A Multi-Sensor Information Fusion Approach for Efficient 3D Reconstruction in Smart Phone

Apurbaa Mallik¹, Brojeshwar Bhowmick² and Shahnawaz Alam³
Tata Consultancy Services, Kolkata, India
¹apurbaa.mallik@tcs.com, ²b.bhowmick@tcs.com, ³shahnawaz.alam@tcs.com

Abstract—Despite the recent advancements, camera pose estimation using smart phone’s sensor data is still error-prone due to various environmental noise and variability of the force applied on the phone during data acquisition. Here, we propose a novel framework to mitigate these drawbacks of camera pose estimation using various sensor’s data. The directional information of accelerometer is utilized to obtain reliable features for position estimation, rather than using its magnitude. Moreover, a multi-sensor data fusion approach is followed for robust rotation estimation. The proposed framework improves the epipolar geometry estimation and produces accurate stereo-correspondence points. This in turn reduces the reprojection error and enhances the quality of 3D reconstruction. Furthermore, the proposed framework utilizes only the CPU of the phone, instead of GPU and takes around 2-3 secs for processing an image. Extensive experimental results show supportive evidence of the effectiveness of the proposed framework in pose estimation problem.

Keywords: 3D reconstruction, sensor data, pose estimation

1. Introduction

One of the fundamental problem in computer vision is to reconstruct a complete 3D scene from a set of multi-view 2D images. Multi-view 3D reconstruction systems have found applications in various domains including robotics, architecture, archaeology, surveillance, human computer interaction and entertainment industry etc. In this regard, the primary challenge is to find way of effectively fusing information from multiple views without a prior knowing the detailed calibration information and minimizing human intervention. In the last few decades, dense 3D reconstruction from multi-view has been explored widely by various researchers and numerous approaches have been proposed in the literature. One of the most important step in estimation of the 3D geometry of an object is obtaining the camera parameters.

Conventionally, camera parameters are estimated using image correspondence to get an accurate 3D model. Various schemes have been proposed by researchers to build 3D structure in near real time using image information [1],[2],[3]. Several industrial software such as Patch Based Multi-View Stereo [4] also provides different solutions to generate dense 3D model of a scene from multi-view images. But, the image based pose estimation schemes require high computation time. Moreover, presence of homogeneous texture and varying lighting condition in the scene may lead to failure of the image based 3D reconstruction scheme. Using cloud based 3D reconstruction framework described in [5], one can exploit the computation power of the server to generate the 3D model. But, it is not a convenient solution to the general users as access to a high processing servers is limited.

Nowadays, smart mobile (both high and low end) devices are equipped with build-in sensors like accelerometer, gyroscope and magnetometer which provide motion and orientation information. These additional sensor’s data have proven to provide computationally efficient 3D reconstruction framework [6],[7],[8],[9]. Project Tango [6] uses these sensor information in their mobile device to model the 3D world. But, their mobile device is equipped with additional hardware such as motion tracking camera and integrated depth sensing tools. Therefore, it increases the infrastructure cost and is not easily available to the common users. Pan et al. have developed a 3D model on mobile phone which could not provide adequate structural information due to the sparse nature of the model [7]. In [8], Tanskanen et al. have utilized smart phone sensor’s data along with image information to calculate accurate motion and orientation parameters for dense 3D reconstruction. But, usage of image information and epipolar geometry [10] increases the complexity of the reconstruction algorithm. For outlier filtering, they have also utilized computationally expensive methods like 5-point algorithm [11] and random sample consensus scheme [12] in their proposed algorithm. Moreover, they have used mobile GPU to achieve real time computation. Recently, Brojeshwar et al. have proposed a framework for real time 3D reconstruction, where mobile sensor’s data is solely used for camera parameter estimation [9]. But, in practical scenarios smart phone sensor’s data may get corrupted due to various environmental and sensor effects like magnetic vibrations, temperature etc [13]. Therefore, in most of the cases it becomes very difficult to accurately and precisely estimate the camera parameters.

