Visual Simultaneous Localization and Mapping

Amulya Huchachar

Abstract : This article describes simultaneous localization and mapping (SLAM) which is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. While this initially appears to be a chicken-and-egg problem there are several algorithms known for solving it. SLAM consists in the concurrent construction of a model of the environment (themap), and the estimation of the state of the robot moving within it. The SLAM community has made astonishing progress over the last 30 years, enabling large-scale real-world applications, and witnessing a steady transition of this technology to industry.

Introduction : SLAM comprises the simultaneous estimation of the state of a robot equipped with on-board sensors, and the con-struction of a model (themap) of the environment that the sensors are perceiving. In simple instances, the robot state is described by its pose (position and orientation), although other quantities may be included in the state, such as robot velocity,sensor biases, and calibration parameters. The map, on the other hand, is a representation of aspects of interest (e.g.position of landmarks, obstacles) describing the environment in which the robot operates.

The need to use a map of the environment is twofold. First, the map is often required to support other tasks; for instance, a map can inform path planning or provide an intuitive visualization for a human operator. Second, the mapallows limiting the error committed in estimating the state of the robot. In the absence of a map, dead-reckoning would quickly drift over time; on the other hand, using a map, e.g.a set of distinguishable landmarks, the robot can “reset” its localization error by re-visiting known areas (so-called loop closure). Therefore, SLAM finds applications in all scenarios in which a prior map is not available and needs to be built.

In some robotics applications the location of a set of landmarks is known a priori. For instance, a robot operating ona factory floor can be provided with a manually-built map of artificial beacons in the environment. Another example is the case in which the robot has access to GPS (the GPS satellites can be considered as moving beacons at known locations). In Such scenarios, SLAM may not be required if localization can be done reliably with respect to the known landmarks.The popularity of the SLAM problem is connected with the emergence of indoor applications of mobile robotics. Indoor Operation rules out the use of GPS to bound the localization error; furthermore, SLAM provides an appealing alternative to user-built maps, showing that robot operation is possible in the absence of an ad hoc localization infrastructure

Semantic 3D environments are increasingly important in multiple fields, especially in robotics. Nowadays, 3D mapping methods only contains odometry or geometrical information of surrounding environments without semantic meanings, which cannot enable robots to infer more information for specific tasks and makes it difficult for human-robot interaction. A map with semantic information allows robots to fully understand their environments, and generalize its navigation capability,just as humans do, and achieves higher-level tasks. Semantic Information will also enable robots to obtain simple cognition or reasoning ability. Robot perception within semantic infor-mation also makes it possible for robots to achieve language-based

human-robot interaction tasks.SemanticSimultaneouslyLocalizationandMapping(SLAM) system mainly involves the 3D mapping and semantic segmentation. Recently, researches on semanticSLAM are mainly focusing on indoor environments or Lidarbased SLAM system for outdoor environments. Visual basedSemantic SLAM is mainly achieved by using RGB-Depth(RGB-D) camera, which can be greatly affected by lighting conditions and not well-suited for outdoor environments.Lidar is more suitable in such environment, but it is much more costly than camera-based SLAM system. Lidar Contains less information than visual information, which makes the study in camera-based semantic SLAM system more meaningful.

This project is inspired by human visual navigation system.Human navigation system greatly relies on visual perception since the visual images contain considerable information such as odometry, geometrical structures, and semantic meanings. Our navigation from one place to another is mainly based on landmark level semantic meanings, visual features and their topological relationship.

This System is performed by using Oriented FAST and RotatedBRIEF (ORB) features, which has good robustness formoving condition and good real-time performances. It can be used in multiple scenes of outdoor environments with great performance in loop closing. We use ORB-SLAM to extract visual features for re-localization.

Applications : Visual SLAM is still in its infancy, commercially speaking. While it has enormous potential in a wide range of settings, it's still an emerging technology. With that said, it is likely to be an important part of augmented reality (AR) applications. Accurately projecting virtual images onto the physical world requires a precise mapping of the physical environment, and only visual SLAM technology is capable of providing this level of accuracy.

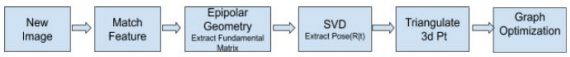
Visual SLAM systems are also used in a wide variety of field robots. For example, rovers and landers for exploring Mars use visual SLAM systems to navigate autonomously. Field robots in agriculture, as well as drones, can use the same technology to independently travel around crop fields. Autonomous vehicles could potentially use visual SLAM systems for mapping and understanding the world around them.

One major potential opportunity for visual SLAM systems is to replace GPS tracking and navigation in certain applications. GPS systems aren't useful indoors, or in big cities where the view of the sky is obstructed, and they're only accurate within a few meters. Visual SLAM systems solve each of these problems as they're not dependent on satellite information and they're taking accurate measurements of the physical world around them.

Visual SLAM technology has many potential applications and demand for this technology will likely increase as it helps augmented reality, autonomous vehicles and other products become more commercially viable.

