

RGBDTAM: A Cost-Effective and Accurate RGB-D Tracking and Mapping System

Alejo Concha and Javier Civera

Abstract—Simultaneous Localization and Mapping using RGB-D cameras has been a fertile research topic in the latest decade, due to the suitability of such sensors for indoor robotics. In this paper we propose a direct RGB-D SLAM algorithm with state-of-the-art accuracy and robustness at a low cost. Our experiments in the RGB-D TUM dataset [34] effectively show a better accuracy and robustness in CPU real time than direct RGB-D SLAM systems that make use of the GPU.

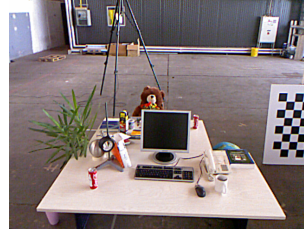
The key ingredients of our approach are mainly two. Firstly, the combination of a semi-dense photometric and dense geometric error for the pose tracking (see Figure 1), which we demonstrate to be the most accurate alternative. And secondly, a model of the multi-view constraints and their errors in the mapping and tracking threads, which adds extra information over other approaches. We release the open-source implementation of our approach¹. The reader is referred to a video with our results² for a more illustrative visualization of its performance.

I. INTRODUCTION

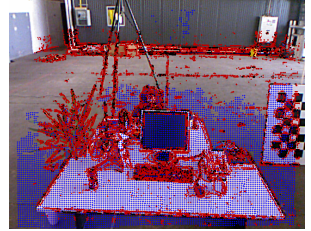
The availability of affordable and accurate RGB-D cameras has caused a profound impact in mobile robotics. Currently, the research lines based on such technology are as varied as object recognition [23], scene recognition and understanding [14], [31], person detection [32] or human-robot interfaces [35].

RGB-D sensors have been used also for Visual Odometry (VO) –i.e., the estimation of the incremental motion of the camera from the sensor content– and SLAM –acronym for Simultaneous Localization and Mapping, aiming at estimating globally consistent scene models in addition to camera ego-motion. Again, the rationale is the same: RGB-D cameras are perfectly suited to indoor robotics, offering accurate, dense and fully observable measurements within a range at a low cost. Achieving the same accuracy in dense 3D reconstructions from RGB-only sequences is still a challenge [5].

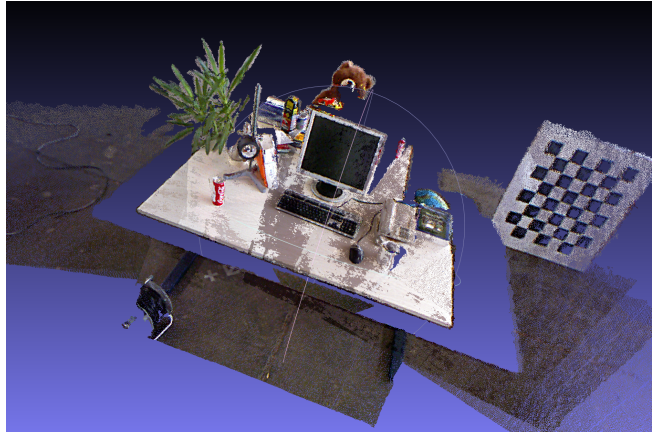
RGB-D cameras, however, have several limitations. One of the most relevant is that they cannot operate under direct sunlight. Also, they have a minimum and maximum depth range, and their depth measurements are noisy for absorbent and reflective surfaces. In principle, these limitations should have a small effect in the specific applications of VO and SLAM, as they can still use multi-view constraints to estimate the ego-motion and the map. However, the state-of-the-art *direct* RGB-D SLAM systems mostly use the depth



(a) Sample frame



(b) Sample frame and map projection



(c) 3D map after back-projecting the depth maps from every keyframe.

FIG. 1: (a) Sample frame for one of our experiments. (b) Same frame with the reprojected map in red and blue. We minimize the photometric error for red points and the geometric error for blue points. Note that distant points are mostly red due to the range limit of the depth sensor. Such points were mapped using multi-view RGB-only constraints. (c) 3D map, composed of the back-projected (non-fused) point clouds from every keyframe.

image constraints and do not fully exploit the information from multiple RGB views.

In this paper we present a direct RGB-D SLAM system that fuses multi-view and depth information. Such fusion extends the range of the maps from the typical few-meters one in RGB-D sensors to potentially infinity. Figure 1 illustrates this addition of distant, multi-view points to the map.

Our second contribution is a thorough analysis of the photometric and geometric residual combination. In our experiments, a semi-dense photometric and dense geometric residual has the highest accuracy and robustness. Our experimental results in a public dataset shows that RGBDTAM outperforms the state of the art in direct RGB-D SLAM.

The intuition of the above is, high-gradient pixels are the most informative for multi-view estimation. If the photomet-

Alejo Concha and Javier Civera are with I3A, Universidad de Zaragoza, Spain alejocb@unizar.es, jcivera@unizar.es

¹Our code can be found in this link <https://github.com/alejocb/rgbdtam>

²<https://www.youtube.com/watch?v=sc-hqtJtHD4>

ric multi-view residual is dense and most of it is composed of low-texture pixels, it is dominated by the noise and hence the estimation is of low accuracy.

