



**HAL**  
open science

## Stereoscopic Image Inpainting using scene geometry

Alexandre Hervieu, Nicolas Papadakis, Aurélie Bugeau, Pau Gargallo I Piracés, Vicent Caselles

► **To cite this version:**

Alexandre Hervieu, Nicolas Papadakis, Aurélie Bugeau, Pau Gargallo I Piracés, Vicent Caselles. Stereoscopic Image Inpainting using scene geometry. IEEE International Conference on Multimedia and Expo (ICME'11), Jul 2011, Barcelona, Spain. pp.1-6, 10.1109/ICME.2011.6011901 . inria-00622591

**HAL Id: inria-00622591**

**<https://inria.hal.science/inria-00622591>**

Submitted on 20 Sep 2011

**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.

# STEREOSCOPIC IMAGE INPAINTING USING SCENE GEOMETRY

A. Hervieux, N. Papadakis, A. Bugeau, P. Gargallo \*

V. Caselles †

Barcelona Media  
Av. Diagonal, 08017 Barcelona  
ahervieu@gmail.com

Universitat Pompeu Fabra  
C/Tànger 122, 08018 Barcelona  
vicent.caselles@upf.edu

## ABSTRACT

In this paper we propose an algorithm for stereoscopic image inpainting, given the inpainting mask in both images. We also assume that depth map is known in one of the images of the stereo pair, taken as reference. This image is clustered in homogeneous color regions using a mean-shift procedure. In each clustered region, depths are fitted by planes and then extended into the mask. Then we inpaint the visible parts of each extended region using a modified exemplar-based inpainting algorithm. Finally, we extend the algorithm to stereoscopic image inpainting. We display some experiments showing the performance of the proposed algorithm.

*Index Terms*— Image Inpainting, Depth-Enhanced Image, Stereoscopic Images

## 1. INTRODUCTION

The recent commercial interest in exhibiting 3D movies or other events like sports or music in theaters has motivated the development of post-production tools in order to assist their acquisition or to eliminate unwanted objects like rigs or cables which may be unavoidable during filming. Image inpainting consists in recovering the missing or corrupted parts of an image so that the reconstructed image looks natural. Our purpose in this paper is to propose an inpainting tool which is able to fill-in the holes in both left and right images with a coherent texture so that the invented parts of both images look like the projection of real 3D objects and the user perceives them as such. In this paper we focus our attention on still images, the video case being necessary to create a flexible tool for the post-production of depth-enhanced imagery.

Image inpainting techniques can be organized into two main categories: geometry and texture oriented methods.

\*A. Hervieux and P. Gargallo acknowledge partial support by IP project "2020 3D Media: Spacial Sound and Vision", financed by EC. A. Hervieux, A. Bugeau and N. Papadakis acknowledge support from the Torres Quevedo program of the Ministerio de Educación y Ciencia in Spain. The authors thank Mediapro for allowing the use of some of their stereoscopic images.

†V. Caselles acknowledges partial support by MICINN project, reference MTM2009-08171, by GRC reference 2009 SGR 773 and by "ICREA Acadèmia" prize for excellence in research funded both by the Generalitat de Catalunya.

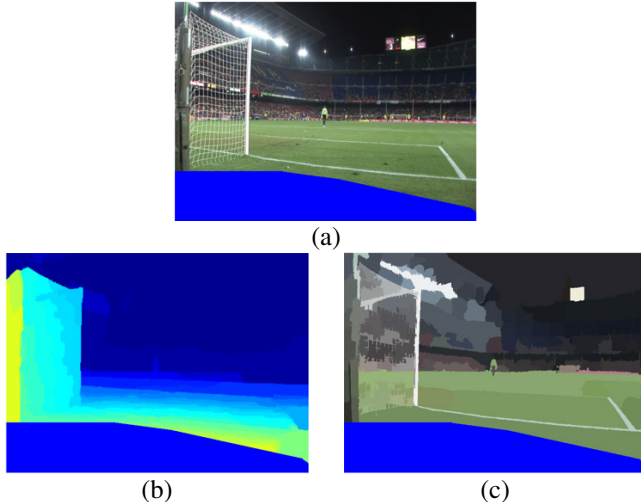
Geometry-based approaches aim at propagating the geometric structure of the image from the boundary towards the interior of the holes. Usually these methods consist in finding the minimum of an energy functional or directly solving a partial differential equation [1, 2, 3]. Its main drawback is that they are not able to restore textures.

