Hong Liu, Yu-long Zhou, Zhao-peng Gu, 2014. Inertial measurement unit-camera calibration based on incomplete inertial sensor information. *Journal of Zhejiang University-SCIENCE C (Computers & Electronics)*, **15**(11):999-1008.
[doi:10.1631/jzus.C1400038]

# Inertial measurement unit-camera calibration based on incomplete inertial sensor information

JZUS C (Comput & Electron)

**Key words:** Calibration, Computer vision, Inertial sensor, Smart phone, Incomplete information

Corresponding author: Zhao-peng Gu
E-mail: guzp@pkusz.edu.cn

# Motivation

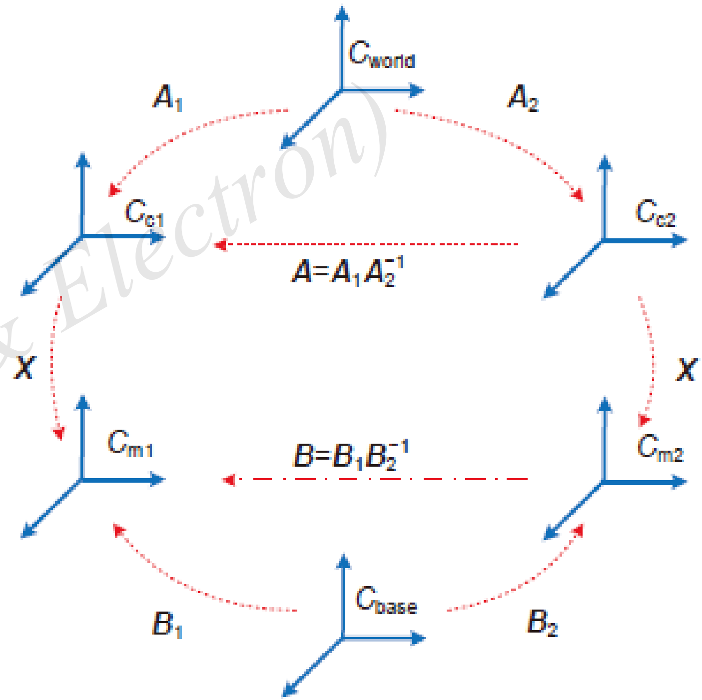
- This paper addresses the incomplete information based IMU camera calibration problem by exploiting the intrinsic restrictions among the coordinate transformations

- Disadvantages of existing methods:
  - Cannot deal with calibration with incomplete IMU information such as just Roll and Pitch
  - Analytical solutions are not given

JZUS C (Comput & Electron)

# Features of our method

- Use only roll and pitch of the IMU output
- Recover IMU Rotation $\boldsymbol{R}_B$ from property of similar matrices
- Using LM optimization to refine final result




$$\boldsymbol{R}_A\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}_B,$$

$$\mathrm{tr}(\boldsymbol{R}_A) = \mathrm{tr}(\widetilde{\boldsymbol{R}_B})$$

JZUS C (Comput & Electron)

# Framework of our method (I)

The algorithms of camera-IMU calibration can be divided into two parts:

