Noname manuscript No.
(will be inserted by the editor)

Incorporating Scene Priors to Dense Monocular Mapping

Alejo Concha · Wajahat Hussain · Luis Montano · Javier Civera

Received: date / Accepted: date

Abstract This paper presents a dense monocular mapping algorithm that improves the accuracy of the state-of-the-art variational and multiview stereo methods by incorporating scene priors into its formulation. Most of the improvement of our proposal is in low-textured image regions and for low-parallax camera motions; two typical failure cases of multiview mapping.

The specific priors we model are the planarity of homogeneous color regions, the repeating geometric primitives of the scene –that can be learned from data– and the Manhattan structure of indoor rooms. We evaluate the performance of our method in our own sequences and in the publicly available NYU dataset, emphasizing its strengths and weaknesses in different cases.

Keywords Monocular SLAM · 3D Reconstruction · Structure from Motion

1 Introduction

Estimating a 3D reconstruction of a scene from 2D images has been one of the most studied topics in the computer vision community for the last four decades. As a result the geometric models for single and multiple views are currently well-known [19]. The topic also has a key importance for robotics, as robots need accurate models of their environment in order to interact safely with it. The sequential 3D estimation of the scene and the camera pose is usually known in the robotics community as visual SLAM, the latter acronym standing for *Simultaneous Localization and Mapping.*

From a geometric point of view, we need at least two views to estimate the depth of a general scene. The standard 3D reconstruction pipeline starts from multiple views of a scene and uses the well-known geometric models to minimize an error related with the goodness of the estimation. The traditional approaches minimize the geometric reprojection error of a sparse set of salient points (e.g., [8,23,33]) while more recent ones use the photometric error [28,10]. These algorithms have two main limitations that are rarely mentioned in the literature, failing in the cases of low-texture scenes and low-parallax camera motions. Both cases are likely to appear in indoor and man-made scenes.

Although single-view reconstruction is an ill-posed problem, meaning that in general depth cannot be estimated from one view,

Address: C/ María de Luna 1, Ada Byron Building, 50018, Zaragoza, Spain
{alejocb,hussain,montano,jcivera}@unizar.es

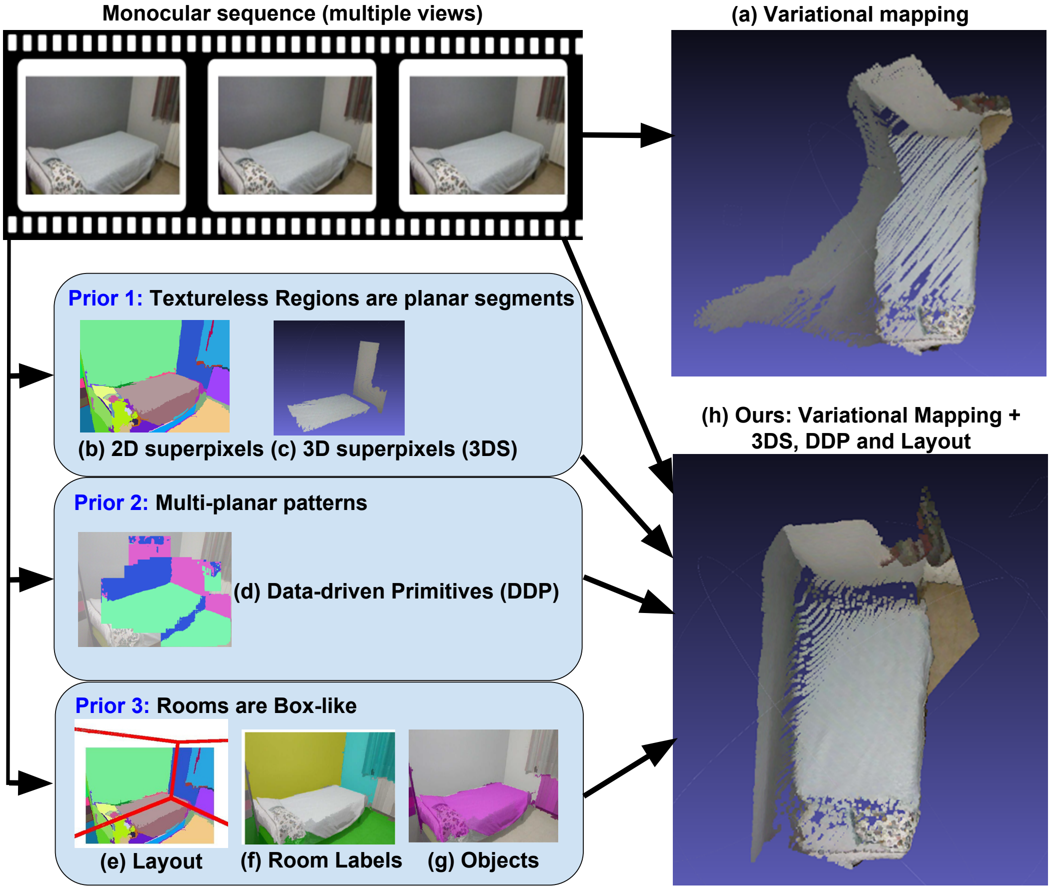

Monocular sequence (multiple views)
(a) Variational mapping
Prior 1: Textureless Regions are planar segments
(b) 2D superpixels
(c) 3D superpixels (3DS)
Prior 2: Multi-planar patterns
(d) Data-driven Primitives (DDP)
Prior 3: Rooms are Box-like
(e) Layout
(f) Room Labels
(g) Objects
(h) Ours: Variational Mapping + 3DS, DDP and Layout

Fig. 1 Incorporating scene priors to dense monocular mapping. (a). Variational mapping fails in textureless regions (top view). Notice for example the large errors in the walls. We use the following information to fix this error. **Prior 1**: Textureless regions are planar segments. We segment the image into superpixels (b) and triangulate them from multiple views (c). **Prior 2**: Man-made scene entities have repeating patterns that can be learned from RGB-D data. (d) shows the detections of such data-driven primitives, capturing the three normals of the scene. **Prior 3**: Indoor scenes are box-like. We fit a box to a sparse reconstruction (e). Given the room layout, we classify the image into the room geometric parts walls-floor-ceiling (f) and clutter (g). This gives us the prior depth and shape for the pixels classified as room geometric parts in (f). (h) shows how the 3D reconstruction is improved when the three scene priors are used.

there are solutions based on exploiting non-geometric cues and assumptions on the scene. For example, [35] creates a piecewise planar reconstruction with user interaction. [22], also using planar assumptions, is able to reconstruct outdoor scenes. [31,9] predict the depth from a single image by learning models from training data. Single-view reconstruction has been proposed for robot navigation and planning [26,32], but its accuracy is usually lower than multiview techniques and might fail catastrophically if the underlying assumptions are not met or the current image is far from the training set.

In this paper we propose the combination of state-of-the-art dense monocular SLAM algorithms (specifically we take [28] as our baseline) with higher-level features, data-driven and scene understanding cues to address the failure cases of low-texture scenes and low-parallax motions. We use 3D superpixels (3DS) [3] to model areas of homogeneous color, data-driven 3D primitives (DDP) to predict the depth of repeating scene patterns from a single view [13] and layout estimation and classification [20] to predict the depth of the walls and ceiling, usually textureless. Our experimental results show that our approach outperforms our baseline [28] in all the cases. Through several sequences, we illustrates the weaknesses and strengths of each of our depth cues.

See figure 1 for an overview of our system. 1(a) shows the 3D reconstruction of a state-of-the-art dense SLAM method in a bedroom scene. Notice the errors in the walls. Observe the scene priors; 3D superpixels (3DS) in (c), data-driven primitives (DDP) in (d) and Layout and room labels in (e) and (f). (h) shows the improved reconstruction.

This paper builds on the previous work [6]. The specific contributions of this paper are

- The evaluation of a new single-view depth prior based on learning geometric primitives from training data.
- The fusion of the three priors. Notice that [6] just evaluated two of the priors separately.
- An extended experimental evaluation of the proposed algorithm, including several sequences from the publicly available NYU dataset.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 gives the details of our proposal. Section 4 presents the experimental results and section 5 concludes.

