Fusion based Dense SLAM

Sun Qinxuan

April 11, 2017

TO BE ADDRESSED

- Basic Concepts
- Some Challenges in SLAM
- State-of-the-art V-SLAM Systems
- Widely Used Techniques in V-SLAM
    - Tracking
    - Mapping
    - Loop Closing
    - Map Optimizing
- Fusion based dense SLAM
    - KinectFusion
    - ElasticFusion

Basic Concepts

SLAM (Simultaneous Localization and Mapping) [1]

- Estimation of the robot state (equipped with sensors).
  - Pose (position and orientation).
  - Velocity.
  - Calibration parameters.
- Construction of a model (*the map*) of the environment.

Basic Concepts
Map representation (3D) [1]

- Different kinds of map representations.
  - Landmark-based (feature-based) sparse representations.
    - Represent the scene as a set of *sparse* landmarks.
    - Each landmark corresponds to discriminative features.
    - Point features (most widely used).
  - Raw dense representations.
    - A large unstructured set of points or polygons.
    - *surfels* used in ElasticFusion.
    - (in monocular SLAM) Direct methods.
  - Boundary and spatial-partitioning dense representations.
    - Explicitly represent surfaces (or boundaries) and volumes.
    - Simple boundary representation: plane-based models.
    - Volume representation: truncated signed-distance function (TSDF).
    - TSDF used in KinectFusion.
  - High-level object-based representations.

Basic Concepts

Map representation (3D) [1]

- Comparison between sparse and dense map representations.
  - Feature-based approaches:
    - High speed.
    - Reliance on feature type, detection and matching thresholds.
    - Problems of incorrect correspondences.
  - Dense, direct methods:
    - Exploit all the information in the image.
    - Outperform feature-based methods in scenes with poor texture and motion blur.
    - require high computing power (GPUs) for real time performance.

Basic Concepts

Figure: Left: feature-based map of a room produced by ORB-SLAM. Right: dense map of a desktop produced by DTAM.

Some Challenges in SLAM

- Robust performance
- Scalability
- High level understanding of the environment

Some Challenges in SLAM

- Robust performance
  - Data association
    - Perceptual aliasing
    - Dynamics in the environment
  - Sensor or actuator degradation

Some Challenges in SLAM

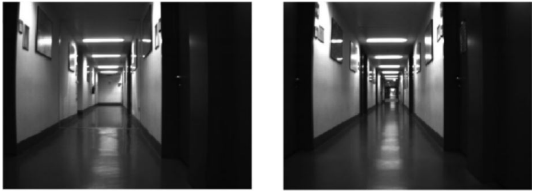


Figure: Perceptual aliasing.

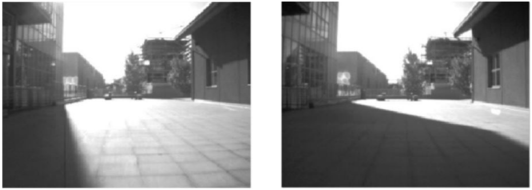


Figure: Dynamics in the environment.

Some Challenges in SLAM

- Scalability

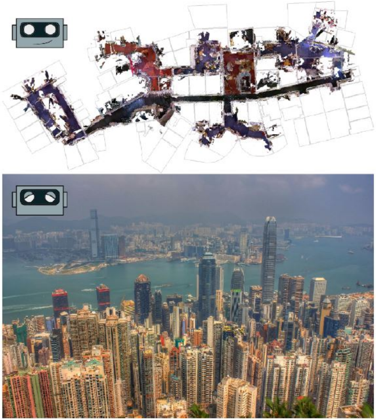


Figure: In some applications, robots need to operate for an extended period of time over large areas.

