



HAL
open science

Depth-guided disocclusion inpainting of synthesized RGB-D images

Pierre Buysens, Olivier Le Meur, Maxime Daisy, David Tschumperlé, Olivier
Lézoray

► **To cite this version:**

Pierre Buysens, Olivier Le Meur, Maxime Daisy, David Tschumperlé, Olivier Lézoray. Depth-guided disocclusion inpainting of synthesized RGB-D images. *IEEE Transactions on Image Processing*, 2017, 26 (2), pp.525-538. 10.1109/TIP.2016.2619263 . hal-01391065

HAL Id: hal-01391065

<https://hal.science/hal-01391065v1>

Submitted on 2 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Depth-guided disocclusion inpainting of synthesized RGB-D images

Pierre Buysens, Olivier Le Meur, Maxime Daisy, David Tschumperlé, Olivier Lézoray

Abstract—We propose to tackle the disocclusion inpainting of RGB-D images when synthesizing new views of a scene by changing its viewpoint. Such a process creates holes both in depth and color images. First, we propose a novel algorithm to perform the depth-map disocclusion inpainting. In particular, this intuitive approach is able to recover the lost structures of the objects and to inpaint the depth-map in a geometrically plausible manner. Then, we propose a depth-guided patch based inpainting method to fill-in the color image. Depth information coming from the reconstructed depth-map is added to each key step of the classical patch-based algorithm from Criminisi *et al.* in an intuitive manner. Relevant comparisons to state-of-the-art inpainting methods for the disocclusion inpainting of both depth and color images are provided and illustrate the effectiveness of our proposed algorithms.

Index Terms—RGB-D view synthesis, depth-map disocclusion inpainting, color image disocclusion inpainting

I. INTRODUCTION AND CONTEXT

3DTV and the more general Free-Viewpoint Rendering (FVR) are promising technologies that stimulate the emergence of new multimedia experiences such as 3D cinema, display and broadcasting, to name a few. In this context, Depth Image Based Rendering (DIBR) has then become an important feature when capturing a scene: it consists in rendering both the classical color image together with a depth-map of the scene that encodes the relative depth of each pixel according to the camera position. Given this pair of images, one can synthesize a new virtual view by modifying the point of view of the scene through a *warping* transformation [1].

A critical issue arises when one wants to synthesize such a virtual view: background objects that lied behind foreground objects that have been uncovered have to be rendered in the new view. These unknown parts of the synthesized view are the so-called *occluded areas*, and filling them is known as the *disocclusion inpainting* process, which is a particular case of the *inpainting* problem. Note that the areas to be disoccluded appear both in the newly rendered color image and the corresponding depth-map at the same location.

Figure 1 illustrates this phenomenon with a right horizontal translation of the viewpoint: the known parts (depicted in red and blue) are partially rendered in the virtual view. A non negligible part of the virtual view is unknown (depicted in

green): the background that was hidden by the red foreground object, and the one that was offscreen.

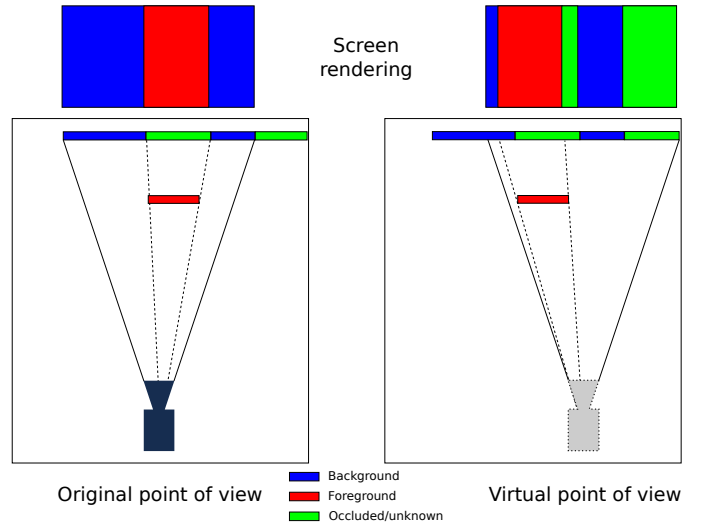


Fig. 1. Illustration of a view synthesis (right) with a right translation of the original view point (left). The corresponding renderings on a screen are depicted on the top of the figure.

In this paper we propose to tackle both depth-map and color image disocclusion inpainting. As pointed by Yoon *et al.* in [2], inpainting the depth-map is definitely not an easy task, and requires dedicated algorithms: Using the reconstructed depth-map to guide the color image disocclusion inpainting algorithm leads to more flexible approaches.

Our contributions are multiple. First, we propose a thorough analysis of the depth-map disocclusion inpainting problem and show that current state-of-the-art approaches cannot properly inpaint common complex cases. We then propose an intuitive 3-steps approach that involves only one *easy-to-adjust* parameter (λ in the following). Compared to state-of-the-art approaches, our method results in better visual reconstruction with a good recovery of lost structures. Second, we extend our previous work [3], [4] on color image disocclusion inpainting. We provide new insights and details of our method, in addition to the inclusion of a smart heuristic for the best patch search scheme. We also compare our approach to representative state-of-the-art methods by providing qualitative and quantitative results on challenging images and reference datasets.

Based on the patch-based inpainting algorithm of Criminisi *et al.* [5], our proposed depth-aware patch-based disocclusion inpainting method for color images revisits each of its key steps by incorporating the depth information in an intuitive way. The proposed approach involves only one extra parameter

Pierre Buysens and Olivier Le Meur are with the IRISA laboratory, UMR 6074, INRIA, CNRS, Université de Rennes 1, Rennes, France.

Maxime Daisy, David Tschumperlé and Olivier Lézoray are with the GREYC laboratory, UMR 6072, CNRS, Université de Caen Normandie, ENSICAEN, Caen, France.

Email : pierre.buysens@gmail.com, olivier.le_meur@irisa.fr, {maxime.daisy, david.tschumperle}@ensicaen.fr, olivier.lezoray@unicaen.fr

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

λ as compared to the seminal approach of [5]. This threshold parameter λ discriminates adjacent pixels into foreground and background according to their respective depths. Particularly, two adjacent pixels p and q belong to the same *object* (foreground or background) if $|\text{depth}(p) - \text{depth}(q)| < \lambda$.

One has to note that, in the following, we deal with *disparity* maps that encode the displacement of each pixel used to synthesize a new view. Computing a depth-map (where a foreground object has a low depth) from a disparity map (where a foreground object has high disparity values) can be easily processed through an inverse mapping function. Figure 2 shows together with a color image its corresponding depth and disparity maps. In the rest of the paper, we use the term *depth-map* and deal specifically with horizontal translation warpings, though more general warpings, such as vertical, rotational or forward/backward warpings, can be also considered with some (small) modifications (not detailed in this paper).



Fig. 2. Example of a color image (excerpt from the *Flowers* image), its corresponding depth-map (middle) and its corresponding disparity-map (right).

The paper is organized as follows: In section II, we review the major methods proposed in the literature for the disocclusion inpainting of both color and depth images. In section III, we first detail the main challenges of depth-map disocclusion inpainting and propose our approach to tackle them. We then compare it to existing approaches on challenging cases. In section IV, we detail our depth-aware patch-based approach for color image disocclusion inpainting. We compare it qualitatively and quantitatively to representative approaches of the state-of-the-art on images gathering both structures and textures difficulties. Finally, we draw some conclusions in section V.

II. RELATED WORK

Here, we review the methods proposed in the literature for the disocclusion inpainting of synthesized RGB-D images. We propose a classification of these methods depending on the concerned modality: most of the state-of-the-art approaches deal with the disocclusion inpainting of color images assuming that the depth-map is already inpainted. Among these works, a few propose a specific approach towards depth-map disocclusion inpainting, and some authors perform a joint disocclusion inpainting of both depth-map and color image at the same time. Note that we only review here the state-of-the-art approaches that deal with horizontal translation warping. Some work deals with more general warpings such as *depth* translation of the viewpoint (closer to the scene) [6], [7], which is clearly beyond the scope of this paper.

Before delving into this review, let us first define some notations used throughout this paper.

A. Notations and definitions

A color image is considered as a function $I : \mathcal{I} \rightarrow \mathbb{R}^n$, where \mathcal{I} defines the image domain, and $n = 3$ for usual color RGB images. Similarly, a depth-map is considered as a function $J : \mathcal{J} \rightarrow \mathbb{R}$, where \mathcal{J} is the depth-map domain. In the following, a RGB-D image is considered as the pair (I, J) that shares the same domain $\mathcal{I} = \mathcal{J}$. As we deal with occluded images, the pair (I, J) contains a set of holes $\Omega = \{\Omega_1, \dots, \Omega_N\}$. Ω represents the mask of (I, J) that has to be filled (i.e., the set of unknown pixels that have to be resynthesized), and $\delta\Omega$ is the boundary of Ω .

