High-performance Visual Odometry with Two-stage Local Binocular BA and GPU

Wei Lu, Zhiyu Xiang* and Jilin Liu
Department of Information Science & Electronic Engineering
Zhejiang University, Hangzhou, China
lwhfh01@zju.edu.cn

Abstract— Visual odometry becomes an important method to deal the localization work in intelligent vehicle and robotics. A high performance visual odometry needs to achieve two requirements: high accuracy and high frequency. So we propose a two-stage local binocular bundle adjustment algorithm doing the optimization and construct a parallel pipeline using GPU acceleration. Finally, our system can run at about 35–40 frames per second with the maximum RMS 3D localization error less than 1%.

I. INTRODUCTION

Motion estimation is very important for autonomous land vehicle (ALV), intelligent robot (IR) and other navigation systems. Traditional navigation sensors like Global Positioning System (GPS) and inertial measurement units (IMU) have been widely used but still suffer many shortcomings and limitations. GPS cannot work indoors or in the shadow of any shelter, let alone on the moon or mars where GPS is unavailable. Low-cost IMU quickly degrades and expensive ones will still drift unless corrected. To solve this problem, a vision based method — visual odometry (VO) was presented by Nister in 2004 [1], which has been rapidly developed and widely used recently. For its high price-performance ratio and lower demand of environment conditions, VO gets more and more attention as an important component of navigation systems.

VO could estimate the motion of a vehicle on which camera is fixed producing sequential, overlapping pictures of the environment. From the view of system input, VO could be divided into two categories by the number of camera eyes: monocular and binocular. And from the view of key technique used, VO could be realized based on optical flow or feature points. Depending on the specific choices of VO’s architecture, the fundamental assumption and output result will be different. Generally speaking, monocular VO cannot estimate the scale information using barely images [2], and optical flow method usually requires that the vehicle’s motion is planar. For the sake of estimating entire 6 DOF (degree of freedom, 3 DOF for position and 3 DOF for orientation) of motion using only images, we focus on the binocular VO based on feature points.

The most important advantage of binocular VO is the rich 3D information reconstructed, whose accuracy is directly related to the feature point algorithm. [3] analyzes and evaluates some most popular features like SIFT [4], SURF and CenSurE [5]. According to their conclusions and our own practice, SIFT feature is more accurate and stable while SURF and CenSurE are faster. [6] constructed a VO using CenSurE feature and run 10Hz on a 2GHz Pentium CPU. [7] proposed a more simplified detector and descriptor, and their VO can run 25Hz on a 3GHz i7 CPU. We want to start with the feature having accurate localization and distinctive descriptor, so the SIFT feature is chosen for its good performance against rotation, scale changes and illumination variance. In order to achieve much higher speed, we use GPU to accelerate the algorithm. By taking full account of the specific conditions of VO, we simplify the SIFT feature greatly, and do the binocular feature tracking at 40Hz with GPU.

Bundle adjustment (BA) [8] is one of the most popular tools for recovering accurate motion and structure from long image sequences. It gives the optimal solution through minimizing a cost function of overall reprojection error. Later, local bundle adjustment (LBA) [9] was presented, applying BA to incremental motion estimation. By using a sliding window, LBA can prevent computation cost from increasing with the length of image sequence. [10] compared LBA with global BA in details and talks about how to select the size of sliding window. Since BA uses non-linear Newton-type optimizers such as Levenberg-Marquardt, accurate initial values can help the algorithm rapidly converge and away from the local minimum. Besides, BA minimizes the sum of squares of reprojection errors, and the reprojections should not be treated equally because of different localization uncertainties.

Our main contribution is an improved BA algorithm over the LBA, which we call it TLBBA (Two-stage Local Binocular Bundle Adjustment). When TLBBA is used in VO, it improves the accuracy dramatically with only comparable computational cost. In the first stage of TLBBA, the single-step motion is optimized by an interleaved bundle adjustment which has a new binocular structure and more reasonable weighting factors using feature uncertainty. It is notable that traditional monocular BA cannot be used between only two views because of insufficient constraint, and it tends to suffer larger errors or even give a wrong solution [11]. The second stage of TLBBA is also benefit from the binocular structure and uncertainty weighting. And with more accurate initial values from the first stage, the multiple-view LBA robustly searches the optimal motion model in a sliding window and quickly converges. Finally, our system can run at about 40Hz with maximum RMS 3D localization error less than 1%, using only images and without manual intervention.

