Vision Control - final project

CPA-SLAM: Consistent Plane-Model Alignment

for Direct RGB-D SLAM

Sun Qinxuan

June 22, 2017

1 Preface

While presenting the ideas and detailed methods proposed in this paper, I will insert some comments of mine into the context, which are all identified with blue characters. (I don't exactly know whether this could be called a preface, but I know no better alternatives.) – by SqX.

2 Paper Overview

This paper proposes an RGB-D SLAM system based on consistent plane-model alignment. The proposed method models the environment with a global plane model and integrates frame-to-keyframe and frame-to-plane alignment. Compared with most of the other plane based SLAM system, this paper makes use of the dense image information available in keyframes for accurate short-term tracking, while uses a global model to reduce drift.

The whole system is illustrated in Figure 1. The major contributions of the paper are as follows.

- An RGB-D SLAM approach is developed which consistently tracks camera motion through direct image alignment towards a keyframe and a global plane model in an EM framework.
- Some spatial constraints are obtained between keyframes and global plane model.
- A real-time capable SLAM system is developed.

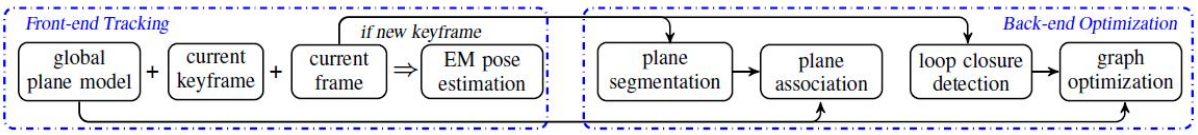

Figure 1: Schematic pipeline of CPA-SLAM system.

3 Method Description

3.1 Preliminaries

For the detailed description, the following notations are used.

k	Keyframe index;
i	Current frame index;
$\Omega \in \mathbb{R}^2$	image domain;
Ω_i	Disjoint segments of image domain;
$\mathbf{v}$	A 3D point;
$\mathbf{n}$	Unit normal at a 3D point;
$\mathbf{x}$	A 2D pixel;
$\mathbf{x} = \rho(\mathbf{v})$	Projection of a 3D point onto the 2D image;
$\mathbf{v} = \rho^{-1}(\mathbf{x})$	Back-projection of a 2D pixel into 3D space;
$\xi_{ji} \in \mathfrak{se}(3)$	Rigid body motion from frame i to j;
$t(\xi, \mathbf{v}) = g(\xi)\mathbf{v}$	Transforming 3D point $\mathbf{v}$ by ξ;
$\omega(\xi, \mathbf{x}) = \rho(t(\xi, \rho^{-1}(\mathbf{x})))$	Warping pixels between frames;
$\pi = (\mathbf{n}^T, d)^T$	Plane parameters with $\mathbf{n}$ representing the unit normal and $-d$ being the distance from the plane to the origin.

3.2 Global Plane Model

The global plane model is defined as a set of planes $\{\pi_m^g\}$ in the world coordinate. It is augmented incrementally. When a new keyframe is produced, it is segmented into K regions where Ω_0 represents the non-planar region and Ω_j is the j-th plane. When associating the local observations with the global model, a correspondence is found is the angle between the plane normals is small and their distances to the origin are similar.

Actually I don't think it's a very good way to correspond the planes. If the parameters of plane are used, it must be guaranteed that the camera pose tracking is accurate enough to make the plane parameters observed in two different frames close enough to achieve the correct correspondences. Also, the threshold here is hard to choose to guarantee a good performance in the system implementation.

3.3 Tracking towards Keyframe and Plane Model

The motion estimation from the current frame i to the keyframe k is achieved by minimizing both photometric error r_I and geometric error r_G. The two errors are defined by (1) and (2), respectively.

$$r_I = I_k(\omega(\xi, \mathbf{x})) - I_i(\mathbf{x}) \tag{1}$$

$$r_G = \begin{cases} \mathbf{n}_k^T \left(g(\xi, \mathbf{v}_i) - \mathbf{v}_k\right) & if\ \omega(\xi, \mathbf{x}_i) \in \Omega_0 \\ \mathbf{n}_{\pi j}^T g(\xi, \mathbf{v}_i) + d_j & if\ \omega(\xi, \mathbf{x}_i) \in \Omega_j \end{cases} \tag{2}$$

Note that the geometric residual is defined as the point-to-plane distance when the current pixel $\mathbf{x}_i$ is warped to the planar region Ω_j.

