet(2010)], the authors enhance the inpainting method in [Criminisi et al(2003)Criminisi, Pérez, and Toyama] by reducing the priority of patches containing a depth discontinuity, and by adding a depth comparison in the search for best matches. In [Gautier et al(2011)Gautier, Le Meur, and Guillemot], the authors use a similar approach but estimate isophotes directions with a more robust tensor computation and constrain the propagation in the direction of the epipole.

The full depth map from the virtual view is most of the time not available, and must be estimated from the input depth map. In [Daribo and Pesquet(2010)], the authors perform a diffusion-based inpainting [Bertalmío et al(2001)Bertalmío, Bertozzi, and Sapiro] on the projected depth map, but both foreground and background are diffused to fill disocclusions. In [Nguyen et al(2009)Nguyen, Do, and Patel], the authors constrain the depth map inpainting in the direction of the epipole, in order that only the background is diffused, but this method fails when a disocclusion is surrounded by foreground depth, as shown in figure 2(b).

As a conclusion, state-of-the-art DIBR methods need a complete depth map at the rendered viewpoint (for backward projection and depth-aided inpainting). However, no fully satisfying method yet exists to obtain a complete and correct depth map, avoiding artifacts generation when used for DIBR. Moreover, most of

disocclusions handling methods proposed in the literature work as a post treatment on the projected view. Connectivity information is not preserved during the projection, and inpainting methods fail to fill in background disocclusions when they are surrounded by foreground objects. The proposed JPF method aims at suppressing such drawbacks. As shown in figure 2, the JPF method enables to recover correct depth information in critical areas. We also propose a full-Z depth-aided inpainting technique which takes into account the high quality of the computed depth map to fill disocclusions with texture from the correct depth.

3 Projection-based disocclusion handling

This section introduces the Joint Projection Filling (JPF) method, which simultaneously handles warping and disocclusion filling, in order to preserve connectivity and fill in disocclusions with background textures.

During warping, there might happen overlapping (several pixels projected at the same position) or disocclusion (no pixels projected at a position). In [McMillan(1995)], a pixel scanning order is introduced to perform the painter's algorithm during the projection. In case of overlapping, this pixel scanning order ensures the pixel just projected at a position to be the foreground pixel so that the z-buffer is not needed. A second property, resulting from the first one, is more helpful to handle disocclusions. If two successive pixels are not adjacent, there is a disocclusion, and the pixel just projected is the background pixel. This second property is exploited to ensure only background pixels are used to fill in disocclusion areas.

The JPF algorithm is described in section 3.1. It is first introduced for rectified cameras and then generalized for non-rectified cameras. Section 3.2 presents a ghosting removal method, based on pixels confidence measure. Finally, section 3.3 presents some synthesized textures and depth map, obtained by the JPF method.

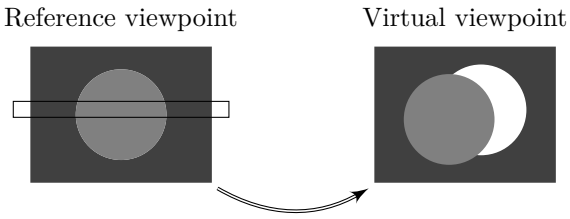
3.1 Disocclusion detection

In the following, we assume that the epipolar geometry is such that the pixels from the reference image are processed sequentially, from top-left to bottom-right, according to McMillan scanning order [McMillan(1995)].

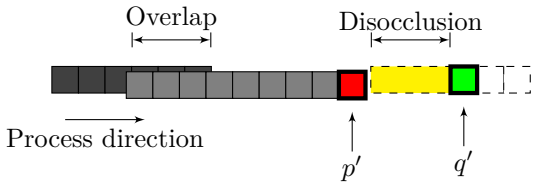
Figure 4(b) presents the principle of the Joint Projection Filling (JPF) method, in the particular case of rectified cameras. Each row is thus independent of the others, reducing the problem to one dimension. Consider a row of pixels from the reference view, and a pixel $p = (p_x, p_y)$ on that row. The pixel p is projected on position $p' = (p'_x, p_y)$ in the synthesized view. After having processed pixel p, the next pixel to be processed is $q = (p_x + 1, p_y)$. Its projected position $q' = (q'_x, p_y)$ verifies one out of the three following equations:

$$\begin{cases} q'_x = p'_x + 1 & \text{Pixels } p' \text{ and } q' \text{ are adjacent.} \\ q'_x < p'_x + 1 & \text{There is an overlap.} \\ q'_x > p'_x + 1 & \text{There is a crack or a disocclusion.} \end{cases} \tag{1}$$

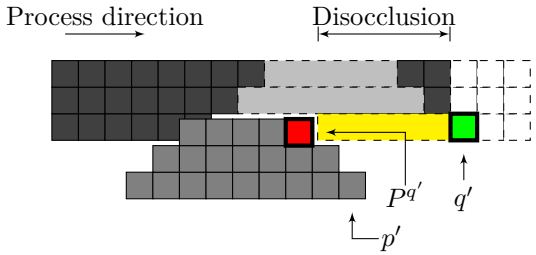
The first and the second cases do not generate artifacts. In the last case, p' and q' are in same order as p and q, but there is a gap between them. In the proposed method, contrary to classical point-based projection, this gap



(a) Projection.



(b) Details of a single row of pixels in the synthesized view for a rectified projection.



(c) Details of some rows of pixels in the synthesized view for a non rectified projection.

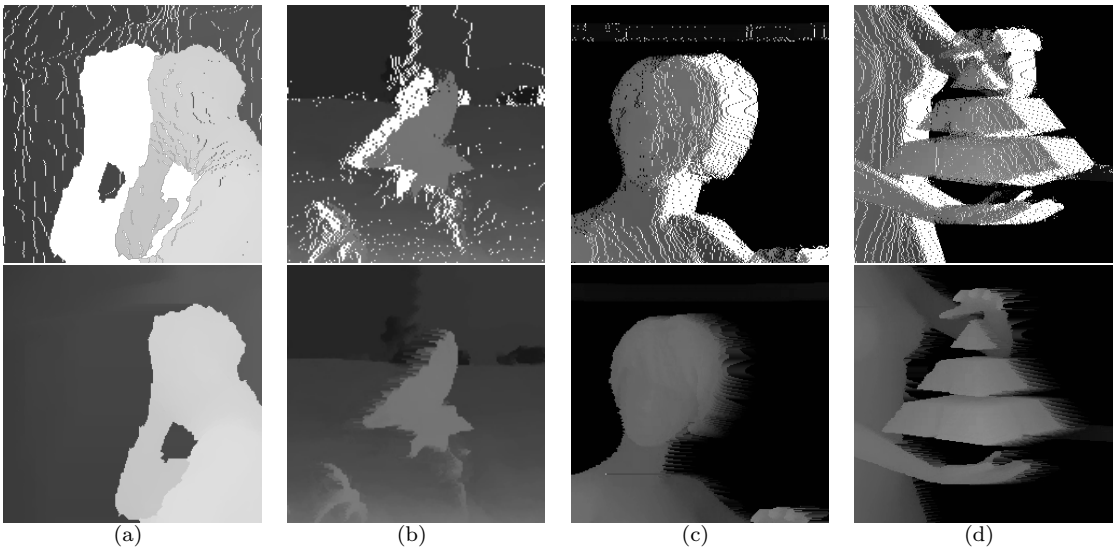
Figure 4: JPF method scheme for rectified 4(b) and non rectified 4(c) cameras. q' is a background pixel which is used to fill in the highlighted disocclusion.

