Absolute scale velocity determination combining visual and inertial measurements for micro aerial vehicles

Jacques Kaiser

INRIA

June 24, 2015

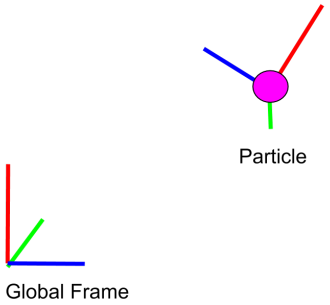
Section 1

Sensor fusion

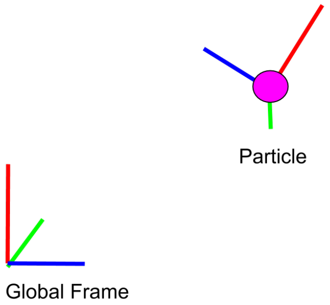
Micro aerial vehicles

Micro aerial vehicles

Micro aerial vehicles



Micro aerial vehicles

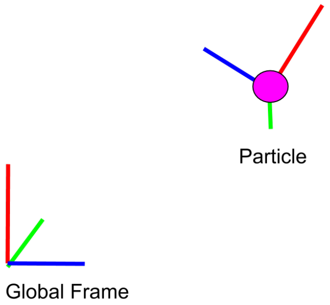


A basic state vector:

$$X = \begin{bmatrix} r \\ \dot{r} \\ q \end{bmatrix}$$

- r position;
- $\dot{r}$ velocity;
- q orientation.

Micro aerial vehicles



A basic state vector:

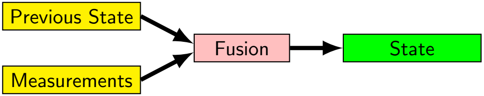
$$X = \begin{bmatrix} r \\ \dot{r} \\ q \end{bmatrix}$$

- r position;
- $\dot{r}$ velocity;
- q orientation.

The goal of sensor fusion is to recover the state X

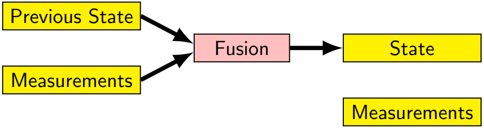
Visual-inertial sensor fusion

Filter based method

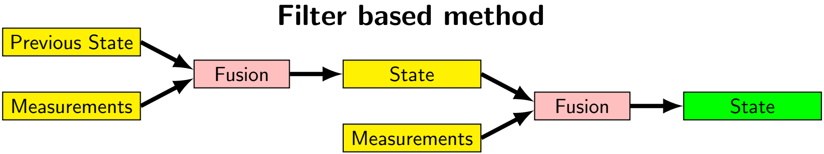


Visual-inertial sensor fusion

Filter based method

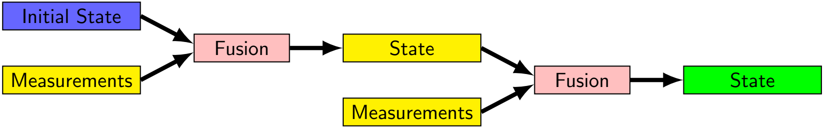


Visual-inertial sensor fusion



Visual-inertial sensor fusion

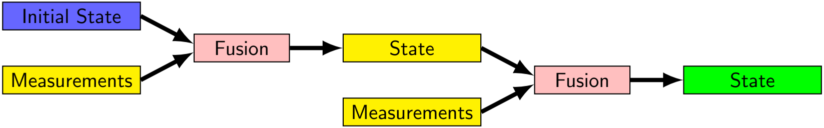
Filter based method



How to recover the **initial state**?

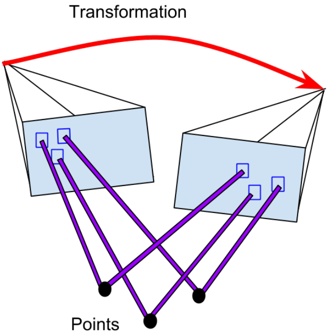
Visual-inertial sensor fusion

Filter based method

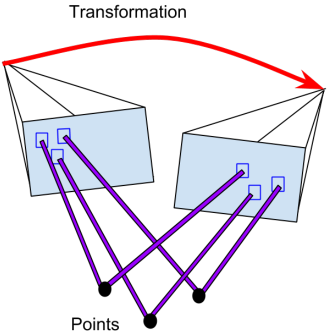


How to recover the **initial state**?
We need a **deterministic solution**

Deterministic solutions in Computer Vision



Deterministic solutions in Computer Vision



But the relative translation and distance to features are recovered only **up to scale**

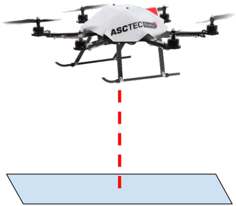
Absolute scale from visual measurements

How big is this building?

