TT-SLAM: Dense Monocular SLAM for Planar Environments

Xi Wang, Marc Christie, Eric Marchand

TT-SLAM: Dense Monocular SLAM for Planar Environments

Xi Wang, Marc Christie, Eric Marchand

Abstract— **This paper proposes a novel visual SLAM method with dense planar reconstruction using a monocular camera: TT-SLAM. The method exploits planar template-based trackers (TT) to compute camera poses and reconstructs a multi-planar scene representation. Multiple homographies are estimated simultaneously by clustering a set of template trackers supported by superpixelized regions. Compared to RANSAC-based multiple homographies method [1], data association and keyframe selection issues are handled by the continuous nature of template trackers. A non-linear optimization process is applied to all the homographies to improve the precision in pose estimation. Experiments show that the proposed method outperforms RANSAC-based multiple homographies method [1] as well as other dense method SLAM techniques such as LSD-SLAM or DPPTAM, and competes with keypoint-based techniques like ORB-SLAM while providing dense planar reconstructions of the environment.**

I. Introduction

Research on SLAM techniques (Simultaneous Localization And Mapping) has drawn a big amount of attention in the robotics community and led to implementations in various use cases: indoor and outdoor, in the urban and in the wild. Sparse SLAM methods either rely on direct alignment of pixel-level information or minimization of re-projection errors on extracted keypoints and similar low-level image features [2], [3], [4].

However, more high-level geometric features such as lines and planes can be exploited and integrated into visual SLAM systems as they provide a more semantic abstraction and are more robust over point-based image features.

Planes, for example, are ubiquitous geometric feature in man-made environments and objects, and enjoy worthwhile characteristics in visual tracking and SLAM tasks. Planar models only require a small set of parameters but can reconstruct complex scenes in a dense fashion. Planar models are also easy to estimate and track using homographies that express relations between image and world spaces. Henceforth, many tracking algorithms are based on single homography transforms: SLAM [5], object visual tracking [6] or robotic visual servoing [7].

While the single homography constraint can be easily exploited in tracking tasks on scenes with a dominant plane, this assumption severely limits applications to more general environments.

A number of contributions have therefore explored the use of multiple plane representations. Wang et al. [1] proposed a ransac-based relative camera pose estimation under multiple planar structures with the help of superpixels. Inspired by this work, this paper proposes a multiple planar SLAM framework using template-based trackers and superpixels to estimate camera trajectories and reconstruct a dense partial mapping from monocular image sequences (see Fig.1).

Our contributions are: (1) a novel method of initializing template trackers with the help of superpixels, (2) a mean shift clustering system to handle planar segmentation and pose estimation, and (3) a non-linear optimization refiner for improving precision and robustness by merging template tracker estimations.

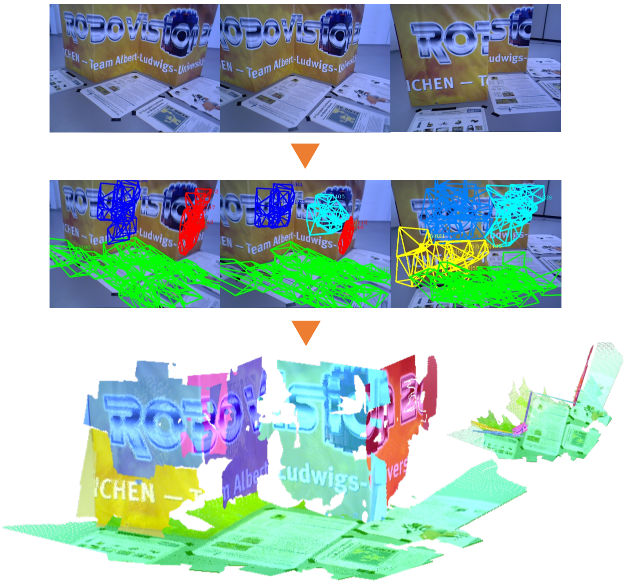


Fig. 1: We propose a visual SLAM method which tracks and clusters template-based trackers, estimates camera poses, and maps three dimensional multi-planar environments on color image sequences acquired by a monocular camera. Each color represents a different plane.

II. Related Work

A range of related work can be found under the topic of estimating camera pose and mapping in planar worlds. Some rely on the single homography assumption in which the workspace is usually one-dominant-planar scenes [8]. Pirchheim and Reitmayr [5] designed and developed a mobile augmented reality SLAM system for single planar environments. Combining the process with IMUs (Inertial Measurement Unit) also helps in improving the precision and eliminating ambiguities during homography decomposition [9], [10].

