WPI

Autonomous Navigation Using ORB SLAM for Indoor Environments

RBE 595 Sensor Fusion and Perception
Project Presentation

Abizer Patanwala
Joy Mehta
Kunal Nandanwar

Abstract

Visual simultaneous localization and mapping (vSLAM) is the technique of concurrently mapping the environment and determining the location and orientation of a camera in relation to its surroundings. The technique merely makes use of the camera's visual inputs. Augmented reality, robotics, and autonomous driving are some uses for vSLAM.

We aim to demonstrate how to handle visual data from a monocular camera to create a map of an indoor area and estimate the trajectory of the camera. We employ ORB-SLAM, the feature-based vSLAM algorithm

Introduction

Autonomous interior navigation of robots has emerged as one of the major challenges for service-oriented robots. The mapping, localization, path planning, and dynamic obstacle avoidance are components of the visual SLAM problem.

Two ways to solve the problem:

1. Using filter based methods such as MonoSlam which uses EKF filter
2. Using optimization based methods: ORB-SLAM, PTAM, DTAM, LSD-SLAM etc.

We choose ORB-SLAM because it is one of the best Visual SLAM algorithm which outperforms many visual inertial algorithms and high Performance of ORB-SLAM in both outdoor and indoor environments, makes it a suitable choice to deploy it in navigation

Literature Review

Place Recognition:

- Williams et al. study evaluated a number of methods for location recognition and came to the conclusion that strategies focused on appearances, such as image-to-image matching, scale better in big contexts than map-to-map or image-to-map approaches.

- For the first time, DBoW2 combined the highly effective FAST feature detector with bags of binary words derived from BRIEF descriptors.

- Comparing this to the SURF and SIFT features that were previously employed in bags of words techniques, the time required for feature extraction was lowered by a factor of more than one order of magnitude.

Literature Review

Map Initialization:

- Because depth cannot be determined from a single image, monocular SLAM requires a process to produce an initial map.
- Tracking a well-known structure first can help find a solution to the issue.
- Initialization techniques based on two views either assume local planar scenes and determine the relative camera pose from a homography or compute an essential matrix that models planar and general scenes using the five-point algorithm.
- If all points in a planar image are closer to one of the camera centres, both reconstruction approaches suffer from a two-fold ambiguity solution and are not adequately controlled under low parallax.
- On the other hand, a unique basic matrix can be generated using the eightpoint approach if a nonplanar scene is perceived with parallax, allowing the relative camera pose to be accurately determined.

Literature Review

Monocular SLAM :

- Over the past decades, several prominent monocular SLAM techniques have been proposed such as MonoSLAM, PTAM, CD-SLAM, ORB-SLAM, ORB-SLAM2, Edge-SLAM, DTAM, LSD-SLAM and DSO.
- Among these only MonoSLAM and Monocular FastSLAM are filter-based techniques. Filtering was initially used to solve monocular SLAM.
- Keyframe-based approaches are more accurate than filtering for the same computing cost, according to research by Strasdat et al.
- MonoSLAM was one of the first algorithms to implement visual SLAM. Majority of the state of the art SLAM techniques are based on visual SLAM which employs optimization techniques rather than filter based.
- PTAM was a pioneering algorithm in the sense that it was the first to separate tracking and mapping threads and apply keyframes to mapping threads

Literature Review

Monocular SLAM :

- DTAM was one of the first methods to perform direct tracking and mapping. LSD-SLAM and DSO also perform direct tracking and mapping.
- With a front-end based on optical flow implemented on a GPU, followed by FAST feature matching and motion-only BA, and a back-end based on sliding-window BA, Strasdat et al. presented a large-scale monocular SLAM system.
- It also uses the survival of the fittest approach for map points and keyframes. This policy improves tracking robustness and enhances lifelong operation because redundant keyframes are dropped.
- Pirker et al. 's proposal for CD-SLAM is a fairly full system with efforts to work in dynamic situations, loop closing, relocalization, and large-scale operation.
- Song et al.'s visual odometry makes use of a temporal sliding window BA back-end and ORB characteristics for tracking.
- ORB-SLAM is an optimization-based method which uses ORB features for mapping, tracking, loop closing and relocalization. It's an extension of PTAM. It performs real-time loop closure based on pose graph optimization.