Some Challenges in SLAM

- Scalability
  - Two ways to reduce complexity of graph optimization
    - Sparsification methods
    - Multi-robot methods

Some Challenges in SLAM

- High level understanding of the environment
  - Semantic SLAM
  - Task related
  - Place/Object classification
  - Properties/Functions

State-of-the-art V-SLAM Systems

Sensor	Map	SLAM (visual odometry)	published	GPU required
monocular	sparse	ORB-SLAM [5]	2015, TRO	No
		SVO [7, 8]	2014, ICRA	No (MAV)
	semi-dense	LSD-SLAM [6]	2014, ECCV	No
	dense	DTAM [9]	2011, ICCV	Yes
RGB-D	sparse	RGBD-SLAM [10]	2014, TRO	No
	dense	DVO [11, 12]	2013, IROS	No
		KinectFusion [13]	2011, ISMAR	Yes
		ElasticFusion [14, 15]	2015, RSS	Yes

State-of-the-art V-SLAM Systems

SLAM (visual odometry)	developers
ORB-SLAM [5]	Instituto de Investigacion en Ingenieria de Aragon Universidad de Zaragoza, Spain
SVO [7, 8]	Robotics and Perception Group University of Zurich, Switzerland
LSD-SLAM [6]	Computer Vision Group, Department of Computer Science Technical University Munich, Germany
DTAM [9]	Robot Vision Research Group, Department of Computing Imperial College London, UK
RGBD-SLAM [10]	Department of Computer Science University of Freiburg, Germany
DVO [11, 12]	Computer Vision Group, Department of Computer Science Technical University of Munich, Germany
KinectFusion [13]	Microsoft
ElasticFusion [14, 15]	Dyson Robotics Laboratory, Department of Computing Imperial College London, UK

Widely Used Techniques in V-SLAM

Widely Used Techniques in V-SLAM

- Tracking
- Mapping
- Loop Closing
- Map Optimizing

Widely Used Techniques in V-SLAM

Tracking

- Feature-based method
- Direct method

Widely Used Techniques in V-SLAM

Tracking

- Feature-based method (ORB-SLAM)
  - Current camera pose prediction via a motion model.
  - Data association - achieved by feature matching (ORB features).
  - Bundle Adjustment.

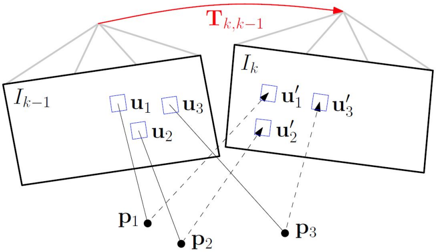
Widely Used Techniques in V-SLAM
Tracking

- Direct method
  - Minimize the projective photometric error.

$$T_{k,k-1} = \underset{T}{\operatorname{argmax}} \int \delta I(T, \mathbf{u}) d\mathbf{u}$$

  - where

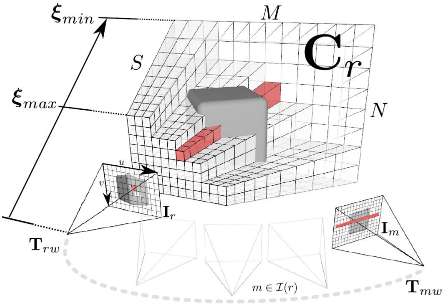
$$\delta I(T, \mathbf{u}) = I_k \left(\pi \left(T \cdot \pi^{-1}(\mathbf{u}, d_u) \right) \right) - I_{k-1}(\mathbf{u})$$



Widely Used Techniques in V-SLAM

Tracking

- Direct method
  - DTAM



Widely Used Techniques in V-SLAM

Tracking

- Direct method
  - DTAM

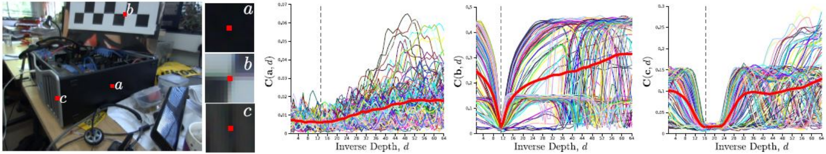
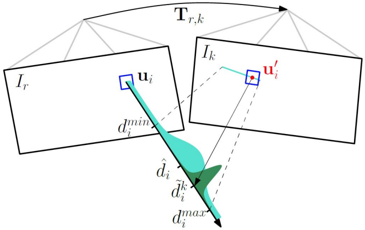


Figure: Plots for the pixel photometric functions.