Another class of approaches addresses worlds through the Manhattan assumption: all three dimensional planes in

Author are with Univ Rennes, Inria, CNRS, Irisa, France.
Email:{xi.wang}@inria.fr
{marc.christie, eric.marchand}@irisa.fr

the environment are perpendicular to each other. Such an assumption is well suited in standard indoor and urban scenarios and simplifies the model, improving performance and precision in specific use-cases [10], [11], [12].

A number of approaches rely on planar scene SLAM and visual tracking systems by exploiting the depth information of RGB-D cameras. Kaess [13] proposes a quaternion formulation for 3D planes to improve convergence speed during optimization. Hsiao et al. [14] extended the previous work to a real-time keyframe-based RGBD planar SLAM: It does keyframe-based local odometry with help of geometric and photometric information for fast pose estimating. Then all the keyframes data are handled by a factor graph map using incremental smoothing and mapping technique (iSAM).

Approaches driven by deep learning neural networks also gained popularity and showed improved performances in many computer vision tasks. Pop-Up SLAM [12] demonstrates good performance for planar scene especially when the environment is texture-less. Yang and Scherer [15] proposed to add 3D object detection by bounding boxes as another constraint for Manhattan structured environments.

The use of superpixels in SLAM techniques has raised interest in the community. A superpixel is a group of pixels sharing spatial and chromatic similarities, usually generated by clustering or segmentation methods: classic works include SLIC [16], SEEDS [17] and graph-segmentation superpixel [18]. In the computer vision and robotic visionary domain, the technique is exploited as it provides a rough planar estimation.

More specifically, Concha and Civera propose to integrate superpixel techniques in sparse [19] and dense [20] SLAM systems to enhance mapping results. The idea consists of a Monte Carlo ranking to find the correspondence and initial 3D pose of superpixel-represented planes. The paper proposes an optimization framework to refine the plane poses with already known camera pose estimated separately from a PTAM system. Later, in DPPTAM [20], superpixels are used in a semi-dense tracking system. Similar to [19], plane estimation is handled in a decoupled fashion to camera pose which is found by semi-dense SLAM system. Ransac and SVD on three dimensional points is used for estimating the plane equation. A dense mapping optimization technique is therefore designed with superpixels information too.

Recently, [1] proposed a coupled estimation of multiple planes and camera pose based on multiple homographies. A dedicated RANSAC is applied on keypoints and ambiguities in plane estimation are eliminated via multiple homographies in order to simultaneously achieve sparse tracking and dense mapping.

Template-based trackers is a well-known technique in robotics to track and estimate planar image patches by registering different primitive geometric models w.r.t various metrics: *e.g.,* sum of square difference (SSD), zero-mean normalized cross-correlation (ZNCC), and mutual information (MI). Planar trackers usually estimate a homography transform between a template patch and query image via optimization method. Many applications are derived from template-based trackers including augmented reality [21], robot control [22], etc. Compared with RANSAC methods (*e.g.,* [1]), using template trackers to continuously extract homographies has the following advantages: 1) it solves well the data association problem when multiple planes are present in the scene; 2) it provides continuous observation of the tracking results, therefore the system has more flexibility to deal with the keyframe selection problem; 3) RANSAC method tends to require higher computational cost when dealing with multiple planes, as template trackers are much lighter and deterministic in terms of results.

Combining the advantage of template tracker and the work of multiple homographies pose estimation [1], we present a novel method of multiple planar vSLAM. It supports: 1) a novel method of tracking camera pose and mapping multiple planar environments simultaneously in a dense fashion; 2) a method of generating, clustering and utilizing template trackers with support of superpixel images for vSLAM applications; 3) a mean of applying homography-based non-linear optimization on template trackers as a refiner for achieving better pose estimation and mapping quality.

III. Overview

We propose TT-SLAM as a novel visual SLAM technique which relies on template trackers (TT) for planar environments. It comprehends the following modules (see overview in Fig. 2): (a) generation and tracking of template trackers: we add template trackers on regions of superpixelizd images and track them in the sequence of images; (b) clustering of decomposed planes: we rely on a mean shift clustering algorithm to group similar decomposed planes from homographies to extract a multi-planar structure; (c) non-linear refiner: we apply a non-linear optimization framework on template trackers to refine camera pose and multiple planes simultaneously on both the single incoming image and whole the image sequence (Bundle Adjustment-like). All the modules are presented in details in the following sections.

IV. Multiple Template Trackers

The main idea of our work is to rely on multiple template trackers to both estimate a camera pose and a dense planar mapping of a 3D scene.

