

Dense Mono Reconstruction: Living with the Pain of the Plain Plane

Pedro Piniés

Lina Maria Paz

Paul Newman

Abstract— This paper is about dense depthmap estimation using a monocular camera in workspaces with extensive texture-less surfaces. Current state of the art techniques have been shown to work in real time with an admirable performance in desktop-size environments. Unfortunately, as we show in this paper, when applied to larger indoor environments, performance often degrades. A common cause is the presence of large affine texture-less areas like by walls, floors, ceilings and drab objects such as chairs and tables. These produce noisy and worse still, grossly erroneous initial seeds for the depthmap that greatly impede successful optimisation.

We solve this problem via the introduction of a new non-local higher-order regularisation term that enforces piecewise affine constraints between image pixels that are far apart in the image. This property leverages the observation that the depth at the edges of bland regions are often well estimated whereas their inner pixels are deeply problematic. A welcome by-product of our proposed technique is an estimate of the surface normals at each pixel. We will show that in terms of implementation, our algorithm is a natural extension of the often used variational approaches. We evaluate the proposed technique using real datasets for which we have ground truth models.

I. INTRODUCTION

Our goal here is to build dense maps of indoor scenes with a drastically reduced reliance on surface texture. As roboticists the case for monocular dense mapping is easily made. With low cost sensors we want to be able to perceive the complete 3D structure of the local workspace. This paper is about filling in the gaps, quite literally, in this competency. Gaps that are caused in dense maps through a paucity of texture.

Current state of the art methods for creating dense depthmaps with a monocular camera are based on powerful variational optimisation algorithms that are able to produce, in real time, dense volumetric reconstructions of desktop-size workspaces under stable lighting conditions [1, 2, 3]. In general, the objective function (or energy function) to be minimised consists of a data term that measures the photoconsistency over a set of consecutive images and a regularisation term that tends to preserve sharp discontinuities between objects located at different depths while simultaneously enforcing depth smoothness for homogeneous surfaces. A key step of the minimisation process involves the application of a primal-dual optimisation scheme which is widely used for solving variational convex energy functions that arise in many image processing problems [4].

Figure(1) shows a target workspace for us. The challenge is in dealing with large plain (and planar) structures like

authors are with the Mobile Robotics Group, University of Oxford, Oxford, England; {ppinies, linapaz, pnewman}@robots.ox.ac.uk



Fig. 1. Example of the success of the dense depthmap creation proposed in this paper after integrating 10 consecutive images for a monocular camera in a challenging environment: a long and elongated office corridor with large texture-less image regions and low-parallax due to the camera moving forwards. First row: (Left) Reference RGB image used to create the depthmap. (Right) Depthmap obtained with the method proposed. Notice the smoothness of the depthmap in all affine surfaces (floor, walls, ceiling) and the consistent U-shape distribution of colours for elements in the corridor that are at the same distance from the camera: red values are used for close objects whereas dark blue colours indicate far points from the camera. Second row: 3D coloured pointcloud obtained from back-projecting depthmap pixels.

floors, walls or ceilings. In such regions the photoconsistency term is of little help (all pixels are similar from all views) and regularisers which promote smoothness “struggle” to propagate information from distant boundaries.

We exploit the following idea: if we impose a strong prior about the predominant shape of the objects in the scene (for indoor environments we consider affine surfaces), we can propagate the estimated depth from pixels at the border of bland objects to their interior pixels by using a constraint

in the regularisation term that favours solutions with the predetermined family shape. This means that, unlike traditional variational methods that just consider local information of neighbouring pixels we need to relate distant pixels in the cost function. Building on the work presented in [5], our contribution is the introduction of a novel non-local higher-order regularisation term which takes cues from a soft-segmentation and yields dense maps across large affine yet utterly bland scene regions.

The closest work we know of to ours is [6] where the authors also consider planar constraints to improve the estimation of monocular depthmaps. In this work, there is a requirement for a nontrivial pre-processing stage to first acquire from external means plane normals before constructing optimisation constraints. This required segmentation of keyframes into a set of superpixels, classification of the superpixel into four predefined classes (wall, floor, ceiling, clutter) or matching superpixels between different keyframes using homographies. In contrast our method requires neither a preprocessing step before the optimisation nor an additional penalty term for regularisation. All of this is implicitly wrapped up in the form and structure of our new objective function.