In parallel to these geometry-oriented approaches, a category of very powerful methods have appeared as an application of texture synthesis [4, 5, 6, 7]. They can be described as exemplar-based methods (they basically exploit the locality and the stationarity at a certain scale of textures) and were introduced by Efros and Leung in the seminal paper [8], where a simple yet effective non-parametric texture synthesis method based on local image patches was proposed. The unknown region  $\Omega$ , called the hole, is filled-in by copying content from the known part  $\Omega^c$  of the image, the complement of the hole. This method is at the origin of many image inpainting algorithms, one of the most effective ones being [4]. This algorithm permits to obtain interesting results for inpainting of both textured and non-textured images.

The above methods deal with 2D images or video and do not take into account the 3D geometry of the objects they are inpainting.

In the case of stereoscopic image inpainting, one needs to take into account depth information to produce coherent inpaintings [9, 10]. In the method proposed by Wang et al. [9] color and depth maps are filled-in for each image in a separate way. The 3D coherence is obtained using an iterative process, but no convergence is ensured. The method in [10] first produces a 3D coherent filling-in of the depth maps using an energy minimization process and then inpaints the image pair simultaneously using a modified version of [4], that ensures the 3D coherence of the generated texture.

In this paper, we also use the depth information, not only to achieve a coherent inpainting of the left and right images, but to improve the inpainting itself. We first concentrate on the inpainting of a single image whose depth map is known. We will refer to such an image as a *depth-enhanced image*. Our main idea is to use the 3D geometry of the scene, given by the depth map, to determine which objects are visible in the hole. The color texture is then synthesized for each of



**Fig. 1.** (a) The region to be inpainted  $\Omega$  is shown in blue. (b) The depth map  $D$  corresponding to (a), with unknown depth in the same blue region. (c) Mean-shift clustering  $C$  of (a).

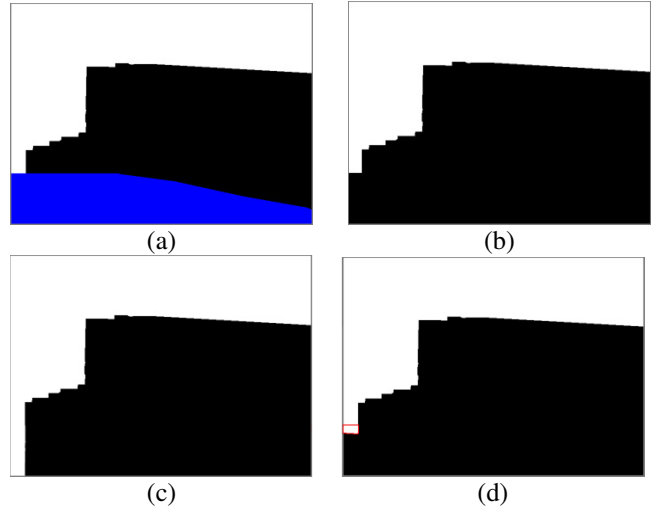
these objects independently as in a texture-by-numbers problem [11]. We can see our approach as a mixture of geometric and texture-based methods, where the geometric step fills-in object labels rather than colors. In addition, we will see that the depth information will greatly help in the geometric reasoning.

## 2. OVERVIEW OF THE ALGORITHM

The input of our algorithm is an image and the corresponding depth map (i.e. a depth-enhanced image). The depth map can be computed from multiple images using a favorite stereo algorithm [12]. The proposed algorithm has three main steps:

- (i) Segment the image into depth layers:
  - (a) Segment the image using color information.
  - (b) Estimate a depth plane for each color segment. Obtain a set of labeled planes.
  - (c) Segment the given depth map  $D$  into depth planes using the model planes computed in (b).
- (ii) Extend the depth layers into the hole using a binary inpainting algorithm. Extract a depth map by computing which of the layers is visible at each pixel (i.e. the closest to the camera).
- (iii) Inpaint the image using the depth-based segmentation. Knowing which layer is visible at each pixel, we use a patch-based texture synthesis method to synthesize the texture corresponding to that layer.