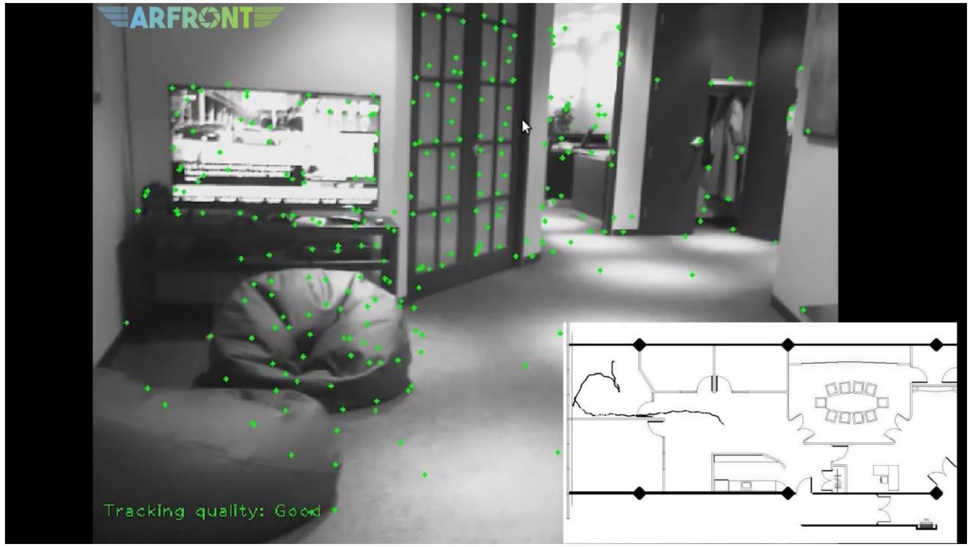
Problem Description

Indoor static environment: home, office, restaurant, warehouse or inside factory.

Aim: To localize a mobile robot, create a map of the surroundings to be used further for motion planning tasks.

Challenge: Lack of textures in the indoor environment, motion planning using a sparse map.

We choose to implement ORB-SLAM because of its comparatively high accuracy as compared to many state-of-the-art algorithms.



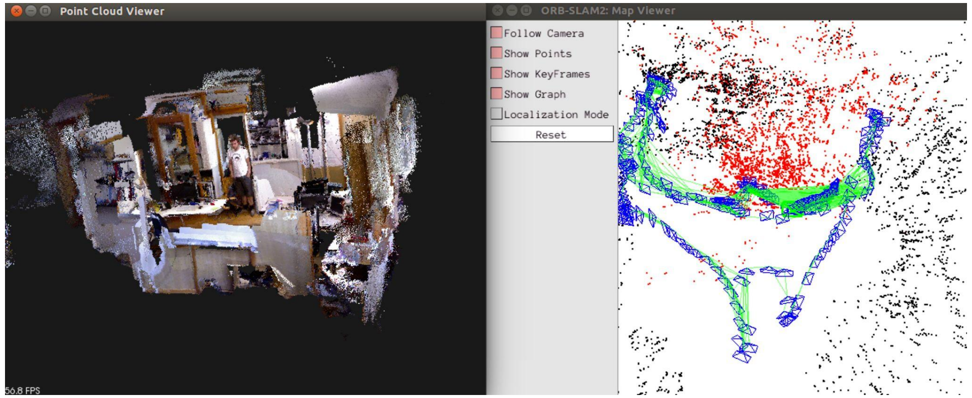


Image Courtesy: https://www.youtube.com/watch?v=XWZfgdVOzAE

Image Courtesy: https://github.com/necusjz/IndoorMapping

System Overview

Pipeline of ORB-SLAM showing tracking, mapping and loop closure steps.

