An Evaluation of the RGB-D SLAM System

Endres, Felix & Hess, Jurgen & Engelhard, Nikolas & Sturm, Jurgen & Cremers, Daniel & Burgard, Wolfram. (2012)

Project 15 – 3D Reconstruction of Infants' Cranial Shape Using Mobile Devices
Seminar Presentation – Tara Tang

Project Summary

- *Clinical Problem:* pediatricians need a reliable method for accurately detecting infant cranial deformities such as DPB and craniosynostosis

- *Goal:* create a software pipeline to reconstruct an accurate 3D model of a baby's head from depth information

- *General approach:* take RGB and depth images of a baby from several viewpoints, and find registration transformation between viewpoints to combine all images into a unified 3D model

Paper Overview

Endres, Felix & Hess, Jurgen & Engelhard, Nikolas & Sturm, Jurgen & Cremers, Daniel & Burgard, Wolfram. (2012). **An Evaluation of the RGB-D SLAM System**. Proceedings - IEEE International Conference on Robotics and Automation. 1691-1696. 10.1109/ICRA.2012.6225199.

- Novel approach to SLAM problem using RGB-D information
- Evaluates algorithm with three feature descriptors: SIFT, SURF, and ORB
- Very relevant to us: outlines a basic approach that we have been trying to follow and adapt to our own project

Problem Background

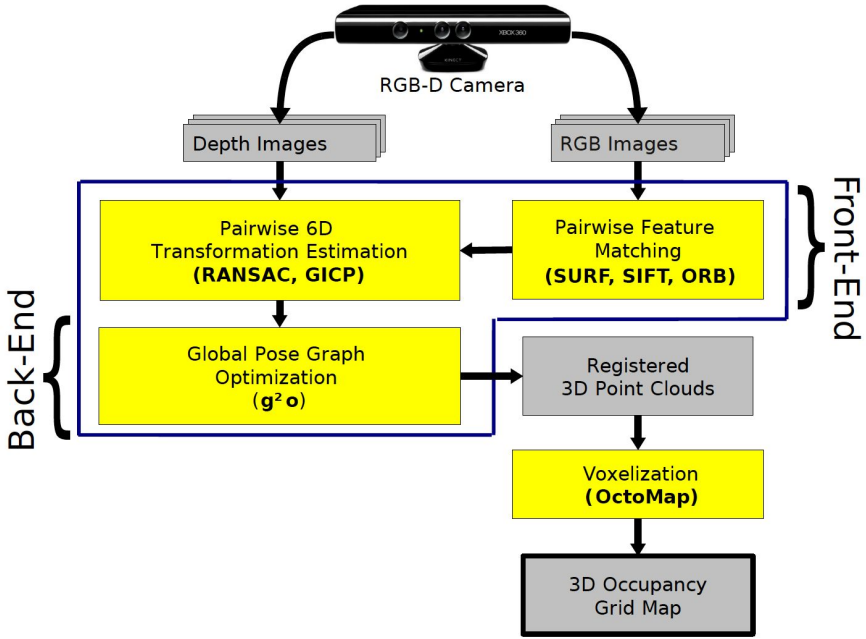
SLAM – **S**imultaneous **L**ocalization **A**nd **M**apping

- Chicken-and-egg problem
  - Camera pose to generate world model
  - World model that localizes camera pose
- Solution is SLAM: estimate both at the same time

RGB-D data

- New sensors like the Microsoft Kinect can capture both color images and corresponding dense depth maps
- Offer a novel approach to SLAM

Approach Overview – Schematic



1. RGB: feature detection and matching (**SIFT**, **SURF**, **ORB**)
2. Depth: project to 3D point clouds
3. Estimate transformation between features (**RANSAC**)
4. Pose graph to store transformations
5. Global pose graph optimization (**g^2o**)
6. Generate voxel occupancy map (**OctoMap**)

Endres et al. (2012)

Step 1: Feature Detection and Mapping

- Three feature detection algorithms implemented in OpenCV:
  - SIFT: **S**cale **I**nvariant **F**eature **T**ransform [2]
  - SURF: **S**peeded **U**p **R**obust **F**eatures [3]
  - ORB: **O**riented FAST and **R**otated **B**RIEF [4]

- Detect keypoints in two RGB images
- Match keypoints between images using keypoint descriptors

Lowe (1999)

Step 2: Project to 3D Point Clouds

- Feature locations in 2D image are projected to 3D space using depth measurement at keypoint center
- Requires knowledge of the camera's intrinsic parameters [5]
  - Describe how camera coordinates relate to image coordinates

$$\text{focal length} = (f_x, f_y)$$
$$\text{optical centers} = (c_x, c_y)$$
$$\text{image pixel} = (u, v)$$
$$\text{depth} = z$$
$$x = \frac{(u - c_x)z}{f_x}$$
$$y = \frac{(v - c_y)z}{f_y}$$

Step 3: Estimate Transformations

- Visual features prone to false positives, inconsistency with corresponding depth image, projection errors at image borders, etc.