is filled in immediately, before processing the projection of the next pixel. The method to fill the gap is adapted to its size. If the gap is small enough, it is considered as a crack. p' and q' are thus assumed to be on same layer, and the gap is filled in by interpolating the two pixels p' and q'. If the gap is too large, it is considered as a disocclusion. p' and q' are thus assumed to be on two distinct depth layer and the gap is filled in by background pixel. The McMillan pixel ordering ensures that q' is the background pixel, which is stretched from position p' to q'. The value of each pixel m between p' and q' is thus estimated as follows:

$$m = \begin{cases} (1-\alpha)p' + \alpha q' & \text{if } d \leq K \\ q' & \text{if } d > K \end{cases} \qquad \text{where} \begin{cases} d = q'_x - p'_x \\ \alpha = \frac{1}{d}(m_x - p'_x) \end{cases} \tag{2}$$

In the simulation results reported in the paper, the threshold K has been fixed to 5 pixels, to handle cracks and small disocclusions.

The algorithm is generalized for non-rectified cameras, as illustrated in figure 4(c). Pixels p' and q' may no longer be on the same row, thus we define pixel $P^{q'}$ as the last pixel projected on row q'_y. Equation (1) is revised,



(a) (b) (c) (d)

Figure 5: Comparison between synthesized depth maps from a forward point-based projection (first row) and from the JPF method (second row). Blurred depth discontinuities in the original depth map produces stretching effects on the synthesized depth maps. Note that McMillan scanning order is from right to left in figures 5(a) and 5(b), whereas it is from left to right in figures 5(c) and 5(d), due to epipole location for input images pairs.

replacing p' with $P^{q'}$, thus q' and $P^{q'}$ are on the same row.

$$\begin{cases} q'_x \leq P_x^{q'} + 1 & \text{There is no artifact.} \\ q'_x > P_x^{q'} + 1 & \text{There is a disocclusion.} \end{cases} \tag{3}$$

As previously, the disocclusion handling method depends on the distance between q'_x and $P_x^{q'}$. The value of each pixel m between $P^{q'}$ and q' is thus estimated as follows:

$$m = \begin{cases} (1-\alpha)P^{q'} + \alpha q' & \text{if } d \leq K \\ q' & \text{if } d > K \end{cases} \qquad \text{where} \begin{cases} d = q'_x - P_x^{q'} \\ \alpha = \frac{1}{d}(m_x - P_x^{q'}) \end{cases} \tag{4}$$

Figure 5 presents the synthesized depth maps obtained with the JPF method, without any ghosting removal technique. Our JPF method has removed all cracks and has filled in the disocclusions with only background pixels. Depth maps from the "Ballet" sequence contain sharp discontinuities, which are preserved by the JPF method (figure 5(a)). Depth maps from other sequences contain some blur along depth discontinuities, due to DCT-based compression. This blur produces some sparse pixels inside the disocclusion area, which are stretched to fill the disocclusion, resulting in an annoying ghosting artifact.

This occlusion-compatible ordering is helpful to detect cracks and disocclusions. Next section explains how to fill in disocclusions while preserving edges sharpness and avoiding ghosting artifacts.

3.2 Disocclusion filling

Pixels along objects boundaries are considered unreliable, because they often contain mixed foreground/background information for texture and depth value. Their projection may thus create ghosting

artifacts in the synthesized views. The JPF method as described in section 3.1 fills in each row of a disoccluded region using a single pixel. When applied on such a "blended" boundary pixel, this method may result in annoying pixel stretching artifacts, as can be seen in figure 5. However, these artifacts can be minimized by adapting the pixel stretching length, according to a pixel confidence measure. The algorithm used to avoid stretching and ghosting artifacts thus proceeds with the following two steps:

In a first step, a confidence measure $\lambda_q \in [0;1]$ is computed for each pixel q by convolving the depth map (Z) with a Difference-Of-Gaussians (DOG) operator as follows:

$$\lambda_q = 1 - (\mathrm{DOG} * \mathrm{Z})(q) \tag{5}$$

The DOG operator is built as the difference of two gaussians: the gaussian G of variance σ^2, and the 2D Dirac delta function δ_2.

$$\begin{aligned} \mathrm{DOG} &= G - \delta_2 \\ G(u,v) &= \frac{1}{\sigma^2} \cdot \phi\left(\frac{u}{\sigma}\right) \cdot \phi\left(\frac{v}{\sigma}\right) \end{aligned} \tag{6}$$

where ϕ is the standard normal distribution. The value of σ, in the experiments described below, has been fixed to 3.

In a second step, the confidence measure is used during the JPF method, to confine pixel stretching. Reusing the notations introduced in section 3.1, suppose that a wide disocclusion is discovered during the projection of pixel q. Instead of filling the whole gap between $P^{q'}$ and q', with color and depth values of q', only a part of the gap is filled in. The rest will be filled with the next pixel which will be projected on that same row j.

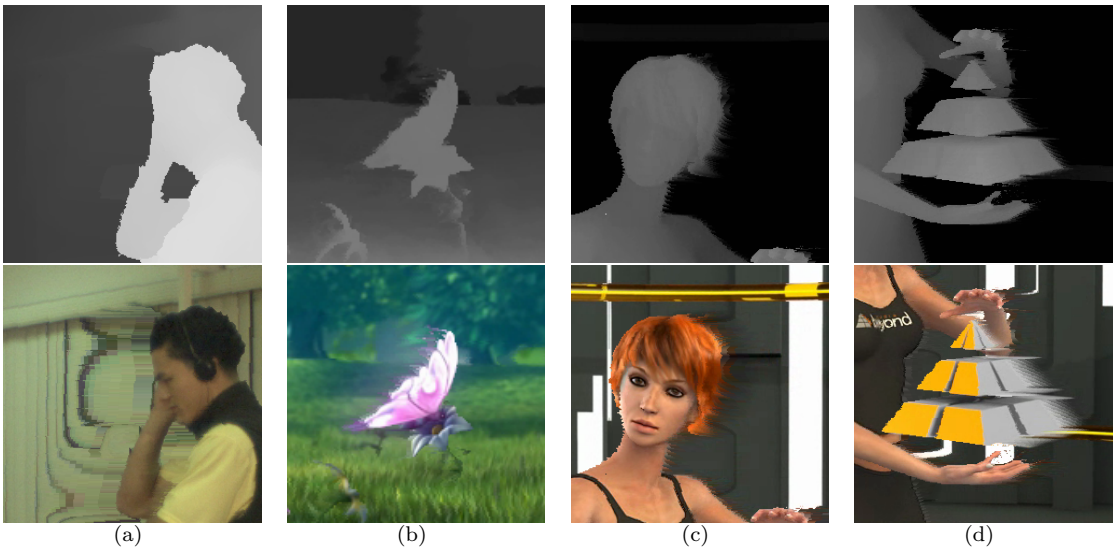
Assume M is a point between $P^{q'}$ and q', defined with the following equation:

