CPA-SLAM: Consistent Plane-Model Alignment for Direct RGB-D SLAM

Lingni Ma, Christian Kerl, Jörg Stückler and Daniel Cremers

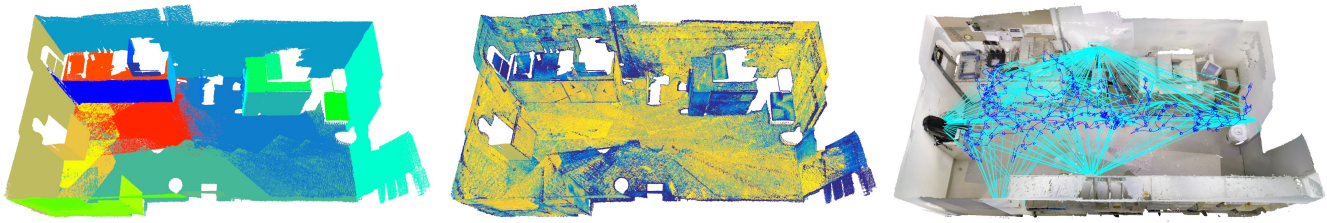


Fig. 1: Demo of our CPA-SLAM algorithm. Left: plane segmentation and association of all keyframes (same color depicts associated planes; association requires a minimal overlap). The found planes include segmentation errors, e.g., objects on tables and close to walls. Middle: average soft association with the plane model from tracking with our EM framework. The probability of a pixel being associated with a plane drops from yellow (1) to blue (0). Right: final map and plane-keyframe constraints used for graph optimization.

***Abstract*— Planes are predominant features of man-made environments which have been exploited in many mapping approaches. In this paper, we propose a real-time capable RGB-D SLAM system that consistently integrates frame-to-keyframe and frame-to-plane alignment. Our method models the environment with a global plane model and – besides direct image alignment – it uses the planes for tracking and global graph optimization. This way, our method makes use of the dense image information available in keyframes for accurate short-term tracking. At the same time it uses a global model to reduce drift. Both components are integrated consistently in an expectation-maximization framework. In experiments, we demonstrate the benefits our approach and its state-of-the-art accuracy on challenging benchmarks.**

I. Introduction

Man-made environments are composed of many objects with simple geometric properties that can be used for reference in visual simultaneous localization and mapping (SLAM) systems. In this process, these objects represent higher-level semantic information in the map, which could be used, e.g., for robots to reason about the world, to communicate information, or to display information to users, for instance, using virtual reality devices. The object states can be concurrently estimated with the motion of the camera. Integrated consistently, camera motion and object state estimation can benefit from each other.

In this paper, we propose a novel formulation for including planes as a global model into a direct, keyframe-based SLAM approach with an expectation-maximization (EM) framework. For tracking and map optimization, we propose an image alignment method that tracks the camera motion towards a reference keyframe and, at the same time, aligns the image with the planes in a global model. The model planes are concurrently extracted from the keyframes and estimated in global coordinates using graph optimization. By including both local frame-to-keyframe and global frame-to-model constraints in direct image alignment, we significantly reduce drift that is a typical problem of pure keyframe-based SLAM methods. Graph constraints between keyframes without overlapping views can be established if they observe the same model plane. An additional benefit of our method is that it provides a compact planar scene representation.

We name our algorithm CPA-SLAM and show a demo in Fig. 1. On the left we visualize for all keyframes their segmentation into planes and the association of these planes based on a minimal overlap citerion. It can be seen that there are some false segmentations, e.g., small objects on and close to surfaces such as walls and tables are included into the plane segments. Shown in the middle, we illustrate that our EM tracking automatically determines not to align these pixels with the plane model. The final model on the right demonstrates the accurate mapping achieved with our method. It also shows the constraints between non-overlapping keyframes that are otherwise difficult to establish without the plane model. The major contributions of this work are summarized as follows:

- We develop an RGB-D SLAM approach that consistently tracks camera motion through direct image alignment towards a keyframe (as in [1]) and a global plane model in an EM framework.
- We further use this alignment method to obtain spatial constraints between keyframes and the global plane model. These are jointly optimized with alignment constraints between keyframes for global consistency.
- We demonstrate a real-time capable SLAM system.

We compare our method to state-of-the-art approaches on challenging benchmark datasets and demonstrate improvements in trajectory estimation accuracy in relation to these approaches.

This work is partially funded by EU project SPENCER (ICT-2011-600877), ERC grant Convex Vision (GA no. 240168) and ERC Proof of Concept grant CopyMe3D (GA no. 632200).

L. Ma, C. Kerl, J. Stückler and D. Cremers are with the Computer Vision Group, Computer Science Institute IX, Technical University of Munich, Germany {lingni,kerl,stueckle,cremers}@in.tum.de

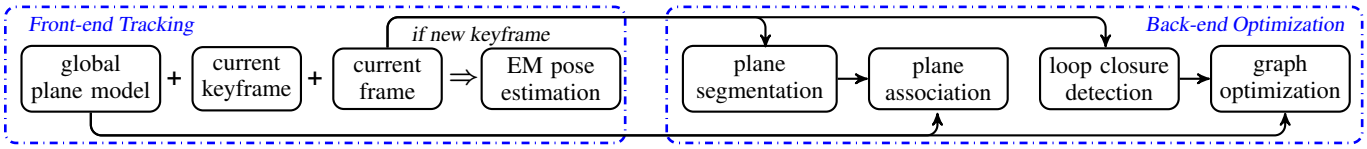


Fig. 2: Schematic pipeline of our CPA-SLAM system.