Steps (i), (ii) and (iii) are object of Sections 3, 4, and 5, respectively. Section 6 describes the extension of the algorithm to stereoscopic image pairs.



**Fig. 2.** Illustration of the region extension procedure with  $N' = 2$  adjacent regions. (a) depth plane segmentation of the known part  $\Omega^c$  of the image (b) of Fig. 1. We then have  $S_2$  in black,  $S_1$  in white and  $\Omega$  in blue. (b) Black background  $S_2$  reconstruction. (c) Extension of the depth layer  $S_1$  (the white region) in the hole  $\Omega$  using [3]. (d) Visible part of the extended regions with respect to the two layers plane equations (the red surrounded region is the visible part of the white extended region).

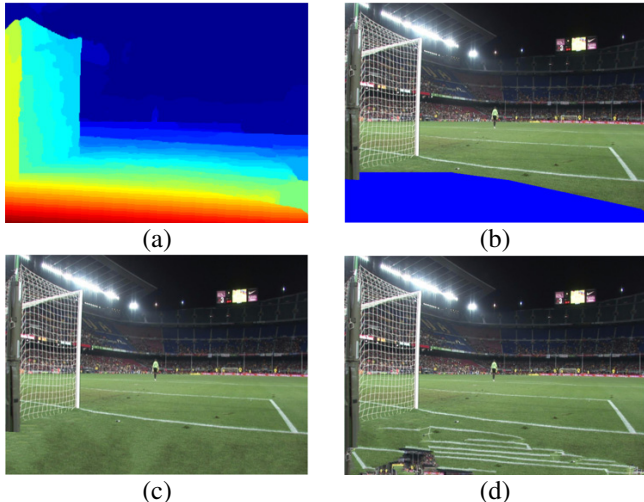
## 3. COLOR AND DEPTH SEGMENTATION

This section presents a way of computing a segmentation of a depth-enhanced image relying both on color and depth information. The idea consists in computing first an over-segmentation using only color information. Then the 3D geometry of each of the color segment, given by the depth map, is approximated by a 3D plane. Finally, in order to obtain regions that correspond as closely as possible to real 3D objects, we compute a second segmentation, this time based on the depths rather than the color.

**Color image segmentation.** The first step of the process consists in computing a segmentation of the image in regions of uniform color. For that we use the mean-shift algorithm [13]. This kernel-based algorithm does not require to fix the number of regions, but it requires to choose the kernel shape and its bandwidth for both spatial and color dimensions. In practice we have chosen a Gaussian kernel with both bandwidths equal to 10. The result of the segmentation is a partition  $C$  of the known part of the image  $\Omega^c$ . In the sequel, we will denote by  $C_i$ ,  $i = 1, \dots, N$ , the  $N$  estimated color clusters adjacent to the hole.

An example is given in Fig. 1. Here, the goal is to remove and fill-in the lower part of image (a), i.e. the hole  $\Omega$  represented by the blue color. The given depth map is shown in (b) and the obtained mean-shift segmentation is shown in (c).

This example will be used in the following sections of the article to explain the algorithm step by step. Indeed, it is a "simple" example, with only  $N = 3$  hole-adjacent regions,



**Fig. 3.** Depth and depth based color inpainting of the soccer image (Fig. 1). (a) The inpainted depth map. (b) Inpainting of the first layer (region surrounded in red Fig. 2.(d)). (c) Final result obtained after filling-in the background region. (d) Result using a 2D exemplar-based algorithm [4].

that shows the processes involved in the proposed method.

**Depth plane estimation.** For each region  $C_i$  adjacent to the hole, we want to approximate its 3D geometry by a plane. For that, we fit a plane (an affine function) to the depths of the pixels in the region. For different reasons, it may happen that no plane fits the given depths with enough accuracy. Indeed, the real shape may not be planar and, even if it is, the given depth map might contain errors. Thus, in order to fit robustly a plane to most of the region, a RANSAC (RANDOM Sample And Consensus) procedure is considered. We use here the version called MLESAC [14]. When using it, we set the noise standard deviation  $\sigma = 0.25$ , the min and max number of iterations as 100 and 1000, respectively. Thanks to this step, the proposed inpainting method is robust to small errors in depth estimation.