Widely Used Techniques in V-SLAM
Mapping (Monocular SLAM)

- Depth filter (SVO)
  - Bayesian framework.
  - Initialized with a high uncertainty.
  - Depth measurement is modeled with a *Gaussian + Uniform* mixture model distribution.
  - Recursive Bayesian update.



Widely Used Techniques in V-SLAM

Loop closing

- Geometric-based method
  - Usually for small-scale loop closure detection.
- Appearance-based method
  - Usually for large-scale loop closure detection.
  - Matching between Keyframes (RGBD-SLAM)
  - Bag of Words [16] (ORB-SLAM)
  - FAB-MAP [17] (LSD-SLAM)

Widely Used Techniques in V-SLAM

Map Optimizing

- Pose graph optimization (RGBD-SLAM, LSD-SLAM)
- Fusion based map update (KinectFusion, ElasticFusion)

Fusion based Dense SLAM

- RGB-D sensor.
- Map-centric approach.
- Fuse the data from a moving sensor into a single global surface model, permitting accurate viewpoint-invariant localization as well as offering the potential for detailed scene understanding.
- Two examples
    - KinectFusion
    - ElasticFusion

Fusion based Dense SLAM

KinectFusion

- Preliminaries.
  - 6DOF camera pose estimation at frame k

$$T_{g,k} = \begin{bmatrix} \mathbf{R}_{g,k} & \mathbf{t}_{g,k} \\ \mathbf{0}^T & \mathbf{1} \end{bmatrix} \in \mathbb{SE}_3.$$

  - $\mathbf{p}_g = \mathrm{T}_{\mathrm{g,k}}\mathbf{p}_{\mathrm{k}}$.
  - Camera calibration matrix K.
  - $\mathbf{q} = \pi(\mathbf{p})$ perspective projection,
    where $\mathbf{p} \in \mathbb{R}^3 = (x,y,z)^T$, $\mathbf{q} \in \mathbb{R}^2 = (x/z, y/z)^T$.
  - Homogeneous vector $\dot{\mathbf{u}} := (\mathbf{u}^T|1)^T$.
  - Raw depth map $\mathrm{R}_{\mathrm{k}}(\mathbf{u}) \in \mathbb{R}$, where $\mathbf{u} \in \mathscr{U} \subset \mathbb{R}^2$

Fusion based Dense SLAM

KinectFusion

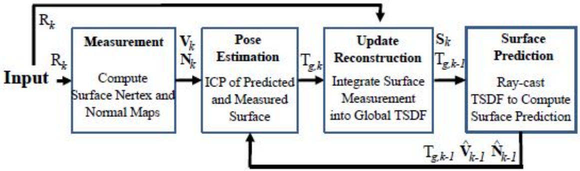


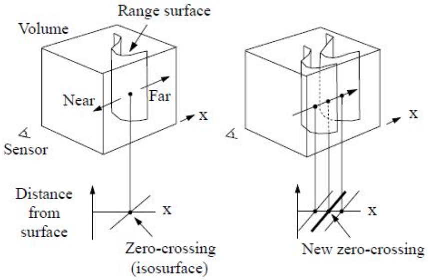
Figure: Overall system workflow of KinectFusion.

Fusion based Dense SLAM
KinectFusion

- Dense map representation.
  - Truncated signed-distance function (TSDF). [18]
    - Global TSDF containing a fusion of frames $1, \ldots, k$

      $$\mathbf{S}_k(\mathbf{p}) \mapsto [\mathrm{F}_\mathrm{k}(\mathbf{p}), \mathrm{W}_\mathrm{k}(\mathbf{p})],$$

      where $\mathrm{F}_\mathrm{k}(\mathbf{p})$ is the truncated signed distance value, $\mathrm{W}_\mathrm{k}(\mathbf{p})$ is the weight.
    - A discretization of TSDF is stored in GPU.