$$M = (1 - \lambda_q^2)P^{q'} + \lambda_q^2 q' \tag{7}$$

The gap between $P^{q'}$ and M is filled in by pixel q', thus pixels on foreground/background boundaries which have low confidence measures are used to fill the disocclusion only for a couple of pixels next to the foreground, where blended pixels are expected to be in the synthesized view.

3.3 Results

This confidence-based interpolation method shifts back unreliable pixels near the discontinuities and only uses reliable pixels to fill in disocclusions. Figure 6 presents the rendering results of the JPF method with confidence-based interpolation. The projected depth maps, shown on the first row, are to be compared with those presented in figure 5. One can see that depth discontinuities are sharpened, producing realistic depth maps. The second row presents the results obtained with the same algorithm applied on texture. Disocclusions are gracefully filled in when the background is uniform, but annoying stretching artifacts appear in case of textured background. This JPF method can be used as a part of a virtual view synthesis algorithm, depending on the



(a) (b) (c) (d)

Figure 6: Warping results of the JPF method.

application. Two use cases are addressed in section 4, either for virtual view extrapolation when only one input view is available, or for intermediate view interpolation when multiple input views are available.

4 Virtual view rendering

The JPF method is designed to synthesize virtual views from one or many input view plus depth video sequences, depending on the final application. Section 4.1 describes a virtual view extrapolation algorithm, which is used when only one input view plus depth video sequence is available. Section 4.2 presents an interpolation algorithm to synthesize intermediate views when multiple video plus depth video sequences are available.

4.1 View extrapolation with full-Z depth-aided inpainting

In order to synthesize a virtual view from only one input view plus depth sequence, the classical rendering scheme, introduced in figure 1, is replaced by the one presented in figure 3. First, the depth map for the virtual view is synthesized by our JPF method, handling ghosting, cracks and disocclusions. Then, the texture of the virtual view is obtained by a classical backward warping followed by the proposed full-Z depth-aided inpainting algorithm.

Our proposed full-Z depth-aided inpainting algorithm is a modification of the depth-aided inpainting method described in [Daribo and Pesquet(2010)], itself based on the exemplar-based inpainting approach, introduced in [Criminisi et al(2003)Criminisi, Pérez, and Toyama]. Section 4.1.1 describes the exemplar-based inpainting approach introduced in [Criminisi et al(2003)Criminisi, Pérez, and Toyama]. Section 4.1.2 presents the depth-aided overlay introduced in [Daribo and Pesquet(2010)], which uses the depth map to drive the inpainting process. Section 4.1.3 describes our proposed modification which takes into account the high quality of the virtual depth map. The importance of the synthesized depth map quality is discussed in section 5, for three different depth-aided inpainting methods.

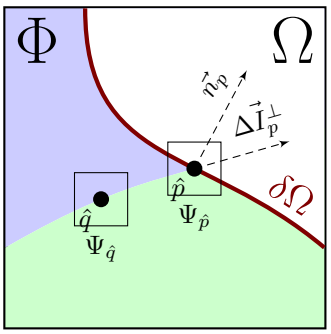


Figure 7: Notation diagram, introduced by Criminisi *et al.* [Criminisi et al(2003)Criminisi, Pérez, and Toyama]. Given the block Ψ_p, n_p is the normal to the contour $\delta\Omega$ of the hole region Ω. Φ is the non-hole region. $\Delta I_p^{\perp}$ is the isophote at point p.

4.1.1 Exemplar-based inpainting approach [Criminisi et al(2003)Criminisi, Pérez, and Toyama]

The inpainting algorithm introduced by Criminisi *et al.* in [Criminisi et al(2003)Criminisi, Pérez, and Toyama] is an exemplar-based inpainting technique. The authors noted that exemplar-based inpainting techniques can replicate both texture and structure, and they demonstrate that the quality of the output image synthesis is highly influenced by the order in which the inpainting is processed. Based on these observations, they describe in [Criminisi et al(2003)Criminisi, Pérez, and Toyama] an inpainting algorithm which iterates the following two steps until all missing pixels have been filled in. First, a priority term is computed, in order to determine the next patch of the image to be filled in. Second, this selected patch is filled in by copying a patch chosen as the best match for the known pixels of the patch to be filled in. These two steps are iterated until all missing pixels have been filled in.

Considering an input image I, and a missing region Ω, the source region Φ is defined as $\Phi = I - \Omega$ (see figure 7). For each pixel p along the frontier $\delta\Omega$, they define the patch Ψ_p centered in point p.

The priority $P(p)$ of the patch Ψ_p is computed as the product of the Confidence term $C(p)$ and the Data term $D(p)$.

$$P(p) = C(p) \cdot D(p) \tag{8}$$

The Confidence term $C(p)$ indicates the reliability of the current patch. It is nearly the ratio between the number of know pixels in the patch, compared to the size of the patch. $C(p)$ is initialized to 0 if $p \in \Omega$ or 1 if $p \in \Phi$, and then it is computed as follows:

$$C(p) = \frac{1}{|\Psi_p|} \sum_{q \in \Psi_p \cap \Phi} C(q) \tag{9}$$

where $|\Psi_p|$ is the number of pixels within the patch Ψ_p.

The Data term $D(p)$ is computed as the scalar product of the isophote direction $\Delta I_p^{\perp}$, and the unit vector n_p, orthogonal to $\delta\Omega$ at point p.

$$D(p) = \frac{|\Delta I_p^{\perp}, n_p|}{\alpha} \tag{10}$$

where α is a normalization factor (e.g. $\alpha = 255$ for a typical gray-level image).

Once all priorities on $\delta\Omega$ are computed, the patch $\Psi_{\hat{p}}$ with the highest priority is selected to be filled in. Then, a template matching algorithm search for the best exemplar $\Psi_{\hat{q}}$ to fill in missing pixels under $\Psi_{\hat{p}}$, as follows:

$$\Psi_{\hat{q}} = \arg\min_{\Psi_q \in \Phi} \{\mathrm{SSD}_{\Phi}(\Psi_{\hat{p}}, \Psi_q)\} \tag{11}$$

where $\mathrm{SSD}_{\Phi}(\cdot,\cdot)$ is the distance between two patches, defined as the Sum of Squared Differences and only computed on pixels from the non-hole region Φ.

