Depth map inpainting under a second-order smoothness prior

Daniel Herrera C.†, Juho Kannala†, Ľubor Ladický‡, and Janne Heikkilä†

†Center for Machine Vision Research
University of Oulu, Finland
{dherrera,jkannala,jheikkila}@ee.oulu.fi

‡Visual Geometry Group
University of Oxford, UK
lubor@robots.ox.ac.uk

Abstract. Many 3D reconstruction methods produce incomplete depth maps. Depth map inpainting can generate visually plausible structures for the missing areas. We present an inpainting method that encourages flat surfaces without favouring fronto-parallel planes. Moreover, it uses a color image to guide the inpainting and align color and depth edges. We implement the algorithm efficiently through graph-cuts. We compare the performance of our method with another inpainting approach used for large datasets and we show the results using several datasets. The depths inpainted with our method are visually plausible and of higher quality.

Keywords: depth map, inpainting, second order prior, graph cut

1 Introduction

Existing 3D reconstruction methods have different strengths and weaknesses. Some are more robust than others, but they all have cases where the reconstruction fails and no depth is estimated for an area of the image. For example, Time-of-Flight cameras and structured lighting methods like the Kinect, cannot reconstruct over or under-exposed areas. Stereo methods fail when there is no visible texture in the surface.

This often results in a semi-dense depth map that has accurate depth for some pixels but no depth for others. Estimating the depth of these missing regions is a severely ill-posed problem since there is very little information to be used. Recovering the true depth would only be possible with very detailed prior knowledge of the scene. However, in some applications (e.g. image-based rendering) it is enough to estimate a visually plausible structure for the scene. Depth map inpainting can then be a good solution to estimate the depth of the missing regions.

We consider two requisites for a scene to have a visually plausible structure. First, a surface is expected to be continuous and smooth. Second, depth discontinuities between surfaces are expected to be aligned with intensity or color

discontinuities. During inpainting, the pixels around the boundary of the missing region can be used to address the smoothness constraint and, since most 3D reconstruction methods produce a depth map aligned with a color image, we can use this color image to satisfy the edge alignment constraint.

Because depth map inpainting is so ill-posed, the prior assumptions on the scene structure play a dominant role. Simple priors lead to poor estimations but complicated priors can be intractable to solve. Our contribution consists of a depth map inpainting method that produces visually plausible structures encouraging piecewise planar surfaces. The solution is efficiently computed through Quadratic Pseudo-Boolean Optimization (QPBO) [1] using a second-order prior that favors a constant depth gradient while aligning depth and color discontinuities.

2 Previous work

Most stereo methods also include a prior on the generated depth map [2]. If a dense depth map is desired, a prior is necessary to regularize areas with low texture. Traditionally, a robust (first order) smoothness prior has been used. However, this favours fronto-parallel planes and leads to a staircase effect, thus higher order priors are recommended [3].

Herrera et al. [4] implemented a planarity constraint directly in their depth map inpainting algorithm. However, color and depth are used independently without exploiting the joint information. Levin et al.'s inpainting algorithm [5] has been succesfully used to inpaint depth maps [6]. Although it was originally meant for colorization, color and depth share a similar relation to intensity and the results are visually pleasing.

Yang et al. [7] exploit the relation between intensity and depth to perform super resolution of depth maps. They use bilateral filtering to align the upsampled depth discontinuities with the high-resolution intensity edges. They obtain visually pleasing results but their formulation cannot be easily extended to other applications. Gandhi et al. [8] also start with a low resolution depth map, but they use it to construct a prior for a stereo camera system. Their prior improves the stereo reconstruction, but depends on an active Time-of-Flight camera.

In some situations, as in image-based rendering, the depth map is only an intermediate step. Fitzgibbon et al. [9] apply the prior directly on the color of the synthesized view, thus ignoring depth ambiguities arising from similar colors. In the multi-view case, the priors do not necessarily have to be applied on the image level. Gargallo and Sturm [10] implement a multi-view depth map prior in 3D space. It enforces depth map overlap and smoothness with discontinuities. The prior is expensive but is crafted so that it can be efficiently applied to small sets of neighbouring points.

Woodford et al. [3] introduced a multi-view stereo method that can efficiently solve a second-order prior using QPBO [1]. They formulate the problem as an energy minimization. The prior is implemented through triplets of neighbouring

pixels. They showed that QPBO can find very good local minima for such an energy even though it is NP-hard to find the global minimum. Our inpainting approach uses a similar formulation for the second-order prior and also uses QPBO to minimize it. However, being an inpainting method, the prior has a greater influence than the data term and the proposal generation scheme is different.

