1 Quick Explanation about SLAM's front and back-ends

SLAM algorithms are mostly separated into a front-end and a back-end [1]. Using this separation the front-end resembles the part of the program which interprets the sensor data by e.g. extracting landmarks and executing data association[1]. The back-end on the other hand describes the part of the algorithm which is responsible for estimating the robot's pose and the map (i.e. the landmark locations). Figure 1 sketches this relationship for a typical SLAM system.

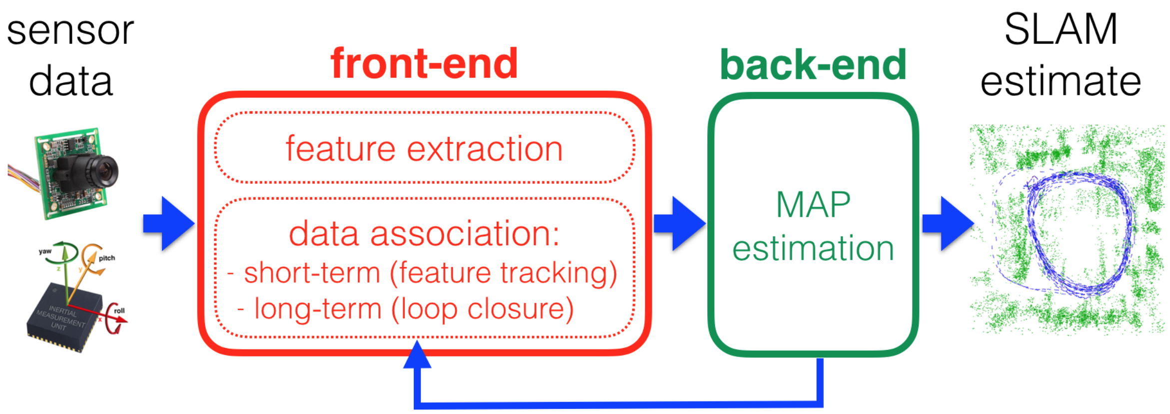

Figure 1: Front-end and back-end in a typical SLAM system. The back-end can provide feedback to the front-end for loop closure detection. [1]

[1]Matching landmark between frames.

Table 1: Monocular graph-based SLAM approaches

Name	Characteristics	
CD-SLAM	*Front-end:*	Histogram of Oriented Cameras (HoC) descriptor [2]
	Back-end:	Keyframe-based BA, graph optimization
	Loop closing:	Yes, using FAB-MAP
	Code:	-
	Refs(year):	[3](2011)
	Notes:	Focus on highly dynamic environments, Keep map proportional to explored space not time, Usage of HoC descriptor in feature based front end
C-KLAM	*Front-end:*	SIFT features
	Back-end:	Custom to C-KLAM, based on BA
	Loop closing:	Possible, but was not achieved in test
	Code:	-
	Refs(year):	[4](2014)
	Notes:	Focus on making use of data between keyframes, Incorporates IMU data in graph optimization
CNN SLAM	*Front-end:*	Dense approach based on LSD SLAM
	Back-end:	$Sim(3)$ optimization and BA
	Loop closing:	Not mentioned in paper
	Code:	-
	Refs(year):	[5](2017)
	Notes:	Uses trained CNN to predict depth maps, Can be used to correct scale drift, Relies on use of GPU, Implements semantic labeling (i.e. distinguish walls and floor)
COP SLAM	*Front-end:*	Any front-end that creates pose-graphs
	Back-end:	Optimizing pose-chains using trajectory bending[6]
	Loop closing:	Is a back-end $\rightarrow$ no detection, only optimization
	Code:	-
	Refs(year):	[7](2013), [8](2015)
	Notes:	Optimizes sparse pose-graph (pose-chain), 50 to 200 times faster than G^2O, Only optimizes the robot pose, not the map, Has an extension which can account for scale-drift