Planar template tracker is a technique which tracks a planar image region on a sequence of frames. The technique outputs a homography transform $\mathbf{H}$ from a reference region in the first image to the current one. In a planar scene, the homography transform ${}^2\mathbf{H}_1 \in \mathbb{SL}(3)$ is used to describe the transformation of a three-dimensional plane from one image I_1 to another I_2. When the camera is intrinsically calibrated, *i.e.* the intrinsic matrix $\mathbf{K}$ is known, all pixels from I_1 and I_2 can be presented as normalized three dimensional coordinates denoted as: $\mathbf{p_1}$ and $\mathbf{p_2} \in \mathbb{R}^3$. The homography matrix is therefore a constraint between those points within the planar region:

$$\mathbf{p_2} = {}^2\mathbf{H}_1\mathbf{p_1}$$

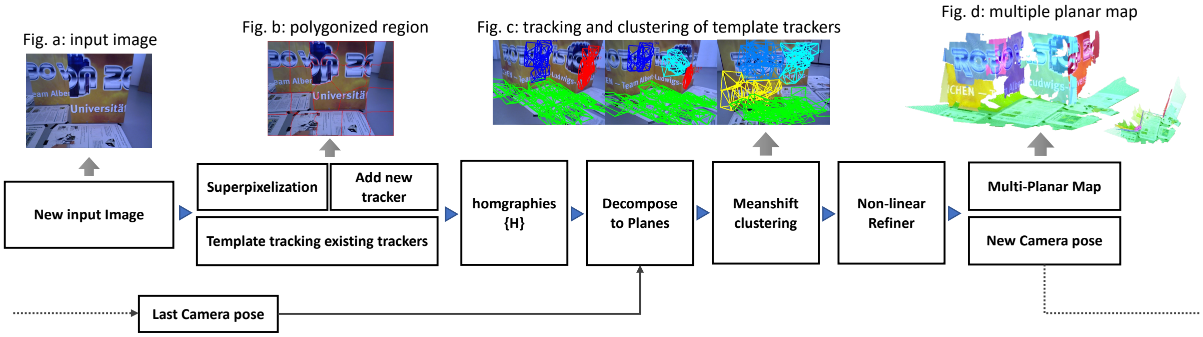


Fig. 2: Pipeline of our system which processes an input image sequence (subfig.a) to perform superpixelization (subfig.b). In subfig.c, tracking and clustering template trackers is performed (different colors represent different found planes in 3D, see subfig.d). Finally, passing through the refiner module, our method is able to recover camera trajectories together with a dense planar environment which conserves well plane perpendicularity without applying any Manhattan assumption.

This transform is actually composed of a rotation matrix ${}^{2}\mathbf{R}_1 \in \mathbb{SO}(3)$, a translation vector ${}^{2}\mathbf{t}_1 \in \mathbb{R}^3$ and a normal vector in the first frame I_1: $\mathbf{n}_1 = (a, b, c)^\top \in \mathbb{R}^3$ (Eq. 1). The three dimensional plane associated is then formulated as $\mathbf{p}^\top \mathbf{n}_1 = d$, where $\mathbf{p} \in \mathbb{R}^3$ are three dimensional points on the plane and d is the perpendicular distance to the origin:

$$ {}^{2}\mathbf{H}_1 = {}^{2}\mathbf{R}_1 + \frac{{}^{2}\mathbf{t}_1}{d} \mathbf{n}_1{}^\top \tag{1} $$

Different methods have been proposed to compute a homography matrix between images, some rely on keypoints [23] and others exploit pixel-level information [24]. For most template tracking problems, it is regarded as a differential image alignment problem at a pixel level.

The objective of differential image alignment is to estimate a displacement ρ of an image template I^* in multiple frames. It can be treated as a frame-to-frame tracking process, where the I^* is usually a Region-of-Interest (RoI) extracted from the reference frame. One then requires a similarity measure f to represent the distance between the reference image and the warped image. With the above definitions, one can describe the differential image alignment problem under an optimization problem:

$$ \widehat{\rho}_t = \arg\max_{\rho} f\left(I^*, w\left(I, \rho\right)\right) \tag{2} $$

where we aim at finding the displacement $\widehat{\rho}_t$ which maximizes the similarity under a given measure f. For the purpose of clarity, the warping function w is an abused notation to define a general transformation of the image I parameterized by ρ. In the context of planar homography estimation, we search on $\rho \in \mathfrak{sl}(3)$ which has 8 parameters. In order to accelerate the searching process, the inverse compositional formulation technique is proposed by precomputing derivatives of the reference image (see more details in [25], [6]).