* corresponding author and e-mail: xiangzy@zju.edu.cn
II. THE PARALLEL PIPELINE OF VO

Our system is a little similar to [12], but adding the new TLBBA optimization and some acceleration methods. As illustrated in Fig. 1, our pipeline consists of two threads: feature tracking thread and motion estimation thread. When the binocular images I<sub>n</sub> captured, the first thread detect and describe features of I<sub>n</sub>, then track the features and reconstruct 3D points P<sub>n</sub> and P<sub>n-1</sub>; The second thread estimate the motion between P<sub>n</sub> and P<sub>n-1</sub>; using RANSAC and quaternion method, then TLBBA is performed here to optimize the result.

A. Feature Detection and Description

We adopt the SiftGPU of Wu [20], where GPU is used to accelerate the SIFT feature’s detection, description and matching, the detail of which is described in [13] and [14]. We use Intel Core 2.8GHz i5 CPU and NVIDIA GTS450 video card. The SiftGPU takes 80 ms to process one 640*480 color image, which seems not fast enough. In our application, because the motion of one single step is relatively small, we can give up some performance on the scale invariance, and just one or two scales retained can afford the tracking task. The slimmed-down SiftGPU could deal the tracking in about 40ms to process binocular 640*480 images. Furthermore, we try to use smaller images (480*300), which will further reduce the computation cost. Though smaller image leads to less feature number and worse initial estimation result, after our TLBBA optimization, the maximum RMS 3D localization error will be still controlled below 1%.

B. The Uncertainty of Feature

SIFT detects feature point through searching extrema on the DOG (difference of Gaussian) pyramid. If a feature is located at \( p = (x, y)^T \) on scale \( \sigma_i \) layer, we denote its DOG response as:

\[
D(p, \sigma_i) = \left( G(p, \sigma_{i+1}) - G(p, \sigma_i) \right) * I(p)
\]  

where \( D(p, \sigma_i) \) is the second derivative of \( C(p, \sigma_i) \) on scale \( \sigma_i \).

According to [15], the uncertainty of feature localization can be reflected by covariance estimates, and measured by the inverse of the Hessian matrix:

\[
\Sigma_i = \pm \begin{bmatrix} D_{xx}(p, \sigma_i) & D_{xy}(p, \sigma_i) \\
D_{xy}(p, \sigma_i) & D_{yy}(p, \sigma_i) \end{bmatrix}^{-1}
\]  

where \( D_{xx}, D_{xy} \) and \( D_{yy} \) are the second derivative of \( D(p, \sigma_i) \). In consideration that features may be on different scale layers, we project all \( \Sigma_i \) to the scale layer \( \sigma_0 \) in order to facilitate comparisons:

\[
\Sigma = \Sigma^0 = \sum_i \left( \frac{\text{res}(D(\bullet, \sigma_0))}{\text{res}(D(\bullet, \sigma_i))} \right)^2
\]  

\( \text{res()} \) is the image resolution. This uncertainty will be used as weighting factor of the reprojection error for each feature point in the TLBBA algorithm.

C. Motion Estimation

After feature matching between the left and right images, we can reconstruct the feature point’s 3D positions in the camera coordinates system through triangulation. In Fig. 2, \( C_l, C_r \) denote binocular camera centers; \( f, B \) denote focal length and baseline; \( (u_0, v_0)^T \) is principal point; \( P = (P_x, P_y, P_z)^T \) is the reconstructed 3-D point; \( \mathbf{p}_l = (x_l, y_l)^T \) and \( \mathbf{p}_r = (x_r, y_r)^T \) are the matched features’ coordinates or measured projections on left and right images respectively. And we have:

\[
\mathbf{p} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \mathbf{F} (\mathbf{p}_l, \mathbf{p}_r) = \frac{B}{x_l - x_r} \begin{bmatrix} x_l - u_0 \\ y_l - v_0 \\ f \end{bmatrix}
\]  

If \( P_l \) and \( P_r \) are 3D points reconstructed by current frame and previous frame respectively, the rotation \( \mathbf{r} \) and translation \( \mathbf{t} \) between them satisfy:
Feature matching between current frame and previous frame could provide hundreds of 3D point pairs, so there may be some outliers caused by mismatch. To reject the influence of outliers, RANSAC method [16] is introduced, and also accelerated by GPU. We use the quaternion method proposed by [17] to estimate the camera motion. Typically, to perform estimation once, we randomly select 3 (the least number required to estimate \( \mathbf{r}, \mathbf{t} \) point pairs from all the matched 3D points, estimate \( \mathbf{r}, \mathbf{t} \) with quaternion method, count the inliers, iterate above steps for enough times and finally select the motion model owning most inliers.