Dolphin-SLAM	*Front-end:*	Sensor dependant, Sonar: HU moments, Image: SURF
	Back-end:	RatSLAM back-end (not really graph-based)
	Loop closing:	Yes, using FAB-MAP
	Code:	`https://github.com/dolphin-slam/dolphin_slam`
	Refs(year):	[9](2015), [10](2016)
	Notes:	Focus on SLAM in underwater scenario, Multiple sensors: sonar, camera, IMU and DVL, Based on RatSLAM
DPPTAM	*Front-end:*	Piecewise planar dense
	Back-end:	Semi-dense map with estimation of planar surfaces, map optimization not mentioned
	Loop closing:	Not mentioned in paper
	Code:	`https://github.com/alejocb/dpptam`
	Refs(year):	[11](2015)
	Notes:	Reconstruction of dense maps using only CPU, Reduced cost due to planar surface estimation via superpixels [12] with the assumption that low color-gradient areas are mostly planar
DSO	*Front-end:*	Sparse and direct
	Back-end:	None (odometry only)
	Loop closing:	No
	Code:	`https://github.com/JakobEngel/dso`
	Refs(year):	[11](2015)
	Notes:	Optimizes camera intrinsics and extrinsics, Works well in low textured areas, Distributes sampled pixels such that, when available, high gradients are used, otherwise takes weak gradients
DTAM	*Front-end:*	Dense method
	Back-end:	Creates dense map, no graph optimization
	Loop closing:	No
	Code:	`https://github.com/anuranbaka/OpenDTAM`
	Refs(year):	[13](2011)
	Notes:	Relies on GPU for computation, Creates feature-rich, textured, dense map, Robust against quick movement and camera defocus

DT SLAM	*Front-end:*	FAST, separated 2D and 3D feature matching
	Back-end:	BA on pose and features of all sub maps
	Loop closing:	Yes
	Code:	`https://github.com/plumonito/dtslam`
	Refs(year):	[14](2014)
	Notes:	Holds off 3D feature triangulation until enough parallax is observed (Deferred triangulation), Can incorporate purely rotational movement frames, Creates sub-maps which can be merged later to avoid scale inconsistencies
FAB-MAP	*Front-end:*	Appearance-based bag of words approach with SURF
	Back-end:	Place recognition in appearance-space, no metric map
	Loop closing:	Yes
	Code:	`https://github.com/arrenglover/openfabmap`
	Refs(year):	[15](2008), [16](2009), [17](2012)
	Notes:	Needs training on an environment similar to the one it is going to be used in, Place recognition via visual bag of words database, No metric map creation, mostly used as a tool for loop closure detection in other approaches like LSD SLAM
LSD SLAM	*Front-end:*	Semi-dense
	Back-end:	Creates semi-dense map, pose-graph optimization (g2o)
	Loop closing:	Yes, small via $sim(3)$ and large via FAB-MAP
	Code:	`https://github.com/tum-vision/lsd_slam`
	Refs(year):	[18](2013), [19](2014)
	Notes:	Semi-dense, keyframe-based, runs on CPU, Makes use of Lie-algebra for tracking and optimization, Scale-drift aware
NID SLAM	*Front-end:*	Semi-dense, NID metric
	Back-end:	Creates semi-dense map, pose-graph optimization (g2o)
	Loop closing:	Yes, using FAB-MAP
	Code:	-
	Refs(year):	[20](2017)
	Notes:	Focus on lighting, weather and structural changes, Relies on GPU for computation, Scale-drift aware (based on LSD SLAM)

ORB SLAM	*Front-end:*	Indirect using ORB feature descriptor
	Back-end:	BA on sub-graphs, uses local maps
	Loop closing:	Yes, using a place recognition database
	Code:	`https://github.com/raulmur/ORB_SLAM`
	Refs(year):	[21](2015)
	Notes:	Uses three parallel threads for tracking, local map creation and loop closing, Uses a bag of words approach for place recognition, Focus on real-time operation, runs on CPU, Focus on long-term localization, not detailed maps
ORB SLAM2	*Front-end:*	Indirect using ORB feature descriptor
	Back-end:	BA on sub-graphs, uses local maps
	Loop closing:	Yes, using a place recognition database
	Code:	`https://github.com/raulmur/ORB_SLAM2`
	Refs(year):	[22](2016)
	Notes:	Extension of ORB SLAM for RGB-D and stereo camera setups
Visual Inertial ORB SLAM	*Front-end:*	Indirect using ORB feature descriptor
	Back-end:	BA on sub-graphs, uses local maps
	Loop closing:	Yes, using a place recognition database
	Code:	-
	Refs(year):	[23](2017)
	Notes:	Modular extension of ORB SLAM for incorporating IMU readings, Takes gyroscope and accelerometer bias into account, Derives scale from IMU measurements
PTAM	*Front-end:*	Indirect, using FAST corner detector
	Back-end:	Local and global BA
	Loop closing:	No
	Code:	`https://github.com/Oxford-PTAM/PTAM-GPL`
	Refs(year):	[24](2007)
	Notes:	First to parallelize tracking and mapping, Uses keyframes to create map, Optimizes map in background when exploring already known areas

