# Radiometric confidence criterion for patch-based inpainting

Julien Fayer, Géraldine Morin, Simone Gasparini, Maxime Daisy, Benjamin Coudrin

# Radiometric confidence criterion for patch-based inpainting

Julien Fayer*†, Géraldine Morin*, Simone Gasparini*, Maxime Daisy†, Benjamin Coudrin†
*University of Toulouse, Toulouse INP - IRIT, Email: {name.surname}@irit.fr
†SAS InnerSense, Email: {name.surname}@innersense.fr

***Abstract*—Diminished Reality (DR) consists in virtually removing objects from a captured scene, thus requiring a coherent filling of the areas originally hidden behind these objects. Indoor DR applications often exploit the planar geometry of the scene to apply an inpainting process on a perspectively undistorted view of the plane. In this paper we propose to integrate a novel, physic-based criterion into classical state-of-the-art inpainting algorithms in order to take into account the variations in image resolution of the undistorted view. The proposed inpainting process selects the patches and avoids the propagation of low-resolution data, *i.e.* patches corresponding to parts of the plane that are far from the camera or seen under an very skew angle. We illustrate the improvements of DR results on synthetic and real images.**

## I. Introduction

Contrary to Augmented Reality, Diminished Reality (DR) [1] aims at virtually removing one or more objects from the image of a scene. The image regions containing the object must then be filled seamlessly with a texture coherent with the surrounding regions in order to provide a diminished, photo-realistic view. To this end, completion methods such as inpainting [2] can be used to fill the region hidden by the object according to the photometric data of their surrounding regions. The results of inpainting approaches fit well for the expectations of Diminished Reality. In order to improve inpainting results in the context of DR, previous works [3], [4] exploit the indoor setting: as the background consists of planes (floor, walls, ceiling), instead of applying inpainting techniques directly on the image, they take advantage of the planar geometry of the scene to better guide the inpainting process. For each plane, they compute the homography between the camera plane and the 3D plane. Perspective rectification is applied to the plane region to allow inpainting on distortion-free images. In the undistorted plane, the texture structure recovers its regularity in size and pattern, which is a great advantage for inpainting. However, inpainting methods do not take into account the quality variation in terms of image resolution of these rectified images: this can lead to degraded results, as discussed in Section III. We propose an approach which first computes a trust criterion on the rectified image and uses it to improves the inpainting process in the rectified image.

The paper is organized as follows: §II reviews the state-of-the-art inpainting techniques, §III highlights the issues in inpainted images due to resolution variation and §IV introduces the proposed method to guide and improve the inpainting process. §V shows how the proposed criterion can be integrated in two classical inpainting algorithms, and the corresponding results are given in §VI. Finally, §VII concludes the paper, proposes perspective and future extensions of our approach.

## II. Related works

Inpainting finds its natural application in image restoration to remove, *e.g.*, scratches and cracks from the frames of old movies or photos, or in special effects to remove entire objects from an image. There are two main approaches for inpainting techniques [5]. **Diffusion-based methods** [6] are used in image restoration to fill or correct small regions of the images for which a mask is provided by the user. These methods are generally based on Partial Differential Equations (PDEs) and a diffusion model that iteratively propagate the information from the outside of the mask along the isophotes, *i.e.* the level lines perpendicular to the gradient vectors of the pixels on the contour. These methods perform well when filling small and smooth regions but are not adapted if a structure or texture needs to be propagated. Moreover, these methods, being iterative, have a high computational cost.

**Patch-based methods** [7], [8] are instead used to fill larger portions of the image by copying either single pixels (*sparsity-based* [9]), entire patches or a mixture of those from other parts of the image (*exemplar-based* [7]). For each pixel $p$ of the mask, they search the most similar patch in the image to the one centered in $p$, and they copy it. The search for this similar patch is the most important but also the most expensive step of the algorithm. Many variants and optimizations have been proposed over the last decade. One of the most effective approaches is *PatchMatch* [10], which efficiently finds for every patch the approximate nearest-neighbour in the image using a randomized cooperative hill climbing strategy. In [11] the search is restricted to the most likely offsets, reducing both the complexity and also enhancing the propagation of the geometric structures of the image. The other critical step in patch-based methods is the selection of $p$ and the order of filling. *Onion-peel* order fills the missing data starting from the pixels on the border and proceeding layer after layer towards the region's center. This sometimes leads to unexpected results at the center of the region and, in general, structures are not propagated inside the region. *Data-aware* methods, instead, give priority to pixels lying on borders of objects, thus favouring the preservation of structures. On the other hand,