How to cope with noisy data?

- RANSAC: **Ran**dom **Sa**mple **C**onsensus [6]
  - Randomly choose 3 matched pairs
  - Rigid transformation
  - Inliers: mutual distance < 3 cm
  - Refine rigid transformation with inliers
  - Iterate; choose transformation with most inliers
- Subset of 20 frames: 3 most recent + uniformly sampled previous images

Steps 4 and 5: Pose Graph + Optimization

- If frame can be matched to any previous frame, store in pose graph
  - Frames = nodes, transformations between frames = edges
  - Otherwise, store at same pose as previous frame with high uncertainty
- Create globally consistent camera trajectory with g^{2}o optimization [7]
  - Cost function: prune edges with high cost

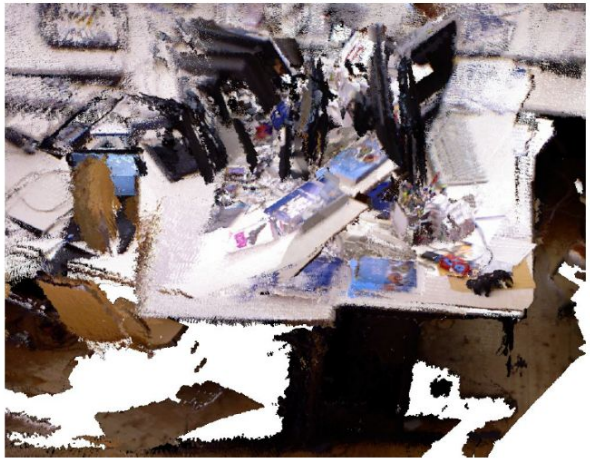
$$\mathbf{F}(\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij})^T \mathbf{\Omega}_{ij} \mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij})$$

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}} \mathbf{F}(\mathbf{x})$$

Step 6: Voxel Occupancy Grid

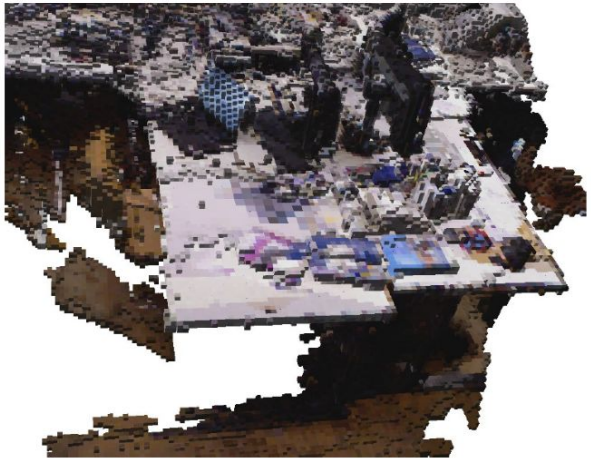
- All point clouds are registered to a common coordinate system
- Combination of all point clouds is large and highly redundant
- Represent world model as 3D occupancy grid map with OctoMap [8]
  - Octree-based mapping framework
  - Probabilistic occupancy estimation: copes with error and noise
  - Explicitly represents free space and unmapped areas

Example Results



(a) Result of frame-to-frame tracking (no loop closures)

(b) Result after graph optimization (loop closures)



(c) Volumetric 3D map after post-processing

Endres et al. (2012)

Evaluation of Results

- RGB-D benchmark with Kinect sequences and synchronized ground-truth
  - SIFTGPU and FLANN matching on FR1 (simplest dataset): average camera velocities = 9 deg/s to 42 deg/s, 0.06 m/s to 0.43 m/s
  - Average accuracy of 9.7 cm and 3.95º, 0.35 s processing time per image