All state-of-the-art methods use *patches* for color image disocclusion inpainting. A patch Ψ_p centered on the pixel p is considered as a function $\Psi_p : \mathcal{N}_p \rightarrow \mathbb{R}^n$ where $\mathcal{N}_p \subset \mathcal{I}$ is the square support of size s of Ψ_p and n is the dimension of the considered image ($n = 1$ for a depth-map, and $n = 3$ for a color image). It is worth noting that Ψ_p can be masked/incomplete (i.e., some of its pixels are unknown). $|\mathcal{N}_p| = s^2$ denotes the size of the support of a patch (i.e., its number of pixels), while $|\Psi_p|$ denotes the number of *known* pixels of the patch Ψ_p . In the case of an incomplete patch (that contains unknown pixels), one has obviously $|\Psi_p| < |\mathcal{N}_p|$.

$\Psi_{\hat{p}}$ denotes a patch that matches Ψ_p according to a given metric:

$$\Psi_{\hat{p}} = \left\{ \Psi_q \mid \arg \min_{q \in (\mathcal{N}_q \cap (\mathcal{I} \setminus \Omega))} d(\Psi_p, \Psi_q) \right\}$$

The widely used distance d to compare the visual similarity of two patches is the *Sum of Square Differences* (SSD):

$$d_{SSD}(\Psi_p, \Psi_q) = \sum_{v \in (\mathcal{N}_p \cap (\mathcal{I} \setminus \Omega))} \|\Psi_p(v) - \Psi_q(v + p - q)\|^2 \quad (1)$$

Note that other patch similarity measures have been used in the literature such as the *Bhattacharya* distance [8] or the *Hellinger* distance [9] combined with the SSD metric. Nevertheless, the advantages of using such distances remains still unclear for the general purpose of image inpainting.

Finally, we denote by $\mathcal{W} : \mathcal{I} \rightarrow \mathcal{I}$ the general *warping* function that transforms an *original* color image I_o into a *synthesized* one $I_s = \mathcal{W}(I_o)$. Since this paper only deals with horizontal translation warpings (to simulate parallel cameras), the same warping function warps also the corresponding depth-map J_o into a *synthesized* depth-map $J_s = \mathcal{W}(J_o)$. Throughout this paper, we perform the synthesis of the novel view through such a warping process that can be done with standard 3D warping equations [10].

B. Disocclusion Inpainting of color images

We consider here that the adjoining depth-map does not contain holes (i.e., it has already been filled).

Since most of state-of-the-art approaches rely on the seminal patch-based inpainting algorithm of Criminisi *et al.* [5], we first review this particular algorithm.

1) *Sketch of patch-based inpainting algorithm:* Back in 2004, a breakthrough inpainting approach was proposed in [5]. Based on the work proposed in [11] for texture reconstruction, this greedy patch-based inpainting algorithm introduces the notion of *priority* that guides the filling order, and consists in iterating the 4 following steps:

- a) A priority term is assigned to each pixel $p \in \delta\Omega$ (exterior border of Ω), and is computed as

$$P(p) = C(p) \times D(p) \quad (2)$$

where $C(p)$ and $D(p)$ are the so-called *confidence* and *data* terms respectively. The first term reflects the number of reliable (known) pixels in \mathcal{N}_p , while the second term is computed from local gradient in \mathcal{N}_p and reflects the structures that enter the mask. They are defined in [5] as:

$$C(p) = \frac{\sum_{q \in (\mathcal{N}_p \cap (\mathcal{I} \setminus \Omega))} C(q)}{|\mathcal{N}_p|} \quad (3)$$

$$D(p) = \frac{\nabla I_p^\perp \cdot \mathbf{n}_p}{\alpha} \quad (4)$$

where α is a normalization factor (that can in fact be ignored), \mathbf{n}_p is the unit vector orthogonal to the boundary of the mask $\delta\Omega$ at p , and $C(p) = 1, \forall p \in \bar{\Omega} = (\mathcal{I} \setminus \Omega)$. The pixel $t \in \delta\Omega$ found to have the maximum priority is chosen as the *target* pixel (i.e., point to reconstruct first).

- b) Given the incomplete target patch Ψ_t centered on t , the second step consists in searching in $\bar{\Omega}$ the patch $\Psi_{\hat{t}}$ that minimizes the SSD between ψ_t and $\Psi_{\hat{t}}$ among the known part of Ψ_t (Eq. 1).
c) Paste the values from $\Psi_{\hat{t}}$ to the unknown pixels of Ψ_t :

$$\Psi_t(q) = \Psi_{\hat{t}}(p) \mid q - t = p - \hat{t}, \forall q \in \mathcal{N}_t \cap \Omega \quad (5)$$

- d) Update $\delta\Omega$ as well as *confidence* and *data* terms, and go back to step a).

Numerous small improvements of this seminal algorithm have been proposed later for the purpose of image inpainting, but reviewing them is clearly beyond the scope of this paper. A comprehensive review can be found in [12].

2) *Depth-aware color image disocclusion inpainting:* Let us now focus on the various modifications proposed in the literature of this patch-based inpainting algorithm for color image disocclusion inpainting. Two angles of attack have been mainly investigated to add depth information into the algorithm of Criminisi *et al.*: (a) modification of the priority term (step 1 of the algorithm of Criminisi *et al.*), and (b) modification of the patch searching/matching process (step 2).

a) *Modifications of the priority term:*

The authors of [13] proposed to add a third multiplicative term as $P(p) = C(p) \times D(p) \times L(p)$ where $L(p)$ is a depth regularity term, defined as the inverse variance of the depth patch Ψ_p^d centered on p :

$$L(p) = \frac{|\mathcal{N}_p|}{|\mathcal{N}_p| + \sum_{q \in (\mathcal{N}_p \cap (\mathcal{I} \setminus \Omega))} (\Psi_p^d(q) - \bar{\Psi}_p^d)^2} \quad (6)$$

where Ψ_p^d is the patch defined on the depth-map centered at p and $\bar{\Psi}_p^d$ is the depth mean value of Ψ_p^d . The rationale behind

this additional term is to favor target pixels p that stand in flat regions ($L(p) \simeq 1$) and to leave pixels that lie at the frontier between foreground and background ($L(p) \ll 1$) for the end of the inpainting process.

Similarly, a new priority term is proposed in [14] that involves an additional depth-related term

$$P(p) = C(p) \times D(p) \times \left(1 - \frac{\bar{\Psi}_p^d}{z_{max}}\right) \quad (7)$$

where $\bar{\Psi}_p^d$ is the average depth of the pixels of Ψ_p^d and z_{max} denotes the global maximum depth value of the depth-map.

In [15], the priority is kept unchanged, but the set of pixels $p \in \delta\Omega$ are discretized between foreground and background pixels, and only those that belong to the background have an effective assignment of their priorities (the priority of the pixels of the boundary that lie on the foreground side of the hole is set to 0).

In a similar manner, authors of [16] proposed to set to 0 the priority of pixels that lie on the right side of the hole if the camera has moved from right to left. The idea behind this trick is that the holes appear in this case to the left of the foreground objects, and by setting to 0 the priority of the pixels that lie on the right side of the hole, it prevents the inpainting process to begin from the foreground. In addition to *one side priority* scheme, they also propose to use a 3D tensor [17] for the *data* term computation by including depth information in addition to the color features to favor both color and geometric structures.

Authors of [18] proposed to modify the filling order such that it also favors the beginning of the algorithm from the background, as well as the propagation of edges inside the mask:

$$P(p) = F(p) \times D(p) \times M(p) \times \frac{|\Psi_p|}{|\mathcal{N}_p|} \quad (8)$$

where $F(p)$ is a binary function such that $F(p) = 0$ if p belongs to the foreground, and $F(p) = 1$ if p belongs to the background, and $M(p)$ is an additional term computed at p as

$$M(p) = \frac{\sum_{r \in \mathcal{N}_p \cap \Omega} \sum_{q \in (\mathcal{N}_p \cap (\mathcal{I} \setminus \Omega))} e^{-\frac{(J(r) - J(q))^2}{2\sigma}}}{|\Psi_p| (|\mathcal{N}_p| - |\Psi_p|)}$$

Finally, authors of [2] propose to compute the priority term as

$$P(p) = C(p)^\alpha \times D(p)^\beta \times E(p)^\gamma \quad (9)$$

where $C(p)$ and $D(p)$ are the initial confidence and data terms (Eq. 3), $E(p)$ is the inverse of the disparity map, and $\{\alpha, \beta, \gamma\}$ are hyperparameters arbitrary fixed by the authors. Beyond these parameters, the rationale behind the additional term $E(p)$ is to give highest priority to the background (low disparity thus high $E(p)$ values) than to the foreground.

b) *Modifications of the searching/matching process:*

Beyond the modifications of the filling order, most of the existing methods propose also to modify the way the best patch is searched (step 2 of the algorithm summarized in Section II-B1).