In the case of the geometric error, all the pixels have a high signal/noise ratio. There are some degenerated cases, though, where some degrees of freedom are not constrained, and those justify the combination of both residuals. As they are complementary, the minimization of both errors achieves the best performance. The photometric error is useless in texture-less scenarios, and the geometric one is useless in structure-less scenarios.

The rest of the paper is organized as follows. Section II describes the related work. Section III gives an overview of the full RGB-D SLAM system. Section IV details the tracking thread of our SLAM system, section V the local mapping thread, and section VI the global mapping and loop closure algorithms we use in our system. Finally, sections VII and VIII show the experimental results and conclusions.

II. RELATED WORK

One of the first approaches for direct RGB-D odometry is KinectFusion [28], which uses only the depth channel D to estimate the odometry and a dense map and discards the RGB information. As its main limitations, it is restricted to small workspaces and will probably fail if the scene does not contain enough geometric structure.

Kintinuous [38] builds on KinectFusion and uses a rolling cyclical buffer that shifts the volume as the camera is moving, hence not being restricted to small workspaces. It also includes loop closing and pose graph optimization for global consistency.

[33] is one of the first approaches that proposes to minimize the photometric error between the current frame and a past frame.

DVO SLAM [19], [20] models the map as a pose graph. The constraints between keyframes are set from the tracking thread, which is based on dense photometric and geometric error minimization. This system also achieves CPU real time but not at full resolution (640×480 pixels).

[22] shows and compares three different alignment strategies for direct tracking, namely the forward-compositional, the inverse-compositional and the efficient second-order minimization approach. In this paper we use the inverse composition, as it is the most efficient of the three.

[7] estimates the relative motion between frames by a least-squares optimization that minimizes the 3D geometric error between corresponding RGB salient points. [16] uses the alignment obtained from feature matching as a seed for a joint optimization of the RGB-D point clouds. In both cases these relative transformations form the edges of a pose graph that are optimized using g^2o [13] and TORO respectively.

Regarding the weighting of the geometric and the photometric error [36], [6], [24] and [37] scale the errors with a heuristic constant. [19] weights both contributions according to their respective covariances. [25] scale each depth error using its squared inverse depth.

[15] proposes to use the inverse depth in the minimization of the geometric reprojection error. We evaluate this parametrization in our system.

ElasticFusion [39] is one of the most recent works and the RGB-D SLAM state of the art in terms of accuracy. The tracking thread uses ICP and dense photometric reprojection error. It achieves global consistency in a map-centric manner by a non-rigid deformation of the map structure, instead of using a more standard pose-centric graph optimization.

Some of these approaches incorporate the multi-view constraints in the tracking thread by weighting the errors with the standard deviation of the depth/inverse depth but they lack a multi-view model in the mapping thread. In our approach we also use multi-view constraints in the mapping. We show in our results that a semi-dense photometric error improves the accuracy and the efficiency of the estimation. [9], [8] and [11] have shown the effectiveness of a semi-dense or sparse photometric error in the optimization of the camera pose, but in a monocular setting.

Table I details the use of multi-view/depth information and semi-dense/dense residuals for several direct RGB-D SLAM methods in the literature. Notice that our approach is the first one using semi-dense RGB residuals and multi-view constraints for mapping in direct RGB-D SLAM. The reader is referred to section VII for the analysis showing that this combination is the best performing one.

It is worth remarking that some *feature-based* RGB-D SLAM methods (e.g., [16], [7], [27]) use multi-view constraints in the mapping thread and run in real time in a standard CPU. Our contribution and analysis is, however, focused on *direct* methods.

III. NOTATION

We follow the standard approach of Parallel Tracking and Mapping, first proposed in [21], and divide our algorithm into two threads.

The mapping thread estimates a scene map \mathcal{M} from a set of m selected keyframes $\{\mathcal{K}_1, \dots, \mathcal{K}_j, \dots, \mathcal{K}_m\}$. Each keyframe $\mathcal{K}_j = \{T_w^j, P^j\}$ is modeled with its pose T_w^j in a world frame w and its associated point cloud $P_w^j = \{p_w^1, \dots, p_w^i, \dots, p_w^n\}$ where each point p_w^i contains photometric and geometric information.

The tracking thread estimates the pose of the current frame by minimizing the geometric and photometric reprojection error of its associated point cloud with respect to a previous keyframe. If the scene is revisited the reference keyframe is selected using the method in section VI-B. If the camera is moving through unexplored areas, new keyframes are created based on the camera motion and the overlap with the current point cloud.

