Layout Aware Visual Tracking and Mapping

Marta Salas*, Wajahat Hussain*, Alejo Concha, Luis Montano, Javier Civera, J. M. M. Montiel

***Abstract*— Nowadays real time visual Simultaneous Localization And Mapping (SLAM) algorithms exist and rely on consistent measurements across multiple views. In indoor environments, where majority of robot's activity takes place, severe occlusions can occur, e.g., when turning around a corner or moving from one room to another. In these situations, SLAM algorithms can not establish correspondences across views, which leads to failures in camera localization or map construction.**

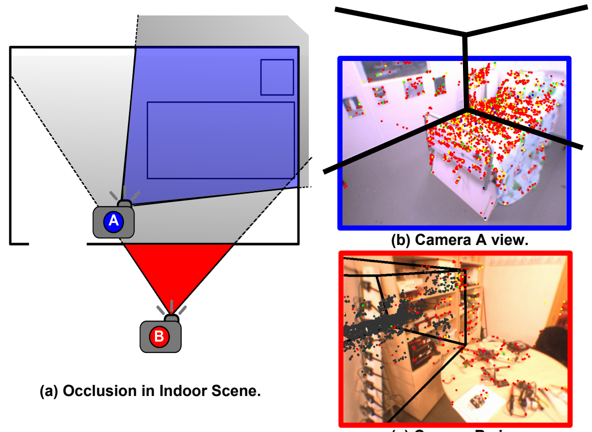
This work takes advantage of the recent scene box layout descriptor to make the above mentioned SLAM systems occlusion aware. This room box reasoning helps the sequential tracker to reason about possible occlusions and therefore look for matches in only potentially visible features instead of the entire map. This increases the life of the tracker, as it does not consider itself lost under the occlusion state.

Additionally, focusing on the potentially visible portion of the map, i.e., the current room features, it improves the computational efficiency without compromising the accuracy. Finally, this room level reasoning helps in better image selection for bundle adjustment. The image bundle coming from the same room has little occlusion, which leads to better dense reconstruction. We demonstrate the superior performance of layout aware SLAM on several long monocular sequences acquired in difficult indoor situations, specifically in a room-room transition and turning around a corner.

I. Introduction

Simultaneous Localization And Mapping (SLAM) from visual sensors has shown great progress in recent years and real-time solutions are at hand nowadays [1], [2], [3], [4]. These systems are mostly based on the geometric consistency of the visual features –points, lines or objects– across multiple views. Geometric consistency means that visual features will find correspondences near their predicted positions. False tracks or detections are unlikely to have this consensus over a large number of observations (images).

This geometric consistency only holds in occlusion-free scenes. As the complexity of the scene grows, occlusions are likely to appear, and the standard SLAM systems might be confused about the state of unfounded correspondence. It may be a false track or a temporary occlusion of a tracked feature. This confusion is usually resolved using heuristics, .e.g., a feature is considered a false track if the ratio of its correspondence-count versus its visible-count is low [1], [5]. This rule assumes a low-degree occlusion and mainly holds for a small-scale translation, an almost-concave room and a wavy camera motion producing quite redundant sequences. Abrupt scene changes (e.g., going out of a room) will make these heuristics, and thus state-of-the-art SLAM systems break down.



(a) Occlusion in Indoor Scene. (b) Camera A view. (c) Camera B view.

Fig. 1. Occlusion in indoor scene causing confusion. **a)** Scenario where the camera starts at *location A*, moves out of the room and points towards the wall of the same room at *location B*. **b)** Camera view at *location A*. The visible map sparse points (RGB points) and the estimated room 3D box layout (black lines) are projected on the image. **c)** Camera view at *location B*. It shows severe occlusion of the map points (grey points) inside the room. Due to our box reasoning, SLAM algorithm is able to reason about the occlusion and proceed with tracking the camera instead of considering itself lost.

Fig. 1 shows a frequently occurring situation in mobile robotics. The camera starts building a sparse map at *location A*. The tracked map points are projected as RGB points in Fig. 1 b. The camera leaves the room and reaches *location B* just outside the room. Now the map points, from inside the room, are unable to find correspondences due to occlusions from the room boundary. The occluded points are shown as gray points in Fig. 1 c. The major portion of the map can not be tracked and the standard SLAM systems consider this as a *lost* state. The ability to recover from this lost state is vital considering the amount of time spent indoors by robots.

We leverage the recent scene 3D box layout descriptors [6], [7], [8], [9], [10], to overcome this confusion. These box layout descriptors model the scene as a set of dominant planes, e.g., cuboid rooms are composed of 6 orthogonal planes. Relaxing the orthogonality incorporates non-cuboid rooms, e.g., at least ceiling-floor are orthogonal to the walls of the room [11], [12]. This scene summarization into dominant planes incorporates the majority of the daily life scenes. For example, a street scene can be summarized by five dominant planes [13].

Similar to [14], [11], [12], we estimate this room 3D box layout from the initial sparse map and the image content as described later on. Fig. 1 b and 1 c show the estimated room

*These authors contributed to the present work in equal parts and share first authorship.

**This research has been funded by the Dirección General de Investigaciń of Spain under projects DPI2012-32168 and DPI2012-32100.