Unlike common applications of template-based trackers where the regions-of-interest are usually known a priori or selected by user interaction, our system needs to automatically determine the regions-of-interest by computing adequate regions in terms of area and location consistent with a rough planar assumption. To address this problem, we rely on superpixel image decomposition. A superpixel is defined as a group of connected pixels sharing strong chromatic consistency (*e.g.,* SLIC [16]). We make an assumption here that each superpixel can be regarded as a potential planar region suitable for template-based trackers.

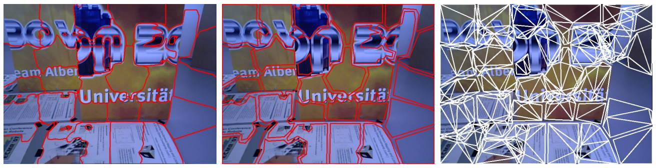


Fig. 3: An example of template tracker generation process. The left image shows the cluster contours of a superpixelized image. Polygonized regions and corresponding template trackers with triangulated RoIs are displayed in the middle and right images respectively.

During the initialization procedure, each superpixel is assigned as a RoI for a template-based tracker in order to track the regions in the following frames. Since superpixel borders are often non-planar and perturbate tracking quality, we propose to simplify the contour of superpixels by applying Teh-Chin chain approximation [26] and Ramer–Douglas–Peucker algorithm [27] on eroded superpixel contour. The regions are then represented as a Delaunay triangulation and considered as a tracking RoI (see Fig.3). Though the superpixels only provide a rough a priori on region planarity, trackers which are assigned with non-planar or multi-planar regions will quickly lead to divergence during the tracking optimization process and can be removed.

By contrast with our previous work [1] where all homographies are estimated from one given keyframe (*i.e.* the same reference image), new template trackers can be considered and added at any time. This reduces the risk of wrong keyframe selection, a issue identified in [1]. A policy is therefore devised to decide when to add new template trackers by selecting superpixels which fail to superpose with an already existing template tracker, by simply measuring their ratio of region overlapping on the image surface. For every new incoming frame, we therefore compare the newly computed superpixels with current valid trackers and add new trackers on those not covered. Our ratio is defined as follows for each superpixel:

$$r = \frac{S_{tt} \cap S_{sp}}{S_{sp}} \tag{3}$$

S_{tt} and S_{sp} are the regions of template tracker and superpixels respectively.

V. Clustering and Decomposition

Once we obtain a set of homographies from different template trackers $\{\mathbf{H}\}$, the next step consists in clustering homographies to obtain a simplified and better multi-planar representation. In our previous work [1], this was achieved by a Winner-Takes-All RANSAC on detected keypoints to identity multiple planes. Here we rely on a mean shift clustering technique to decide if some trackers belong to the same plane.

Clustering is a task of grouping similar data together and classifying according to specific metrics: classic works including K-means [28], mean shift [29], etc. Clustering is popular in computer vision and visionary robotics applications as it's able to reveal patterns from data aspect: *e.g.,* [30] use mean shift technique for estimating undrifted rotation from vanishing points in indoor scenarios to decouple the rotation and translation in SLAM.

In our work, we expect a clustering system to separate different trackers and group similar ones as they are tracking the same three dimensional plane. As we do not know in advance the number of planes in the scene, it makes mean shift clustering an appropriate method to deal with the case as it doesn't require an initial seed number, unlike other clustering methods. Ideally, if all the trackers are initialized at the same reference frame, we may directly apply mean shift on the space of homography $\mathbf{H} \in \mathbb{SL}(3)$. However, because of the aforementioned trackers adding policy, classification cannot be performed directly on the homography space since we are dealing with trackers initialized from different reference frames. Instead, since pose estimation is a sequential tracking problem, we propose to perform the classification on the decomposed planes represented in world coordinates (see Eq. 1), and clustering them in the space of plane parameters $\mathbf{\Pi} = \{\mathbf{n}, d\}$ where $\mathbf{n}$ is the normal vector of the plane and d is the perpendicular distance to the origin.

A classical issue, however, is the ambiguity in homography decomposition. Inevitably, decomposing a single homography yields two sets of results of $\mathbf{R}, \mathbf{t}, \mathbf{n}$ which both are geometrically valid. Without extra information, at least two ambiguities exist even after applying positive depth condition, unless one element among $\mathbf{R}, \mathbf{t}, \mathbf{n}$ is known in a priori, *e.g.,* by IMU information or known surface normal. For multiple planar homographies, we addressed the problem [1] by proposing voting on the common direction of the translational vector. We adopt the same method in this work for not only eliminating ambiguities but also filtering low quality template trackers by measuring their translational vector to the voted common direction: if none of translational vectors is close enough to the common direction among ambiguity sets, we consider the template tracker itself may be wrongly initialized or assigned with non-planar regions.