IV. ROBUST RGB-D TRACKING

For the camera motion estimation we minimize a functional which is composed of two terms –the photometric

	Tracking		Mapping		RGB-tracking		D-tracking	
	D	MV	D	MV	Dense	Semi-dense	Dense	Semi-dense
Newcombe et al. 2011 [28]	✓		✓				✓	
Whelan et al. 2012 [38]	✓		✓		✓		✓	
Kerl et al. 2013 [19]	✓		✓		✓		✓	
Meilland & Comport 2013 [25]	✓	✓	✓		✓		✓	
Gutierrez et al. 2015 [15]	✓	✓	✓		✓		✓	
Whelan et al. 2016 [39]	✓		✓		✓		✓	
Jaimez et al. 2017 [18]	✓	✓			✓		✓	
RGBDTAM	✓	✓	✓	✓		✓	✓	

TABLE I: State-of-the-art tracking-and-mapping RGB-D VO/SLAM systems; and their use of depth/multi-view constraints and dense/semi-dense residuals. D stands for depth and MV stands for multi-view.

error r_{ph} and the geometric error r_g . r_{ph} and r_g will be defined in the following subsections.

$$\{\hat{T}, \hat{a}, \hat{b}\} = \arg \min_{T, a, b} r_{ph} + \lambda r_g. \quad (1)$$

a and b are the gain and brightness of the current image and \hat{T} is the estimated incremental motion of the current camera pose. λ is a learned constant weighting the photometric and geometric terms. Notice that we only optimize T , a and b , therefore we keep the point cloud fixed –for efficiency reasons– and do not optimize jointly the poses and the points. We use a minimal parametrization for the camera pose, the rotation R is mapped into the tangent space $\mathfrak{so}(3)$ of the rotation group $SO(3)$ at the identity. Therefore the increments –the angular increment $\delta\omega$ and the increment for the translation δt – are defined as follows:

$$T = \begin{bmatrix} \exp_{\mathfrak{so}(3)}(\delta\omega) & \delta t \\ 0_{1 \times 3} & 1 \end{bmatrix}. \quad (2)$$

We estimate the transformation T_w^f from the current camera frame f to the global reference frame w using Gauss-Newton optimization and the inverse compositional approach [1] in equation 1.

The update for the current camera pose T_w^f is as follows

$$T_w^f \leftarrow T_w^f \hat{T}^{-1}. \quad (3)$$

A. Photometric error (r_{ph})

We minimize the photometric error only for those pixels belonging to Canny edges [2]. Their inverse depth is estimated using the mapping method described in section V.

The photometric error for the tracking thread is as follows:

$$r_{ph} = \sum_{i=1}^n w_p \left(\frac{(I_k(\pi(T_w^k p_w^i)) - a I_f(\pi(T_w^f \hat{T}^{-1} p_w^i)) + b)^2}{\sigma_{ph}^2} \right). \quad (4)$$

The first term $I_k(\pi(T_w^k p_w^i))$ is the intensity of the 3D point p_w^i in a keyframe I_k and the second term $I_f(\pi(T_w^f \hat{T}^{-1} p_w^i))$ is the intensity of the same 3D point in the current frame I_f . $\pi()$ is the projection function. Global illumination changes are addressed by estimating a and b , which are the gain and brightness of the current frame with respect to the current

keyframe. w_p is the Geman-McClure robust cost function, used to remove the influence of occlusions and dynamic objects.

B. Covariance-weighted Geometric error (r_g)

The second term in equation 1 is related to the depth measurements. The 3D point cloud is aligned with the current camera and the error between the inverse depth of the points $\frac{1}{e_z^T T_w^f p_w^i}$ and the measured inverse depth from the depth channel D_f is minimized.

$$r_g = \sum_{i=1}^n w_p \left(\frac{\left(\frac{1}{e_z^T T_w^f p_w^i} - D_f(\pi(T_w^f T^{-1} p_w^i)) \right)^2}{\sigma_g^2} \right). \quad (5)$$

w_p is again the Geman-McClure robust cost function. e_z is a 3D vector defined as $e_z = [0, 0, 1]$

Contrary the photometric residual (detailed in section IV-A), in this case a dense optimization is better than a semi-dense one. We homogeneously subsample the pixels used in the geometric error, in order to achieve CPU real-time performance. We use four pyramid levels (from 80×60 to 640×480). For the first level we use all pixels. For the second, third and fourth levels we use one in every two, three and four pixels respectively –horizontally and vertically.

Covariance Propagation for Structured Light Cameras: In order to estimate a value for the standard deviation of the geometric residual σ_g , we model the depth error of RGB-D cameras as that of the stereo. Our analysis is valid for RGB-D sensors based on structured light patterns, such as the Kinect v1 or the Google Tango.

Focusing our analysis in the epipolar plane, the stereo depth z only depends on the disparity d , the camera focal length f and the baseline b

$$z = \frac{fb}{d}. \quad (6)$$

For the inverse depth ρ

$$\rho = \frac{d}{fb}. \quad (7)$$

Assuming a disparity error with standard deviation σ_d (and no error for the focal length and the baseline), a first-order propagation gives the following standard deviation for the depth error

$$\sigma_z = \frac{\partial z}{\partial d} \sigma_d = \frac{fb}{d^2} \sigma_d = \frac{z^2}{fb} \sigma_d. \quad (8)$$

The inverse depth parametrization [3] is linearly dependent on the disparity. The first order error propagation gives, for a fixed baseline, a constant uncertainty in the inverse depth.