Rat-SLAM	*Front-end:*	Not specified
	Back-end:	Neural network based on the hippocampus of rodents
	Loop closing:	Yes, via local view cells
	Code:	`https://github.com/davidmball/ratslam`
	Refs(year):	[25](2004), [26](2005), [27](2006), [28](2008), [29](2013)
	Notes:	Back-end that can make use of different input data both visual and internal (IMU), Composed of pose cells, local view cells, and experience map
RD SLAM	*Front-end:*	Indirect, using SIFT
	Back-end:	BA based on PTAM
	Loop closing:	No
	Code:	`http://www.zjucvg.net/rdslam/rdslam.html`
	Refs(year):	[30](2013)
	Notes:	Based on PTAM, Focus on dynamic environments with changes in structure and illumination, According to author still fails frequently, Relies on GPU due to SIFT features
REBVO	*Front-end:*	Between direct and indirect, using edges as features
	Back-end:	None (odometry only)
	Loop closing:	No
	Code:	`https://github.com/JuanTarrio/rebvo`
	Refs(year):	[31](2016)
	Notes:	Tracks edges instead of features or pixels, Focus on running on embedded devices, Has extension for IMU integration, Depth estimated via EKF
REMODE	*Front-end:*	Direct, dense method, bayesian estimation for depth
	Back-end:	Creates dense map, no graph optimization
	Loop closing:	No
	Code:	`https://github.com/uzh-rpg/rpg_open_remode`
	Refs(year):	[32](2014)
	Notes:	Depth estimation via a bayesian scheme, Can smooth camera measurement noise, Relies on GPU for processing

RFM SLAM	*Front-end:*	Indirect, but not specified which features
	Back-end:	Splits orientation and position estimation for 2D case
	Loop closing:	Yes
	Code:	-
	Refs(year):	[33](2016)
	Notes:	Separates orientation and position estimation, 2D case can be extended to 3D, Focus on computationally cheaper pose optimization
RK SLAM	*Front-end:*	Indirect, FAST, uses homographies for tracking
	Back-end:	Local and global optimization using BA
	Loop closing:	Yes
	Code:	`http://www.zjucvg.net/rkslam/rkslam.html`
	Refs(year):	[34](2016)
	Notes:	Focus on fast motion and rotation, Extracts and matches 3D planes in keyframes, Has extension to incorporate IMU data
Seq SLAM	*Front-end:*	Looks for minimum in difference of image sequences
	Back-end:	No map creation or localization, only place recognition
	Loop closing:	Yes
	Code:	`https://github.com/subokita/OpenSeqSLAM`
	Refs(year):	[35](2012)
	Notes:	Focus on extreme changes in environment, Like FAB-MAP no real SLAM approach which localizes a robot and builds a map, rather used for loop closure
SVO	*Front-end:*	Semi-direct, dense pixel batches and FAST
	Back-end:	BA for pose, keeps a small map of fixed size
	Loop closing:	No
	Code:	`https://github.com/uzh-rpg/rpg_svo`
	Refs(year):	[36](2014), [37](2017)
	Notes:	Focus on runtime, runs on embedded devices, Comes with two settings: *fast* and *accurate*, Fixed number of keyframes in map, Needs high FPS $\sim$ 60 FPS

References

[1] Cesar Cadena et al. "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age". In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332. ISSN: 15523098. DOI: 10.1109/TRO.2016.2624754. arXiv: arXiv:1606.05830v2.

[2] Katrin Pirker, Matthias Rüther, and Horst Bischof. "Histogram of Oriented Cameras - A New Descriptor for Visual SLAM in Dynamic Environments". In: *British Machine Vision Conference* (2010), pp. 76.1–76.12. DOI: 10.5244/C.24.76.

[3] Katrin Pirker, M Ruther, and Horst Bischof. "CD SLAM - continuous localization and mapping in a dynamic world". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* (2011), pp. 3990–3997.

[4] Esha D. Nerurkar, Kejian J. Wu, and Stergios I. Roumeliotis. "C-KLAM: Constrained keyframe-based localization and mapping". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2014), pp. 3638–3643. ISSN: 10504729. DOI: 10.1109/ICRA.2014.6907385.

