Stereo Visual SLAM

James Yang

***Abstract*— In this document, we discuss a basic implementation of visual simultaneous localization and mapping using a temporal sequence of stereo images. The algorithm makes use of epipolar geometry to create 3D points and, at various points, Levenberg – Marquardt algorithm for minimizing reprojection error between multiple views. The full algorithm is demonstrated and future work is discussed at the end.**

I. INTRODUCTION

Stereo visual SLAM is the process of recreating the 3-D structure of an environment using a set of 2-D images. While a monocular sequence can be used using the same mathematical principles, the reconstruction is only true up to a scale factor. Stereo visual SLAM adds a constraint that two images are taken simultaneously over time by two separate cameras separated by some known distance. With this additional constraint, the scale factor problem is eliminated, and 3-D reconstruction can be achieved with proper units and scale.

Stereo visual SLAM is highly desirable in many modern robotics applications such as autonomous cars, drone delivery, and other tasks involving outdoor navigation. Many of these systems seek to replace very expensive LIDAR systems with cheaper alternatives. Cameras prove to be a highly robust and accurate alternative that can also provide very dense data at a significantly lower cost.

In the following, the paper first discusses the mathematical principles behind point triangulation. The paper then discusses reprojection error minimization using local bundle adjustment via the Levenberg-Marquardt algorithm. The paper then concludes with results using the KITTI dataset[1][2][3], final thoughts, and future work.

II. STEREO VSLAM PIPELINE

The stereo VSLAM algorithm is an iterative process outlined in the following:

1) Matches features between two sets of sequential stereo image pairs.
2) Triangulate the features matched between the first stereo pair in 3-space.
3) Estimate the pose of the second stereo image pair using the features associated with the triangulated points.
4) Perform local bundle adjustment to minimize projection error.
5) Repeat these steps using the second pair of images.

This process assumes a known set of initial poses. Given $\mathbf{x}_L(t)$ and $\mathbf{x}_R(t)$ are the trajectories of the left and right cameras, we seed the initial poses as $\mathbf{x}_L(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ and $\mathbf{x}_L(0) = \mathbf{t}_R$ where $\mathbf{t}_R$ is the constant translation between the left and right cameras.

At any given point, we assume that the right camera is always offset from the left camera by the vector $\mathbf{t}_R$. Ergo, given the rotation of the left camera $\mathbf{R}_L(t)$, we create a homogeneous transformation,

$$\mathbf{H}_L(t) = \begin{bmatrix} \mathbf{R}_L(t) & \mathbf{x}_L(t) \\ \mathbf{0} & 1 \end{bmatrix}$$

where the pose of the right camera can be extracted from the homogeneous transformation $\mathbf{H}_R(t)$ easily calculated as

$$\mathbf{H}_R(t) = \mathbf{H}_L(t) \begin{bmatrix} \mathbf{I} & \mathbf{t}_R \\ \mathbf{0} & 1 \end{bmatrix}$$

III. FEATURE TRIANGULATION

A. Outlier rejection using RANSAC

To triangulate features, we first reject potential outliers that may have resulted from measurement noise. To do this, we use the epipolar constraint that given the projection of a 3-D point onto two separate images, there exists a nullspace F also known as the fundamental matrix such that

$$x_l{}^T \mathbf{F} x_r = 0$$

Using an 8 point RANSAC algorithm, we find the fundamental matrix that maximizes the number of points that fall under the epipolar constraint. Those points are then used for triangulation. Figures 1 and 2 show an example of outlier rejection.

B. Triangulation

Using the projective geometry equation

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

*This work was not supported by any organization

[1]Albert Author is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands albert.author@papercept.net

[2]Bernard D. Researcheris with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA b.d.researcher@ieee.org

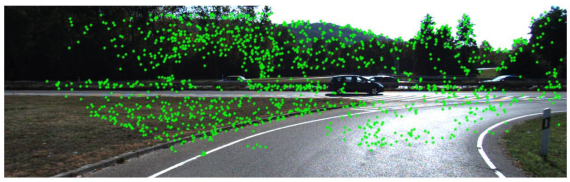

Fig. 1: Pre-RANSAC features.

