

Relative Pose of IMU-Camera Calibration Based on BP Network

Shasha Guo¹, Jing Xu¹, Ming Fang¹ & Ying Tian¹

¹ School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China

Correspondence: Ming Fang, School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China. Tel: 86-431-8558-3331. E-mail: fangming@cust.edu.cn

Received: August 14, 2017

Accepted: August 26, 2017 Online Published: September 9, 2017

doi:10.5539/mas.v11n10p15

URL: <https://doi.org/10.5539/mas.v11n10p15>

Abstract

There are many applications of the combination of IMU (Inertial Measurements Unit) and camera in fields of electronic image stabilization, enhancement reality and navigation where camera-IMU relative pose calibration is one of the key technologies, which may effectively avoid the cases of insufficient feature points, unclear texture, blurred image, etc. In this paper, a new camera-IMU relative pose calibration method is proposed by establishing a BP neural network model. Thus we can obtain the transform from IMU inertial measurements to images and achieve camera-IMU relative pose calibration. The advantage of our method is the application of BP neural network using Levenberg-Marquardt algorithm, avoiding more complex calculations for the whole process. And it is convenient for the application of camera-IMU combination system. Meanwhile, nonlinearities and noises are compensated while training and the impact of gravity can be ignored. Our experimental results demonstrated that this method can achieve camera-IMU relative pose calibration and the accuracy of quaternion estimation has reached about 0.01.

Keywords: BP Network, camera-IMU, pose, calibration

1. Introduction

We can't get correct camera motion trajectory by a single visual sensor in cases that image features are scarce, the texture is not obvious, the image is blurred and illumination changes, etc., which may influence the effect of image stabilization and tracking. However, inertial sensor IMU can still provide estimate of their own attitude measurements and predictive information without considering the image and external scenes. Therefore, in many fields such as electronic image stabilization, enhancement reality, navigation etc., combination system of visual sensor and inertial sensor is used. They use IMU data to reflect the motion trajectory of the camera by estimating the rigid body transformation relationship between IMU and camera, among which camera-IMU relative pose calibration is very important.

Traditional camera-IMU relative pose calibration methods are mostly based on Extend Kalman Filter (EKF) (Li M. & Mourikis A I., 2000; Jia & Evans, 2007; Li M., Kim B H & Mourikis A I., 2013; Jia C. & Evans B L., 2014.) or unscented Kalman Filter (UKF) (SJ Julier & JK Uhlmann, 2004). And the motion estimation are mainly obtained according to image characteristics, and then predicted and updated according to IMU data while visual tracking is failed.

Jonathan Kelly (Kelly & Sukhatme, 2009) employs an unscented Kalman filter to estimate the relative pose of different sensors, which implemented based on hardware support devices. They have achieved camera-IMU calibration where the accuracy of their experimental translation error is 0.43cm, 0.23cm, 0.24cm in the x direction, the y direction and the z direction respectively, and the three angle error of Euler angle is all about 0.06 degree. Though the method is effective, the error accuracy is not high, external hardware support is needed as well.

BP network (Chen Ming, 2013) can be used to implement a mapping function from input to output, which has been proved to have the ability to implement any complex nonlinear mapping. So it is suitable for complex problems of internal mechanisms. Considering this, we propose a new algorithm based on BP neural network to achieve camera-IMU online calibration, framework is shown in Figure 1. While there are less feature points, input IMU data to trained model and the prediction quaternion $quat_{IMU}$ is $finalquat$ representing relative rotation between adjacent frames. While feature points are adequate, compare quaternion from images $quat_{img}$ and trained

model output $quat_{IMU}$, if errors exceed a certain threshold, feedback quaternion from images to trained model to update. Now we have implement camera-IMU offline and the comparison module and feedback will be implemented in the future..