II. Related Work

Several approaches to visual SLAM have exploit planes as typical features in man-made environments. Gee et al. [2] use planes to reduce the amount of interest points in a monocular Kalman-filter based SLAM approach. In [3], planes are also included in the state-space of an EKF method to monocular SLAM. This method focusses on the incremental optimization of the planes in a global frame for augmented reality applications. In [4], orthogonality of planes in indoor environments is exploited to improve the consistency of plane models and SLAM. Martinez-Carranza et al. [5] propose a unified parameterization for points and planes in a monocular EKF-SLAM system.

In RGB-D SLAM, dense depth enables the detection of texureless planes. Dou et al. [6] combine planes and interest points in frame-to-frame matching and bundle adjustment. They detect planes using a Hough-voting scheme. The corresponding planes between frames are found through RANSAC which yields plane tracks throughout the RGB-D sequence. Global planes are instantiated from the tracks and used within bundle adjustment. Since there may exist several tracks of the same plane entity, the planes are merged according to distance criteria. Taguchi et al. [7] also use interest points and planes for SLAM with RGB-D sensors. They directly use various combinations of point and plane observations in a RANSAC framework to determine correspondence and camera pose between frames and a global map. Trevor et al. [8] use RANSAC to find the major planes in a scene from RGB-D and 2D laser measurements. SLAM is performed in an EKF framework, associating plane observations with global planes in the map. Salas-Moreno et al. [9] integrate incremental plane mapping into point-based fusion [10]. Besides isolated surfels, they also label surfels to belong to the same planar structure and enforce planarity constraints in their estimates.

Some works also have tackled the problem of including object detection into SLAM. Salas-Moreno et al. [11] detect objects from a model database and estimate their poses in individual frames. The poses of these objects in the global frame are then estimated through graph optimization. Another similar work is semantic bundle adjustment [12] in which the objects are detected and tracked, and included as 6-DoF landmarks in a bundle adjustment framework. In contrast to our approach, both methods do not handle a seamless transition between object and remaining image measurements already in the camera tracking part.

Our method applies direct alignment towards keyframe and plane model consistently in an EM framework to estimate camera motion. We also use both keyframes and planes to obtain spatial constraints for global graph optimization.

III. Direct SLAM with Model Planes

Our SLAM system has a mixed nature of frame-to-keyframe and frame-to-model online tracking. As a global model, we maintain a set of planes that originate from keyframe segmentation to present the large smooth regions in the scene. We optimize the global plane model with all observations from the keyframes.

Fig. 2 illustrates our SLAM system. At the front-end tracking, each RGB-D frame is aligned towards its nearest keyframe as well as the global plane model in an EM framework. An RGB-D frame becomes a new reference keyframe, if the certainty of motion estimation drops below a threshold. When a new keyframe is produced, an iteration of back-end optimization starts. In this process, we first segment the keyframe and associate the detected planes with the global model. Planes that are failed to be associated with existing planes in the model are included as new planes into the model. We further search for loop closures between keyframes and establish constraints between keyframe and the global model given the keyframe plane observations. The graph is then optimized to correct the keyframe poses and the plane model. The front-end tracking and back-end optimization can be processed in parallel.

In the following section, we will first explain the incremental construction of the plane model. We then detail our algorithm for camera tracking through direct image alignment towards a keyframe and the plane model. Last, we explain how we optimize the plane model and the camera poses through graph optimization.

A. Preliminaries

We denote the index of a keyframe by k and the current frame by i. An image is considered as a 2D continuous domain $\Omega \subset \mathbb{R}^2$ which can be segmented into disjoint regions Ω_j. We denote a 3D point by $\mathbf{v}$, its unit normal by $\mathbf{n}$, and a 2D pixel by $\mathbf{x}$. The projection of a 3D point onto the 2D image is $\mathbf{x} = \rho(\mathbf{v})$, and the back-projection is $\mathbf{v} = \rho^{-1}(\mathbf{x})$. To represent the rigid body motion in $\mathbb{SE}(3)$, we use the minimal parameterization with twist coordinate $\boldsymbol{\xi}$ of Lie algebra $\mathfrak{se}(3)$. The exponential map $g(\boldsymbol{\xi}) = \exp(\widehat{\boldsymbol{\xi}})$ converts the twist coordinate to a transformation matrix, and the log map g^{-1} operates vice versa. The subscript of $\boldsymbol{\xi}$ specifies direction: $\boldsymbol{\xi}_{ji}$ transforms from frame i to j, and $\boldsymbol{\xi}_i = \boldsymbol{\xi}_{wi}$ transforms from frame i to the world. The inverse motion is denoted by $\boldsymbol{\xi}_i^{-1} = \boldsymbol{\xi}_{iw}$. We further define function $t(\boldsymbol{\xi}, \mathbf{v}) = g(\boldsymbol{\xi})\mathbf{v}$ to transform 3D points, and $\omega(\boldsymbol{\xi}, \mathbf{x}) = \rho\left(t\left(\boldsymbol{\xi}, \rho^{-1}(\mathbf{x})\right)\right)$ to warp pixels between frames.

We parametrize planes using the Hessian form $\boldsymbol{\pi} = (\mathbf{n}^\mathsf{T}, d)^\mathsf{T}$, where $\mathbf{n}$ is the unit plane normal, and $-d$ is the distance from the plane to the origin. The distance of any point $\mathbf{v}$ to the plane is calculated by $r_\mathcal{D} = \mathbf{n}^\mathsf{T}\mathbf{v} + d$. A plane observed in frame k is transformed into world coordinates by $t(\boldsymbol{\xi}_k, \boldsymbol{\pi}) = g(\boldsymbol{\xi}_k)^{-\mathsf{T}}\boldsymbol{\pi}$.