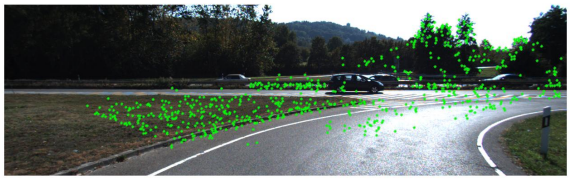

Fig. 2: Post-RANSAC features. Outliers are rejected.

where

$$\mathbf{P} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$$

we have the expectation that

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} - \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{0}$$

We can reformulate this problem as

$$\left[\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}_\times \mathbf{P} \right] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

where given

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}_\times = \begin{bmatrix} 0 & -1 & y \\ 1 & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

Given a second projection of the same point to another image, the equation can be expanded to

$$\begin{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times & \mathbf{P}_1 \\ \begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}_\times & \mathbf{P}_2 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{0}$$

we reduce the problem to 4 degrees of freedom, by which we can use singular value decomposition to solve for $\mathbf{X}$. Letting

$$A = \begin{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ 1 \end{bmatrix}_\times & \mathbf{P}_1 \\ \begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix}_\times & \mathbf{P}_2 \end{bmatrix}$$

we decompose A as the following

$$A = U \Sigma V^T$$

where

$$V = [v_1 \mid v_2 \mid v_3 \mid v_4]$$

We further see that

$$v_4 = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

By this, we have

$$\mathbf{X} = \begin{bmatrix} a/d \\ b/d \\ c/d \end{bmatrix}$$

Figure 3 shows an instance where linearly triangulated points are reprojected to the image plane. It is noted that the estimation, while decent, is not quite of the desired quality.

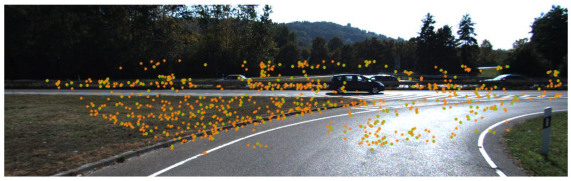

Fig. 3: Reprojection of linearly estimated 3-D points.

C. Non-linear error minimization

Given the triangulation points from the previous step, the points can be further refined to reduce the reprojection error resulting from noise in the calculations in triangulation. Since these 3-D points were calculated using two different views, we would ideally see that the point $\mathbf{X}$ would project onto the two frames at $\mathbf{x}_1$ and $\mathbf{x}_2$.

We see that

$$\mathbf{P} = \begin{bmatrix} -p_1- \\ -p_2- \\ -p_3- \end{bmatrix}$$

Using $\mathbf{P}_1$ and $\mathbf{P}_2$ from triangulation, we solve for the correct $\mathbf{X}$ that minimizes the reprojection error. This problem is formulated as the following

$$\min_{\mathbf{X}} \sum_i^2 \left|\left| \mathbf{x}_i - \frac{1}{p_{i,3}\mathbf{X}} \begin{bmatrix} p_{i,1}\mathbf{X} \\ p_{i,2}\mathbf{X} \end{bmatrix} \right|\right|_2^2$$

This problem is solved using the Levenberg-Marquardt algorithm.

Figure 4 shows an instance of the refined 3-D points reprojected onto the image plane. We see extremely accurate results post-error minimization.

Fig. 4: Reprojection of refined 3-D points.

IV. NEW POSE CALCULATION

A. Estimation

Using the refined 3-D points, the next pair of images can be located by minimizing the projection error of the 3-D point onto the new images. The new equation we solve for is very similar to the triangulation equation, but instead of solving for $\mathbf{X}$, we are now solving for $\mathbf{P}$. Let

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$

Using our definition for $\mathbf{x}$ from before, we have the new problem

$$\begin{bmatrix} \begin{bmatrix} \mathbf{0}_{1\times4} & -\tilde{\mathbf{X}}_1^T & y_1\tilde{\mathbf{X}}_1^T \\ \tilde{\mathbf{X}}_1^T & \mathbf{0}_{1\times4} & -x_1\tilde{\mathbf{X}}_1^T \\ -y_1\tilde{\mathbf{X}}_1^T & x_1\tilde{\mathbf{X}}_1^T & \mathbf{0}_{1\times4} \end{bmatrix} \\ \begin{bmatrix} \mathbf{0}_{1\times4} & -\tilde{\mathbf{X}}_2^T & y_2\tilde{\mathbf{X}}_2^T \\ \tilde{\mathbf{X}}_2^T & \mathbf{0}_{1\times4} & -x_2\tilde{\mathbf{X}}_1^T \\ -y_2\tilde{\mathbf{X}}_2^T & x_2\tilde{\mathbf{X}}_2^T & \mathbf{0}_{1\times4} \end{bmatrix} \\ \vdots \end{bmatrix} \begin{bmatrix} p_1^T \\ p_2^T \\ p_3^T \end{bmatrix} = \mathbf{0}$$

$\mathbf{P}$ can solved using singular value decomposition.