II. PROBLEM OVERVIEW

A. Variational Optimisation

When a monocular camera is used, the state of the art algorithm [2] to create a depthmap $\xi(\mathbf{u})$ from a set of images $\mathbf{I}_\xi = \{I_0(\mathbf{u}), \dots, I_n(\mathbf{u})\}$ solves the following variational problem

$$\min_{\xi} E_R(\xi) + E_D(\xi) \quad (1)$$

where $E_D(\xi)$ is a nonconvex data term that calculates the average photometric error ρ between a reference image $I_r(\mathbf{u})$ and the warp of the remaining images in set \mathbf{I}_ξ

$$E_D(\xi) = \lambda \int_{\Omega} \frac{1}{|\mathbf{I}_\xi|} \sum_{k \in \mathbf{I}_\xi} \rho(I_r(\mathbf{u}), I_k(\mathbf{u}), \xi(\mathbf{u})) d\mathbf{u} \quad (2)$$

and $E_R(\xi)$ is a regularisation term, usually a Total Variation (TV) or Huber regulariser, that is able to preserve depth discontinuities while smoothing homogeneous regions

$$E_R(\xi) = \int_{\Omega} w(\mathbf{u}) \|\nabla \xi(\mathbf{u})\|_1 d\mathbf{u} \quad (3)$$

In eq.(2) λ is a parameter used to define the trade-off between the regulariser and the data term whereas $w(\mathbf{u})$ in eq.(3) is a per pixel weight based on the gradient of the reference image that reduces smoothing effects of the regulariser across image edges.

The optimisation problem in eq.(1) is solved using an iterative alternating optimisation method based on an exhaustive search step that involves the nonconvex data term $E_D(\xi)$, and a Primal-Dual algorithm [4] that solves a convex cost function [7] involving the regularisation term $E_R(\xi)$.

B. The Consequences of Little Texture

We have tested our own CUDA implementation of the previous (TV) algorithm on a NVIDIA GeForce GT 650M 1024 MB card using different synthetic scenarios found in [8, 9] obtaining a median depthmap error that is usually below 2 centimetres, after 800 iterations in 500 ms.

In Figure(2) we illustrate the performance of the algorithm for one of the synthetic datasets and the results obtained in two real scenarios for desktop-sized and office-sized environments. For the desktop-sized example we applied a sideways “wavy” movement to the camera, typical of this kind of experiment, which improves the parallax and therefore the estimated depth. For the office-size example the camera was mounted on a robot that was moving forward, which is, in fact, the most challenging movement for calculating parallax, but the natural one for collecting 3D models of indoor and outdoor environments. In all three datasets, the image set \mathbf{I}_ξ comprised 10 consecutive images.

Observe that for the majority of the pixels in the initial seed of the synthetic dataset shown in Figure(2(b)) we have, a reasonable estimate of the depth. At first sight this appears to be true even in low-textured areas like the right wall. This surprising result may be explained by the shadows cast by some objects on the wall (like the computer monitor) and the fact that the illumination pattern of the synthetic rendered scene is probably just a Lambertian approximation of the usually more complex lighting effects and reflectances found in real scenes. For the headset dataset we can see that the initial depthmap is clearly wrong for some of the pixels of the table, Figure(2(f)). However the presence of cables, papers, the headset and the calibration pattern, that have a good estimate and occupy most of the image, help the regulariser to propagate “good” estimates from correct to wrong pixels. Finally, Figure (2(j)) shows an extremely noisy initial seed for the office-size real environment due to the fact that most of the scene consists of plain white walls and the robot was moving in the direction of the field of view. Although the regulariser improves the initial solution, it cannot cope with the vast number of initial wrong pixels and the final depthmap obtained is of worse quality, as can be seen in the 3D pointcloud reconstruction shown in Figure (2(l)).

C. Proposed Solution Requirements

In order to deal with the problems discussed in the last example of the previous section (see Figure 2(j)) in the context of indoor environments our regulariser has to satisfy two requirements. Firstly, it has to be able to cope with many pixels corresponding to weakly textured areas since their initial depthmap estimates will be in gross error in most of the cases. Secondly, it has to favour affine solutions since it is likely that many pixels in the depthmap belong to the same 3D plane.