B. Global Plane Model

We define the global plane model as a set of planes in the world coordinates $\{\boldsymbol{\pi}_m^g\}$. This model represents large flat surfaces in a scene. Each plane $\boldsymbol{\pi}_m^g$ in the model is associated with a list of independent local observations $\boldsymbol{\pi}_{mk}$ in the keyframes. The global plane model is estimated through graph optimization using all local observations, which we detail in Section III-F.

The global plane model is augmented incrementally. Whenever a new keyframe is produced, it is segmented into K regions, where Ω_0 is the non-planar region and $\Omega_j\,, j > 0$ is the jth plane. For efficient plane detection, we apply the agglomerative hierarchical clustering (AHC) algorithm proposed by Feng et al. [13]. For each plane segment, least squares plane fitting is used to estimate its parameters. We then associate the local observations with the global model. A correspondence is found if the angle between the plane normals is small ($< 15°$) and their distances to the origin are similar. We further confirm the association by warping the current plane segment into other keyframes and examine the overlaps. If a local plane observation fails to be associated, it is added to the global model as a new element.

C. Tracking towards Keyframe and Plane Model

We now describe motion estimation from the current frame i to the keyframe k by minimizing both photometric error $r_\mathcal{I}$ and geometric error $r_\mathcal{G}$. To simplify the notation, we drop the subscript of $\boldsymbol{\xi}$ in this section.

1) Formulation: The photometric residual is defined on the intensity image assuming photoconsistency as

$$r_\mathcal{I} = I_k\big(\omega(\boldsymbol{\xi}, \mathbf{x})\big) - I_i(\mathbf{x}). \tag{1}$$

The geometric residual is defined by

$$r_\mathcal{G} = \begin{cases} \mathbf{n}_k^\mathsf{T}\big(g(\boldsymbol{\xi}, \mathbf{v}_i) - \mathbf{v}_k\big) & \text{if } \omega(\boldsymbol{\xi}, \mathbf{x}_i) \in \Omega_0 \\ \mathbf{n}_{\pi j}^\mathsf{T} g(\boldsymbol{\xi}, \mathbf{v}_i) + d_j & \text{if } \omega(\boldsymbol{\xi}, \mathbf{x}_i) \in \Omega_j \end{cases}, \tag{2}$$

which depends on whether the current pixel $\mathbf{x}_i$ is warped to the non-planar region Ω_0 or the jth planar region Ω_j of the keyframe. In case of Ω_0, the geometric residual resembles a variant of ICP [14]. Otherwise, the residual is the distance to the corresponding global plane transformed into the keyframe, thus, $(\mathbf{n}_{\pi j}^\mathsf{T}, d_j)^\mathsf{T} = t(\boldsymbol{\xi}_k^{-1}, \boldsymbol{\pi}_m^g)$.

Combining the photometric and geometric error into one variable $\mathbf{r} = (r_\mathcal{I}, r_\mathcal{G})^\mathsf{T}$, and with N correspondences between keyframe and current frame, we find the camera motion by minimizing the following non-linear weighted least squares,

$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \sum_n^N \sum_k^K \gamma_{nk} w_{nk} \mathbf{r}_n^\mathsf{T} \boldsymbol{\Sigma}_k^{-1} \mathbf{r}_n. \tag{3}$$

The weight w_{nk} is used to enhance the robustness against outliers and can be iteratively estimated. In our case, the weights are derived from a Student-t distribution as proposed in [1]. The variable $\boldsymbol{\gamma}_n \in \mathbb{R}^K$ is the labeling that indicates which region the residual belongs to. Accordingly, γ_{n0} refers to non-planar region Ω_0 and γ_{nj} refers planar region Ω_j. Instead of a hard labeling $\gamma_{nk} \in \{0, 1\}$, we use the soft labeling $\gamma_{nk} \in [0, 1]$ to increase robustness. This objective can be efficiently optimized with the Gauss-Newton method. In the following we will address how to concurrently determine $\boldsymbol{\gamma}, \boldsymbol{w}, \boldsymbol{\Sigma}$ in a probabilistic formulation.

2) A Probabilistic View on Motion Estimation: The optimization problem in eq. (3) cannot be solved directly, because the parameters $\boldsymbol{\gamma}, \boldsymbol{w}, \boldsymbol{\Sigma}$ also need to be estimated in addition to the motion $\boldsymbol{\xi}$. To solve for both, we motivate the same energy function from a probabilistic point of view and show that optimizing eq. (3) is equivalent to optimizing a mixture of bivariate t-distributions in an EM framework.

Suppose that $K - 1$ planes are visible in the keyframe. For each pixel observation with residual $\mathbf{r}_n$, there is a corresponding indicator $\mathbf{z}_n \in \mathbb{B}^K$ that tells which segment the geometric observation comes from. As an indicator, $\mathbf{z}_n$ satisfies $z_{nk} \in \{0, 1\}$ and $\sum_k^K z_{nk} = 1$. Now, assume the following probability

$$p(z_{nk} = 1) = \eta_k,\; p(\mathbf{r}_n | z_{nk} = 1) = p_t(\mathbf{r}_n; \mathbf{0}, \boldsymbol{\Sigma}_k, \nu_k), \tag{4}$$

which can also be written in a compact form,

$$p(\mathbf{z}_n) = \prod_k^K \eta_k^{z_{nk}},\; p(\mathbf{r}_n | \mathbf{z}_n) = \prod_k^K p_t(\mathbf{r}_n; \mathbf{0}, \boldsymbol{\Sigma}_k, \nu_k)^{z_{nk}}. \tag{5}$$