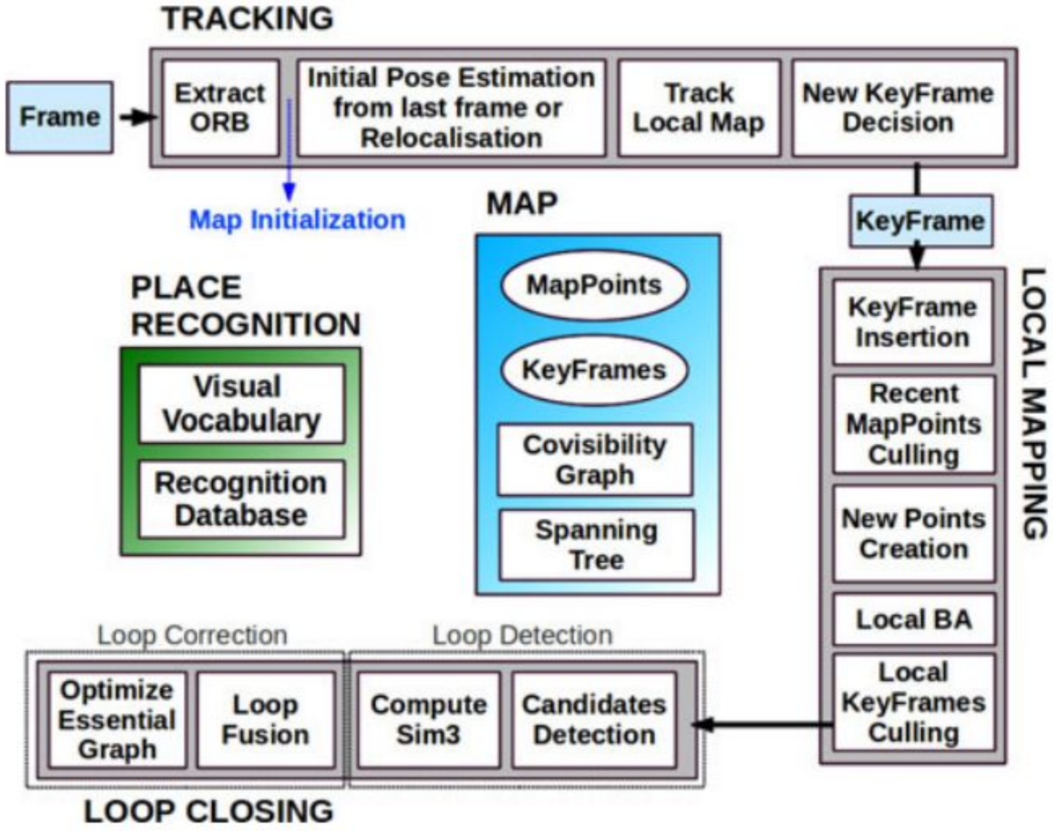


Image Courtesy: R. Mur-Artal, J. M. M. Montiel and J. D. Tardos, ”ORB-SLAM: A ´ Versatile and Accurate Monocular SLAM System,”
in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163,
Oct. 2015, doi: 10.1109/TRO.2015.2463671.

System Overview

Feature Choice

- We selected ORB, which are multi-scale FAST corners that are orientated and have a 256-bit descriptor attached.
- They offer strong perspective invariance and are very quick to compute and match. This makes it possible to match them across a wide range of baselines, improving BA accuracy.

Three Threads: Tracking, Local Mapping and Loop Closing

- Our system uses three concurrent threads for tracking, local mapping, and loop closure. For every frame, the tracking is in charge of localizing the camera and choosing when to add a new keyframe.

Map points and Key Frames

- A typical ORB description Di is the related ORB descriptor in the keyframes when the point is detected, and it has the smallest hamming distance compared to the other associated descriptors.
- If a distortion model is given, all ORB features taken from the frame, whether or whether they are linked to a map point, have coordinates that are undistorted.

System Overview

Bags of Words Place Recognition

- To conduct loop identification and relocalization, the system incorporates an inbuilt bags of words place recognition module. The descriptor space sometimes referred to as the visual language, may be discretized into visual words. The ORB descriptors that are taken from a sizable collection of photos are used to generate the vocabulary offline.

Map Initialization

- To triangulate a starting set of map points, the map initialization computes the relative posture between two frames. This technique should choose a decent two-view configuration, or a configuration with noticeable parallax, without requiring human interaction and should be independent of the scene (planar or generic).

Loop Closing

- The loop closing thread uses Ki, the final keyframe that the local mapping processed, to search for and end loops.

Feature Choice

- One of the key concepts in our system's design is the use of the same characteristics for location identification, frame-rate relocalization, and loop detection that are utilized for mapping and tracking.
- By doing so, we may effectively use our method without having to extrapolate the depth of the recognition features from nearby SLAM data.
- We selected ORB, which are multi-scale FAST corners that are orientated and have a 256-bit descriptor attached.
- They offer strong perspective invariance and are very quick to compute and match. This makes it possible to match them across a wide range of baselines, improving BA accuracy.

Three Threads: Tracking, Local Mapping and Loop Closing

- Our system uses three concurrent threads for tracking, local mapping, and loop closure. For every frame, the tracking is in charge of localizing the camera and choosing when to add a new keyframe.
- The location recognition module is used to carry out global relocalization whenever the tracking is lost.
- Following a reprojection search for matches with the nearby map locations, the camera posture is once more optimized. The tracking thread ultimately selects whether to insert a new keyframe.
- New keyframes are processed by the local mapping, and local BA is used to generate the best reconstruction possible around the camera posture. In order to triangulate new points, related keyframes in the covisibility graph are searched for new correspondences for mismatched ORB in the new keyframe.