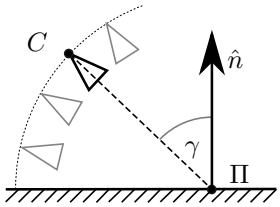


Fig. 1. Camera positions w.r.t. the plane $\Pi$ according to the angle $\gamma$.

a known issue of the *PatchMatch* approach is that it cannot handle well regular textures, *i.e.*, textures embedding regular patterns or structures. Other methods have been proposed to handle regular textures by performing a **statistical analysis** of the texture that allows to find the map of the dominant directions (or *offsets*) [12], [13]: the inpainting problem is then cast as a global minimization of an energy function written in terms of the offset map that enforces the structure and texture consistency. In [14], the minimization problem is solved via **graph cuts** in order to reduce the computational complexity.

The latest works, however, uses information such as the camera position and the 3D structure of the scene to improve the final rendering. In the context of DR applications [15], [3], [16], the objects to be removed usually appear lying on a plane or occluding other planes in the background (eg. floor or walls). In this case, the inpainting is not applied directly to the image. First the perspective distortion is removed and the original structure of the texture is recovered by computing the rectifying homography. An inpainting algorithm is then applied to the undistorted view, and the completed image is re-projected into the original image through the inverse homography. However, if the area to be filled belongs to a plane having a skew angle w.r.t. camera optical axis (*e.g.*, the floor of a room), the photometric information of the undistorted image has usually a large variation of resolution. Applying inpainting independently of this resolution variation generates a blur in the reconstructed data, including in the final rendering after reprojection as explained in the next section. Eisenacher *et al.* [17] proposed a user-driven texture synthesis algorithm: the user can describe the local geometry supporting the texture which can be then re-synthesized it on new surfaces. Although it is not strictly speaking an inpainting method, this is, to the best of our knowledge, the only work that, like ours, tries to took into account the different resolution of the image according to the 3D scene geometry.

## III. Problem Statement

In our indoor DR application, we apply inpainting techniques to the perspectively undistorted view of a plane. The undistorted view has a non-uniform quality of data and possible interpolation artefacts, which affects the final inpainting result. We designed a simple experiment to illustrate such effect. Consider a textured plane $\Pi$ seen by a camera $C$ under an angle $\gamma$ w.r.t. the plane normal $\hat{n}$ (see Fig. 1): we rendered two synthetic images, each with a different value of $\gamma$ ($60°$ and $85°$, respectively). Fig. 2 shows the two images of the textured plane $\Pi$ (column (a)), along with the outline of the region to inpaint, and the undistorted views of $\Pi$ (col. (b)). Once *PatchMatch* is applied to the undistorted view (col. (c)) and the result is projected back on the original image (col. (d)), we can clearly see that low-resolution data corresponding to the most perspectively distorted part of the image has been used to fill the mask region: the result is visually unsatisfying with a noticeable blur effect in the inpainted region. This simple example shows the influence of the perspective distortion on the result of a classic inpainting technique. In this work we propose to introduce a novel criterion in order to better guide the completion process and thus avoid the propagation of low-resolution data.

## IV. Radiometric Confidence Criterion

We first seek a criterion that associates to each image pixel a score characterizing the quality of its projection when it is mapped by the rectifying homography. In particular, our score relies on Bouguer's law, which measures the light flux $\mathrm{d}F$ of a light point source $I$ over a surface $\mathrm{d}S$ seen under a solid angle $\mathrm{d}\Omega$ (see Fig. 3):

$$\mathrm{d}F = I\,\mathrm{d}\Omega = I\frac{\cos\theta}{r^2}\,\mathrm{d}S. \tag{1}$$

Thus the light flux $\mathrm{d}F$ over the surface unit $\mathrm{d}S$ is

$$\frac{\mathrm{d}F}{\mathrm{d}S} = I\frac{\cos\theta}{r^2}. \tag{2}$$

We deduce, for each point $P$ of the image plane, a $\mathrm{trust}$ score as the ratio between the point light flux $\mathrm{d}F$ over an infinitesimal area around P on the image plane and the flux around its projected point $Q$ on the plane $\Pi$. Using (2), we compute at the points $P$ and $Q$ :

$$\left(\frac{\mathrm{d}F}{\mathrm{d}S}\right)(P) = I\frac{\cos\phi}{r_P^2} = I\frac{\cos^3\phi}{f^2} \tag{3}$$

and

$$\left(\frac{\mathrm{d}F}{\mathrm{d}S}\right)(Q) = I\frac{\cos\theta}{r_Q^2} = I\frac{\cos^3\theta}{d^2}, \tag{4}$$