The authors are with the Aragón Institute of Engineering Research (I3A), Universidad de Zaragoza, Spain {msalasg,hussain,alejocb, montano, jcivera,josemari}@unizar.es

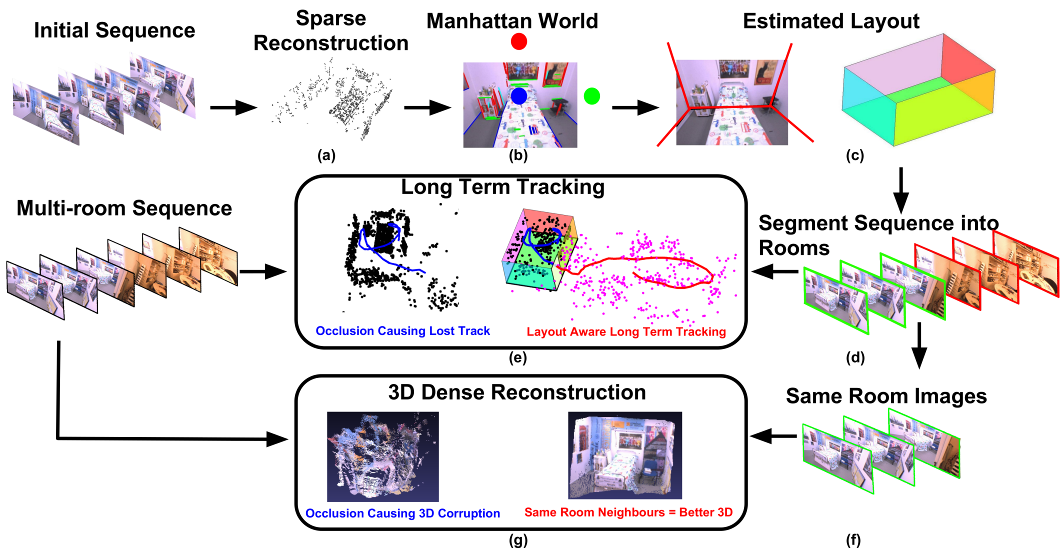


Fig. 2. Overview of our approach, the use of the 3D box layout information makes the tracking and the mapping more robust. **a)-c)** 3D box layout estimation. **d)** The estimated layout segments the images of the sequence into rooms. **e)** Standard SLAM loses tracking due to occlusion, while layout-aware SLAM results in a long-term camera tracking since it can reason about the occluded points inside the room (black) and the visible points outside the room (magenta). **f)** Layout aware selection of image bundle (neighbouring frames) for dense reconstruction. **g)** naïve selection of neighbouring frames, spanning different rooms, suffers from occlusion corrupting the 3D reconstruction. Layout-awareness helps in selecting images from the same room which reduces the occlusion and results in a better dense reconstruction. (Best viewed in color)

layout in black lines. Given this box layout prior, our layout aware SLAM is able to distinguish between the occluded and the false tracks. This aids the system to recover from the lost state.

Our contribution is the introduction of this box layout reasoning in the sequential mapping process which has threefold advantages.

1) **Long Term Tracking**. The tracker is able to distinguish between the occluded and the false tracks. This makes tracker confident in difficult situations, e.g., room transitions, turning around a corner.
2) **Light Weight Mapping**. This room box level reasoning allows to focus on only the current room features instead of the entire map. This increases the computational efficiency without reducing the accuracy of the map.
3) **Better Dense Reconstruction**. Random image bundle selection for dense reconstruction suffers from occlusion. Our system selects images from the same room which avoids this occlusion problem.

The reader can look at the video (`http://webdiis.unizar.es/~msalasg/videos/IROS.mp4`) for a quick overview. The rest of the paper is organized as follows. Section II describes the related work. An overview of our proposal is given in section III, and a more detailed view in section IV. Section V shows the experimental results and, finally, section VI concludes and points out the lines for future work.

II. Related Work

SLAM methods can be based on sparse features as in [1], or dense tracking as in [3]. These algorithms process the video stream sequentially, which means they can only use images up to the present. There are offline reconstruction methods ([15], [16]) which might be more resilient to occlusion since they process the images in a batch. Our aim is online estimation and leveraging the exciting advances in single image scene understanding.

The single image scene layout methods model the room as a set of dominant planes [6], [17], [18]. The indoors consist mainly of cuboid rooms with 6 planes, i.e., ceiling, floor and the four walls. Given this room box layout, it becomes easier to reason about the remaining content in the room, e.g., objects are found mainly on the floor, doors are found in the walls [19]. We are not the first ones to fuse this scene layout estimation with multi-view geometry. As shown in [14], [11], [12], multiview geometric constraints improve the robustness of the scene 3D layout. In our work, the flow of information is in the opposite direction, i.e., room layout reasoning aids the multiview sequential mapping. Alejo et al. [20] use this 3D box layout, in combination with the dense geometric labels in the 2D image space, to improve the dense reconstruction, provided by DTAM [3], in texture-less areas. Their reconstruction is limited to single room scenarios. We utilize the 3D box layout to select images coming from the same room, which improves the DTAM reconstruction in multi-room scenario. In an interesting work Liu et al. [21] use this 3D layout to localize the camera in the 2D floor plan. In our work, we do not assume the availability of a known map, i.e., floor plan, to improve monocular mapping.

In a few notable works object level semantic reasoning is fused with the multiview geometry. Bao et al. [22] propose the joint estimation of a sparse set of salient points and the objects from multiple views. In that work, the geometric constraints help to remove the false object detections and the detections, in turn, provide constraints to improve the camera pose estimation. Similarly Salas-Moreno et al. [4], used object detection for loop closure resulting in improved camera poses. Complimentary to object level semantic reasoning, our proposal introduces for the first time scene level semantic constraints to improve the geometric mapping estimation.