Although, the magnitude of acceleration is greatly affected due to variability in applied force on the phone but the direction of movement is relatively robust. Keeping this in mind, we propose a robust camera pose estimation method leveraging the direction information obtained
from accelerometer values. The position estimation using accelerometer direction is inspired from [14]. Along with position estimation, the rotation of camera is also improved using fusion of accelerometer, magnetometer and gyroscope data. In this regard, the main contributions of the proposed framework can be enumerated as below-

- Use of directional component of noisy accelerometer values to compute camera position accurately. We show that, even in presence of noise in absolute value of accelerometer readings, the direction is a reliable statistic to compute position.
- Use of multi-sensor’s data fusion to compute camera rotation robustly.
- Robust computation of position and rotation improves the estimation of epipolar geometry which in turn produce better geometric filtered point correspondence, thus improving the quality of reconstruction.
- The proposed 3D reconstruction framework executes in real-time on CPU of the phone. Therefore, it can run on inexpensive smart phones which may be devoid or have low end GPU and is not capable of executing high computational activities.

The rest of the paper is organized as follows : An overview of the proposed framework is presented in Section 2. Section 3 describes the data acquisition procedure of the proposed system. The novel camera calibration methodology based on sensor’s data is explained in Section 4. The final 3D model estimation scheme is described in Section 5. Section 6, contains the experimental results and conclusion is drawn in Section 7.

2. Proposed Framework

In the proposed framework, a robust and computationally efficient camera pose estimation method based on relatively stable features of various sensor’s data is described. The main stages of the 3D reconstruction pipeline are as follows-

- Data acquisition
- Camera calibration procedure
- 3D model estimation

Fig. 1, shows our 3D reconstruction pipeline. The user captures the sensor’s data and images using a mobile data capture application. These acquired data are processed using the proposed 3D reconstruction framework to generate the dense 3D model in real-time.

3. Data Acquisition

The user moves their smart phone to get multiple view images of the subject. Simultaneously, data from different mobile sensors such as accelerometer, gyroscope and magnetometer is also captured. With respect to the mobile coordinate system, the accelerometer and gyroscope provides the linear acceleration \(\text{m/sec}^2\) and the angular velocity \(\text{rad/sec}\) along X-axis, Y-axis and Z-axis, respectively. The sensor’s data is used to compute the camera pose with respect to the world coordinate system. The internal clocks of the mobile sensors are time synchronized to obtain uniform data.

4. Camera Calibration Procedure

Camera calibration is of two types - internal and external calibration. Internal calibration estimates the focal length and principle point of a camera. The internal parameter remains the same for a particular resolution of an image of a camera. Therefore, the internal parameter is calibrated only once for a particular device. The external calibration estimates the rotation and position of the camera with respect to world coordinate system.

4.1 Rotation from Smart Phone Sensor Data

Orientation readings of the phone can be obtained by various combination of sensor’s data. Here, orientation can be represented using the Euler angles — azimuth angle (rotation of XY plane), row angle (rotation of XZ plane) and pitch angle (rotation of YZ plane). Using a combination of accelerometer and magnetometer data, we can estimate the orientation readings of the phone \(o^{am}\). Similarly, gyroscope data also provides the orientation of the phone \(o^g\) [13]. However, both \(o^{am}\) and \(o^g\) are prone to various noise. In presence of magnetic vibrations, the signal exhibits short interval variations and diverges from its true value. Therefore, in the proposed scheme low pass filter is applied on \(o^{am}\) to obtain a smoother signal. Though \(o^g\) does not get affected by magnetic field, but it is influenced by constant gyro bias and calibration noise. This is known as the gyro drift. We apply high pass filter on \(o^g\), to preserve the high frequency component of the signal and nullify the effect of gyro drift. Similar to [15], for every \(i^{th}\) sensor reading, we calculate the fused orientation parameter by combining the filtered \(o^{am}\) and \(o^g\) as,

\[
o_i = \lambda o^{am}_i + (1-\lambda) o^g_i
\]

where \(\lambda\) is a tuning parameter and \(o_i\) is the \(i^{th}\) reading of the fused orientation of the phone. From every sample of \(o_i\),
fused rotation matrix $R$ is estimated and used for calculation of image positions.