The marginal probability of $\mathbf{r}_n$ is therefore,

$$p(\mathbf{r}_n) = \sum_k^K \eta_k p_t(\mathbf{r}_n; \mathbf{0}, \boldsymbol{\Sigma}_k, \nu_k), \tag{6}$$

which is a mixed model of bivariate t-distributions. Seeking motion $\boldsymbol{\xi}$ by maximum-likelihood estimation, yields

$$\begin{aligned} \boldsymbol{\xi}^* &= \arg\max_{\boldsymbol{\xi}} \log p(\mathbf{r} | \boldsymbol{\xi}) \\ &= \arg\max_{\boldsymbol{\xi}} \sum_n^N \log\Big(\sum_k^K \eta_k p_t(\mathbf{r}_n; \mathbf{0}, \boldsymbol{\Sigma}_k, \nu_k)\Big). \end{aligned} \tag{7}$$

With the distribution of $\mathbf{r}$ being a mixed model, the logarithm acts outside the summation and the direct optimization no longer yields a weighted least squares form. Even worse, there is no closed-form solution for the hyper-parameter $\boldsymbol{\Sigma}_k$.

The indicator $\mathbf{z}_n$ is a latent variable which we cannot observe directly. Therefore, the observed data $\mathbf{r}$ are incomplete, while $\{\mathbf{r}, \mathbf{z}\}$ are complete. Using eq. (5), the complete log-likelihood is,

$$\begin{aligned} \log p(\mathbf{r}, \mathbf{z}) &= \log \prod_n^N \prod_k^K \big(\eta_k p_t(\mathbf{r}_n; \mathbf{0}, \boldsymbol{\Sigma}_k, \nu_k)\big)^{z_{nk}} \\ &= \sum_n^N \sum_k^K z_{nk} \log\big(\eta_k p_t(\mathbf{r}_n; \mathbf{0}, \boldsymbol{\Sigma}_k, \nu_k)\big). \end{aligned} \tag{8}$$

This shows that with the knowledge of the indicators $\mathbf{z}$ we can regain the simple form of the optimization problems to estimate motion and hyper-parameters.

3) Tracking as Expectation-Maximization: The EM algorithm provides a probabilistic formalism to estimate the parameters of posterior probability functions with latent variables as in eq. (7). It constructs a lower bound on the log-likelihood by optimizing the Kullback-Leibler divergence between a simpler approximation and the actual posterior probability (see [15] for details). In EM, one therefore optimizes the conditional expectation of the log joint probability, conditioned on the posterior probability of the latent variables. In our case, this is

$$\mathbb{E}_{p(\mathbf{z}|\mathbf{r})}[\log p(\mathbf{r},\mathbf{z})] = \sum_n^N \sum_k^K \gamma_{nk} \log\left(\eta_k p_t(\mathbf{r}_n; \mathbf{0}, \mathbf{\Sigma}_k, \nu_k)\right), \tag{9}$$

where we define the mixing coefficient

$$\gamma_{nk} = \mathbb{E}_{p(\mathbf{z}|\mathbf{r})}[z_{nk}] = p(z_{nk}|\mathbf{r}_n). \tag{10}$$

Further writing out the EM objective function yields,

$$\mathbb{E}_{p(\mathbf{z}|\mathbf{r})}[\log p(\mathbf{r},\mathbf{z})] = \sum_n^N \sum_k^K \frac{\gamma_{nk}\cdot(\nu_k+2)}{\nu_k + \mathbf{r}_n^\mathsf{T}\mathbf{\Sigma}_k^{-1}\mathbf{r}_n} \mathbf{r}_n^\mathsf{T}\mathbf{\Sigma}_k^{-1}\mathbf{r}_n. \tag{11}$$

Apparently, this corresponds to the previous objective function in eq. (3) by setting

$$w_{nk} = \frac{\nu_k + 2}{\nu_k + \mathbf{r}_n^\mathsf{T}\mathbf{\Sigma}_k^{-1}\mathbf{r}_n}. \tag{12}$$

Now we can deduct the EM steps. In the $(t+1)$ E-step, we estimate the posterior probability of $\mathbf{z}$ holding parameters $\boldsymbol{\xi}, \boldsymbol{\eta}, \mathbf{\Sigma}$ from the t M-step fixed. Working out the mathematical details, this yields

$$\gamma_{nk}^{t+1} = \frac{\eta_k^t p_t(\mathbf{r}_n; \mathbf{0}, \mathbf{\Sigma}_k^t, \nu_k)}{\sum_j^K \eta_j^t p_t(\mathbf{r}_n; \mathbf{0}, \mathbf{\Sigma}_j^t, \nu_k)}. \tag{13}$$

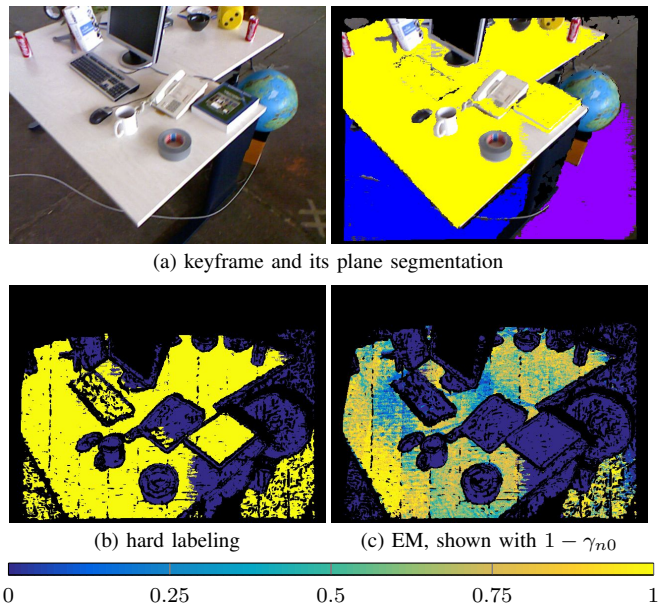
In the $(t+1)$ M-step, we in turn solve for the motion estimation holding the mixing coefficients from the $(t+1)$ E-step fixed. To this end, eq. (11) is iteratively linearized with first-order Taylor approximation of $\mathbf{r}$. At each iteration, this yields a normal equation to solve for an increment $\boldsymbol{\Delta\xi}$ on the motion,