$$\sigma_\rho = \frac{\partial \rho}{\partial d} \sigma_d = \frac{\sigma_d}{fb}. \quad (9)$$

C. Scaling parameters

As we combine residuals of different magnitudes, we need to scale them according to their covariances. For the geometric error we propagate its uncertainty using equations 8 and 9. For the photometric error we use the median absolute deviation of the residuals of the previous frame to extract a robust estimation of the standard deviation.

$$\sigma_{ph} = 1.482 * \text{median}(r_{ph} - \text{median}(r_{ph})). \quad (10)$$

V. RGB-D MAPPING

We add a new keyframe in the map when the percentage of pixels that is visible from the previous keyframe is below a threshold. The mapping thread estimates a semi-dense map as soon as possible, in order to minimize the tracking failure risk.

Every pixel may have up to two sources of information to estimate its inverse depth: The raw depth sensor reading (ρ_1) and multi-view geometry (ρ_2).

For the multi-view triangulation we follow an approach similar to [9], [4]. The inverse depth ρ_2 for every high-gradient pixel u^* in a keyframe I_j is estimated by minimizing its photometric error r_{ph}^o with respect to several overlapping views I_o .

$$\hat{\rho}_2 = \arg \min_{\rho_2} r_{ph}, \quad (11)$$

with

$$r_{ph} = \sum_o \left\| (I_j(s_{u^*}) - I_o(G(s_{u^*}, T_w^j, T_w^o, \rho))) \right\|_2^2. \quad (12)$$

s_{u^*} are the pixel coordinates of the template (we use a one-dimensional patch, similarly to [9]) around the pixel u^* and G is the function that backprojects the template from the new keyframe I_j to the 3D world and projects it back to each overlapping image I_o .

These two contributions are fused using their uncertainties as follows

$$\rho = \frac{\sum_{j=1}^2 \frac{\rho_j}{\sigma_j^2}}{\sum_{j=1}^2 \frac{1}{\sigma_j^2}}, \quad \sigma = \frac{1}{\sum_{j=1}^2 \frac{1}{\sigma_j^2}}. \quad (13)$$

The uncertainties σ_j are estimated using equation 9. Notice that we do not fuse the inverse depth map of the current keyframe with the inverse depth map of the previous keyframes or the 3D model. There is a reason for this. We do not optimize jointly the pose of the keyframes and the 3D point cloud (as it is done in most direct algorithms [29], [9]). The fusion of different depth maps transfers the errors of each keyframe to the 3D map, resulting in a less accurate localization of the camera.

VI. LOOP CLOSURE AND MAP REUSE

A. Loop closure

The back-end of our algorithm is composed of loop closure detection and pose-graph optimization over the keyframes. We used the open library DBow2 [12] for appearance-based loop closure and the vocabulary created by the ORB-SLAM authors [26]. The ratio between the best match (a previous keyframe) and a neighboring keyframe of the current keyframe is calculated. If this ratio is higher than a threshold (0.5 in our experiments), the previous keyframe becomes a candidate for loop closing.

Once the candidate has been selected, we search for ORB [30] correspondences in both keyframes and use RANSAC [10] to get the 6 DOF transformation between the sparse point clouds. We use the Horn's method [17] to calculate this transformation T_j^k between keyframes j and k . We define a point as an inlier if the reprojection error –taking into account the pyramid level– is smaller than a threshold. If a minimum number of inliers is found the loop closure is accepted. Once the loop is detected the 6-DoF poses of all the keyframes are refined using pose-graph optimization with the g^2o library [13]. The following functional is minimized:

$$\{\hat{T}_w^1, \dots, \hat{T}_w^j, \dots, \hat{T}_w^k, \dots, \hat{T}_w^m\} = \arg \min_{\{T_w^1, \dots, T_w^m\}} \sum_{j,k} r_{j,k}^\top \Lambda_{j,k} r_{j,k} \quad (14)$$

Where $\{T_w^1, \dots, T_w^j, \dots, T_w^k, \dots, T_w^m\}$ are the poses of the n keyframes in the map. $\Lambda_{j,k}$ is the information matrix, which we set to the identity. $r_{j,k}$ is the residual for the edge j, k which is defined as follows:

$$r_{j,k} = \log(T_j^k T_k^w T_w^j). \quad (15)$$

B. Map reuse

Instead of continuously creating new keyframes we adopt a conservative strategy that privileges the use of the already existing ones. Our system looks for overlapping keyframes in a specific area before creating a new one, and in this manner we reduce the accumulated drift. Again, we use DBow2 [12] to obtain a list of candidate keyframes imaging the current tracked area. We propose two heuristic rules to discard invalid candidates:

- An overlap of at least 80% between the previous keyframe and the current frame is required.
- The photometric and geometric reprojection error are required to be smaller than 3 times the standard deviation.

ation of both errors. For the photometric error we set $\sigma_{ph} = 15$;

After applying these heuristics to remove loop outliers, we select the oldest candidate keyframe and use it for tracking. Notice that the pose of the old keyframe is taken after the optimization of the pose-graph functional –equation 14.