The result of this step is a set of  $N$  3D planes  $\Pi_i$ ,  $i = 1, \dots, N$ , each of which approximates the depths of a region  $C_i$ . In the example shown in Fig. 1,  $N = 3$  regions adjacent to the hole are observed (the green, light gray and dark gray ones). For each of these regions, a plane (not shown in the figure) is fitted to the depth map.

**Plane segmentation of the depth data.** As output of the previous step we have the image segmented into a set of regions with uniform color and whose depth map is approximated by a plane. Notice that a single plane may fit different regions since a single planar object may contain different colors. Thus, we need to improve our initial color segmentation using the depth planes.

For that we propose to segment the given depth map into depth planes using a graph-cut approach taking the previously computed depth planes as labels. Thus, for each pixel of

the known part of the image domain  $\Omega^c$ , we choose a plane  $\Pi_i$ . We require that plane to approximate well the original depth of the pixel via a per-pixel energy, and we add a spatial regularization term, in order to avoid sparse segmentations. We solve this multi-label optimization problem using the  $\alpha$ -expansion graph-cut algorithm [15], and we get the final segmentation  $S_i$  of the domain  $\Omega^c$ , with  $i = 1, \dots, N'$ ,  $N' \leq N$ .

In the example shown in Fig. 1, the proposed segmentation by planes of the depth data has fused the two planes representing the light and dark gray regions given by the color segmentation. Indeed, those two regions have a similar plane equation since they correspond to the same object, *i.e.*, the dark post at the left part of the initial image. As there were only  $N = 3$  planes adjacent to the hole, this step has reduced the number of planes to  $N' = 2$ . The final depth segmentation is shown in Fig. 2.(a), as black and white regions.

As a result of the process described in this section, we have now a set of image regions  $S_i$ , whose geometry is approximated by a set of planes  $\Pi_i$ . In next section we describe how to extend these regions into the hole to be inpainted.

#### 4. DEPTH INPAINTING

In this section, we present a scheme to fill-in the depth-map in the hole. Since the depth map is segmented into planar regions, the idea is to extend the shape of these regions, and then determine which region is actually visible at each pixel of the hole.

**Extension of the depth layers.** In order to ensure that each pixel in the hole will be covered by at least one of the depth layers, we start by extending the background layer to cover the entire hole. We define the background layer as the depth layer that is most distant to the camera. By convention, the background will correspond to the depth layer  $S_{N'}$ . In the example of Fig. 2, the plane in black is deeper than the white one over all the blue mask. Hence, background reconstruction is here done simply by filling-in the blue mask with the black plane as shown in Fig. 2.(b).

Next, each of the remaining depth layers  $S_i$ ,  $i = 1, \dots, N' - 1$ , is extended into the hole using an image smoothing algorithm [3]. For each layer, we consider the characteristic function of its corresponding image region. This is a binary function that takes the value 0 outside the region, and 1 inside, and it is unknown inside the hole to be inpainted. We then apply the smoothing algorithm (with gradient and tensor smoothness parameters set to 1 and 5, respectively) to the values of the function inside the hole, while we hold constant the known values outside the hole that serve as boundary condition. The result is a smoothed characteristic function that has to be thresholded in order to obtain a binary segmentation. A threshold value of 0.5 was chosen.

This produces a geometric inpainting of the regions. The result is a set of overlapping regions. Each pixel in the hole is covered by at least one (the background) region.

**Depth reconstruction.** Finally, the depth of each pixel in the hole is easily obtained by computing which region is visible on it. That is, the closest to the camera among the regions that contain the pixel. Indeed, several extended regions may contain the pixel because the depth extension has been done independently for each depth layer.

While simple, this is a key step of the proposed algorithm. By computing the visibility of the regions, we determine where a region ends because it becomes hidden by another one. This kind of geometric reasoning would not be possible if we were not using the depth maps, and is the main reason for the improvements brought by the present technique over the 2D inpainting methods.

In the example, we see that the extension of post layer stops when it hits the ground. In Fig. 2.(c), we see the infinite extension of the post layer produced by the smoothing procedure, while in Fig. 2.(d), we can see the post stopping on the ground. See Fig. 3.(a),(b) and (c) for the corresponding depth and image results. The same propagation stopping effect can be seen in the inpainting of the cone in Fig. 6.