$$\begin{aligned} &\sum_n^N \sum_k^K \gamma_{nk}^{t+1} w_{nk}^{t+1} \mathbf{J}_n^\mathsf{T} \mathbf{\Sigma}_k^{-1} \mathbf{J}_n \Delta\boldsymbol{\xi} \\ &= -\sum_n^N \sum_k^K \gamma_{nk}^{t+1} w_{nk}^{t+1} \mathbf{J}_n^\mathsf{T} \mathbf{\Sigma}_k^{-1} \mathbf{r}_n. \end{aligned} \tag{14}$$

The hyper-parameters are then updated by

$$\eta_k^{t+1} = \frac{1}{N}\sum_n^N \gamma_{nk}^{t+1}, \;\; \mathbf{\Sigma}_k^{t+1} = \frac{\sum_n^N \gamma_{nk}^{t+1} w_{nk}^{t+1} \mathbf{r}_n \mathbf{r}_n^\mathsf{T}}{\sum_n^N \gamma_{nk}^{t+1}}. \tag{15}$$

From the above deduction, we see the alternating optimization principle of EM. The E-step computes the soft labeling given the current parameter values, while the M-step reestimates the parameters based on the latest soft labeling. To implement the EM framework, we use projective data association [16] to guide the iterative EM steps. At each M-step, we warp the current frame into the keyframe and find correspondences for pixels if possible. With the segmented keyframe, the data association propogates the keyframe labeling to the current frame. Since we have small motion between frames and a good initial guess through tracking, the label propogation is mostly correct. Therefore, if a point is associated with plane j, we compute γ_{nj} according to eq. (13) and set $\gamma_{n0} = 1 - \gamma_{nj}$. Otherwise, we set $\gamma_{n0} = 1$. This implementation efficiently approximates the original EM solution due to the fact that a point most likely belongs to only one plane (up to plane intersections).



(a) keyframe and its plane segmentation

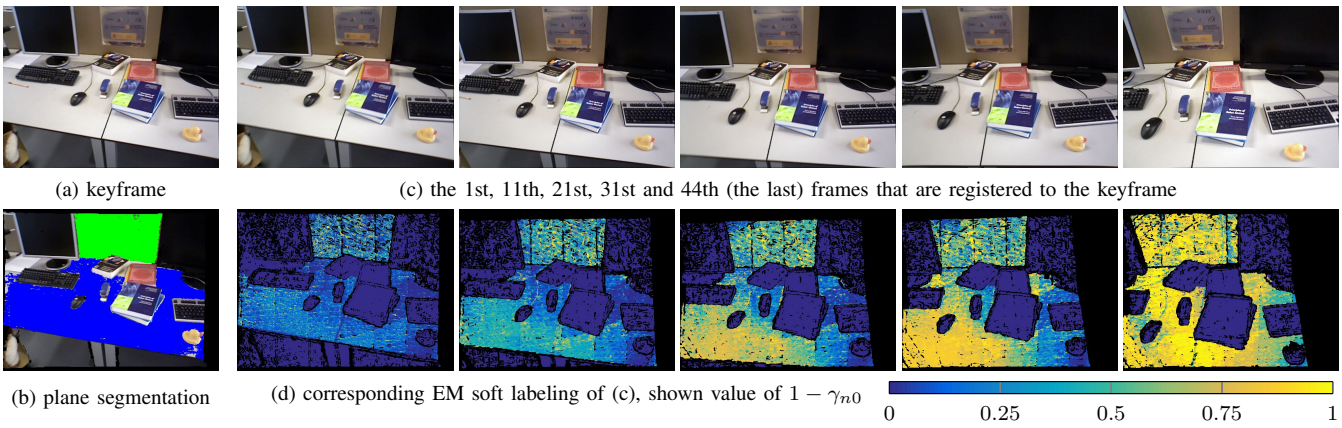
(b) hard labeling (c) EM, shown with $1-\gamma_{n0}$

Fig. 3: Comparison between the hard labeling and EM soft labeling to associate planar points in the current frame. The soft labeling is more robust against the false segmentation in the keyframe, e.g., the keyboard and the book are assigned 0 probability to being on the plane of the table.

D. Properties of the EM Formulation

While our tracking method estimates camera motion, it also estimates the segmentation of the current frame via soft labeling. Note that a hard labeling can be easily obtained using projective data association. However, soft labeling is preferred, as it is more robust against false segmentation. We illustrate this in Fig. 3, where the plane segmentation contains outliers (e.g., the keyboard and the book). These false detections are difficult to avoid due to noise and extreme scenarios with two close parallel planes.