Authors of [13] and [16] incorporate the depth-map into the SSD computation as a fourth channel. While authors of [13] perform a weighting of this depth *channel* through an additional weighting parameter, authors of [16] chose to consider it as important as the color channels. Moreover, authors of [16] retrieve the $K = 5$ best patches and perform a linear combination of these best patches weighted by their respective SSD.

This search scheme favors the selection of patches that have similar depths without preventing picking a patch in the foreground.

Another scheme is to restrict the search only to patches that have lower depths. This is especially employed by [2], [15] with a tolerance parameter ϵ fixed by the authors. By restricting the search of the best patch to candidate patches of lower depths, it becomes impossible to inpaint the background with a foreground patch, which is the desired property.

Finally, authors of [18] formulate the matching process as a multiplicative combination between two similarity terms. The first one, computed on the non-masked part of Ψ_t , is related to the similarity of both color and depth, while the second one, computed on the masked part of Ψ_t , is only related to the similarity between depths.

These two angles of attack mainly investigated in the literature involve the modifications of the definition of the *priority* term and/or the best patch search scheme. Most of these modifications consider only the case of simple holes (a hole that is only surrounded by background and a single foreground object), and are demonstrated on few images. Beyond the extra parameters that have been tuned manually (which can be tedious), the robustness of the proposed *priority* terms is definitely not guaranteed with more complex (and indeed common) cases involving multiple objects at different depths that overlap.

In Section IV, we tackle the color image disocclusion inpainting by proposing natural and efficient modifications of the priority term computation, the best patch search scheme, and also the copy/paste scheme. We also propose to specifically deal with the aliasing problem (related to the matting problem) by providing a simple and efficient method. To our knowledge, no state-of-the-art method specifically deal with this phenomenon.

C. Disocclusion Inpainting of depth-maps

As opposed to the disocclusion inpainting of color images, the problem of the depth-maps disocclusion inpainting has received considerably less attention.

It is worth noting the depth-map disocclusion inpainting method used in [15], [18] which fills a given hole with a line-based algorithm. Given a pixel p to inpaint, the leftmost and rightmost known pixels are found, and the one with the minimal depth (that corresponds to the background) is retained as the depth value of the drawn line.

Arguing that the depth-map is smooth by nature, authors of [13] proposed to inpaint the depth-map with an isotropic diffusion inpainting scheme, previously defined in [19].

The authors of [2] recently proposed to use the original depth-map J_o to help finding the *good* depth values to inpaint a given hole. Using such a scheme relies on the inverse warping function \mathcal{W}^{-1} , and allows to properly inpaint, with background values, holes that are only surrounded by a foreground object. The effective inpainting is then processed with a line drawing scheme. The original depth-map was also used in our previous work [20]. However, we proposed to use a superpixel-based process both to get a more robust method and to modelize the depth-map.

Finally, it is worth mentioning depth-maps inpainting methods such as [21] dedicated to the restoration of depth-maps obtained from a kinect for instance. Although dealing with a problem neighbor to the disocclusion inpainting, these kind of methods use the fully known color image, which is not the case in disocclusion inpainting (i.e., the color image also contains holes).

The main flaw of these approaches (as for the color image disocclusion inpainting methods), is that they only work well in the case of simple holes, and do not scale to complex ones involving multiple objects at different depths that overlap. In such cases, some object structures have to be recovered and current state-of-the-art methods are insufficient.

In Section III, we tackle the depth-map disocclusion inpainting by proposing a simple 3-steps algorithm. The core of our proposal is the structure propagation step that allows to reconstruct missing structure of the objects prior to the effective inpainting. A structure completion step is indeed blatantly missing from the state-of-the-art methods.

D. Joint inpainting

Among the widely used scheme that separates the depth-map disocclusion inpainting from the color image one, a few works aimed at performing a joint disocclusion inpainting.

Authors of [22] proposed first a modification of the priority term to enforce the priorities of the background pixels:

$$P(p) = (C(p) + D(p) + L(p)) \times \left(z_{max} - \overline{\Psi}_p^d \right) \quad (10)$$

where $L(p)$ is the depth-related term defined in [13] (Eq. 6), z_{max} is the maximum depth value of the entire depth-map, and $\overline{\Psi}_p^d$ the average depth value of the known pixels of the patch Ψ_p^d defined on the depth-map. The algorithm consists then in alternating texture and depth pixels inpainting.

Authors of [23] also inpaint both color image and depth-map in parallel. The background boundary of a hole is identified by a classification performed on the depth values of the pixels of this boundary, and a priority value is only given to these pixels (the priority of the remaining pixels is set to 0). The adjoining background is then set as the source region, and the effective inpainting is performed with a linear combination of the K best patches found in this source region.

Finally, authors of [24] propose a superpixel-based approach to perform a joint disocclusion inpainting of both the color image and the depth-map. The main advantage is that superpixels, as opposed to patches, theoretically contain only foreground (or background) pixels, and this avoids the

typical foreground leaking problem.

Performing the disocclusion inpainting of the color and the depth images at the same time seems appealing. However, since the main challenges of color image and depth-map inpainting are completely different by nature, it seems more than challenging to provide a robust algorithm that is able to manage both simple and complex cases at the same time.

III. DEPTH-MAP DISOCCLUSION INPAINTING

Before detailing our proposed depth-map disocclusion inpainting method, we first review the main challenges and show why current state-of-the-art methods are insufficient to deal with.

A. Problem analysis

The disocclusion inpainting problem arises when the newly synthesized viewpoint reveals areas that were initially hidden. The disocclusion inpainting of depth-map is definitely a complex task, and have been unfortunately tackled with methods that are not good enough to manage complex geometric cases.

We now detail the main challenges of such a disocclusion inpainting and illustrate them through the *Art* and *Backpack* images (from the Middlebury dataset [25]) which contains many foreground objects of different depths (Figure 3):

- As illustrated in Figure 3 (right), some holes (like the one pointed by the red arrow) have to be filled with background values but are only surrounded with foreground values. Inpainting such a hole with the diffusion approach of [13] or with the line-based method of [15], [18] leads to filling it with foreground values. To our knowledge, only the methods [2], [20] that make use of the original depth-map J_o are able to fill it with the *correct* background values.
- When the scene contains multiple foreground objects that overlap, the structure of a partly hidden foreground object (rings in Figure 3) may be broken in the synthesized depth-map (green arrows on the right image of Figure 3). Moreover, the structures are not necessarily horizontal. To our knowledge, no state-of-the-art method are currently able to deal with this phenomenon.
- The same phenomenon of broken structures may also appear when a foreground object touches the border of the original depth-map J_o . These structures, which are not necessarily horizontal, have to be propagated (green arrow in Figure 3, middle image).
- Apart from the holes that have to be filled, it is always assumed in state-of-the-art algorithms that the depths of the warped depth-map are correct and do not have to be overwritten. In case of multiple foreground objects that overlap, it may not always be the case. Figure 3 (right) illustrates such a case (rightmost green arrows): the structure of the ring has to be completed, and this implies overwriting some background values that have been put there by the warping process.

Apart from these challenges, the background planes are not necessarily parallel to the camera plane. Inpainting them with horizontal lines of constant depth (as proposed in [2], [15], [18]) necessarily leads to inconsistencies in the resulting depth-map.

B. Proposed method

Preliminaries: Our proposed method relies on a diffusion technique as in [13] but with several modifications to tackle all the challenges detailed above. First it is necessary to underline that holes mainly appear in the synthesized view when there are depth discontinuities in the original depth-map. Fortunately, these discontinuities can be easily detected through the use of morphological operators. We denote NB_o^+ the *exterior* narrow band of the original depth-map, that is the binary image defined as the exterior morphological gradient thresholded by λ and computed as:

$$NB_o^+ = \mathcal{T}(\delta(J_o) - J_o, \lambda) \quad (11)$$

where $\delta(\cdot)$ and $\mathcal{T}(\cdot, \lambda)$ are the dilation and the λ -thresholding operators respectively. Here, $\delta(J_o) - J_o$ is indeed the *extern* morphological gradient of J_o [26].

This narrow band image states for each pixel if it belongs to the background or not, according to its direct neighborhood. Warping this narrow band image leads to the *exterior* narrow band of the synthesized depth-map:

$$NB_s^+ = \mathcal{W}(NB_o^+) \quad (12)$$

This narrow band defines the starting pixels for the *diffusion*. Figure 4 shows in green the narrow bands on both the original (left) and synthesized (middle) depth-maps. The main problem of the diffusion arises when multiple objects with different depths overlap each other. Figure 4 (right) shows the result of the diffusion-based inpainting scheme [13] while taking NB_s^+ as source. Important inconsistencies can be noticed, especially where the rings are broken. Moreover, when multiple objects with different depths are involved, a given hole has to be filled both with background and foreground depth values. In such a case, the diffusion-based inpainting scheme [13] produces smooth transitions (Figure 4, right) between background and foreground depths, and the typical sharpness of the contours is lost. Finally, and as pointed in Section III-A (first point), there may remain small holes that have not been inpainted since their boundaries do not contain any pixel of the exterior narrow band NB_s^+ (see Figure 4, right).