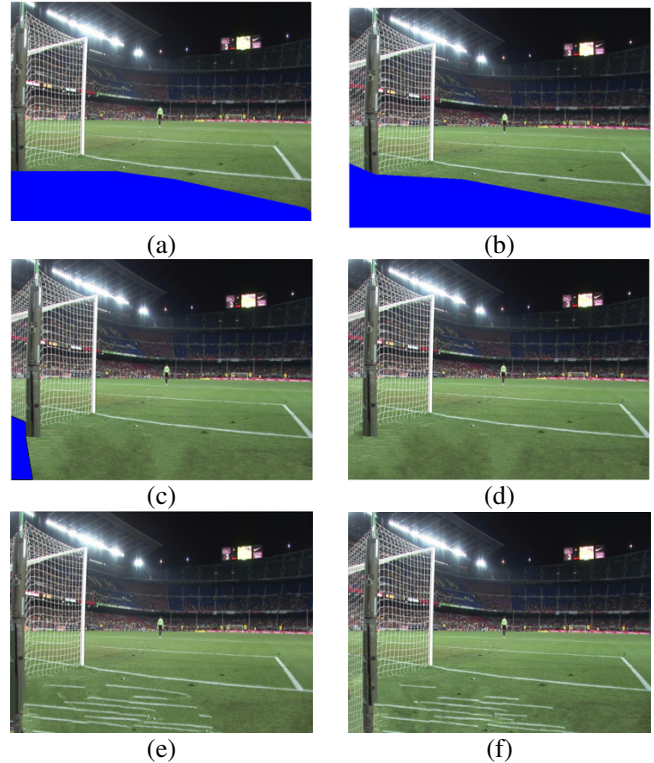
## 5. INPAINTING USING DEPTH LAYERS

The final step of the algorithm consists in inpainting the color image. From the previous steps, for every pixel in the hole, we know its depth and its depth layer. The idea is therefore to give to every pixel the texture corresponding to its depth layer. This is done using the patch-based inpainting of Criminisi et al. [4] for each visible depth layer in the hole. This step is similar in spirit to the texture-by-number problem [11].

The original method, essentially, fills-in the hole by copying patches of the rest of the image inside the hole, taking care that the copied patches fit well into their surrounding. The entire image is used to search for the best patches. Here, since we have a segmentation of the image into depth layers, we can restrict the search for patches to the corresponding depth layers. Therefore, each region of the hole is inpainted by copying patches of the same region outside the hole. In the experiments, we use patches of size  $9 \times 9$ .

In addition to restricting the search for patches to single depth layers, we noticed that reducing the search domain even more improved the results. In effect, a depth layer can be rather large and contain different textures. So one can obtain better results by using only patches that are close to the hole, which are more likely to have the adequate texture. To find the correct region for searching the patches, we compute a mean-shift segmentation of the color image on each depth layer using a Gaussian kernel whose bandwidth (for both space and color dimensions) is 5. This gives a set of subregions of the depth layer. Then, we use only the subregions adjacent to the hole as a source for copying patches.

Fig. 3 illustrates the inpainting process of image in Fig. 1.(a). Fig. 3.(b) is the result of inpainting the depth layer  $S_1$  corresponding to the dark post (*i.e.* the red surrounded re-



**Fig. 4.** Illustration of the stereoscopic inpainting technique. (a), (b) a stereoscopic image pair representing the same scene as in Fig. 1, with the regions to inpaint in blue. (c) Partial filling-in of image (b) obtained by transferring the inpainted image (Fig. 3.(c)) using the inpainted depth map Fig. 3.(a). Unfilled-in region remains in blue. (d) Final inpainting result after filling-in the remaining blue area using the algorithm in [10]. (e), (f) Results obtained using [9].

gion in Fig. 2.(d)). Fig. 3.(c) shows the final result, where the background layer  $S_2$  has also been inpainted with its corresponding texture. The subregion segmentation proposed above is responsible for the fact that only grass texture, without white lines, has been copied into the background layer. Fig. 3.(d) shows the result of the original inpainting algorithm [4] without search restriction.