Sequence Name	Length	Duration	Avg. Angular Velocity	Avg. Transl. Velocity	Frames	Total Runtime	g²o Runtime	Transl. RMSE	Rot. RMSE
FR1 360	5.82 m	28.69 s	41.60 deg/s	0.21 m/s	745	145 s	0.66 s	0.103 m	3.41°
FR1 desk2	10.16 m	24.86 s	29.31 deg/s	0.43 m/s	614	176 s	0.68 s	0.102 m	3.81°
FR1 desk	9.26 m	23.40 s	23.33 deg/s	0.41 m/s	575	199 s	1.31 s	0.049 m	2.43°
FR1 floor	12.57 m	49.87 s	15.07 deg/s	0.26 m/s	1214	488 s	3.93 s	0.055 m	2.35°
FR1 plant	14.80 m	41.53 s	27.89 deg/s	0.37 m/s	1112	424 s	1.28 s	0.142 m	6.34°
FR1 room	15.99 m	48.90 s	29.88 deg/s	0.33 m/s	1332	423 s	1.56 s	0.219 m	9.04°
FR1 rpy	1.66 m	27.67 s	50.15 deg/s	0.06 m/s	687	243 s	10.26 s	0.042 m	2.50°
FR1 teddy	15.71 m	50.82 s	21.32 deg/s	0.32 m/s	1395	556 s	1.72 s	0.138 m	4.75°
FR1 xyz	7.11 m	30.09 s	8.92 deg/s	0.24 m/s	788	365 s	40.09 s	0.021 m	0.90°

Endres et al. (2012)

Evaluation of Results

- SIFTGPU and SURF similar, ORB is less effective and failed twice

	Success	Transl. RMSE (Avg. ± Std. Dev.)	Rot. RMSE (Avg. ± Std. Dev.)
SIFTGPU	9/9	0.097 m ± 0.063 m	3.95° ± 2.47°
SURF	9/9	0.098 m ± 0.078 m	3.39° ± 1.55°
ORB	7/9	0.215 m ± 0.189 m	7.75° ± 5.55°

Endres et al. (2012)

Evaluation of Results

- ORB faster than SIFT and SURF by one order of magnitude
- FLANN faster than BF by a factor of 2
- Ratio of graph optimization vs. total runtime below 6%

Type	Count Avg. ± Std. Dev.	Runtime Detection + Extraction Avg. ± Std. Dev.
SURF	1733 ± 153	0.34 s + 0.34 s
ORB	1117 ± 558	0.018 s + 0.0086 s
SIFTGPU	1918 ± 599	0.19 s

Matcher	Runtime (Avg. ± Std. Dev)
FLANN	0.203 s ± 0.078 s
Brute Force	0.386 s ± 0.120 s

Endres et al. (2012)

Pros and Cons

Pros

- Very detailed and concise
- Written well – easy to understand and follow
- Open source code for evaluation and comparison

Cons

- How is the camera pose transformation calculated?
- Why not discard a frame that cannot be matched?
- Details for loop closure optimization
- How to determine “reliable” estimates that don’t require global optimization?

Relevance to Our Project

- Great introduction to the SLAM problem and how we might solve it
- Although implementation is quite different, good base for our project
- Ideas for reducing pose graph error and selecting frames for registration
- Comparison of feature detectors and matchers can help us optimize ours

References

[1] Endres, Felix & Hess, Jurgen & Engelhard, Nikolas & Sturm, Jurgen & Cremers, Daniel & Burgard, Wolfram. (2012). An evaluation of the RGB-D SLAM system. Proceedings - IEEE International Conference on Robotics and Automation. 1691-1696. 10.1109/ICRA.2012.6225199.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 1150-1157 vol.2, doi: 10.1109/ICCV.1999.790410.

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool (2008). Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, Volume 110, Issue 3. 346-359, ISSN 1077-3142, https://doi.org/10.1016/j.cviu.2007.09.014.

[4] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2564-2571, doi: 10.1109/ICCV.2011.6126544.

[5] Szeliski, Richard. (2011). *Computer Vision: Algorithms and Applications* (1st ed.). London: Springer.

[6] Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 6 (June 1981), 381–395. doi: https://doi.org/10.1145/358669.358692

[7] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige and W. Burgard, "G2o: A general framework for graph optimization," 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 3607-3613, doi: 10.1109/ICRA.2011.5979949.

[8] K.M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation, 2010.