Figure 5 shows an instance of the refined 3-D points reprojected onto the new image using the linearly estimated camera pose. We can see many of the points not overlapping.

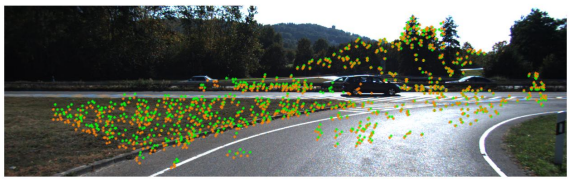

Fig. 5: Reprojection of refined 3-D points onto new camera pose.

B. Non-linear error minimization

In the same way that the 3-D triangulation was refined, the camera pose is also refined. The problem is formulated as the following:

$$\min_{\mathbf{P}} \sum_i^n \left|\left| \mathbf{x}_i - \frac{1}{p_3\mathbf{X}_i} \begin{bmatrix} p_1\mathbf{X}_i \\ p_2\mathbf{X}_i \end{bmatrix} \right|\right|_2^2$$

Figure 6 shows an instance of the refined 3-D points reprojected onto the new image using the linearly estimated camera pose. We can see some points not overlapping.

Fig. 6: Reprojection of refined 3-D points onto new camera pose.

V. LOCAL BUNDLE ADJUSTMENT

Given a set of refined camera poses and 3-D points, all parameters are further refined to minimize the reprojection error of all the points in all the camera views. One constraint is that one of the camera poses remains fixed in space, and the rest may be optimized upon. Given m different views of n different points, we solve the following:

$$\min_{\mathbf{P}_k,\mathbf{X}_l,\mathbf{x}_l} \sum_j^m \sum_i^n \left|\left| \mathbf{x}_{i,j} - \frac{1}{p_{j,3}} \mathbf{X}_{i,j} \begin{bmatrix} p_{j,1}\mathbf{X}_{i,j} \\ p_{j,2}\mathbf{X}_{i,j} \end{bmatrix} \right|\right|_2^2$$

where $k \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, n\}$.

This is also accomplished using the Levenberg-Marquardt algorithm.

VI. RESULTS

Figure 7 shows part of one of the reconstructed scenes.

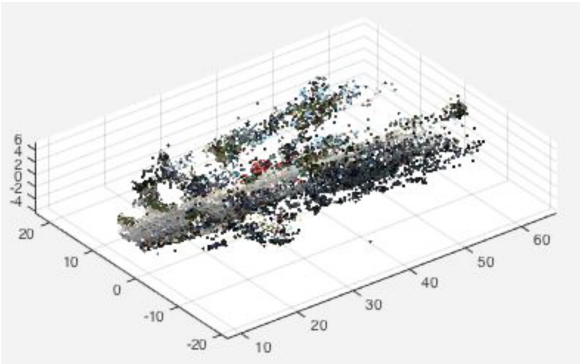

Fig. 7: Sample reconstruction.

Sample videos are attached, labeled reconstruct_x.avi. It is noted that while on most straightaways the algorithm performs quite well, at certain points in the algorithm's progression the

camera's pose suddenly teleports to the origin. As of the writing of this document, it is still unknown as to why this occurs, though potential issues could be in enforcing the special orthogonality constraint of the rotation matrix in estimating camera pose.

VII. CONCLUSION

Based on initial results, it is clear that some sort of probabilistic state estimation is necessary to accommodate large deviations from previous pose estimations. Future work will be focused on implementing a Kalman Filter to smooth out sudden movements as well as keeping track of the triangulated features. A probabilistic model for the camera poses and triangulation is expected to produce far superior results than brute-forcing each calculation. Furthermore, optical flow can be used to generate a better motion model, and could be a beneficial addition to the algorithm.

VIII. RUNNING THE CODE

Run binocular_VSLAM.m in the VSLAM directory. This code does not run in real time.

IX. CONCLUSIONS

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

REFERENCES

[1] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[2] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.