With the ability to segment the data coming from different rooms, our system manages to focus on the current room reducing the complexity of the method. Other efficient submapping techniques exist, but they either need the complete graph to reason on the partition or are based on heuristics [23], [24]. Our layout aware mapping method utilizes the scene understanding to make these decisions in an online manner.

III. Overview

The overview of our pipeline is shown in Fig. 2. It takes a moving camera video stream of a room as input and builds a map composed of a set I of N geometrically well-conditioned image keyframes $I = \{I_1, I_2..I_N\}$ with camera poses $C = \{C_1, C_2..C_N\}$ and a sparse set $\mathbf{X}$ of P points $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2..\mathbf{X}_P\}$ using PTAM [1] (Fig. 2 a). The camera pose $C_j = [R_j|\mathbf{t}_j]$ is a 3x4 matrix, where R is the 3x3 rotation matrix and $\mathbf{t}$ is the 3x1 translation vector. We assume that the camera is placed inside a cuboid room which is mostly opaque (no glass walls). Under these assumptions, the tracked map points $\mathbf{X}$, after the little initial motion, largely belong to this room (Fig. 2 a).

This PTAM output is utilized to build the room 3D box layout, M. As mentioned earlier, this 3D box layout is composed of 6 planes (Fig. 2 c). In order to estimate this 3D box layout, we try to fit the planes to the initial sparse point cloud, by leveraging the Manhattan world assumption [25], i.e., the entities are aligned along three orthogonal directions. These Manhattan plane orientations are provided by clustering the detected line segments in images (RGB line segments in Fig. 2 b), into 3 dominant clusters (RGB dots in Fig. 2 b).

So far, most of the map content (keyframes, tracked points) belongs to this first room. As there is little content visible from the outside world, there is little occlusion. Therefore, for this first room, our policy is to track the entire map content, what allows the camera to transit from the first room to the outside environment. The transition happens when the camera crosses the boundary of the already estimated 3D box layout. At the transit point, we place a door quadrilateral (white quadrilateral in Fig. 2 e) of standard size to remember the entry/exit point. Given this 3D box layout, our system is able to distinguish between the tracked points (Fig. 2 f) and the keyframes (Fig. 2 d) of the first room from that of the current environment.

Our policy for the current room is to only track the points outside the first room (magenta points in Fig. 2 f) and the points visible from the first room through the door quadrilateral. Without this map segmentation, the tracker is confused by the occlusion from the room boundary and hence considers itself lost (terminated blue camera track in Fig. 2 f). Secondly, an image bundle coming from the same room (green keyframes in Fig. 2 f) has little occluded common content. This improves the dense reconstruction (Fig. 2 f). Finally, with this room level map segmentation, we can concentrate our optimization on the current room instead of the whole building. This enables efficient map management.

Algorithm 1 Algorithm for multiview layout estimation

INPUT: $\mathbf{X} = \{X_1, X_2..X_P\}$ *% Sparse point set*
$C = \{C_1, C_2..C_N\}$ *% Camera poses*
$I = \{I_1, I_2..I_N\}$ *% Keyframe Images*
K *% Camera calibration*
OUTPUT: $M = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, \boldsymbol{\pi}_4, \boldsymbol{\pi}_5, \boldsymbol{\pi}_6\}$ *where* $\boldsymbol{\pi} = [\mathbf{n}\ d]^T$

Get Manhattan directions
for $j = 1$ to N **do**
  $[\mathbf{v}_1^j, \mathbf{v}_2^j, \mathbf{v}_3^j] = get\_image\_vanishing\_points(I_j)$
  $[\mathbf{V}_1^j, \mathbf{V}_2^j, \mathbf{V}_3^j] = R_j^{-1} K^{-1} [\mathbf{v}_1^j, \mathbf{v}_2^j, \mathbf{v}_3^j]$ *%Project into 3D*
end for
$[\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3] \leftarrow [\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3]$ *where* $\mathbf{n}_k = med([\mathbf{V}_k^1, .., \mathbf{V}_k^N])$
Get room plane hypotheses
for $k = 1$ to 3 **do** *%loop over Manhattan directions*
  for all $\mathbf{X_i}$ in $\mathbf{X}$ **do** *%loop over all points*
    $d_{ki} = \mathbf{n}_k^T \mathbf{X}_i = n_x.X_{ix} + n_y.X_{iy} + n_z.X_{iz}$
  end for
  $[d_M^k, d_m^k] = [max(d_{ki}), min(d_{ki})]$
  $[\boldsymbol{\eta}_{k1}..\boldsymbol{\eta}_{kD}] = get\_equally\_spaced\_planes(\mathbf{n}_k, [d_M^k, d_m^k])$
  Prune valid planes
  for $r = 1$ to D **do** *%loop over plane hypotheses*
    $score_r = get\_plane\_hypothesis\_score(\mathbf{X}, \boldsymbol{\eta}_{kr})$
    if $score_r > 6$ **then**
      $\boldsymbol{\pi}_k^{valid} \cup \boldsymbol{\eta}_{kr}$ *%* $\boldsymbol{\pi}_k^{valid}$ *is the set of valid planes*
    end if
  end for
  Select two extreme planes in opposite direction
  $\boldsymbol{\pi}_1^k = get\_plane\_with\_max\_d(\boldsymbol{\pi}_k^{valid})$ *% one wall*
  $\boldsymbol{\pi}_2^k = get\_plane\_with\_min\_d(\boldsymbol{\pi}_k^{valid})$ *% opposite wall*
end for
Output
$\{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, \boldsymbol{\pi}_4, \boldsymbol{\pi}_5, \boldsymbol{\pi}_6\} \leftarrow \{\boldsymbol{\pi}_1^1, \boldsymbol{\pi}_2^1, \boldsymbol{\pi}_1^2, \boldsymbol{\pi}_2^2, \boldsymbol{\pi}_1^3, \boldsymbol{\pi}_2^3\}$