Another property of our EM tracking is demonstrated in Fig. 4, which shows the soft labeling of subsequent frames that are aligned to the same keyframe. In this example, tracking trusts the keyframe data more when the current frame has a small temporal and spatial distance to the keyframe, while it trusts the global plane model more when the current frame is further away. This is logical and as expected. With frames



(a) keyframe

(c) the 1st, 11th, 21st, 31st and 44th (the last) frames that are registered to the keyframe

(b) plane segmentation

(d) corresponding EM soft labeling of (c), shown value of $1 - \gamma_{n0}$

Fig. 4: Properties of our tracking method: While EM trusts the keyframe data more if the current frame has a small temporal and spatial distance to the keyframe, it otherwise relies more on the global planar map. See Sec. III-D for details.

being close to the keyframe, the measurements very well correspond to keyframe data and, hence, direct alignment to the keyframe yields reliable and accurate registration. However, when a frame moves away from the keyframe both temporally and spatially, the difference between the measurements increases and drift accumulates. In such cases, the global plane model becomes beneficial.

E. Keyframe Selection and Loop Closure Detection

Following the work of [1], keyframes are selected by examining the uncertainty of motion estimation. The normal equation (14) provides an approximate Hessian matrix $\mathbf{H}$,

$$\mathbf{H} = \sum_{n}^{N} \sum_{k}^{K} \gamma_{nk}^{t+1} w_{nk}^{t+1} \mathbf{J}_n^\mathsf{T} \mathbf{\Sigma}_k^{-1} \mathbf{J}_n. \tag{16}$$

Its inverse gives a lower bound on the covariance of the estimated motion, i.e., $\mathbf{\Sigma}_{\boldsymbol{\xi}} \approx \mathbf{H}^{-1}$. Assuming $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\xi}^*, \mathbf{\Sigma}_{\boldsymbol{\xi}})$, we can extract the uncertainty embedded in covariance into a scalar value using the differential entropy $h(\boldsymbol{\xi}) = 3\big(1 + \ln(2\pi)\big) + 0.5 \ln(|\mathbf{\Sigma}_{\boldsymbol{\xi}}|)$. Given the current keyframe k, we test the entropy ratio for every consecutive frame tracked towards the keyframe by $\alpha = h(\boldsymbol{\xi}_{k+j})/h(\boldsymbol{\xi}_{k+1})$. Whenever α drops below a pre-defined threshold, the $(k+j)$th frame is selected as the new keyframe. Empirically, we find the value range $0.9 \sim 0.95$ to generate good performance.

Whenever a new keyframe is produced, we find loop closures by comparing the current keyframe to previous keyframes via a spatial search. To register two keyframes, we use direct image alignment and initialize the estimation with the transformation computed from their poses. The same ratio test is performed to determine a successful closure. After the last keyframe being produced, we run an additional loop closure search for all keyframes.

F. Joint Pose and Plane Graph Optimization

On the global scale, we optimize the keyframe poses and the model planes for consistency in a graph

$$\mathbf{\Theta}^* = \arg\min_{\mathbf{\Theta}} \sum_{i,j} \mathbf{e}_{ij}^\mathsf{T} \mathbf{H}_{ij} \mathbf{e}_{ij}, \tag{17}$$

where $\mathbf{e}_{ij}$ is the error of the edge connecting vertices i, j and $\mathbf{H}_{ij}$ is the information matrix. The variable $\mathbf{\Theta}$ is the list of the parameters to be optimized. In our case, $\mathbf{\Theta} = (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_N, \boldsymbol{\pi}_1^g, \boldsymbol{\pi}_2^g, \ldots, \boldsymbol{\pi}_M^g)$. Since there are two types of vertices, keyframe poses and global planes, the graph also consists of two types of edges: between two poses and between a plane and a keyframe pose.

For an edge connecting two poses $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ with the measured constraint $\boldsymbol{\xi}_{ji}$, the edge error is calculated by

$$\mathbf{e}_{ij} = g^{-1}\big(g(\boldsymbol{\xi}_i^{-1}) g(\boldsymbol{\xi}_j) g(\boldsymbol{\xi}_{ji})\big), \tag{18}$$

The Hessian in eq. (16) is used as the information matrix.

Now we define the error for an edge connecting a global plane $\boldsymbol{\pi}_j^g$ and a keyframe pose $\boldsymbol{\xi}_i$, given the local plane observation $\boldsymbol{\pi}_{ji}$ in the keyframe. Notice that the Hessian plane equation is an over-parameterization of 3D planes, since a plane has only three degrees of freedom. Therefore, the Hessian form will lead to complications in the optimization, which requires extra constraints to ensure the unit length of the plane normal. To avoid this problem, we use the minimal parameterization $\boldsymbol{\tau} = (\phi, \psi, d)$, where ϕ and ψ are the azimuth and elevation angle of the normal, respectively. The following conversion between the Hessian and the minimal representation applies,

$$\boldsymbol{\tau} = q(\boldsymbol{\pi}) = (\phi = \arctan \frac{n_y}{n_x}, \psi = \arccos n_z, d)^\mathsf{T}, \tag{19}$$

$$q^{-1}(\boldsymbol{\tau}) = (\cos\phi \cos\psi, \sin\phi \cos\psi, -\sin\psi, d)^\mathsf{T}. \tag{20}$$

To avoid the singularities of the minimal representation, we force the angle to fall into $(-\pi, \pi]$. The error for plane-keyframe edges is then defined as

$$\mathbf{e}_{ij} = q(\boldsymbol{\pi}_j^g) - q\big(t(\boldsymbol{\xi}_i, \boldsymbol{\pi}_{ji})\big). \tag{21}$$

The information matrix for plane-keyframe edges is set to isotropic using $\sigma_\pi^{-2} I$.

We optimize the graph when a new keyframe is inserted into the graph using the g2o [17] framework.