III. NON-LOCAL TOTAL GENERALISED VARIATION IN-PAINTING

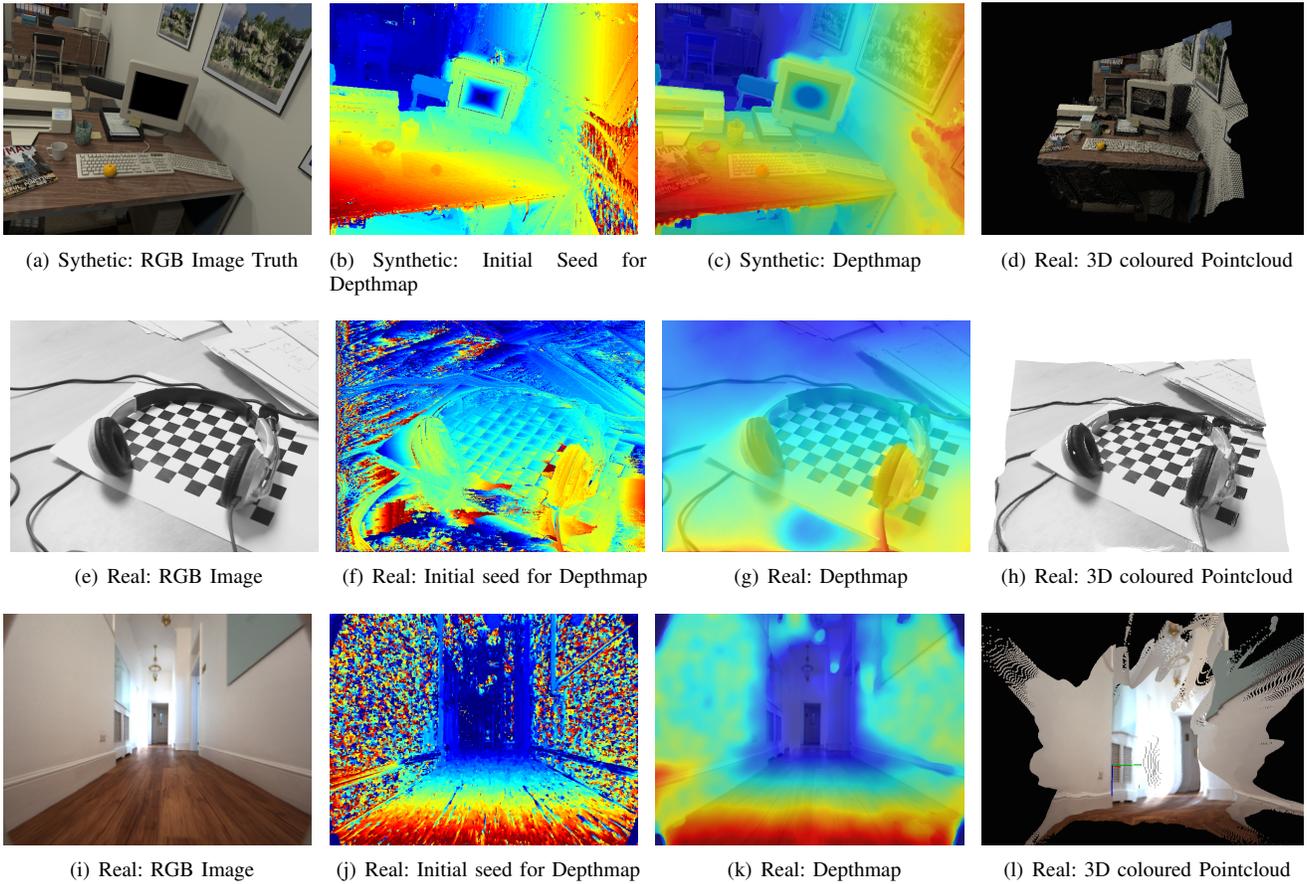


Fig. 2. Three examples of creation of depthmaps using a state of the art TV algorithm for synthetic (first row), real desktop-size (second row) and real office-size datasets (third row). For each dataset the first column shows the RGB images I_r used as reference to calculate the depthmap, the second column shows the initial seed obtained by just minimising $E_D(\xi)$ using exhaustive search, the third column contains the final depthmap created after regularisation and the fourth column shows the corresponding coloured pointcloud obtained from back-projecting the pixels estimated in the depthmaps. Observe that for the office-size environment we obtain an extremely noisy initial seed due to the lack of texture in the images (walls and ceiling) and the type of movement applied to the camera. As a result, the final depthmap calculated is of lower quality than the ones obtained for the previous datasets as can be verified in the corresponding 3D pointcloud in 2(l).