2 Related Work

2.1 Dense Monocular Mapping

Real-time and dense 3D reconstructions of small-size environments from monocular sequences were first achieved in [17,28,34]. The problem is formulated as the minimization of an energy composed of a photometric and a regularization term; the first one modeling the photometric consistency of corresponding pixels and the second one the smoothness of regions with low image gradients. A typical limitation of standard regularizers based on the Total Variation or the Huber norm is that they have high errors in large low-textured image regions. [10] estimates the depth only for high-gradient pixels, producing semidense maps. In contrast, our proposal produces fully dense maps. [30] uses a non-local regularizer, able to propagate information from distant pixels and obtain more accurate reconstructions. Instead of relying in the regularizer, our proposal introduces new features (3D superpixels [3]), 3D primitives learned from data and floor-ceiling-walls-clutter classification to the formulation. Our proposal improves over the state of the art in the case of textureless regions. But it also improves in the low-parallax case, as our two latest cues use single view –zero-parallax– information.

2.2 Data-Driven Depth Cues

There are several works that use machine learning and high-level cues to improve multiview reconstructions. [1] jointly optimize 3D objects and sparse keypoints achieving a better performance in both tasks than the performance achieved optimizing them separately. [29] detects patches based on gradients in the images and looks for them in a RGB-D dataset to infer depth information and use it to fill low texture areas in keypoint-based Structure from Motion. Differently from them we estimate *fully dense* 3D reconstructions. [2] and [7] use object constraints to improve 3D dense reconstructions. Our approach aims to reconstruct scenes instead of objects.

2.3 Manhattan And Piece-Wise Planar Models

[14,16,37] used the Manhattan assumption to fill textureless gaps in sparse 3D reconstructions. [25,3,4] and [12,36] have used superpixels and indoor scene understanding respectively to fill textureless gaps in sparse 3D reconstructions. Our contribution is to fuse the previously mentioned cues and a new one –data-driven primitives– in a dense variational formulation. Our main advantages over them are the estimation of pixelwise reconstructions –the previously referred ones are not fully dense.

3 Dense Mapping Using Scene Priors

3.1 Problem Formulation

Our aim is to estimate the inverse depth $\boldsymbol{\rho}(\mathbf{u})$ for every pixel $\mathbf{u}$ of a reference keyframe $\mathbf{I}_r$ using a set of overlapping views $\{\mathbf{I}_1, \ldots, \mathbf{I}_j, \ldots\}$. In order to do that we minimize a global energy function $E_{\boldsymbol{\rho}}$; which is the weighted sum of a photometric error data term $\mathbf{C}(\mathbf{u}, \boldsymbol{\rho}(\mathbf{u}))$, a regularization term $\mathbf{R}(\mathbf{u}, \boldsymbol{\rho}(\mathbf{u}))$ and our newly proposed term which is a summation of the three scene priors $\boldsymbol{\rho}_1$, $\boldsymbol{\rho}_2$ and $\boldsymbol{\rho}_3$

$$E_{\boldsymbol{\rho}} = \int (\lambda_0 \mathbf{C}(\mathbf{u}, \boldsymbol{\rho}(\mathbf{u})) + \mathbf{R}(\mathbf{u}, \boldsymbol{\rho}(\mathbf{u})) + \\ + \sum_{\pi=1}^{3} \frac{\lambda_\pi}{2} \mathbf{P}(\mathbf{u}, \boldsymbol{\rho}(\mathbf{u}), \boldsymbol{\rho}_\pi(\mathbf{u})) \partial \mathbf{u} \tag{1}$$

λ_0 and λ_π are the weighting factor that account for the relative importance of the energy terms.

3.2 The Scene Priors

To extract our three scene priors we need two preprocessing steps. We extract first a set of salient points $\mathbf{u}^* \in \mathbf{u}$ in every keyframe of the sequence, compute correspondences and estimate the salient points' 3D positions which we defined as $\mathbf{p} = \left(\mathbf{p}_1^\top \ \ldots \ \mathbf{p}_i^\top \ \ldots \ \mathbf{p}_n^\top\right)^\top$ and camera poses $\mathbf{c} = \left(\mathbf{c}_1^\top \ \ldots \ \mathbf{c}_r^\top \ \ldots \ \mathbf{c}_j^\top \ \ldots \ \mathbf{c}_m^\top\right)^\top$ using a standard Bundle Adjustment optimization [33].

In the second preprocessing step, we segment every reference keyframe $\mathbf{I}_r$ into a set of superpixels $\mathcal{S}_r = \{s_1^r, \ldots, s_l^r, \ldots, s_t^r\}$ using the algorithm by Felzenszwalb et al. [11].

*3.2.1 3D Superpixels (*3DS*)*

We assume that the superpixels $\mathcal{S}_r = \{s_1^r, \ldots, s_l^r, \ldots, s_t^r\}$ correspond to approximately planar areas in the scene. We will estimate their 3D parameters using [3], which we will summarize here for completeness.

We can estimate the geometric parameters $\mathbf{\Pi} = \left(\pi_1^\top \ \ldots \ \pi_k^\top \ \ldots \ \pi_q^\top\right)^\top$ for the q planar superpixels $\{s_1, \ldots, s_k, \ldots, s_q\}$ that were matched in two or more keyframes with the following optimization

$$\hat{\mathbf{\Pi}} = \arg\min_{\mathbf{\Pi}} \sum_{r=1}^{m} \sum_{k=1}^{q} F(\epsilon_{s_k}^r) \ . \tag{2}$$

$\epsilon_{s_k}^r = \mathbf{u}_{s_k}^r - \mathbf{h}\left(\mathbf{u}_{s_k^h}^j, \pi_k^h, \mathbf{c}_r, \mathbf{c}_j\right)$ stands for the reprojection error of the superpixel s_k *contours* in the keyframe $\mathbf{I}_r$. As we are approximating the superpixels by planar surfaces, $\mathbf{h}$ stands for a homography model. We use a robust function of the error $F(*)$ to avoid the influence of outliers. Superpixels π_k are parametrized by its plane normal $\mathbf{n}_k$ and distance to the origin d_k.

The superpixel correspondences between several views are computed as follows. We first search for pairwise correspondences between two keyframes $\mathbf{I}_r$ and $\mathbf{I}_j$ using a Monte Carlo approach. For every superpixel $\mathbf{s}_k$ in $\mathbf{I}_r$ we create several plane hypotheses π_k^h. The plane hypothesis are ranked according to the reprojection error of the superpixel contours in image $\mathbf{I}_j$

$$\epsilon_{s_k^h} = ||\mathbf{u}_{s_k^h}^j - \mathbf{h}\left(\mathbf{u}_{s_k^h}^r, \pi_k^h, \mathbf{c}_r, \mathbf{c}_j\right)|| \tag{3}$$

The planar superpixel hypotheses π_k^h with the smallest error $\epsilon_{s_k^h}$ are taken as the initial seed for the optimization of equation 2.

The scene prior inverse depth $\boldsymbol{\rho}_1(\mathbf{u})$ for every pixel $\mathbf{u} \in s_k$ is computed as the intersection of its backprojected ray and the plane π_k

$$\boldsymbol{\rho}_1(\mathbf{u}) = || - \frac{\mathbf{u}\mathbf{K}_r^{-1}\mathbf{R}_r\mathbf{n}_k}{d_k\mathbf{K}_r^{-1}\mathbf{u}}|| \ . \quad (4)$$

Where $\mathbf{R}_r$ is the rotation matrix of the keyframe $\mathbf{I}_r$ and $\mathbf{K}_r$ is its internal calibration matrix.

*3.2.2 Data-Driven Primitives (*DDP*)*

A data-driven primitive is a repetitive and distinctive image gradient pattern with an associated 3D pattern. The models for such patterns can be learned from RGB-D training data. At test time, and from a single view, the gradient patterns can be detected and their depth can be predicted. Imagine, for example, the case of a room corner. It is a primitive that appears frequently indoors, it has a clear 3D pattern and several associated image patterns depending on the viewpoint.