Three Threads: Tracking, Local Mapping and Loop Closing

- Following creation, a strict point culling strategy is used to ensure that only high-quality points are kept, depending on the data acquired during tracking. Duplicate keyframes must be removed using local mapping.
- Every new keyframe triggers a search for loops by the loop closure. When a loop is found, a similarity transformation that describes the drift collected in the loop is computed.
- The duplicated points are then fused once both sides of the loop have been aligned.
- To establish global consistency, a pose graph optimization over similarity constraints is then carried out.

Map Points & Key Frames

- Each map point pi stores:
  - Its 3D coordinate system location is Xw,i.
  - The direction of observation ni is the average unit vector of all of its directions of observation (the rays that join the point with the optical centre of the keyframes that observe it).
  - A typical ORB description Di is the related ORB descriptor in the keyframes when the point is detected, and it has the smallest hamming distance compared to the other associated descriptors.
  - The scale invariance limits of the ORB features, specify the maximum dmax and lowest ddmin distances at which the point may be seen.

Map Points & Key Frames

- Each keyframe Ki store:
  - The rigid body transformation known as the camera posture Tiw transfers points from the outside world to the camera coordinate system.
  - The internal components of the camera, such as the focal length and main point.
  - If a distortion model is given, all ORB features taken from the frame, whether or whether they are linked to a map point, have coordinates that are undistorted.
- The creation of map points and keyframes follows a liberal policy, while the detection of superfluous keyframes and incorrectly matched or untrackable map points is handled by a more stringent culling method.
- In addition, despite having fewer points than PTAM, our maps have a elatively low number of outliers.

Covisibility Graph and Essential Graph

- The creation of map points and keyframes follows a liberal policy, while the detection of superfluo
- Our system's covisibility information, which is shown as an undirected weighted graph, is quite helpful for a number of jobs. Each node is a keyframe, and if two keyframes share observations of the same map points (at least 15), then there is an edge between them. The weight θ of the edge is determined by the number of shared map points.
- We carry out a pose graph optimization that disperses the loop-closing mistake over the network in order to repair a loop policy.

Bags of Words Place Recognition

- To conduct loop identification and relocalization, the system incorporates an inbuilt bags of words place recognition module. The descriptor space sometimes referred to as the visual language, may be discretized into visual words.
- The ORB descriptors that are taken from a sizable collection of photos are used to generate the vocabulary offline.
- As demonstrated in our prior work, provided the pictures are sufficiently generic, the same language may be utilized for multiple contexts and yet produce acceptable results.
- The method gradually creates a database with an inverted index, which records each visual word in the lexicon in which keyframes it has been observed. This allows for extremely efficient database searching. When a keyframe is eliminated through the culling process, the database is also updated.

Automatic Map Initialization

- To triangulate a starting set of map points, the map initialization computes the relative posture between two frames.
- This technique should choose a decent two-view configuration, or a configuration with noticeable parallax, without requiring human interaction and should be independent of the scene (planar or generic).
- By spotting low-parallax scenarios and the well-known twofold planar ambiguity, our technique avoids initializing a damaged map and only initializes when it is guaranteed that the two-view setup is secure.
- Our algorithm's stages are as follows:
  - Find initial correspondences
  - Parallel computation of the two models
  - Model Selection
  - Motion and Structure from Motion recovery 5) Bundle adjustment

Tracking

- ORB Extraction
  - At 8 scale levels and a scale factor of 1.2, we extract FAST corners. We discovered that extracting 1000 corners from images with dimensions between 512 x 384 to 752 x 480 pixels was appropriate.
- Initial Pose Estimation from Previous Frame
  - If tracking was successful for the previous frame, we forecast the camera posture using a constant velocity motion model and do a guided search of the map points seen there.
  - We utilize a broader search of the map points around their location in the previous frame if not enough matches were discovered (i.e., the motion model is obviously broken). The identified correspondences are then used to optimize the stance.
- Initial Pose Estimation via Global Relocalization
  - In the event that tracking is lost, the frame is converted to a bag of words, and keyframe candidates for global relocalization are sought by searching the recognition database.