The main contribution of our work is the model based on BP algorithm, which uses nonlinear least squares method--Levenberg-Marquardt (LM) (Mor, 1978) algorithm to convert time domain data to spatial domain data and keep their correspondence, and through which we can obtain accurate prediction, i.e. our method can achieve accurate camera-IMU calibration. Besides, calibrate relative pose IMU and camera by training, avoiding traditional complex calibration on industrial applications, for example, by extracting corners on calibration plate. And everyone without specialized knowledge can also use an IMU and apply to image stabilization.

The reminder of paper is organized as follows. In Section 2, we introduce our prior work on time synchronization in our lab. In Section 3 and Section 4, we introduce our algorithm and provide our experimental results and analyze the experimental results. Our results show excellent performance. And Section 5 is conclusion.

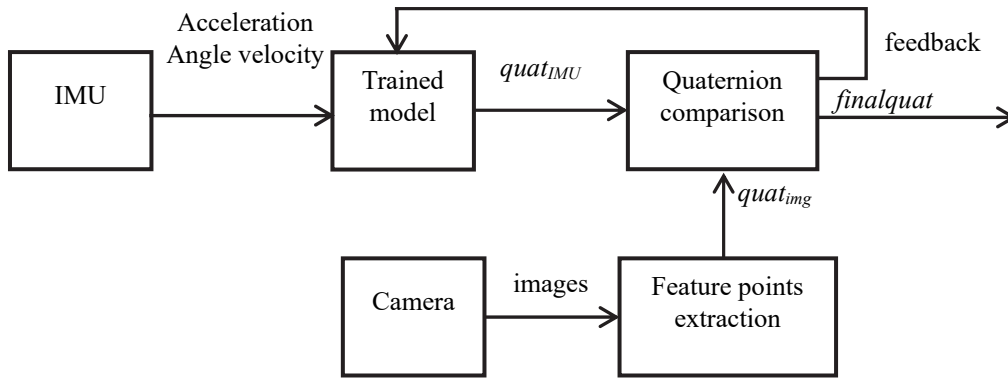


Figure 1. Framework of proposed calibration method

2. Prior Work

Given that the start time, sampling frequency and transmission delay are not the same (S. Gupta, U. Dalal & V.N. Mishra, 2015) between the visual sensors and the inertial sensor, data acquired at the same time is often asynchronous, so it is needed for time registration to ensure the matched relationship between different sensors. We employ Newton interpolation method to achieve the time synchronization (Tian Y & Fang M, 2016) between camera and IMU. First, extend their frequencies to a same frequency according to the minimum common multiples of the two sensor frequencies, use central difference method to interpolate for the set of extension points to ensure the acquisition frequency synchronization. After that through the minimum error matching of the data from the static to the motion part on the rigid fixed camera-IMU specified, and thus achieve time synchronization on the two datasets.

3. Algorithm

As the IMU acquisition frequency is much higher than the camera's acquisition frequency, there are multiple IMU data changes during two adjacent frames captured by a camera. Considering the IMU data changes may reflect the transform between two adjacent frames, that is, after a transformation, it is likely to obtain camera pose by multiple sets of IMU data in same time interval, regardless of translation, we propose a new method to calibrate the transform between camera and IMU based BP algorithm.

Fix the camera and IMU rigidly and begin to collect data at the same time. Calculate rotate matrix of the adjacent images and represent it by quaternion and take the quaternion as output, take the IMU data acquired during the same time interval including acceleration (A. Modh, M. Dabhi, L.N. Mishra & V.N. Mishra, 2015) and angular velocity as the input. Quaternion (Liu J F, 2004) is composed by a real unit and three imaginary units. Three-dimensional rotation can be represented as

$$q = q_0 + q_1 i_1 + q_2 i_2 + q_3 i_3 \quad (1)$$

Where q_0 is the real part of quaternion, q_1 , q_2 and q_3 are the three imaginary parts of quaternion, i_1 , i_2 and i_3 can

be interpreted as a rotation from the aspect of geometric.