FUNCTION $get\_plane\_hypothesis\_score(\mathbf{X}, \boldsymbol{\eta})$
for all $\mathbf{X_i}$ in $\mathbf{X}$ **do** *%loop over all points*
  $\tilde{\mathbf{X}}_i = [\mathbf{X}_i|1]$ *% point in homogeneous coordinates*
  $d_i = \boldsymbol{\eta}^T \tilde{\mathbf{X}}_i = n_x.X_{ix} + n_y.X_{iy} + n_z.X_{iz} + d$
  if $abs(d_i) < \varepsilon$ **then**
    score ++
  end if
end for

IV. Occlusion Aware Tracking and Mapping

A. Layout Estimation M

At this stage, after the short camera motion inside the room, PTAM provides us with the sparse point cloud $\mathbf{X}$ and the keyframe images I with their poses C. Tsai et al. [11] have shown that this small initial camera motion of ∼7-8 seconds and the above mentioned multi-view data is enough to estimate the room layout in real time. We use a modified version given in Algorithm 1.

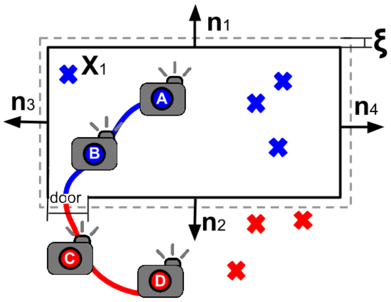


Fig. 3. Room layout enables the capability to segment the map components coming from different rooms. Layout is represented as a black box, its planes normal vectors point towards outside the room. The cameras A and B, and blue map points, which are inside the room, are separated from cameras C and D, and red map points which are outside the room. ξ is a threshold which defines room transition.

Similar to prior work, we model the room as a Cuboid. The room 3D box layout M is composed of six planes $M = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, \boldsymbol{\pi}_4, \boldsymbol{\pi}_5, \boldsymbol{\pi}_6\}$. Every plane $\boldsymbol{\pi} = [\boldsymbol{n} \;\; d]$ is parametrized by its plane normal unit vector $\boldsymbol{n}$ and the distance to the origin d. In order to fit these planes to the sparse point cloud, we need to know the plane normal orientation. Indoors, the dominant planes are aligned along 3 directions. These directions are provided by the detected 2D line segments in the keyframe images I. Look at the RGB lines segments in Fig. 2 b. The RGB lines segments form 3 dominant clusters. The weighted centroids of these clusters, known as vanishing points $[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$, are shown as RGB points in Fig. 2 b. We use the code provided by [6] to estimate these points for all keyframes images I. We back project these points to get the 3 vanishing directions $[\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3]$ where $\mathbf{V} = R_j^{-1} K^{-1} \mathbf{v}$. Next we take the median of each vanishing direction from all keyframes, i.e., $\mathbf{n}_k = med([\mathbf{V}_k^1, \mathbf{V}_k^2, \ldots, \mathbf{V}_k^N])$, to provide robustness against single image vanishing point estimation failure.

Next we generate equally spaced D plane hypotheses per orientation $\mathbf{n}$. The range bounds for these hypotheses are given by the max and the min value of $d_k = \{\boldsymbol{n}_k \boldsymbol{X}_i \;\; \forall i\}$. $D = 25$ in our experiments. The score of each plane hypothesis $\boldsymbol{\eta}$ is the count of points that lie close to it, i.e., $||\boldsymbol{\eta}^T \tilde{\mathbf{X}}|| < \varepsilon$.

Given the pruned hypotheses for each orientation, we select the two furthest planes in opposite directions. These planes correspond to opposite entities of the room, i.e., floor-ceiling, right-left wall and front-back wall. For a detailed online layout estimation method have a look at Tsai et al. [11].

B. Layout-Conscious Parallel Tracking and Mapping

Tracking and mapping process are carried out simultaneously. Tracking the camera against a known map involves predicting a pose of the camera and establishing correspondences of the current image with the tracked map points. A standard solution to consider the tracking lost, while exploring the environment, is to set a threshold on the fraction of the predicted visible points which found correspondences. When a sudden occlusion happens, like in the two rooms scenario we are proposing, most of the map points are occluded once the camera leaves the room and faces towards one of the room walls. While the camera is inside the room, all the features are tracked. However, when the camera crosses one of the walls, current camera pose $C_c = [R_c | \mathbf{t_c}]$ is computed using the tracked points which satisfy the condition:

$$\boldsymbol{\pi}^\top \tilde{\mathbf{X}}_i > \xi \vee (\boldsymbol{\pi}^\top \tilde{\mathbf{X}}_i < \xi \wedge proj(\mathbf{C}_c, \mathbf{X}_i) \in door) \quad \forall \boldsymbol{\pi} \in M \tag{1}$$

where $\tilde{\mathbf{X}}_i$ is homogeneous representation of $\mathbf{X}_i$. ξ is a threshold which defines the room transition, with $\xi = 0.01$ in our experiments. This condition checks, given that the current camera pose C_c, predicted according to a velocity model, is out the room ($\boldsymbol{\pi}^\top \tilde{\mathbf{t}}_c > \xi$), which map points are outside the room M or are visible through the door quadrilateral. Note that point-wall and camera-wall distances are all negative because layout normals point towards outside the room. Given that the current camera is *camera C* (Fig. 3), this condition holds for all the points colored in red and the blue point ($\mathbf{X}_1$) visible through the door. While standard algorithms (e.g. [1]) would consider the camera lost, our room layout helps to distinguish between those points spanning different rooms.