## 6. STEREOSCOPIC IMAGE INPAINTING

Having extended the depth map into the hole, it is easy to adapt the method to inpaint stereoscopic image pairs.

Let us consider that one of the stereoscopic images, call it  $I_1$ , is already filled-in using the method presented above. Using the inpainted depth map, we can fill-in the corresponding pixels of the hole of the other image  $I_2$ . Indeed, given the depth map and the relative position of the cameras, we know which pixel of  $I_1$  corresponds to which pixel of  $I_2$ . Therefore, we copy the generated texture from  $I_1$  to  $I_2$ . Notice that, due to occlusions, some parts in  $I_2$  may have no corresponding point in  $I_1$  and are not filled-in. To obtain a complete inpainting of image  $I_2$ , these remaining parts are filled-in using the

stereoscopic inpainting algorithm in [10].

An experiment of stereoscopic inpainting is shown in Fig. 4. Figs.4.(a) and (b) are the given stereoscopic image pair. The process described in the previous sections is then applied to Fig.4.(a) to obtain the inpainted depth map (Fig. 3.(a)) and the inpainted image (Fig. 3.(c)). These informations are then transferred to image (b), so that a partial inpainting is obtained (Fig. 4.(c)). The final result including the filling in of the remaining parts is shown in Fig. 4.(d).

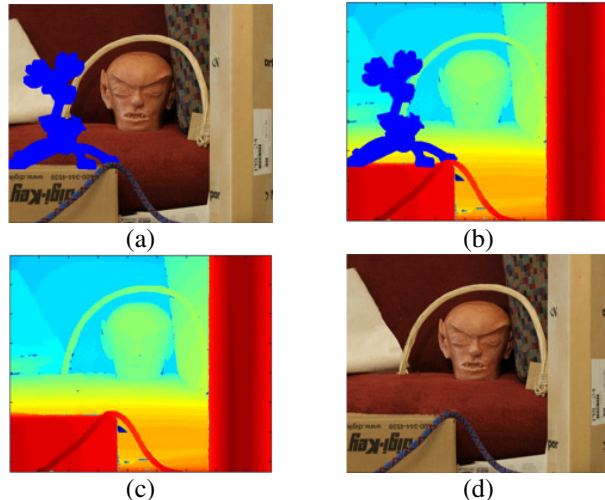
## 7. EXPERIMENTAL RESULTS AND COMPARISONS

**Depth-enhanced image inpainting results.** For the example shown in Fig. 5, the first row shows the initial image with the inpainting mask (in blue) and the corresponding depth map. The second row contains the complete depth map and inpainting results obtained using the proposed method. Another example is shown in Fig. 3, where image (c) shows the result of our method, while image (d) is the result of Criminisi’s method [4]. Another example is shown in Fig. 6.

The comparison with Criminisi’s algorithm [4] highlights the importance of depth related information in inpainting issues. Indeed, our algorithm benefits from depth information by ensuring 3D coherence thanks to background reconstruction and depth-layer extensions. The 3D coherence is illustrated by the filled-in depth map available in Figs. 3, 5. The reconstructed depth maps look coherent with the unmasked depths, and this coherence is propagated into the color images. On the contrary, using only color information may lead to visible artifacts, *i.e.* results that do not fit any perceptually coherent architecture. This can be seen in Fig. 6. Our algorithm manages to properly extend the geometry of the objects surrounding the hole. Small artifacts are still present in the generated textures, but the global architecture is well reconstructed (see Fig. 6.(c)). On the other hand, the 2D algorithm [4] just copies patches that match well with each other, but does not construct any meaningful geometry. Note that the final image generated by our algorithm is a superposition of different textures and generates patches that were not available in the original image.

**Stereoscopic image inpainting results.** Figs. 7.(a) and (b) show the initial stereo pair with the inpainting mask (in blue). Figs. 7.(c) and (d) show the inpainting results obtained by our method. Figs. 7.(e) and (f) show the results obtained using the simultaneous depth-map/stereo image inpainting scheme in [10]. Figs. 7.(g) and (h) show the results obtained using the stereo image inpainting algorithm described in [9]. The visual comparison with the two existing schemes highlights the effectiveness of the novel method.