[5] Keisuke Tateno et al. "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction". In: (2017). arXiv: 1704.03489. URL: http://arxiv.org/abs/1704.03489.

[6] Gijs Dubbelman, Isaac Esteban, and Klamer Schutte. "Efficient trajectory bending with applications to loop closure". In: *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings* (2010), pp. 4836–4842. ISSN: 2153-0858. DOI: 10.1109/IROS.2010.5652656.

[7] Gijs Dubbelman and Brett Browning. "Closed-form Online Pose-chain SLAM". In: *IEEE International Conference on Robotics and Automation*. 2013. URL: http://gijsdubbelman.com/joomla/media/pdf/icra2013.pdf.

[8] Gijs Dubbelman and Brett Browning. "COP-SLAM: Closed-Form Online Pose-Chain Optimization for Visual SLAM". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1194–1213. ISSN: 15523098. DOI: 10.1109/TRO.2015.2473455.

[9] Luan Silveira et al. "An open-source bio-inspired solution to underwater SLAM". In: *IFAC-PapersOnLine* 28.2 (2015), pp. 212–217. ISSN: 24058963. DOI: 10.1016/j.ifacol.2015.06.035.

[10] Guilherme B. Zaffari et al. "Exploring the DolphinSLAM's parameters". In: (2016). DOI: 10.1109/OCEANSAP.2016.7485531.

[11] Alejo Concha and Javier Civera. "DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence". In: *IEEE International Conference on Intelligent Robots and Systems* 2015-Decem (2015), pp. 5686–5693. ISSN: 21530866. DOI: 10.1109/IROS.2015.7354184.

[12] Pedro F Felzenszwalb and Daniel P Huttenlocher. "Efficient Graph-Based Image Segmentation". In: (2015), pp. 1–26. ISSN: 1573-1405. DOI: 10.1023/B:VISI.0000022288.19776.77. URL: papers3://publication/uuid/D1250C05-2FC7-4954-A734-E33EBBEECB95.

[13] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. "DTAM: Dense tracking and mapping in real-time". In: *Proceedings of the IEEE International Conference on Computer Vision* (2011), pp. 2320–2327. ISSN: 1550-5499. DOI: 10.1109/ICCV.2011.6126513.

[14] Herrera C. Daniel et al. "DT-SLAM: Deferred triangulation for robust SLAM". In: *Proceedings - 2014 International Conference on 3D Vision, 3DV 2014* (2014), pp. 609–616. DOI: 10.1109/3DV.2014.49.

[15] M. Cummins and P. Newman. "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance". In: *The International Journal of Robotics Research* 27.6 (2008), pp. 647–665. ISSN: 0278-3649. DOI: 10.1177/0278364908090961. URL: http://ijr.sagepub.com/cgi/doi/10.1177/0278364908090961.

[16] Mark Cummins and Paul Newman. "Highly Scalable Appearance-Only SLAM - FAB-MAP 2.0". In: *Rss* (2009), pp. 1–8. ISSN: 10504729. DOI: 10.1109/ROBOT.2008.4543473.

[17] Arren Glover et al. "OpenFABMAP: An Open Source Toolbox for Appearance-based Loop Closure Detection". In: *International Conference on Robotics and Automation*. May. Saint Paul, MN, 2012. DOI: 10.1109/ICRA.2012.6224843. URL: https://eprints.qut.edu.au/50317/1/glover{_}ICRA2012{_}final.pdf.

[18] Jakob Engel, Jurgen Sturm, and Daniel Cremers. "Semi-dense visual odometry for a monocular camera". In: *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 1449–1456. ISSN: 1550-5499. DOI: 10.1109/ICCV.2013.183.

[19] Jakob Engel, Thomas Sch, and Daniel Cremers. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: (2014), pp. 834–849. ISSN: 16113349. DOI: 10.1007/978-3-319-10605-2_54.

[20] G Pascoe et al. "NID-SLAM: Robust Monocular SLAM using Normalised Information Distance". In: *Computer Vision and Pattern Recognition* (2017). URL: http://www.robots.ox.ac.uk/{~}mobile/Papers/2017CVPR{_}pascoe.pdf.

[21] Raul Mur-Artal, J. M M Montiel, and Juan D. Tardos. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163. ISSN: 15523098. DOI: 10.1109/TRO.2015.2463671. arXiv: 1502.00956.