Combining the photometric and geometric error into one variable $\mathbf{r} = (r_I, r_G)^T$, and the camera motion is calculated by minimizing the following non-linear weighted least

squares

$$\xi^* = \arg\min_{\xi} \sum_{n}^{N} \sum_{k}^{K} \gamma_{nk} \omega_{nk} \mathbf{r}_n^T \Sigma_k^{-1} \mathbf{r}_n. \tag{3}$$

The weight ω_{nk} is derived from a Student-t distribution as proposed in [1] and the variable $\gamma_n \in \mathbb{R}^K$ is the labeling that indicates which region the residual belongs to. Note that a soft labeling $\gamma_{nk} \in [0, 1]$ is used to increase robustness.

The idea of soft labeling here is interesting. It describes how likely a point belongs to a plane rather than whether the point belongs to the plane.

Since the parameters γ, ω and Σ also need to be estimated in addition to the motion ξ, the optimization of (3) cannot be solved directly.

Suppose that $K - 1$ planes are visible in the keyframe. Assume that there exists an indicator $\mathbf{z}_n \in \mathbb{B}^K$ that tells which segment the pixel comes from. It satisfied $z_{nk} \in \{0, 1\}$ and $\sum_k^K z_{nk} = 1$. As a result, the variable $\mathbf{z}_n$ can be seen as a lattent variable with the following probability

$$p(\mathbf{z}_n) = \prod_{k}^{K} \eta_k^{z_{nk}}, \tag{4}$$

$$p(\mathbf{r}_n | \mathbf{z}_n) = \prod_{k}^{K} p_t(\mathbf{r}_n; 0, \Sigma_k, \nu_k)^{z_{nk}}. \tag{5}$$

The EM algorithm provides a probabilistic formalism to estimate the parameters of posterior probability functions with latent variables. In EM, the conditional expectation of the log joint probability is optimized, which is conditioned on the posterior probability of the latent variable. In this case, the log joint probability can be computed as

$$\begin{aligned} \log p(\mathbf{r}, \mathbf{z}) &= \log \prod_{n}^{N} \prod_{k}^{K} \left(\eta_k p_t(\mathbf{r}_n; 0, \Sigma_k, \nu_k)\right)^{z_{nk}} \\ &= \sum_{n}^{N} \sum_{k}^{K} z_{nk} \log \left(\eta_k p_t(\mathbf{r}_n; 0, \Sigma_k, \nu_k)\right). \end{aligned} \tag{6}$$

Then the conditional expectation is computed as

$$\mathbb{E}_{p(\mathbf{z}|\mathbf{r})} \left[\log p(\mathbf{r}, \mathbf{z})\right] = \sum_{n}^{N} \sum_{k}^{K} z_{nk} \gamma_{nk} \log \left(\eta_k p_t(\mathbf{r}_n; 0, \Sigma_k, \nu_k)\right), \tag{7}$$

where

$$\gamma_{nk} = \mathbb{E}_{p(\mathbf{z}|\mathbf{r})}[z_{nk}] = p(z_{nk} | \mathbf{r}_n). \tag{8}$$

So in the E-step of the EM process, the soft label γ_{nk} is calculated holding all the parameters from the last time step. In the M-step, the motion estimation and other unknown parameters is solved by maximizing (7). The robustness of soft labeling is illustrated in Figure 2.

It is said in the paper that the iterative EM steps are guides by the projective data association [2], which propagates the keyframe labeling to the current frame. Essentially, it's similar to the ICP process. Instead of determining the point pair as correspondence, they associate points with plane segments if possible. In the EM process, the soft labeling is treated as a hidden variable, and estimated in each iteration together with the camera

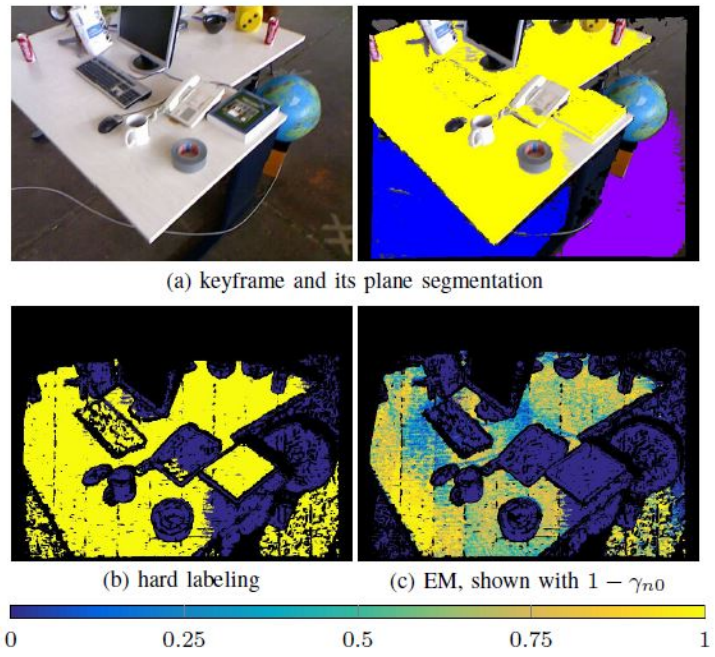

Figure 2: Comparison between the hard labeling and EM soft labeling to associate planar points in the current frame. The soft labeling is more robust against the false segmentation in the keyframe, e.g., the keyboard and the book are assigned 0 probability to being on the plane of the table.

motion. It's kind of like the data association part in ICP process. Since the correspondences between two scans are unknown, so they need to be estimated before calculating the camera motion, and re-estimated in the iterative process.