To find the optimal solution of the \( n(n > 3) \) inliers, traditional operation here is a least square method minimizing the sum of squares of errors (SSE):

\[
\left( \mathbf{r}^*, \mathbf{t}^* \right) = \arg \min_{\mathbf{r}, \mathbf{t}} \sum_{j=1}^{n} \left\| \mathbf{e}_j \right\|^2
\]

\[
\mathbf{e}_j = \mathbf{p}_{cj} - (\mathbf{r} \mathbf{p}_{pj} + \mathbf{t})
\]

where \( \mathbf{e}_j \) is a 3-vector which represents the 3D space position error between the current frame point \( \mathbf{p}_{cj} \) and the point transformed from previous frame: \( \hat{\mathbf{p}}_{cj} = (\mathbf{r} \mathbf{p}_{pj} + \mathbf{t}). \)

The solution from (6) and (7) is not accurate enough due to two weaknesses: The first one is the lack of considering the nonuniformity of the error distribution in 3-D space, i.e., distant points usually introduce bigger errors. The other one is treating each point pair equally, without considering the different localization uncertainties of reconstructed points.

### III. TLBBA ALGORITHM

#### A. First stage: two-view binocular iterative optimization

In consideration of the nonuniformity of error distribution, we reproject the 3D point’s position error \( \mathbf{e}_j \) to 2D images:

\[
\mathbf{e}'_j = F^{-1}(\mathbf{p}_{cj}) - F^{-1}(\mathbf{r} \mathbf{p}_{pj} + \mathbf{t}) = \begin{bmatrix} \hat{\mathbf{p}}_{cl} \\ \hat{\mathbf{p}}_{cr} \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{cl} \\ \mathbf{p}_{cr} \end{bmatrix}
\]

where \( \Phi^{-1} \) is a function reprojecting \( \mathbf{p}_{pj} \) to both current and previous binocular image pairs. To find the optimal solution, we optimize the motion and structure in an interleaved mode. And the cost function optimizing \( \mathbf{r}^* \) and \( \mathbf{t}^* \) can be established in a similar way:

\[
\left( \mathbf{r}^*, \mathbf{t}^* \right) = \arg \min_{\mathbf{r}, \mathbf{t}} \sum_{j=1}^{n} \left\| \mathbf{e}'_j \right\|^2
\]

Equations (10) and (11) are performed iteratively to increasingly improve the accuracy of estimation. This stage of optimization can converge very quickly in practice, usually within about 3 or 4 iterations. This method is somewhat like the resection-intersection method or named interleaved bundle adjustment [18], which works in monocular and multiple-view.
circumstance. Our new method is distinct with the binocular and two-view characteristic. It is notable that the traditional monocular resection-intersection method cannot work between only two frames, because the constraint is not enough and estimation error may be enlarged. The binocular version of cost function gives much stronger constraint to 3D points, which is the key improvement of binocular structure. From (4), we can easily get the reprojections’ coordinates:

\[
\hat{x}_i = u_0 + \frac{P_x f_x}{P_z} \\
\hat{y}_i = v_0 + \frac{P_y f_y}{P_z} \\
\hat{z}_i = u_0 - \frac{(B - P_x f_x)}{P_z} = \hat{x}_i - \frac{B f_x}{P_z}
\]

As (12) shows, in monocular model, measurements \( x_i \) and \( y_i \) can only constrain the ratio \( P_x : P_y : P_z \), then the optimized position of \( P \) may freely move along the ray \( C_p p_i \) connecting camera centre and the projection (see Fig. 2). The more projections the \( P \) has, the stronger the \( P \) constrained. So monocular BA works well in multiple view reconstruction, but cannot be used between only two views where there is insufficient constraint. Typically, in a weak-texture environment, most features may live only 2 or 3 views, and then the multiple-view problem will degrade to a two-view problem. In this case, monocular LBA is prone to suffer much larger errors and even converge to a wrong solution. In contrast, binocular LBA never worries about this, because additional measurement \( x_r \) closely constrain \( P_z \) together with the help of \( x_i \). \( P_z \) cannot vary proportional to \( P_x \), and \( P \) is fixed as the intersection point of \( C_p p_i \) and \( C_p p_r \).

Equation (13) indicates that if \( P_z \) leaves the correct position, \( \hat{x}_i \) or \( \hat{z}_i \) must move away from its corresponding measurement and produce a big reprojection error.

B. Second stage: weighted binocular LBA

LBA can achieve similar precision as global BA, while keeping the computation time at a pretty low level. LBA uses a sliding window to limit the number of parameters to be adjusted. After accumulating \( R \), \( t \) to global rotation \( R \) and translation \( T \), 3D point \( P \) will also has its global position \( P^g \). If the window size is \( M \), and \( N \) points are observed, the cost function of weighted binocular LBA is expressed as:

\[
(R^*, T^*) = \arg \min \sum_{i=1}^{N} \sum_{j=1}^{M} \epsilon^T \Sigma^{-1} \epsilon
\]

\[
\epsilon = [p_{lji} - F^{-1}(R^*, T^*, P^g)] - [p_{rji} - \hat{p}_{rji}]
\]

IV. EXPERIMENTAL RESULT

We test the proposed two-stage local binocular bundle adjustment VO in our campus scene and Karlsruhe data set. In our campus experiments, images are captured by Bumble Bee2 camera, Pioneer 3-AT (Fig. 3(a)) is used as vehicle platform for short distance experiments while an outdoor electric vehicle (Fig. 3(b)) for the long distance. Especially, the estimated trajectories of short distance are compared to the ground truth provided by SOKKIA SRX1 total station surveying instrument (Fig. 3(c)). The experimental results will show the robustness and accuracy of our TLBBA algorithm in the challenging environment.