After decomposition, we obtain a set of planes represented in world coordinates, denoted as $\{\mathbf{\Pi}\}$. Instead of clustering naively on the space of planes $\mathbf{\Pi} = \{\mathbf{n}, d\}$ where the euclidean distance not defined properly, a hierarchical mean shift scheme is applied by considering first the normal vectors $\{\mathbf{n}\}$, then second the d parameter and the on-image barycenter position $\{d, p_c\}$ of each template trackers for grouping planes locally. We utilize the euclidean metric on both hierarchies of clustering and find the results are good enough though the space of plane normal has its own geodesic metric on the sphere group (see Fig. 4 for clustering results and correspondent depth image).

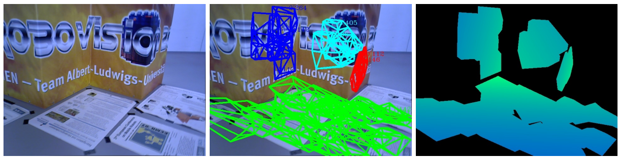


Fig. 4: Clustered and matched template trackers in middle subfig (same color represents same clustered 3D plane), and correspondent depth is generated on trackers region (right).

VI. Non-linear Multi-Plane Refiner and BA

A. Non-linear Refiner on Current Image

Given the clustering performed on the image planes, we then design a refining process to better exploit the information from multiple trackers and improve the estimation of both the camera pose $\mathbf{q} \in \mathfrak{se}(3) \in \mathbb{R}^6$ (the minimal representation of transformation $\{\mathbf{R}, \mathbf{t}\}$) and planar equations $\mathbf{\Pi}$ simultaneously. In traditional SLAM systems, this process is usually handled by a non-linear optimization framework, which minimizes the re-projection error on image space of extracted landmarks such as keypoints (Bundle Adjustment).

To handle homography transformations, a similar process can be applied via a non-linear least square Gauss-Newton optimization process which minimizes the re-projection error E between pixels $(\mathbf{p}_2^n - {}^2\mathbf{H}_1\mathbf{p}_1^n)^2$, $n = 1, .., N_p$ as number of pixels, w.r.t. camera pose $\mathbf{q}$ and the plane parameter $\mathbf{\Pi}_1 = \{\mathbf{n}_1, d\}$. This is expressed as:

$$\{\widehat{\mathbf{q}}, \widehat{\mathbf{\Pi}}_1\} = \operatorname*{argmin}_{\mathbf{q},\mathbf{\Pi}_1} E(\mathbf{q}) = \operatorname*{argmin}_{\mathbf{q},\mathbf{\Pi}_1} \sum_{n}^{N_p} \left(\mathbf{p}_2^n - {}^2\mathbf{H}_1\mathbf{p}_1^n\right)^2 \tag{4}$$

To compute the re-projection error, we use vertices from the Delaunay triangulation process of each template tracker.

Similarly to [1], sharing multiple homographies in a static environment can be interpreted as the relation of a set of homographies estimated by trackers $\{\mathbf{H}^i\}$ and a shared transformation in world coordinate frame ${}^w\mathbf{T}_o \in \mathbb{SE}(3)$ (o represents origin of the frame) represented by local transforms ${}^w\mathbf{T}_{r_i}$ (from the reference frame r_i of template tracker i to its current position) for all trackers, where $i = 1, .., N_{tt}$ as the number of trackers:

$$\begin{aligned} {}^{w}\mathbf{T}_{r} &= {}^{w}\mathbf{T}_{o}\,{}^{r}\mathbf{T}_{o}^{-1}, {}^{w}\mathbf{T}_{r} = \{{}^{w}\mathbf{R}_{r}, {}^{w}\mathbf{t}_{r}\} \\ \mathbf{H}^{i} &= {}^{w}\mathbf{R}_{r_i} + \frac{{}^{w}\mathbf{t}_{r_i}}{d_i}\mathbf{n}_{r_i}{}^{\top} \end{aligned} \tag{5}$$

We can therefore propose a refiner for estimating camera pose and planar equations simultaneously from multiple trackers homographies. Note that we already know the correspondence mapping from $\{\mathbf{\Pi}^i\}$ to clustered and grouped planes $\{\mathbf{\Pi}^c\}$ by mean shift and data association. Rather than considering each plane separately for each tracker, during the optimization process we group planes together in $\{\mathbf{\Pi}^i\}$ following the mean shift clustering.