Simultaneously, a map is built using a Bundle Adjustment optimization, which minimizes the reprojection error using the Levenberg-Marquardt algorithm:

$$\{\{C_1..C_N\}\{\boldsymbol{X}_1..\boldsymbol{X}_P\}\} = \underset{\{\{C\}\{\boldsymbol{X}\}\}}{\arg\min} \sum_{i=1}^{P} \sum_{j=1}^{N} \boldsymbol{e}_{ji}^T \Lambda \boldsymbol{e}_{ji}, \tag{2}$$

where the reprojection error $\mathbf{e}_{ji}$ for the 3D point $\mathbf{X}_i$ in camera C_j with 2D measurement $\mathbf{x}_{ij}$ is

$$\boldsymbol{e}_{ij} = l \cdot (\mathbf{x}_{ij} - proj(C_j, \mathbf{X}_i)), \tag{3}$$

where l is an indicator function that models the occlusion

$$l = \begin{cases} 1 & \text{if } \boldsymbol{\pi}^\top \tilde{\mathbf{X}}_i < \xi \text{ and } \boldsymbol{\pi}^\top \tilde{\mathbf{t}}_j < \xi \;\; \forall \boldsymbol{\pi} \in M \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The indicator function checks point-wall and camera-wall distances and excludes those map points and cameras that do not belong to the same room M from the optimization. This layout based segmentation of the entities, coming from different rooms, sets an upper bound to the complexity of the optimization to single room complexity for indoor scenes with multiple rooms. Given that current camera is the *camera D* (Fig. 3), only camera poses and map points coming from outside the room are optimized (red entities in Fig. 3).

C. Layout-Conscious Dense Reconstruction

The dense reconstruction, given that the camera poses are known, is computed as a minimization of the weighted sum of a photometric error data term $C(\mathbf{u}, \rho)$ and a regularization term $G(\mathbf{u}, \rho)$ for every pixel $\mathbf{u}$ in the reference image I_r and its corresponding pixel in every other image I_j at a hypothesized inverse distance ρ. This equation is solved using the variational method proposed in DTAM [3]

$$\rho = \underset{\rho}{\arg\min} \sum_{\mathbf{u}} (\lambda_1 C(\mathbf{u}, \rho) + G(\mathbf{u}, \rho)), \tag{5}$$

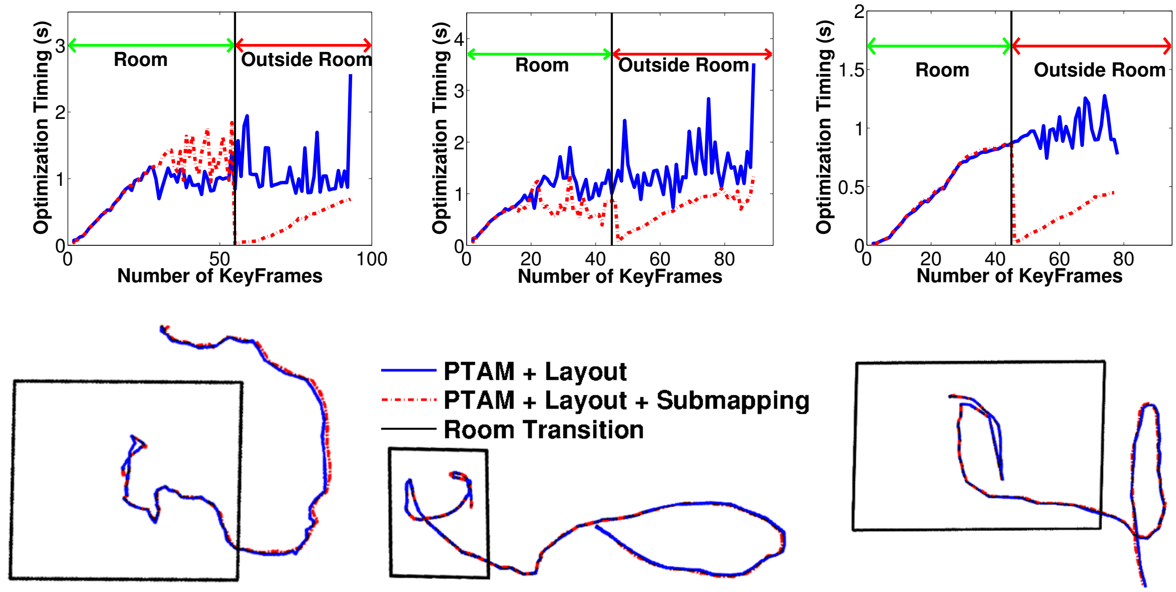


Fig. 4. **Efficient map management using the 3D box layout reasoning**. We segment the cameras (**C**) and the sparse tracked points (**X**) belonging to the different rooms using the layout. This allows updating the map of the current room only instead of the whole buidling. **1st row** shows the complete optimization timing for the three sequences spanning 2 rooms with (PTAM + Layout + submapping) and without (PTAM + Layout) map splitting. The red dotted lines show that our map splitting is atleast 2 times faster compared to the blue lines at the end of the sequences. **2nd row** shows that this map splitting does not degrade the camera pose accuracy. Similarity makes the red and the blue tracks appear as one track. (best viewed in color)

where λ_1 is a weighting factor that accounts for the relative importance of the photometric and regularization terms, and the regularizer is based on the Huber norm of the gradient of the inverse depth map $||\nabla\rho(\mathbf{u})||_{\varepsilon}$. The regularization term handles the outliers and assumes the scene evolves smoothly. When a heavy occlusion occurs this smooth condition does not hold and this term can not handle it.

