DPPTAM: Dense Piecewise Planar Tracking and Mapping from a Monocular Sequence

Alejo Concha I3A
Universidad
de Zaragoza, Spain
alejocb@unizar.es

Javier Civera I3A
Universidad
de Zaragoza, Spain
jcivera@unizar.es

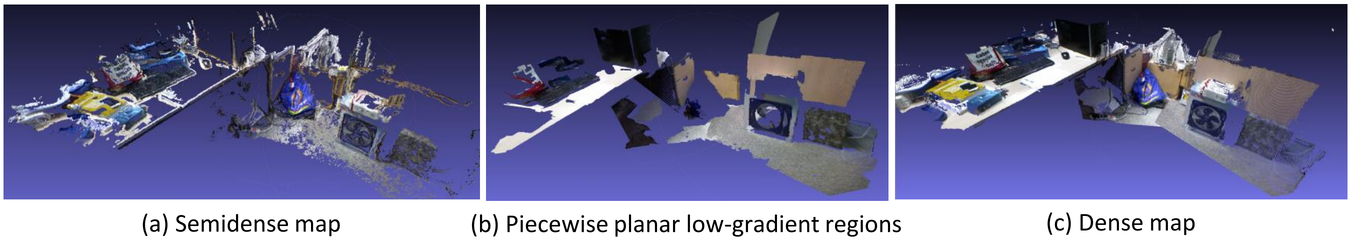


Fig. 1: Illustrative results for our approach. (a) shows a semidense map estimated from a monocular sequence in a desktop environment. Notice that it only contains high-gradient areas. (b) shows a map of the low-gradient areas, under the planar assumption. (c) is the dense map of the scene, built from the two previous contributions.

***Abstract*— This paper proposes a direct monocular SLAM algorithm that estimates a dense reconstruction of a scene in real-time on a CPU. Highly textured image areas are mapped using standard direct mapping techniques [1], that minimize the photometric error across different views. We make the assumption that homogeneous-color regions belong to approximately planar areas. Our contribution is a new algorithm for the estimation of such planar areas, based on the information of a superpixel segmentation and the semidense map from highly textured areas.**

We compare our approach against several alternatives using the public TUM dataset [2] and additional live experiments with a hand-held camera. We demonstrate that our proposal for piecewise planar monocular SLAM is faster, more accurate and more robust than the piecewise planar baseline [3]. In addition, our experimental results show how the depth regularization of monocular maps can damage its accuracy, being the piecewise planar assumption a reasonable option in indoor scenarios.

I. Introduction

SLAM, standing for Simultaneous Localization and Mapping, aims to estimate the pose of a mobile sensor and a map of its surrounding environment in real-time. Monocular SLAM, relying on a single camera as the only input, has become a particularly valuable research topic during the last decade. The small size, low weight and low consumption of a monocular camera make it an excellent sensor for autonomous robots –Micro Aerial Vehicles (MAVs) [4], driverless cars [5] or underwater vehicles [6]–, augmented reality demos [7] and 3D scanners [8].

One of the hardest challenges in monocular SLAM is the estimation of a *fully dense* map of the imaged scene. A monocular camera is a bearing-only sensor; and its pixel depths are estimated from their correspondences in other views. These correspondences are found by comparing the photometric patterns in the candidate pixel neighborhoods. As a result pixels in textureless areas cannot be reliably matched across views and accurate 3D reconstructions are usually limited to areas of high image gradients.

In this paper we follow the line initiated in [3], [9] and model the environment with 3D points for high-gradient areas and 3D planes for low-gradient areas. The assumption made is that image areas with low photometric gradients are mostly planar; which is met in most indoors and man-made scenes. Low-gradient image areas are segmented using superpixels [10]. Our experiments, using standard public datasets, show that this assumption allows to estimate dense and accurate indoor maps using a monocular camera. See an illustrative result of our approach in figure 1.

The contribution of this paper is a new initialization scheme for the piecewise planar areas that is more efficient, robust and accurate than the baseline used in [3]. We compare several monocular SLAM alternatives including semidense, piecewise planar and dense; and discuss their performance. We show that our piecewise planar monocular SLAM improves the accuracy and density of a semidense algorithm with a lower computational cost than a dense one.

The rest of the paper is organized as follows. Next section describes the related work. Section III gives an overview of our system. Section IV details the direct-based methods that we use to estimate semidense maps and track the camera pose. Section V details our proposal for piecewise planar reconstructions. Finally, section VI shows our experimental results and section VII concludes.

II. Related work

A. Direct SLAM

Direct visual SLAM [1] refers to a class of SLAM algorithms, recently appeared, that uses the raw pixel intensity values to estimate a map of the environment and the camera motion. This is in contrast to the more traditional feature-based methods [11], [12] that used the image coordinates of a set of salient point correspondences. In principle, direct methods are not limited to salient points and hence can exploit the information from *every* pixel of the image –with some limitations we discuss below. We can classify such methods as semidense and dense. Note that some of the systems presented below also use features for camera localization.