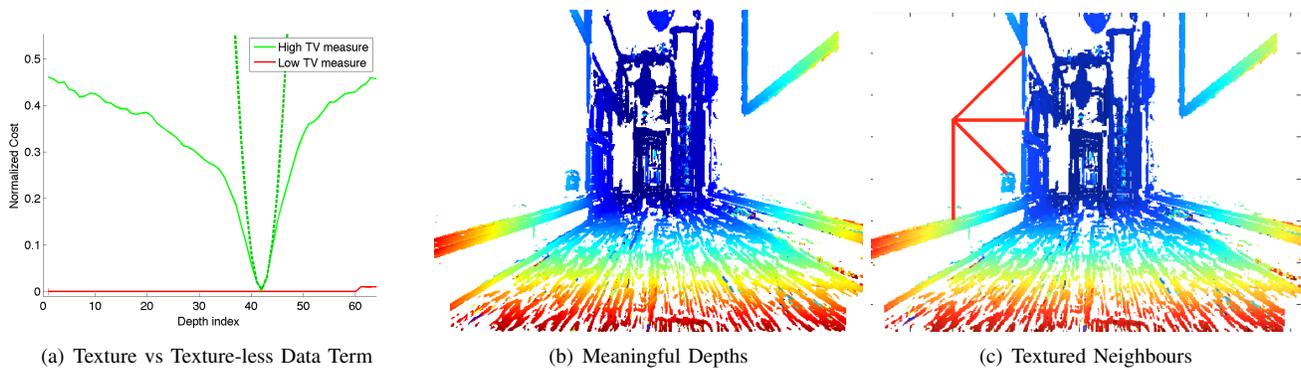


Fig. 3. Subfigure 3(a) shows the profile of the data term $E_D(\xi(\mathbf{u}))$ for a textured pixel \mathbf{u}_t (solid green) and texture-less one \mathbf{u}_{tL} (red) as well as a quadratic approximation of the data term at the minimum cost of \mathbf{u}_t (dashed green). The practical importance of this graph is that pixels well localised in depth use to have high curvatures at the solution. Subfigure 3(b) shows meaningful depth pixels that have been selected by thresholding the curvature at the initial depth solution. Subfigure 3(c) shows a simple heuristic used to select for texture-less pixels \mathbf{u}_{tL} a set of non-local meaningful candidates $\mathcal{N}(\mathbf{u}_{tL})$ with proper depth estimates. In the example shown four candidates, connected with red lines, have been found for the central pixel of the left wall

A. Primal-Dual Algorithm Review

In this section we give a brief review of the standard primal-dual optimisation algorithm since it will be used to

optimise the energy function proposed in this paper. Space precludes a step by step introduction to this machinery and with regret we must point the reader to [4] for a more detailed

Algorithm 1 primal-dual

```

1: {Initialization of variables:}
2:  $\tau, \sigma > 0, \theta \in [0, 1]$ 
3:  $\xi^0 = \eta, \mathbf{p}^0 = 0, \bar{\xi}^0 = \xi^0$ 
4: while  $k \leq N$  do
5:   {Dual step:}
6:    $\mathbf{p}^{k+1} = \Pi_{\mathbf{p}}(\mathbf{p}^k + \sigma K \bar{\xi}^k)$ 
7:   {Primal step:}
8:    $\xi^{k+1} = \Pi_{\xi}(\xi^k - \tau K^T \mathbf{p}^{k+1})$ 
9:   {Relaxation step:}
10:   $\bar{\xi}^{k+1} = \xi^{k+1} + \theta(\xi^{k+1} - \xi^k)$ 
11: end while

```

description ¹.

Given a noisy signal $\eta(\mathbf{u})$ a standard application of the primal-dual algorithm consists of calculating a denoised signal $\xi(\mathbf{u})$ by minimising the following energy function

$$\min_{\xi} \int_{\Omega} \alpha |\nabla(\xi(\mathbf{u}))|_1 + \lambda (\xi(\mathbf{u}) - \eta(\mathbf{u}))^2 d\mathbf{u} \quad (4)$$