The priority computation step and the best matches duplication step are iterated until all missing pixels have been filled in. The exemplar-based inpainting method produces good results for object removal and image restoration, but is not well suited to address the problem of disocclusions handling, because disocclusions should be filled in with background texture only.

4.1.2 Extension to depth-aided inpainting [Daribo and Pesquet(2010)]

The Criminisi's inpainting method has been adapted in [Daribo and Pesquet(2010)], to address the specific problem of disocclusion handling. The authors propose two major modifications, which are the addition of a new term $L(p)$ into the priority function evaluation, and the consideration of the depth into the SSD computation.

The first modification consists in introducing the Level regularity term $L(p)$ as a third term into the priority function $P(p)$.

$$P(p) = C(p) \cdot D(p) \cdot L(p) \tag{12}$$

The Level regularity term $L(p)$ is defined as the inverse variance of the depth patch Z_p:

$$L(p) = \frac{|Z_p|}{|Z_p| + \sum_{q \in Z_p \cap \Phi} \left(Z_p(q) - \bar{Z}_p\right)^2} \tag{13}$$

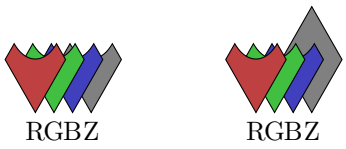
where $|Z_p|$ is the area (in terms of number of pixels) of depth patch Z_p centered in p, $Z_p(q)$ is the depth value at pixel location q under Z_p, and $\bar{Z}_p$ the mean value.

The second modification consists in adding depth into SSD computation, computed during the best match search. Equation 11 is thus modified as follows:

$$\Psi_{\hat{q}} = \arg\min_{\Psi_q \in \Phi} \{\mathrm{SSD}_{\Phi}(\Psi_{\hat{p}}, \Psi_q) + \alpha\, \mathrm{SSD}_{\Phi}(Z_{\hat{p}}, Z_q)\} \tag{14}$$

where the parameter α controls the importance given to the depth distance minimization.

The Level regularity term $L(p)$ gives more priority to patches with a constant depth level, which is expected to favor background pixels over foreground ones. Considering depth into the SSD computation favors patches which are at the same depth as the patch to be copied.



(a) (b)

Figure 8: Part of the patch $\Psi_{\hat{p}}$ which is involved in SSD computation. In [Daribo and Pesquet(2010)], authors compute SSD with color (RGB) and depth (Z) information only from the known part of the patch $\Psi_{\hat{p}} \cap \phi$, as shown in figure 8(a). Instead, we use the depth of the full patch to compute SSD, as shown in figure 8(b).

4.1.3 Proposed full-Z depth-aided inpainting

The synthesized depth map does not contain holes, thanks to the JPF method which projects the input depth map onto the virtual viewpoint while filling cracks and disocclusions. The patch $\Psi\hat{p}$ to be filled in contains thus a depth value for each pixel, even for pixels in the hole region Ω. These depth values are close to the ground truth, because disocclusions are only filled in with background depth. The proposed modification is to use the depth value of all pixels in the patch, including those whose color is not known. Equation 14 is thus modified as follows:

$$\Psi_{\hat{q}} = \arg\min_{\Psi_q \in \Phi} \{\mathrm{SSD}_{\Phi}(\Psi_{\hat{p}}, \Psi_q) + \alpha\, \mathrm{SSD}_{\Phi \cup \Omega}(Z_{\hat{p}}, Z_q)\} \tag{15}$$

Figure 8 shows the part of patch $\Psi\hat{p}$ which is involved in SSD computation.

Results of the proposed full-Z depth-aided inpainting method are analyzed in section 5, and compared with results from the two other depth-aided inpainting methods.

4.2 Intermediate view interpolation with Floating Texture

Intermediate view rendering methods fill in disocclusions with texture information from many input views. The classical scheme works as follows: Intermediate views are first synthesized by projecting each input view onto the virtual viewpoint, using the backward projection described in figure 1; The backward projection removes cracks and sampling artifacts with three time-consuming steps, which are a forward warping step, a filtering step and a backward warping step; The final rendered view is then computed by blending intermediate views together. Disocclusions are thus filled in with corresponding textures from side views. Depending on the correctness of the estimated depth maps, and the accuracy of the cameras calibration, depth information coming from each view may be inconsistent, resulting in blurring artifacts after blending, as shown in figure 10(a).

The proposed solution adapts the Floating Texture approach, introduced in [Eisemann et al(2008)Eisemann, De Decker, Magnor, Bekaert, de Aguiar, Ahmed, Theobalt, and Sellent], which uses an optical-flow-based warping to correct for local texture misalignments. Figure 9 presents each step of the proposed intermediate view interpolation with the Floating Texture registration. Each input view plus depth sequence is first forward projected onto the virtual viewpoint. The JPF method is used to warp both color and depth maps. This projection method allows handling cracks without the need of a backward projection. Each intermediate view is then registered using an optical-flow-based warping technique. Optical

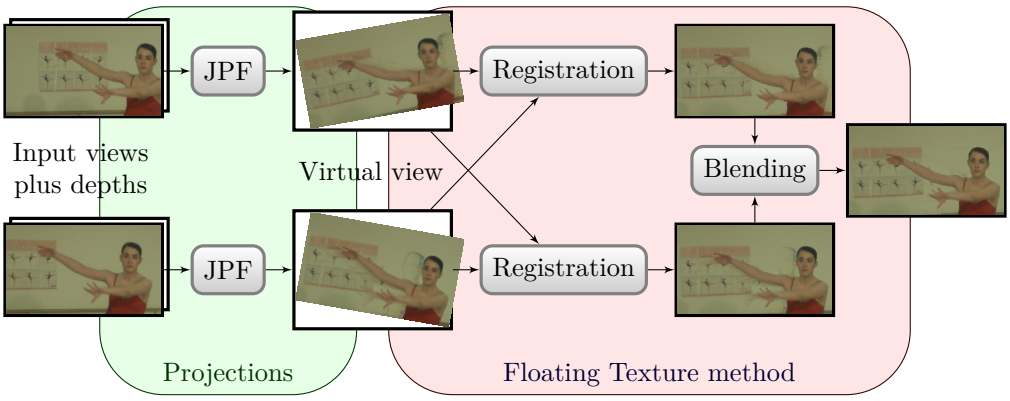


Figure 9: Floating Texture scheme for intermediate view synthesis. Each input view is projected onto the virtual viewpoint, using the Joint Projection Filling (JPF) method to fill in disocclusions and preserve contours. Projected views are realigned with the Floating Texture algorithm, then blended together with a weighting based on the virtual view position and confidence score.