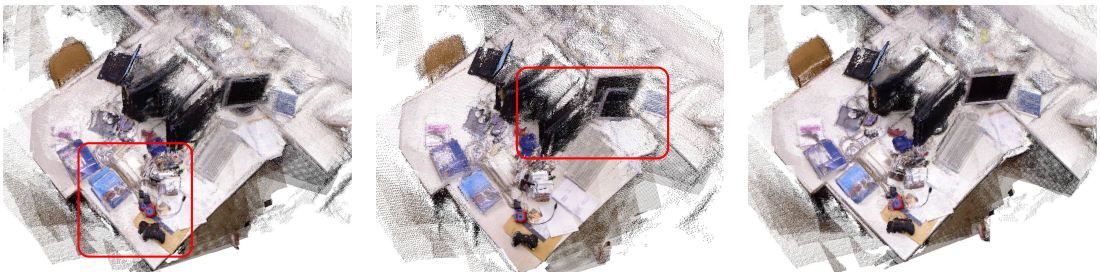


Fig. 5: Comparison of accumulated point clouds from keyframes using SLAM. From left to right: tracking without planes, hard labeling the planes, and soft labeling the planes (our EM approach).

IV. Experimental Results

In this section, we evaluate our EM tracking method and the overall SLAM algorithm. Two public datasets with groundtruth trajectories are used in the assessment: the TUM RGB-D benchmark [18] and the ICL-NUIM synthetic scenes [19]. We also evaluate our algorithm with the Stanford scene3D dataset [20], which has longer and more complicated trajectories. The EM tracking is implemented with CUDA and run on an NVidia GTX780 GPU with 2304 cores, 3.7 GHz, and 3 GB memory. The remaining SLAM methods are implemented in C++ to run on CPU and evaluated with an Intel Core i7-2660, 3.4 GHz and 8 GB RAM.

In the first experiment, we evaluate the performance of our EM tracking method by comparing with two other variants: a) tracking without planes and b) tracking with planes using hard-labeling. We set $\alpha = 0.9$ for keyframe selection in all methods and compare the obtained trajectories without the final graph optimization. Table I shows the root mean square error (RMSE) of the absolute trajectory error (ATE) and Fig. 5 gives a visual comparison of the alignment results. It can be seen that our tracking improves trajectory accuracy and yields better consistent models.

In the second experiment, we evaluate the benefits of the global plane model in our full CPA-SLAM system. Table II presents a comparison between our method and two other SLAM algorithms that use planes: dense planar SLAM [9] and point-plane SLAM [7]. Our EM-based algorithm achieves much better accuracy on the sequences. These improvements come from joining the advantages of keyframe-based and model-based tracking into a flexible and robust EM framework.

We also compare our CPA-SLAM algorithm to several state-of-the-art RGB-D SLAM systems, including DVO-SLAM [1], Kintinuous with deformable mapping [21], MRSMap [22], and RGB-D SLAM [23]. We report the ATE of the final trajectories in Table III. The results showcase that our SLAM algorithm performs better or on par with the state-of-the-art algorithms on the sequences. For sequences with many planar structures or containing heavy noise, e.g., fr3/nst, iclnuim/lr2noisy and iclnuim/lr3noisy, our algorithm demonstrates most advantages in reducing tracking errors. Fig. 6 shows the output of our SLAM methods. It can be seen that with a global plane model, constraints between non-overlapping keyframes are established, as long as they observe the same global plane. As can be seen, geometry is accurately reconstructed from the estimated trajectory.

TABLE I: RMSE of absolute trajectory error (no final optimization) of tracking methods: without plane model, plane model with hard labeling and plane model with soft EM labeling (bold marks the best).

dataset	without plane	hard labeling	soft labeling
fr1/desk	0.034	0.080	**0.030**
fr1/plant	**0.050**	0.072	0.073
fr2/desk	0.097	0.134	**0.095**
fr3/office	0.086	0.077	**0.076**
fr3/structure_texture_near	0.049	**0.028**	0.036
fr3/nst	0.076	**0.032**	**0.032**
iclnuim/lr3	**0.002**	0.049	**0.002**
iclnuim/lr3noisy	0.028	0.024	**0.019**

TABLE II: Comparison of our CPA-SLAM to other SLAM algorithms that use planes. The RMSE of the absolute trajectory error (m) is shown and the results of other methods are cited from the original papers.

dataset	CPA-SLAM	planar SLAM [9]	point-plane SLAM [7]
iclnuim/lr0noisy	**0.007**	0.246	–
iclnuim/lr1noisy	**0.006**	0.017	–
fr1/xyz	**0.011**	–	0.024
fr1/floor	0.085	–	**0.065**

Table IV presents the runtime of the computationally intensive parts of our method. Tracking achieves real-time performance with approximately 50 fps. Plane segmentation is also efficiently performed with approximately 100 fps. Combining frame tracking, plane segmentation and association, our implementation requires around 40 ms to process one keyframe. Searching loop closures and optimizing the graph requires relatively long time, however, this can be run on parallel CPU threads. As a result, our method is capable of real-time performance for 30 fps RGB-D mapping.