For images, estimate the relative rotation between adjacent images by extracting ORB (ORiented Brief) feature points. Represent the relative rotation by unit quaternion and regard the quaternion training labels.

Then establish BP neural network model to obtain the relationship between IMU data and camera pose estimated by images. For hidden layer, we use tan-sigmoid as activation function to compress the data between -1 and 1, shown in equation (2). And then a linear output layer is added to the network.

$$\text{tansig}(n) = 2/(1+e^{-2n})-1 \quad (2)$$

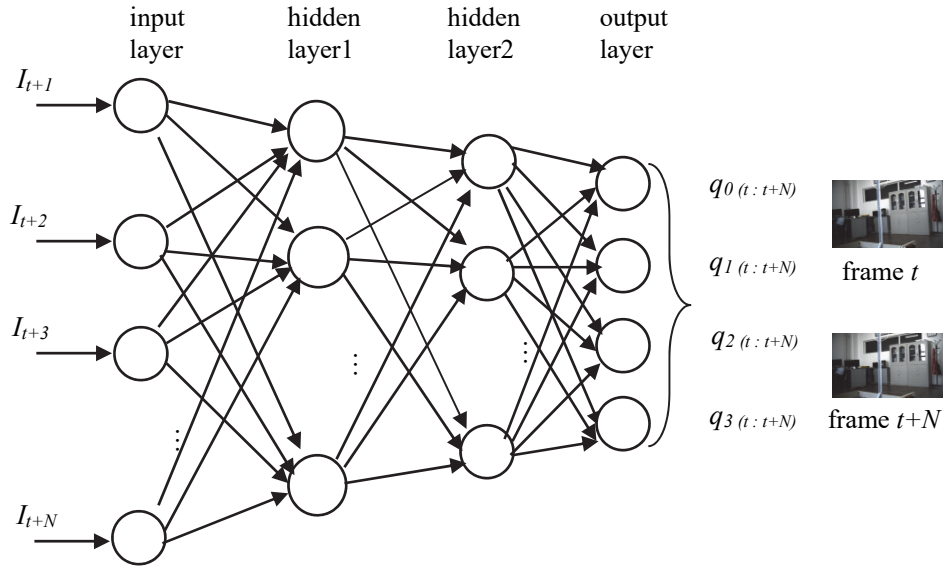


Figure 2. BP network model of the proposed method

The basic idea of BP algorithm is the signal positive transmission and error back propagation while learning. The learning rule of BP network is to use the steepest descent method to continuously adjust the weight and threshold of the network by back propagation to minimize the square error. BP neural network model topology includes input layer, hidden layer and output layer. The number of hidden layer is not sure. And we establish a BP network of three layers, as shown in Figure 2 to find the relationship F in equation (3) between IMU data and images. Quaternion represents the rotate matrix from frame t to frame $T+N$ and I_t is a six dimensional vector representing IMU inertial measurements data acquired at time of t , including three values of acceleration $[a_x, a_y, a_z]$ and three values of angle velocity $[w_x, w_y, w_z]$.

$$[q_0, q_1, q_2, q_3]_{(t:t+N)} = F(I_t : I_{t+N}) \quad (3)$$

$$I_t = [a_{xt}, a_{yt}, a_{zt}, w_{xt}, w_{yt}, w_{zt}] \quad (4)$$

Assume the frequency of camera is f_c and the frequency of IMU is f_i , the number of IMU data measured while camera captured per frame $N=f_i/f_c$.