Fusion based Dense SLAM

KinectFusion

- Dense map representation.
  - Truncated signed-distance function (TSDF).
    - TSDF created from data of k-th frame.
    - For a point $\mathbf{p}$ in global frame, and a raw depth map $\mathrm{R_k}$ with a known $\mathrm{T_{g,k}}$

$$\mathrm{F}_{\mathrm{R_k}}(\mathbf{p}) = \psi(\lambda^{-1}\|\mathbf{t}_{\mathrm{g,k}} - \mathbf{p}\|_2 - \mathrm{R_k}(\mathbf{x})),$$

$$\lambda = \|\mathrm{K}^{-1}\dot{\mathbf{x}}\|_2,$$

$$\mathbf{x} = \left\lfloor \pi(\mathrm{K}\mathrm{T}_{g,k}^{-1}\mathbf{p}) \right\rfloor,$$

$$\psi(\eta) = \begin{cases} \min(1, \frac{\eta}{\mu})\mathrm{sgn}(\eta) & \text{iff } \eta \geq -\mu \\ \textit{null} & \textit{otherwise} \end{cases}.$$

Fusion based Dense SLAM

KinectFusion

- Dense map representation.
  - Truncated signed-distance function (TSDF).
    - De-noise the global TSDF from multiple noisy TSDF measurements.
    - Update rules

$$\mathrm{F}_k(\mathbf{p}) = \frac{\mathrm{W}_{k-1}(\mathbf{p})\mathrm{F}_{k-1}(\mathbf{p}) + \mathrm{W}_{\mathrm{R}_k}(\mathbf{p})\mathrm{F}_{\mathrm{R}_k}(\mathbf{p})}{\mathrm{W}_{k-1}(\mathbf{p}) + \mathrm{W}_{\mathrm{R}_k}(\mathbf{p})}$$

$$\mathrm{W}_k(\mathbf{p}) = \mathrm{F}_{k-1}(\mathbf{p}) + \mathrm{F}_{\mathrm{R}_k}(\mathbf{p})$$

Fusion based Dense SLAM

KinectFusion

- Surface prediction.
  - Surface prediction from ray casting the TSDF. [19]
    - Each pixel's corresponding ray, $\mathrm{T}_{g,k}\mathrm{K}^{-1}\dot{\mathbf{u}}$.
    - March starting from minimum depth and stopping when a zero crossing is found.
    - $\mathrm{R}_{g,k}\hat{\mathrm{N}}_k = \hat{\mathrm{N}}_k^g(\mathbf{u}) = \nu[\nabla \mathrm{F}(\mathbf{p})],\ \nabla \mathrm{F}(\mathbf{p}) = \left[\frac{\partial \mathrm{F}}{\partial x}, \frac{\partial \mathrm{F}}{\partial y}, \frac{\partial \mathrm{F}}{\partial z}\right]^T$

Fusion based Dense SLAM

KinectFusion

- Sensor pose estimation.
  - Two assumptions:
    - Small motion from one frame to the next (due to high tracking frame-rate).
    - GPU enables a fully parallelized processing pipeline.
  - Align a live surface measurement $(\mathbf{V}_k, \mathbf{N}_k)$ against the model prediction from the previous frame $(\hat{\mathbf{V}}_k, \hat{\mathbf{N}}_k)$.
  - Projective data association [20] and point-plane metric [21].
  - Global energy to minimize,

$$\mathbf{E}(\mathrm{T}_{g,k}) = \sum_{\mathbf{u}\in\mathscr{U}} \left\| \left(\mathrm{T}_{g,k}\dot{\mathbf{V}}_k(\mathbf{u}) - \hat{\mathbf{V}}^g_{k-1}(\hat{\mathbf{u}})\right)^T \hat{\mathbf{N}}^g_{k-1}(\hat{\mathbf{u}}) \right\|_2 .$$