3 Energy framework

In the depth map inpainting problem we have two images, the depth map $\mathbf{Z}$ and the color image $\mathbf{I}$, both in the same reference frame. The depth map is incomplete, for example because it was produced by the Kinect. Therefore it has regions with missing values that we wish to estimate, whereas the color image is complete.

We define our energy over these two images as a combination of a data term and a smoothness term. The data term applies to individual pixels while the smoothness term is defined over pixel triplets.

$$E = \sum_{p} E_d(p) + \lambda \sum_{\{p,q,r\} \in \mathcal{N}} E_s(p,q,r), \tag{1}$$

where p, q, and r are pixel coordinates of the form $[u, v]^\top$. $\mathcal{N}$ contains all three consecutive pixels, horizontally and vertically, so that p is the left or top-most pixel and r the right or bottom-most. Finally, λ adjusts the relative weight between the terms.

3.1 Data term

In most inpainting problems we know nothing about the missing values, thus no data term is needed. However, in some cases, like when inpainting depth maps acquired with the Kinect, the pixels around the boundary of the missing region are known to be noisy. In these cases we can include the pixels around the boundary in the inpainting process and include a data term that favours the observed values

$$E_d(p) = \begin{cases} \rho_d(\hat{\phi}(p) - \phi(Z(p))) & \text{if } \exists \hat{\phi}_p \\ 0 & \text{else,} \end{cases} \tag{2}$$

where $\hat{\phi}(p)$ is the observed depth value for pixel p in measurement units and $Z(p)$ is the inpainted depth value in metric units for pixel p. The function $\phi(\cdot)$ transforms the depth from metric units to measurement units and depends on the measurement device. ρ_d is the robust measure

$$\rho_d(x) = \min(x^2, \tau_d^2) \tag{3}$$

that limits the contribution of each pixel. The argument τ_d depends on the measurement units and the expected noise of a measurement that is not an outlier.

3.2 Second-order prior

The smoothness prior penalizes changes in the depth derivative, thus encouraging regions of constant derivative (i.e. flat planes). The cost for each triplet is

$$E_s(p,q,r) = W_s(p,q,r)\rho_s(Z(p) - 2Z(q) + Z(r)). \tag{4}$$

The argument $Z(p)-2Z(q)+Z(r)$ directly measures the second derivative of the depth map. ρ_s is a robust measure that limits the contribution of each triplet and allows discontinuities

$$\rho_s(x) = \min\left(|x|, \tau_s\right). \tag{5}$$

We note that the absolute value was used for ρ_s instead of the square of the argument because QPBO is not able to minimize the energy otherwise.

The triplet weighting term W_s is used to decrease the strength of the prior in areas where the color gradient is high. This weighting is defined over pixel triplets but the color gradient is defined over pixel pairs. We define a pairwise weighting and take the minimum weight of the two pairs of consecutive pixels, i.e. $W_s(p,q,r) = \min(W(p,q), W(q,r))$.

The pairwise weighting $W(p,q)$ can be for pixels that are horizontally or vertically consecutive. However, we use the color gradient magnitude in either case and not just it's horizontal or vertical component. Slanted edges may have a weak component in one direction but we still consider this pixel part of an edge and it should not enforce the smoothness constraint any further. The weighting function is then defined as

$$W(p,q) = \exp\left(\frac{-|\nabla I(p)|^2}{2\sigma_I^2}\right), \tag{6}$$

regardless of whether it is a horizontal or vertical pair. σ_I^2 is the variance of the expected noise in the color gradient. The color gradient itself is defined as

$$\nabla I_x(p) = |I(p) - I(p + [1,0]^\top)| \tag{7}$$

$$\nabla I_y(p) = |I(p) - I(p + [0,1]^\top)| \tag{8}$$

$$|\nabla I(p)| = \sqrt{\nabla I_x(p)^2 + \nabla I_y(p)^2}. \tag{9}$$

We used a simple Euclidean distance in RGB space to compute the magnitude of a color difference in Equations 7 and 8. It is expected that a better color distance measure would be more robust, however it was not necessary in our experiments.