Tracking

- Track Local Map
  - We can project the map into the frame and look for further map point correspondences after we have an estimate of the camera posture and the first set of feature matches.
  - We merely project a local map in order to limit the complexity of huge maps.
  - This local map includes a set of keyframes K1, which have map points in common with the current frame, as well as a set K2, which are keyframes K1's covisibility graph neighbours.
- New Keyframe Decision
  - Choosing whether or not to generate the current frame as a new keyframe is a final step. We will strive to insert keyframes as quickly as feasible since it makes the tracking more resilient to demanding camera motions, often rotations, and because the local mapping has a way to cull unnecessary keyframes

Local Mapping

- KeyFrame Insertion
  - The covisibility graph is initially updated, with a new node added for Ki and the edges arising from the shared map points with other keyframes updated.
  - After that, we compute the keyframe’s bag-of- words representation, which will aid in data association for triangulating new points.
- Recent Map Points Culling
  - Map points must pass a stringent test within the first three keyframes following the creation in order to be kept in the map. This test verifies that the points are trackable and have not been incorrectly triangulated, that is, as a result of erroneous data association.
  - A point must meet these two requirements:
    - More than 25% of the frames in which the point is expected to be visible must be found by the tracking.
    - Map points must be viewed from at least three keyframes if more than one keyframe has passed after they were created.
-

Local Mapping

- New MapPoint Creation
  - By triangulating ORB from linked keyframes Kc in the covisibility graph, new map points are produced. We look for a match with another unmatched point in a different keyframe for every unmatched ORB in Ki.
- Local Bundle Adjustment
  - The current processed keyframe Ki, every keyframe linked to it in the covisibility graph Kc, and every map point viewed by those keyframes are all optimized by the local BA.
  - The optimization includes any other keyframes that view those points but are not related to the keyframe being processed at the time; yet, they all stay fixed.
- Local Keyframe Culling
  - The local mapping seeks to find unnecessary keyframes and remove them in order to maintain a compact reconstruction.
  - Every keyframe in Kc that has 90% of the map points visible in at least another three keyframes of the same or finer size is discarded. The scale condition makes sure that map points have the keyframes that allow for the most accurate measurements.

Loop Closing

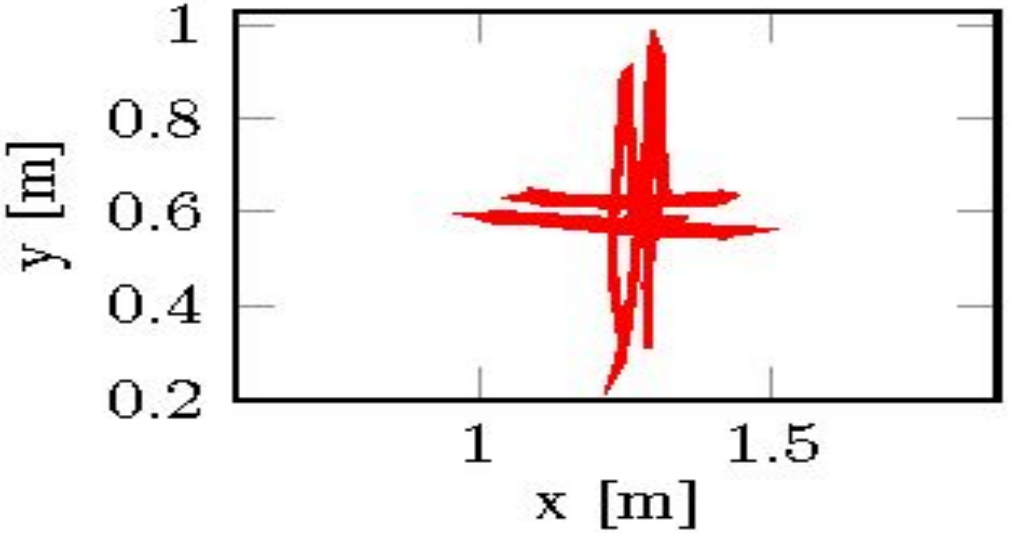
- The loop closing thread uses Ki, the final keyframe that the local mapping processed, to search for and end loops.
- Loop Candidates Detection
  - The bag of words vector of Ki is compared to all of its neighbours in the covisibility graph first (θmin = 30), and the neighbour with the lowest score, smin, is kept.
- Compute the Similarity Transformation
  - We begin by calculating correspondences between the loop candidate keyframes and the ORB associated with the map points in the current keyframe. For each loop candidate at this time, we have 3D to 3D correspondences.
- Loop Fusion
  - Fusing duplicate map points and adding new edges to the covisibility graph, which will attach the loop closure, is the first stage in the loop rectification process.
  - All map points viewed by the loop keyframe and its neighbours are projected onto Ki, and the area immediately around the projection is examined for matches

Experiments

- For evaluating the ORB SLAM performance we have used TUM RGBD dataset. The sequence used is freiburg_xyz
- The input images of the dataset are shown below.