Regard the three-layer BP neural network training as a numerical optimization, and the error surface is just a highly nonlinear function (L.N. Mishra, 2017; Deepmala, 2014) of the weights. Here, we employ Levenberg-Marquardt algorithm, i.e. with a gradient seeking maximum (minimum) value of the algorithm to find a set of optimal parameters vector to minimum value function, which compromises the Newton method and gradient method. In part of evaluating, we use Mean Squared Error (MSE) as evaluation to describe the accuracy of the prediction model. MSE is the mean expected square value of the differences between the estimated value and the true value from group. The smaller MSE is, the higher the accuracy of the prediction model is.

$$MSE = [\sum_{i=1}^n (true_i - predicted_i)^2] / n \quad (5)$$

In equation (5), i refers to groups number of data, $true_i$ here is quaternion obtained from frame i to frame $i+1$, $predicted_i$ is estimated value by IMU data of group i .

4. Experiments

4.1 Experimental Platform

Fix the IMU and the camera rigidly together to form a visual-inertial system, as shown in Figure 3. Then collect images and inertial measurements data at the same time synchronously based on our preliminary work.

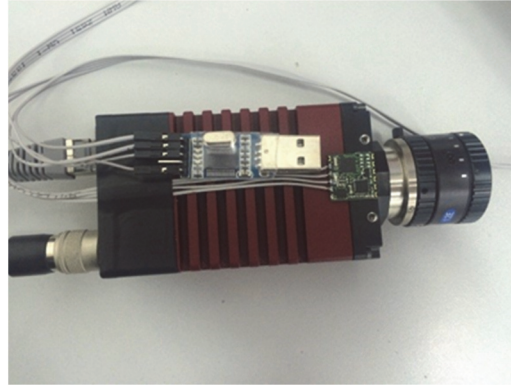


Figure 3. Combination system of IMU and camera

In our experiments, we use industrial camera GX1910 as visual sensor, the resolution is 1920*1080, and keep its acquisition frequency to be 10fps; for inertial sensor, we use MPU6050 which provides three-axis angular rate and linear acceleration measurements at 100 Hz, and the baud rate is set to 9600.

4.2 Experimental Dataset

Start the combination system of camera and IMU at the same time. We collect two data sets, one is fast sequence and the other is slow sequence. For the fast sequence, we shake the combination system intensely and quickly, thus IMU data during interval of two frames may change significantly. While the IMU data changes for per N group are not obvious, and even there are no changes.

From part 4.1, $N = f_i / f_c = 10$, so for each sequence, there are ten sets of IMU data including acceleration and angle velocity corresponding two images. Thus the training input data for each group is a 60 dimension vector. The training source data needed be acquired in scenes where there are clear and quantities of features points. Then optical flow method is used to extract the images feature points to calculate rotate matrix between adjacent images to generate training labels automatically, i.e., pose of the camera. For convenience, we use quaternion instead of rotate matrix, thus in fact the training output is a 4-dimension vector. We have captured 2001 frames image and 20000 sets of IMU data correspondingly for each sequence.