4 Optimization

We now have an energy E that is a continuous function of a depth map and its color image. We reduce the minimization of this energy to a series of binary problems in a similar fashion to [3, 11]. Each binary problem can then be represented as a graph-cut problem.

4.1 Binary problems

The binary problems are a generalization of the α-expansion algorithm [1]. Suppose we have a current depth map estimate $\mathbf{Z}^{[t]}$ and a proposal estimate $\mathbf{Z}'$. We have one binary variable for each pixel b_p that indicates if it retains the depth from the current estimate or takes the depth from the proposal. Thus, the depth after this binary problem is

$$Z(p)^{[t+1]} = (1 - b_p)Z(p)^{[t]} + b_p Z(p)'. \tag{10}$$

In the basic α-expansion algorithm [12] each proposal would be a fronto-parallel plane, however this leads to a poor local minimum in our case. We use more complicated proposals as detailed in the next section.

Each binary problem can then be represented as a graph-cut problem. However, as for [3], this leads to a non-submodular graph. We use QPBO [1] which is able to optimize non-submodular energies.

QPBO is not always able to label all pixels. Pixels with a set label ("0" or "1") are guaranteed to be the optimal labelling for this binary problem. If the "unknown" labels are set to "0", QPBO guarantees that the energy will not increase, i.e. $E^{[t+1]} \leq E^{[t]}$. The number of unlabelled nodes depends strongly on the chosen energy and the image structure. For example, using a truncated square function for ρ_s produces almost only unlabelled pixels, while using a truncated absolute value produces very few unlabelled pixels. We select a label for the unknown pixels using an approach named QPBO-R in [3]. The unlabelled pixels are split into *strongly connected regions* as in [13]. For each region we independently select the labelling "0" or "1" which gives the lowest total energy.

4.2 Generating proposals

It is necessary to generate meaningful proposals for the optimization to work. In order to do this, pixels with valid depth are used to generate candidate planes.

First, the pixels to be inpainted are separated into connected regions and each connected region is processed separately. For each region the algorithm iterates around its boundary, generating one proposal for each boundary pixel. At each boundary pixel, three random pixels with valid depths are selected from a 10x10 neighbourhood. Each of these three random pixels defines a 3D point $P = [u, v, z]^\top$, where u and v are pixel coordinates and z is the metric depth. A plane is fitted to these three points. The proposal depth for the region is then obtained by calculating the depth of this plane at each pixel.

5 Experiments

We demonstrate the performance of our algorithm by inpainting depth map holes in a synthetic example and in several real life datasets. We compare our results with those produced with Levin et al.'s [5] approach. We selected Levin et al.'s approach as a comparison because it has been used for depth map inpainting in the NYU dataset [6].

5.1 Synthetic example

To demonstrate the advantages of using a second-order prior over a first-order we inpaint with a synthetic example. We generated a synthetic image of two curved surfaces that intersect each other, shown in Figure 1. The surfaces have two distinct colors. Salt and pepper noise with a magnitude of 4mm was added to the depth map so that the surfaces were not perfectly smooth. The pixels to be inpainted lie across the surface boundary.

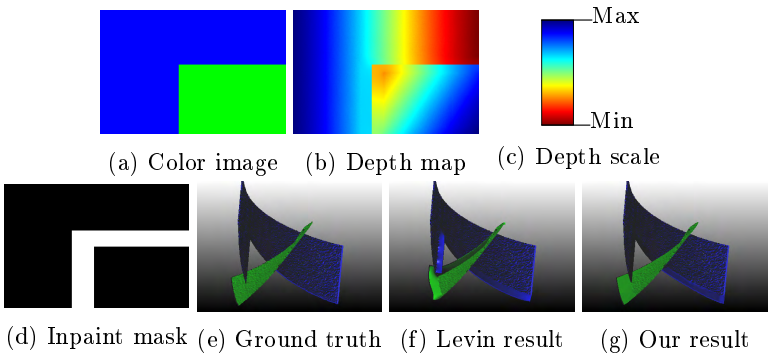


Fig. 1: Ground truth and results of the synthetic tests. Results rendered in 3D.