A. Ground truth comparison

In the short distance experiments, the estimated trajectory with our improved binocular visual odometry is compared to a total station whose measurement accuracy is at 2mm level. In the Karlsruhe data set, an OXTS RT 3000 GPS/IMU system is used to provide the ground truth, whose accuracy is 0.05~2 m. Not only path comparison, but also 3D point-to-point localization error will be shown as evaluations of our system.

If \( \hat{L}_m \) is the estimated location of the vehicle at frame \( m \), and \( L_m \) is the location from total station, we define RMS 3D localization error as:

\[
E_{lm} = \|\hat{L}_m - L_m\|
\]

and accumulated distance error as:

\[
E_{dm} = \sum_{i=1}^{m} (\|\hat{L}_i - \hat{L}_{i-1}\| - \|L_i - L_{i-1}\|)
\]
B. Short distance experiments

This experiment is about 96m long, conducted at 10 AM, a summer hot day with strong lighting. The trees produced large area shadow, and the grass appears to be over-exposed due to reflection of light (Fig. 4(a)). The pitch angle of vehicle may change suddenly caused by the uneven lawn surface (Fig. 4(b)). We see the estimated trajectory of vehicle as Fig. 4(c). From this view, we can see the red points suffer large errors, especially in the vertical direction. These errors are mainly introduced by the inaccurate 3D reconstructions. If the disparity of projections on binocular images is small, then the reconstructed 3D point has large uncertainty due to the discrete resolution. And in some cases, the nearer points are hardly tracked because of big changes in the pitch angle, and the distant 3D points reconstructed with larger uncertainties make the motion estimation inaccurate. In contrast, the blue points distribute much close to the ground truth.

Then we look at the RMS 3D error in Fig. 5(a), it is obvious that our TLBBA significantly improve the VO’s localization accuracy. Someone may notice that this error curve fluctuates significantly. If local incremental motion estimation’s error drops, there may be two reasons: using global information or introducing reversed error (two wrongs make a right). To avoid interruptions as the later situation, we use the accumulated distance error (defined in (17)) to help evaluate the algorithms, which is more convicive to scale estimation. In Fig. 5(b), the green line dropping closely to red line, which means the traditional LBA is helpless to accumulated distance error, and even make it worse after frame 250. On the other hand, the TLBBA can reduce this error significantly for its strong constrain to 3D information and reliable scale estimation. In Table 1, we give the numerical results, and we can see that the TLBBA improves the accuracy dramatically with only comparable computation cost in contrast to traditional LBA.

C. Long distance experiments

Since the distance of this experiment is much longer (about 1019m), the total station cannot provide the ground truth, and GPS is also unavailable due to the shades of trees along the road. So we compare the experimental results with the map from Google Earth, which also provides localization information with high precision. As illustrated in Fig. 6, our VO system works well facing complex environment.

D. Experimental results of Karlsruhe data set

The Karlsruhe sequences are grayscale stereo sequences taken from a moving platform driving through a mid-size city. In [7] and [19], the author reported their experimental results with 2.2% average translation error at about 25Hz. In contrast, our method only suffers 1% maximum error at 40Hz. We show the results of “0010 sequences” (about 430m) as Fig. 7.

<table>
<thead>
<tr>
<th>Performance</th>
<th>Optimization method</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS 3D(m)</td>
<td>Without optimization</td>
</tr>
<tr>
<td></td>
<td>1.11 (1.15%)</td>
</tr>
<tr>
<td>accumulated distance error(m)</td>
<td>-2.57</td>
</tr>
<tr>
<td>time for motion estimation(s)</td>
<td>2.53</td>
</tr>
</tbody>
</table>
V. CONCLUSION

We presented a two-stage local binocular bundle adjustment (TLBBA) method taking advantages of binocular structure and feature uncertainty weighting. TLBBA improves the VO's accuracy dramatically with only comparable computational cost. With the acceleration of GPU and parallel processing, the new improved VO can estimate the vehicle’s motion at 35–40 frames per second with the maximum RMS 3D localization error less than 1%. Looking into the future, we are going to research on improving the robustness of VO under complex dynamic environments and integrate it into off-road autonomous vehicles to enhance the localization ability.

REFERENCES