1) Semidense SLAM: Semidense visual SLAM only makes use of the high-gradient image pixels, as those are the only ones producing reliable matches.

SVO [13] –standing for semidirect visual odometry– uses feature correspondences as an implicit result of direct motion estimation instead of an explicit feature extraction and matching. The direct tracking is refined with bundle adjustment.

LSD-SLAM [1] –standing for Large Scale Direct Monocular SLAM– performs a probabilistic filtering-based depth map estimation which is tracked using direct image alignment. LSD-SLAM includes a pose graph optimization and loop closure to extend the algorithm to large scale scenarios. As it main weakness, the low textured areas are not reconstructed.

[14] also builds a probabilistic semidense approach. Differently from [1] it is built on top of a feature based SLAM and again low textured areas are not reconstructed.

2) Dense SLAM: Differently from the above ones, dense visual SLAM methods aim to estimate a depth for every pixel both high and low-gradient ones. [7], [15] where the first ones presenting dense results in real-time using a monocular camera. They not only minimize the difference between image intensities, but include a regularization term enforcing smooth solutions. This latest term is crucial for reconstructing low-gradient pixels. GPU processing is usually required to achieve real-time.

REMODE [16] (standing for Regularized Monocular Dense reconstruction) propose to integrate a Bayesian estimation of the inverse depth into the variational formulation. Uncertainty of the inverse depth is used to decrease the regularization in those areas with a low inverse depth uncertainty. Bayesian estimation offers a natural way to reject unreliable measurements in an on-line fashion. The camera pose optimization is based on features, similarly to [14].

B. Piecewise Planar Models from Visual Data

Piecewise planar and Manhattan models are a popular choice to obtain offline dense reconstructions in man-made environments. [17] achieves impressive results from a stereo sequence. [18] hypothesizes planes based on a sparse 3D reconstruction and tests their photometric compatibility in several views. [19] uses the Manhattan assumption –three dominant perpendicular directions– and superpixel classification to extract a room layout from a single view. [20], [21] use a multiview sparse feature map to estimate a more robust layout.

[3] assumes that the homogeneous-color regions from a superpixel segmentation are planar, and estimates a map composed of such planar areas and salient points. Planar areas are initialized by superpixel triangulation. Our contribution is an initialization based on superpixels and a semidense map that is faster and more accurate. [9] uses multiview superpixels and layout to estimate an accurate dense map using direct methods.

III. Overview

Figure 2 shows a simplified scheme of our algorithm. The computation is divided into three threads. The first one tracks the camera pose for every sequence frame I_n using a semidense map (section IV-A). The semidense map is the output of the second thread, that estimates the inverse depth ρ_u for the high-gradient pixels of a keyframe I_k (section IV-B). The keyframes are selected from the sequence frames using certain heuristics. Finally, the third thread estimates at a lower frame rate a dense map of the scene using the piecewise planar assumption and regularization (section V).

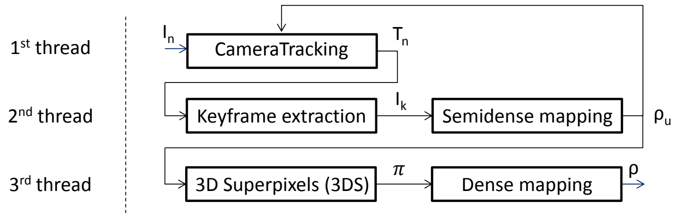


Fig. 2: Overview of our approach.

IV. Tracking and Mapping High-Gradient Pixels

A. Tracking High-Gradient Pixels

The transformation from the current camera frame to the global frame T_n is estimated based on the photometric reprojection error r^{p_u} using the inverse compositional approach [22]. The photometric error for the i^{th} pixel p_u^i is defined as

$$r^{p_u^i} = (I_k(F(T_k\hat{T}p_u^i)) - I_n(F(T_n p_u^i))) \qquad (1)$$

where F is the pinhole camera model. In p_u^i the subindex u stands for points of high gradient, to differentiate it from a general point p^i. The tracking thread only uses a subset of image points, composed of high-gradient points and superpixel contours. Sections IV-B and V-A detail how the 3D position for those is obtained.

We seek to estimate the transformation $\hat{T}$ from the closest keyframe I_k to the current frame I_n. T_k is the transformation from the last keyframe to the global reference frame. The seed for the transformation T_n comes from a constant velocity motion model –although this step is ignored if the photometric reprojection error is higher after applying it.