Specifically, we use the approach of [13]. Each primitive is defined by $< \mathbf{w}, \mathbf{N}, \mathbf{y} >$; where $\mathbf{w}$ is the weight of an *SVM* classifier, $\mathbf{N} = \{\mathbf{n}(\mathbf{u})\}$ is the set of normals for each pixel $\mathbf{u}$ of the primitive patch, and $\mathbf{y} = \{0, 1\}^m$ is an indicator vector where $y_i = 1$ if the training patch $\mathbf{x}_i$ is an instance of such primitive. Each patch has a geometric component $\mathbf{x}_i^G$ and an appearance component $\mathbf{x}_i^A$ (HOG). In order to build the *SVM* classifiers $\mathbf{w}$ the following objective function is minimized on m training images

$$\arg\min_{\mathbf{y},\mathbf{w}} R(\mathbf{w}) + \sum_{i=1}^{m} \left(\Delta(\mathbf{N}, \mathbf{x}_i^G) + c_2 L(\mathbf{w}, \mathbf{x}_i^A, y_i) \right) \quad (5)$$

Where R is the classifier regularizer, each c_i trades off between terms, and L is the loss function. Notice that the above classifiers will provide a set of sparse detections of some geometric primitives in the test images. Dense results can be achieved by the propagation of these sparse detections to the entire image. But we have observed that such propagation might be innacurate if only a small number of primitives is detected. In order to keep the geometric primitives as accurate as possible, we only consider the sparse detections.

Similarly to section 3.2.1 we extract superpixels and assume that they correspond to approximately planar areas in the scene. For every superpixel π_k its plane normal $\mathbf{n}_k$ and distance to the origin d_k are estimated. For each superpixel, the common normal direction is the most voted one from the geometric primitives. The distance d_k is estimated using the approach of section 3.2.3; and the inverse depth prior $\boldsymbol{\rho}_2(\mathbf{u})$ for every pixel $\mathbf{u} \in s_k$ is computed as the intersection of its backprojected ray and the plane π_k (equation 4).

3.2.3 Layout

One of the goals of indoor scene understanding is the estimation of the rough geometry of a room –its layout– and the classification of every image pixel $\mathbf{u}$ into the wall, floor, ceiling or clutter classes. In this paper we basically use the algorithm of [20] and extend it to a multiview case. For an overview of the layout and the labelling algorithm see figure 2.

The main assumption is that we are in a cuboid room. The geometric model of the room layout $\mathcal{L}$ will be composed of six planes $\mathcal{L} = \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6\}$. Every plane π_k will be parametrized by its plane normal $\mathbf{n}_k$ and distance to the origin d_k. We first estimate the plane normals $\mathbf{n}_k$ by extracting the vanishing points $\mathbf{v}_k^r$ from the dominant directions of the room in every keyframe $\mathbf{I}_r$ [24]. These vanishing points are estimated by clustering the detected 2D line segments in the keyframe in three dominant clusters. Figure 2 b shows the vanishing points as red, green and blue circles. We backproject them to the 3D world $\mathbf{V}_k^r = \mathbf{K}_r^{-1}\mathbf{v}_k^r$ ($\mathbf{K}_r$ standing for the calibration matrix), group them into three clusters, and take their centroids.

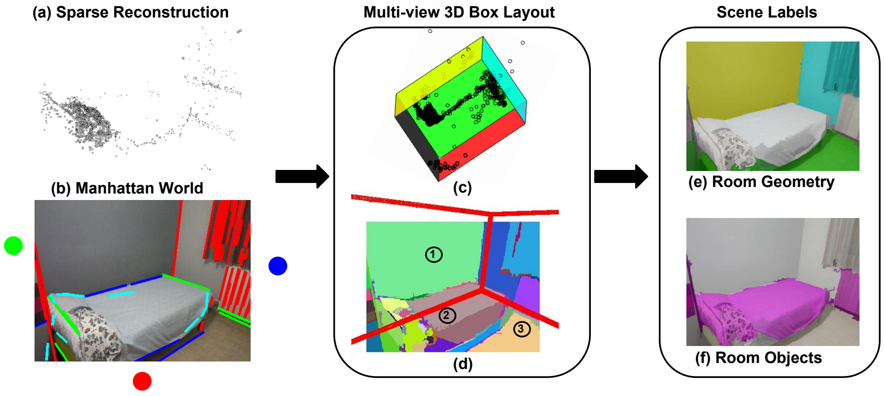


Fig. 2 Overview of the layout and the labelling algorithm. See section 3.2.3 for details.

In order to estimate the room layout box, we will fit planes to the sparse reconstruction $\mathbf{p} = \begin{pmatrix} \mathbf{p}_1^\top & \ldots & \mathbf{p}_i^\top & \ldots & \mathbf{p}_n^\top \end{pmatrix}^\top$ of figure 2. For this plane fitting, we start from the 3 dominant orientations of the room; the Manhattan directions provided by the vanishing points. For each orientation, we hypothesize N_{hyp} planes at different distances. Specifically, $N_{hyp} = 25$ in our experiments. A plane hypothesis is considered valid if it is supported by a minimum number of points (6 in our experiments). A point supports the hypothesis if it is within a predefined threshold. Finally, out of the winning planes, we select 6 extremal planes consisting the 3D box layout (figure 2 c)

Next, leveraging this 3D box layout, we label every superpixel from the segmentation $\mathcal{S}_r = \{s_1^r, \ldots, s_l^r, \ldots, s_t^r\}$ into 4 different classes $\{W, F, C, Cl\}$ –wall, floor, ceiling and clutter respectively. See [21] for details on the superpixel features and the classification algorithm. One of the most informative features for this classification is the interposition feature. The superpixels belonging to the room geometry must be totally contained in one of the projected box polygons. The superpixels belonging to the object clutter can cross the boundary between two polygons of the project layout box. For example, in figure 2 d, superpixels numbered 1 and 3 are totally contained in the wall and the floor polygon. Hence, they get the room geometry labels (figure 2 e). The superpixel numbered 2 is crossing the red line of the projected box layout. Only 3D objects have this physical property and hence it is labelled as clutter (figure 2 f). For more details see [20]. Notice that this method only tells us where the objects are but it does not give us the orientation nor the depth prior for the clutter (object) region. Therefore, we will not constraint the depth of the pixels $\mathbf{u} \in Cl$ that are labeled as clutter. For the rest of the pixels $\mathbf{u} \in \{W, F, C\}$ we will compute an *a priori* inverse depth $\boldsymbol{\rho}_3(\mathbf{u})$ from the intersection between the backprojected ray $\mathbf{K}_r^{-1}\mathbf{u}$ and the layout plane $\pi_k \in \mathcal{L}$ where it has been classified using equation 4.

3.3 The Photometric Cost $(\mathbf{C}(\mathbf{u}, \boldsymbol{\rho}(\mathbf{u})))$.

As in [28], our photometric error is based on color difference between the reference image and the set of short-baseline images. Every pixel $\mathbf{u}$ of the reference image $\mathbf{I}_r$ is first backprojected at an inverse distance $\boldsymbol{\rho}$ and projected again in every close image $\mathbf{I}_j$.

$$\mathbf{u}^j = \mathbf{T}_{rj}(\mathbf{u}, \boldsymbol{\rho}) = \mathbf{K}\mathbf{R}^\top \left(\begin{pmatrix} \frac{\mathbf{K}^{-1}\mathbf{u}}{||\mathbf{K}^{-1}\mathbf{u}||} \\ \boldsymbol{\rho} \end{pmatrix} - \mathbf{t} \right) \tag{6}$$

Where $\mathbf{T}$,$\mathbf{R}$ and $\mathbf{t}$ and respectively the relative transformation, rotation and translation between keyframe r and frame j. The photometric error is the summation of the color error between every pixel in the reference image and its corresponding in every other image at an hypothesized inverse distance $\boldsymbol{\rho}$.

$$\mathbf{C}(\mathbf{u},\boldsymbol{\rho}(\mathbf{u})) = \frac{1}{|I_s|}\sum_{j=1,j\neq r}^{m} f\left(\epsilon(\mathbf{I}_j,\mathbf{I}_r,\mathbf{u},\boldsymbol{\rho})\right) \quad (7)$$

$$\epsilon(\mathbf{I}_j,\mathbf{I}_r,\mathbf{u},\boldsymbol{\rho}) = \mathbf{I}_r(\mathbf{u}) - \mathbf{I}_j(\mathbf{T}_{rj}(\mathbf{u},\boldsymbol{\rho})) \quad (8)$$

Where I_s is the number of images used in the reconstruction, used for normalization of the photometric term. Differently from [6] we use a robust cost function –Tukey's cost function f– in the photometric term instead of L_1 norm, which improves the accuracy of the reconstruction in depth discontinuities due to the influence of outliers in occlusions [5].