(a) Without Floating Texture.

(b) With Floating Texture.

(c) Optical flow [Zach et al(2007)Zach, Pock, and Bischof].

Figure 10: Results of Floating Texture algorithm for view synthesis from multiple views.

flows are computed with the algorithm described in [Zach et al(2007)Zach, Pock, and Bischof], which provides the best results for Floating Texture registration. Finally, the registered views are blended together using a weighting scheme based on the confidence score, computed in section 3.2.

Figure 10 presents an intermediate view, rendered by our proposed method. Blending intermediate views together, weighted by the confidence score, allows us to fill in disocclusions with real texture. Blurring artifacts may appear if intermediate views are misaligned, as shown in figure 10(a). Applying the Floating Texture approach removes blur on contours and texture details are enhanced, as shown in figure 10(b). One of the flow fields estimated with the method proposed in [Zach et al(2007)Zach, Pock, and Bischof] is shown in figure 10(c). Colors represent directions, and luminosities represent norms.

5 Rendering Results

This section compares virtual view synthesis results obtained when using three depth-aided inpainting techniques for occlusion handling. For each inpainting techniques, the virtual depth maps are synthesized either by the classical scheme, shown in figure 1, or by the JPF method.

Figure 11 shows inpainting results of the algorithm presented in [Daribo and Pesquet(2010)]. Figure 12 shows inpainting results of the algorithm presented in [Gautier et al(2011)Gautier, Le Meur, and Guillemot]. Figure 13 shows inpainting results of the proposed full-Z depth-aided inpainting algorithm. In each figure, the

first column shows a virtual view synthesized by the backward projection, where disocclusions appear in white. The second column shows the virtual depth map where disocclusions are filled in with a Navier-strokes inpainting algorithm [Bertalmío et al(2001)Bertalmío, Bertozzi, and Sapiro], whereas the fourth column shows the depth map synthesized with our JPF method. The third and the fifth columns show the results of the depth-aided inpainting method, led by the depth map respectively presented in column 2 and 4.

One can observe that the depth maps shown in column 2 are not realistic because depth discontinuities do not fit with object boundaries. This is due to the depth map inpainting method, which fills disocclusions with both background and foreground values. On the contrary, depth maps presented in column 4 are closer to the ground truth, thanks to the JPF method. Small details are well preserved by the projection, as fingers on row 3 or blades of grass on row 4.

The influence of the virtual depth map can be observed by comparing column 3 and 5 of each figure. Errors in depth map from column 2 are amplified by every depth-aided inpainting method, because some foreground patches are selected to fill in disocclusions. The resulting images, shown in column 3, contain more artifacts than the ones obtained with a correct depth map.

Depth-aided inpainting methods can be compared with each other by analyzing the fifth column of each figure. Rendering results shown in figures 11 and 12 still contains blur artifacts along boundaries, even if the correct depth map is used to conduct the inpainting process. The proposed full-Z depth-aided inpainting method preserves small details, as fingers on row 3 or blades of grass on row 4.

As a conclusion, the quality of the rendered view is strongly dependent on the quality of the virtual depth map, no matter the depth-aided inpainting method. Synthesizing high quality virtual depth map is thus an interesting challenge for DIBR techniques. The JPF method is well suited for this purpose, because connectivity information is used during the forward projection. Moreover, the proposed full-Z depth-aided inpainting method improves upon state-of-the-art methods by taking into account the correctness of the synthesized depth map.

6 Conclusion

This article describes two DIBR techniques, relying on the Joint Projection Filling (JPF) method to improve the rendering quality.

The JPF method is based on McMillan's occlusion-compatible ordering introduced in [McMillan(1995)]. It allows cracks handling and disocclusions filling by ensuring that only background pixels are used during interpolation. The confidence-based pixels shifting avoids ghosting artifacts and texture stretching while sharpening discontinuities. In terms of computational complexity, this JPF method is equivalent to classical point-based projection and can be used with non-rectified views. Synthesized depth maps are very similar to ground truth depth maps.

The first DIBR technique is a virtual view extrapolation, based on a depth-aided inpainting technique. The JPF method is used here to synthesize the depth map of the virtual view without errors, in order to conduct the depth-aided inpainting. The depth-aided inpainting takes into account the high quality of the generated depth

map to select correct patches to be duplicated.

The second DIBR technique is an virtual view interpolation method, which uses the Floating Texture algorithm [Eisemann et al(2008)Eisemann, De Decker, Magnor, Bekaert, de Aguiar, Ahmed, Theobalt, and Sellent] to sharpen the final view. The JPF method is used here to produce intermediate views without cracks nor disocclusions, in order to process the optical flow estimation.

References

[Bertalmío et al(2001)Bertalmío, Bertozzi, and Sapiro] Bertalmío M, Bertozzi A, Sapiro G (2001) Navier-stokes, fluid dynamics, and image and video inpainting. In: Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on, Los Alamitos, CA, USA, vol 1, pp 355–362, DOI 10.1109/CVPR.2001.990497 6, 16, 21, 22, 23

[Chan et al(2007)Chan, Shum, and Ng] Chan S, Shum HY, Ng KT (2007) Image-based rendering and synthesis. Signal Processing Magazine, IEEE 24(6):22–33, DOI 10.1109/MSP.2007.905702 2

[Criminisi et al(2003)Criminisi, Pérez, and Toyama] Criminisi A, Pérez P, Toyama K (2003) Object removal by exemplar-based inpainting. In: Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on, vol 2, pp 721–728, DOI 10.1109/CVPR.2003.1211538 6, 11, 12

[Daribo and Pesquet(2010)] Daribo I, Pesquet B (2010) Depth-aided image inpainting for novel view synthesis. Multimedia Signal Processing (MMSP), IEEE International Workshop on pp 167–170, DOI 10.1109/MMSP.2010.5662013 6, 11, 13, 14, 15, 21

[Do et al(2009)Do, Zinger, Morvan, and de With] Do L, Zinger S, Morvan Y, de With PHN (2009) Quality improving techniques in dibr for free-viewpoint video. In: 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, pp 1 –4, DOI 10.1109/3DTV.2009.5069627 5, 6

[Eisemann et al(2008)Eisemann, De Decker, Magnor, Bekaert, de Aguiar, Ahmed, Theobalt, and Sellent] Eisemann M, De Decker B, Magnor M, Bekaert P, de Aguiar E, Ahmed N, Theobalt C, Sellent A (2008) Floating textures. Computer Graphics Forum (Proc of Eurographics) 27(2):409–418, received the Best Student Paper Award at Eurographics 2008 6, 14, 17

[Gautier et al(2011)Gautier, Le Meur, and Guillemot] Gautier J, Le Meur O, Guillemot C (2011) Depth-based image completion for view synthesis. In: 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video 6, 15, 22