The tracking minimization is as follows

$$\hat{T} = \arg\min_{T} r^{p_u}. \tag{2}$$

$$r^{p_u} = \sum_{i=1}^{n} w_i (r^{p_u^i})^2. \tag{3}$$

The residuals are reweighted (w_i) with a robust cost function to remove the influence of outliers –in particular, occlusions.

For the optimization we use a minimal parametrization of the camera pose. The rigid body transformation T is mapped to the tangent space *se(3)* of the euclidean space *SE(3)* at the identity. The tangent space is also named the *twist coordinates* $\varepsilon = (w,v)^t \in \mathbb{R}^6$, where $w \in \mathbb{R}^3$ is the angular velocity and $v \in \mathbb{R}^3$ is the linear velocity. ε is mapped into *SE(3)* by the exponential map $T = exp_{se(3)}(\varepsilon)$ and the inverse is done by the logarithmic map $\varepsilon = log_{SE(3)}(T)$.

In the inverse compositional approach the update for the current camera pose T_n is calculated as

$$T_n = T_n \hat{T}^{-1} \tag{4}$$

$\hat{T}$ is calculated applying the Gauss-Newton update in the energy functional of equation 3

$$\delta\hat{\varepsilon} = -(J^T W J)^{-1} J^T W r \tag{5}$$

$$\hat{T} = exp_{se(3)}(\delta\hat{\varepsilon}) \tag{6}$$

Where W and r are the matrix for the weights of the Tukey's robust cost function and the residuals vector respectively. J is the jacobian of the residual $J = \frac{\partial r}{\partial \varepsilon}$. To obtain it we use the chain rule:

$$J = \frac{\partial r}{\partial \varepsilon} = J_F^r J_{T_k}^F J_{\varepsilon}^{T_k} \tag{7}$$

Where J_F^r are the gradients of the residual reference keyframe, $J_{T_k}^F$ is the derivative of the projection model with respect to the transformationT_k and $J_{\varepsilon}^{T_k}$ is the derivative of the transformation T_k with respect to the motion ε.

Note that using the inverse compositional approach the Jacobians are always calculated in the last keyframe, and there is no need to update them until a new frame becomes a keyframe since the transformation T_k, the points p_u and the gradients of the keyframe J_F^r are constant during the optimization. This approach significantly accelerates the motion estimation.

To bootstrap our system we follow a similar approach to [1], assigning as the depth map for the first frame a plane parallel to the image plane at random depth. The depth map converges to the ground truth after a few keyframes in most of the cases.

B. Mapping High-Gradient Pixels

Rapid camera motions require high-frequency map updates for the camera pose not to lose track. Similarly to [1], our system maintains a semidense map of high-gradient points that can be quickly updated and serves for camera tracking as described in section IV-A. This semidense map is not only used for tracking, but also for the estimation of the planar surfaces described in section V.

For each high-gradient pixel u, its inverse depth ρ_u is estimated by minimizing the photometric error r_{ph}^o for several overlapping views.

$$\hat{\rho}_u = \arg\min_{\rho_u} r_{ph}^o \tag{8}$$

$$r_{ph}^o = ||(I_k(s_u^k) - I_o(s_u^o)|| \tag{9}$$

$$s_u^o = G(s_u^k, T_k, T_o, \rho_u) \tag{10}$$

s_u^k are the pixel coordinates of a template around the pixel u in the image I_k. G is the function that backprojects the template s_u^k from the keyframe I_k at a distance ρ_u and then projects it to the overlapping image I_o.

For the first overlapping image we perform and exhaustive search in the epipolar line. In the rest of the images the search space is constrained by the current depth estimation and its uncertainty.

The optimization is performed using pixel coordinates and the optimal pixel coordinate $\hat{s}_u^o$ is then transformed into its corresponding optimal inverse depth $\hat{\rho}_u(\hat{s}_u^o)$. We repeat this process in 10 overlapping images yielding 10 inverse depth hypotheses $\hat{\rho}_{u[1-10]}$for every high-gradient pixel in the reference keyframe. $\sigma_{\hat{\rho}_u[1-10]}$ is approximated assuming an uncertainty of one pixel in the overlapping image:

$$\sigma_{\hat{\rho}_u[1-10]} = (\hat{\rho}_{u[1-10]}(\hat{s}_{u[1-10]}^o) - \hat{\rho}_{u[1-10]}(\hat{s}_{u[1-10]}^o + 1)) \tag{11}$$

We perform three additional procedures to remove potential outliers from our estimation and regularize the solution.