3.4 The Gradient Regularizer ($\mathbf{R}(\mathbf{u},\boldsymbol{\rho}(\mathbf{u}))$).

The gradient regularizer is the Huber norm of the weighted gradient of the inverse depth map $||\nabla\boldsymbol{\rho}(\mathbf{u})||_\epsilon$

$$\mathbf{R}(\mathbf{u},\boldsymbol{\rho}(\mathbf{u})) = \mathbf{g}(\mathbf{u})||\nabla\boldsymbol{\rho}(\mathbf{u})||_\epsilon \quad (9)$$

Depth discontinuities often coincides with contours. $\mathbf{g}(\mathbf{u})$ is a per-pixel weight that decreases the regularization strength for high-gradient pixels.

$$g(\mathbf{u}) = e^{-\alpha||\nabla\mathbf{I}_r(\mathbf{u})||_2} \quad (10)$$

Where α is a constant.

3.5 The terms associated with the scene priors ($\mathbf{P}(\mathbf{u},\boldsymbol{\rho}(\mathbf{u}),\boldsymbol{\rho}_\pi(\mathbf{u}))$).

The scene prior terms model the distance from every point to its estimated planar prior (or priors) $\boldsymbol{\rho}_\pi$ detailed in section 3.2. Differently from [6] we use iterative reweighted least squares to be robust against outliers [5]. This is of key importance to deal with classification or segmentation errors. In those cases the cost function of the error should saturate for large values and have a limited influence on the solution.

$$\mathbf{P}(\mathbf{u},\boldsymbol{\rho}(\mathbf{u}),\boldsymbol{\rho}_\pi(\mathbf{u})) = \mathbf{w}_\pi\left(\boldsymbol{\rho}(\mathbf{u}) - \boldsymbol{\rho}_\pi(\mathbf{u})\right)^2 \quad (11)$$

$\mathbf{w}_\pi$ is the Tukey's cost function weight. In the areas of the image where we do not have a planar constraint (areas classified as clutter in the Manhattan layout, small and textured superpixels and areas where we did not detect any geometric primitive) we set $\lambda_\pi = 0$. We set the lambda of 3D superpixels $\lambda_1 = 10$ and we set a smaller lambda for the other two priors $\lambda_2 = 5$ and $\lambda_3 = 5$. The reason is that superpixels are based on multiview geometry whereas layout and geometric primitives use learning which is more prone to large errors.

3.6 Solution.

The energy is composed of two convex terms $g(\mathbf{u})||\nabla\boldsymbol{\rho}(\mathbf{u})||_\epsilon + \sum_{\pi=1}^{3}\frac{1}{2}\lambda_\pi w_\pi\left(\boldsymbol{\rho}(\mathbf{u}) - \boldsymbol{\rho}_\pi(\mathbf{u})\right)^2$ and a non-convex term $\lambda_0\mathbf{C}(\mathbf{u},\boldsymbol{\rho}(\mathbf{u}))$. The convex terms and the non-convex term are optimized differently and an auxiliary variable $\mathbf{a}$ is used to couple these two terms:

$$E_{\boldsymbol{\rho},\mathbf{a}} = \int \Big(\lambda_0\mathbf{C}(\mathbf{u},\mathbf{a}(\mathbf{u})) + \mathbf{g}(\mathbf{u})||\nabla\boldsymbol{\rho}(\mathbf{u})||_\epsilon + \sum_{\pi=1}^{3}\frac{1}{2}\lambda_\pi\mathbf{w}_\pi\left(\boldsymbol{\rho}(\mathbf{u}) - \boldsymbol{\rho}_\pi(\mathbf{u})\right)^2 + \frac{1}{2\theta}(\boldsymbol{\rho}(\mathbf{u}) - \mathbf{a}(\mathbf{u}))^2\Big)\partial\mathbf{u} \quad (12)$$

The coupling term $\frac{1}{2\theta}(\boldsymbol{\rho}(\mathbf{u}) - \mathbf{a}(\mathbf{u}))^2$ will enforce $\boldsymbol{\rho}$ and $\mathbf{a}$ to become the same as the parameter θ is initialized in 0.2 and it is derived to 0 iteratively. Therefore, equation 12 will result in the original energy 1.

The non-convex term will be optimized by sampling and the convex terms will be efficiently optimized using a primal-dual approach.

The convex terms are converted to their primal-dual formulation using the Legendre-Fenchel transformation (details and proofs in [18]). The energy in the equation 12 is then minimized as follows

$$\hat{\boldsymbol{\rho}} = \underset{\mathbf{q}, ||\mathbf{q}||_2 \leq 1}{\arg\max} \left\{ \underset{\boldsymbol{\rho}, \mathbf{a}}{\arg\min} \, E(\boldsymbol{\rho}, \mathbf{a}, \mathbf{q}) \right\} \tag{13}$$

$$\begin{aligned} E(\boldsymbol{\rho}, \mathbf{a}, \mathbf{q}) = \Big\{ \big\langle \mathbf{g}\mathbf{A}\boldsymbol{\rho}, \mathbf{q} \big\rangle - \delta_q(\mathbf{q}) - \frac{\epsilon}{2} ||\mathbf{q}||_2^2 + \\ \sum_{\pi=1}^{3} \frac{1}{2} \lambda_\pi \mathbf{w}_\pi \left(\boldsymbol{\rho} - \boldsymbol{\rho}_\pi \right)^2 + \frac{1}{2\theta} (\boldsymbol{\rho} - \mathbf{a})^2 + \\ \lambda_0 \mathbf{C}(\mathbf{a}) \Big\} \end{aligned} \tag{14}$$

Where $\mathbf{q}$ is the dual variable, $\mathbf{A}\boldsymbol{\rho}$ stands for the gradient of $\boldsymbol{\rho}$, ϵ is the threshold of the Huber norm which determines when L_1 or L_2 norm are used [28] and δ_q is an indicator function [18].

For the dual variable $\mathbf{q}$ the energy has to be maximized, therefore a gradient ascent step $\frac{\partial \mathbf{E}(\boldsymbol{\rho}, \mathbf{a}, q)}{\partial \mathbf{q}} = \nabla(q)$ is computed:

$$\frac{\partial E(\boldsymbol{\rho}, \mathbf{a}, \mathbf{q})}{\partial \mathbf{q}} = \mathbf{g}\mathbf{A}\boldsymbol{\rho} - \epsilon \mathbf{q} \tag{15}$$

Discretizing $\nabla(q) = \frac{\mathbf{q}^{(n+1)} - \mathbf{q}^n}{\sigma_q}$ and rearranging terms:

$$\frac{\mathbf{q}^{(n+1)} - \mathbf{q}^n}{\sigma_q} = \mathbf{g}\mathbf{A}\boldsymbol{\rho}^n - \epsilon \mathbf{q}^{(n+1)} \tag{16}$$

Where σ_q is the differentiation step.

$$\mathbf{q}^{(n+1)} = \left(\mathbf{q}^n + \sigma_q \mathbf{g}\mathbf{A}\boldsymbol{\rho}^n \right) / \left(1 + \sigma_q \epsilon \right) \tag{17}$$

$$\mathbf{q}^{(n+1)} = \mathbf{q}^{(n+1)} / \max(1, ||\mathbf{q}^{(n+1)}||_1) \tag{18}$$

In the case of the variable $\boldsymbol{\rho}$, the energy is minimized, therefore a gradient descent step $\frac{\partial \mathbf{E}(\boldsymbol{\rho}, \mathbf{a}, \mathbf{q})}{\partial \boldsymbol{\rho}} = \nabla(\boldsymbol{\rho})$ is computed. Using the divergence theorem $\frac{\partial \langle \mathbf{A}\boldsymbol{\rho}, \mathbf{q} \rangle}{\partial \boldsymbol{\rho}} = -div(\mathbf{q}) = \mathbf{A}^T \mathbf{q}$, where $\mathbf{A}^T \mathbf{q}$ forms the negative divergence of $\mathbf{q}$:

$$\begin{aligned} \frac{\partial E(\boldsymbol{\rho}, \mathbf{a}, q)}{\partial \boldsymbol{\rho}} = \mathbf{g}\mathbf{A}^T \mathbf{q} + \frac{1}{\theta} (\boldsymbol{\rho} - \mathbf{a}) + \\ \sum_{\pi=1}^{3} \lambda_\pi \mathbf{w}_\pi \left(\boldsymbol{\rho} - \boldsymbol{\rho}_\pi \right) \end{aligned} \tag{19}$$