The photometric error is the summation of the color error between every pixel in the reference image I_r and its corresponding in the neighbour image I_j

$$C(\mathbf{u},\rho(\mathbf{u})) = \frac{1}{|Z|}\sum_{j=1, j\neq r}^{N} ||l_d \cdot (I_r(\mathbf{u}) - I_j(T_{rj}(\mathbf{u},\rho)))||_1, \quad (6)$$

where T_{rj} defines a relative transformation for every pixel $\mathbf{u}$ of the reference image, which is first backprojected at the inverse distance ρ, and then, it is projected again in every close j-th image that is in the same room, which is given by an indicator function l_d. Z is a normalization factor given by the number of images taken into account in the summation.

$$l_d = \begin{cases} 1 & if\ \boldsymbol{\pi}^\top\tilde{\mathbf{t}}_r < \xi\ and\ \boldsymbol{\pi}^\top\tilde{\mathbf{t}}_j < \xi \quad \forall\boldsymbol{\pi}\in M \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

This indicator function excludes those cameras which are not in the same room M as the reference one. Given that *camera B* is the reference frame (Fig. 3), only cameras on the blue trajectory, e.g., *camera A*, are used for the dense reconstruction.

V. EXPERIMENTS

Four RGB image sequences, around one minute long each, have been acquired with a moving camera in indoor scenarios. Although some public datasets are available (e.g. [26] and [27]), these datasets provide sequences for single room scenarios only.

Our sequences span more than one room to demonstrate the occlusion effect. The rooms are approximately cuboid. We compare our layout-conscious tracking and mapping with the two state-of-the-art baselines in sequential mapping: PTAM [1] and our DTAM [3] mapping implementation. A video summarizing our experiments is attached as supplementary material.

A. Long-term Tracking Using the 3D Layout Reasoning

Fig. 6 shows the camera tracking results for three sequences, each spanning two rooms to show the occlusion effect. There are three scenarios, i.e., laboratory+corridor, bedroom+laboratory and class+coffee room.

In each sequence, after mapping the first room, where the layout is estimated using this map, the camera moves to the next room. While in the next room, when the camera points towards the first room it faces heavy occlusion due to room walls (1^{st} Seq. Image 4, 2^{nd} Seq. Image 5 and 3^{rd} Seq. Image 6). The room layout (black lines) segments the points present in different rooms. The points in the current room are colored and the points from the previous room are black.

This discrimination ability allows confident camera tracking under occlusion, e.g, around corners or while leaving the room. In absence of this room layout information, the inability to correspond the map points leads to a tracking failure (images labelled L in Fig. 6).

Table I shows that the layout conscious PTAM becomes more robust and ends up successfully tracking more frames.

B. Efficient Map Management Using the 3D Layout Reasoning

PTAM can build real-time sequential maps of small scale scenarios limited to single-rooms. Building real-time maps of

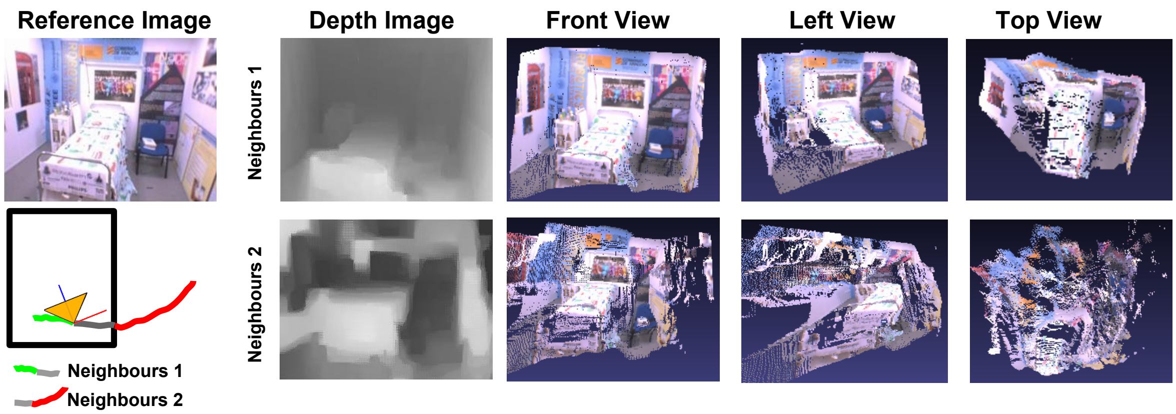


Fig. 5. **Better dense reconstruction using the 3D box layout reasoning**. This figure shows how the occlusion affects the 3D dense reconstruction. The goal is to reconstruct every pixel in a reference image. Few neighbouring images, to this reference image, are selected to compute pixel correspondences. The layout figure (**2nd row, 1st column**) shows the reference camera location (yellow triangle) with 2 neighbour sets, i.e., set 1(green+gray), set 2(blue+gray). **Row 1** shows the result of our layout-aware neighbour selection (set 1). **Row 2** shows the reconstruction result, using layout-unaware neighbour selection (set 2), that included several images outside the room having partial occlusion. The improvement of our proposal is visually noticeable.