- *Gradient direction.* The inverse depth of the pixels whose epipolar line is perpendicular to the gradient direction cannot be reliably estimated from stereo [1]. We only estimate the depth for pixels having gradients around the epipolar direction and within a certain threshold.
- *Temporal consistency.* An estimated inverse depth is likely to be an inlier if the inverse depth hypotheses from several image pairs are similar. If the inverse depth hypotheses span over the epipolar line, the estimated inverse depth might be an outlier [23]. Inverse depths are sorted and we look for compatible values between at least 5 out of the 10 hypotheses. We calculate the ratio between the difference of the maximum and minimum optimal inverse depths (5 at least) and their global standard deviation $\sigma_{\hat{\rho}_u}$.

$$(\hat{\rho}_u^{max}{}_{[i,i+n]} - \hat{\rho}_u^{min}{}_{[i,i+n]})/\sigma_{\hat{\rho}_u} < 2 \tag{12}$$

$$\sigma_{\hat{\rho}_u} = \sqrt{\left(\sum_{k=i}^{i+n} \frac{1}{\sigma_{\hat{\rho}_u[k]}^{\;2}}\right)^{-1}} \tag{13}$$

The test is therefore repeated for $n = [4,...,9]$ and for $i = [1,...,10-n]$ spanning all different hypotheses combinations. The final optimal inverse depth $\hat{\rho}_u$ is the average of the temporally consistent hypotheses.

- *Spatial consistency.* Applying the smooth world assumption, neighboring pixels should have similar inverse depths. We run a test for the spatial similarity of the contiguous pixels inverse depths. The equations 12 and 13 are also applied for this test. Instead of computing them using the inverse depth hypotheses for every pixel, they are computed using the optimal inverse depths values of the pixel and its neighbors. $n = max(\#neighbors - 1, 1)$ and $i = 1$ in this case, therefore we require at least one match between the pixel and its neighbors. Again, we perform the average of the spatially consistent optimal inverse depths to smooth the final depth map.

Finally, the inverse depth estimation is scaled against the previous map. This helps to keep the scale in sequences with large changes in depth.

The 3D points with less uncertainty will be used for robust tracking –section IV-A– , reliable 3D superpixel estimation –section V-A– and variational mapping –section V-B.

V. Mapping Low-Gradient Pixels

A. 3D superpixels

The accurate semidense mapping from section IV-B and the 2D superpixels are used to efficiently estimate 3D planar superpixels.

First, each keyframe I_k is segmented into a set of superpixels $\mathscr{S}_k = \{s_1,\ldots,s_i,\ldots,s_m\}$ using the algorithm of [10]. Each 3D point p_u from the semidense map is then projected on the keyframe $u = F(T_k p_u)$. The 3D points p_u are assigned to the superpixels if their projections u lie within a threshold ξ –see algorithm 1.

Algorithm 1 Point to Superpixel Contour Assignment

```
procedure POINT_SUPERPIXEL_MATCHING($M,\mathscr{S}$)
    for $p_u \in M$ do                          ▷ For every point in the map
        $p_u \in \varnothing$
        for $s_i \in \mathscr{S}$ do                ▷ For every superpixel
            $u = F(T_k p_u)$                    ▷ Point's projection
            if $distance(u,\mathscr{C}(s_i)) < \xi$ then    ▷ If the point's projection is within a distance to the superpixel contour
                $p_u \in \mathscr{C}(s_i)$          ▷ The point belongs to the contour
            end if
        end for
    end for
end procedure
```

The 3D points associated to the contour of every superpixel $p_u \in \mathscr{C}(s_i)$ are used to robustly fit a plane π_i using singular value decomposition. We use RANSAC [24] for outlier rejection and consider three additional metrics to evaluate the quality of the estimated plane.

- *Normalized residuals test.* We calculate the ratio between the distances of the 3D points to the plane and the distances between the 3D points to themselves. If this ratio is less than a threshold –0.05 in this paper– the match is accepted.
- *Degenerated cases.* We look for degenerated cases where multiple solutions occur. For example, some contours might be close to a 3D line and have one dominant dimension. We avoid this cases by seeking for degenerate rank in the singular value decomposition.
- *Active search, temporal consistency.* Following a similar approach than [3], we actively search the 3D superpixels in the superpixels of neighboring frames by calculating the error between the reprojected contour and the contours of the potential matches in the neighbors frames. The reprojection error for a 3D contour point $p_u \in \mathscr{C}(s_i)$ of a superpixel s_i in a camera T_j is computed using the standard pinhole model F.

$$\varepsilon_j = u_{s_i}^j - F(T_j p_u) \tag{14}$$

  Where $u_{s_i}^j$ stands for the closest point to $F(T_j p_u)$ in contour of superpixel s_i in camera j. If enough overlapping in the reprojection is achieved for superpixel s_i in camera j, the match is accepted. If at least two matches are achieved for s_i, the 3D superpixel is accepted.

This active search of superpixels is of key importance, as it can reject the erroneous data association between 2D superpixels and 3D contours. This erroneous data association comes from the fact that a contour is surrounded by at least 2 superpixels and it is not possible to discern what superpixel corresponds to the contour using only one view. The active search seeks for consistency between multiple views and it helps mitigate the problem.