Details of the proposed method: We now propose to tackle the main challenges exposed above with a simple and intuitive approach. It can roughly be decomposed into 3 steps: 1) Detect the points where a structure is potentially broken, and propagate these structures, 2) Diffuse the depths from well-defined regions to fill-in the unknown depths (with a possible overwrite of background values), 3) Inpaint the remaining holes (essentially those of the first point in Section III-A) with the use of the original depth-map.

1) *Structure propagation:* A structure is potentially broken when multiple objects with different depths overlap. To propagate these structures, we first detect the set of points that are located at the junction between 3 areas of different depths. Since these so-called *t-junctions* $tj(o)_{i,i \in \{1, \dots, n\}}$ belong to the *exterior* boundary of an object and to the *interior* boundary

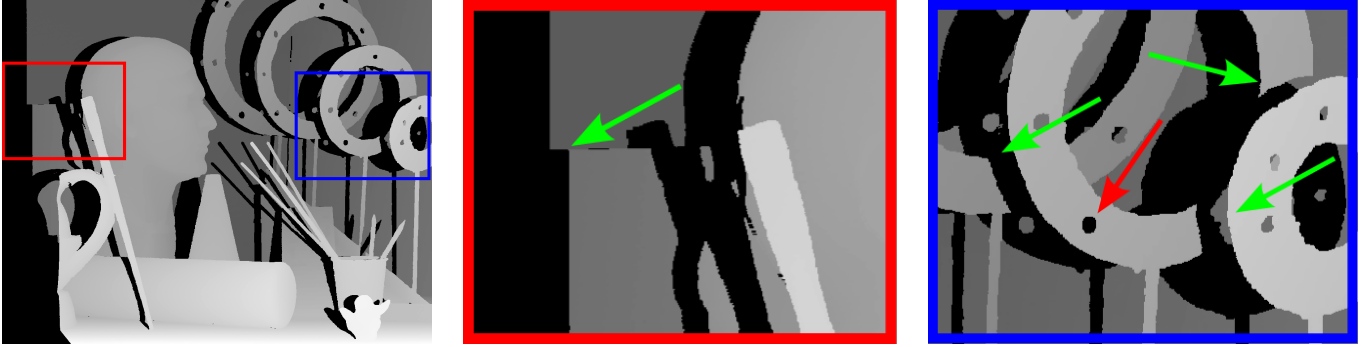


Fig. 3. Illustration of the main difficulties of the depth-map inpainting on the *Art* image. *Left*: synthesized depth-map (J_s). *Middle* and *right*: zoomed parts of the left image. The arrows point to the main challenges of the depth-map disocclusion inpainting problem.



Fig. 4. *Left*: exterior narrow band NB_o^+ superimposed on the original depth-map J_o . *Middle*: exterior narrow band NB_s^+ superimposed on the synthesized depth-map J_s . Inpainting result with [13] while taking the exterior narrow band NB_s^+ (middle) as source for the diffusion. See text for details.

of another object, they are detected in the original depth-map as:

$$\{tj(o)_{i,i \in \{1, \dots, n\}}\} = \{p \in J_o \mid p \in (NB_o^- \cap NB_o^+)\} \quad (13)$$

where NB_o^+ is the exterior narrow band (eq. 11) and NB_o^- is the interior narrow band of the original depth-map, computed as:

$$NB_o^- = \mathcal{T}(J_o - \epsilon(J_o), \lambda) \quad (14)$$

where $\epsilon(\cdot)$ is the erosion operator. Here $J_o - \epsilon(J_o)$ is indeed the *intern* morphological gradient of J_o [26].

Once these *t-junctions* are detected in J_o , the corresponding *t-junctions* in the synthesized depth-map J_s are retrieved with the warping function \mathcal{W} . Figures 5 and 6 (first row) plot as blue dots the *t-junctions* points in both original and synthesized depth-maps for the *Backpack* and *Art* images respectively. Note that there are fewer *t-junctions* in J_s than in J_o : because of the warping process, some of them are now masked by a foreground object.

The structures of the partially hidden foreground object can now be retrieved from these *t-junctions*. In this work, we simply extend these structures linearly, whose directions are found by looking at the interface between the foreground object to extend and the background in the original depth-map J_o . Checking this interface into a small neighborhood around the considered *t-junction* is sufficient to retrieve a *good* enough direction to propagate the linear structure. Starting from a *t-*

junction, the contour is then propagated until one of these conditions is reached:

- the extended line reaches an object of higher depth,
- two lines having the same depth (depth of their respective *t-junctions*) do cross,
- the line reaches the border of the image.

It is important to note here that these lines can go through some known pixels whose depth are lower (i.e., *background* pixels).

Figure 6 (second row) shows some of these lines in red together with their associated *t-junctions* depicted in blue. Notice how some lines go through background pixels (second row, right image).

2) *Depth diffusion*: Beyond the sharpness of the structures that is well recovered, one important advantage of the structure propagation scheme detailed above is as follows: a given hole that has to be filled both with background and foreground values is now split (according to the extended lines) into homogeneous parts. Such a hole is shown in Figure 6 (bottom right): the parts of the hole that lie between two extended lines can be filled with the depth of the foreground ring, while the rest of the hole can be filled with background values only.

The effective inpainting can now be carried out safely via a diffusion scheme with the extended lines acting as barriers. Starting from each border of NB_s^+ , the created fronts are stopped if one of these conditions is met:

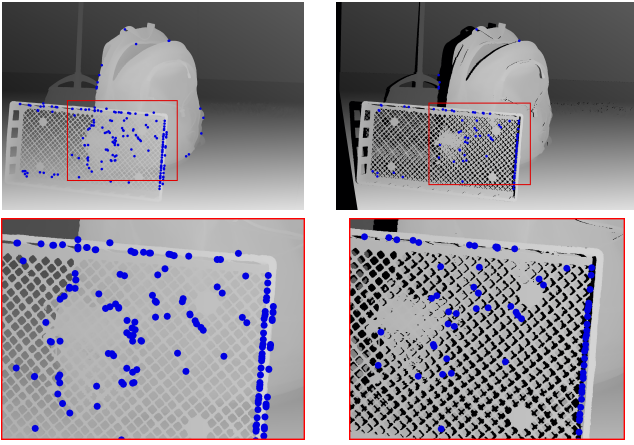


Fig. 5. Illustration of the t -junctions on the original (left) and synthesized (right) *Backpack* image. Bottom: excerpts on the grid part of the image.

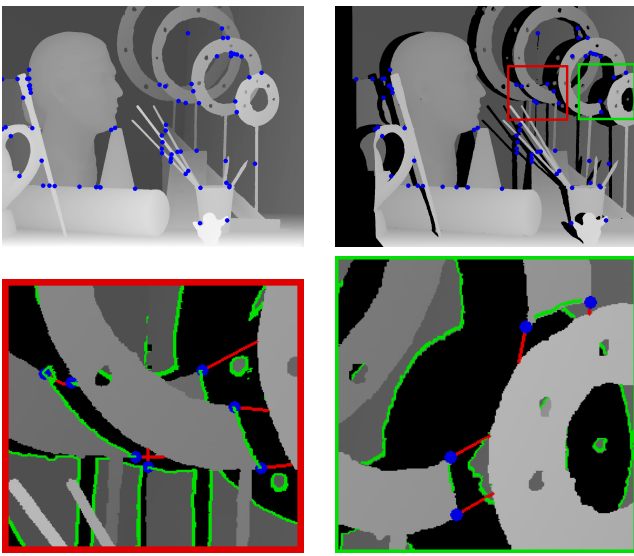


Fig. 6. Top: plots of the t -junctions on the *original* (left) and synthesized (right) depth-maps as blue dots. Bottom: zoomed parts of the synthesized depth-maps with t -junctions depicted as blue dots, extended structures as red lines, and exterior narrow bands NB_s^+ in green.

- the diffused depth is lower than an existing one, i.e., the front reaches a foreground object,
- the front reaches an extended line.

It is important to note here that the diffusion is allowed to overwrite depth values that are lower (w.r.t. λ) than those that are diffused.

Figure 7 (middle) shows the result of the diffusion step on the synthesized depth-map (left). Thanks to the extended lines, the sharp structures of the rings are well recovered, and the result is consistent with respect to the geometry of the objects.

3) *Inpainting remaining lonely regions*: After the two first steps of our algorithm, some holes may not have been inpainted. These holes are not surrounded by pixels of NB_s^+ , and then have not been filled during the diffusion process. Since they are surrounded only by foreground depth values, it is necessary to retrieve *good* background depth values for filling them. As suggested in [2], [3], a solution is to use depth

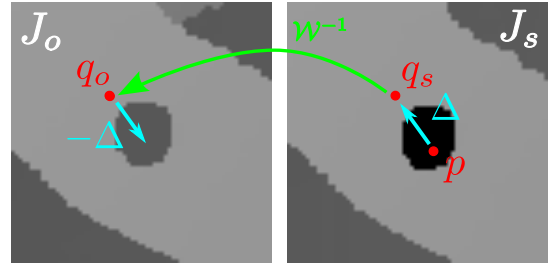


Fig. 9. Illustration of the inpainting of the remaining holes. *Left*: original depth-map J_o , *right*: synthesized depth-map J_s . See text for details.

values from the original depth-map.