Experiments

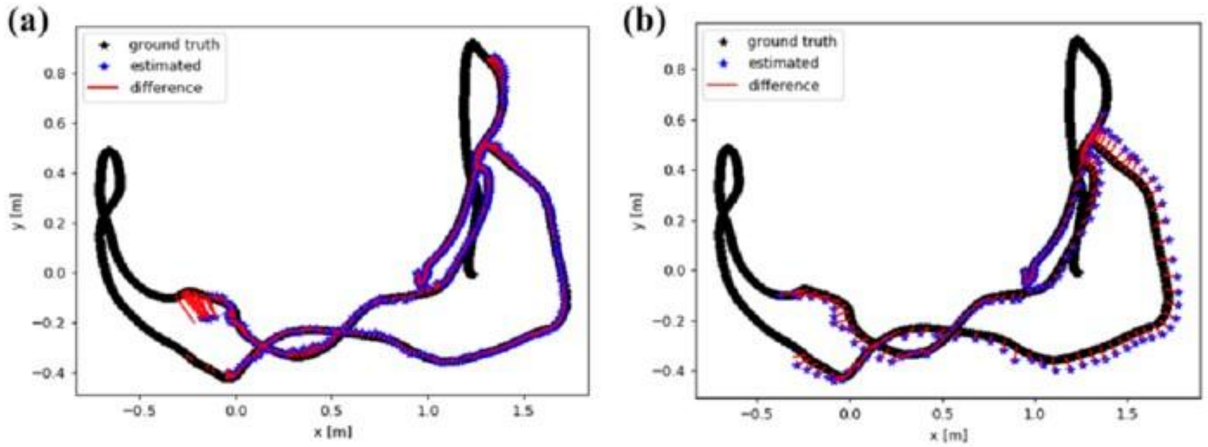
- The Localization output of ORB SLAM is shown in figure below. The figure shows the path of hand held camera, extracted using tracking pipeline of ORB SLAM algorithm



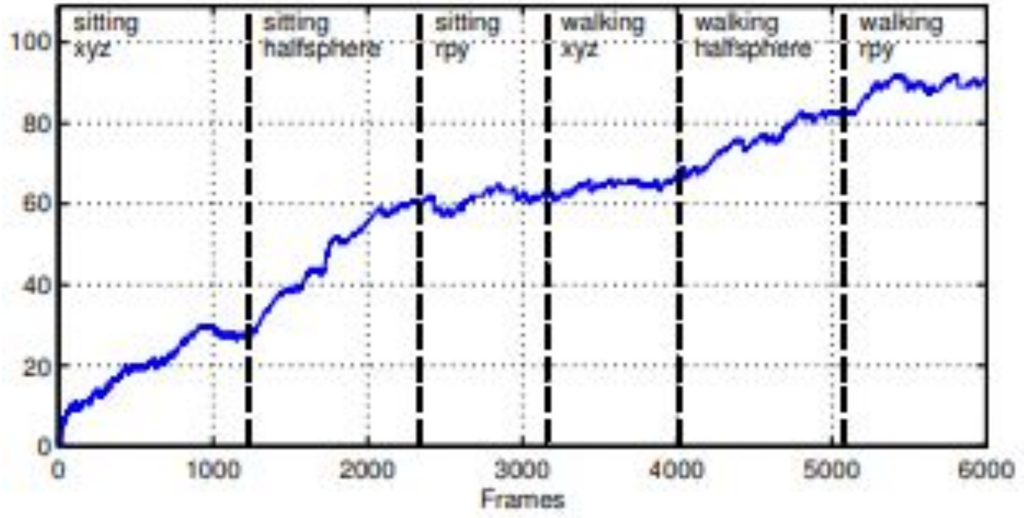
Experiments

- The algorithm is robust to shaky motion and capable of lifelong experiment
- Even when the camera is observing the scene from various angles in a perfectly static environment, ORB SLAM is able to constraint the number of keyframes
- Another important feature of ORB SLAM is that its initialization is more robust as compared to other SLAM methods
- In the freiburg_xyz dataset ORB SLAM doesn’t initialize unless the distinction between the first and the next keyframe is enough to extract meaningful points and correct pose
- The map culling procedure in the implementation ensures that as the no of keyframes grow, the map points does not grow at a large rate