The ability to sense the location of a camera, as well as the environment around it, without knowing either data points beforehand is incredibly difficult. Visual SLAM systems are proving to be effective at tackling this challenge, however, and are emerging as one of the most sophisticated embedded vision technologies available.

Pipeline :



Method :

Pinhole Camera Model: The visual SLAM gets the information from the cameras so

it's important to know about the pinhole model of the camera which works as a sensor in this project. The pinhole camera model describes the mathematical relationship between the coordinates of a point in three-dimensional space and its projection onto the image plane of an ideal pinhole camera. The principle of a pinhole camera is simple. Objects from the scene reflect light in all directions. The size of the aperture is so small that among the many rays that are reflected off at P, a point on the surface of an object in the scene, only one of these rays enter the camera (in reality it's never exactly one ray, but more a bundle of light rays or photons composing a very narrow beam of light)

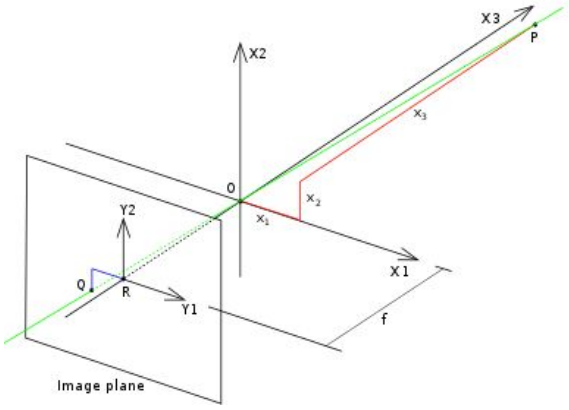


Fig 1 : Geometry of pinhole camera

Camera Calibration : Since the lenses these cameras use are cheap lenses there is some distortion in the images, to solve those distractions it's necessary to do camera calibration. Geometric camera calibration, also referred to as *camera resectioning*, estimates the parameters of a lens and image sensor of an image or video camera. You can use these parameters to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene. These tasks are used in applications such as machine vision to detect and measure objects. They are also used in robotics, for navigation systems, and 3-D scene reconstruction. Today's cheap pinhole cameras introduce a lot of distortion to images. Two major distortions are radial distortion and tangential distortion.

Due to radial distortion, straight lines will appear curved. Its effect is more as we move away from the center of image. For example, one image is shown below, where two edges of a chess board are marked with red lines. But you can see that the border is not a straight line and doesn't match with the red line. All the expected straight lines are bulged out.
This distortion is solved as follows:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Similarly, another distortion is the tangential distortion which occurs because the image taking lense is not aligned perfectly parallel to the imaging plane. So some areas in image may look nearer than expected. It is solved as below:

$$x_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$
$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

In short, we need to find five parameters, known as distortion coefficients given by:

$$Distortion\ coefficients = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3)$$

In addition to this, we need to find a few more information, like intrinsic and extrinsic parameters of a camera. Intrinsic parameters are specific to a camera. It includes information like focal length (f_x, f_y), optical centers (c_x, c_y) etc. It is also called camera matrix. It depends on the camera only, so once calculated, it can be stored for future purposes. It is expressed as a 3x3 matrix:

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Extrinsic parameters corresponds to rotation and translation vectors which translates a coordinate of a 3D point to a coordinate system.

Feature Matching : Once we have multiple frames of images, we need to identify the features matching in both the images. I have used the ORB method for feature matching in this project. ORB is basically a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance. First it uses FAST to find keypoints, then apply Harris corner measure to find top N points among them. It also uses pyra mid to produce multiscale-features. It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this corner point to centroid gives the orientation. To improve the rotation invariance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the patch. ORBuses BRIEF descriptors. But we have already seen that BRIEF performs poorly with rotation. So what ORB does is to "steer" BRIEF according to the orientation of keypoints. For any feature set of n binary tests at location (x_i, y_i), define a $2 \times n$ matrix, S which contains the coordinates of these pixels. Then using the orientation of patch, θ, its rotation matrix is found and rotates the S to get steered(rotated) version S_θ.

Shi and C. Tomasi made a small modification to it in their paper **Good Features to Track**[1] which shows better results compared to Harris Corner Detector.

The scoring function in Harris Corner Detector was given by:

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

Instead of this, Shi-Tomasi proposed:

$$R = min(\lambda_1, \lambda_2)$$

Epipolar geometry : We need to find the fundamental matrix and pose estimation from 3D image to 2D mapping, using the pose estimation we need to do the triangulation. When we take an image using pin-hole camera, we lose important information, i.e. depth of the image. Or how far is each point in the image from the camera because it is a 3D-to-2D conversion. So it is an important question whether we can find the depth information using these cameras. And the answer is to use more than one camera. Our eyes works in similar way where we use two cameras (two eyes) which is called stereo vision. So let's see what OpenCV provides in this field.