If we discretise this equation we obtain

$$\min_{\xi} \alpha |K\xi|_1 + \lambda \|\xi - \eta\|_2 \quad (5)$$

where ξ and η are the images represented in vectorised form and operator K is a discretised version of the gradient operator. Algorithm 1 shows the basic steps of the primal dual algorithm where \mathbf{p} is an internal dual variable used during the optimisation and σ and τ are parameters that control the step size of the algorithm. Finally, the proximal map operators for the primal and the dual steps [10] can be calculated for each pixel ij individually using

$$\Pi_{\xi}(\xi_{ij}) = \frac{\xi_{ij} + \lambda\tau\eta_{ij}}{1 + \lambda\tau} \quad (6)$$

$$\Pi_{\mathbf{p}}(p_{ij}) = \frac{p_{ij}}{\max(1, |p_{ij}|/\alpha)} \quad (7)$$

B. Assessing Depth Estimates and Finding non-local Matchings

As we saw in Figure(2(j)) most of the pixels that correspond to low-textured regions have noisy and meaningless depth values for the initial seed obtained from exhaustive search of the data term $E_D(\xi)$. Figure(3(a)) shows, for a textured pixel \mathbf{u}_t (light green), the values taken by the data term $E_D(\xi(\mathbf{u}_t))$ along a range of inverse depth values $[\xi_{min}, \xi_{max}]$ and also the corresponding cost profile for a texture-less one \mathbf{u}_{tl} (red). Notice that for the textured pixel there is a clear minimum $\xi(\mathbf{u}_t)^*$ of the data term that corresponds to a well estimated depth whereas the low-textured pixel shows a flat profile that explains why any small noise in the original intensity images can randomly change the position of the minimum. According to these figures, we use the curvature of a second order approximation of the

¹because of space constraints we must proceed assuming that the reviewer is cognisant of primal-dual optimisation

data term at the minimum cost (shown in dashed green for the textured pixel only) as a measure of the reliability of the initial depth estimated $\xi(\mathbf{u}_t)^*$. Using this measure we can select only those pixels of the initial depthmap that have meaningful depths as can be seen in Figure(3(b)). During the optimisation process we will disregard invalid depth estimates of texture-less pixels by just setting $\lambda = 0$ in Equation(4). As explained in [4] this is equivalent to generate an interpolation solution (also known as in-painting in the computer vision literature) for those pixels that depends on the regulariser chosen.

During the optimisation, in order to speed up the transfer of depth information to invalid pixels we need to select, in addition to a local neighbourhood for each pixel, a set of potential meaningful non-local pixel candidates with valid depths. In this paper we use a simple approach to find potential candidates by just looking for the closest valid pixels along the main 8 star directions (E, NE, N, NW, W, SW, S, SE) as shown in Figure(3(c)). We denote the non-local neighbourhood of a texture-less pixel \mathbf{u}_{tl} by $\mathcal{N}(\mathbf{u}_{tl})$

C. Geometric constraint for pixels belonging to the same affine surface

In this Section we show that if two pixels \mathbf{u}_1 and \mathbf{u}_2 in an inverse depth image $\xi(\mathbf{u})$ belong to the same planar surface in 3D their inverse depth values $\xi(\mathbf{u}_1)$ and $\xi(\mathbf{u}_2)$ are constrained by the following affine equation

$$\xi(\mathbf{u}_1) - \xi(\mathbf{u}_2) = \langle \mathbf{w}, \mathbf{u}_1 - \mathbf{u}_2 \rangle \quad (8)$$

where $\langle \bullet, \bullet \rangle$ is the inner product between two vectors and \mathbf{w} is the normal vector that describes the orientation of the affine surface.

Given a camera with intrinsic parameters (f_u, f_v, c_u, c_v) , where f is the focal length and c the optical centre, the corresponding 3D point $\mathbf{x} = (x, y, z)$ for a pixel $\mathbf{u} = (u, v)$ in the image can be calculated using the following back-projection equation

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{\xi(\mathbf{u})} \begin{bmatrix} \frac{u-c_u}{f_u} \\ \frac{v-c_v}{f_v} \\ 1 \end{bmatrix} \quad (9)$$

If, in addition, the corresponding point \mathbf{x} belongs to a plane $\mathbf{p} = (\mathbf{n}, d)$ in 3D it has to accomplish

$$d = \langle \mathbf{n}, \mathbf{x} \rangle = n_x x + n_y y + n_z z \quad (10)$$

where $\mathbf{n} = (n_x, n_y, n_z)$ is the unitary normal vector of the plane and d is the orthogonal distance to the origin.