where $f$ is the camera focal length, and $d$ is the distance between $C$ and $\Pi$. Note that $r_P = \frac{f}{\cos\phi}$ and $r_Q = \frac{d}{\cos\theta}$. We define our criterion as the ratio between (3) and (4)

$$\mathrm{trust}(P) = \frac{\left(\frac{\mathrm{d}F}{\mathrm{d}S}\right)(Q)}{\left(\frac{\mathrm{d}F}{\mathrm{d}S}\right)(P)} = \left(\frac{f}{d}\right)^2\left(\frac{\cos\theta}{\cos\phi}\right)^3. \tag{5}$$

whose value decreases as the distance $d$ increases and for increasing angles $\theta$. Higher values of $\mathrm{trust}$ are pixels of good quality, whereas lower values denote points that are likely to represent regions of the plane far from the camera and/or seen under a very skew angle.

We consider $P$ as the center of the pixel $p$ of the image and we define the confidence map $\mathcal{C}$ for each pixel of the rectified image domain $U \times V \subset \mathbb{N}^2$:

$$\mathcal{C} : \begin{cases} U \times V & \longrightarrow \mathbb{R} \\ p & \longmapsto \mathcal{C}(p) = \mathrm{trust}(P) \end{cases} \tag{6}$$

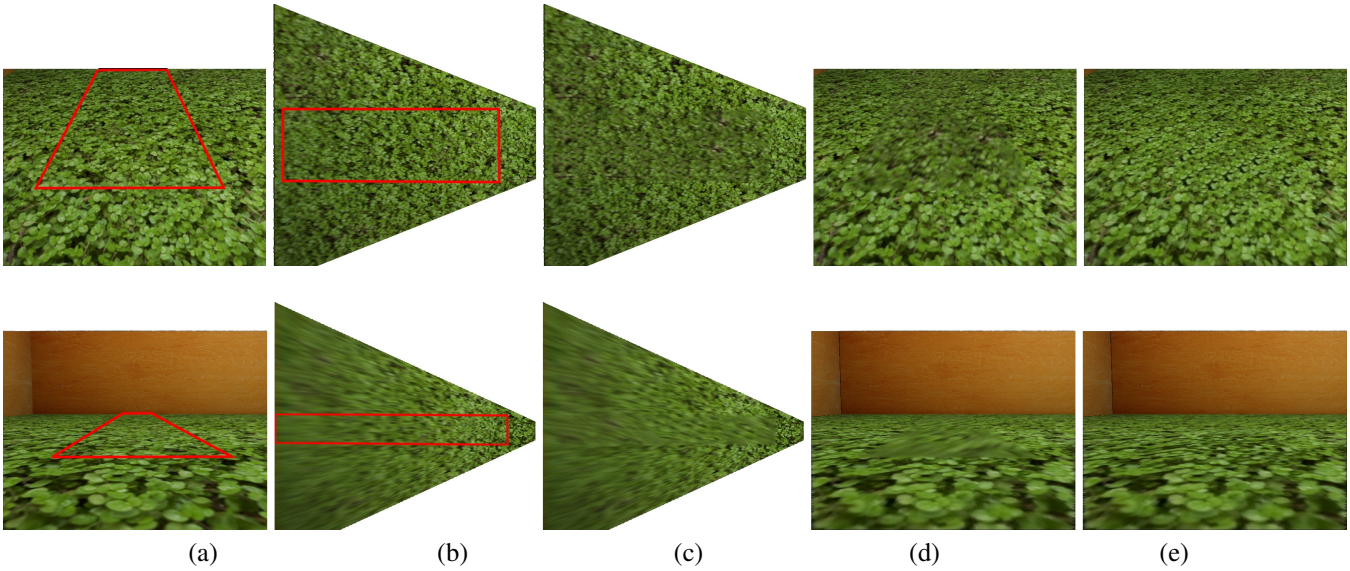


Fig. 2. Perspective influence on inpainting processing on two synthetic images rendered with $\gamma = 60°$ (first row) and $\gamma = 85°$ (second row). From right to left: (a) input image with the outline of the zone that will be removed and completed using inpainting, (b) rectified image of the floor, (c) rectified image completed using classical inpainting [10], (d) remapped output, (e) our proposed solution.