1. Recovery of $\boldsymbol{R}_B$

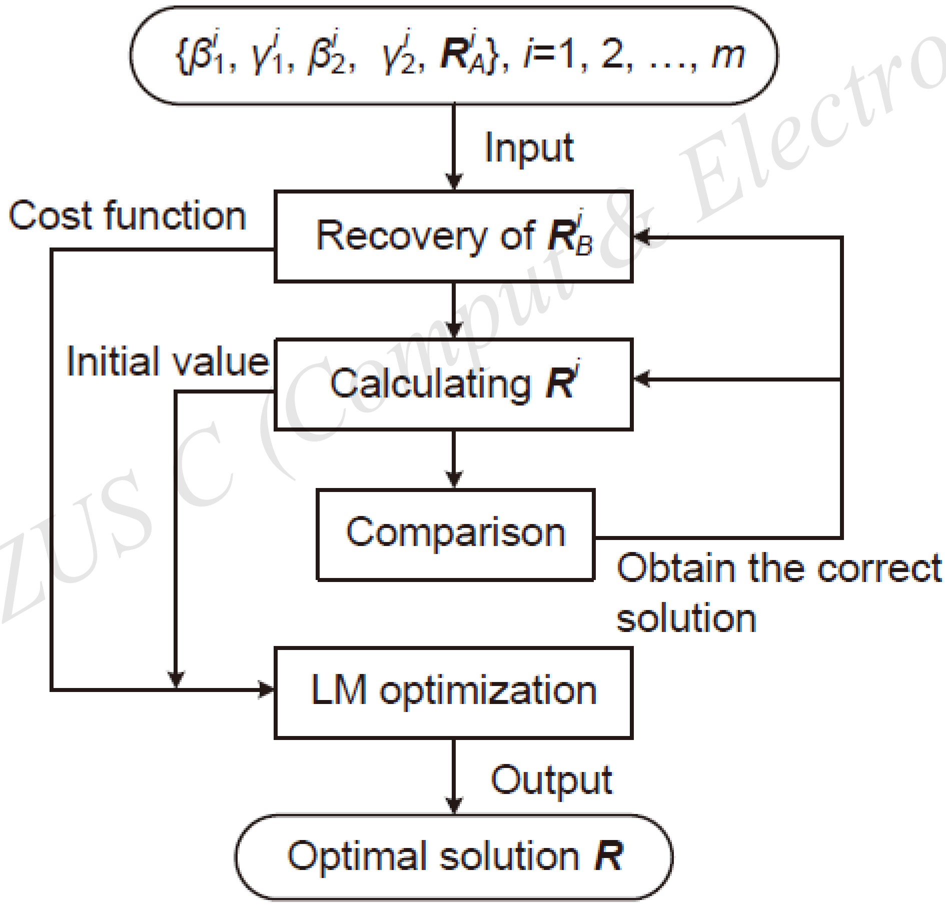
**Algorithm 1** Recovery of $\boldsymbol{R}_B$

**Input:** $\boldsymbol{R}_A$ and four Euler angles $\beta_1, \beta_2, \gamma_1, \gamma_2$
**Output:** $\boldsymbol{R}_B$
**Begin**
  Calculate $\boldsymbol{O}_i$ and $\boldsymbol{P}_i$, $i=1,2$
  Calculate $\boldsymbol{T}$
  Formulate $\widetilde{\boldsymbol{R}_B} = \text{yaw}(\alpha)\boldsymbol{T}$
  Compute $\text{tr}(\boldsymbol{R}_A)$ and $\text{tr}(\widetilde{\boldsymbol{R}_B})$
  Solve $\alpha$ from Eq. (17)
  Recover $\boldsymbol{R}_B = \boldsymbol{O}_1\boldsymbol{P}_1\text{yaw}(\alpha)\boldsymbol{P}_2^{\text{T}}\boldsymbol{O}_2^{\text{T}}$
**End**

JZUS C (Comput & Electron)

# Framework of our method (II)

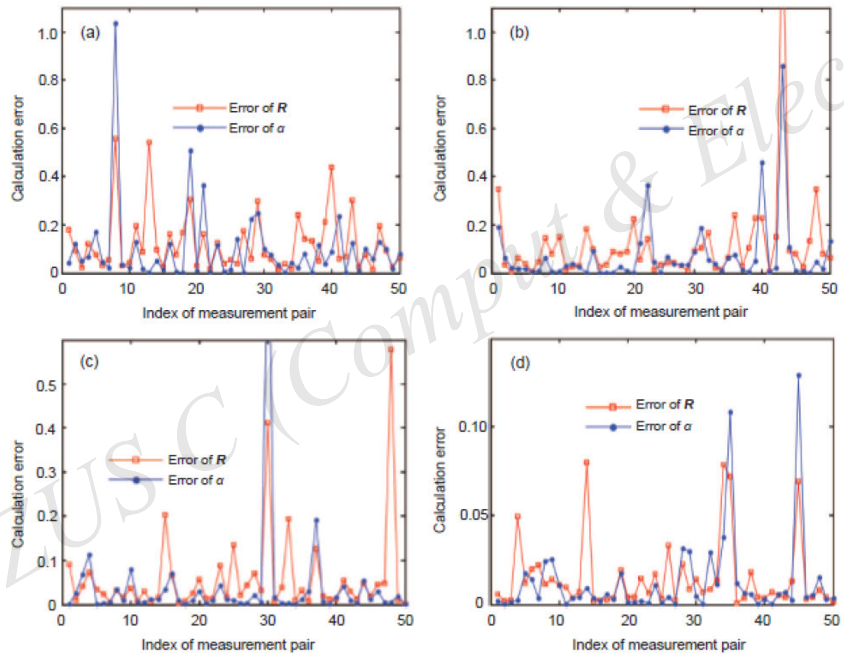
2. Calculating ***R*** and LM optimization