Substituting \mathbf{x} in Eq.(9) into Eq.(10) we obtain

$$\begin{aligned} d &= \frac{1}{\xi(\mathbf{u})} \left(n_x \frac{u-c_u}{f_u} + n_y \frac{v-c_v}{f_v} + n_z \right) \\ \xi(\mathbf{u}) &= \frac{n_x}{f_u d} u + \frac{n_y}{f_v d} v + \left(\frac{n_z}{d} - \frac{n_x c_u}{f_u d} - \frac{n_y c_v}{f_v d} \right) \\ &= \langle \mathbf{w}, \mathbf{u} \rangle + const \end{aligned} \quad (11)$$

where $\mathbf{w} = (w_u, w_v) = (\frac{n_x}{f_u d}, \frac{n_y}{f_v d})$ codifies the projection of the 3D surface normals into the image plane. Finally if \mathbf{u}_1 and \mathbf{u}_2 belong to the same 3D plane then from eq.(11)

$$\begin{aligned} \xi(\mathbf{u}_1) - \xi(\mathbf{u}_2) &= (\langle \mathbf{w}, \mathbf{u}_1 \rangle + const) - (\langle \mathbf{w}, \mathbf{u}_2 \rangle + const) \\ &= \langle \mathbf{w}, \mathbf{u}_1 - \mathbf{u}_2 \rangle \quad \square \end{aligned} \quad (12)$$

D. Optimisation Algorithm

We first create a semi-dense depthmap $\eta(\mathbf{u})$ containing meaningful depth estimates using the method explained in subsection III-B – Figure(3(b)) shows an example. Making use of the geometric constraint explained in subsection III-C the proposed energy function to be minimised is given by

$$\min_{\xi, \mathbf{w}} \int \int_{\Omega} \alpha_1(\mathbf{u}_1, \mathbf{u}_2) |\xi(\mathbf{u}_1) - \xi(\mathbf{u}_2) - \langle \mathbf{w}(\mathbf{u}_1), \mathbf{u}_1 - \mathbf{u}_2 \rangle|_1 \quad (13)$$

$$+ \alpha_2(\mathbf{u}_1, \mathbf{u}_2) |w_1(\mathbf{u}_1) - w_1(\mathbf{u}_2)|_1 \quad (14)$$

$$+ \alpha_2(\mathbf{u}_1, \mathbf{u}_2) |w_2(\mathbf{u}_1) - w_2(\mathbf{u}_2)|_1 d\mathbf{u}_1 d\mathbf{u}_2 \quad (15)$$

$$+ \lambda(\mathbf{u}) \int_{\Omega} (\xi(\mathbf{u}) - \eta(\mathbf{u}))^2 d\mathbf{u} \quad (16)$$

Equation (13) is the part of the regulariser in which we favour affine surfaces between pixels \mathbf{u}_1 and \mathbf{u}_2 using the constraint shown in eq.(8). Equations (14) and (15) impose a total variation constraint on the components of the estimated projected normals $\mathbf{w}(\mathbf{u}) = (w_1(\mathbf{u}), w_2(\mathbf{u}))$. Thus this term tries to impose similar normal vectors for homogeneous surfaces allowing at the same time large discontinuities between different surfaces. As it is explained in [5] this regularisation term can be considered as a non-local extension of a Total Generalised Variation norm [11]. Finally, eq.(16) is a standard data term that enforces pixels for which $\lambda(\mathbf{u}) \neq 0$ to be close to the input depthmap $\eta(\mathbf{u})$.

Coefficients $\alpha_1(\mathbf{u}_1, \mathbf{u}_2)$ and $\alpha_2(\mathbf{u}_1, \mathbf{u}_2)$ are used to incorporate soft-segmentation cues in the regulariser. In the current implementation these weights are based on the intensity similarity between pixels \mathbf{u}_1 and \mathbf{u}_2 in the reference image:

$$\alpha_1(\mathbf{u}_1, \mathbf{u}_2) \propto \exp\left(-\frac{|I_r(\mathbf{u}_1) - I_r(\mathbf{u}_2)|}{\sigma_i}\right) \quad (17)$$

where σ_i controls the influence of the neighbouring pixels. In addition, the coefficients are used as support weights to control the local and non-local influence of pixels. For all pixels support weight values are calculated for a local 7×7 window. To allow the transmission of depth information from meaningful depths to a texture-less pixel \mathbf{u}_{tl} we calculate additional support weights with its non-local neighbours $\mathcal{N}(\mathbf{u}_{tl})$.