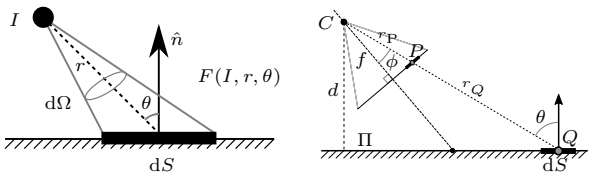


Fig. 3. Left: Bouguer's law for a light source of intensity $I$ on a surface $\mathrm{d}S$. Right: the point $Q$ is the projection of the point $P$ of the image plane on the plane $\Pi$ and its surrounding surface $\mathrm{d}S$

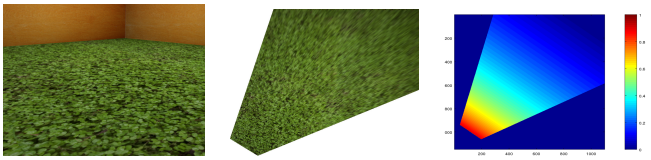

Fig. 4. Left: input image of a floor plane (green texture). Center: rectified image of the plane. Right: confidence map $\mathcal{C}$ for the rectified image, the Jet colormap (high values in red and low values in blue) shows that the value of $\mathrm{trust}$ decreases for the pixels that have a lower quality, *i.e.* those on the top right part of the image.

with $\mathcal{C}(p) = 0$ if $p$ is not a projection of a point of the plane $\Pi$. Fig. 4 shows an example of the map computed for a synthetic image. In the next section we show how we integrate our criterion into two classic inpainting algorithms. Note that our criterion is not restricted to these two methods, but it can be easily integrated for filtering (as shown in §VI) or in other inpainting approaches.

## V. Resolution-aware inpainting

We integrate the proposed criterion in two state-of-the-art inpainting methods, *PatchMatch* and statistic-Graphcut. First, let us define a predicate we use in both approaches to handle the selection of a new candidate through the map $\mathcal{C}$ and avoid

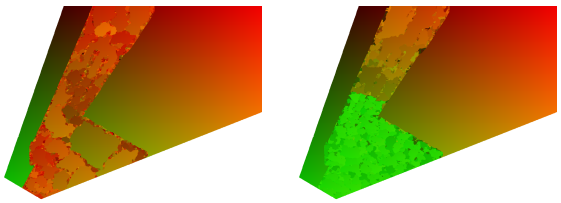

Fig. 5. The effects of our modifications of *PatchMatch*. In a rectified image $I$ like the one in Fig. 4, we used a colormap to identify each pixel position and we show inside the mask the final NNF map, *i.e.* each color inside the mask indicate from which zone of $I$ the pixel has been taken. On the left, the original *PatchMatch* NNF contains many reddish pixels coming from zones of $I$ with "lower quality data". On the right, our computed NNF uses pixels from "higher data quality" regions of $I$ (greenish pixels are closer to the camera).

the selection of "low resolution" data. The predicate returns $\mathrm{false}$ (reject new candidate) or $\mathrm{true}$ (accept) and it is defined as:

$$\mathrm{validation}(p, q, \mathcal{C}, \alpha) = \mathcal{C}(q) \leq \alpha\, \mathcal{C}(p), \tag{7}$$

where $p$ is the pixel to complete, $q$ is the candidate proposed by the approach, and $\alpha \in ]0, 1]$ is a tolerance parameter.

### A. Adapted PatchMatch approach

*PatchMatch* [10] efficiently finds matches between patches of an image as detailed in Alg. 1. It defines a nearest neighbour field (NNF) as a function of offsets $map : \mathbb{R}^2 \to \mathbb{R}^2$ that binds any patch $a$ of an image $A$ to a patch $b$ of an image $B$ according to a distance $D$ (typically, the Sum of Squared Differences (SSD)). The NNF is initialized randomly and updated with a multi-resolution pyramid. Two iterative updates are performed: (a) the offsets with best scores are propagated to adjacent pixels (1.12-15), and (b) a random search in the neighbourhood of all offsets with best distance scores (1.16).

**Algorithm 1** *PatchMatch* algorithm with trust criterion

**Require:** Rectified image $I$, rectified binary mask $M$, trust map $\mathcal{C}$
**Ensure:** Rectified inpainted image $I_{\text{inpainted}}$
**procedure** PATCHMATCH($I, M, \mathcal{C}$)
  Apply diffusion-inpainting in $I_{|M}$ ▷ $I_{|M}$ is the restriction of $I$ to $M$
  Create multi-resolution pyramid for $I$ and $M$ with $N$ floors
  **for** $i \leftarrow 1 : N$ **do**
    **if** $i = 1$ **then**
      $map \leftarrow$ RAND( ) ▷ First floor, initialisation of NNF map
    **else**
      $map \leftarrow$ RESIZE($map, 2$) ▷ gets the NNF of previous floor
    **end if**
    **for** $j \leftarrow 1 : n_{\text{iter}}$ **do**
      **for all** $p \in I^i_{|M}$ **do** ▷ $I^i_{|M}$ is the image of $i$th floor restricted to $M$
        NEIGHBORUPDATE($p, -1, 0$) ▷ Propagation steps
        NEIGHBORUPDATE($p, 1, 0$)
        NEIGHBORUPDATE($p, 0, -1$)
        NEIGHBORUPDATE($p, 0, 1$)
        RANDOMUPDATE($p$) ▷ Randomization step
      **end for**
    **end for**
    APPLYNNF($I_{|M}$)
  **end for**
**end procedure**