Discretizing $\nabla(\boldsymbol{\rho}) = \frac{\boldsymbol{\rho}^{(n+1)} - \boldsymbol{\rho}^n}{\sigma_\rho}$and rearranging terms:

$$\begin{aligned} \frac{\boldsymbol{\rho}^{(n+1)} - \boldsymbol{\rho}^n}{\sigma_\rho} = -\mathbf{g}\mathbf{A}^T \mathbf{q}^{(n+1)} \\ - \frac{1}{\theta^n} (\boldsymbol{\rho}^{(n+1)} - \mathbf{a}^n) \\ - \sum_{\pi=1}^{3} \lambda_\pi \mathbf{w}_\pi \left(\boldsymbol{\rho}^{(n+1)} - \boldsymbol{\rho}_\pi \right) \end{aligned} \tag{20}$$

Where σ_ρ is the differentiation step.

$$\boldsymbol{\rho}^{(n+1)} = \frac{\left(\boldsymbol{\rho}^n + \sigma_\rho \left(-\mathbf{g}\mathbf{A}^T \mathbf{q}^{(n+1)} + \frac{\mathbf{a}^n}{\theta^n} + \sum_{\pi=1}^{3} \lambda_\pi \mathbf{w}_\pi \boldsymbol{\rho}_\pi \right) \right)}{(1 + \frac{\sigma_\rho}{\theta^n} + \sum_{\pi=1}^{3} \lambda_\pi \mathbf{w}_\pi \sigma_\rho)} \tag{21}$$

The remaining non-convex function is minimized using a point-wise search for each $\mathbf{a}$ in the range $\mathbf{a} = [\boldsymbol{\rho}_{min}, \boldsymbol{\rho}_{max}]$:

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\arg\min} \, E^{aux}(\boldsymbol{\rho}, \mathbf{a}) \tag{22}$$

$$E^{aux}(\boldsymbol{\rho}, \mathbf{a}) = \frac{1}{2\theta} (\boldsymbol{\rho} - \mathbf{a})^2 + \lambda_0 \mathbf{C}(\mathbf{a}))) \tag{23}$$

Finally, we use the acceleration of the non-convex solution recommended in [28] and also we achieve sub-sample accuracy by performing a single Newton step using numerical derivative in the discrete variable **a**:

$$\hat{\mathbf{a}}^{(n+1)} = \hat{\mathbf{a}}^{(n+1)} - \frac{\nabla E^{aux}}{\nabla^2 E^{aux}} \tag{24}$$

Equations 17, 18, 21, 22 and 24 are computed iteratively while $\theta^{(n+1)} = \theta^{n}(1 - 0.001 * n)$ is higher than 0.0001.

For the initialization of the iterative optimization we will use the photometric depth in the high-gradient image regions and the average of the depths of the scene priors for textureless areas. We have observed that this initialization has better convergence than a photometric one.

4 Experimental Results

We have evaluated different aspects of our proposal in indoor and outdoor sequences of small and middle-size scenes. For every indoor experiment we have a RGB-D sequence. We used the D channel as the ground truth depth for the scene and our algorithm used the RGB data. We used our own sequences and sequences from the public NYU dataset [27]. In both cases the camera used was the Microsoft Kinect. The outdoor experiments were recorded with a RGB camera and we only show qualitative results, due to the limitations of *RGB-D* sensors under direct sunlight.

We divided our results on two subsets. Section 4.1 presents results on low texture scenes. Section 4.2 presents results on low parallax camera motions using the sequences from the NYU dataset.

4.1 Low Texture Scenes

4.1.1 Indoors

We have evaluated the performance of 3D superpixels (3DS) as a prior for direct mapping with 5 indoor sequences (*Bookshelf*, *Desktop*, *Corner1*, *Corner2* and *Wall*). The experiments in this section deviate from the assumptions of the other two priors–layout and geometric primitives–, as most of them are close ups. We will only evaluate the improvement obtained using 3DS. 3DS is a more general prior than Layout and data-driven primitives (DDP), as it can be applied in any scene. DDP requires scenes similar to the training set and Layout requires a global view of the indoor scene. On the other hand, the triangulation of superpixels require a high-parallax camera motion while the other two perform reasonably even for the single-view case.

Sequence	**Mean Error [cm]**	
	DTAM	**Ours (3DS)**
Bookshelf (3DS)	2.9	2.7
Desktop (3DS)	4.4	2.9
Corner1 (3DS)	6.6	3.2
Corner2 (3DS)	18.5	13.7
Wall (3DS)	30.4	10.3
Lab (3DS)	28.2	10.2
Lab (Layout)		15.5
Lab (3DS + Layout)		10.5

Table 1 Mean of the estimated depth error for the standard *DTAM* and our approach using 3DS.

Figure 3 and table 1 show the qualitative and quantitative results for these experiments. The so-called –*Bookshelf* experiment is a clear textured scenario where the photometric term is already very informative and the reconstruction is quite accurate with standard dense mapping. But even in this case, 3DS improves the mean error 6%. In the other four sequences there are larger textureless areas and the gradient-based regularization produces larger errors. In these latest cases, 3DS improves the 3D reconstructions significantly.

We have used a larger sequence recorded in our *Lab* to compare 3DS and Layout. Find a qualitative summary of the results in figure 4, the 3D maps obtained in figure 5 and the quantitative results in table 1. 3DS and Layout mean errors are 10.2 cm and 15.5 cm respectively, both smaller than the *DTAM* baseline error (28.2 cm). 3DS outperforms Layout in this case because the sequence was recorded with large camera translations ?and hence high

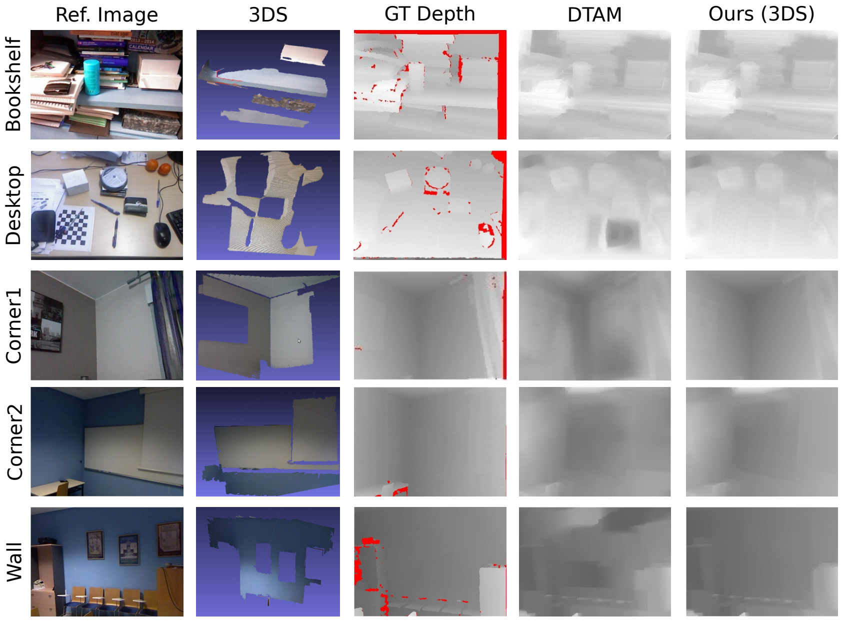


Fig. 3 Indoor experiments, high-parallax camera motion, close-ups. 1^{st} column: Reference image. 2^{nd} column: 3D superpixels. 3^{rd} column: ground truth depth –red stands for no-depth-data. 4^{th} column: *DTAM* depth. 5^{th} column: Ours, using 3DS. Notice how this latest column is visually closer to the ground truth than the *DTAM* one.

parallax. This is the best configuration for 3DS. You can observe a large error in the layout in the last row of figure 4. The red line standing for a corner is wrongly estimated at the middle of a wall. Our approach using Layout, in the last column, has a high depth error.

Notice in the last row of table 1 that the combination of 3DS and Layout is worse than 3DS alone. The reason is that the different energy terms in the optimization are weighted with the parameters λ_π, that we learn from training data.