We notice that our method directly ensures the 3D coherence between the stereoscopic images, in opposition to [9]. In this method, 3D coherence is imposed by an iterative process that compares the results obtained on both sides. Results are considered satisfactory if the inpainted disparity and color are close enough [9]. Hence, 3D artifacts may be created: the in-



**Fig. 5.** Depth-enhanced image inpainting illustration. (a) initial image with inpainting mask (in blue). (b) Depth map. (c) inpainted depth map depth map obtained with the proposed method. The small blue region near the box corresponds to points where depth was not computed. (d) inpainting result obtained using our algorithm.

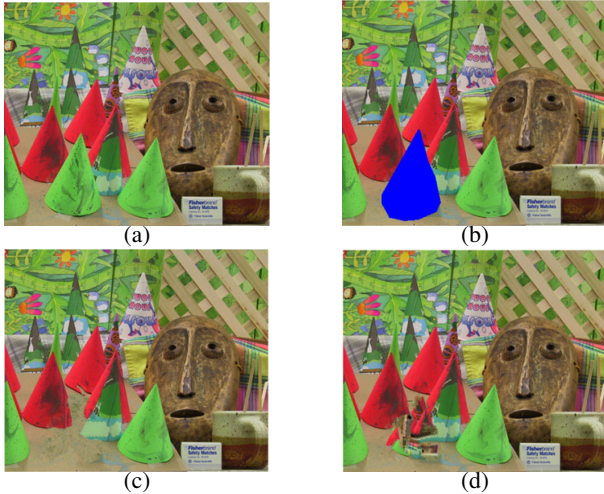
painted parts of the two images need not match, regarding disparity maps. This is observed in the two experiments shown here. For example, in Fig. 4, the algorithm creates white lines that are not 3D consistent between the two images while our method ensures perfect correspondence of the inpainted parts of the two images according to the disparity maps.

By comparing with the method proposed in [10] which also produces 3D coherent results, we see the advantages of taking into account 3D reconstruction using depth layer extensions. Indeed, the method in [10] is a patch-based inpainting algorithm extended to 3D, taking benefit of the depth information by choosing patches beyond or at the same depth than the center pixel of the patch to be filled-in, thanks to an already inpainted depth map (that satisfies the visibility constraints between both stereoscopic images) obtained using a total variation regularization. In our paper, by realizing a geometrical extension of the layers adjacent to the inpainting mask, we allow recovering a better depth inpainting.

Finally, we note that we look for patches belonging to the same 3D planes, whereas in [10] the authors search for patches at same depth with respect to the camera distance. The results visually show the importance of 3D reconstruction (see, for instance, Fig. 7)

## 8. CONCLUSIONS

This paper proposes a novel stereoscopic image inpainting algorithm based on a three-step algorithm for inpainting depth-enhanced images. Our inpainting algorithm for depth enhanced images extends the geometry of the objects one by one into the hole and uses their depth to determine which object should be visible. This leads to natural extension of the objects’ geometry into the hole, while the final texture syn-