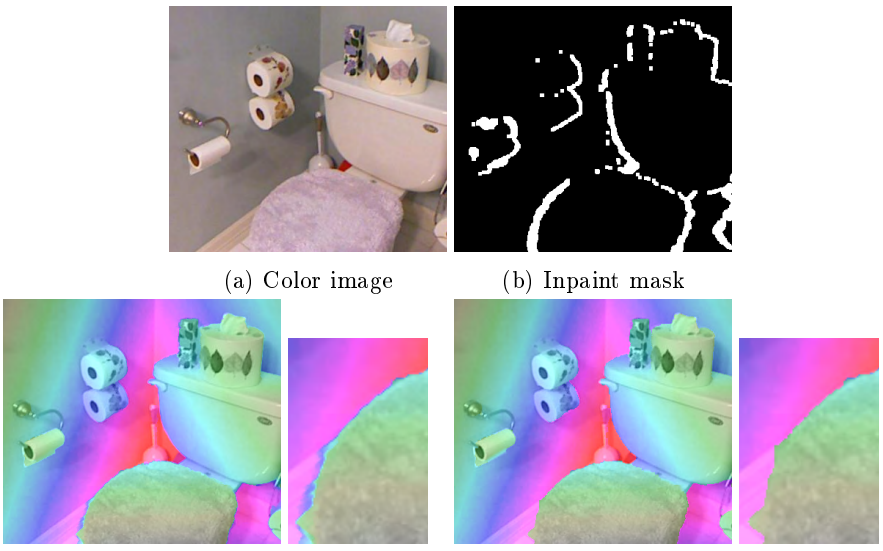
Levin's approach uses a first-order prior, i.e. it favors constant depth. Whereas our second-order prior favors constant depth derivative. This is clearly seen in the results of Figure 1. Levin's approach correctly separates the surfaces but fills the missing pixels with a constant depth, while our method provides a smooth result that matches the ground truth shape. The root-mean-square error (RMSE) for Levin's result is 34.6mm while ours is only 5.2mm. The RMSE of our approach is only 1mm larger than the original noise.

5.2 Inpainting the Kinect

We applied both algorithms to inpaint holes in depth maps captured with the Kinect. The Kinect is not able to reconstruct all surfaces and thus the depth maps produced have holes. There is no ground truth for these holes however, so the comparison can only be done qualitatively. In addition to the missing depth, it is also known that the pixels in the boundary of the hole are noisy. Therefore the mask for the pixels to inpaint was dilated with a 5x5 square element. The data term of Eq. 2 was used to include these noisy values.

Figure 2 shows the inpainting of an image from the NYU depth dataset V2 [6]. The inpainted depth map is overlayed on the color image to show the alignment of the depth and color edges. We see that both algorithms align the edges properly. However, Levin's approach produces smooth edges, which creates

artefacts in the form of pixels with intermediate depth. Our approach produces aligned and sharp boundaries.



(a) Color image (b) Inpaint mask

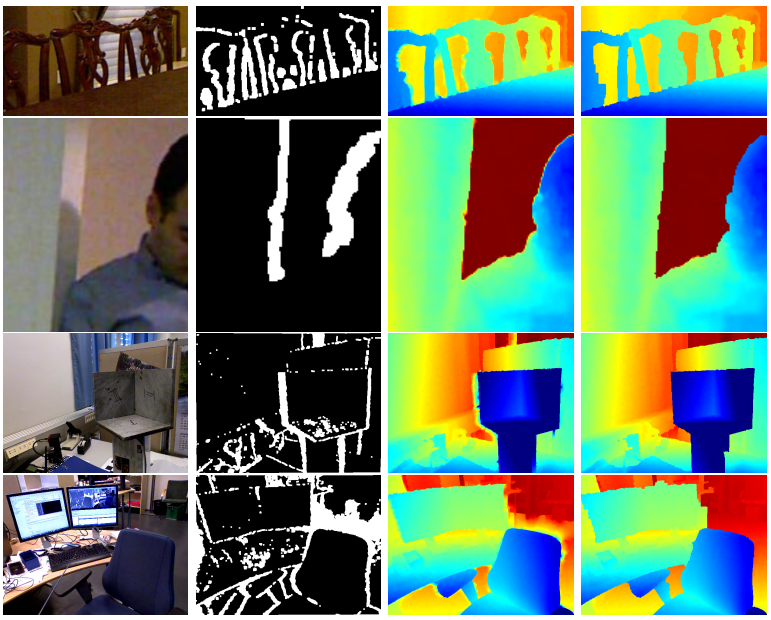
(c) Levin's result (full and zoom) (d) Our result (full and zoom)

Fig. 2: Edge alignment with Kinect inpainting. The zoomed area shows the lower left edge of the toilet seat. Both algorithms align the edges but our result is sharper. The similar color between the seat and the floor tile causes both algorithms to extend the surface over the tile.