Fusion based Dense SLAM

KinectFusion

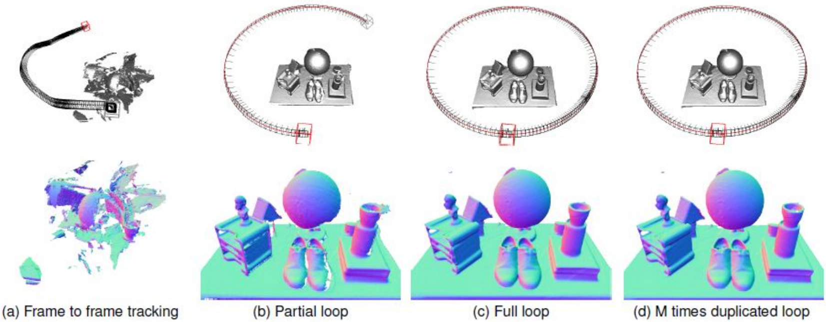


Figure: Circular motion experiment.

Fusion based Dense SLAM

ElasticFusion

- Preliminaries.
  - A pixel coordinate $\mathbf{u} \in \Omega \subset \mathbb{N}^2$.
  - Depth map D, $d : \Omega \to \mathbb{R}$.
  - Color image C, $\mathbf{c} : \Omega \to \mathbb{N}^3$.
  - 3D back-projection $\mathbf{p}(\mathbf{u}, D) = \mathrm{K}^{-1}\mathbf{u}d(\mathbf{u})$.
  - Perspective projection $\mathbf{u} = \pi(\mathrm{K}\mathbf{p})$.
  - Intensity image $I(\mathbf{u}, C) = \mathbf{c}(\mathbf{u})^T\mathbf{i}$, $\mathbf{i} = [0.114, 0.299, 0.587]^T$.
  - Global pose of camera

$$\mathbf{P}_t = \begin{bmatrix} \mathbf{R}_t & bft_t \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{S}E_3$$

Fusion based Dense SLAM

ElasticFusion

- Map representation.
  - An unordered list of surfels M.
  - Each surfel M^s:
    - position $\mathbf{p} \in \mathbb{R}^3$
    - normal $\mathbf{n} \in \mathbb{R}^3$
    - color $\mathbf{c} \in \mathbb{N}^3$
    - weight $\omega \in \mathbb{R}$
    - radius $r \in \mathbb{R}$ ($r = \frac{d\sqrt{2}}{f|\mathbf{n}_z|}$)
    - initialized timestamp t_0
    - last updated timestamp t

Fusion based Dense SLAM
ElasticFusion

- Pose estimation.

$$E_{track} = E_{icp} + \omega_{rgb} E_{rgb}$$

- Geometric term:

$$E_{icp} = \sum_k \left(\left(\mathbf{v}^k - \exp(\hat{\xi}) \mathrm{T} \mathbf{v}_t^k \right) \cdot \mathbf{n}^k \right)^2.$$

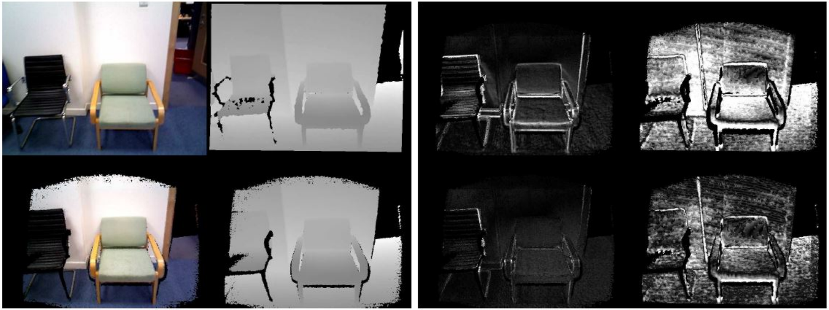
- Photometric term:

$$E_{rgb} = \sum_{\mathbf{u} \in \Omega} \left(I(\mathbf{u}, C_t^l) - I\left(\pi(\mathrm{K} \exp(\hat{\xi}) \mathrm{T} \mathbf{p}(\mathbf{u}, D_t^l)), \hat{C}_{t-1}^a \right) \right)^2,$$

where D_t^l and C_t^l are the current depth and color images, $\hat{D}_{t-1}^a$ and $\hat{C}_{t-1}^a$ are the predicted active model from the last frame.