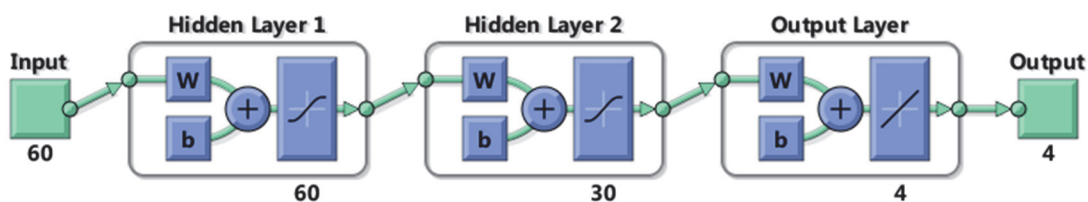


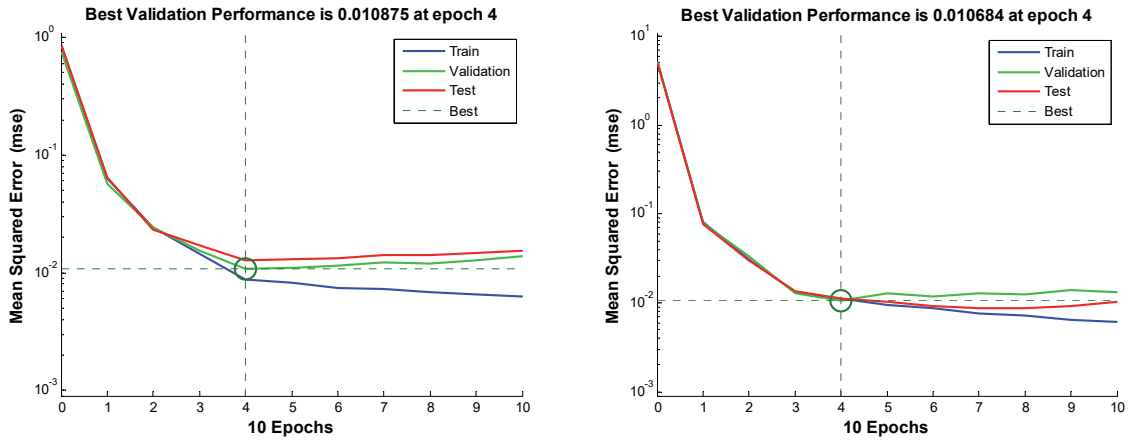
Figure 4. Network established in the experiment

Figure 4 is the three-layer network established in the experiment, there are sixty nodes in the first hidden layer, thirty nodes in the second hidden layer and four nodes in the output layer and each layer we use tan-sigmoid activation function. While training, Levenberg-Marquardt algorithm is used to get optimal parameters.

4.3 Experimental Results

Preprocess the IMU data and images and transform them to training input and output data needed. After preprocessing, we get 2000 sets of training source data for each sequence which are divided to two parts to train and test respectively in sequential. Among all the training source data, the proportion of training data and final test data is 80%, 20%. Training sequence is used to train model tune model. While training, input data is actually

divided to three parts, training data, validation data and test data to a percent of 4:1:1 in order to achieve optimal. Final test sequence is used to test the trained model. Figure 5 shows the convergence tendency of the error goal for both fast sequence and slow sequence.