Let us assume that m is the total number of support weights α_1 different from zero and that the images have n pixels. We can express the proposed energy function in a more compact matrix form after discretisation

$$\min_{\xi} |\alpha \circ K \xi^*|_1 + \|\lambda \circ (\xi^* - \eta^*)\|_2 \quad (18)$$

where \circ represents point-wise multiplication, $\alpha = [\alpha_1^T, \alpha_2^T, \alpha_2^T]^T$ is an $3m \times 1$ vector containing all support

TABLE I
MEDIAN ERROR

Range = [1.655 3.445] [m]	
Dataset 1	4 cm
Dataset 2	6.74 cm
Dataset 3	3.62 cm

weights, $\xi^* = [\xi^T, \mathbf{w}_1^T, \mathbf{w}_2^T]^T$ is a $3n \times 1$ extended vector containing the optimised depths ξ and the first \mathbf{w}_1 and second \mathbf{w}_2 components of the normals, $\eta^* = [\eta^T, 0_{1 \times 2n}]^T$ is a $3n \times 1$ extended input vector with the semi-dense depthmap η and additional padded zeros to match ξ^* size (we make the corresponding $\lambda = 0$ in λ for this additional components) and K is a sparse selection matrix that takes into account distances between matched pixels. Finally, and most importantly, note that the expressions in eq.(18) and eq.(5) are almost identical allowing us to make use of Algorithm 1 to solve the minimisation problem.

IV. EXPERIMENTS

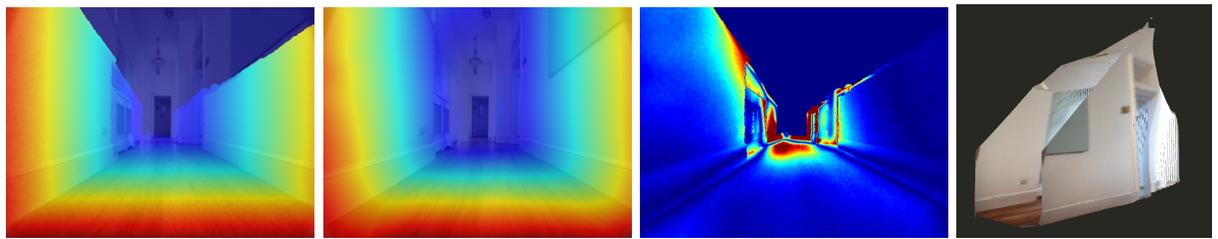
We evaluate our algorithm using three real datasets for which we have ground truth models computed from a push-broom laser. By projecting the laser points into the reference image we generate ground truth depthmaps that are used for comparisons. Since not all laser points get projected we use the corresponding subset to obtain our statistics. Figure(4) shows and compares ground truth depthmaps with the ones created using the technique proposed in this paper. Figure(5) shows histograms of the depth errors. For visualisation purposes all errors are saturated to a maximum of 0.5 metres. Finally, table I shows the median error for each dataset .

V. CONCLUSIONS

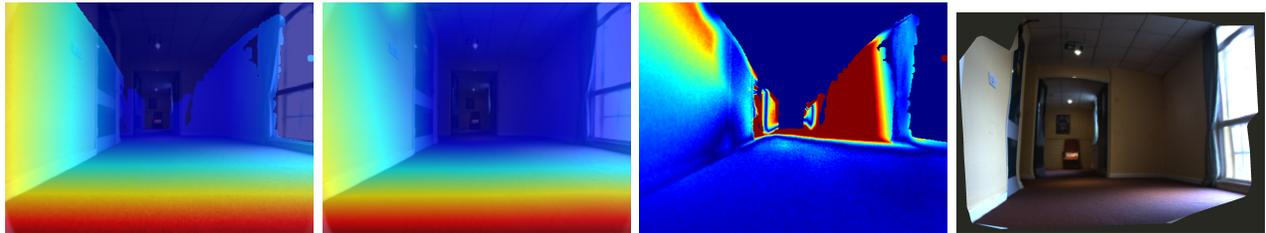
We have introduced a method which allows us to reconstruct dense depthmaps in workspaces which contain textureless yet affine surfaces - the plain plane. We have shown how to frame the problem as a non-convex optimisation problem which includes an energy term designed to propagate depth image across the scene - in particular from boundaries to interiors. We have further shown that the optimisation can be expressed in a familiar form which admits primal-dual optimisation. We have demonstrated the efficacy of our approach on a variety of data gathered from a robot moving on a trajectories designed to challenge the reconstruction process.