Fusion based Dense SLAM

ElasticFusion



(i) (ii)

Fusion based Dense SLAM

ElasticFusion

- Local loop closure.
    - Divide M into two disjoint sets Θ (active set) and Ψ (inactive set) according to the timestamp M_t^s.
    - Align Θ and Ψ.
- Global loop closure. [22]
    - Randomized fern encoding.

Fusion based Dense SLAM

ElasticFusion

- Deformation graph.
- Each node G^n:
  - timestamp $G^n_{t_0}$
  - position G^n_g
  - set of neighboring nodes $\mathcal{N}(G^n)$
  - affine transformation G^n_{R} and G^n_{t}

Fusion based Dense SLAM

ElasticFusion

- Graph construction.
  - Sample from M s.t. $|G| \ll |M|$.
  - G is ordered over n on $G^n_{t_0}$ s.t. $G^n_{t_0} \geq G^{n-1}_{t_0}, \ldots, G^0_{t_0}$.
  - Define $\mathcal{N}(G^n) = G^{n\pm 1}, \ldots, G^{n\pm k/2}$.

ElasticFusion

- Deformation graph.
- Deformed position of a surfel.

$$\hat{M}^s_{\mathbf{p}} = \sum_{n \in I(M^s, G)} \omega^n(M^s) \left[G^n_{\mathrm{R}}(M^s_{\mathbf{p}} - G^n_g) + G^n_g + G^n_t \right]$$

$$\hat{M}^s_{\mathbf{n}} = \sum_{n \in I(M^s, G)} \omega^n(M^s) {G^n_{\mathrm{R}}}^{-1^T} M^s_{\mathbf{n}}$$

where $I(M^s, G)$ is a set of influencing nodes in graph which M^s identifies. (Algorithm 1)

Fusion based Dense SLAM

ElasticFusion

Algorithm 1: Deformation Graph Application

Input: $\mathcal{M}^s$ surfel to be deformed
$\mathcal{G}$ set of deformation nodes
α number of nodes to explore

Output: $\hat{\mathcal{M}}^s$ deformed surfel

do

```
// Find closest node in time
```

$c \leftarrow \arg\min_i \left\| \mathcal{M}^s_{t_0} - \mathcal{G}^i_{t_0} \right\|_1$

```
// Get set of temporally nearby nodes
```

$\mathcal{I} \leftarrow \emptyset$

for $i \leftarrow -\alpha/2$ **to** $\alpha/2$ **do**

$\quad \mathcal{I}^{i+\alpha/2} \leftarrow c + i$

sort_by_euclidean_distance($\mathcal{I}, \mathcal{G}, \mathcal{M}^s_{\mathbf{p}}$)

```
// Take closest k as influencing nodes
```

$\mathcal{I}(\mathcal{M}^s, \mathcal{G}) \leftarrow \mathcal{I}^{0 \rightarrow k-1}$

```
// Compute weights
```

$h \leftarrow 0$

$d_{max} \leftarrow \left\| \mathcal{M}^s_{\mathbf{p}} - \mathcal{G}^{\mathcal{I}^k}_{\mathbf{g}} \right\|_2$

for $n \in \mathcal{I}(\mathcal{M}^s, \mathcal{G})$ **do**

$\quad w^n(\mathcal{M}^s) \leftarrow (1 - \left\| \mathcal{M}^s_{\mathbf{p}} - \mathcal{G}^n_{\mathbf{g}} \right\|_2 / d_{max})^2$

$\quad h \leftarrow h + w^n(\mathcal{M}^s)$

```
// Apply transformations
```

$\hat{\mathcal{M}}^s_{\mathbf{p}} = \sum_{n \in \mathcal{I}(\mathcal{M}^s, \mathcal{G})} \frac{w^n(\mathcal{M}^s)}{h} \left[\mathcal{G}^n_{\mathbf{R}} (\mathcal{M}^s_{\mathbf{p}} - \mathcal{G}^n_{\mathbf{g}}) + \mathcal{G}^n_{\mathbf{g}} + \mathcal{G}^n_{\mathbf{t}} \right]$