[Hartley and Zisserman(2004)] Hartley RI, Zisserman A (2004) Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, ISBN: 0521540518 2

[Hasinoff et al(2006)Hasinoff, Kang, and Szeliski] Hasinoff SW, Kang SB, Szeliski R (2006) Boundary matting for view synthesis. Comput Vis Image Underst 103:22–32, DOI 10.1016/j.cviu.2006.02.005, URL http://portal.acm.org/citation.cfm?id=1148410.1148412 5

[Jantet et al(2009)Jantet, Morin, and Guillemot] Jantet V, Morin L, Guillemot C (2009) Incremental-ldi for multi-view coding. In: 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, Potsdam, Germany, pp 1–4, DOI 10.1109/3DTV.2009.5069647 6

[Kauff et al(2007)Kauff, Atzpadin, Fehn, Müller, Schreer, Smolic, and Tanger] Kauff P, Atzpadin N, Fehn C, Müller M, Schreer O, Smolic A, Tanger R (2007) Depth map creation and image-based rendering for advanced 3dtv services providing interoperability and scalability. Signal Processing: Image Communication 22:217–234, DOI 10.1016/j.image.2006.11.013, URL http://portal.acm.org/citation.cfm?id=1231529.1231663 4

[McMillan(1995)] McMillan L (1995) A list-priority rendering algorithm for redisplaying projected surfaces. Tech. Rep. 95-005, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.1759 4, 7, 16

[Mori et al(2009)Mori, Fukushima, Yendo, Fujii, and Tanimoto] Mori Y, Fukushima N, Yendo T, Fujii T, Tanimoto M (2009) View generation with 3d warping using depth information for ftv. Image Commun 24:65–72, DOI 10.1016/j.image.2008.10.013, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4547850 5

[Müller et al(2008a)Müller, Smolic, Dix, Kauff, and Wiegand] Müller K, Smolic A, Dix K, Kauff P, Wiegand T (2008a) Reliability-based generation and view synthesis in layered depth video. Multimedia Signal Processing (MMSP), IEEE International 10th Workshop on pp 34–39, DOI 10.1109/MMSP.2008.4665045 6

[Müller et al(2008b)Müller, Smolic, Dix, Merkle, Kauff, and Wiegand] Müller K, Smolic A, Dix K, Merkle P, Kauff P, Wiegand T (2008b) View synthesis for advanced 3d video systems. EURASIP Journal on Image and Video Processing p 11, DOI 10.1155/2008/438148 4, 5

[Nguyen et al(2009)Nguyen, Do, and Patel] Nguyen QH, Do MN, Patel SJ (2009) Depth image-based rendering from multiple cameras with 3d propagation algorithm. In: Proceedings of the 2nd International Conference on Immersive Telecommunications, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, IMMERSCOM '09, vol 6, pp 1–6, URL http://portal.acm.org/citation.cfm?id=1594108.1594116 3, 5, 6

[Oh et al(2009)Oh, Yea, and Ho] Oh KJ, Yea S, Ho YS (2009) Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video. In: Picture Coding Symposium (PCS), IEEE Press, Piscataway, NJ, USA, pp 233–236, DOI 10.1109/PCS.2009.5167450 6

[Pfister et al(2000)Pfister, Zwicker, van Baar, and Gross] Pfister H, Zwicker M, van Baar J, Gross M (2000) Surfels-surface elements as rendering primitives. pp 335–342, URL http://www.merl.com/papers/TR2000-10/ 5

[Rusinkiewicz and Levoy(2000)] Rusinkiewicz S, Levoy M (2000) Qsplat: a multiresolution point rendering system for large meshes. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '00, pp 343–352, DOI 10.1145/344779.344940 5

[Sarim et al(2009)Sarim, Hilton, and Guillemaut] Sarim M, Hilton A, Guillemaut JY (2009) Wide-baseline matte propagation for indoor scenes. In: Conference Visual Media Production (CVMP), Proceedings of, IEEE Computer Society, Washington, DC, USA, CVMP '09, pp 195–204, DOI 10.1109/CVMP.2009.6 5

[Shade et al(1998)Shade, Gortler, He, and Szeliski] Shade J, Gortler S, He Lw, Szeliski R (1998) Layered depth images. In: SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM, New York, NY, USA, pp 231–242, DOI 10.1145/280814.280882, URL http://grail.cs.washington.edu/projects/ldi/ 6

[Shum and Kang(2000)] Shum HY, Kang SB (2000) A review of image-based rendering techniques. Institute of Electrical and Electronics Engineers, Inc., URL http://research.microsoft.com/pubs/68826/review_image_rendering.pdf 2

[Sourimant(2010)] Sourimant G (2010) Depth maps estimation and use for 3dtv. Technical Report 0379, INRIA Rennes Bretagne Atlantique, Rennes, France 2

[Tauber et al(2007)Tauber, Li, and Drew] Tauber Z, Li ZN, Drew M (2007) Review and preview: Disocclusion by inpainting for image-based rendering. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37(4):527 –540, DOI 10.1109/TSMCC.2006.886967 6

[Telea(2004)] Telea A (2004) An image inpainting technique based on the fast marching method. Journal of Graphics, GPU, and Game Tools 9(1):23–34 6

[Wang and Cohen(2007)] Wang J, Cohen MF (2007) Image and video matting: a survey. Found Trends Comput Graph Vis 3:97–175, DOI 10.1561/0600000019, URL http://portal.acm.org/citation.cfm?id=1391083.1391084 5

[Yoon et al(2007)Yoon, Lee, Kim, and Ho] Yoon SU, Lee EK, Kim SY, Ho YS (2007) A framework for representation and processing of multi-view video using the concept of layered depth image. Journal of VLSI Signal Processing Systems for Signal Image and Video Technology 46:87–102, URL http://vclab.gist.ac.kr/papers/01/2007/1_2007_SUYOON_A_Framework_for_Representation_and_Processing_ 6

[Zach et al(2007)Zach, Pock, and Bischof] Zach C, Pock T, Bischof H (2007) A duality based approach for realtime tv-l1 optical flow. In: Pattern recognition, 29th DAGM conference on, Springer-Verlag, Berlin, Heidelberg, pp 214–223, URL http://portal.acm.org/citation.cfm?id=1771530.1771554 15

[Zhang and Chen(2004)] Zhang C, Chen T (2004) A survey on image-based rendering–representation, sampling and compression. Signal Processing: Image Communication 19(1):1–28, DOI 10.1016/j.image.2003.07.001, URL http://www.sciencedirect.com/science/article/B6V08-49NV92R-2/2/73e2a2577939e1d6dd4c124d4d0fabcc 2