The whole pipeline of plane estimation from planes and superpixels is summarized in algorithm 2. We have observed three main advantages of this approach over the baseline [3].

- *Map completeness.* [3] needs relatively large parallax to initialize a superpixel by triangulation. Superpixels on high-parallax views might be quite different and hence difficult to match. We overcome these limitations by initializing directly in the reference keyframe using the existing 3D semidense map. As a result, we are able to initialize a higher number of superpixels.
- *Higher Accuracy.* In [3] the triangulation was done from two views. In this paper we incorporate the 3D information of a very accurate semidense map, estimated from more than two views.
- *Lower cost.* The initialization of [3] is very expensive due to a Montecarlo search over the space of plane configurations. Our initialization is a least-squares plane-fitting problem with closed form solution. See table I

Algorithm 2 Plane from Points

```
procedure PLANE_FROM_POINTS($p_u \in \mathscr{C}(s_i)$)
    $d = f_1(p_u)$    ▷ Average distance between the points
    $e_{min} = \infty$    ▷ Minimum error
    $e_{max} = 0.05$    ▷ Maximum normalized error allowed.
    $\pi_i$    ▷ Optimal plane
    $p = 0.99$    ▷ Probability for selecting only inliers
    $w = 0.5$    ▷ Inliers ratio
    $n_{hyp} = \frac{\log 1-p}{\log(1-w)^4}$    ▷ Number of hypotheses
    for $n \in n_{hyp}$ do    ▷ For number of hypotheses
        $p_u^* \in p_u$    ▷ 4 random points $\in p_u$
        $[U,S,V,\pi] = svd(p_u^*)$    ▷ SVD for $p_u^*$
        $e = f_2(\pi p_u)/d$    ▷ Normalized (d) Robust ($f_2$) error ($\pi p_u$ )
        $matchings = M(p_u,R,t)$    ▷ Matchings in active search. Temporal consistency, Equation 14. It depends on the pose of the neighboring cameras and the superpixel extraction in them.

        $inlier = TRUE$
        if $e > e_{max}$ then    ▷ Plane bad fitted.
            $inlier = FALSE$
        end if

        if $rank(p_u^*) < 3$ then    ▷ Degenerated case.
            $inlier = FALSE$
        end if

        if $matchings < 2$ then
            $inlier = FALSE$
        end if

        if $inlier == TRUE \cap e < e_{min}$ then
            $\pi_i = \pi$
            $e_{min} = e$
            $update(w)$    ▷ Update inlier ratio
            $n_{hyp} = \frac{\log 1-p}{\log(1-w)^4}$    ▷ Update hypotheses
        end if
    end for
     return $\pi_i$
end procedure
```

Method	Computational cost [ms]		
	Contour extraction	***Init.***	***Opt.***
This paper	**∼90 ms**	**∼20 ms**	∼5 ms
[3]	∼180 ms	∼370 ms	∼5 ms

TABLE I: Cost comparison between [3] and this paper.

for a time comparison.

B. Dense Mapping

A fully dense reconstruction –one depth for each pixel– can be estimated using a similar approach to [9]. The functional to minimize is a sum of three terms over the image domain Ω.

$$\begin{aligned} E_\rho &= \int_\Omega (\lambda_1 C(s_u,\rho(s_u)) + G(u,\rho(u)) + \\ &+ \frac{\lambda_2}{2} M(u,\rho(u),\rho_p(u)) \partial u \end{aligned} \quad (15)$$

The first term $C(s_u,\rho(s_u))$ is based on color difference between the reference image and the set of short-baseline images. Every patch s_u of the reference image I_r is first backprojected at an inverse distance ρ and projected again in every close image I_j.

$$\varepsilon(I_j,I_r,u,\rho) = I_r(u) - I_j(T_{rj}(u,\rho)) \quad (16)$$

λ_1 is a weighting factor that accounts for the relative importance of the photometric and gradient regularization terms.

$G(u^r,\rho(u))$ regularizes the solution. The specific form of this cost is

$$G(u^r,\rho(u)) = g(u^r)||\nabla\rho(u)||_\varepsilon \quad (17)$$

where $||\nabla\rho(u)||_\varepsilon$ is the Huber norm of the gradient of the inverse depth map and $g(u)$ is a per-pixel weight that decreases the regularization strength across image contours:

$$g(u) = e^{-\alpha||\nabla I_r(u)||_2} \quad (18)$$

Where α is a constant. The third term measures how far is the estimated depth from a piecewise planar reconstruction based on superpixels

$$M(u,\rho(u),\rho_p(u)) = w||\rho(u) - \rho_p(u)||_2^2 \quad (19)$$

ρ_p is the inverse depth prior coming from 3D superpixels (see section V-A).

w the weight of Tukey's cost function. Finally, we use the sub-sample accuracy method and the acceleration of the non-convex solution, both recommended in [7]. The functional is minimized following the primal-dual approach. For the details see [9].