V. Conclusion

In this paper, we proposed a novel method that combines direct image alignment and global model alignment for RGB-D SLAM. Our method tracks camera motion towards the nearest keyframe and the global plane model in an EM framework. This reduces drift and establishes constraints among non-overlapping keyframes that observe the same plane. The keyframe poses and the plane model are optimized in one graph concurrently. Our method exhibits state-of-the-art accuracy on publicly available benchmark datasets and is capable of real-time performance. In future work, we will

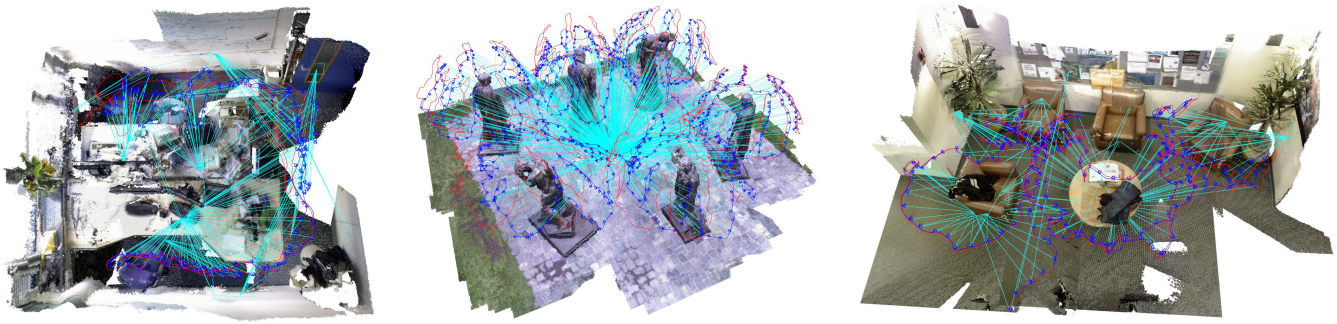


Fig. 6: Fused model by our SLAM methods. The trajectories with and without graph optimization are shown in blue and red, and the constraints between keyframe poses and planes are shown in cyan.

TABLE III: The RMSE of the absolute trajectory error (m) of our CPA-SLAM approach in comparison to state-of-the-art algorithms (bold marks best).

SLAM system	fr1/desk	fr1/desk2	fr1/plant	fr1/room	fr1/rpy	fr1/xyz	fr2/desk	fr2/xyz	fr3/office	fr3/nst	iclnuim/lr2noisy	iclnuim/lr3noisy
CPA-SLAM	**0.018**	**0.029**	0.029	0.055	0.024	**0.011**	0.046	**0.014**	**0.025**	**0.016**	**0.089**	**0.009**
DVO-SLAM	0.021	0.046	0.028	**0.053**	**0.020**	**0.011**	**0.017**	0.018	0.035	0.038	0.339	0.152
Kintinous	0.037	0.071	0.047	0.075	0.028	0.017	0.034	0.029	0.030	0.031	0.129	0.864
MRSMap	0.043	0.049	**0.026**	0.069	0.027	0.013	0.052	0.020	0.042	1.530	0.331	1.127
RGB-D SLAM	0.023	0.043	0.091	0.084	0.026	0.014	0.095	0.026	–	–	–	–

TABLE IV: The average runtime performance of our algorithm in ms. Frame tracking runs well within the typical camera frame-rate of 30 fps. Plane segmentation and association has to run only at each keyframe creation. The comparably slow graph optimization runs in a background thread. Therefore, our RGB-D SLAM is able to perform in realtime.

	fr3/office	fr3/nst	iclnuim/lr3noisy
keyframes/ frames	118/ 2489	66/ 1637	59/ 1241
global planes/ local planes	18/ 197	1/ 66	7/ 64
frame tracking (ms)	20.9	24.1	23.2
plane segmentation (ms)	9.3	14.4	9.8
plane association (ms)	10.7	8.4	8.4
graph optimization (ms)	75.8	24.0	22.0

consider the integration of further types of geometric shapes and complex objects into our SLAM system. One important question here is how to learn and acquire object models on-the-fly, and how to come to adequate object hypotheses.

References

[1] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robot Systems (IROS)*, 2013.

[2] A. P. Gee, D. Chekhlov, W. Mayol-Cuevas, and A. Calway, "Discovering planes and collapsing the state space in visual SLAM," in *Proc. of the British Machine Vision Conference (BMVC)*, 2007.

[3] F. Servant, E. Marchand, P. Houlier, and I. Marchal, "Visual planes-based simultaneous localization and model refinement for augmented reality," in *Proc. of the Int. Conf. on Pattern Recognition (ICPR)*, 2008.

[4] J. Stückler and S. Behnke, "Orthogonal wall correction for visual motion estimation.," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2008.

[5] J. Martinez-Carranza and A. Calway, "Unifying planar and point mapping in monocular slam," in *Proc. of the British Machine Vision Conference (BMVC)*, 2010.

[6] M. Dou, L. Guan, J.-M. Frahm, and H. Fuchs, "Exploring high-level plane primitives for indoor 3D reconstruction with a hand-held RGB-D camera," in *Computer Vision - ACCV Workshops*, pp. 94–108, 2012.

[7] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[8] A. Trevor, J. Rogers, and H. Christensen, "Planar surface SLAM with 3D and 2D sensors," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[9] R. F. Salas-Moreno, B. Glocker, P. H. J. Kelly, and A. J. Davison, "Dense planar SLAM," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2014.

[10] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3D reconstruction in dynamic scenes using point-based fusion," in *Proc. of Int. Conf. on 3D Vision (3DV)*, 2013.

[11] 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 *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1352–1359, 2013.

[12] N. Fioraio and L. di Stefano, "Joint detection, tracking and mapping by semantic bundle adjustment.," in *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1538–1545, 2013.

[13] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[14] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. PAMI.*, vol. 14, pp. 239–256, Feb 1992.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006.

[16] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. of IEEE ISMAR*, 2011.

[17] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

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

[19] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[20] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Trans. Graph.*, vol. 32, pp. 112:1–112:8, July 2013.

[21] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. Leonard, and J. McDonald, "Real-time large scale dense RGB-D SLAM with volumetric fusion," *Int. J. of Robotics Research, IJRR*, 2015.

[22] J. Stückler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3d modeling and tracking," *J. of Visual Communication and Image Representation (JVCI)*, vol. 25, pp. 137–147, Jan. 2014.

[23] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.