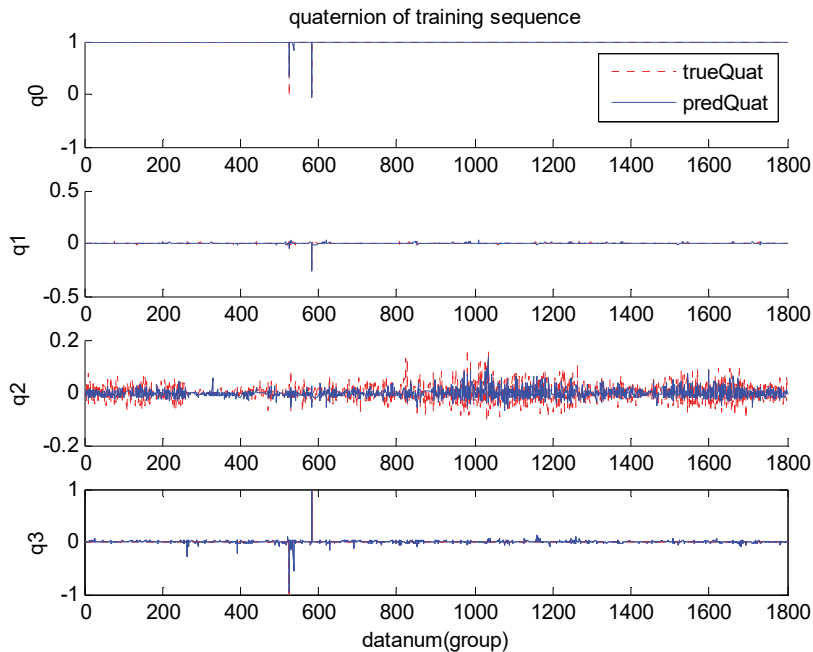


(a) Error curve for fast sequence

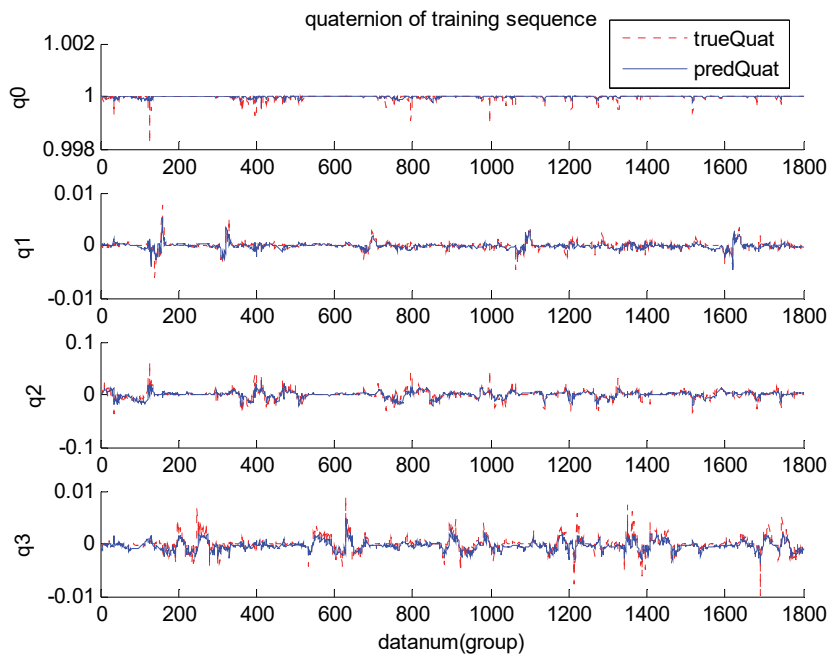
(b) Error curve for slow sequence

Figure 5. Error goal convergence tendency

Figure 5(a) shows that the optimal error goal for fast sequence is 0.010875 and Figure 5(b) shows the optimal error for slow sequence is 0.010684. And these results demonstrate that the precision of trained model for both fast and slow sequence is almost the same. Next, input training IMU data and test IMU data of both sequences to their own trained model respectively to predict. Figure 5(a) and Figure 5(b) show predicted quaternion and truth quaternion of training data for both fast sequence and slow sequence. Figure 6(a) and Figure 6(b) show predicted quaternion and truth quaternion of test data for both fast sequence and slow sequence. Here the true quaternion refer to quaternion obtained by extracting image feature points. In Figure 6 and Figure 7, horizontal axis notes data of each group in sequential, vertical axis notes value of quaternion respectively. Each group data includes a 4-dimension quaternion vector obtained by adjacent frames and a 60-dimension imu data vector, composed by 10 groups IMU data corresponding the two frames.