Finally, we have used three more sequences (*Bedroom1*, *Bedroom2* and *Kitchen*) to further evaluate the performance of our algorithm in a high-parallax low-texture case, this time using the three scene priors and comparing against the baseline *DTAM* and also against the state-of-the-art batch approach *PMVS* [15]. See figure 7 and table 3 to observe the distribution of the errors in these experiments. Note that in the Bedroom2 and the Kitchen experiment the solution for standard DTAM is already quite accurate and we only slightly outperform it. For the case of Bedroom1 the baseline DTAM leads to big errors because of the large untextured wall. This error is fixed by Layout and DDP but the algorithm did not find a 3D Superpixel for the large wall, so the error is close to the DTAM baseline. Notice that we obtained competitive results in the comparison against PMVS. Note also in figure 11 that PMVS creates semidense maps and leaves holes in low textured areas, whereas we achieve fully dense reconstructions..

4.1.2 Outdoors

We have performed two outdoor experiments –in a building corner and a façade– to evaluate 3D superpixels in outdoor scenes. Figure 8 summarizes the results. Observe how in both cases the low texture walls are not planar for the *DTAM* baseline. 3D superpixels are able to improve the results and estimate the correct planar surfaces.

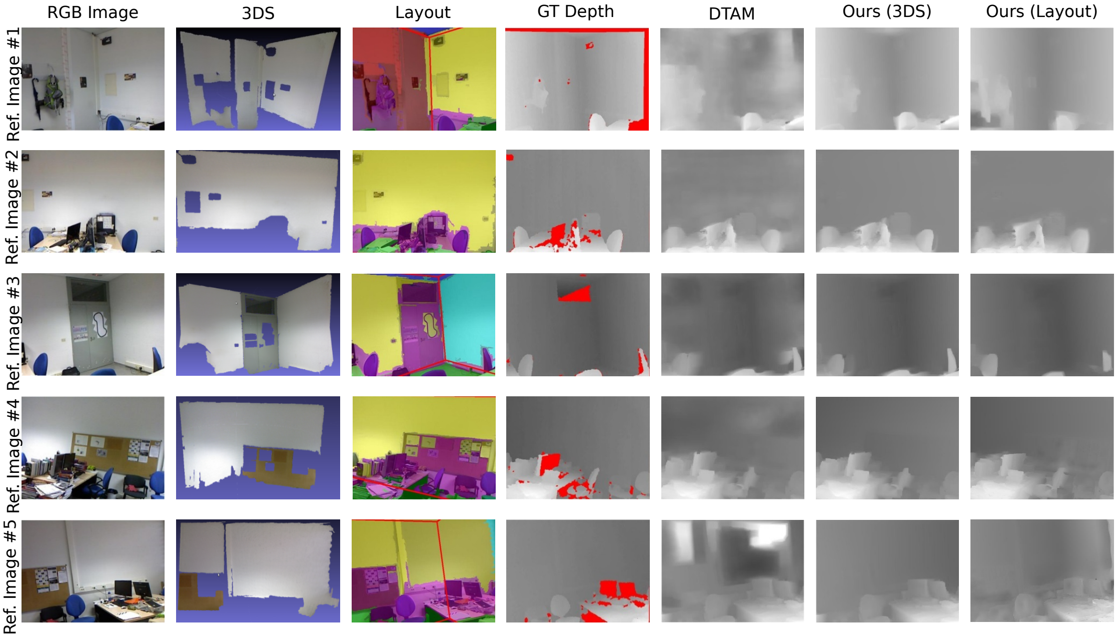
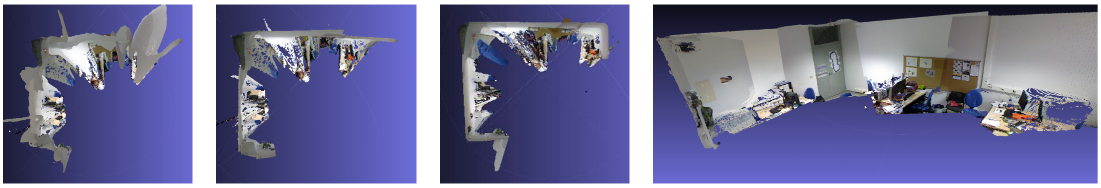


Fig. 4 Lab experiment. Each row shows the results for a reference image. 1^{st} column: *RGB* image. 2^{nd} column: 3D superpixels. 3^{rd} column: Room layout and labels. Red lines stand for the projected box. Magenta stands for clutter, green for floor and dark blue for ceiling. Other colors stand for walls. 4^{th} column: ground truth depth –red stands for no-depth-data. 5^{th}: column *DTAM* depth. 6^{th} column: Our approach, using 3DS. 7^{th} column: Our approach, using Layout. The improvement of the depth maps of *DTAM* with planarity constraints against the standard *DTAM* is visually noticeable.



(a) *DTAM*. Top view

(b) Ours (3DS). Top view

(c) Ours (Layout). Top view

(d) Ours (3DS). Side view

Fig. 5 3D maps for the *Lab* experiment. Notice the large *DTAM* errors in 5(a) and the more accurate reconstructions in 5(b) –using the layout– and 5(c) –using 3D superpixels. Notice the differences: 5(b) shows small misalignments, while 5(c) is globally consistent but with large errors in the objects and final parts of two walls due to wrong labels and layout errors. 5(d) shows a side view of *DTAM* using 3D superpixels. Quantitative results are in table 1.

4.2 Low Parallax Camera Motion

We have used the NYU dataset [27] to evaluate the performance of our algorithm in low-parallax camera motion sequences. The first thing to remark is that 3D superpixels will perform badly for this case. In our experience, in order to have an accurate estimation of the 3D superpixel the baseline has to be greater than 0.2 times the average depth of the keyframe. This constraint does not hold for the sequences tested in this dataset, so the results for 3DS

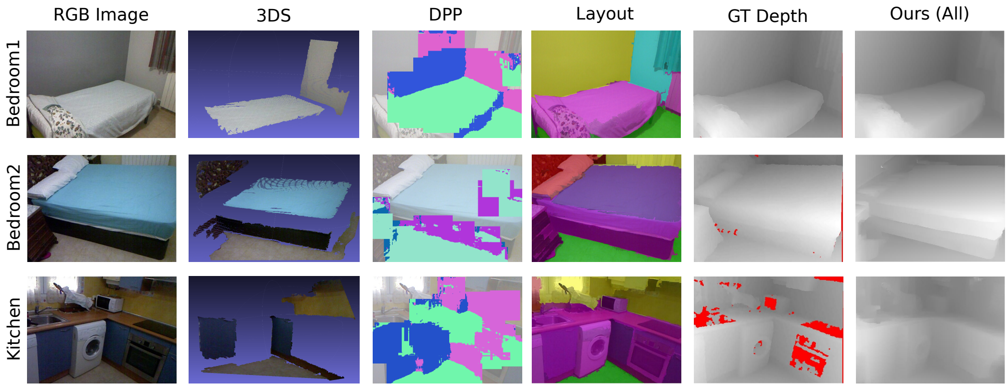
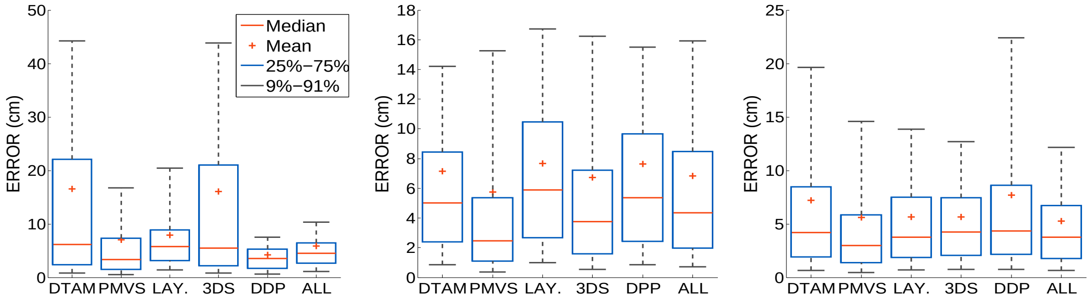


Fig. 6 Results from the *Bedroom1*, *Bedroom2* and *Kitchen* sequence.



(a) *Bedroom1*

(b) *Bedroom2*

(c) *Kitchen*

Fig. 7 Box and Whiskers plots showing the depth error distribution for the indoor high-parallax sequences.