$\hat{\mathcal{M}}^s_{\mathbf{n}} = \sum_{n \in \mathcal{I}(\mathcal{M}^s, \mathcal{G})} \frac{w^n(\mathcal{M}^s)}{h} \mathcal{G}^{n\;-1\top}_{\mathbf{R}} \mathcal{M}^s_{\mathbf{n}}$

Fusion based Dense SLAM

ElasticFusion

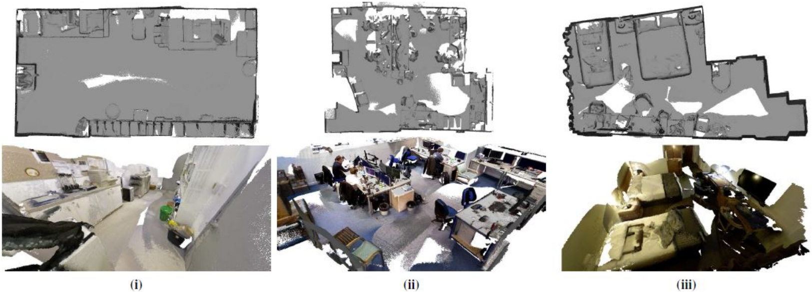


Figure: ElasticFusion experiment.

REFERENCE

- a survey of SLAM, TRO, 2017.
- T. Bailey and H. F. Durrant-Whyte. Simultaneous Localisation and Mapping (SLAM): Part II. Robotics and Autonomous Systems (RAS), 13(3):108117, 2006.
- H. F. Durrant-Whyte and T. Bailey. Simultaneous Localisation and Mapping (SLAM): Part I. IEEE Robotics and Automation Magazine, 13(2):99110, 2006.
- G. Dissanayake, S. Huang, Z. Wang, and R. Ranasinghe. A review of recent developments in Simultaneous Localization and Mapping. In International Conference on Industrial and Information Systems, pages 477482. IEEE, 2011.
- ORB-SLAM: A Versatile and Accurate Monocular SLAM System, TRO, 2015.
- LSD-SLAM: Large-Scale Direct Monocular SLAM, ECCV, 2014.
- SVO: Fast Semi-Direct Monocular Visual Odometry, ICRA, 2014.
- SVO: Fast Semi-Direct Monocular Visual Odometry, TRO, 2016.
- DTAM: Dense Tracking and Mapping in Real-Time, ICCV, 2011.
- 3-D Mapping With an RGB-D Camera, TRO, 2014.
- Dense Visual SLAM for RGB-D Cameras, IROS, 2013.

- Robust Odometry Estimation for RGB-D Cameras, ICRA, 2013.
- KinectFusion, ISMAR, 2011.
- ElasticFusion, RSS, 2015.
- ElasticFusion, IJRR, 2016.
- Bags of Binary Words for Fast Place Recognition in Image Sequences, TRO, 2012.
- FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance, IJRR, 2008.
- B. Curless and M. Levoy. A volumetric method for building complex models from range images. In ACM Transactions on Graphics (SIGGRAPH), 1996.
- S. Parker, P. Shirley, Y. Livnat, C. Hansen, and P. Sloan. Interactive ray tracing for isosurface rendering. In Proceedings of Visualization, 1998.
- G. Blais and M. D. Levine. Registering multiview range data to create 3D computer objects. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 17(8):820824, 1995.
- Y. Chen and G. Medioni. Object modeling by registration of multiple range images. Image and Vision Computing (IVC), 10(3):145155, 1992.

B. Glocker, J. Shotton, A. Criminisi, and S. Izadi. Real-Time RGB-D Camera Relocalization via Randomized Ferns for Keyframe Encoding. IEEE Transactions on Visualization and Computer Graphics, 21(5):571583, 2015.