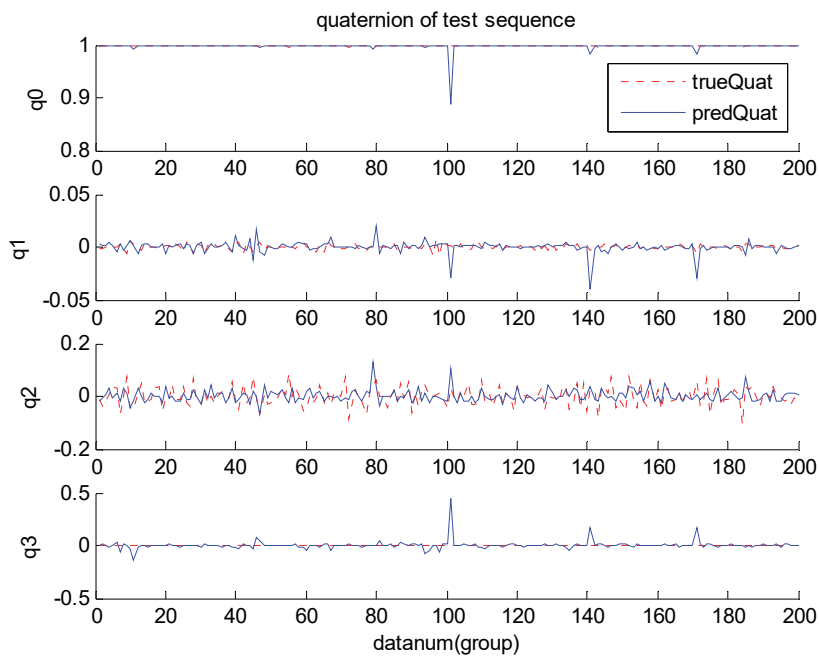


(a) Quaternion of training of fast sequence

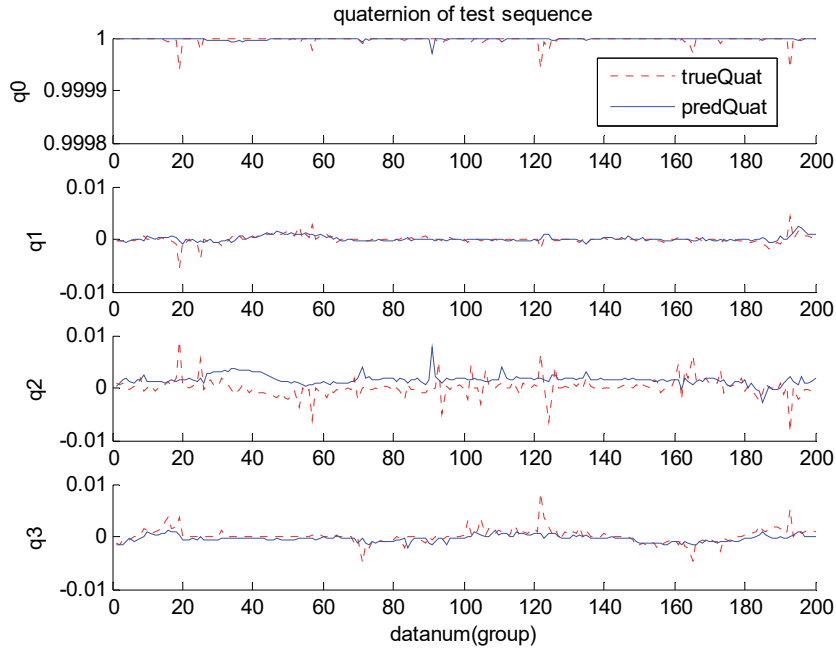


(b) Quaternion of training for slow sequence

Figure 6. Quaternion of training data



(a) Quaternion of test for fast sequence



(b) Quaternion of test for slow sequence

Figure 7. Quaternion of test data

By comparing Figure 6(a) and Figure 6(b), Figure 7(a) and Figure 7(b), we can know the method works better on slow sequence and the prediction quaternion can better reflect changes of true changes. A sudden change exceeding a certain threshold indicates a false predictor. For this problem, in practical application of visual-inertial combination system, the value can be replaced by measurement from images. And the threshold is determined by the error convergence of training model.

Table 1. Mean orientation error comparison

sequences	mean orientation error							
	training data				test data			
	q ₀	q ₁	q ₂	q ₃	q ₀	q ₁	q ₂	q ₃
fast sequence	0.0009	0.0038	0.0195	0.0123	0.0014	0.0029	0.0261	0.0144
slow sequence	0.0000	0.0004	0.0037	0.0007	0.0000	0.0004	0.0019	0.0008

For each data sequence, calculate mean orientation error of prediction and true quaternion obtained from images. Table 1 explains comparison of mean orientation error of each data sequence. Taking the number of input data and the probability of error values appears into consideration, and in conjunction with Figure 6 and Figure 7, the table clearly demonstrates that error of training data is lower than that of test data. Besides, there is some relation between four variables of quaternion, and the combination of the four corresponding variables of quaternion can reflect an actual pose. In short, errors in Table 1 indicate that BP neural network can indeed be trained to achieve camera-IMU relative pose calibration and predict relative camera pose from IMU data.