JZUS C (Comput & Electron)

# Simulation & results

For comparison, the mean of the two *e* is regarded as the error of  for each pair since there are two measurements in each pair. The experimental results are:

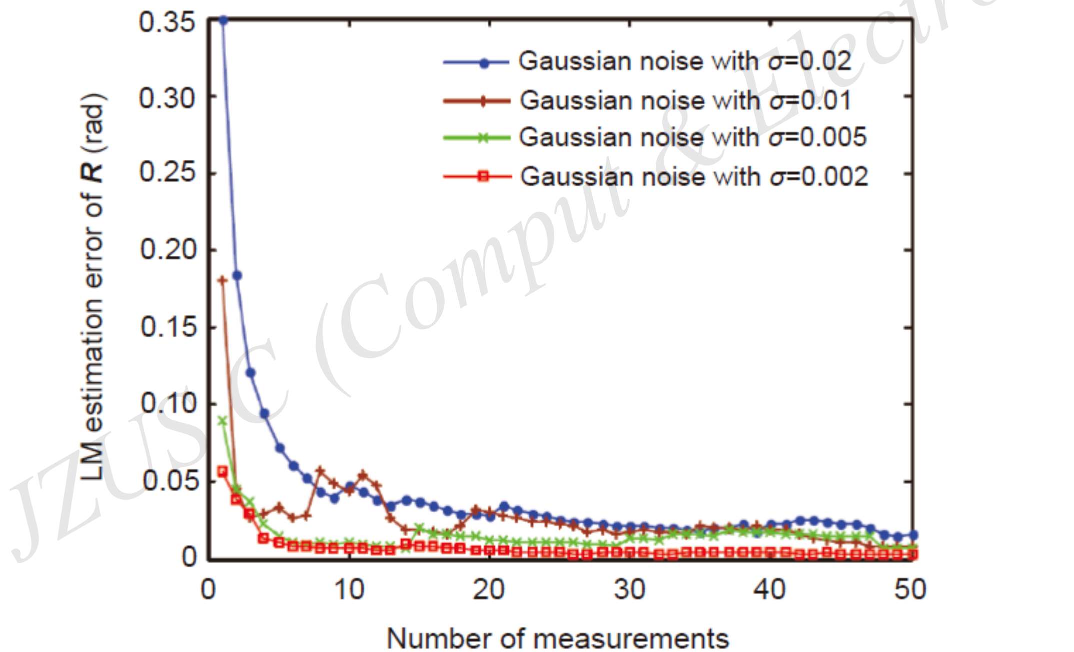



JZUS C (Comput & Electron)

Most errors of ***R*** are within the limits of 20*σ* whereas a few errors are relatively large which is not surprising due to the strong noise there. The two curves generally followed the same trends. Inconsistencies of the two curves may result from the error of $\boldsymbol{R}_A$ or other computational errors