For a pixel p to inpaint, this third step (illustrated in Figure 9) mainly consists in:

- getting the nearest known pixel $q_s \in J_s$, and the offset $\Delta = p - q_s$,
- getting its counterpart q_o in the original depth-map J_o by inverse warping $q_o = \mathcal{W}^{-1}(q_s)$,
- retrieving the good depth by taking the opposite offset $-\Delta$ from q_o .

Since these lonely holes are generally small, the adjoining small hole in J_o pointed by the offset $-\Delta$ from $q_o \in J_o$ can be safely modeled by a linear plane, and thus linearly interpolated to get the depth value of each point p of the hole.

Figure 7 (right) shows the effect of this step that effectively inpaints the remaining holes.

C. Results and Comparisons

In this section, we provide visual results of our depth-map inpainting method and comparisons with state-of-the-art approaches [13], [15], [2], [20]. Figure 8 shows and compares depth-map disocclusion inpainting results for images *Adirondack*, *Art*, and *Backpack* from the Middlebury dataset [25]. This dataset provides high resolution pairs of stereo images together with their respective disparity maps.

The sizes of the images *Adirondack*, *Art*, and *Backpack* are 2880×1988 , 1390×1110 , and 2940×2016 respectively, while the number of pixels to inpaint is *464K*, *347K*, and *634K* respectively. We noticed here that our proposed method may actually inpaint more pixels since it allows incorrect background pixels to be overwritten.

Figure 8 shows that our approach clearly gives better disocclusion inpainting results. The structures are recovered in a more natural manner, and no major inconsistencies can be observed contrary to other state-of-the-art methods.

IV. DEPTH-AWARE COLOR IMAGE DISOCCLUSION INPAINTING

In this section, we detail our proposed method for the disocclusion inpainting of the color image I_s . In the following, we consider that the adjoining synthesized depth-map J_s does not contain holes anymore (i.e., it has already been inpainted with the method detailed in Section III-B).



Fig. 7. Inpainting of rings. Left: synthesized depth-map with the holes in black. Middle: result after the line extension and diffusion process. Right: final result after inpainting the remaining holes.

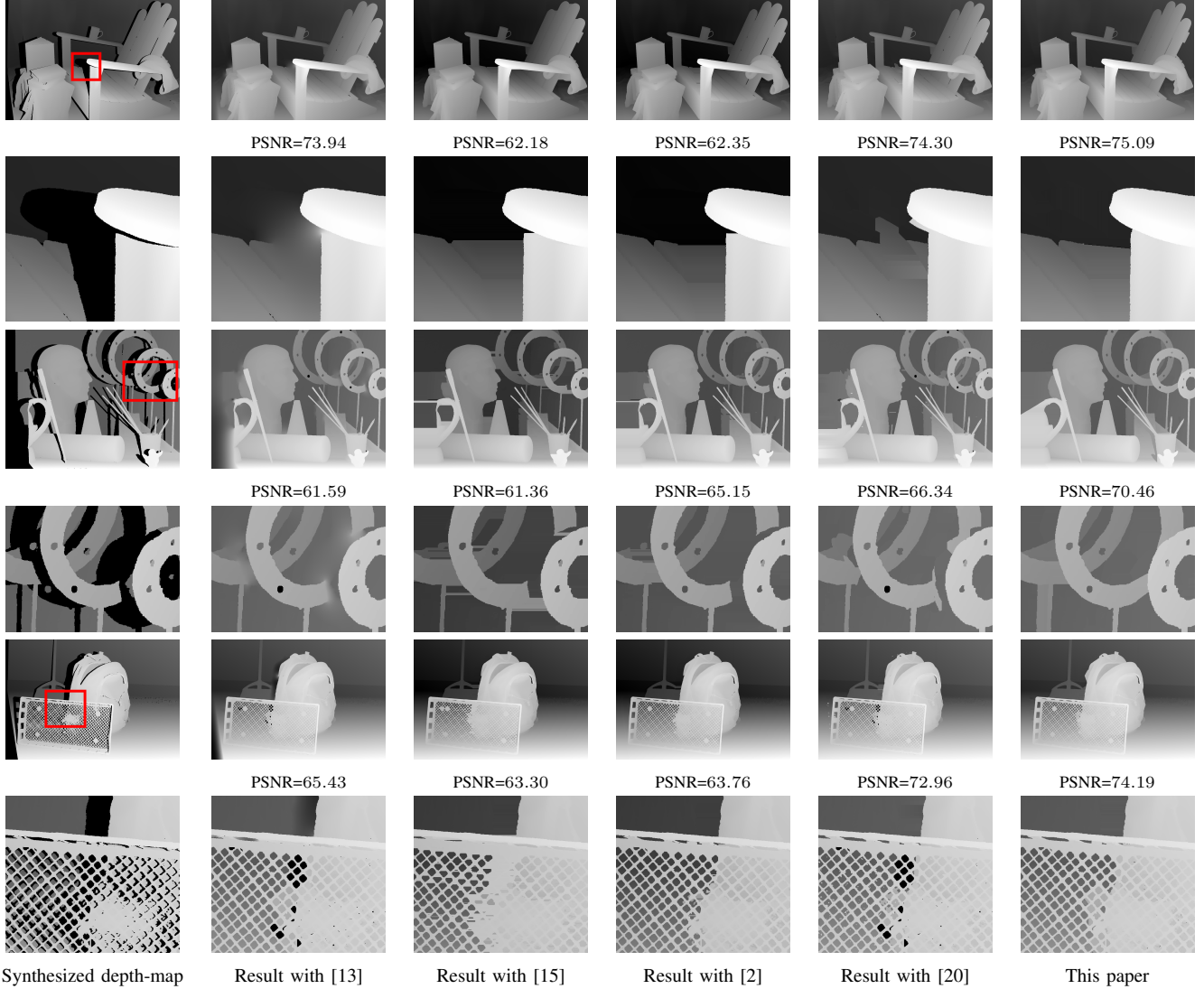


Fig. 8. Depth-map disocclusion inpainting results on the following images: *Adirondack* (rows 1 and 2), *Art* (rows 3 and 4), and *Backpack* (rows 5 and 6). From left to right: synthesized depth-map with holes in black, then results with [13], [15], [2], [20], and our proposed method. The parameters (i.e., the patch size for [2], and the relative superpixel size for [20]) of each method have been manually tuned to get the best results.

A. Patch-based inpainting revisited

Our proposed color image disocclusion inpainting method is also based on the seminal patch-based inpainting algorithm of Criminisi *et al.*[5] (detailed in Section II-B1).

We now revisit each of the 3 main steps of this algorithm (i.e., priority computation, search scheme, and data copy) to

make them use the information from the depth-map.

1) *Depth-aware priority term*: In our previous work [4], we showed that the priority term, and particularly the *data term*, is very sensitive. Modifying it by adding extra terms to take into account the depth, as most of state-of-the-art approaches proceed (Section II-B2), tends to even lower its

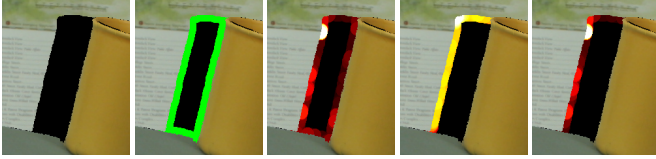


Fig. 10. Illustration of the effect of the proposed *confidence* term. From left to right: hole to inpaint, interior boundaries of the mask (in green), and, with a heat color map, the *data*, *confidence*, and *priority* terms respectively. Since the *confidence* terms for the pixels lying on the border of the cup are equal to 0, their *priorities* are also equal to 0, and the filling process starts from the background.

robustness. Moreover, some state-of-the-art approaches add extra parameters that can be particularly difficult to tune.

In this paper, we propose to avoid adding an extra term to the initial priority term $P(p) = C(p) \times D(p)$. As detailed below, the depth information is not incorporated into the *data* term $D(p)$ but rather in the *confidence* one $C(p)$ in an obvious way.

First, we chose to use the *data* term that we previously proposed in [4] for the purpose of general inpainting. This *data* term, based on structure tensors, showed some merit in comparison to the original one (more details on this can be found in [4]).

Second, we make a simple ascertainment: the confidence term is supposed to count the *reliable* information (known pixels) around a pixel p , such that these known pixels are used for the patch matching (and the effective inpainting). Since it is quite counter-intuitive to use foreground pixels that lie near a background pixel p that we want to inpaint, we then propose to define the *reliable data* around a pixel p as the number of pixels that are at the same depth as p (with respect to the threshold λ):

$$C(p) = \frac{1}{|\mathcal{N}_p|} \sum_{\substack{q \in (\mathcal{N}_p \cap (\mathcal{I} \setminus \Omega)) \\ |J_s(p) - J_s(q)| < \lambda}} C(q) \quad (15)$$

with $C(p) = 1, \forall p \in \bar{\Omega}$.