REFERENCES

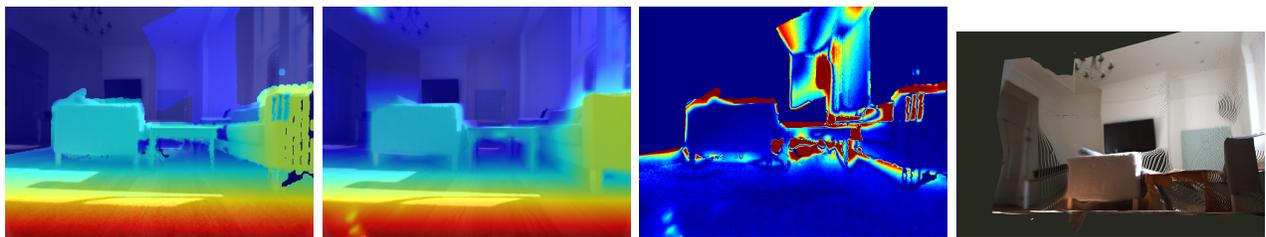
- [1] J. Stuehmer, S. Gumhold, and D. Cremers, "Real-Time Dense Geometry from a Handheld Camera," Darmstadt, Germany, September 2010, pp. 11–20.
- [2] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proceedings of the 2011 International Conference on Computer Vision ICCV*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2320–2327.
- [3] G. Graber, T. Pock, and H. Bischof, "Online 3D reconstruction using Convex Optimization," in *1st Workshop on Live Dense Reconstruction From Moving Cameras, ICCV 2011*, 2011.



(a) Dataset 1: Ground Truth Depthmap (b) Dataset 1: Estimated Depthmap (c) Dataset 1: Depthmap Error (d) Dataset 1: 3D coloured Pointcloud (Side view)

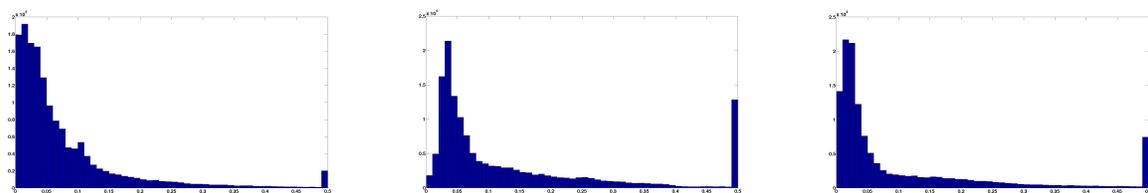


(e) Dataset 2: Ground Truth Depthmap (f) Dataset 2: Estimated Depthmap (g) Dataset 2: Depthmap Error (h) Dataset 2: 3D coloured Pointcloud



(i) Dataset 3: Ground Truth Depthmap (j) Dataset 3: Estimated Depthmap (k) Dataset 3: Depthmap Error (l) Dataset 3: 3D coloured Pointcloud

Fig. 4. Dense depthmap results obtained for 3 Datasets. First column presents ground truth depthmaps calculated from projecting laser points from a pushbroom laser. Second column shows the estimated depthmap. Third column calculates the absolute difference between the ground truth and the estimated depthmap. Errors range from 0 (dark blue) to 0.5 (dark red) metres. Fourth column shows the corresponding coloured pointcloud.



(a) Dataset 1: Histogram of errors in meters (b) Dataset 2: Histogram of errors in meters (c) Dataset 3: Histogram of errors in meters

Fig. 5. Histograms with depth errors for each dataset

[4] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, May 2011.

[5] R. Ranftl, K. Bredies, and T. Pock, "Non-local total generalized variation for optical flow estimation," in *Computer Vision—ECCV 2014*. Springer, 2014, pp. 439–454.

[6] A. Concha, W. Hussain, L. Montano, and J. Civera, "Manhattan and piecewise-planar constraints for dense monocular mapping," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[7] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005. [Online]. Available: <http://dx.doi.org/10.1007/s10107-004-0552-5>

[8] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison, "Real-time camera tracking: when is high frame-rate best?" in *Proceedings of the 12th European conference on Computer Vision - Volume Part VII*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 222–235.

[9] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014.

[10] R. T. Rockafellar, *Convex Analysis*. Princeton, New Jersey: Princeton University Press, 1970.

[11] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM J. Img. Sci.*, vol. 3, no. 3, pp. 492–526, Sep. 2010.