**function** NEIGHBORUPDATE($p, dx, dy$)
  $\delta \leftarrow (dx, dy)$
  $q \leftarrow map(p + \delta)$
  **if** SSD($p, q$) $\leq$ SSD($p, map(p)$) **AND** validation($p, q, \mathcal{C}, 1$) **then**
    $map(p) \leftarrow map(p) - \delta$
  **end if**
**end function**

**function** RANDOMUPDATE($p$)
  $\delta \leftarrow$ (RAND( ), RAND( ))
  $q \leftarrow map(p) + \delta$
  **if** SSD($p, q$) $\leq$ SSD($p, map(p)$) **AND** validation($p, q, \mathcal{C}, 1$) **then**
    $map(p) \leftarrow q$
  **end if**
**end function**

**Algorithm 2** Statistic+Graphcut algorithm with trust criterion

**Require:** Rectified image $I$, rectified binary mask $M$, trust map $\mathcal{C}$
**Ensure:** Rectified inpainted image $I_{\text{inpainted}}$
**procedure** STATISTIC+GRAPHCUT($I, M, \mathcal{C}$)
  $I_r \leftarrow$ DOWNSAMPLE($I$)
  $M_r \leftarrow$ DOWNSAMPLE($M$)
  $O \leftarrow$ COMPUTEOFFSETS($I_r, M_r$) ▷ Statistic step
  $labels \leftarrow$ COMPUTELABELS($I_r, M_r, O, \mathcal{C}$)
  $labels \leftarrow$ UPSAMPLE($labels$)
  $I_{\text{inpainted}} \leftarrow$ APPLYLABELS($labels$)
**end procedure**

**function** COMPUTELABELS($I, M, O, \mathcal{C}$)
  $I_e \leftarrow I_{|M \cup \partial M}$ ▷ $\partial M$ is the boundary of $M$
  **for all** $p : (x, y) \in I_e$ **do**
    **if** $p \in \partial M$ **then**
      $labels(p) \leftarrow 0$ ▷ Match adjacent pixels to null vector
    **else**
      $labels(p) \leftarrow$ RAND($1, |O|$) ▷ Random initialisation
    **end if**
    $labels \leftarrow$ GRAPHCUT($I_e, O, \mathcal{C}, E_{\text{data}}, E_{\text{reg}}$)
  **end for**
  **return** labels
**end function**

**function** $E_{\text{DATA}}(p, t, I, M, \mathcal{C})$
  **if** $p \in \partial M$ **OR** $((p + t) \notin M$ **AND** validation$(p, p + t, \mathcal{C}, \alpha))$ **then**
    **return** 0
  **end if**
  **return** $+\infty$
**end function**

In *PatchMatch* the patches are treated identically regardless of their quality, in particular in the low resolution levels of the pyramid. This leads to a spread of low quality data throughout the mask. To counter that, in the propagation step, beside verifying that the candidate patch has a better similarity than the current considered patch, we also use validation to verify that the center pixel $q$ of the candidate patch has a larger confidence than the mask pixel $p$. Note we are using $\alpha = 1$ in validation for this case. Our modifications are shown in red in Alg. 1. Fig. 5 visually illustrates the effect of our modifications.

### B. Adapted statistic and graphcut approach

As mentioned in Section II, *statistic analysis* followed by *graphcut* is more adapted than *PatchMatch* for inpainting textures with a regular pattern. Alg. 2 details the steps of the considered algorithm [18], [19]. *Statistic analysis* consists in calculating for each patch of the known zone $I_{|M^C}$ its associated offset (2.4). The offset of a patch centered in $p$ is the translation vector $t$ such that:

$$I(p+t) = \underset{u, \|u\| > \tau}{\arg\min}\ \text{sim}\left(I(p), I(p+u)\right)$$

where sim is a similarity function (based on a Sum of Squared Differences) and $\tau$ is a threshold to avoid selecting patches which are too close to the considered patch. Then, for each offset, its occurrence is computed and the most expensive $K$ offsets are selected (they should characterize the regularity of a structured texture). Finally, *Graphcut* is performed (2.5) to create the labels map by minimizing an energy function in the form