In order to have this confidence term performing correctly, a slight modification is also done on the boundary computation: we do not consider the *exterior* boundary of the masked part, but the *interior* one. By considering the *interior* boundary of a hole, known pixels belonging to a foreground object lying near a background boundary pixel will not be taken into account for the confidence term computation of the background pixels. Figure 10 shows this advantage and details the full computation of the priority terms for all the pixels of the boundary of a hole. Since the pixels of the boundary that lie near the cup (foreground) belong to the background, their confidence term is equal to 0 (Figure 10, fourth image). Such pixels have a priority equal to 0 and are then inpainted at the very end of the process, which is a desired behavior.

2) *Depth-aware search scheme*: The proposed lookup strategy to retrieve the patch that best matches the known part of the target patch Ψ_t is composed of 3 components:

- Since the synthesized color image I_s is computed from the original color image I_o with the warping function

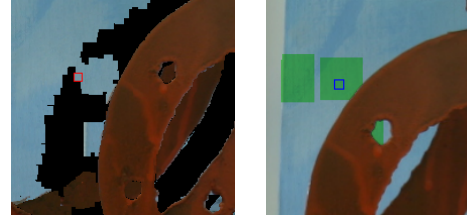


Fig. 11. Illustration of the proposed *depth-aware* search. Left: hole being inpainted (in black), with the target patch depicted in red. Right: search space (in green) in the original view, with the best patch depicted in blue. See text for details.

\mathcal{W} , we can safely claim that most of data of I_s can also be found in I_o ($\text{data}(I_s) \subset \text{data}(I_o)$). From this ascertainment, we constrain the search of the best patch (eq. 1) in I_o only. Thanks to the offset map (i.e., disparity map) used for the initial warping, search areas can easily be retrieved in I_o at the *good* locations by taking the opposite of the offsets.

- Only patches that have the same depth as t (with respect to λ) are considered as candidate patches (step similar to [15], [2]):

$$\Psi_t = \left\{ \Psi_p \in I_o \mid \arg \min_{|J_o(p) - J_s(t)| < \lambda} d_{SSD}(\Psi_t, \Psi_p) \right\} \quad (16)$$

- The search area in I_o is *split* into many sub-windows according to the method proposed in [4]. In short, we propose to use the sources (i.e., locations of the best patches) of previously pasted patches as preferred search areas for the patch that is being inpainted.

Many advantages result from this search scheme: 1) Thanks to the *split* search method, the overall coherence of the reconstructed part is better maintained than with the classical search based on a single window centered at t in I_s . 2) Since patches are searched exclusively in I_o , one does not have to care about incomplete patches (i.e., all patches of I_o are always fully defined). 3) Since candidate patches are constrained to have the same depth, all the patches that are irrelevant for the inpainting of the patch are discarded: foreground patches are automatically discarded when inpainting a background patch. 4) Thanks to the *split* search scheme and to the discard of many irrelevant candidate patches (that are not at the same depth), the search space is greatly reduced which implies computation times shortened. As the retrieval of the best patch is known to be the most time consuming part of patch-based methods, this gain is clearly not negligible.

Figure 11 (right) shows the search area (in green) for the background target patch (red square, left image). The *sub-windows* come from the previously pasted patches, and do not contain the foreground rings. The best patch Ψ_t is depicted in blue (right image).

3) *Depth-aware patch copy*: Once the best patch has been found, masked pixels of Ψ_t are filled with the values of known pixels of Ψ_t (Eq. 5). To our knowledge, all of the literature methods use this paste scheme to fill-in Ψ_t . This works well in practice when dealing with a hole that is only surrounded by the background and a single foreground object, since the

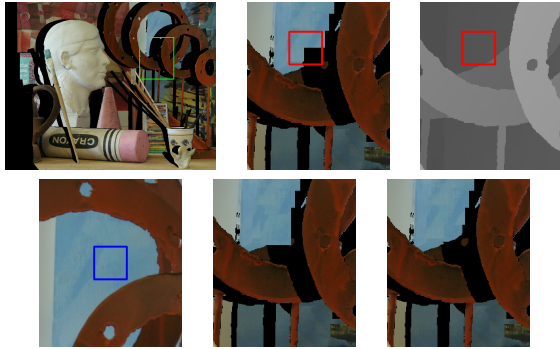


Fig. 12. Illustration of the proposed *depth-aware* copy on an iteration. Top, from left to right: image to inpaint, zoom of the image being inpainted with the target patch depicted in red, adjoining depth-map. Bottom, from left to right: best patch found elsewhere in the original image (depicted in blue), result with the classical copy scheme, and result with our proposed approach. One can notice that no blue pixels are copied into the ring with our approach (bottom right) contrary to the classical copy scheme (bottom middle).

hole has to be filled with background values only. In more complex cases where one has to deal with multiple objects at different depths that overlap, the hole may have to be filled with background and *intermediate* foreground values. This is the case when a foreground object is partially masked by another object with a higher depth. In such a case, the classical copy scheme is clearly not optimal since the underlying depth of the mask is not taken into account.

To properly deal with these not-so-rare cases, we propose a *depth-aware* copy scheme, which copies pixel values from $\Psi_{\hat{t}}$ to unknown pixels of Ψ_t only if their adjoining depths are equal (w.r.t. λ):

$$\Psi_t(q) = \Psi_{\hat{t}}(p) \left| \begin{array}{l} q - t = p - \hat{t}, \forall q \in \mathcal{N}_t \cap \Omega \\ |J_o(p) - J_s(q)| < \lambda \end{array} \right. \quad (17)$$

Figure 12 illustrates the benefits of this *depth-aware* copy scheme where the mask to inpaint contains background values but also foreground values of the masked ring. With the proposed copy strategy, no background pixel values are copied on a masked foreground object (and vice-versa). To our knowledge, none of current state-of-the-art methods propose such a copy feature.

4) *Dealing with object aliasing*: One important problem when processing both color image and depth-map, is that the color of the objects often *bleeds* into other objects or into the background. While the frontier between two objects is sharp and well defined in the depth-map, it can be several pixels wide in the color image (Fig. 13, first row). The color of a pixel of this *narrow band* that belongs to the background (according to its depth) may be composed of a mixture of background and foreground colors. If these phenomena are not properly taken into account during the process, the inpainting algorithm is likely to copy these mixed pixels in pure background/foreground regions (Fig. 13, third row, middle column).

In order to tackle this serious *narrow band* problem, we first propose to compute the *exterior* narrow band of the synthesized depth-map (similar to Eq. 11):

$$NB_s^+ = \mathcal{T}(\delta(J_s) - J_s, \lambda) \quad (18)$$

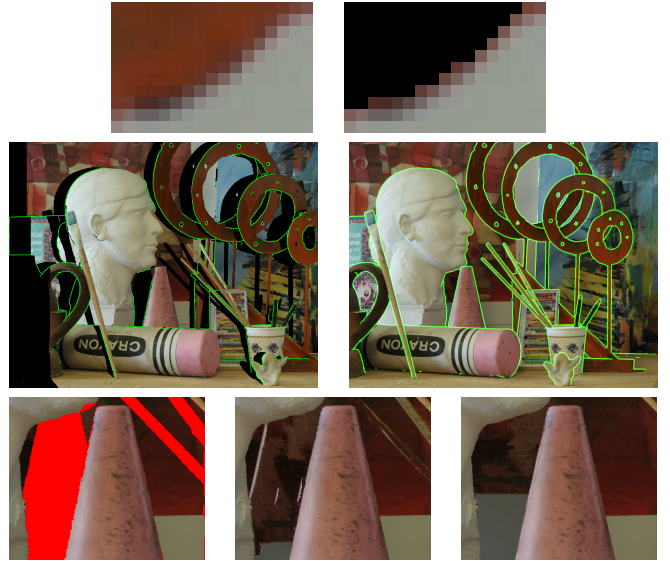


Fig. 13. Illustration of the *narrow-band* problem and the proposed solution. First row: zoom on an image with (left) and without (right) the foreground object (removed w.r.t. its depth). One can notice the remaining color of the foreground object *bleeding* into the background. Second row: superposition of the synthesized (left) and original (right) narrow-bands (depicted in green) on the color images. Third row: a masked part of the *Art* image (left, with the mask in red for visualization purposes), the inpainting result without (middle) and with (right) the proposed *narrow-band* process.

where $\delta(\cdot)$ and $\mathcal{T}(\cdot, \lambda)$ are the dilation and the λ -thresholding operators respectively. The dilation size here essentially depends on the resolution of the image (fixed to 5 in the rest of the paper). Note that NB_s^+ is now different from the one computed for the depth-map inpainting (Eq. 12). Since the depth-map is assumed to be inpainted, it is now fully known.