Absolute scale from visual measurements

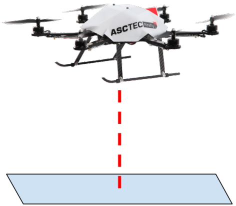
Methods to recover the absolute scale

Methods to recover the absolute scale



Methods to recover the absolute scale

Not suited to unknown environments



Not precise, works only in hover

Inertial Measurement Unit (IMU)

The IMU consists of two sensors providing **physical quantities**:

- Accelerometer: linear acceleration - gravity (m/s^2);
- Gyroscope: angular velocity (rad/s).

Title

absolute scale velocity determination combining visual and inertial measurements for micro aerial vehicles

Table of Contents

Section 2

The closed-form solution

The Closed-Form Solution - 2014

Requires:

- Calibrated camera;
- Inertial Measurement Unit (IMU);
- External Camera IMU transformation.

Output:

- Initial velocity;
- Distance to point-features;
- Attitude.

The Closed-Form Solution - 2014

Requires:

- Calibrated camera;
- Inertial Measurement Unit (IMU);
- External Camera IMU transformation.

Output:

- Initial velocity;
- Distance to point-features;
- Attitude.

$$S_j = \lambda_1^i \mu_1^i - V t_j - G \frac{t_j^2}{2} - \lambda_j^i \mu_j^i$$

The Closed-Form Solution - 2014

Requires:

- Calibrated camera;
- Inertial Measurement Unit (IMU);
- External Camera IMU transformation.

Output:

- Initial velocity;
- Distance to point-features;
- Attitude.

$$S_j = \lambda_1^i \mu_1^i - V t_j - G \frac{t_j^2}{2} - \lambda_j^i \mu_j^i$$

The Overconstrained Linear System

$$S_j = \lambda_1^i \mu_1^i - V t_j - G \frac{t_j^2}{2} - \lambda_j^i \mu_j^i$$

Valid for every point-features i at any time t_j:

The Overconstrained Linear System

$$S_j = \lambda_1^i \mu_1^i - V t_j - G \frac{t_j^2}{2} - \lambda_j^i \mu_j^i$$

Valid for every point-features i at any time t_j:

$$\left[\begin{array}{c} S_2 = \lambda_1^1 \mu_1^1 - V t_2 - G \frac{t_2^2}{2} - \lambda_2^1 \mu_2^1 \\ S_2 = \lambda_1^2 \mu_1^2 - V t_2 - G \frac{t_2^2}{2} - \lambda_2^2 \mu_2^2 \\ \vdots \\ S_3 = \lambda_1^1 \mu_1^1 - V t_3 - G \frac{t_3^2}{2} - \lambda_3^1 \mu_3^1 \\ \vdots \\ S_N = \lambda_1^{n_i} \mu_1^{n_i} - V t_N - G \frac{t_N^2}{2} - \lambda_N^{n_i} \mu_N^{n_i} \end{array}\right.$$

The Overconstrained Linear System

$$S_j = \lambda_1^i \mu_1^i - V t_j - G \frac{t_j^2}{2} - \lambda_j^i \mu_j^i$$

Valid for every point-features i at any time t_j:

$$\left[\begin{array}{c} S_2 = \lambda_1^1 \mu_1^1 - V t_2 - G \frac{t_2^2}{2} - \lambda_2^1 \mu_2^1 \\ S_2 = \lambda_1^2 \mu_1^2 - V t_2 - G \frac{t_2^2}{2} - \lambda_2^2 \mu_2^2 \\ \vdots \\ S_3 = \lambda_1^1 \mu_1^1 - V t_3 - G \frac{t_3^2}{2} - \lambda_3^1 \mu_3^1 \\ \vdots \\ S_N = \lambda_1^{n_i} \mu_1^{n_i} - V t_N - G \frac{t_N^2}{2} - \lambda_N^{n_i} \mu_N^{n_i} \end{array} \right.$$