Results

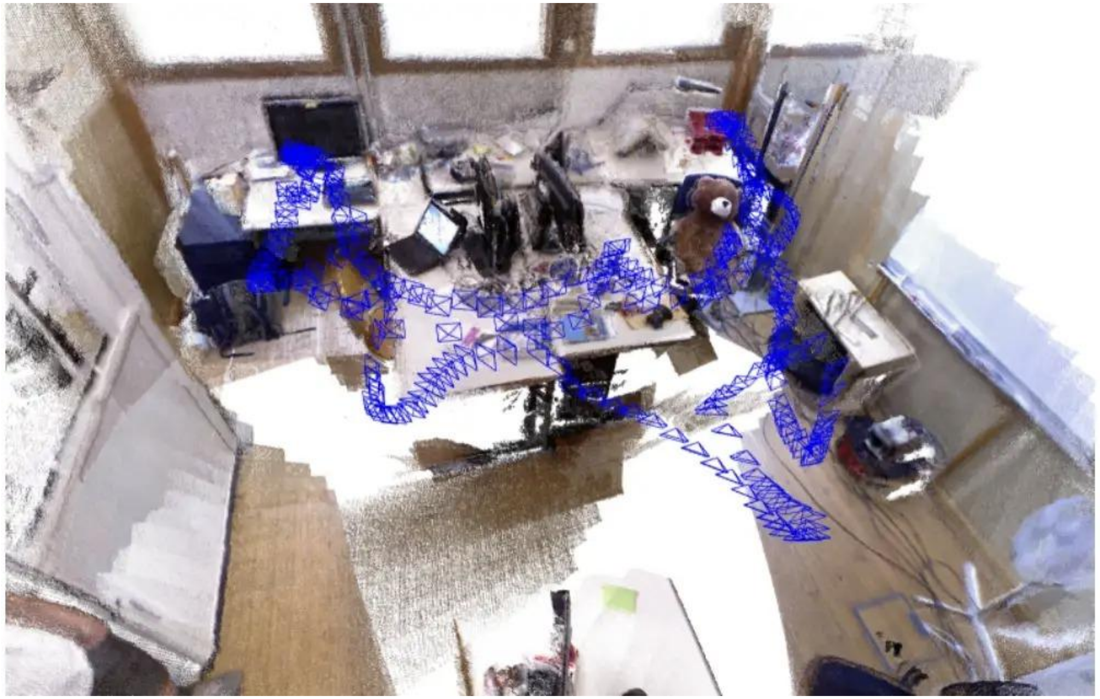


Results on TUM dataset

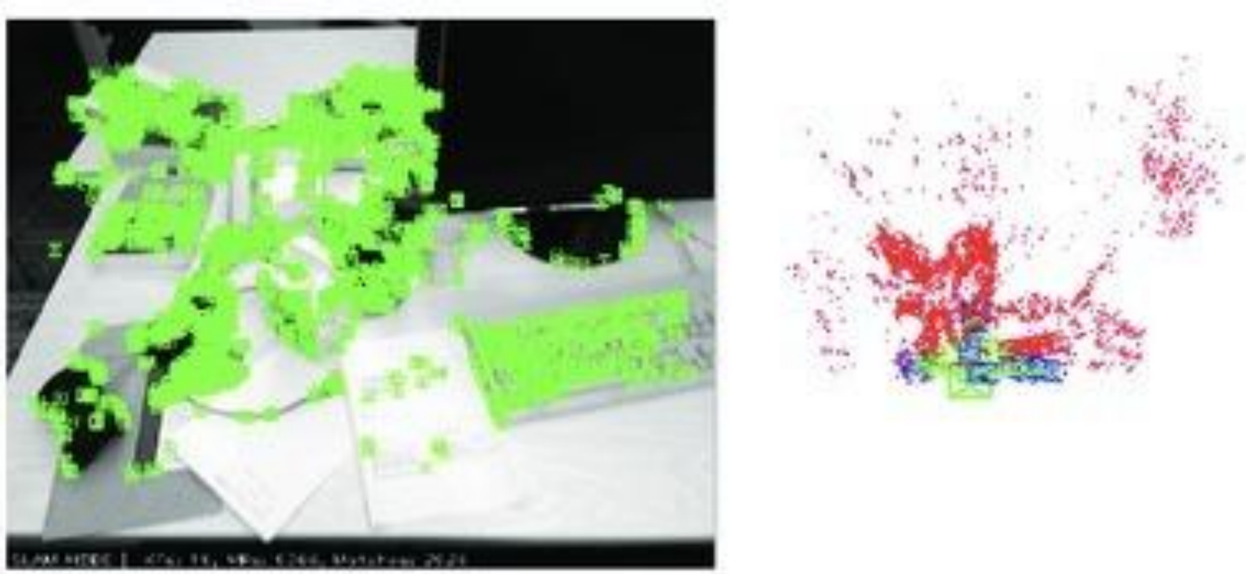


Evolution of the number of keyframes in the map

Results



Point Cloud of a room scene



The scenes of ORB-SLAM changed on TUM dataset

freiburg1_xyz

References

[1] Masunga, Nsingi. “Mobile Robot Navigation in Indoor Environments by using the Odometer and Ultrasonic data.” (1999).