We also propose to discard areas that are estimated with a large error. These areas mostly correspond to far areas due to low parallax, textureless areas not reconstructed with superpixels, and areas in the borders of the image. We detect these uninformative areas using the map superpixels and semidense points. We classify every superpixel as a high informative area or a poor informative area. We differentiate between large superpixels –low texture areas– and small superpixels – high texture areas. We only classify large superpixels as a high informative area if we have found a 3D superpixel in the reference image or in the neighbors images. For the rest of the superpixels, we will classify them as a high-informative area if most of the contour of the superpixel is already estimated by the accurate semidense approach. The rest of the superpixels will be ignored and then will not be reconstructed. Our results applying this technique are denoted as *Semidense mapping filtered* in the experimental section VI.

VI. Experiments

We have used the public TUM dataset [2] to evaluate the accuracy and computational cost of our algorithm. Also, we tested our system online with a hand-held camera. An illustrative video of such experiments can be found in the video accompanying the paper [1].

A. Comparison against [3]

Seq.	Keyfr.	*Error ratio,* $\frac{[3]}{ours}$	*Compl. ratio,* $\frac{ours}{[3]}$
fr3 str tex far	1	6.24	1.75
	2	1.17	1.16
	3	0.42	6.78
	4	0.78	1.76
	5	1.61	1.01
	6	0.55	1.23
fr2 xyz	1	0.59	1.05
	2	2.23	1.00
	3	1.08	4.42
	4	13.4	1.46
	5	11.3	1.48
	6	1.71	1.12
	7	34.5	0.76
fr3 nstr tex near	1	1.13	7.56
	2	0.98	18.70
	3	1.452	1.53
	4	3.02	1.41
	5	1.87	4.40
	6	1.42	0.86
	7	5.95	3.66
	8	2.20	10.10
	9	2.84	2.64
	10	1.03	2.77
	11	1.13	6.74
	12	4.31	8.54
	13	3.97	10.49
	14	1.18	1.46
	15	0.20	2.86
	16	0.24	2.10
	17	1.84	1.59

TABLE II: Error and completeness ratios between us and [3]. Numbers higher than 1 means us outperforming.

Tables II shows a quantitative comparison of our approach against the superpixel initialization of [3]. We report the error ratio defined as the mean reconstruction error of [3] over our mean reconstruction error; and the completeness ratio defined as our percentage of reconstructed pixels (over the total image pixels) over the percentage of reconstructed pixels of [3]. These ratios are defined so that numbers higher than 1 denote that we are outperforming [3]. For absolute accuracy and completeness results, the reader is referred to table III.

Notice that we are more accurate than [3] in most of the keyframes. There are two main reasons for that. The first one is the use of the semidense map for the initialization, that filters out most of the superpixel segmentation noise. In [3] we triangulated directly from the superpixel correspondences. The second one is the three rejection tests defined in section section V-A. We have observed that ratios greater than 3 correspond to estimation failures of our previous work [3] that are now correctly rejected by our three tests.

Notice also how we are able to reconstruct more superpixels than [3] (completeness ratio higher than one for most of the keyframes in table II). Again, the use of the semidense map makes our approach more resilient to the low repeatability of superpixels.

B. Comparison of direct mapping alternatives

This section compares several alternatives for direct monocular SLAM in real time in terms of depth accuracy, cost and map completeness. The approaches considered are semidense mapping, 3D superpixels mapping (3DS), dense mapping, semidense mapping filtered (see section V-B for the difference between dense mapping and dense mapping filtered) and several combinations of them.

Table III shows the quantitative results of the comparison. We report the mean and median errors over all the keyframes of the sequence for every mapping alternative; and the completeness of the map over the total number of image pixels. Table IV shows their mean computational cost.

		Error [cm]		
Seq.	***Mapping approach***	***Mean***	***Median***	***Compl.***
fr3 str tex far	Semidense	5.49	3.93	0.37
	3DS [3]	6.17	4.65	0.17
	3DS (ours)	4.14	3.61	0.25
	Semidense + 3DS (ours)	4.20	3.43	0.45
	Dense	25.12	6.18	1.00
	Dense + 3DS (ours)	23.96	4.98	1.00
	Semidense filtered	6.78	4.57	0.62
	Semidense filtered + 3DS (ours)	5.52	3.71	0.62
fr2 xyz	Semidense	6.35	2.50	0.16
	3DS [3]	14.87	3.19	0.11
	3DS (ours)	1.94	1.86	0.12
	Semidense + 3DS (ours)	3.13	2.01	0.23
	Dense	31.86	9.26	1.00
	Dense + 3DS (ours)	29.30	6.57	1.00
	Semidense filtered	12.03	5.38	0.29
	Semidense filtered + 3DS (ours)	6.76	2.74	0.29
fr3 nstr tex near	Semidense	3.03	2.46	0.25
	3DS [3]	2.97	2.70	0.45
	3DS (ours)	2.78	1.96	0.41
	Semidense + 3DS (ours)	2.84	2.49	0.50
	Dense	27.16	11.22	1.00
	Dense + 3DS (ours)	23.18	6.31	1.00
	Semidense filtered	8.48	4.41	0.55
	Semidense filtered + 3DS (ours)	3.04	2.32	0.55