If these heuristics do not hold for any previous keyframe, then we try to close the loop following the approach described in the previous section VI-A. Finally, if these described strategies do not succeed we assume the system is exploring new areas and create a new keyframe.

VII. EXPERIMENTAL RESULTS

For our experimental results we use the publicly available TUM dataset [34]. We have done an exhaustive evaluation in all the static sequences of the dataset. We run our code on every sequence 5 times with different random initialization parameters and report the median of the 5 trajectory errors.

a) Comparison against direct RGB-D SLAM: We compare our approach against ElasticFusion [39], a state-of-the-art direct RGB-D SLAM system. We do not include other direct approaches in the evaluation, as [39] outperforms all the previous baselines.

Table II shows the trajectory error (RMSE). RGBDTAM shows better accuracy and robustness in most of the sequences. We used the same sequences than [39]. RGBDTAM is in general more robust than ElasticFusion in sequences with poor structure and texture in close objects (sequences fr2 coke, fr2 dishes, fr2 metallic sphere, fr3 cabinet). The semi-dense photometric residual is more robust in these sequences, as textureless areas do not contribute to the optimization.

The higher accuracy, higher robustness and lower cost of RGBDTAM with respect to ElasticFusion is remarkable and deserves further elaboration. ElasticFusion aims to estimate a map-centric global representation by an non-rigid fusion of the RGB-D keyframes; estimating visually appealing maps with an appearance of global consistency. RGBDTAM, by adopting the more traditional pose-centric approach, focuses on the solid probabilistic integration of the most informative data, and presumably this is the reason of its higher accuracy. We believe that the two approaches are complementary and valuable and would like to combine their respective strengths in future work.

b) Comparison against feature-based RGB-D SLAM: We compare RGBDTAM against ORB-SLAM2 [27], the state-of-the-art feature-based RGB-D SLAM system.

Table III shows the trajectory error (RMSE) in the TUM dataset. RGBDTAM has worse accuracy in most of the sequences. We have used the same sequences than the original paper [27]. We believe the reason for our worse performance is that RGBDTAM alternates between tracking and triangulation, lacking a joint optimization of poses and points in the mapping thread.

#	Sequence Name	RMSE [cm]	
		[39]	RGBDTAM
1	fr1 360	10.8	10.1
2	fr1 desk	2.0	2.7
3	fr1 desk2	4.8	4.2
4	fr1 floor	-	-
5	fr1 plant	2.2	2.5
6	fr1 room	6.8	15.5
7	fr1 rpy	2.5	2.1
8	fr1 teddy	8.3	8.1
9	fr1 xyz	1.1	1.0
10	fr2 360 hemisphere	-	-
11	fr2 360 kidnap	-	-
12	fr2 coke	-	6.0
13	fr2 desk	7.1	2.7
14	fr2 dishes	-	3.6
15	fr2 large no loop	-	-
16	fr2 large with loop	-	-
17	fr2 metallic sphere	-	-
18	fr2 metallic sphere 2	-	5.2
19	fr2 pioneer 360	-	-
20	fr2 pioneer slam	-	-
21	fr2 pioneer slam2	-	-
22	fr2 pioneer slam3	-	-
23	fr2 rpy	1.5	0.2
24	fr2 xyz	1.1	0.7
25	fr3 cabinet	-	5.7
26	fr3 large cabinet	9.9	7.0
27	fr3 long office household	1.7	2.7
28	fr3 nostr. notext. far	-	-
29	fr3 nostr. notext. near withloop	-	-
30	fr3 nostr. text. far	7.4	2.6
31	fr3 nostr. text. near withloop	1.6	1.0
32	fr3 str. notext. far	3.0	1.3
33	fr3 str. notext. near	2.1	4.4
34	fr3 str. text. far	1.3	1.0
35	fr3 str. text. near	1.5	1.0
36	fr3 teddy	4.9	-

TABLE II: RMSE for ElasticFusion [39] and RGBDTAM in the static sequences of [34]. Results from [39] are reported with best per-sequence-parameters; ours are with best per-dataset-parameters.

#	Sequence Name	RMSE [cm]	
		[27]	RGBDTAM
2	fr1 desk	1.6	2.7
3	fr1 desk2	2.2	4.2
6	fr1 room	4.8	15.5
13	fr2 desk	0.9	2.7
24	fr2 xyz	0.4	0.4
27	fr3 long office household	1.0	2.7
29	fr3 nstr	1.9	1.6

TABLE III: RMSE for ORBSLAM2 [27] and RGBDTAM

c) Evaluation of the residual configuration: Table IV shows a comparison between different configurations in the tracking thread. Observe that minimizing the photometric error for a semi-dense subset of high-gradient pixels (*PS* row in the table) is in general more accurate than using the full image (*PD* row in the table). Notice also that a semi-dense approach is more robust. When using a dense approach the

camera track was lost in some sequences with very little texture, where the noise of textureless areas in the image and other artifacts (such as reflections) dominated the solution.