$$\{\widehat{\mathbf{q_w}}, \{\widehat{\mathbf{\Pi}^c_w}\}\} = \arg\min_{\mathbf{q_w}, \{\mathbf{\Pi}^c_w\}} \sum_{i}^{N_{tt}} \sum_{n}^{N_{p_i}} \left(\mathbf{p}^n_{w_i} - {}^{w}\mathbf{H}_{r_i}\mathbf{p}^n_{r_i}\right)^2 \tag{6}$$

with $\mathbf{p}^n_w$ and $\mathbf{p}^n_{r_i}$ are the vertices of tracking regions from the current frame and the corresponded reference frame of the template tracker i respectively, their sum quantity is N_{p_i} and number of trackers N_{tt}. Remember that the camera pose $\widehat{\mathbf{q_w}}$ and planar equation $\widehat{\mathbf{\Pi}^c_w}$ are actually in the world coordinates, thus a transform of Eq. 5 from global coordinate to local coordinate is mandatory as the homography is defined only between the reference frame and current one. For simplicity, we denote ${}^{w}\mathbf{H}_{r_i}$ by an abuse of notation and hide the transform in Eq. 6.

A warm start for the optimization can be given directly from the last camera pose and also by searching for the previous global planar results for each template tracker. With the help of template trackers, plane data association is no longer a problem as we already know which template tracker generates each plane. Simple searching and comparing of trackers is performed.

B. Bundle Adjustment-like Refiner

The plane map refiner consists of an optimization framework that refines all keyframes' poses and their common planes found by the plane matching process. Each keyframe contains multiple planes and their vertices. Once the *joint plane* information is gained over different keyframes, like global BA for point-based SLAMs, this procedure eliminates the drifting problem, alleviates scale ambiguity, and refines camera trajectory w.r.t whole sequence.

By analogy, we propose a Bundle Adjustment (BA) system for refining every frame's pose and the joint plane information by mutually minimizing their re-projection error:

$$\arg\min_{\mathbf{q}_t, \{\mathbf{\Pi}^c_t\}} \sum_{t}^{N_t} \sum_{i}^{N_{tt}} \sum_{n}^{N_{p_i}} \left(\mathbf{p}^n_t - {}^{t}\mathbf{H}_{r_i}\mathbf{p}^n_{r_i}\right)^2 \tag{7}$$

where t and i are the index of frame and tracker number, N_t and N_{tt} represent the total frame and template trackers number respectively.

C. Planar Map

1) Plane merging and keyframes: We also deploy a plane merging scheme to fuse close planes given a metric on plane normal vector $\mathbf{n}$ and orthogonal distance d. Ideally, we don't rely on well-selected keyframes such as [1] since keypoint homographies are prone to errors with insufficient translations. In contrast, template trackers allow us to track planes along the sequence, and wait until the estimation is stable before generating keyframes.

2) Template rejection: Unlike RANSAC-based methods, template trackers maximize the similarity of all pixels in the region. This makes the outlier rejection critical for a SLAM system: any ill-tracked template tracker is capable of adding noise in the overall camera and plane estimation. Besides applying a robust loss function such as Huber loss [31], we also propose a template rejection procedure for preventing ill-tracked templates. Three main points are chosen here to filter out bad trackers:

- The lack of convergence or high tracking cost led by tracker's optimization, which usually occurs when initialising on texture-less or non-planar regions.
- The voting distance during the ambiguity elimination process: if none of the computed solution is close to the common voted translational direction.
- Unstable templates: we monitor each template in terms of their plane equations and prune trackers which fail to generate stable plane in measure of their parameters.

VII. Experiments and Discussions

We test our proposed method in two different scenarios: indoor and outdoor environments.

For the indoor environments, we test three levels of difficulty and complexity from simplest to most complex: single plane scenario, multiple planes scenario, and a complex multiple planar real room.

Single (`fr_nstr_loop`) and multiple (`fr_str_far`) plane scenarios are tested with the TUM RGB-D dataset [32] which is also utilised by many planar or dense SLAM methods [20], [33], [1]. The scene is composed of rich textured planar structures and relative homogeneous color distribution area. It raises challenges for superpixel decomposition and template trackers as sometimes RoIs might spawn at the middle line of two different planes and mislead the following estimations. However, the proposed system handles well the single and multiple planar scenes, as displayed in Table. I for the comparison of Absolute Pose Error (APE) with ORB-SLAM [2], LSD-SLAM [3], Multi-Level Mapping [33], DPPTAM [20] and our previous work: a ransac-based multi-planar method [1]. We demonstrate in subtables `fr_nstr_loop` and `fr_str_far` for single and multiple plane scenarios. Our method outperforms all dense and RANSAC methods and reaches a good level of precision against a state-of-the-art monocular sparse keypoint-based SLAM [2] which only provides a sparse point cloud mapping. One explanation about the precision drop comparing with [2] in the single planar scene (`fr_nstr_loop`) is that without using keypoints and specially designed relocalization module, the system tends to accumulate errors along the tracking and is negatively influenced by motion blur taken during the image acquisition. It also explains other dense methods' ill-performance. Comparison of APE along the sequence `fr_str_far` is shown in Fig. 5, our method yields lower level error along the whole trajectory. Generated planar maps are viewed in Fig. 6. Dense planar maps are created by reprojecting tracker regions according to computed planar equations at each frame. It's observed that the map conserves well the perpendicularity without applying any Manhattan assumptions.