Figure 3 shows more inpaintings of Kinect images. The first two images are also from the NYU dataset and the last three were taken by ourselves. In all instances our algorithm produces sharper depth edges that are aligned with the strong color edges. In cases where the color image doesn't provide strong enough edges to determine a clear boundary (see Fig. 2 and the chairs in Fig. 3) our method still produces a sharp boundary but is not able to align it properly.

5.3 Quantitative comparison

To obtain a quantitative comparison we perform inpainting in two datasets that have depth map ground truth. The first is the Middlebury 2005 stereo dataset [14]. It contains six images with ground truth but the depth is heavily quantized. The second are the multi-view dense stereo evaluation images from Strecha et al. [15]. We used six of the images with a ground truth model (i.e. fountain-P11 and Herz-Jesu-P8). In both datasets artificial holes were created manually to demonstrate inpainting performance.



(a) Color image (b) Inpaint mask (c) Levin's result (d) Our result

Fig. 3: Kinect inpainting results. Our algorithm produces sharper boundaries, specially over large areas (see the upper part of the chair).

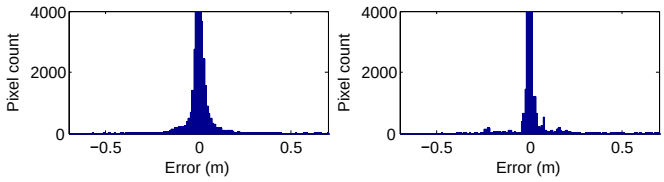
Figure 7 shows the inpainting results on the Middlebury dataset. Our algorithm has consistently sharper boundaries that align well with the color image. Table 1 presents the root mean squared error (RMSE) of both algorithms. Although our results look more visually plausible than Levin's, the RMSE is worse for our approach. However, the robust root-median-squared error (RMdSE) is consistently worst for Levin's results.

This is not yet conclusive, because the RMSE is susceptible to outliers, whereas the RMdSE is robust to outliers but can ignore up to 50% of the data. Thus we analyse the structure of the errors more carefully. Figure 4 shows histograms of the errors over all Middlebury images. Our error distribution is clearly narrower, implying better performance. This means that a few outliers are disturbing the RMSE measure.

To eliminate outliers but avoid using the median measure, we drop 2% of the pixels with the highest error and use the remaining pixels to calculate the RMSE. Figure 5 shows the pixels that were dropped for a couple of images. We see that they are mostly at the object boundaries, which is to be expected from small misalignments with the ground truth. Table 2 shows the RMSE and RMdSE without these pixels. The RMSE now shows clearly better performance for our algorithm, confirming the suspicion that outliers were skewing the mean of the error distribution. These outliers mean that for some pixels the result provided

Table 1: Error measure for the Middlebury dataset. Levin's approach has better RMSE but our error is lower with the robust RMdSE.

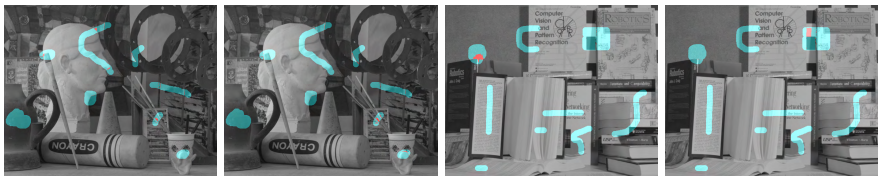
File	RMSE (mm)		RMdSE (mm)	
	Levin's	Ours	Levin's	Ours
Art	**35.3**	36.8	3.1	**1.5**
Books	23.3	**16.1**	1.5	**0.0**
Dolls	**12.0**	16.4	2.9	**2.4**
Laundry	37.6	**31.2**	1.1	**0.0**
Moebius	**25.5**	39.0	2.9	**1.0**
Reindeer	**27.2**	32.2	3.3	**1.1**
Total	**27.7**	29.7	2.4	**0.7**



(a) Levin's (b) Ours

Fig. 4: Histogram of errors. Levin's approach has a wider error distribution. The peak is out of scale for both graphs.

by the proposed approach may be slightly further away from the ground truth than the overly smooth result by Levin's approach. This is expected because the problem is severely ill posed, but in practice these cases are not very significant and appear rarely. Moreover, they are still more visually plausible because they are sharper and align with color edges.