Fig. 2 shows a graph in which the azimuth angle readings of $\mathbf{o}$ (in red colour), $\mathbf{o}^{\text{rm}}$ (in blue colour) and $\mathbf{g}$ (in green colour) are presented. It is evident from Fig. 2, the azimuth angle of $\mathbf{o}$ is closest to the ground truth. Similar results are obtained using the pitch and row angles of the orientation parameters. In Table 1, we have compared the mean value of all orientation parameters $\mathbf{o}$, $\mathbf{o}^{\text{rm}}$ and $\mathbf{g}$ in static condition where $i = 1, 2, \ldots, 1700$. From both Fig. 2 and Table 1, it is evident that $\mathbf{o}$ provides higher accuracy and is closer to the ground truth than $\mathbf{o}^{\text{rm}}$ and $\mathbf{g}$. The accuracy obtained using this filtering reduces reprojection error in the triangulation stage as illustrated in Section 6.

Table 1: Comparative result of $\text{mean}(\mathbf{o})$, $\text{mean}(\mathbf{o}^{\text{rm}})$ and $\text{mean}(\mathbf{g})$ with respect to ground truth (GT) in degree (deg).

<table>
<thead>
<tr>
<th>GT (deg)</th>
<th>$\text{mean}(\mathbf{o})$ (deg)</th>
<th>$\text{mean}(\mathbf{o}^{\text{rm}})$ (deg)</th>
<th>$\text{mean}(\mathbf{g})$ (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.0,0.0</td>
<td>[15.6,0.0,0.1]</td>
<td>[17.3,0.4,1.6]</td>
<td>[9.9,-1.9,8.6]</td>
</tr>
<tr>
<td>30.0,0.0</td>
<td>[30.3,0.0,-0.1]</td>
<td>[32.8,0.8,1.7]</td>
<td>[26.3,-4.3,-8.5]</td>
</tr>
<tr>
<td>45.0,0.0</td>
<td>[44.2,0.1,0.2]</td>
<td>[46.3,0.9,1.7]</td>
<td>[41.4,-4.0,-7.4]</td>
</tr>
<tr>
<td>90.0,0.0</td>
<td>[88.1,-0.3,0.2]</td>
<td>[93.5,1.5,1.7]</td>
<td>[85.5,-6.3,-7.7]</td>
</tr>
</tbody>
</table>

4.2 Position from Smart Phone Sensor Data

The accelerometer of the smart phone provides the raw acceleration readings $\mathbf{a}^{\text{raw}}$ in m/sec$^2$. But, environmental and sensor conditions affect the accelerometer data and make it noisy [13]. This ultimately results in erroneous position estimation. Some of these noises which influence the accelerometer data are static bias and random noise.

Theoretically, in static condition acceleration should be zero. But, due to the presence of static bias $\mathbf{B}$, the accelerometer data deviates from its ideal value and in turn gives an error in estimated position. Therefore, we obtain the bias corrected acceleration value $\mathbf{a}^B$, by subtracting $\mathbf{B}$ from $\mathbf{a}^{\text{raw}}$. Static bias $\mathbf{B}$ is estimated by averaging $\mathbf{a}^{\text{raw}}$ in static condition [13]. Fig. 3 shows $\mathbf{a}^{\text{raw}}$ and $\mathbf{a}^B$ (represented using red and blue lines respectively) along with their estimated positions. The positions estimated from $\mathbf{a}^B$ (represented using green circles) exhibit higher accuracy than the positions estimated using $\mathbf{a}^{\text{raw}}$ (marked by red circles). Furthermore, computing image positions from accelerometer value using Newton’s law of motion produce erroneous estimate as shown in Fig 3. Therefore, the accelerometer data needs further refinement.

