Parallel Tracking and Mapping (PTAM)

Jerrin Bright[1]

[1]Vellore Institute of Technology, School of Mechanical Engineering,
Chennai, Tamil Nadu, India
Jerrin.bright2018@vitstudent.ac.in

Abstract: PTAM is a very computationally cheap and efficient SLAM technique used especially in AR VR applications considering its computation capacity. It uses a parallel threading method whereby tracking and mapping occurs simultaneously. We will be discussing this PTAM SLAM technique in detail with in-depth conceptual in this work starting with tracking, mapping to the refinement using Bundle Adjustment techniques to reduce the reprojection error rates when the estimating the pose of the camera.

Keywords: *PTAM, SLAM, Bundle Adjustment, Mapping*

1. INTRODUCTION

PTAM is an BA algorithm/ technique. Bundle Adjustment can be defined as simultaneous refining of the 3D coordinates describing the scene geometry, the parameters of the relative motion, and the optical characteristics of the camera employed to acquire the images. The tracking and mapping are done parallelly to save the computation cost. The architecture of PTAM is visualized in Fig. 1.

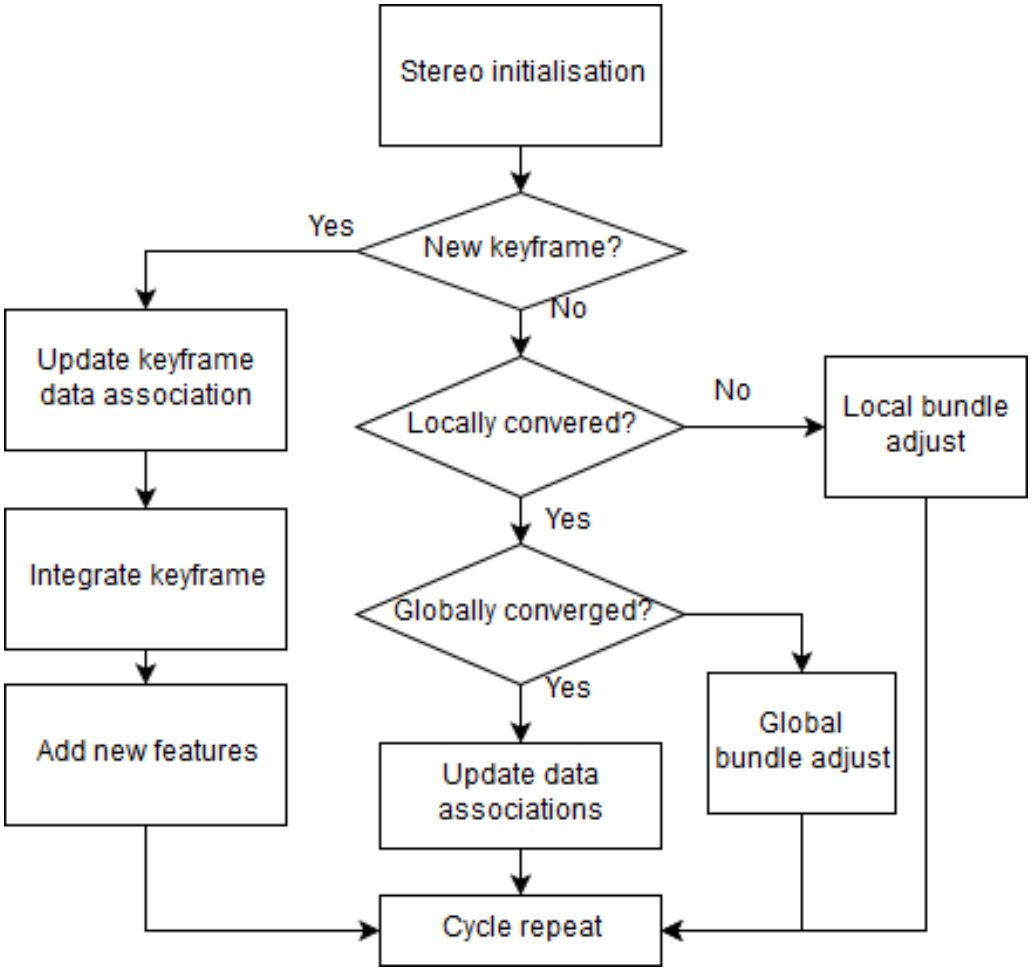

Figure 1. Architecture of PTAM

The five important steps involved in PTAM are: Tracking and mapping is separated and run in 2 parallel threads; Mapping is based on keyframes, which are processed using BA; The map is densely initialized from a 5-Point Algorithm; New points are initialized with an epipolar search and Large numbers of points are mapped. We will be discussing each and every step mentioned above in detail in this work.

The two major phases are Tracking and Mapping which is done parallelly in different threads (nowadays, all computers have more than one core). The major advantage of processing separately is:

- Tracking will not be slaved to the map making procedure
- Thorough processing can be done as the computational burden to update a map in every frame.
- Every single frame isn't necessary to be used, as there will be a lot of redundance, especially when the robot is hovering.

2. Tracking

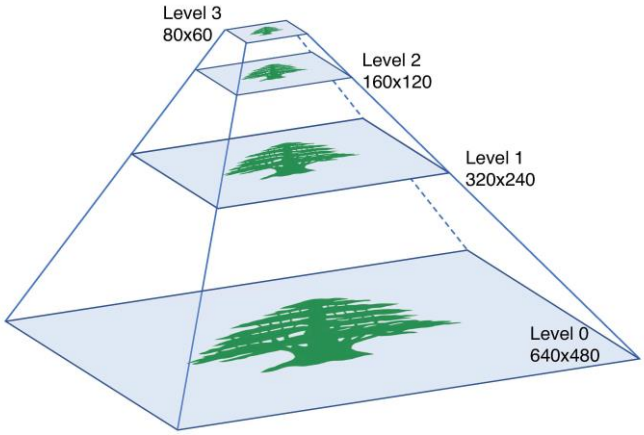

Figure 2. Pyramid representation of an image