TABLE III: Mean and median depth errors and map completeness for several mapping alternatives in 3 sequences of the TUM dataset.

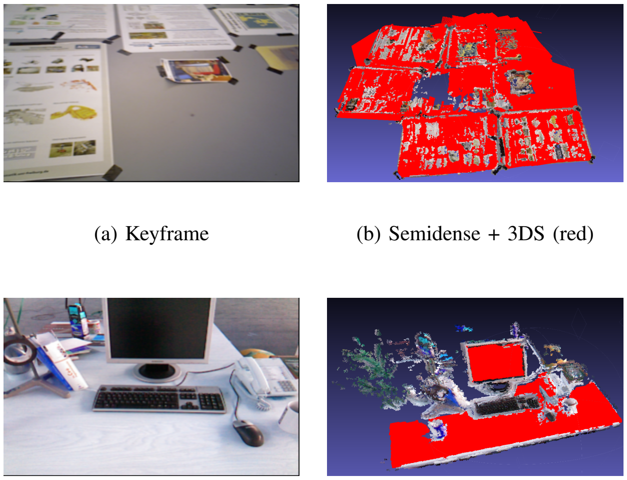
Observe how the 3D superpixels improve the accuracy of semidense and dense maps by comparing *Semidense* vs. *Semidense + 3DS (ours)* and *Dense* vs. *Dense + 3DS (ours)* in the three sequences. If the piecewise planar assumption holds in the current scene, which is usually the case in man-made ones, this will always be the case. Notice that the superpixel initialization proposed in this paper *3DS (ours)* always outperform the baseline initialization *3DS [3]*. Notice also how, in the semidense case, the addition of 3D superpixels increases the density of the map (in table

[1] The video is also available online at `https://youtu.be/SY_bBx7Ut-4`.

III, *Semidense + 3DS (ours)* has higher completeness than *Semidense*).

Table III shows that the accuracy of dense mapping is still limited. In our results, the dense mapping errors are 5 times bigger than the semidense mapping ones. Although 3DS improves the accuracy of dense maps, it is still much lower than the semidense one. Notice that the mean of the dense mapping error is always much larger than its median, suggesting that the depth error distribution has a long tail. Our approach *Semidense filtered* is able to eliminate such large depth errors by filtering out uninformative pixels (see section V-B for details). But the map completeness is reduced to values similar to *Semidense + 3DS* approaches, with similar accuracy and at a higher cost caused by regularizing the dense reconstruction (see the costs at table IV). *Semidense filtered* does not offer then a significant improvement over *Semidense + 3DS*, confirming that *3DS* can have an important role in dense monocular mapping indoors.

Figures 3 shows the 3D reconstruction of our proposal in two of the sequences of the TUM dataset. The 3D superpixels are in red. Notice the completeness and accuracy of the maps, and how the 3D superpixels (in red in the figure) play a key role in achieving a high completeness. Figure 4 shows a visual comparison between a semidense, dense and semidense + 3DS map.



(a) Keyframe

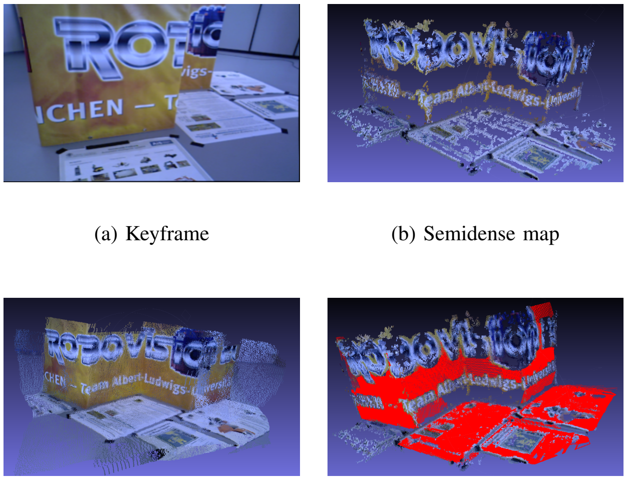
(b) Semidense + 3DS (red)

(c) Keyframe

(d) Semidense + 3DS (red)

Fig. 3: Mapping results of our proposal (*Semidense + 3DS (ours)*). (a) and (c) are selected keyframes of two sequences, (b) and (d) are the estimated maps.