While acquiring data, the user’s hands can shake. It gives an illusion of movement in the accelerometer data, leading to wrong position estimate. Therefore, the region of active mobile movement ($\text{AMM}$) is determined from $\mathbf{a}^B$ to estimate the position correctly. While the phone is in static condition, we calculate the standard deviation of the accelerometer data $\text{B}^\text{std} = \text{std}(\mathbf{a}_i^{\text{raw}} - \mathbf{g}_i^{\text{raw}})$. Here, $\text{std}()$ represents the standard deviation function and $\mathbf{g}_i^{\text{raw}}$ is the gravitational projection of $\mathbf{a}_i^{\text{raw}}$ for the $i$th sensor’s data in static condition. While user is capturing the data, the region of $\text{AMM}$ is identified as,

$$\text{AMM} = \begin{cases} 1 & \text{if, } 3\text{B}^\text{std} < \text{mean}(\mathbf{a}_i^B - \mathbf{g}_i^B) < -3\text{B}^\text{std} \\ 0 & \text{otherwise} \end{cases}$$

where, $\mathbf{g}_i^B$ is the gravitational projection of $\mathbf{a}_i^B$ for the $i$th sensor’s data.

If “$\text{AMM} = 1$”, the phone is said to be in motion; otherwise it is in static condition. In Fig. 4(a), the red box marks the region of $\text{AMM}$ on $\mathbf{a}^B$ (denoted by green line). It is evident from Fig. 4(b), that the position estimated using the region of $\text{AMM}$ (indicated by green circles) is more precise than the positions estimated without using the region of $\text{AMM}$ (represented using red circles).

For every sample of sensor’s data where the mobile is in motion, the acceleration reading is used to estimate the image position. Assuming, the time difference between two readings of accelerometer data in motion be $\Delta t$ where, $\Delta t = t_{i-1} - t_i$. Using Newton’s law of motion, we calculate the displacement of the phone from its previous position in $t_{i-1}$ for the $i$th sample of sensor’s data reading as,

$$\mathbf{s}_i = \mathbf{u}_i \Delta t + 0.5R_i(\mathbf{a}_i^B - \mathbf{g}_i^B - \mathbf{B})\Delta t^2$$
where $u_i$ is the initial velocity and $R_i$ is the fused rotation matrix (as discussed in section 4.1) for $i^{th}$ sensor’s data. We use eq. (3) to estimate the relative position between two consecutive images. This process is similar to the relative position estimation scheme described in [9].

Fig. 5, shows a view-graph where its vertices represents the initial relative image positions. Here, $d$, $e$ and $f$ are the images taken in time $t_d$, $t_e$, and $t_f$ where $t_d < t_e < t_f$. We use eq. (3) to estimate the relative image position between $d$ and $e$ as $c_{de}$. Similarly, we estimate $c_{ef}$. While calculating the relative image position, the former image is taken as the origin. Ideally, the acceleration at the former image should be zero. Therefore, we calculate $c_{de}$ assuming $a_d = 0$ where $a_d$ is the acceleration at $d$. But, as the accelerometer provides noisy data, the acceleration at the image positions may not be zero. Therefore, $c_{de}$ is calculated using non zero acceleration values at the intermediate image positions that is, $a_e \neq 0$. In the same way, $c_{df}$ is estimated by taking $a_e \neq 0$ and $a_f \neq 0$. This concept is extended to estimated the global image positions $c_d$, $c_e$ and $c_f$ that is, image positions with respect to the global origin. In this paper, the position of the first image of the dataset is considered as the global origin.

Due to the presence of various random noise, the sensor’s data gets corrupted and as a result initial image positions become imprecise. Therefore, there is a need to optimize the initial image positions. In [9], the image positions are optimized using iterative reweighted least square position av-
eraging. Though position estimation methodology described in [9] gives good accuracy on datasets with smooth user transitions. But, we may not get precise position estimate using this method due to variability of force applied on the phone and environmental noise during data acquisition. The primary reason behind this is the force applied on the phone effects the accelerometer values. It is experimentally seen that in such conditions, the magnitude of the acceleration data gets noisy, but the direction of the movement can be reliably estimated. Using Newton’s law of motion as given in eq. (3), the image positions are computed from the filtered accelerometer data. Then, we calculate the pair-wise unit direction vector between the mobile position at $t_i$ and $t_j$, $\hat{e}_{ij}$ where $j = i + 1$ as,

$$\hat{e}_{ij} = \frac{\epsilon_{ij}}{||\epsilon_{ij}||_2} \quad (4)$$

This concept can be extended for calculating the unit direction vectors between the camera positions. Therefore, for every pair of image positions $d$ and $e$, we get a unit directional vector, $\hat{c}_{de}$. Then, similar to [14], the global position for those pairwise direction can be obtained using $\rho\left(\frac{|c_i-c_e|}{c_{de}}\right)$ where, $\rho()$ is the pseudo huber loss function [10] and $\hat{c}_{de}$ is unit directional vector between image position $d$ and $e$. The earlier estimated initial global positions $c_d$ and $c_e$, is provided as initialization parameter to the optimization function.