PTAM generated a 4-level pyramid representation of every frame it obtains as shown in the fig. 2. It uses this structured data to enhance the features robustness towards scale changes and to increase the convergence ratio of the pose estimates. Each level is formed by blurring the level before it and sub-sampling by a factor of 2.

2.1 Image Acquisition

The captured images are converted into 8 bits per pixel greyscale for tracking purposes. On each of the pyramid levels, FAST-10 corner detector algorithm is run resulting in a blob-like cluster of corner regions.

2.2 Camera Pose and Projection

Projecting map points into the image is vital for tracking. This is done by first converting the world coordinate frame to a camera-centered coordinate frame. This conversion is done by left multiplication with the camera pose. CW represents frame C from frame W. J^{th} point of the map is represented by pj. The transformation between the camera-centered coordinate frame and the world is called E_{CW}. Thus, in the first equation we will find a particular point on the map in the coordinate frame W.

$$\boldsymbol{p}_{jC} = E_{CW}\boldsymbol{p}_{jW}$$

CamProj() is a camera calibrated projection model used to projecting the points in the camera frame to the image.

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \text{CamProj}(E_{CW}\boldsymbol{p}_{iW})$$

The below equations show the pin-hole camera projection parameters including focal length, principal points and the distortion

$$\text{CamProj}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \begin{bmatrix} f_u & 0 \\ 0 & f_v \end{bmatrix} \frac{\mathrm{r}'}{\mathrm{r}} \begin{pmatrix} \frac{x}{z} \\ \frac{y}{z} \end{pmatrix}$$

$$r = \sqrt{\frac{x^2 + y^2}{z^2}}$$

$$r' = \frac{1}{\omega} \arctan(2r \tan\frac{\omega}{2})$$

The major requirement of tracking is to observe change in the change in the position of the map with respect to change in the camera pose. This is shown in the below equation which is done using the left multiplication with respect to the camera motion which is represented by the letter M.

$$E'_{CW} = ME_{CW} = \exp(\boldsymbol{\mu})E_{CW}$$

μ is the 6-vector parameter used here, representing 3 translational and 3 rotation axis and magnitude.