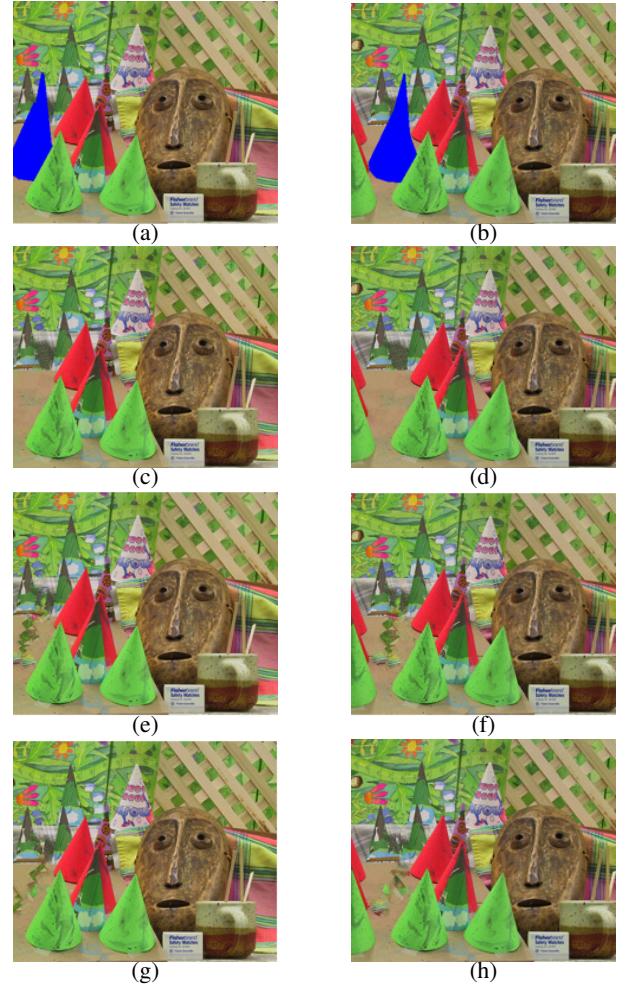


**Fig. 6.** Depth-enhanced image inpainting illustration. (a) initial image. (b) the inpainting mask (in blue). (c) the inpainting result obtained using the proposed algorithm. (d) inpainting result obtained using [4].

thesis step generates a plausible texture for each object. Our experiments show that, by taking into account the 3D geometry of the scene, we are able to produce much more plausible inpainting results than state-of-the-art 2D methods. Our experiments on stereoscopic image inpainting compare favorably to the two other existing methods for the same problem.

## 9. REFERENCES

- [1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *ACM SIGGRAPH*, 2000, pp. 417–424.
- [2] S. Masnou and J.M. Morel, “Level lines based disocclusion,” in *IEEE ICIP*. IEEE, 2002, pp. 259–263.
- [3] D. Tschumperlé, “Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE’s,” *Int. J. Comp. Vis.*, vol. 68, no. 1, pp. 65–82, 2006.
- [4] A. Criminisi, P. Pérez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Trans. on Im. Proc.*, vol. 13, no. 9, pp. 1200–1212, 2004.
- [5] N. Komodakis and G. Tziritas, “Image completion using efficient belief propagation via priority scheduling and dynamic pruning,” *IEEE Trans. on Image Processing*, vol. 16, no. 11, pp. 2649–2661, 2007.
- [6] Y. Wexler, E. Shechtman, and M. Irani, “Space-time completion of video,” *IEEE Trans. on PAMI*, pp. 463–476, 2007.
- [7] Y. Yang, Y. Zhu, and Q. Peng, “Image completion using structural priority belief propagation,” in *ACM Int. Conf. on Multimedia*, 2009, pp. 717–720.
- [8] A.A. Efros and T.K. Leung, “Texture synthesis by non-parametric sampling,” in *IEEE ICCV*, 2002, vol. 2, pp. 1033–1038.
- [9] L. Wang, H. Jin, R. Yang, and M. Gong, “Stereoscopic inpainting: Joint color and depth completion from stereo images,” in *CVPR*, 2008, pp. 1–8.



**Fig. 7.** Stereoscopic image inpainting illustrations. (a), (b) a stereoscopic image pair with the inpainting mask (in blue). (c) and (d) the inpainting results obtained using our method. (e) and (f) results obtained using [10]. (g) and (h) results obtained using [9].

- [10] A. Hervieu, N. Papadakis, A. Bugeau, P. Gargallo, and V. Caselles, “Stereoscopic image inpainting: distinct depth maps and images inpainting,” in *ICPR*, 2010.
- [11] A. Hertzmann, C.E. Jacobs, N. Oliver, B. Curless, and D.H. Salesin, “Image analogies,” in *ACM SIGGRAPH*. Citeseer, 2001, pp. 327–340.
- [12] V. Kolmogorov and R. Zabih, “Multi-camera scene reconstruction via graph cuts,” *ECCV*, pp. 8–40, 2002.
- [13] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. on PAMI*, vol. 24, no. 5, pp. 603–619, 2002.
- [14] P.H.S. Torr and A. Zisserman, “MLESC: A new robust estimator with application to estimating image geometry,” *Comp. Vis. and Im. Unders.*, vol. 78, no. 1, pp. 138–156, 2000.
- [15] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Trans. on PAMI*, vol. 26, no. 9, pp. 1124–1137, 2004.