Sequence	# frames in the Seq.	# tracked frames with PTAM	# tracked frames with PTAM+Layout (**ours**)
lab	2150	1250	2150
bedroom	2520	1624	2520
classroom	1980	1311	1980

TABLE I

NUMBER OF TRACKED FRAMES WITH AND WITHOUT 3D BOX LAYOUT REASONING IN PTAM.

scenes composed of more than single room requires efficient map management.

We use our layout aware formulation to segment the map entities belonging to the different rooms in order to partition the estimation semantically. These layout-aware partitions set an upper bound on the complexity of our algorithm to the size of the largest room. By comparing the Bundle Adjustment (BA) optimization timing (Fig. 4, 1^{st} row) of the whole map versus the layout-conscious BA (explained in Section IV-B), it can be seen that our proposal is at least twice faster than the baseline in all the cases. When comparing the estimated camera trajectories of both methods (Fig. 4, 2^{nd} row), it is seen that this efficiency is achieved without degrading the performance. Note that while in the room the optimization timing might be different due to a different number of map points on each experiment.

There are other efficient map management methods that segment the maps based on heuristics, e.g., start a new map after X sec or Y features [23], [24]. In our proposal, the submapping is based on the semantic content of the scene. For heuristics based submapping, the related entities exist in different maps. Thus, efficiency is achieved by segmenting these related entities.

C. Better Dense Reconstruction Using the 3D Layout Reasoning

As explained in Section IV-C, in order to estimate a dense reconstruction a reference image and a set of neighbouring images are selected to optimize according to equation 5. The selection of the neighbours affects the quality of the reconstruction, as shown in Fig. 5.

We run our layout aware dense reconstruction algorithm over a sequence with 348 frames, which was recorded in the bedroom-laboratory scenario shown in Section V-A. The top view (Fig. 5, 1^{st} column, 2^{nd} row) of the layout figure shows the location of the reference image as a yellow triangle, it corresponds to frame number 33 of the sequence.

First, we consider a neighbourhood inside the room (frames from 1 to 55) given that the box layout is estimated and we can distinguish those frames which span different rooms. The resulting reconstruction along with novel views is shown in Fig. 5, 1^{st} row. Then, we consider a neighbourhood spanning different rooms (frames 28 to 348) as if the box layout was unknown. The resulting 3D reconstruction is affected by the occlusion (Fig. 5, 2^{nd} row).

The results show that, qualitatively, our layout aware approach is able to produce an accurate reconstruction, fixing the noticeable degradation, due to occlusion, of the standard formulation. Although the neighbourhood set 2 represents an extreme case, crossing room boundaries is a frequently occurring event for indoor robotics.

VI. CONCLUSION

We have shown in this paper, for the first time, how leveraging scene level reasoning improves the quality of online geometric mapping. Our experiments show that room level reasoning improves tracking, submapping and dense reconstruction over state of the art.

So far we have tested this 3D layout based occlusion reasoning in a very simple setting of a single cuboid room plus a small external environment. The indoor scenes can be more complex than simple cuboid representation. There are few recent works which have shown robust layout estimation in complex indoor scenarios, i.e., connecting corridors [14] and non-cuboid scenes [11], [12]. Future work includes fusing these robust layout methods with online geometric mapping. In this work our focus is robust camera tracking for feature-based mapping (PTAM) system. There are recent feature-less works, e.g. LSD SLAM, which might not be affected by occlusion as much as feature-based methods.