$$E = E_{\text{data}}(p, t) + E_{\text{reg}}(p, q, t_p, t_q),$$

where $E_{\text{data}}$ measures the cost of associating the offset $t$ to pixel $p$, and $E_{\text{reg}}$ is the regularization term used to express the visual consistency of neighbour pixel values pointed by the offset map. Like *PatchMatch*, this approach is multi-resolution and there is no distinction between the "low" and "high" quality data during the energy minimisation. We integrate our criterion into $E_{\text{data}}$: for a pixel $p$ and a offset $t$, $E_{\text{data}}(p, t)$ is set to 0 if $p + t$ does not belong to the mask AND if validation$(p, p + t, C, \alpha) = 1$, otherwise it is set to $+\infty$. It allows to penalize the couple $(p, t)$ if trust$(p+t) <$ trust$(p)$.

In the case of regular textures, it may happen that we do not find an offset that allows to complete a pixel because the only offsets considered point to pixels in the mask. In this case, we could allow a pixel $p$ with a $\mathcal{C}(p)$ value to use an offset pointing to a pixel $q$ with a smaller $\mathcal{C}(q)$ value but close to $\mathcal{C}(p)$. For that, we reset the tolerance parameter $\alpha$ in (7): at the beginning of the Graphcut step, $\alpha$ is set to 1. If we lack candidates for finding the right label for a mask pixel, we decrease $\alpha$ by a fixed value in order to loosen the constraint and use data of lower slightly quality.

We show in the next section some results based on our adapted algorithm.

## VI. Experimental Results

We implemented the proposed method in C++, using CImg [20] for image manipulation and its *PatchMatch* implementation on Android devices. For perspective removal we use the

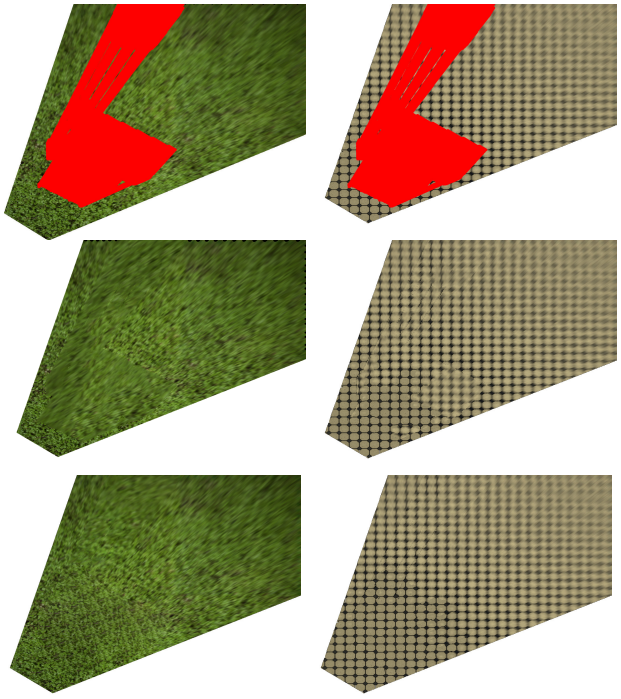

Fig. 6. Results from *PatchMatch* (left) and *Statistic analysis and Graphcut* (right) on an empty area coloured in red (first row) without validation criterion (second row) and with validation criterion (third row).

interpolation implementation by [21], while for pixel labeling in the offsets approach we use the implementation of Graphcut [18], [19], [22]. We compute the plane homography by using its corners which are automatically computed in synthesis cases or with ARCore [23].

Fig. 6 shows the results for two examples of undistorted views with synthetic textures, a stochastic and a regular texture. The mask, in red, has been defined by the removal of objects (see the first row of Fig. 6). We use the state of the art *PatchMatch* algorithm for the first case and the Graphcut algorithm for the second case. The second row of the Fig. 6 shows that the original result is very blurred because low quality data has been propagated. The third row (with the validation criterion) shows the propagation has been handled without degrading the inpainting process.

Fig. 7 shows results of the final output of our DR pipeline on a synthetic image (top row) and two real images (second and third row). For the rectified images shown in the last section, we applied a luminosity adjustment using a diffusion based inpainting method in order to create a more photo-realistic output. For each case, we have a propagation of the "low" resolution data to the "high" resolution data zone in the classic approaches, whereas our validation criterion added in the inpainting methods handles better the propagation.