Figure 13 (second row) displays the superposition of the *narrow-bands* images NB_s^+ and NB_o^+ (in green) on the color images for both the synthesized (left) and original (right) views.

These *narrow-bands* are incorporated within the inpainting process as follows. Given a target pixel t , two different cases appear:

- If $t \notin NB_s^+$, the target pixel t does not lie at the frontier between the background and a foreground object. In such a case, one does not want to copy *bleeding* pixels, so the search of the best patch $\Psi_{\hat{t}}$ is restricted to $\Psi_p \in I_o$ such that Ψ_p contains no pixels of NB_o^+ .
- If $t \in NB_s^+$, the target pixel t lies at the frontier between the background and a foreground object. In such a case, one allows the copy of *bleeding* pixels to eventually copy the smooth transition between the background and the foreground. The search of the best patch $\Psi_{\hat{t}}$ is then processed without restrictions on NB_o^+ .

This simple mechanism avoids (1) the copy of *bleeding* pixels in unwanted areas, while (2) authorizing it at the frontier between the background and the foreground for the best possible reconstruction. Figure 13 (third row, third column) shows the benefits of this scheme: no *bleeding* pixels are copied in unwanted parts of the mask, and the artifacts (third row, second column) are discarded. Note that, since the depth-

maps J_s and J_o are fully known, NB_s^+ and NB_o^+ do not change during the inpainting, and this process is implemented at marginal cost using integral images [27].

B. Evaluation

We evaluate and compare our proposed color image disocclusion inpainting method with respect to two representative approaches of the state-of-the-art [16], [2].

As for the depth-map disocclusion inpainting evaluation, we used the Middlebury stereo dataset [28], [25] which consists of a collection of pairs of stereo images whose sizes vary from 1.4M to 6M pixels. The synthesis is performed from view₁ to view₀ for the 2014 sub-dataset, and from view₅ to view₁ for the rest. We set $\lambda = 4$ for all the experiments.

Figure 14 provides qualitative results obtained with our method together with excerpts of reconstructed images obtained with the methods of [16] and [2]. For fair comparison purposes, we use the same depth-map to guide the inpainting for the three methods: the groundtruth depth-map (available in the dataset) for the *Art*, *Dolls*, *Midd2* and *Moebius* images (rows 2, 4, 5, and 6), and the depth-map inpainted with our dedicated algorithm (Section III) for the *Adirondack* and *Backpack* images (rows 1 and 3). Parameters (i.e., the patch size and the lookup size) have been manually tuned for each method to get the best visual results.

Thanks to the proposed depth-aware patch-based disocclusion inpainting method, our inpainting results show no leaks nor major reconstruction artefacts. Our depth-aware copy scheme avoids inconsistencies between the foreground and the background, even when the images are composed of multiple objects at different depths. Moreover, our robust priority term allows to reconstruct both the structures and the textures with an equal or better quality than with the state-of-the-art methods.

As a quantitative comparison, Figure 16 plots the PSNR obtained with the methods of [16], [2], and ours for many images of the dataset. As we synthesize a view that exists in the dataset, the ground truth can be used for comparison purposes. Note that the same parameters as above (i.e., the patch size and the lookup size) have been manually tuned for each method to get the best result, and these parameters may differ from the one obtained for the figure 14 since a high PSNR does not systematically mean a visually pleasant reconstruction [29].

The last experiment (Figure 15) consists in using our color image disocclusion inpainting method with depth-maps inpainted with [13], [15], [2], [20], and our proposed algorithm (Section III-B). Since our color image disocclusion inpainting method heavily relies on the depth-map (as all the state-of-the-art methods do), a bad depth-map inpainting result automatically implies great inconsistencies in the final result. This experiment underlines the importance of having good depth-map reconstructions.

Finally, it is important to note that our proposed method is quite fast in practice (inpainting at roughly 1500px/s on a single core), and can be easily parallelized: Since the best patches are searched in I_o only (that is fully known),

multiple holes can safely be inpainted in parallel rather than sequentially (as it is actually done in our implementation). Moreover, as the space search is *divided* into several sub-windows, the search for the best patch Ψ_i could also be processed in parallel through these sub-windows to further accelerate the process.

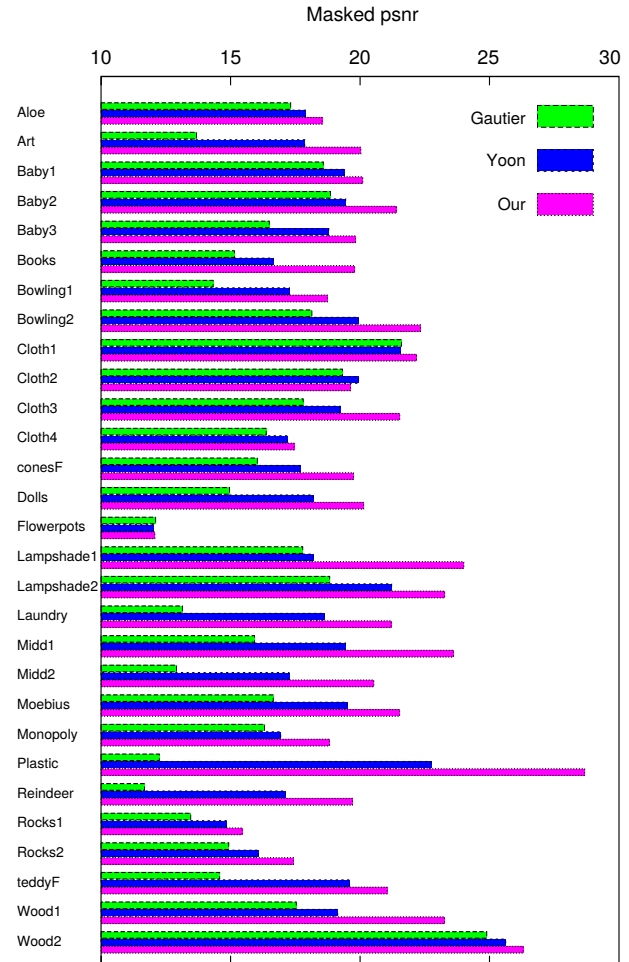


Fig. 16. PSNR of the color inpainted images obtained with the methods of Gautier *et al.* [16], Yoon *et al.* [2], and ours on stereo images of the Middlebury dataset [25].

C. Discussion

Our proposed method involves only one single extra parameter λ compared to the algorithm of Criminisi *et al.*. This λ -thresholding, used throughout our approach, essentially depends on the range of the depth-map, and allows to differentiate two adjacent pixels that belong to different objects. We fixed it manually (by simply looking to depth-maps values) to the same value ($\lambda = 4$) for all the experiments since the images used in this paper have been acquired by the same depth camera. Depending on the depth-map values, the effect of a too large λ is that no differentiation between close objects is possible. On the contrary, a too small value for λ implies that small depth variations inside an object are considered as object contours. This parameter is indeed crucial for the whole

processing, but is fortunately easy-to-tune manually. Further work includes the automatic adjustment of its value.

Note that we use a simple linear extension in our depth-map inpainting algorithm for simplicity purposes (Section III-B1). This step can clearly be improved by using Euler spirals for instance [30] in place of the linear propagation used.

Main remaining parameters are the patch size $|\mathcal{N}_p|$ and the relative size of the space search. This latter parameter directly provides the sizes of the sub-windows, and it is generally admitted in the literature that too big values for this parameter does not provide better results than constraining the search space with smallest search space sizes. Moreover, by applying our proposed lookup search scheme which *divides* the search space into many sub-windows, we indeed constrain the search of the best patch $\Psi_{\hat{i}}$ to more relevant patches in term of final global consistency (see [4] for more details). A typical value for the space search size is 3 to 4 times the value of the patch size.

Adjusting automatically the last major parameter (i.e., the patch size) remains an open question in the inpainting community. It indeed depends on the relative sizes of the objects (structures/textures) that lie around a given hole. While large patch sizes are preferable for macro-textures generation, small patch sizes could be preferable for structures propagation. We previously showed in [4] (Section IV-B) that the multi-windows search scheme allows indeed to copy/paste larger structures and textures with smallest patch sizes. Beyond other advantages of this search scheme (see [4]), its leading advantage is that the sensitivity of the patch size parameter is reduced in comparison with the classical (centered single-windowed) search scheme.

The complexity of the whole algorithm can be detailed as follows:

- Regarding the proposed depth-map disocclusion inpainting algorithm, the complexity is mainly governed by the diffusion step, which is quasi-linear (w.r.t. the number of unknown pixels). The *t-junctions* detection and the small number of structure propagation have indeed a negligible cost.
- Regarding the proposed color image disocclusion inpainting algorithm, it is well-known that the search for the best patch is the main bottleneck of patch-based algorithm. We previously showed in [4] (Section V) that our multi-windows search scheme induces 2 to 5 less patch comparison computations than with the classical (centered single-windowed) search scheme.