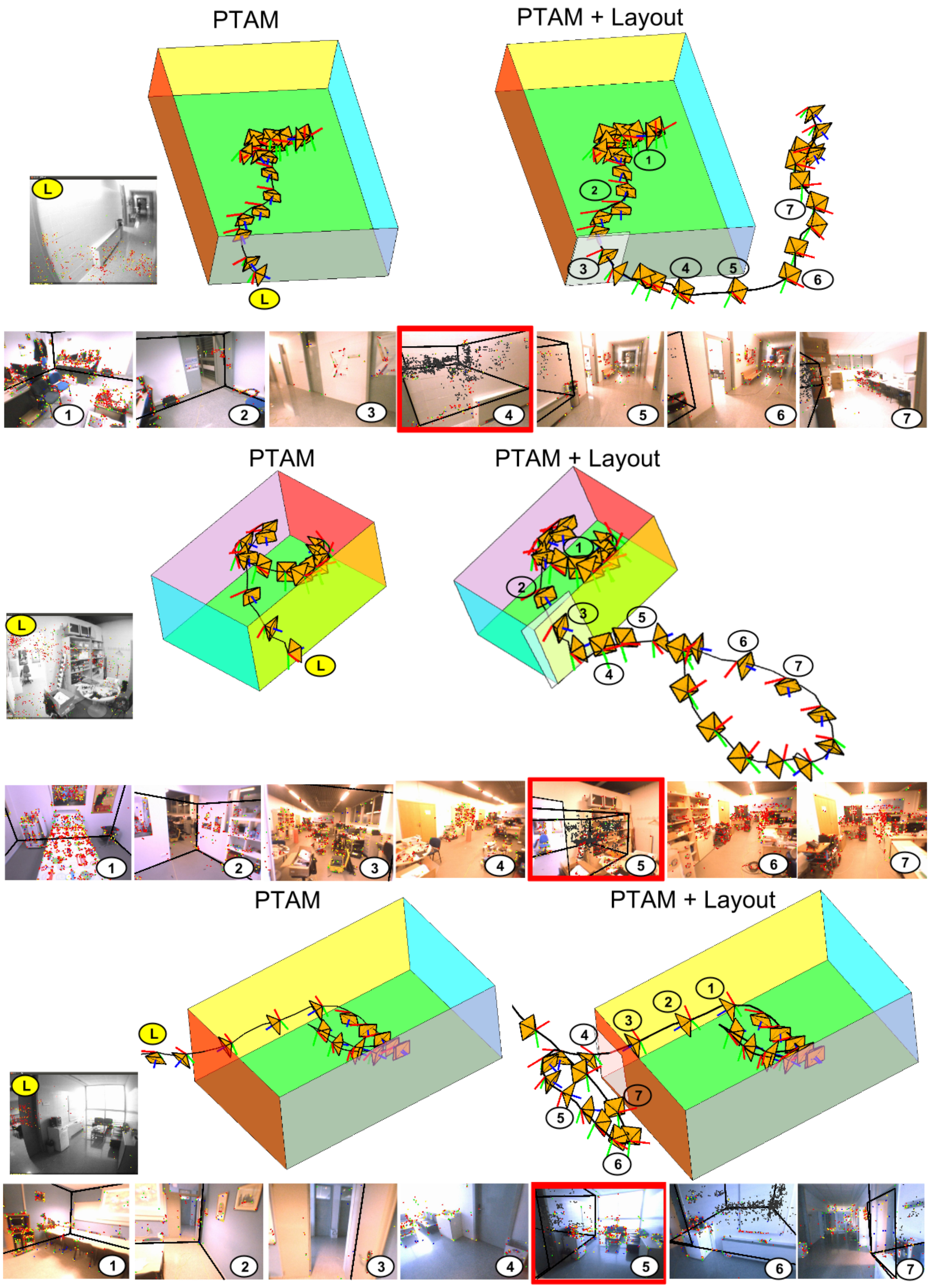


Fig. 6. **Long-term tracking using the 3D box layout reasoning**. The estimated room layout is shown as the colored 3D cuboid for the three sequences. Once the camera moves across the room boundaries, the tracked features are occluded. The image sequences below the 3D cuboids show the projected room layout (black lines), the features in the current room (RGB dots) and the occluded features from the previous room (gray dots). The standard tracking methods can not distinguish between the occluded and the non-occluded points and therefore are lost. Images with label **L** show PTAM searching for these occluded points elsewhere. Our occlusion reasoning provides long and confident camera tracking in these complex situations (red bordered frames). (Best viewed in color) Follow the link for a video showing these results: `http://webdiis.unizar.es/~msalasg/videos/IROS_tracking.mp4`

However, Mur et al. [28] recently showed that feature-based methods still outperform these feature-less direct methods in accurate geometric mapping. As this method [28] does not explicitly model room level occlusion, it can benefit from our proposal.

ACKNOWLEDGMENT

The authors would like to thank Yasir Latif for dataset collection, multiple iterations of the manuscript and insightful discussions. We are also grateful to the anonymous reviewers for improving the current work.

REFERENCES

[1] 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.

[2] G. Klein and D. Murray, "Improving the agility of keyframe-based slam," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, 2008.

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

[4] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *CVPR*, 2013.

[5] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, October 2008.

[6] 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.

[7] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade, "Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces." in *NIPS*, vol. 1, no. 2. Vancouver, BC, 2010, p. 3.

[8] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun, "Efficient structured prediction for 3d indoor scene understanding," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2815–2822.

[9] L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard, "Bayesian geometric modeling of indoor scenes," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2719–2726.

[10] S. Ramalingam, J. K. Pillai, A. Jain, and Y. Taguchi, "Manhattan junction catalogue for spatial reasoning of indoor scenes," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3065–3072.

[11] 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.

[12] A. Furlan, S. Miller, D. G. Sorrenti, L. Fei-Fei, and S. Savarese, "Free your camera: 3d indoor scene understanding from arbitrary camera motion," in *British Machine Vision Conference 2013*. BMVA (British Machine Vision Association), 2013.

[13] V. Nedovic, A. W. Smeulders, A. Redert, and J.-M. Geusebroek, "Stages as models of scene geometry," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1673–1687, 2010.

[14] 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)*. IEEE, 2011, pp. 2228–2235.

[15] N. Snavely, S. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, 2008.

[16] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Reconstructing building interiors from images," in *Proc. Int. Conf. on Computer Vision*, 2009, pp. 80–87.

[17] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese, "Understanding indoor scenes using 3d geometric phrases," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 33–40.

[18] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun, "Box in the box: Joint 3d layout and object reasoning from single images."

[19] D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 3–15, 2008.

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

[21] C. Liu, A. G. Schwing, K. Kundu, R. Urtasun, and S. Fidler, "Rent3d: Floor-plan priors for monocular layout estimation," in *CVPR*, 2015.

[22] S. Y. Bao and S. Savarese, "Semantic structure from motion," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2025–2032.

[23] H. Strasdat, A. J. Davison, J. Montiel, and K. Konolige, "Double window optimisation for constant time visual slam," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2352–2359.

[24] L. M. Paz, J. D. Tardós, and J. Neira, "Divide and conquer: Ekf slam in," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1107–1120, 2008.

[25] C. Rother, "A new approach to vanishing point detection in architectural environments," *Image and Vision Computing*, 2002.

[26] 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)*, 2012.

[27] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[28] R. Mur-Artal and J. Tardos, "Probabilistic semi-dense mapping from highly accurate feature-based monocular slam," in *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.