L16: Brief Survey on Visual SLAM

Perception in Robotics, Term 2, 2018

Gonzalo Ferrer

Skolkovo Institute of Science and Technology

December 7, 2018

Mono SLAM [1] (Davison 2007)

"...first successful application of the SLAM methodology from mobile robotics to the "pure vision" domain of a single uncontrolled camera"

Idea: online-SLAM using EKF with the following particularities:

- Sparse but persistent features (image patches).
- General motion model for smooth camera movement.
- Active approach to mapping and measurement.
- Feature initialization and feature orientation estimation.

[1] Davison et al. "MonoSLAM: Real-Time Single Camera SLAM", PAMI 2007

Mono SLAM [1] (Davison 2007)

Videos.

[1] Davison et al. "MonoSLAM: Real-Time Single Camera SLAM", PAMI 2007

PTAM [2] (Klein 2007)

- Introducing the concept of Keyframe.
- Splits Tracking and Mapping, running in two parallel threads.
- Mapping is based on keyframes, which are processed using batch techniques.
- Camera pose is tracked (localized) based on keyframes.

More on `http://www.robots.ox.ac.uk/~gk/youtube.html`

[2] Klein and Murray "Parallel Tracking and Mapping for Small AR Workspaces", ISMAR 2007

Filtering vs Keyframes [3] (Strasdat 2011)

MonoSLAM vs PTAM for real-time applications.

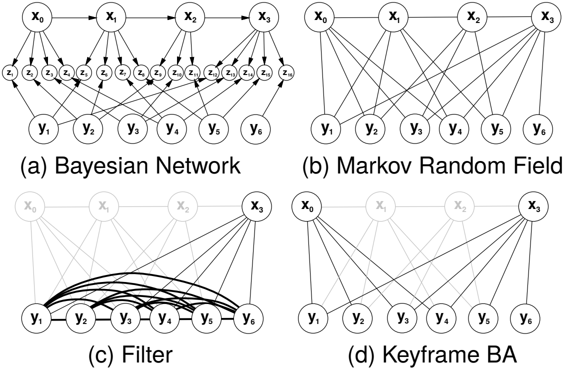


Fig. 1. (a) Bayesian network for SLAM/SFM. (b) SLAM/SFM as markov random f eld without representing the measurements explicitly. (c) and (d) visualise how inference progressed in a f lter and with keyframe-based optimisation.

---

[3] Strasdat, Montiel and Davison “Real-time Monocular SLAM: Why Filter”, ICRA 2010

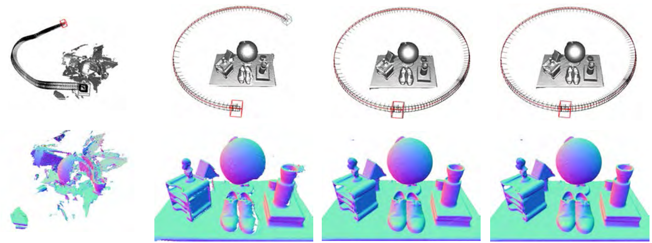
Filtering vs Keyframes [3] (Strasdat 2011)

After a fair comparison from both methods:

- Increasing the number of features improves results, while increasing the number of intermediate frames has a minor impact.
- Filtering is only superior for very small processing budgets and smoothing is superior otherwise.

[3] Strasdat, Montiel and Davison "Real-time Monocular SLAM: Why Filter", ICRA 2010

KinectFusion [4] (Newcombe 2011)



(a) Frame to frame tracking (b) Partial loop (c) Full loop (d) M times duplicated loop

- The idea is to marginalize maps at each observation.
- They maintain an explicit map of the environment (fine grid) which allows for a "highly certain" alignment of observations.

[4] Newcombe, et al. "Real-Time Dense Surface Mapping and Tracking", ISMAR 2011

KinectFusion [4] (Newcombe 2011)

At every time step it is equivalent to marginalize the current pose, and simply aggregates information to the map:

$$\begin{aligned} p(m|z_{1:t}, u_{1:t}) &= \int_{x_t} p(m, x_t|z_{1:t}, u_{1:t}) dx_t \\ &= \int_{x_t} p(m|x_t, z_{1:t}, u_{1:t}) p(x_t|z_{1:t}, u_{1:t}) dx_t \\ &\sim \int_{x_t} p(m|x_t, z_{1:t}, u_{1:t}) \delta(x_t|z_{1:t}, u_{1:t}) dx_t \end{aligned}$$

`https://www.youtube.com/watch?v=quGhaggn3cQ`

[4] Newcombe, et al. "Real-Time Dense Surface Mapping and Tracking", ISMAR 2011

DTAM [5] (Newcombe 2011)

- Monocular dense (every pixel) reconstruction (PTAM with dense frame alignments).
- Volumetric reconstruction of each keyframe with millions of vertices.
- Camera motion estimation at frame-rate.
- Optimization over a trajectory of sparse keyframes (SLAM, Bundle Adjustment, Structure from Motion).