Beyond not-well chosen parameters, our method can give unsatisfactory disocclusion inpainting results on difficult images. Figure 17 shows some bad results, with inconsistencies in the texture reconstruction (left column) or with the leftmost hole (right column). The first problem (left column) is due to the lack of texture repeatability: since the *stop sign* does not fully exist in the original image I_o , our patch-based approach cannot find significant patches to copy/paste. The second failure example (right column) highlights the *offscreen* problem: a large portion of the synthesized image was off-screen in the original image, and has then to be reconstructed. This hole (often the largest) is particularly difficult to inpaint. For information, the number of pixels of the leftmost hole, the



Fig. 17. Examples of failure case on the *Jadeplant* (left) and *conesF* (right) images. From top to bottom: images to inpaint with mask depicted in blue, our reconstructed results, and zoomed excerpts of the results. Inconsistencies can be observed on the *stop sign* reconstruction (left), or on the bottom of the red cone (right).

total number of pixels to inpaint, and the size of the images are 428K, 1466K, 2632 × 1988, and 218K, 403K, 1800 × 1500 for the left and right images of figure 17 respectively.

V. CONCLUSION

In this paper we have presented an efficient disocclusion inpainting method for RGB-D view synthesis. First, the disocclusion inpainting of the depth-map is processed with a novel and intuitive approach that is able to recover the lost structures of the objects. Second, many modifications are made to the classical patch-based inpainting algorithm to take into account the previously reconstructed depths for the disocclusion inpainting of color images. These intuitive modifications result in an algorithm that is efficient in terms of quality, robustness and speed. Further work includes the extension of these methods to more general warping schemes, and to the difficult challenge of video processing.

REFERENCES

- [1] C. 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] S. S. Yoon, H. Sohn, and Y. M. Ro, “Inter-view consistent hole filling in view extrapolation for multi-view image generation,” in *Image Processing (ICIP), 2014 21th IEEE International Conference on*, 2014, pp. 2883–2887.
- [3] P. Buysens, M. Daisy, D. Tschumperlé, and O. Lézoray, “Depth-aware patch-based image disocclusion for virtual view synthesis,” in *SIGGRAPH Asia 2015 Technical Briefs*, 2015, p. 2.

- [4] —, “Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions,” *Image Processing, IEEE Transactions on*, vol. 24, no. 6, pp. 1809–1824, 2015.
- [5] A. Criminisi, P. Pérez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *Image Processing, IEEE Transactions on*, vol. 13, no. 9, pp. 1200–1212, 2004.
- [6] Y. Mao, G. Cheung, A. Ortega, and Y. Ji, “Expansion hole filling in depth-image-based rendering using graph-based interpolation,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1859–1863.
- [7] Y. Mao, G. Cheung, and Y. Ji, “Image interpolation for dibr view synthesis using graph fourier transform,” in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*. IEEE, 2014, pp. 1–4.
- [8] O. Le Meur, M. Ebdelli, and C. Guillemot, “Hierarchical super-resolution-based inpainting,” *Image Processing, IEEE Transactions on*, vol. 22, no. 10, pp. 3779–3790, 2013.
- [9] R. Martínez-Noriega, A. Roumy, and G. Blanchard, “Exemplar-based image inpainting: Fast priority and coherent nearest neighbor search,” in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, 2012, pp. 1–6.
- [10] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, “View synthesis techniques for 3d video,” in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2009, pp. 74 430T–74 430T.
- [11] A. Efros, T. K. Leung *et al.*, “Texture synthesis by non-parametric sampling,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999, pp. 1033–1038.
- [12] C. Guillemot and O. Le Meur, “Image inpainting: Overview and recent advances,” *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 127–144, 2014.
- [13] I. Daribo and B. Pesquet-Popescu, “Depth-aided image inpainting for novel view synthesis,” in *IEEE International Workshop on Multimedia Signal Processing*, 2010, pp. 167–170.
- [14] L. Ma, L. Do, and P. de With, “Depth-guided inpainting algorithm for free-viewpoint video,” in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 1721–1724.
- [15] 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,” *Multimedia, IEEE Transactions on*, vol. 13, no. 3, pp. 453–465, 2011.
- [16] J. Gautier, O. Le Meur, and C. Guillemot, “Depth-based image completion for view synthesis,” in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, 2011, pp. 1–4.
- [17] S. Di Zenzo, “A note on the gradient of a multi-image,” *Computer vision, graphics, and image processing*, vol. 33, no. 1, pp. 116–125, 1986.
- [18] X. Xu, L.-M. Po, C.-H. Cheung, L. Feng, K.-H. Ng, and K.-W. Cheung, “Depth-aided exemplar-based hole filling for dibr view synthesis,” in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, 2013, pp. 2840–2843.
- [19] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, “Navier-stokes, fluid dynamics, and image and video inpainting,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. 1–355.
- [20] P. Buysse, M. Daisy, D. Tschumperlé, and O. Lézoray, “Superpixel-based depth map inpainting for RGB-D view synthesis,” in *Image Processing (ICIP), 2015 22th IEEE International Conference on*, Québec City, Canada, Sep. 2015.
- [21] F. Qi, J. Han, P. Wang, G. Shi, and F. Li, “Structure guided fusion for depth map inpainting,” *Pattern Recognition Letters*, vol. 34, no. 1, pp. 70–76, 2013.
- [22] S. Reel, G. Cheung, P. Wong, and L. S. Dooley, “Joint texture-depth pixel inpainting of disocclusion holes in virtual view synthesis,” in *Signal and Information Processing Association Annual Summit and Conference*, 2013, pp. 1–7.
- [23] S. M. Muddala, M. Sjöström, and R. Olsson, “Depth-based inpainting for disocclusion filling,” in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*, 2014, pp. 1–4.
- [24] M. Schmeing and J. Xiaoyi, “Faithful Disocclusion Filling in Depth Image Based Rendering Using Superpixel-Based Inpainting,” *Multimedia, IEEE transactions on*, vol. 17, no. 12, pp. 2160–2173, 2015.
- [25] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, “High-resolution stereo datasets with subpixel-accurate ground truth,” in *Pattern Recognition*. Springer, 2014, pp. 31–42.
- [26] P. Soille, *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.
- [27] F. C. Crow, “Summed-area tables for texture mapping,” *ACM SIG-GRAPH computer graphics*, vol. 18, no. 3, pp. 207–212, 1984.
- [28] D. Scharstein and R. Szeliski, “High-accuracy stereo depth maps using structured light,” in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, 2003, pp. 1–195.
- [29] A. I. Oncu, F. Deger, and J. Y. Hardeberg, “Evaluation of digital inpainting quality in the context of artwork restoration,” in *European Conference on Computer Vision*. Springer, 2012, pp. 561–570.
- [30] F. Cao, Y. Gousseau, S. Masnou, and P. Pérez, “Geometrically guided exemplar-based inpainting,” *SIAM Journal on Imaging Sciences*, vol. 4, no. 4, pp. 1143–1179, 2011.



Fig. 14. Inpainting comparisons. From left to right: masked image, our result, and inpainting result excerpts with [16], with [2], and with our method. For information, the mask sizes are (from top to bottom), 526K pixels for the *Adirondack* image, 401K pixels for the *Art* image, 837K pixels for the *Backpack* image, 263K pixels for the *Dolls* image, 157K pixels for the *Midd2* image, and 253K pixels for the *Moebius* image.

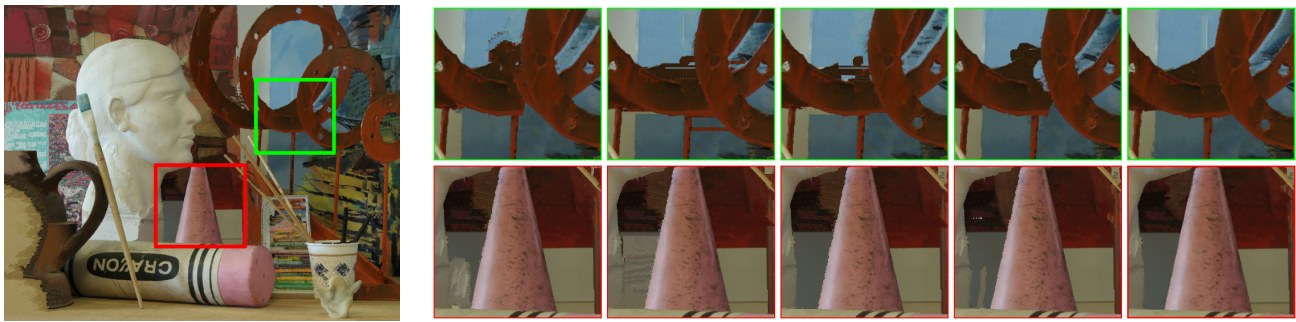


Fig. 15. Inpainting comparisons using the same color image inpainting algorithm (our), but with depth-maps coming from different depth-map inpainting algorithms. From left to right: full result using depth-map inpainted with the proposed method, then inpainting excerpts using depth-map reconstructed results obtained with [13], [15], [2], [20], and the proposed algorithm.