Table IV shows the computational cost results for the mapping alternatives of this section, measured in a 3.5 GHz Intel Core i7-3770K processor with 8.0 GB of RAM memory. Notice first here that our approach has a computational cost 5 times lower than the baseline. Also observe that the low cost of semidense mapping and 3DS makes their combination an interesting alternative to dense variational mapping.



(a) Keyframe

(b) Semidense map

(c) Dense map

(d) Semidense +3DS (red)

Fig. 4: Mapping Results. (a) is a selected keyframe. (b) is the semidense map. (c) is the dense map. (d) is the semidense map using our approach (*Semidense + 3DS (ours)*).

Mapping approach	**Computational cost [ms]**
Semidense	∼350
3DS [3]	∼555
3DS (ours)	∼115
Semidense + 3D Sup(ours)	∼465
Dense	∼1800
Dense + 3DS (ours)	∼1965
Semidense filtered	∼1800
Semidense filtered + 3DS (ours)	∼1965

TABLE IV: Average computational cost for the direct monocular mapping alternatives.

VII. Conclusions

We have presented in this paper a direct SLAM algorithm for dense tracking and mapping using a monocular camera. Our approach leverages the piecewise planar assumption in indoor scenes to estimate accurate maps in real-time in a CPU. We think this is an interesting alternative to dense monocular SLAM, producing denser maps than standard semidense approaches with a small overload.

The specific contribution of this paper is a novel approach to estimate planar 3D superpixels based on the image segmentation and also on the estimated semidense map of high-gradient points. We have validated this approach in standard datasets and performed live experiments with a hand-held camera. Our algorithm has shown to outperform the baseline for superpixel initialization, both in accuracy and computational cost. The full pipeline runs in real-time in a standard CPU.

Acknowledgments

This research was funded by the Spanish government with the projects IPT-2012-1309-430000 and DPI2012-32168

REFERENCES

[1] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 834–849.

[2] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[3] A. Concha and J. Civera, "Using superpixels in monocular SLAM," in *IEEE International Conference on Robotics and Automation*, Hong Kong, June 2014.

[4] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, 2011, pp. 20–25. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2011.5980357

[5] C. McManus, W. Churchill, A. Napier, B. Davis, and P. Newman, "Distraction suppression for vision-based pose estimation at city scales," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 3762–3769.

[6] A. Concha, P. Drews-Jr, M. Campos, and J. Civera, "Real-time dense 3d mapping of underwater environments from monocular images," in *MTS/IEEE OCEANS*, 2015.

[7] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2320–2327.

[8] J. Sturm, E. Bylow, F. Kahl, and D. Cremers, "CopyMe3D: Scanning and printing persons in 3D," in *German Conference on Pattern Recognition (GCPR)*, Saarbrücken, Germany, September 2013.

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

[10] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[11] 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.

[12] A. J. Davison, N. D. Molton, I. D. Reid, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. June, pp. 1052–1067, 2007.

[13] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[14] R. Mur and J. D. Tardos, "Probabilistic semi-dense mapping from highly accurate feature-based monocular slam," in *Robotics Science and Systems*, 2015.

[15] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Pattern Recognition*. Springer, 2010, pp. 11–20.

[16] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time," in *2014 IEEE International Conference on Robotics and Automation (ICRA 2014)*, 2014.

[17] D. Gallup, J.-M. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1418–1425.

[18] A. Argiles, J. Civera, and L. Montesano, "Dense multi-planar scene estimation from a sparse set of images," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 4448–4454.

[19] V. Hedau, D. Hoiem, and D. Forsyth, "Recovering the spatial layout of cluttered rooms," in *Computer vision, 2009 IEEE 12th international conference on*. IEEE, 2009, pp. 1849–1856.

[20] G. Tsai, C. Xu, J. Liu, and B. Kuipers, "Real-time indoor scene understanding using bayesian filtering with motion cues," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 121–128.

[21] A. Flint, D. Murray, and I. Reid, "Manhattan scene understanding using monocular, stereo, and 3D features," in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2228–2235.

[22] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Int. J. Comput. Vision*, vol. 56, no. 3, pp. 221–255, Feb. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000011205.11775.fd

[23] G. Vogiatzis and C. Hernndez, "Video-based, real-time multi view stereo," in *Image and Vision Computing. Volume 29, Issue 7, June 2011, Pages 434441*, 2011.

[24] R. Raguram, J. Frahm, and M. Pollefeys, "A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus," in *Proceedings of the European Conference on Computer Vision*, 2008, pp. 500–513.