# Simulation & results

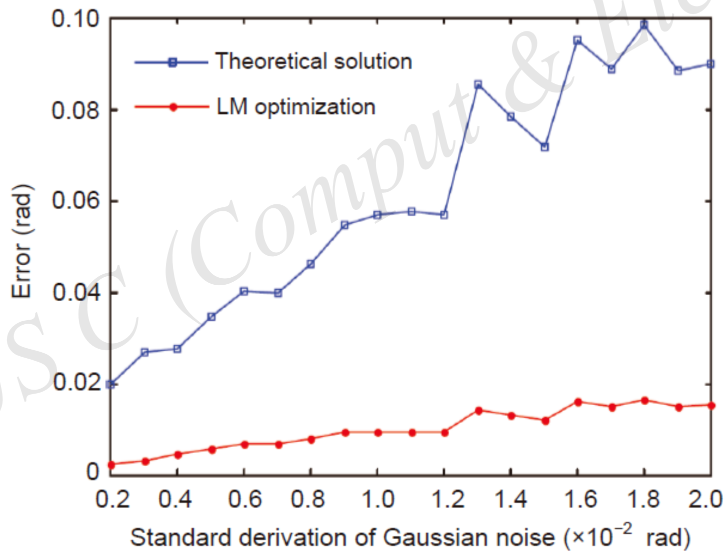
The number of the measurements *m* is assigned from 1 to 50 in the cost function. As a consequence, the LM algorithm is called 200 times in four Gaussian noise environments. The experimental results are:




It is seen that the error curves converge as *m* increases and the converged error value is about the same as $\sigma$

# Simulation & results

To better tell how the errors of the estimates grow with the Gaussian noise, we carried out experiments in 19 Gaussian noise environments with standard derivations ranging from 0.002 to 0.02 using both the theoretical method and LM optimization

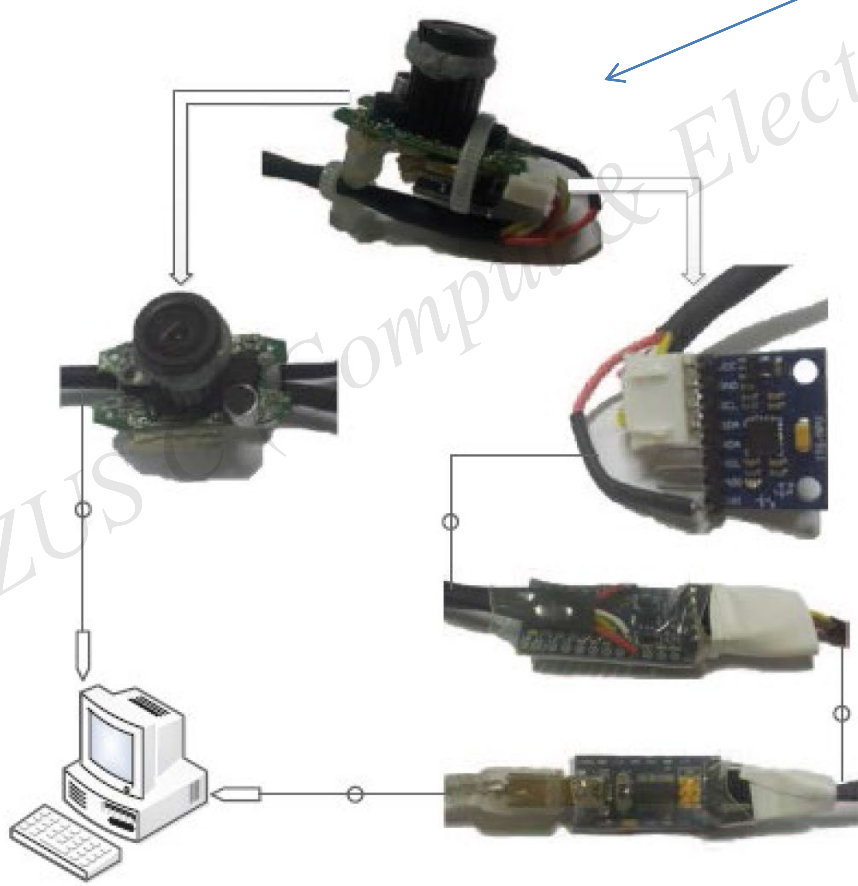



The average error of the theoretical method is about 5*σ*–10*σ* and the error of LM optimization is about *σ*

JZUS C (Comput & Electron)

# Real experiments & results

Experiment setup:



# Real experiments & results

Result:

$$\widehat{\boldsymbol{R}} = \begin{bmatrix} -0.1478 & 0.9836 & -0.1029 \\ -0.9890 & -0.1464 & 0.0205 \\ 0.0051 & 0.1048 & 0.9945 \end{bmatrix} \approx \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

JZUS C (Comput & Electron)