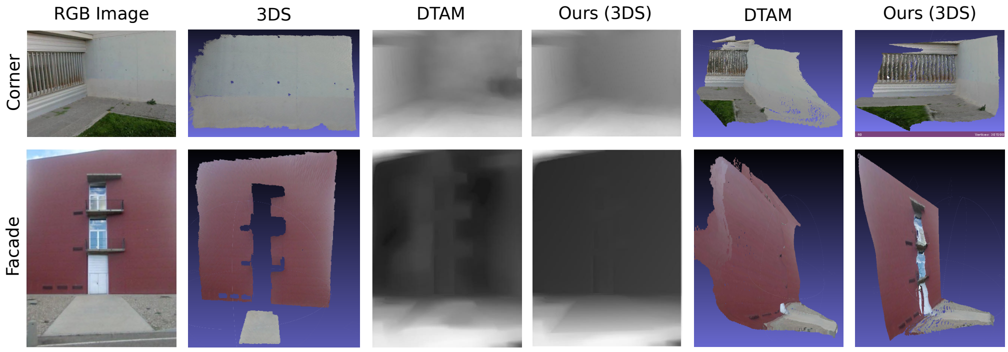


Fig. 8 Outdoor results, in a Corner and a Façade. The improvement of 3DS can be noticed visually.

are the same than the baseline *DTAM* and we only present results for DDP and Layout. As previously said, this is a clear limitation of 3DS –and in general of multiview geometry– and an advantage of DDP and Layout, that give reasonable results even in the single-view case.

We have performed 4 reconstructions of the NYU dataset, that we will denote as NYU #1, #2, #3 and #4 and that corresponds to the sequences *printer room 0001 rect* (#1 and #2), *bedroom 0106 rect* (#3) and *bedroom 0110 rect* (#4) of the dataset. Figure 9 shows the Box-

Sequence	**Mean Error[cm]**		
	DTAM	***PMVS* (%)**	**Ours**
Bedroom1 (3DS)	15.8	7.0 (18%)	15.0
Bedroom1 (DDS)			**4.2**
Bedroom1 (Lay.)			7.9
Bedroom1 (All)			5.9
Bedroom2 (3DS)	7.1	**5.7** (22%)	**6.7**
Bedroom2 (DDP)			7.6
Bedroom2 (Lay.)			7.7
Bedroom2 (All)			6.8
Kitchen (3DS)	7.2	5.5 (20%)	5.6
Kitchen (DDP)			7.7
Kitchen (Lay.)			5.7
Kitchen (All)			**5.2**

Table 2 Mean of the estimated depth error for *DTAM*, *PMVS* and ours in high-parallax low-texture sequences. (%) is the percentage of pixels reconstructed by *PMVS*, notice that *PMVS* only reconstruct high texture pixels.

and-Whiskers plot of the depth error in this sequences for the baselines *DTAM* and *PMVS* and our dense mapping algorithm using Layout, DDP and both. Notice first the huge error of *PMVS* compared with the rest of the approaches. The reason is, being a multiview stereo algorithm, it is very affected by low-parallax measurements. The magnitude of the error –one order of magnitude higher than the others– can be seen in table 3, that shows the mean error values. *DTAM* is less affected by the low parallax; but still the use of scene priors improves its accuracy. See the mean values in table 3.

Sequence	**Mean Error[cm]**		
	DTAM	***PMVS* (%)**	**Ours**
#1 (Lay.)	9.7	157.5 (3%)	10.4
#1 (DDP)			**7.9**
#1 (All)			9.0
#2 (Lay.)	21.2	43.8 (8%)	8.4
#2 (DDP)			9.2
#2 (All)			**7.6**
#3 (Lay.)	22.2	246.0 (2%)	**12.5**
#3 (DDP)			19.4
#3 (All)			14.5
#4 (Lay.)	42.3	288.4 (9%)	23.8
#4 (DDP)			39.1
#4 (All)			**20.9**

Table 3 Mean of the estimated depth error for *DTAM*, *PMVS* and ours in low-parallax sequences. (%) is the percentage of pixels reconstructed by *PMVS*, notice that *PMVS* only reconstruct high texture pixels.

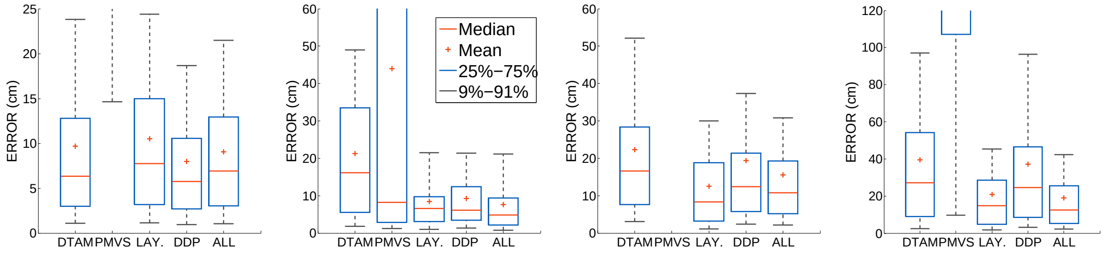
The performance of the different scene priors on these 4 NYU scenes can be better appreciated in figure 10. Observe that in the experiment *NYU #1* the Layout is wrongly labeled (some cupboards are labeled as walls). This is the reason for the Layout algorithm performing slightly worse than *DTAM* in this sequence (see the mean values in table 3). The labeling has also big errors in *NYU #4*, where part of the floor is labeled as clutter. But in this case the texture in the floor allows to reconstruct it more accurately than *DTAM*. In any case, this is precisely the limitation of DDP and Layout. As they rely on data-driven models, their accuracy can be low if the test image is very different than the training ones.

Finally, figure 11 shows the comparison of our approach against *PMVS* in our high-parallax sequences and the low-parallax NYU ones. Notice first in the high-parallax sequences that *PMVS* is a semidense approach that only reconstructs high-gradient pixels. Our approach has the fundamental advantage over *PMVS* of doing a full 3D reconstruction, as seen in the figure.

Secondly, observe the bad results of *PMVS* in the low-parallax sequences of figure 11. Our approach, leveraging the single-view cues given by the Data-Driven Primitives and the Layout of the room, is able to reconstruct the scene with high accuracy even if the geometric constraints are weak.

5 Conclusion

In this paper we have presented an algorithm that fuses several scene priors and depth cues in a dense mapping algorithm based on variational methods. Although the multiview geometric constraints stand out as the preferred ones for monocular map building, their results are poor in low-textured areas and for low-parallax motions. We show how the use of 1) Superpixels as mid-level features, 2) Data-Driven Primitives that appear frequently and can be discovered from training samples, and 3) the rough room Layout estimation and pixel



(a) NYU #1 (b) NYU #2 (c) NYU #3 (d) NYU #4

Fig. 9 Box and Whiskers plots showing the depth error distribution for 4 indoor low-parallax sequences of the NYU dataset.

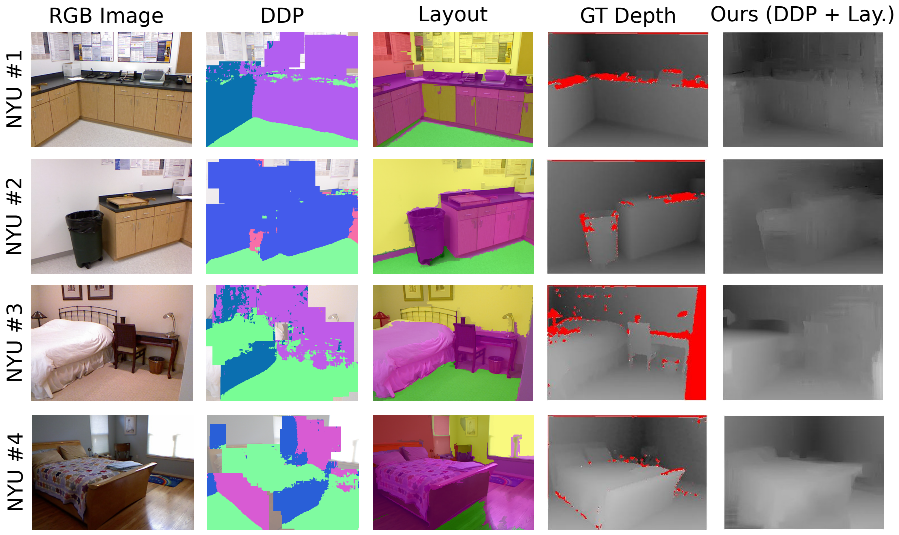


Fig. 10 Overview of the DDP and Layout results, the Ground Truth depth and our estimated depth in the NYU dataset sequences. We are able to estimate accurate reconstructions for these low-parallax sequences.

labeling can improve the 3D reconstructions in the two failure cases mentioned before.