[2] Gatesichapakorn, Sukkpranhachai et al. “ROS-based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera.” 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP) (2019): 151-154.

[3] Biswas, Joydeep and Manuela M. Veloso. “Depth camera based indoor mobile robot localization and navigation.” 2012 IEEE International Conference on Robotics and Automation (2012): 1697-1702.

[4] Gatesichapakorn, Sukkpranhachai et al. “ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera.” 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP) (2019): 151-154.

[5] Weingarten, Jan W. and Roland Siegwart. “EKF-based 3D SLAM for structured environment reconstruction.” 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (2005): 3834-3839.

[6] Cheng, Jiantong et al. “Compressed Unscented Kalman filter-based SLAM.” 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014) (2014): 1602-1607.

[7] T ̈ornqvist, David et al. “Particle Filter SLAM with High Dimensional Vehicle Model.” Journal of Intelligent and Robotic Systems 55 (2009): 249-266. [8] Strasdat, Hauke et al. “Visual SLAM: Why filter?” Image Vision Comput. 30 (2012): 65-77.

[8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-time single camera SLAM,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[9] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in Proc. IEEE ACM Int. Symp. Mixed Augmented Reality, Nara, Japan, Nov. 2007, pp. 225–234.

References

[10] R. Mur-Artal, J. M. M. Montiel and J. D. Tardos, ”ORB-SLAM: A ´ Versatile and Accurate Monocular SLAM System,” in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.

[11] Mur-Artal, Raul and Juan D. Tardos. “ORB-SLAM2: An Open-Source ´ SLAM System for Monocular, Stereo, and RGB-D Cameras.” IEEE Transactions on Robotics 33 (2017): 1255-1262.

[12] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. D. Tardos, ´ “A comparison of loop closing techniques in monocular SLAM,” Robot. Auton. Syst., vol. 57, no. 12, pp. 1188–1197, 2009.

[13] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” in Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog., New York, NY, USA, Jun. 2006, vol. 2, pp. 2161–2168.

[14] M. Cummins and P. Newman, “Appearance-only SLAM at large scale with FAB-MAP 2.0,” Int. J. Robot. Res., vol. 30, no. 9, pp. 1100–1123, 2011.

[15] D. Galvez-Lopez and J. D. Tardos, “Bags of binary words for fast place ´ recognition in image sequences,” IEEE Trans. Robot., vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[16] Strasdat, Hauke et al. “Visual SLAM: Why filter?” Image Vision Comput. 30 (2012): 65-77.

[17] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, “Double window optimisation for constant time visual SLAM,” in Proc. IEEE Int. Conf. Comput. Vision, Barcelona, Spain, Nov. 2011, pp. 2352–2359.

[18] H. Strasdat, J. M. M. Montiel, and A. J. Davison, “Scale drift-aware large scale monocular SLAM,” presented at the Proc. Robot.: Sci. Syst., Zaragoza, Spain, Jun. 2010.

References

[19] K. Pirker, M. Ruther, and H. Bischof, “CD SLAM-continuous localization and mapping in a dynamic world,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., San Francisco, CA, USA, Sep. 2011, pp. 3990–3997.

[20] S. Song, M. Chandraker, and C. C. Guest, “Parallel, real-time monocular visual odometry,” in Proc. IEEE Int. Conf. Robot. Autom., 2013, pp. 4698–4705.

[21] Myriam Servieres, Val ` erie Renaudin, Alexis Dupuis, Nicolas Antigny, ´ ”Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking”, Journal of Sensors, vol. 2021, Article ID 2054828, 26 pages, 2021.

[22] M. Filipenko and I. Afanasyev, ”Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment,” 2018 International Conference on Intelligent Systems (IS), 2018, pp. 400-407, doi: 10.1109/IS.2018.8710464.

[23] Maity, Soumyadip, Arindam Saha, and Brojeshwar Bhowmick. ”Edge slam: Edge points based monocular visual slam.” Proceedings of the IEEE International Conference on Computer Vision Workshops. 2017.

[24] Newcombe, R.A.; Lovegrove, S.J.; Davison, A.J. DTAM: Dense tracking and mapping in real-time. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2320–2327.

[25] Engel, Jakob, Thomas Schops, and Daniel Cremers. ”LSD-SLAM: ¨ Large-scale direct monocular SLAM.” European conference on computer vision. Springer, Cham, 2014.

[26] Engel, Jakob, Vladlen Koltun, and Daniel Cremers. ”Direct sparse odometry.” IEEE transactions on pattern analysis and machine intelligence 40.3 (2017): 611-625.