5. Conclusion

In this paper, a new approach is proposed where a BP neural network using Levenberg-Marquardt algorithm is trained to estimate the relationship from IMU data to relative camera pose. The presented experimental results show that the method is reliable and after training, the predictive value is accurate in view of rotation. For future work, we will try to fusion visual and inertial measurements and achieve online calibration in real-time, and then apply the method to electronic real-time image stabilization.

Acknowledgments

The authors are very grateful to the editorial board members and the reviewers of esteemed journal for their valuable suggestions and constructive comments, which gratefully helped them to improve the research paper. This work is supported by the program of Jilin Science and Technology Development Plan (No.20170307002GX).

References

- Chen, M. (2013). Principle and example of MATLAB Neural Network. Tsinghua University press.
- Deepmala, A. (2014). Study on Fixed Point Theorems for Nonlinear Contractions and its Applications, Ph.D. Thesis (2014), Pt. Ravishankar Shukla University, Raipur 492010, Chhatisgarh, India.
- Gupta, S., Dalal, U., & Mishra, V. N. (2015). Performance on ICI self cancellation in FFT-OFDM and DCT-OFDM system, *Journal of Function Spaces*, Volume 2015, Article ID 854753, 7 pages.
- Jia, C., & Evans, B. L. (2007). Probabilistic 3-D motion estimation for rolling shutter video rectification from visual and inertial measurements. *International Workshop on Multimedia Signal Processing* (pp. 203-208). <https://doi.org/10.1109/mmisp.2012.6343441>
- Jia, C., & Evans, B. L. (2014). Online calibration and synchronization of cellphone camera and gyroscope (pp. 731-734). <https://doi.org/10.1109/globalsip.2013.6736995>
- Julier, S. J., & Uhlmann, J. K. (2004). Unscented Filtering and Nonlinear Estimation. <https://doi.org/10.1109/jproc.2004.837637>
- Kelly, J., & Sukhatme, G. S. (2009). Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Selfcalibration. *International Journal of Robotics Research*, 30(1), 56-79. <https://doi.org/10.1109/cira.2009.5423178>
- Li, M., & Mourikis, A. I. (2000). Improving the accuracy of EKF-based visual-inertial odometry. *Logic for Programming and Automated Reasoning* (pp.362-380). <https://doi.org/10.1109/icra.2012.6225229>
- Li, M., Kim, B. H., & Mourikis, A. I. (2013). Real-time motion tracking on a cellphone using inertial sensing and a rollingshutter camera (pp. 4712-4719). <https://doi.org/10.1109/icra.2013.6631248>
- Liu, J. F. (2004). Three dimensional rotation represented by quaternion.
- Mishra, L. N. (2017). On Existence Theorems for Some Generalized Nonlinear Functional-Integral Equations with Applications. <https://doi.org/10.2298/fil1707081n>
- Modh, A., Dabhi, M., Mishra, L. N., & Mishra, V. N. (2015). Wireless Network Controlled Robot using a Website, Android Application or Simple hand Gestures. *Journal of Computer Networks*, 3(1), 1-5. <https://doi.org/10.12691/jcn-3-1-1>
- Mor, J. J. (1978). The Levenberg-Marquardt algorithm: Implementation and theory (pp. 105-116).
- Tian, Y., & Fang, M. (2016). A time synchronization method for inertial sensor and visual sensor. *Intelligent Control and Automation (WCICA)*, 836-839. <https://doi.org/10.1109/wcica.2016.7578516>

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).