2.3 Patching

To find a single point p in the map, a fixed range image search is done around the points of the predicted image. To perform this search, the corresponding patch must first be warped to take account of viewpoint changes between the patch's first observation and the current camera position. So, the warping matrix is shown below where u and v are the horizontal and vertical pixel displacement in the patch's source pyramidal level and u_c and v_c are pixel displacement of the current camera frame.

$$A = \begin{bmatrix} \frac{\partial u_c}{\partial u_s} & \frac{\partial u_c}{\partial v_s} \\ \frac{\partial v_c}{\partial u_s} & \frac{\partial v_c}{\partial v_s} \end{bmatrix}$$

The differential of the A matrix tells us in which pyramid level, the patch should be searched.

2.4 Pose Update

From the patch observations obtained the pose of the camera can be computed. From each observation, patches pose is found with an assumption of noise measured. Thus, we will calculate the pose by iterating continuously till convergence is observed from any set of measurements. Obj is the objective function and **σ** represents the robust estimate of the standard deviation of the distribution derived.

$$\boldsymbol{\mu}' = \underset{\boldsymbol{\mu}}{\text{argmin}} \sum_{j \in S} \text{Obj}\left(\frac{|\boldsymbol{e}_j|}{\sigma_j}, \sigma_T\right) \qquad \boldsymbol{e}_j = \begin{pmatrix} \hat{u}_j \\ \hat{v}_j \end{pmatrix} - \text{CamProj}\left(\exp\left(\boldsymbol{\mu}\right) E_{CW}\boldsymbol{p}_j\right)$$

The pose update is thus computed iteratively by curtailing the objective function of the reprojection error.

2.5 Tracking

At first, a frame will be taken from the live video and a prior estimate for the camera is generated from the motion model. Depending on the prior pose estimate the map points will be visualized. To increase

the tracking resilience to rapid camera movement the patch search and pose update will be done twice in PTAM. A small amt (50) of coarse scale features will be searched in the image. It is done in the highest level of the pyramid representation of the image. From these coarse matched features, the camera pose will be measured and updated. A large number of points (1000) are now re-projected from the remaining potential points obtained and searched in the image over a smaller range.

3. Mapping

Map building occurs in 2 steps. They are: Building the initial map using stereo technique and Map will be continuously refined and expanded as new keyframes are added by the tracking system.

3.1 Initial Map

We use 5-point algorithm to initialize the map initially. The initial map will consist of only 2 keyframes taken by user upon which the 5-point algorithm and RANSAC can estimate and triangulate the base map.

3.2 Refining and Expanding Map

As the camera moves away from the 2 keyframes formed in the initial map, new keyframes and map features are thus added to the system to allow the map to expand.

4. Refinements

4.1 Bundle Adjustments

It is defined as simultaneous refinement of the 3D coordinate describing the scene geometry and the optical characteristics of the camera giving input images. It is basically applied as the very last step of feature-based 3D reconstruction. It is used to remove the noise pertaining the image features observed.

4.2 Data Association Refinement

Once BA has converged, data association will be done. New measurements will be made in the old keyframes. When new features are added, they will be checked if it is an inlier or outlier. If outlier it will be cast off the map. This will be very useful in cases where tracking goes wrong which happens. If the measurements are given low weights by the M-estimator in the BA refinement process, they will be considered to be outlier. A second chance is given by the data association refinement, where its keypoint will be re-measured in the keyframe with a very tight search region. If outlier even after this measurement, it will be cast off permanently. If it is an inlier, it will be added to the map.

This process is given the least priority in the hierarchy. Only if there is no process at hand to be done, this refinement will be done. That's, only if there are no other keyframes to be measured, it is done and once a new keyframe comes in it will be abruptly stopped.

5. References

[1]. https://github.com/uoip/stereo_ptam
[2]. https://www.robots.ox.ac.uk/~gk/PTAM/
[3]. https://github.com/ethz-asl/ethzasl_ptam
[4]. https://github.com/Oxford-PTAM/PTAM-GPL