Compare now the errors between a semi-dense and a dense point cloud for the geometric depth error (*GIDS* and *GIDD* rows respectively). Notice that in this case a dense approach is more accurate. We homogeneously selected a subset of the points of the geometric point cloud in order to achieve real-time performance. We have observed that this reduction of the dimensionality of the geometric error does not impact the performance of our approach. Notice in table IV that the subsampled version *GIDD* only obtains slightly less accurate results than the fully dense version *GIDD**. The best configuration in terms of efficiency, accuracy and robustness is the one that fuses the semi-dense photometric error and a subsampled of the dense geometric error in the optimization (*PS+GIDD* row).

d) Depth vs inverse depth in the geometric reprojection error: The last two rows in Table IV reports the comparison between the the inverse depth and the depth in the minimization of the geometric error for the best performing configuration of the tracking thread.

We obtained slightly better results for the inverse depth case, but the difference is small due to the range limits of RGB-D sensors. The inverse depth is particularly useful for distant points, for which a depth sensor does not measure its depth.

e) Computational time: All the experiments were run in CPU real-time (the average time being 35.86ms per frame) on a laptop with a 3.5 GHz Intel Core i7-3770K processor and 8.0 GB of RAM memory. Notice that ElasticFusion needs GPU processing and hence, being the comparison difficult, their cost will be presumably higher than ours.

For the semi-dense (contour based) photometric error we track up to 8K points per keyframe. For the geometric error, we homogeneously subsample it as explained in section V. This configuration is the reported in tables II, III and IV.

f) Failure modes: Notice in Table II that RGDTAM failed in 13 sequences out of 36. For the first 9 sequences there were missing frames that led to tracking failure in our system. In sequence 17 a relatively big and slow-moving dynamic object was not rejected as outlier by the robust cost function. For the next two sequences the scene did not contain structure nor texture, as the camera was moving on top of a textureless floor. Both the geometric and photometric errors were uninformative. For the last case (sequence number 36), the camera came very close to an object and performed a pure rotation. The object was closest than the minimum depth range and the multi-view mapping was not able to estimate a map without parallax.

g) Qualitative results.: Figure 2 shows several 3D maps obtained by our system in the TUM dataset. Notice the high accuracy of the map, even when we do not fuse the point clouds from different keyframes. Notice also that the 3D reconstruction in Fig. 2(c) corresponds to a structureless but textured scene, and hence accurate thanks to the

photometric part of the residual. On the other side, the 3D reconstruction in Fig. 2(d) is of a textureless scene of rich structure, and hence only possible thanks to the geometric part of the residual. See table IV for quantitative results in these sequences.

VIII. CONCLUSIONS

In this paper we have presented a direct RGB-D SLAM system with loop closure and map reuse capabilities. Our main contribution is the integration of RGB multi-view constraints in both the tracking and mapping thread. Such multi-view constraints increase the accuracy of the estimation due to two factors. First, the addition of distant points, out of the RGB-D sensor range, to the map. And second, the extra accuracy gained in high-parallax configurations.

We have compared different settings for the photometric and the geometric residual in the tracking thread, concluding that a combination of a semi-dense photometric error and a dense geometric error is the best combination in terms of accuracy and robustness. We have evaluated the minimization of the depth and inverse depth in the geometric error. The inverse depth parametrization is slightly more accurate in our results. Finally, we have shown that our approach outperforms the state of the art on direct RGB-D SLAM systems in terms of trajectory accuracy. Our system is also amongst the ones with the lowest cost, running in real time on a standard CPU.

ACKNOWLEDGMENTS

This research has been partially funded by the Spanish government (project DPI2015-67275) and the Aragón regional government (Grupo DGA T04-FSE).

REFERENCES

- [1] Simon Baker and Iain Matthews. Lucas-Kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.
- [2] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [3] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008.
- [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, Wajahat Hussain, Luis Montano, and Javier Civera. Incorporating scene priors to dense monocular mapping. *Autonomous Robots*, 39(3):279–292, 2015.
- [6] Dima Damen, Andrew Gee, Walterio Mayol-Cuevas, and Andrew Calway. Egocentric real-time workspace monitoring using an RGB-D camera. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1029–1036. IEEE, 2012.
- [7] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-D mapping with an RGB-D camera. *IEEE Transactions on Robotics*, 30(1):177–187, 2014.
- [8] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [9] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *European Conference on Computer Vision*, pages 834–849, 2014.