[Zitnick et al(2004)Zitnick, Kang, Uyttendaele, Winder, and Szeliski] Zitnick CL, Kang SB, Uyttendaele M, Winder S, Szeliski R (2004) High-quality video view interpolation using a layered representation. ACM Trans Graph 23(3):600–608, DOI 10.1145/1015706.1015766, URL http://research.microsoft.com/~larryz/videoviewinterpolation.htm 5

[Zwicker et al(2002)Zwicker, Pfister, van Baar, and Gross] Zwicker M, Pfister H, van Baar J, Gross M (2002) Ewa splatting. Visualization and Computer Graphics, IEEE Transactions on 8:223–238, DOI 10.1109/TVCG.2002.1021576 5

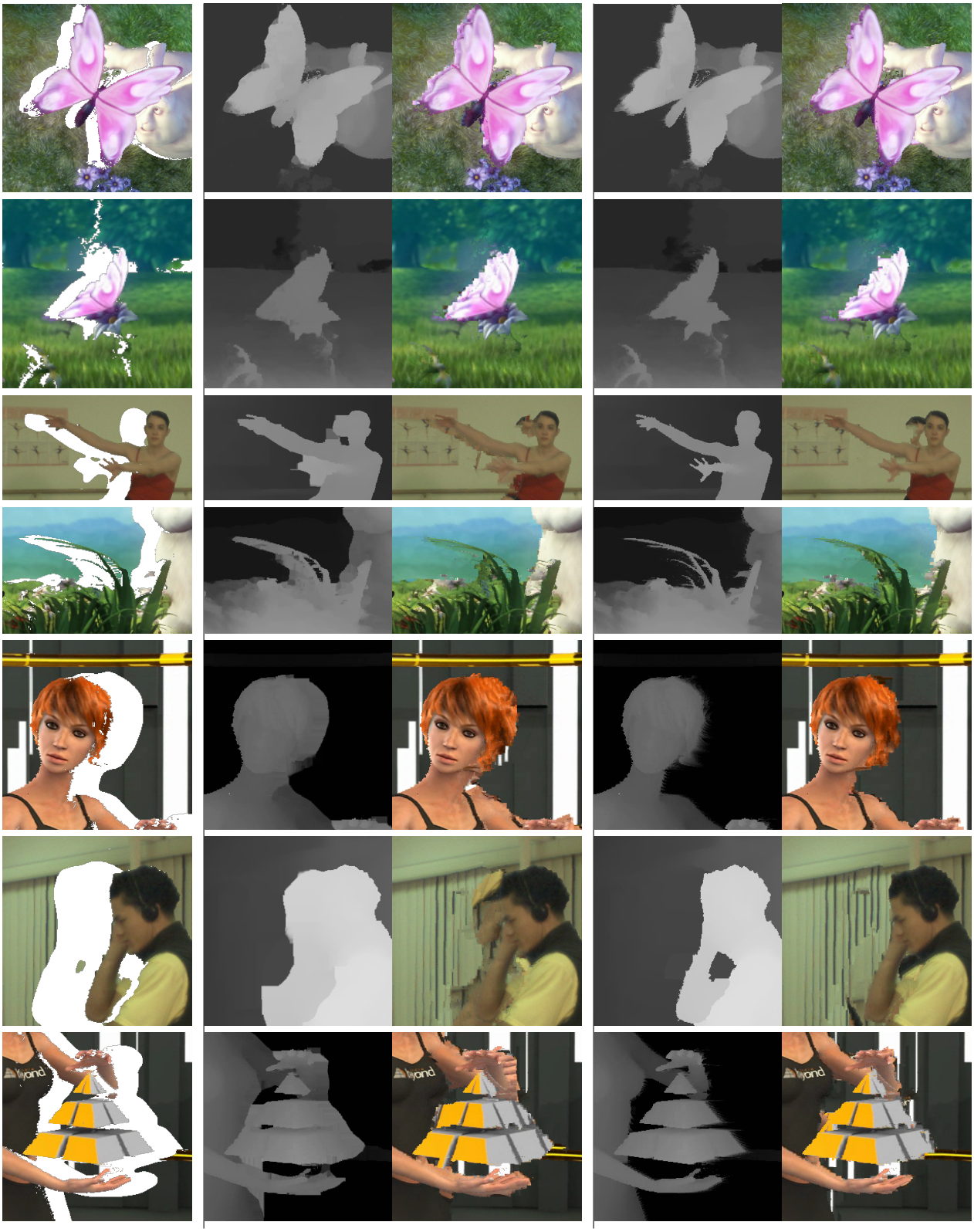


Figure 11: Results for Daribo depth-aided inpainting [Daribo and Pesquet(2010)]. The first column shows a synthesized view with disocclusions. Columns 2 and 4 present the synthesized depth maps, obtained respectively with a Navier-strokes inpainting algorithm [Bertalmío et al(2001)Bertalmío, Bertozzi, and Sapiro] and with our JPF method. Columns 3 and 5 exhibit the results of the inpainting of the texture shown in column 1, guided by the depth map respectively presented in columns 2 and 4.