Why dense methods? they use all data in the image to get more complete, accurate and robust results (see MonoSLAM vs PTAM [3])

[5] Newcombe, Lovegrove and Davison "DTAM: Dense Tracking and Mapping in Real-Time", ICCV 2011

DTAM [5], Dense mapping

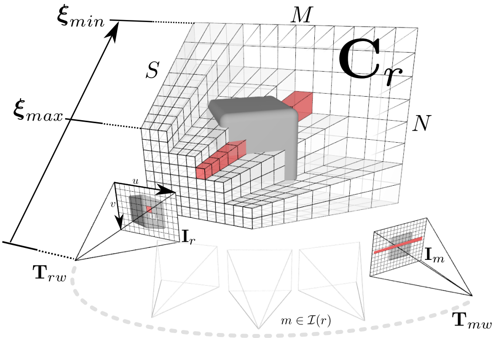


Figure 1. A keyframe r consists of a reference image I_r with pose T_{rw} and data cost volume C_r. Each pixel of the reference frame u_r has an associated row of entries $C_r(u)$ (shown in red) that store the average photometric error or cost $C_r(u, d)$ computed for each inverse depth $d \in D$ in the inverse depth range $D = [\xi_{min}, \xi_{max}]$. We use tens to hundreds of video frames indexed as $m \in I(r)$, where $I(r)$ is the set of frames nearby and overlapping r, to compute the values stored in the cost volume.

[5] Newcombe, Lovegrove and Davison "DTAM: Dense Tracking and Mapping in Real-Time", ICCV 2011

DTAM [5] Incremental cost volume

A sequence of images is required.

Figure 3. Incremental cost volume construction; we show the current inverse depth map extracted as the current minimum cost for each pixel row $d_u^{min} = \arg\min_d C(u, d)$ as 2, 10 and 30 overlapping images are used in the data term (left). Also shown is the regularised solution that we solve to provide each keyframe inverse depth map (4th from left). In comparison to the nearly 300×10^3 points estimated in our keyframe, we show the ≈ 1000 point features used in the same frame for localisation in PTAM ([6]). Estimating camera pose from such a fully dense model enables tracking robustness during rapid camera motion.

Tracking: Given a dense model (keyframe) we can align it with our current image observation by projecting the volume into an image by a virtual pose and minimizing the error between observation and keyframe.

`http://youtu.be/Df9WhgibCQA`

---

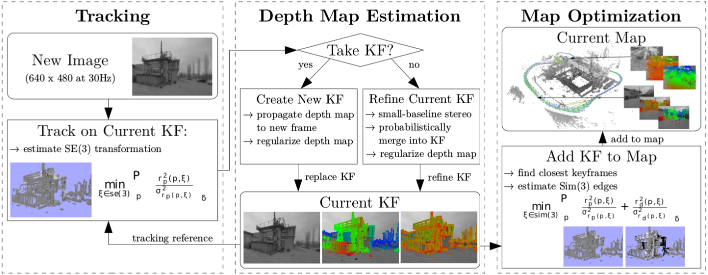
[5] Newcombe, Lovegrove and Davison "DTAM: Dense Tracking and Mapping in Real-Time", ICCV 2011

LSD-SLAM [6] (Engel 2014)

- Semi dense depth reconstructions: Only those pixels with non-zero gradient are considered.
- Large-scale maps.
- Scale-drift and Scale ambiguity solved.
- Global map as a pose graph consisting of keyframes.

[6] Engel, Schops and Cremers "LSD-SLAM: Large-Scale Direct Monocular SLAM", ECCV 2014

LSD-SLAM [6] (Engel 2014)



(See Sec. 3.3) (See Sec. 3.4) (See Sec. 3.2, 3.5 and 3.6)

- Tracking of new camera poses w.r.t keyframe.
- Depth map estimation, refines current keyframe or creates a new one.
- Map optimization (SLAM, Bundle Adjustment, Structure from Motion).

[6] Engel, Schops and Cremers "LSD-SLAM: Large-Scale Direct Monocular SLAM", ECCV 2014