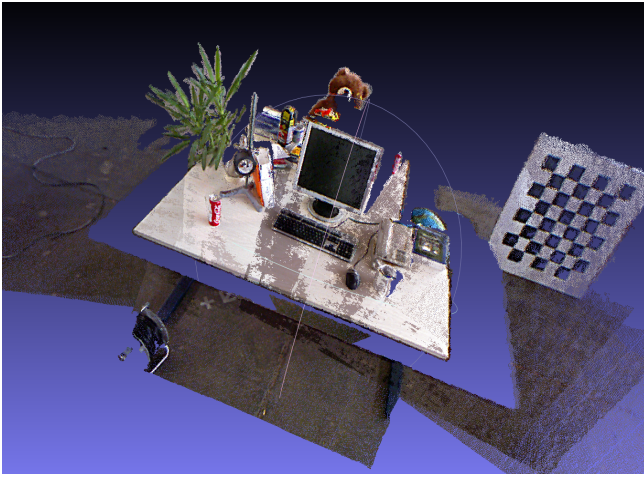
#Seq.	1	2	3	4	5	6	7	8	9	10	11	12
PS [RMSE cm].	9.3	2.7	6.4	-	4.4	13.5	2.5	14.1	1.0	-	-	8.8
PD [RMSE cm].	12.4	5.0	8.4	-	8.2	25.0	2.3	12.6	1.0	-	-	-
GIDS [RMSE cm].	12.7	3.2	6.5	-	6.9	12.5	5.4	12.5	6.3	-	-	-
GIDD [RMSE cm].	13.4	3.4	6.6	-	6.1	12.9	1.7	9.5	1.3	-	-	-
GIDD* [RMSE cm].	13.2	3.0	6.3	-	5.0	15.1	1.8	8.1	1.2	-	-	-
PS + GIDD [RMSE cm].	10.1	2.7	4.2	-	2.5	15.5	2.1	8.1	1.0	-	-	6.0
PS + GDD [RMSE cm].	9.4	2.5	4.3	-	2.9	16.1	2.3	8.0	1.0	-	-	7.2

#Seq.	13	14	15	16	17	18	19	20	21	22	23	24
PS [RMSE cm].	2.6	3.6	-	-	-	9.3	-	-	-	-	0.2	0.5
PD [RMSE cm].	12.1	15.7	-	-	-	-	-	-	-	-	0.3	0.4
GIDS [RMSE cm].	8.3	-	-	-	-	-	-	-	-	-	3.6	1.9
GIDD [RMSE cm].	11.9	-	-	-	-	-	-	-	-	-	3.3	1.8
GIDD* [RMSE cm].	11.7	-	-	-	-	-	-	-	-	-	3.2	1.7
PS + GIDD [RMSE cm].	2.7	3.6	-	-	-	5.2	-	-	-	-	0.2	0.7
PS + GDD [RMSE cm].	2.3	3.3	-	-	-	5.4	-	-	-	-	0.2	0.6

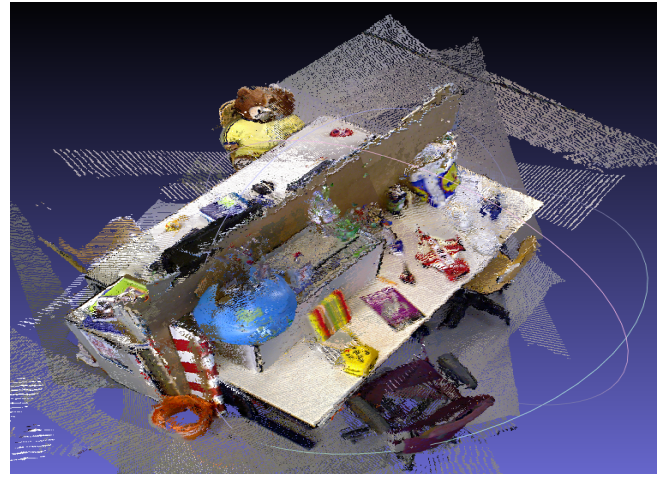
#Seq.	25	26	27	28	29	30	31	32	33	34	35	36
PS [RMSE cm].	-	5.3	2.7	-	-	3.3	1.5	7.5	-	1.2	1.3	-
PD [RMSE cm].	-	-	-	-	-	-	13.5	2.9	-	2.0	1.2	-
GIDS [RMSE cm].	9.7	-	-	-	-	-	-	4.2	3.3	3.4	6.5	-
GIDD [RMSE cm].	-	-	-	-	-	-	-	8.0	3.0	4.0	5.1	-
GIDD* [RMSE cm].	-	-	-	-	-	-	-	7.1	2.6	3.3	6.2	-
PS + GIDD [RMSE cm].	5.7	7.0	2.7	-	-	2.6	1.0	1.3	4.4	1.0	1.0	-
PS + GDD [RMSE cm].	8.8	8.5	2.7	-	-	3.5	1.1	1.4	4.0	0.9	1.2	-

TABLE IV: RMSE for different RGBDTAM configurations. *PS* stands for photometric semi-dense, and *PD* for photometric dense. *GIDD* and *GIDD** stand for geometric inverse depth dense. *GIDD* only uses a subsample of the points (homogeneously distributed). *GDD* stands for geometric depth dense. *GIDS* stands for geometric inverse depth semi-dense. The combination of a semi-dense photometric and a dense –subsampled– geometric is the most accurate and robust.