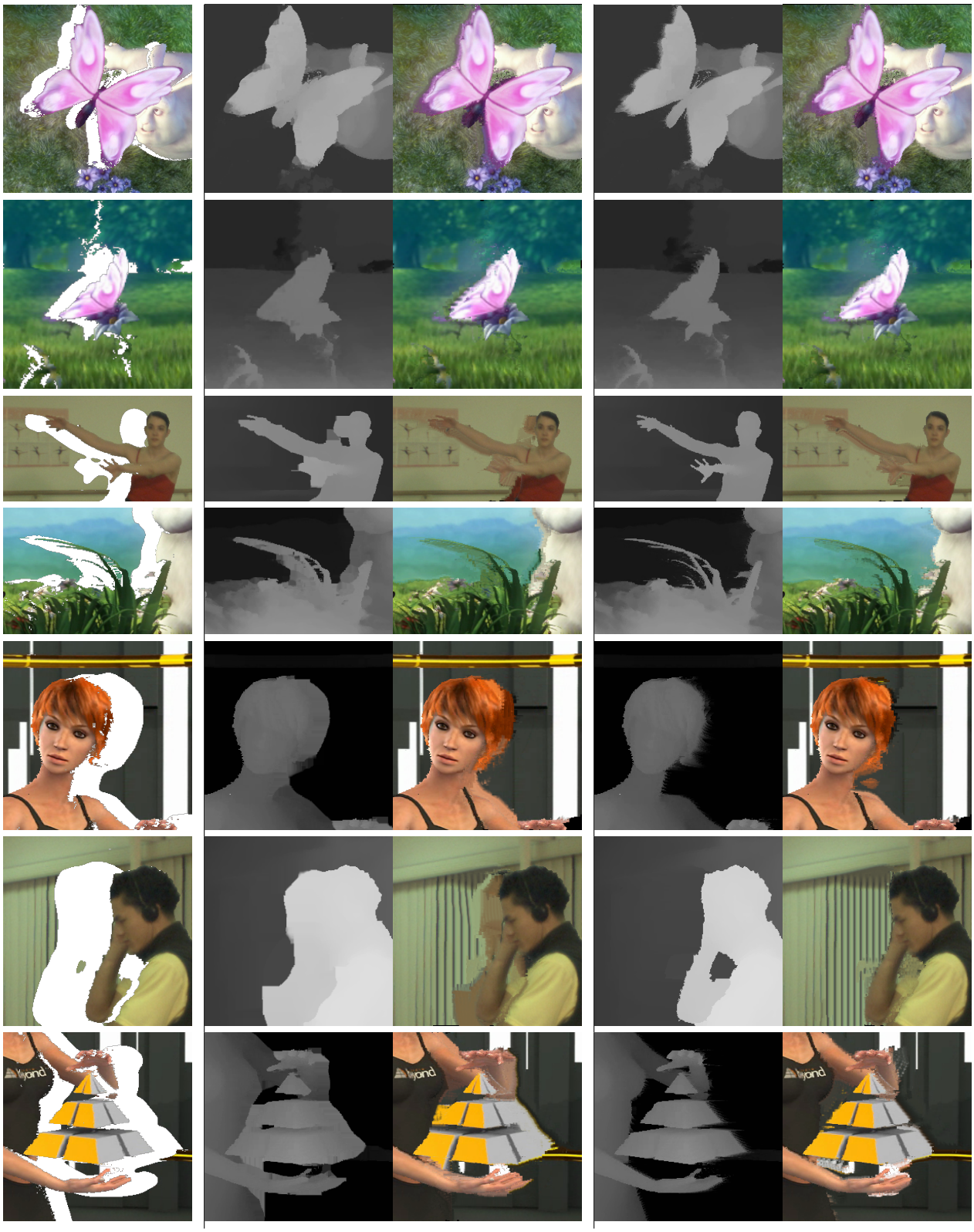


Figure 12: Results for Gautier depth-aided inpainting [Gautier et al(2011)Gautier, Le Meur, and Guillemot]. The first column shows a synthesized view with disocclusions. Columns 2 and 4 present the synthesized depth maps, obtained respectively with a Navier-strokes inpainting algorithm [Bertalmío et al(2001)Bertalmío, Bertozzi, and Sapiro] and with our JPF method. Columns 3 and 5 exhibit the results of the inpainting of the texture shown in column 1, guided by the depth map respectively presented in columns 2 and 4.

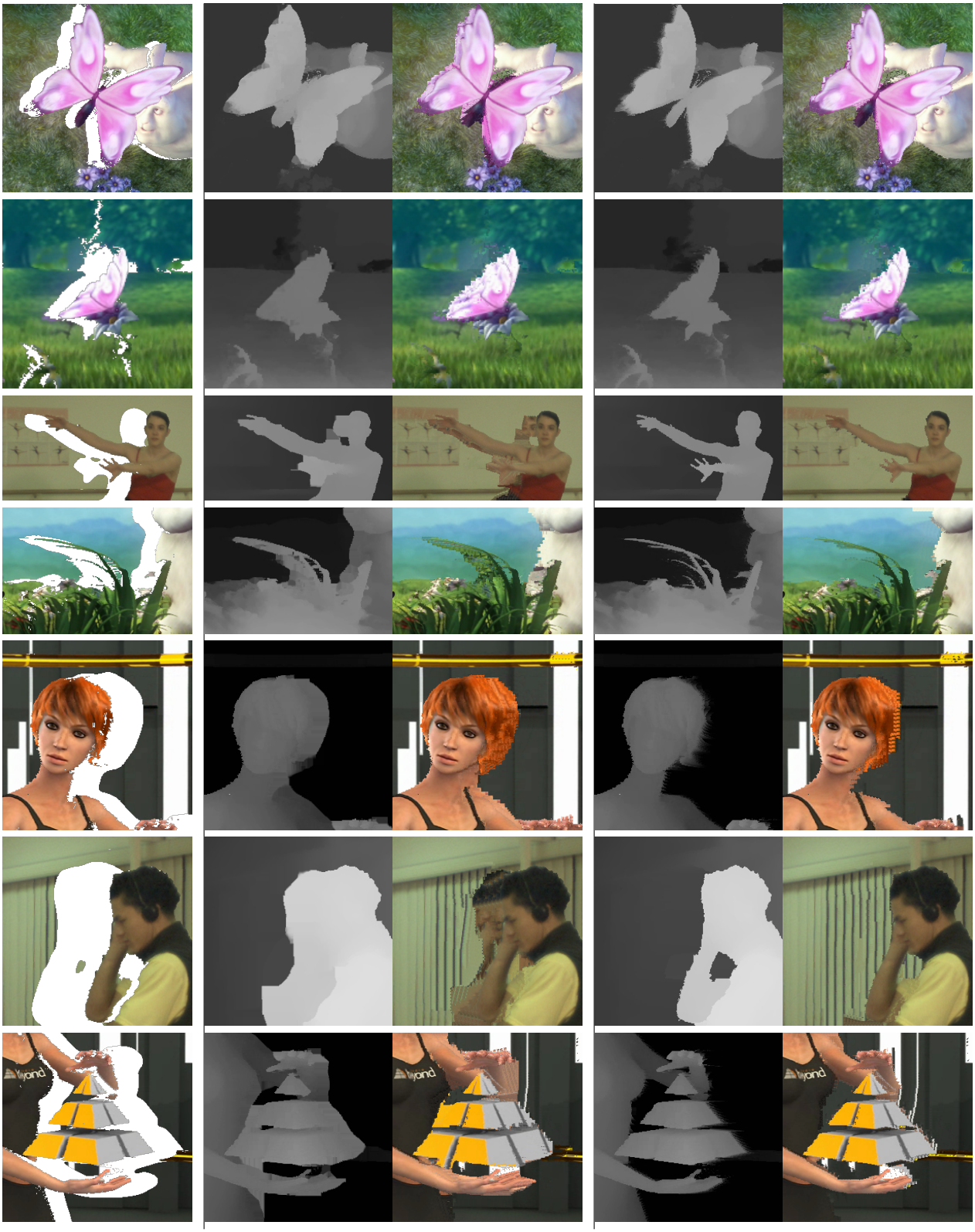


Figure 13: Results for proposed full-Z depth-aided inpainting. The first column shows a synthesized view with disocclusions. Columns 2 and 4 present the synthesized depth maps, obtained respectively with a Navier-strokes inpainting algorithm [Bertalmío et al(2001)Bertalmío, Bertozzi, and Sapiro] and with our JPF method. Columns 3 and 5 exhibit the results of the inpainting of the texture shown in column 1, guided by the depth map respectively presented in columns 2 and 4.