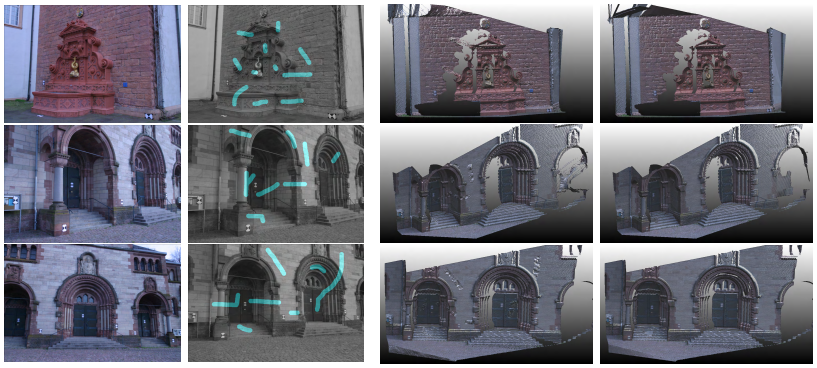
(a) Levin's (b) Ours (c) Levin's (d) Ours

Fig. 5: Dropped pixels. Red areas show the 2% of pixels with highest error.

We also compare the performance of both algorithms with the Strecha dataset. Figure 6 shows some of the inpainting results and Table 3 shows the evaluation with ground truth. We observe the same behaviour as with the Middlebury dataset.

Table 2: Error measure for the Middlebury dataset without the 2% highest errors. Our approach has consistently better performance.

File	RMSE (mm)		RMdSE (mm)	
	Levin's	Ours	Levin's	Ours
Art	20.1	**4.7**	3.0	**1.4**
Books	13.1	**8.6**	1.4	**0.0**
Dolls	8.2	**6.6**	2.8	**2.3**
Laundry	21.1	**3.1**	1.0	**0.0**
Moebius	**17.0**	22.6	2.8	**0.8**
Reindeer	14.8	**13.7**	3.2	**1.0**
Total	16.1	**12.3**	2.3	**0.6**



(a) Color (b) Mask (c) Levin's result (d) Our result

Fig. 6: Inpainting with the Strecha dataset. Results are plotted in 3D. Levin's results are noisier than ours and the boundaries are not always sharp.

6 Conclusions

We presented a method to inpaint depth maps using a second-order prior and a color image as guidance. The second-order prior encourages smooth and planar surfaces without favouring fronto-parallel planes. The algorithm has been efficiently implemented using graph-cuts.

The inpainting performance of our algorithm has been demonstrated, both qualitatively and quantitatively, on a variety of datasets. Our results show a few outliers that are not consistent with our ground truth. However, the ground truth is only one instance of the possible visually plausible structures that could be inpainted. Given the ill-posed nature of the problem, this kind of outliers are expected. Moreover, they are not significant and appear rarely.