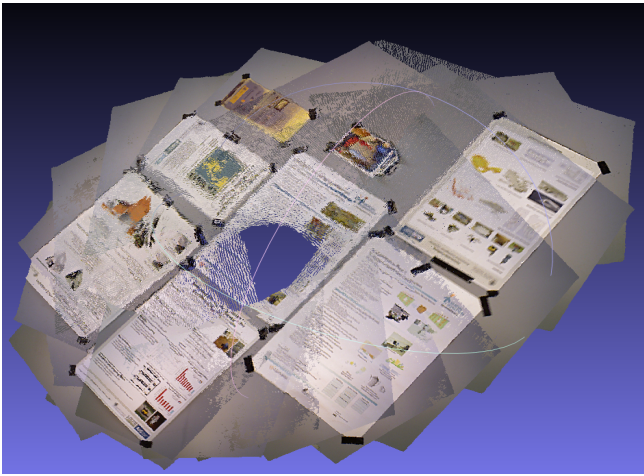
- [10] M. A. Fischler and R. C. Bolles. Random sample consensus, a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381 – 395, 1981.
- [11] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation*, pages 15–22, 2014.
- [12] Dorian Galvez-Lopez and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012.
- [13] G Grisetti, H Strasdat, K Konolige, and W Burgard. g2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation*, 2011.
- [14] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Indoor scene understanding with RGB-D images: Bottom-up segmentation, object detection and semantic segmentation. *International Journal of Computer Vision*, 112(2):133–149, 2015.
- [15] Daniel Gutiérrez-Gómez, Walterio Mayol-Cuevas, and JJ Guerrero. Inverse depth for accurate photometric and geometric error minimisation in RGB-D dense visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 83–89, 2015.
- [16] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, 2012.
- [17] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.
- [18] Mariano Jaimez, Christian Kerl, Javier Gonzalez-Jimenez, and Daniel Cremers. Fast Odometry and Scene Flow from RGB-D Cameras based on Geometric Clustering. In *Proc. International Conference on Robotics and Automation (ICRA)*, 2017.
- [19] Christian Kerl, Jurgen Sturm, and Daniel Cremers. Dense visual SLAM for RGB-D cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106, 2013.
- [20] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for RGB-D cameras. In *2013 IEEE International Conference on Robotics and Automation*, pages 3748–3754, 2013.
- [21] 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.
- [22] Sebastian Klose, Philipp Heise, and Alois Knoll. Efficient compositional approaches for real-time robust direct visual odometry from RGB-D data. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1100–1106, 2013.
- [23] K. Lai, L. Bo, X. Ren, and D. Fox. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In *IEEE International Conference on Robotics & Automation (ICRA)*, 2011.
- [24] Maxime Meilland and Andrew I Comport. On unifying key-frame and voxel-based dense visual SLAM at large scales. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3677–3683, 2013.
- [25] Maxime Meilland and Andrew I Comport. Super-resolution 3D tracking and mapping. In *Robotics and Automation (ICRA)*, 2013 *IEEE International Conference on*, pages 5717–5723. IEEE, 2013.
- [26] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [27] Raúl Mur-Artal and Juan D Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 2017.
- [28] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR)*, 2011 *10th IEEE international symposium on*, pages 127–136. IEEE, 2011.
- [29] 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*, pages 2320–2327, 2011.



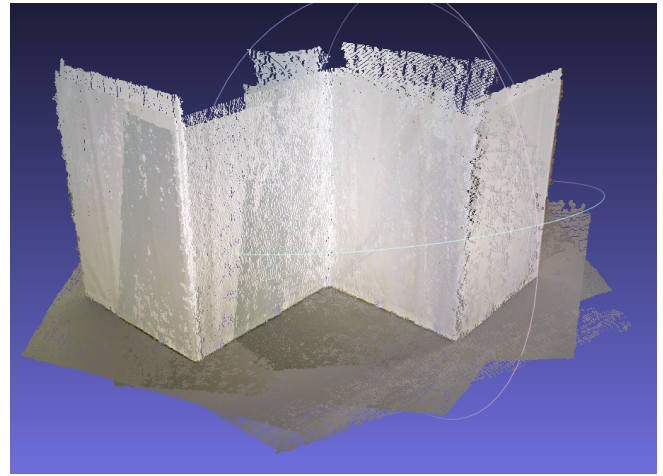
(a) Sequence fr2 rpy



(b) Sequence fr3 household long office



(c) Sequence fr3 no structure texture near with loop



(d) Sequence fr3 structure no texture far

FIG. 2: Qualitative results. Depth maps are not fused. They are back projected from every keyframe.

- [30] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 IEEE international conference on Computer Vision*, pages 2564–2571, 2011.
- [31] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [32] Luciano Spinello and Kai O Arras. People detection in RGB-D data. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3838–3843, 2011.
- [33] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Real-time visual odometry from dense RGB-D images. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 719–722. IEEE, 2011.
- [34] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012.
- [35] Jesus Suarez and Robin R Murphy. Hand gesture recognition with depth images: A review. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 411–417. IEEE, 2012.
- [36] Tommi Tykkälä, Cédric Audras, and Andrew I Comport. Direct iterative closest point for real-time visual odometry. In *2011 IEEE International Conference on Computer Vision Workshops*, pages 2050–2056, 2011.
- [37] Thomas Whelan, Hordur Johannsson, Michael Kaess, John J Leonard, and John McDonald. Robust real-time visual odometry for dense RGB-D mapping. In *2013 IEEE International Conference on Robotics and Automation*, pages 5724–5731, 2013.
- [38] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous: Spatially extended KinectFusion. 2012.
- [39] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.