In Table 2, using root mean square error we compare the accuracy of relative image positions estimations based on [9], our initial position estimation method and our direction based position estimation method respectively. The data is collected by moving the phone in a planer surface. We consider the euclidean distance between two image positions as the ground truth. In most of the cases, the error using the direction based position estimation method has reduced significantly compared to other two methods. This indicates improvement in the accuracy of our position estimation method. The usefulness of the optimized image positions are evaluated through triangulation and the obtained results are shown in Fig. 6. Fig. 6(a) is the original image. Fig. 6(b) is the triangulation result obtained from optimized image positions as against Fig. 6(b), where image positions are estimated using [9]. It is evident in Fig. 6(c), image positions are not accurate due to which the structure estimated is imperfect as marked in red. On the other hand, in Fig. 6(b) the human is clearly reconstructed.

## 5. 3D Model Estimation

Stereo-correspondence estimation is one of the essential stage in 3D reconstruction framework. A low computation stereo-correspondence finding algorithm using a combination of gradient and colour information [9] is used. The proposed precise camera pose computation yields accurate epipolar geometry [10] which leads to better stereo-correspondence estimation. This improves the quality of 3D reconstruction and reduces the reprojection error [10]. The results given in Fig. 10 and Table 4 supports our claim. A detailed explanation is provided in Section 6.Using the estimated correspondences and global camera parameters, the initial 3D points by an established method called triangulation [16] is generated. In the final stage of our framework, using well known bundle adjustment algorithm, simultaneously the dense point cloud and cameras parameters [17] is optimized.

6. Experimental Results

For testing the proposed 3D reconstruction method, LG Nexus 5 smart phone is used. It has Quad-core 2.3 GHz Krait CPU and 2GB RAM. The sensor’s data is recorded at a frequency of 50 Hz and the images are captured with resolution 640x480. Extensive experiments are carried out for qualitative and quantitative evaluation of the proposed framework. Quantitative evaluation are performed based on the analysis of reprojection error [10] and time requirement for camera pose estimation. Whereas, qualitative experiments include assessment of visual quality of the estimated 3D model by testing the proposed framework on different datasets of varying shape and size in general illumination.
In image based epipolar geometry estimation, computation time is dependent on number of stereo-correspondences. Table 3 shows a typical time requirement for computing fundamental matrix between pair of images using 8-point algorithm [10]. Instead of that, using smart-phone sensors for computing epipolar geometry is efficient as it is independent of number of stereo-correspondence points. Also, the computation time is very less. In Table 3, the epipolar geometry computation time is 0.0193 sec using sensor’s data. As a result, for 1,00,000 stereo correspondence points the proposed epipolar geometry estimation method reduces the computation burden by approximately 701 times.

Table 3: Time requirement for epipolar geometry estimation with varying number of stereo-correspondence points.

<table>
<thead>
<tr>
<th>Number of stereo corresponding points</th>
<th>Epipolar geometry estimation (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,00,000</td>
<td>13.3253</td>
</tr>
<tr>
<td>1,25,000</td>
<td>15.9598</td>
</tr>
<tr>
<td>2,00,000</td>
<td>26.2119</td>
</tr>
<tr>
<td>5,00,000</td>
<td>62.2099</td>
</tr>
</tbody>
</table>

The accuracy of the proposed camera pose algorithm is evaluated using estimated reprojection error in the triangulation stage. Higher accuracy of the camera pose indicates lower reprojection error. Different combinations of rotations (R) and positions (P) calibrated using [9] and the proposed scheme are used to evaluate the camera pose estimation performance. These combinations are briefly described below:

- **S1** - Both R and P estimated using [9]
- **S2** - R estimated using [9] and P estimated using our method
- **S3** - R estimated using our method and P estimated using [9]
- **S4** - Both R and P estimated using our method.