Our implementation gives improved results with stochastic textures, even in complex situations. In the regular texture case, we adjust the parameter $\alpha$ as explained in §V. However, our method supports the propagation of the high quality data, possibly creating a resolution discontinuity at the boundaries of the mask. Thus, after the inpainting process, we apply a Gaussian blur whose kernel varies for each pixel $p$ according to their confidence $\mathcal{C}(p)$. At each pixel $p$ the kernel width $\sigma$ is chosen inversely proportional to the square root of $\mathcal{C}(p)$.

## VII. Conclusion and Future Work

We proposed a criterion to better guide the selection of patch in inpainting methods that takes advantage of the planar geometry of the scene to better guide the inpainting process and avoid the propagation of low resolution data. We have shown these improvements handle the blur effect without degrading the quality of the final output. We have also used the criterion a posteriori to recover a continuous quality in the inpainted data.

As future work, we plan to extend our approach to the multi-view setting. In such scenario the global rectified image of the plane is generated from all the views. The trust map is not any more continuous, as the $\mathcal{C}(p)$ values at the boundaries between the projections of views are not continuous. Pixels of the rectified plane may come from multiple original images, thus leading to the problem of properly defining the confidence when multiple sources are available.

## Acknowledgments

This research was supported by the CIFRE ANRT 2016/0139 and the regional project FEDER-FSE Midi-Pyrénées et Garonne REALISM n.15056690.

## References

[1] S. Mori, S. Ikeda, and H. Saito, "A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 17, dec 2017. [Online]. Available: http://doi.org/10.1186/s41074-017-0028-1

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publ. Co., 2000, pp. 417–424. [Online]. Available: http://dx.doi.org/10.1145/344779.344972

[3] N. Kawai, T. Sato, and N. Yokoya, "Diminished reality based on image inpainting considering background geometry," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 99, pp. 1–1, 2015. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2015.2462368

[4] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proceedings of the Europeran Conference on Computer Vision (ECCV)*, 2012. [Online]. Available: https://doi.org/10.1007/978-3-642-33715-4_54

[5] M. Bertalmío, V. Caselles, S. Masnou, and G. Sapiro, "Inpainting," in *Computer Vision*. Boston, MA: Springer US, 2014, pp. 401–416. [Online]. Available: http://doi.org/10.1007/978-0-387-31439-6_249

[6] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436 – 449, 2001. [Online]. Available: http://dx.doi.org/10.1006/jvci.2001.0487

[7] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 2, June 2003, pp. II–721–II–728 vol.2. [Online]. Available: https://doi.org/10.1109/CVPR.2003.1211538

[8] P. Buyssens, M. Daisy, D. Tschumperle, and O. Lezoray, "Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1809–1824, June 2015. [Online]. Available: http://doi.org/10.1109/TIP.2015.2411437