The second experiment of the indoor scene is a drone dataset EuRoc [34]: a drone recorded grey-level dataset in a test room of a flight sequence. We take a segment ($\sim$ 400 frames) of the scene `v1_01_easy` as the environment is not specifically designed for planar SLAM and some texture-less sections and regions fail template trackers. As shown by the results in the third section of Table. I, we also achieve a good level of precision compared to all dense methods and even better than ORB-SLAM [2] on the median error metric.

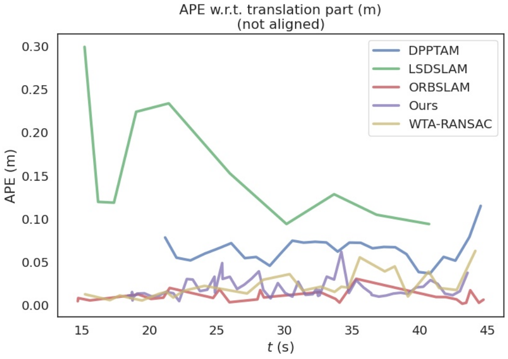
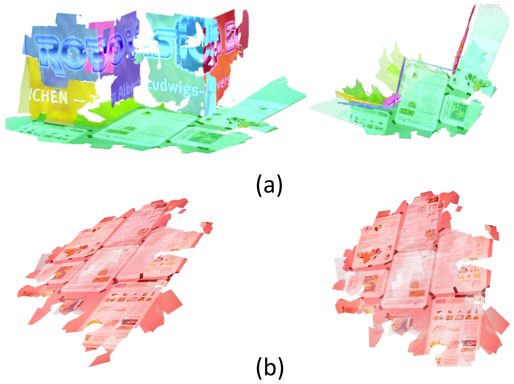


Fig. 5: Absolute Pose Error (APE) metric for the sequence `fr_str_far` of Dataset [32] shows that our dense mapping method outperforms all dense and semi-dense methods and reaches a decent precision level compared to ORB-SLAM which only provides a sparse point cloud map.



(a)

(b)

Fig. 6: 3D multiple (subfig a) and single plane map (subfig b) of the dataset TUM [32] generated by our method. Our proposed method is able to estimate camera trajectory and planar map representation simultaneously.

Data	*Methods*	*Mean (m)*	*Median (m)*	*RMSE (m)*
f3_str_far	[2]	**0.010**	**0.009**	**0.012**
	[3]	0.157	0.124	0.170
	[33]	-	-	0.17
	[20]	0.063	0.063	0.065
	[1]	0.023	0.017	0.027
	TT-SLAM	**0.018**	**0.014**	**0.021**
f3_nstr_loop	[2]	**0.012**	**0.011**	**0.013**
	[3]	0.733	0.649	0.867
	[33]	-	-	0.22
	[20]	0.180*	0.159*	0.197*
	TT-SLAM	**0.110**	**0.098**	**0.120**
v1_01_easy	[2]	**0.091**	0.085	**0.094**
	[3]	1.205	1.107	1.406
	[20]	x	x	x
	TT-SLAM	**0.099**	**0.080**	**0.112**

TABLE I: ATE Evaluation: The proposed method (TT-SLAM) outperforms DPPTAM [20], LSD-SLAM [3] and Multi-Level Mapping [33], ransac-based pose estimation from multi-homographies [1]. Despite behind ORB-SLAM [2] performance (a keypoint sparse SLAM without planar assumption), our approach provides a dense map representation. (***** means lost a portion during tracking, **-** means no reported data, **x** means initialization failure)

For the outdoor experiment, we test our system on a sequence from a hand-held gray-level dataset [35], in a scene of a corridor-like environment. Fig. 7 displays that our system retrieves the corridor's perpendicular planar structure as well as a camera trajectory from the input sequence.