[22] Raul Mur-Artal and Juan D. Tardos. "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras". In: (2016). ISSN: 15523098. DOI: 10.1109/TRO.2012.2197158. arXiv: 1610.06475. URL: http://arxiv.org/abs/1610.06475.

[23] Raul Mur-Artal and Juan D. Tardos. “Visual-Inertial Monocular SLAM with Map Reuse”. 2017. URL: http://arxiv.org/abs/1610.05949{\%}0Ahttp://dx.doi.org/10.1109/LRA.2017.2653359.

[24] Georg Klein and David Murray. “Parallel tracking and mapping for small AR workspaces”. In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR* (2007). ISSN: 00472778. DOI: 10.1109/ISMAR.2007.4538852. arXiv: arXiv:1407.5736v1.

[25] Michael Milford, Gordon Wyeth, and David Prasser. “RatSLAM: a hippocampal model for simultaneous localization and mapping”. In: *Robotics and Automation, ...* May 2004 (2004), pp. 403–408. ISSN: 1050-4729. DOI: 10.1109/ROBOT.2004.1307183. URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=1307183.

[26] Michael J. Milford, David Prasser, and Gordon F. Wyeth. “Experience Mapping : Producing Spatially Continuous Environment Representations using RatSLAM”. In: *Proceedings of Australasian Conference on Robotics and Automation 2005* (2005), pp. 1–10. URL: http://eprints.qut.edu.au/32840/.

[27] Michael Milford, Gordon Wyeth, and David Prasser. “RatSLAM on the edge: Revealing a coherent representation from an overloaded rat brain”. In: *IEEE International Conference on Intelligent Robots and Systems* (2006), pp. 4060–4065. DOI: 10.1109/IROS.2006.281869.

[28] Michael John Milford. *Robot Navigation from Nature*. Vol. 41. 2008, p. 196. ISBN: 9783540775195. DOI: 10.1007/978-3-540-77520-1.

[29] David Ball et al. “OpenRatSLAM: An open source brain-based SLAM system”. In: *Autonomous Robots* 34.3 (2013), pp. 149–176. ISSN: 09295593. DOI: 10.1007/s10514-012-9317-9.

[30] Wei Tan. “Robust Monocular SLAM in Dynamic Environments”. In: *IEEE International Symposium on Mixed and Augmented Reality 2013*. 2013. DOI: 978-1-4799-2869-9/13.

[31] Juan Jose Tarrio and Sol Pedre. “Realtime edge-based visual odometry for a monocular camera”. In: *Proceedings of the IEEE International Conference on Computer Vision* 11-18-Dece (2016), pp. 702–710. ISSN: 15505499. DOI: 10.1109/ICCV.2015.87.

[32] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. “REMODE: Probabilistic, monocular dense reconstruction in real time”. In: *Proceedings - IEEE International Conference on Robotics and Automation* (2014), pp. 2609–2616. ISSN: 10504729. DOI: 10.1109/ICRA.2014.6907233.

[33] Saurav Agarwal, Vikram Shree, and Suman Chakravorty. “RFM-SLAM: Exploiting Relative Feature Measurements to Separate Orientation and Position Estimation in SLAM”. In: (2016). arXiv: 1609.05235. URL: http://arxiv.org/abs/1609.05235.

[34] Haomin Liu, Guofeng Zhang, and Hujun Bao. "Robust Keyframe-based Monocular SLAM for Augmented Reality". In: *IEEE International Symposium on Mixed and Augmented Reality Robust*. 2016. ISBN: 9781509036417. DOI: 10.1109/ISMAR.2016.24.

[35] Michael J. Milford and Gordon F. Wyeth. "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2012), pp. 1643–1649. ISSN: 10504729. DOI: 10.1109/ICRA.2012.6224623.

[36] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. "SVO: Fast semi-direct monocular visual odometry". In: *Proceedings - IEEE International Conference on Robotics and Automation* May (2014), pp. 15–22. ISSN: 10504729. DOI: 10.1109/ICRA.2014.6906584. arXiv: arXiv:1606.05830v2.

[37] Christian Forster et al. "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems". In: *IEEE Transactions on Robotics* 33.2 (2017), pp. 249–265. ISSN: 15523098. DOI: 10.1109/TRO.2016.2623335. arXiv: 1204.3968.