Our experimental results show that 3D superpixels offer the highest accuracy, but they suffer from the multiview geometry limitations. Firstly, their accuracy decreases if the parallax is low. And secondly, superpixel matching can be difficult in certain cases. Their use as mid-level features is then recommended only with strict thresholds in the parallax angle and descriptor distances. We think that superpixels can be an excellent mid-level feature for mapping low texture regions if mid-baseline correspondences can be found.

Data-Driven Primitives and Layout estimation and labeling are techniques designed for the single-view case, hence being more robust to low-parallax motions. In this paper we use a multiview version of the second one for robustness, but it works reasonably well for single images. Both cues improve the reconstruction if the camera motion is small, and also in low-textured areas. The reason for the latest is 1) data-driven primitives capture nonlocal primitives and hence cover some textureless areas, and 2) the Layout is a global scene model. As their main limitation, their data-based nature makes them inaccurate as the im-

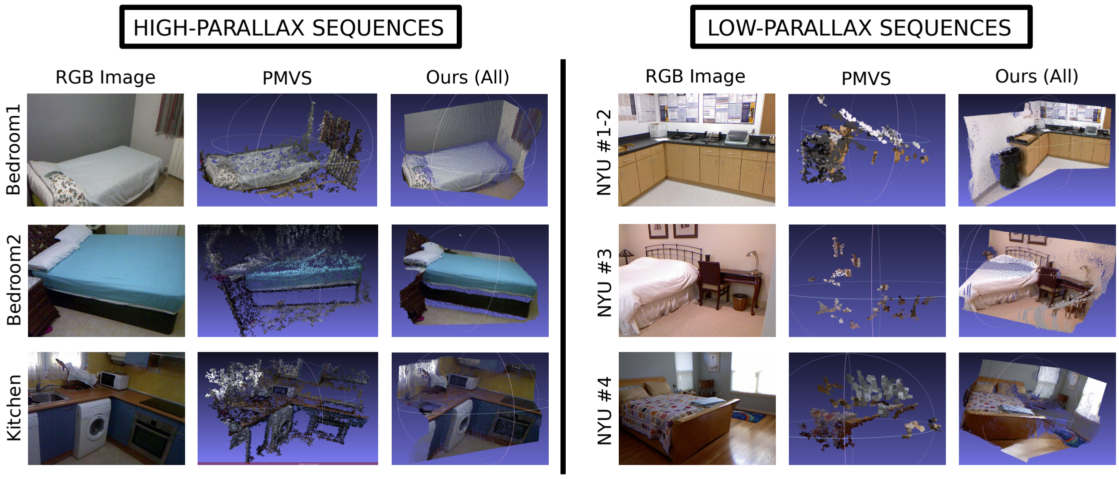


Fig. 11 Qualitative comparison of our approach against *PMVS* in our high-parallax sequences (left) and the NYU low-parallax sequences (right). Notice the sparsity of *PMVS* in textureless areas and our dense results. Also notice the bad 3D maps produced by *PMVS* in the low-parallax cases and how our algorithm produces reasonable results.

age differs from the training samples. In this case, more training data or more sophisticated learning techniques could alleviate this problem.

For future work, we would like to study the potential of this research for a robust and real-time implementation. Regarding robustness, our main concern is that data-driven techniques can give large errors that are difficult to predict. Regarding real-time, we are quite confident that the techniques we used are low-cost. [3] already demonstrated that 3D superpixels can be reconstructed in real-time. [12,36] estimated a multiview layout –without labeling the image– in real-time. Finally, although there is no experimental evidence of real-time for Data-Driven Primitives, it consists of HOG features extraction and *SVM* classification. Both algorithms require low computation.

Acknowledgments

This research was funded by the Spanish government with the projects DPI2012 -32168, DPI2012-32100 and IPT-2012-1309-430000. We would like to thank Marta Salas for her help with the NYU dataset.

References

1. Sid Yingze Bao and Silvio Savarese. Semantic structure from motion. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2025–2032. IEEE, 2011.
2. Y Bao, Manmohan Chandraker, Yuanqing Lin, and Silvio Savarese. Dense object reconstruction with semantic priors. In *26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
3. Alejo Concha and Javier Civera. Using superpixels in monocular SLAM. In *IEEE International Conference on Robotics and Automation*, Hong Kong, June 2014.
4. Alejo Concha and Javier Civera. DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence. In *IEEE/RSJ International Conference on Intelligent Systems and Robots*, Hamburg, Germany, September 2015.
5. Alejo Concha and Javier Civera. An evaluation of robust cost functions for RGB direct mapping. In *European Conference on Mobile Robotics (ECMR15)*, Lincoln, UK, September 2015.
6. Alejo Concha, Wajahat Hussain, Luis Montano, and Javier Civera. Manhattan and

piecewise-planar constraints for dense monocular mapping. In *Robotics:Science and Systems*, 2014.

7. Amaury Dame, Victor A Prisacariu, Carl Y Ren, and Ian Reid. Dense reconstruction using 3D object shape priors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1288–1295, 2013.
8. A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. 29(6), 2007.
9. David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014.
10. Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular slam. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014.
11. Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
12. Alex Flint, David Murray, and Ian Reid. Manhattan scene understanding using monocular, stereo, and 3D features. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2228–2235, 2011.
13. David F. Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3D primitives for single image understanding. In *ICCV*, 2013.
14. Y. Furukawa, B. Curless, S.M. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *Proc. Int. Conf. on Computer Vision*, pages 80–87, 2009.
15. Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.
16. David Gallup, J-M Frahm, and Marc Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1418–1425. IEEE, 2010.
17. Gottfried Graber, Thomas Pock, and Horst Bischof. Online 3d reconstruction using convex optimization. In *2011 IEEE International Conference on Computer Vision Workshops*, pages 708–711, 2011.
18. A. Handa, R. Newcombe, A. Angeli, and A. Davison. Applications of Legendre-Fenchel transformation to computer vision problems. In *Technical Report DTR11-7, Imperial College*, 2011.
19. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, 2004.
20. Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *Computer vision, 2009 IEEE 12th international conference on*, pages 1849–1856. IEEE, 2009.
21. D. Hoiem, A.A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.
22. Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 577–584. ACM, 2005.
23. G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.
24. Jana Košecká and Wei Zhang. Video compass. In *Computer VisionECCV 2002*, pages 476–490. Springer, 2006.
25. Branislav Mičušík and Jana Košecká. Multiview superpixel stereo in urban environments. *International journal of computer vision*, 89(1):106–119, 2010.
26. Bart Nabbe, Derek Hoiem, Alexei AA Efros, and Martial Hebert. Opportunistic use of vision to push back the path-planning horizon. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2388–2393. IEEE, 2006.
27. Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.
28. Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. DTAM: Dense tracking and mapping in real-time. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2320–2327, 2011.
29. Andrew Owens, Jianxiong Xiao, Antonio Torralba, and William Freeman. Shape anchors for data-driven multi-view reconstruction. In *2013 IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.
30. P. Piniés, L. M. Paz, and Paul Newman. Dense Mono Reconstruction: Living with the Pain of the Plain Plane. *Proc. of the 2015 IEEE International Conference on Robotics and Automation*, pages 5226–5231, 2015.
31. Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, pages 1161–1168, 2005.
32. Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Depth perception from a single still image. In *AAAI*, pages 1571–1576, 2008.

33. N. Snavely, S.M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.
34. Jan Stühmer, Stefan Gumhold, and Daniel Cremers. Real-time dense geometry from a handheld camera. In *Pattern Recognition*, pages 11–20. Springer, 2010.
35. Peter Sturm, Steve Maybank, et al. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *The 10th British Machine Vision Conference (BMVC'99)*, pages 265–274, 1999.
36. Grace Tsai, Changhai Xu, Jingen Liu, and Benjamin Kuipers. Real-time indoor scene understanding using bayesian filtering with motion cues. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 121–128. IEEE, 2011.
37. Carlos A Vanegas, Daniel G Aliaga, and Bedrich Benes. Building reconstruction using manhattan-world grammars. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 358–365. IEEE, 2010.