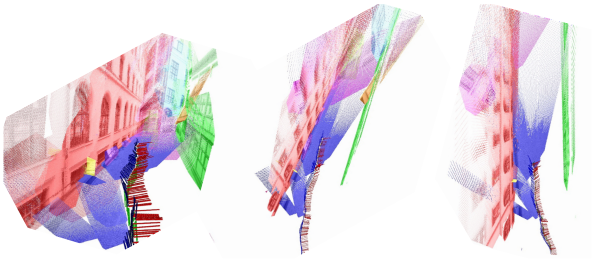


Fig. 7: Reconstructing on the dataset [35], coordinates represent the camera poses. The multi-planar environment is well conserved without applying Manhattan assumptions.

VIII. Conclusion

We proposed a novel way of estimating camera pose and generating dense planar mapping via template trackers. Trackers are created from superpixelized image regions. A mean shift clustering technique is applied to merge similar planes. Finally, a optimization-based refiner is designed to achieve better performance.

Our perspective comprehends three directions: first relying on heterogeneous information such as keypoints, and depth information to improve robustness and tracking quality. The second direction consists in using a deep-learning segmentation and planar region detection rather than superpixels. Third, we aim at exploiting planar maps for relocalization tasks and data association in general SLAM systems.

References

[1] X. Wang, M. Christie, and E. Marchand, "Relative pose estimation and planar reconstruction via superpixel-driven multiple homographies," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'20*, 2020.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Trans. on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct 2015.

[3] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[4] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Mar. 2018.

[5] C. Pirchheim and G. Reitmayr, "Homography-based planar mapping and tracking for mobile phones," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 27–36.

[6] A. Dame and E. Marchand, "Second-order optimization of mutual information for real-time image registration," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4190–4203, 2012.

[7] É. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 53–70, 2005.

[8] S. Benhimane and E. Malis, "Homography-based 2d visual tracking and servoing," *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 661–676, 2007.

[9] B. Guan, P. Vasseur, C. Demonceaux, and F. Fraundorfer, "Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2320–2327.

[10] O. Saurer, F. Fraundorfer, and M. Pollefeys, "Homography based visual odometry with known vertical direction and weak manhattan world assumption," in *ViCoMoR 2012: 2nd Workshop on Visual Control of Mobile Robots (ViCoMoR): Half Day Workshop: October 11th, 2012, Vilamoura, Algarve, Portugal, in conjunction with the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, 2012, pp. 25–30.

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

[12] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up slam: Semantic monocular plane slam for low-texture environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1222–1229.

[13] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4605–4611.

[14] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5110–5117.

[15] S. Yang and S. Scherer, "Monocular object and plane slam in structured environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3145–3152, 2019.

[16] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[17] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "Seeds: Superpixels extracted via energy-driven sampling," in *European conference on computer vision*. Springer, 2012, pp. 13–26.

[18] 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.

[19] A. Concha and J. Civera, "Using superpixels in monocular slam," in *2014 IEEE international conference on robotics and automation (ICRA)*, 2014, pp. 365–372.

[20] ——, "Dense Piecewise Planar Tracking and Mapping from a Monocular Sequence," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[21] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: a hands-on survey," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 12, pp. 2633–2651, 2015.

[22] F. Spindler, "Vision-based robot control with visp," in *ICRA 2018-Tutorial on Vision-based Robot Control*, 2018.

[23] Y. Kanazawa and H. Kawakami, "Detection of planar regions with uncalibrated stereo using distribution of feature points," in *In British Machine Vision Conference*, 2004, pp. 247–256.

[24] A. Agarwal, C. Jawahar, and P. Narayanan, "A survey of planar homography estimation techniques."

[25] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International journal of computer vision*, vol. 56, no. 3, pp. 221–255, 2004.

[26] C.-H. Teh and R. T. Chin, "On the detection of dominant points on digital curves," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, no. 8, pp. 859–872, 1989.

[27] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer graphics and image processing*, vol. 1, no. 3, pp. 244–256, 1972.

[28] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," 1967.

[29] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on information theory*, vol. 21, no. 1, pp. 32–40, 1975.

[30] P. Kim, B. Coltin, and H. Jin Kim, "Linear rgb-d slam for planar environments," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 333–348.

[31] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*. Springer, 1992, pp. 492–518.

[32] 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.

[33] W. N. Greene, K. Ok, P. Lommel, and N. Roy, "Multi-level mapping: Real-time dense monocular slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 833–840.

[34] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[35] J. Engel, V. Usenko, and D. Cremers, "A photometrically calibrated benchmark for monocular visual odometry," in *arXiv:1607.02555*, July 2016.