3.4 Keyframe Selection and Loop Closure Detection

Keyframes are selected by examining the uncertainty of motion estimation [1]. The Hessian matrix $\mathbf{H}$ can be approximated given the normal equation in the optimization process. The covariance of the estimated motion is approximated by the inverse of $\mathbf{H}$, i.e., $\Sigma_\xi \approx \mathbf{H}^{-1}$. Assuming $\xi \sim \mathcal{N}(\xi^*, \Sigma_\xi)$, the differential entropy of a multivariate normal distribution is defined as

$$h(\xi) = 3(1 + \ln(2\pi)) + 0.5 \ln(|\Sigma_\xi|). \tag{9}$$

The entropy ratio for every frame track towards the keyframe is

$$\alpha = h(\xi_{k+j})/h(\xi_{k+1}). \tag{10}$$

Whenever α drops below a pre-defined threshold, the $(k + j)$-th frame is selected as the keyframe. And whenever a keyframe is produced, the loop closure is detected by comparing the current keyframe to previous keyframes via a spatial search and the direct image alignment is used to register two frames.

The loop closure detection hasn't utilized any information of the global plane model. I think the global plane model should be exploited in the loop closure detection, which might improve the work in this paper.

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**

Figure 3: 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	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**

Figure 4: Comparison of 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.

3.5 Joint Pose and Plane Graph Optimization

The keyframe poses and the model planes are together optimized in a graph

$$\Theta^* = \arg\min_{\Theta} \sum_{i,j} \mathbf{e}_{ij}^T \mathbf{H}_{ij} \mathbf{e}_{ij}, \tag{11}$$

where $\Theta = (\xi_1, \xi_2, ..., \xi_N, \pi_1^g, \pi_2^g, ..., \pi_M^g)$ is the parameters to be optimized. The graph contains two types of nodes: keyframe poses and global planes, and two types of edges: between two poses and between a plane and a pose.

For an edge connecting two poses ξ_i and ξ_j with the measured constraint ξ_{ij}, the edge error is defined as

$$\mathbf{e}_{ij} = g^{-1}(g(\xi_i^{-1})g(\xi_j)g(\xi_{ij})). \tag{12}$$

And the error for plane-keyframe edges is defined as

$$\mathbf{e}_{ij} = q(\pi_j^g) - q(t(\xi_i, \pi_{ij})). \tag{13}$$

4 Experimental Evaluation

Some experimental results are shown in Figure 3, Figure 4 and Figure 5. The EM tracking is implemented with CUDA and run on an NVidia GTX780 GPU.

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.

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	–	–	–	–

Figure 5: The RMSE of the absolute trajectory error (m) of CPA-SLAM approach in comparison to state-of-the-art algorithms (bold marks best).

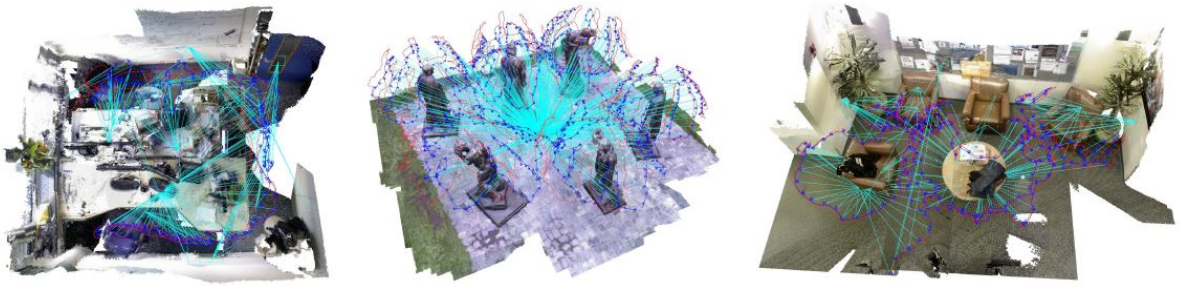

Figure 6: Fused model by proposed SLAM methods. The trajectories with and without graph optimization are shown in blue and red, and the constraints between keyframe poses and planes are shown in cyan.

[2] 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.