In Table 4, the mean reprojection error using combination **S1, S2, S3 and S4** with varying number of 3D points is shown. The reduction of reprojection error in combination **S2, S3 and S4** against **S1**, indicates improvement of the proposed estimated camera pose with respect to the camera pose estimated using [9]. Apart from camera pose, the reprojection error is dependent on stereo-correspondences. We have used a simplistic stereo-correspondence finding method to have a real-time system. Therefore, in presence of homogeneous texture and varying lighting condition we may get noisy output. In these cases, despite of having accurate camera pose, the reprojection error is not reduced to a large extend. Usage of high computation stereo-correspondence finding algorithm will further reduce the reprojection error at the cost of increase in system execution time.

Table 4: Comparison of reprojection error using the camera pose combination – **S1, S2, S3 and S4**

<table>
<thead>
<tr>
<th>No. of 3D points</th>
<th>S1 (pixel)</th>
<th>S2 (pixel)</th>
<th>S3 (pixel)</th>
<th>S4 (pixel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>101994</td>
<td>10.74</td>
<td>7.23</td>
<td>7.22</td>
<td>5.08</td>
</tr>
<tr>
<td>112016</td>
<td>9.12</td>
<td>7.02</td>
<td>8.43</td>
<td>6.90</td>
</tr>
<tr>
<td>100124</td>
<td>18.79</td>
<td>14.54</td>
<td>14.23</td>
<td>13.37</td>
</tr>
<tr>
<td>84495</td>
<td>35.92</td>
<td>31.62</td>
<td>30.29</td>
<td>29.64</td>
</tr>
</tbody>
</table>

Different stages of our 3D reconstruction algorithm are executed parallely on the mobile CPU. The reconstruction stages takes around 2-3 seconds for every image. The stereo-correspondence algorithm takes less than a second and can be estimated while the user captures the data. The computation time for our camera pose estimation method is in the order of microseconds. The 3D reconstruction pipeline stages which require considerable amount of time are triangulation and global bundle optimization. The time requirement for both the stages is similar to [9].

The proposed framework is tested on different indoor and outdoor datasets of varying shape and size in general illumination. Fig. 7, shows a reconstructed dense 3D model of an outdoor structure of height 1.76 meters using 6 images. The 3D model of the objects in Fig. 8 of height 3.04 metres and Fig. 9 of height 0.25 meters are reconstructed using 8 images and 5 images respectively taken in indoor condition.

![Fig. 7: (a) is an outdoor image of an object of height 1.76 meters; (b) and (c) are the front and side view the 3D model respectively.](image)

The proposed method is compared with the scheme described in [9] and the comparisons are given in Fig. 10 using 3D model outputs from triangulation stage. The dense 3D models shown in Fig. 10(b) and Fig. 10(c) are the triangulation outputs where camera pose is estimated using [9] and our method respectively. Due to imprecise camera pose, the estimated 3D structure in Fig. 10(c) is imperfect as marked in red. Whereas, Fig. 10(b) gives comparatively accurate model because of better camera calibrations and stereo-correspondences. As, evident from Fig. 10(d-f), the accuracy of the final 3D model increases after our final stage of global bundle optimization.

![Fig. 10: (d-f) are the front and side view the 3D model respectively.](image)
7. Conclusion

In this paper, a computationally efficient on-device 3D reconstruction framework which runs on the CPU of the phone is proposed. The framework comprises of a robust and computationally effective camera pose estimation method based on relatively stable features of the smart phone sensor’s data. The direction based position estimation method works robustly in practical scenarios where sensor’s data is corrupted due to variability of force applied on the phone or environment noise. Rotation computation using fusion of multi-sensor’s data also provides enhanced result. The estimated precise camera parameter improves the epipolar geometry which in turn gives accurate stereo-correspondence points. This reduces the reprojection error and improves the quality of 3D reconstruction. The proposed novel 3D reconstruction scheme can be extended in other applications like augmented reality, animation, robot vision, communication and bio-metric authentication.

References