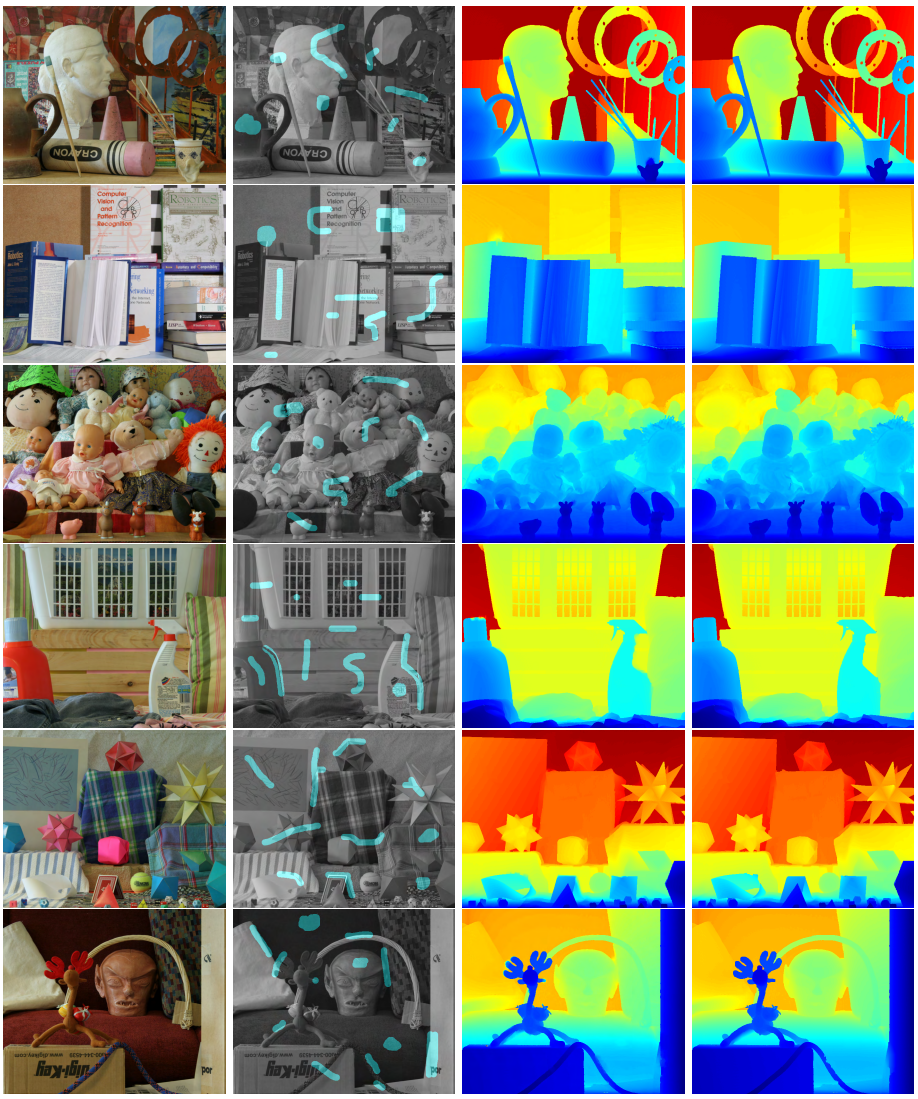
The results obtained demonstrate that the inpainted structures are visually plausible, i.e. the edges are sharp and aligned with color edges. We also demonstrate better performance than another approach used for depth map inpainting in the literature.

Table 3: Error measure for the Strecha dataset.

	RMSE (mm)		RMdSE (mm)	
	Levin's	Ours	Levin's	Ours
All pixels	**119.5**	145.7	17.1	**4.1**
2% dropped	61.8	**54.3**	16.4	**3.9**

References

1. Rother, C., Kolmogorov, V., Lempitsky, V., Szummer, M.: Optimizing binary mrfs via extended roof duality. In: CVPR. (2007)
2. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR, IEEE (2006) 519–528
3. Woodford, O., Torr, P., Reid, I., Fitzgibbon, A.: Global stereo reconstruction under second-order smoothness priors. PAMI **31**(12) (2009) 2115–2128
4. Herrera C., D., Kannala, J., Heikkilä, J.: Generating dense depth maps using a patch cloud and local planar surface models. In: 3DTV-CON, IEEE (2011)
5. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: SIGGRAPH. (2004)
6. Silberman, N., Kohli, P., Hoiem, D., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: ECCV. (2012)
7. Yang, Q., Yang, R., Davis, J., Nistér, D.: Spatial-depth super resolution for range images. In: CVPR, IEEE (2007)
8. Gandhi, V., Cech, J., Horaud, R.: High-resolution depth maps based on ToF-stereo fusion. In: ICRA. (2012) 4742–4749
9. Fitzgibbon, A., Wexler, Y., Zisserman, A., et al.: Image-based rendering using image-based priors. In: ICCV. Volume 2. (2003) 1176–1183
10. Gargallo, P., Sturm, P.: Bayesian 3d modeling from images using multiple depth maps. In: CVPR. Volume 2., IEEE (2005) 885–891
11. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? PAMI **26**(2) (2004) 147–159
12. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI **23** (2001) 1222âĂŞ–1239
13. Billionnet, A., Jaumard, B.: A decomposition method for minimizing quadratic pseudoboolean functions. Operations Research Letters **8** (1989) 161–163
14. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: CVPR. (2007)
15. Strecha, C., von Hansen, W., Van Gool, L., Fua, P., U., T.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: CVPR. (2008)



(a) Color (b) Mask (c) Levin's result (d) Our result

Fig. 7: Inpainting with the Middlebury 2005 dataset. The artificial holes were done across object boundaries which are the most challenging cases. The depth edges are clearly sharper for our algorithm.