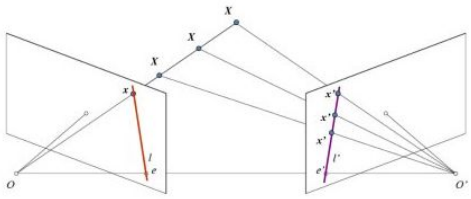


Fig 2 : Triangulation

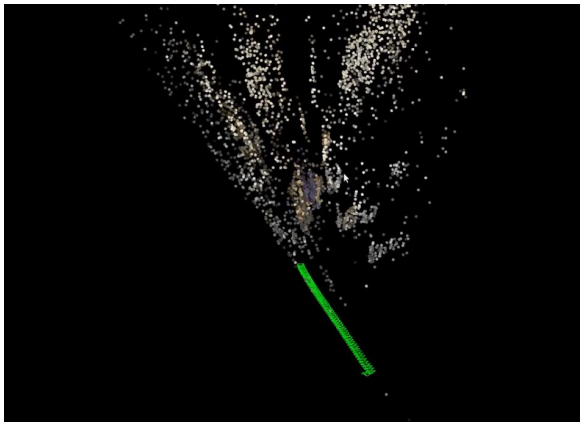
we can't find the 3D point corresponding to the point x in image because every point on the line OX projects to the same point on the image plane. But consider the right image also. Now different points on the line OX project to different points (x′) in the right plane. So with these two images, we can triangulate the correct 3D point. This is the whole idea.The projection of the different points on OX form a line on the right plane (line l′). We call it epiline corresponding to the point x. It means, to find the point x on

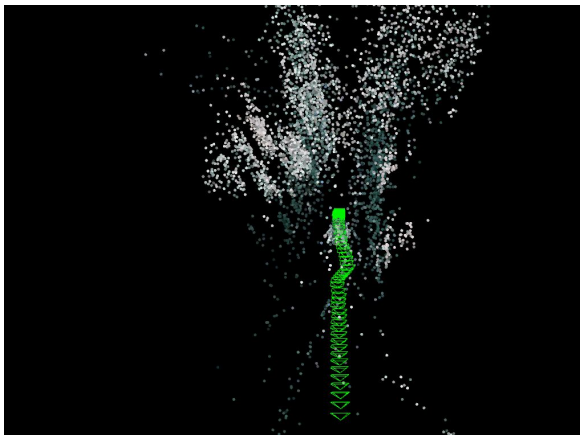
the right image, search along this epiline. It should be somewhere on this line. (Think of it this way, to find the matching point in another image, you need not search the whole image, just search along the epiline. So it provides better performance and accuracy). This is called Epipolar Constraint. Similarly all points will have its corresponding epilines in the other image. The plane XOO′ is called the Epipolar Plane.

We estimate a set of system parameters together with poses of targets in the world. Any sensor that can observe points in the world and is modeled by a measurement equation with gaussian error characteristics can be used with our method. We can easily combine sensor measurements from different coordinate spaces,for example, from lasers (3D points) and cameras (2D image projections). Thetrade-off in accuracy of sensors is handled in a theoretically sound way via co-variances in the measurement equations.

Given a set of images depicting a number of 3D points from different viewpoints, bundle adjustment can be defined as the problem of simultaneously refining the 3D coordinates describing the scene geometry as well as the parameters of the relative motion and the optical characteristics of the camera(s) employed to acquire the images, according to an optimality criterion involving the corresponding image projections of all points.

Experiment : I have taken 2 video recordings of the vehicle movement on the road and fed into the product to get the 3D projection and the vehicle trajectory. And the results looks like below.





Reference: [1]Shi and C. Tomasi. Good Features to Track. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, June 1994.Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary R. Bradski: ORB: An efficient alternative to SIFT or SURF. ICCV 2011: 2564-2571.

https://link.springer.com/chapter/10.1007/978-3-642-28572-1_15

https://scipy-cookbook.readthedocs.io/items/bundle_adjustment.html

https://people.eecs.berkeley.edu/~pabbeel/cs287-fa09/readings/Durrant-Whyte_Bailey_SLAM-tutorial-I.pdf

https://arxiv.org/pdf/1606.05830.pdf

https://arxiv.org/pdf/1902.03747v2.pdf

https://arxiv.org/pdf/2001.01028v1.pdf

https://en.wikipedia.org/wiki/Pinhole_camera

https://en.wikipedia.org/wiki/Pinhole_camera_model

https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping

https://ipsjcva.springeropen.com/articles/10.1186/s41074-017-0027-2

https://www.visiononline.org/blog-article.cfm/What-is-Visual-SLAM-Technology-and-What-is-it-Used-For/99

https://en.wikipedia.org/wiki/Bundle_adjustment

http://www.cs.jhu.edu/~misha/ReadingSeminar/Papers/Triggs00.pdf

https://homes.cs.washington.edu/~sagarwal/bal.pdf

https://en.wikipedia.org/wiki/Epipolar_geometry

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT10/node3.html

http://www.cs.toronto.edu/~jepson/csc420/notes/epiPolarGeom.pdf

https://docs.opencv.org/master/d7/d53/tutorial_py_pose.html

https://docs.opencv.org/master/dc/d2c/tutorial_real_time_pose.html

https://openslam-org.github.io/g2o.html

https://fzheng.me/2016/03/15/g2o-demo/

https://www.ros.org/news/2011/05/g2o-framework-for-graph-optimization-released.html