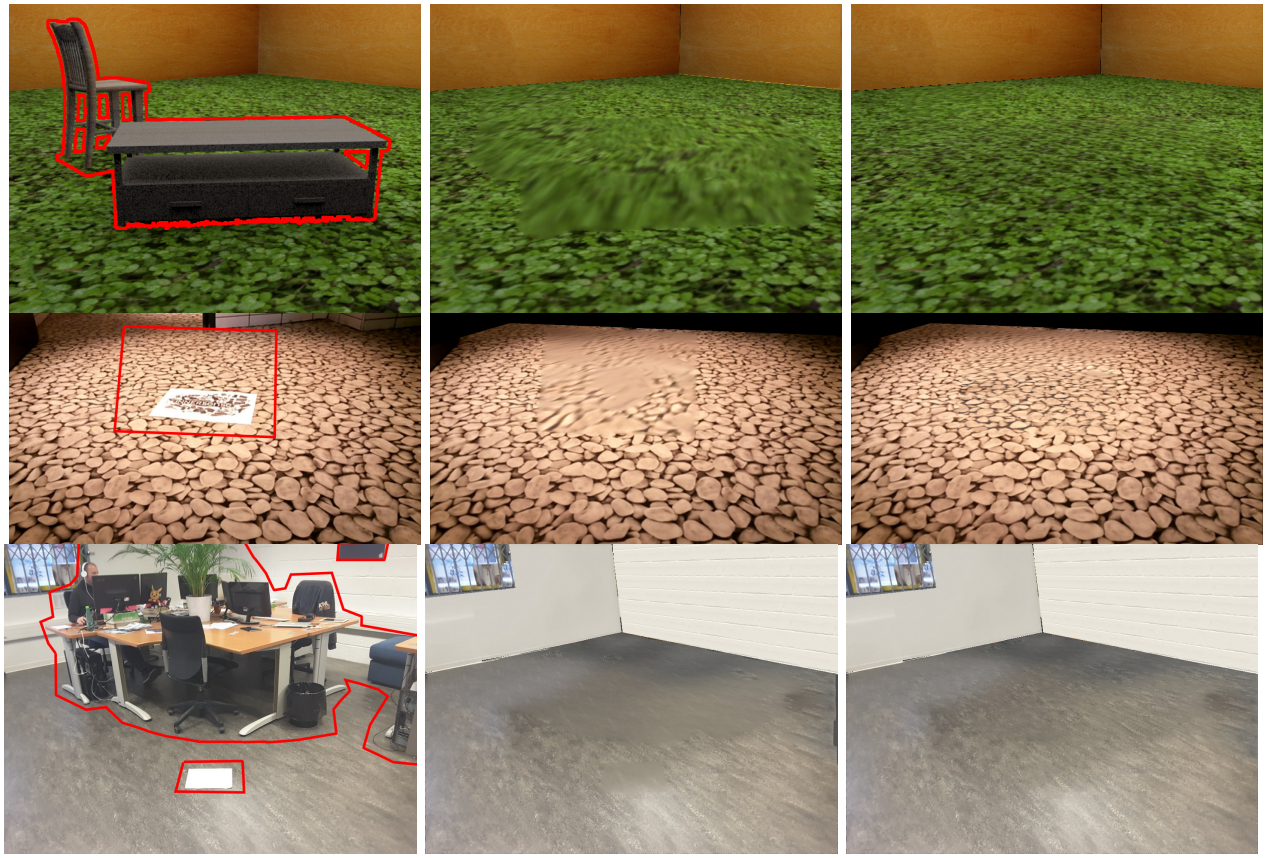

Fig. 7. Results from a global Diminished Reality pipeline. First column: input images, second column: result with a state-of-the-art approach, third column: result with our approach. For the third case (third row), an inpainting + trust criterion is applied for the ground and the left wall, a graphcut synthesis [24] is applied for the right wall.


[9] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1033–1038 vol.2. [Online]. Available: http://doi.org/10.1109/ICCV.1999.790383

[10] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, Aug. 2009. [Online]. Available: http://doi.acm.org/10.1145/1531326.1531330

[11] G. Chican and M. Tamaazousti, "Constrained patchmatch for image completion," in *Proceedings of the 10th International Symposium on Advances in Visual Computing ISVC*. Springer International Publishing, Dec. 2014. [Online]. Available: https://doi.org/10.1007/978-3-319-14249-4_53

[12] K. He and J. Sun, "Statistics of patch offsets for image completion," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 16–29. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33709-3_2

[13] M. Köppel, M. B. Makhlouf, K. Müller, and T. Wiegand, "Fast image completion method using patch offset statistics," in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP)*, Sept 2015, pp. 1795–1799. [Online]. Available: http://doi.org/10.1109/ICIP.2015.7351110

[14] Yunqiang Liu and V. Caselles, "Exemplar-Based Image Inpainting Using Multiscale Graph Cuts," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1699–1711, may 2013. [Online]. Available: http://doi.org/10.1109/TIP.2012.2218828

[15] S. Siltanen, "Diminished reality for augmented reality interior design," *The Visual Computer*, pp. 1–16, 2015. [Online]. Available: http://dx.doi.org/10.1007/s00371-015-1174-z

[16] D. Pavic, V. Schönefeld, and L. Kobbelt, "Interactive image completion with perspective correction," *The Visual Computer*, vol. 22, no. 9, pp. 671–681, 2006. [Online]. Available: http://dx.doi.org/10.1007/s00371-006-0050-2

[17] C. Eisenacher, S. Lefebvre, and M. Stamminger, "Texture synthesis from photographs," *Computer Graphics Forum*, vol. 27, no. 2, pp. 419–428, 2008. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2008.01139.x

[18] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, sep 2004. [Online]. Available: http://doi.org/10.1109/TPAMI.2004.60

[19] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001. [Online]. Available: http://doi.org/10.1109/34.969114

[20] D. Tschumperlé, "The CImg Library," in *IPOL 2012 Meeting on Image Processing Libraries*, Cachan, France, Jun. 2012, p. 4 pp. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00927458

[21] P. Thevenaz, T. Blu, and M. Unser, "Interpolation revisited [medical images application]," *IEEE Transactions on Medical Imaging*, vol. 19, no. 7, pp. 739–758, jul 2000. [Online]. Available: http://doi.org/10.1109/42.875199

[22] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, feb 2004. [Online]. Available: http://doi.org/10.1109/TPAMI.2004.1262177

[23] "Arcore," https://developers.google.com/ar/.

[24] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 277–286, Jul. 2003. [Online]. Available: http://doi.acm.org/10.1145/882262.882264