$$\Xi X = S$$

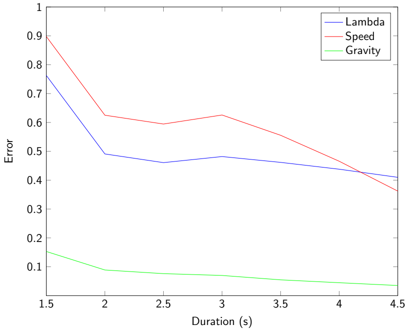
Problem: not robust in practice

"**A closed-form solution** *for state estimation with a visual-inertial system that does not require initialization was presented. However, this approach is* **not suitable for systems that rely on noisy sensor data**"

— Matthias Faessler, ICRA 2015

Problem: not robust in practice

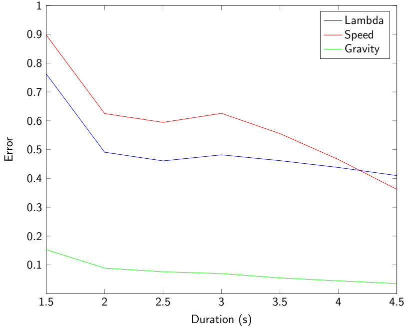
> "**A closed-form solution** *for state estimation with a visual-inertial system that does not require initialization was presented. However, this approach is* **not suitable for systems that rely on noisy sensor data**"
>
> — Matthias Faessler, ICRA 2015



Problem: not robust in practice

*"***A closed-form solution** *for state estimation with a visual-inertial system that does not require initialization was presented. However, this approach is* **not suitable for systems that rely on noisy sensor data***"*

— Matthias Faessler, ICRA 2015



50% relative error on speed and distance estimation

Improving the performance

What makes the estimations so bad?

Possible bottlenecks

- Motion;

Improving the performance

What makes the estimations so bad?

Possible bottlenecks

- Motion;
- Observed features;

Improving the performance

What makes the estimations so bad?

Possible bottlenecks

- Motion;
- Observed features;
- Noisy sensors.

Improving the performance

What makes the estimations so bad?

Possible bottlenecks

- Motion;
- Observed features;
- Noisy sensors.

Sensors provide measurements affected by a Gaussian noise:

$$N(\mu + B, \sigma^2)$$

Accelerometer bias

In dead reckoning task:

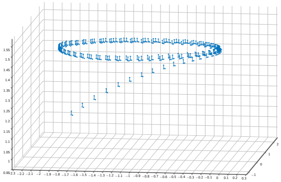


Figure : Ground truth motion

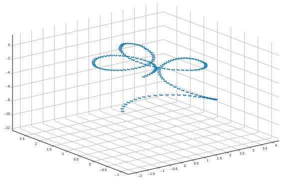


Figure : Dead reckoning with accelerometer bias

In the closed-form solution:

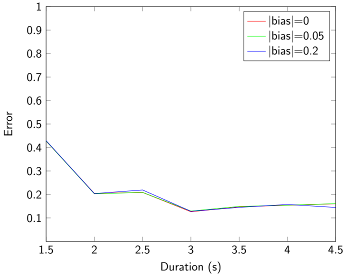


Figure : Speed estimation error with varying accelerometer bias

Gyroscope bias

In dead reckoning task:

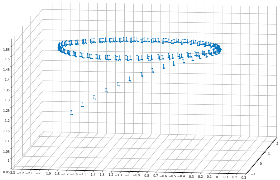


Figure : Ground truth motion

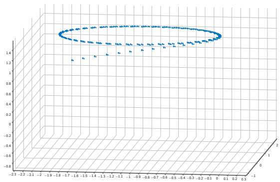


Figure : Dead reckoning with gyroscope bias

In the closed-form solution:

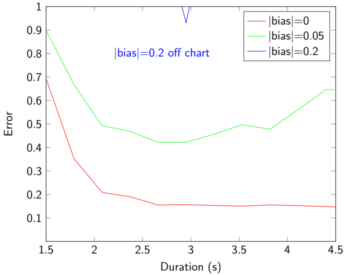


Figure : Speed estimation error with varying gyroscope bias

Estimating the gyroscope bias

- When solving $\Xi X = S$, we are solving $\text{argmin}_X \, ||\Xi X - S||^2$;

Estimating the gyroscope bias

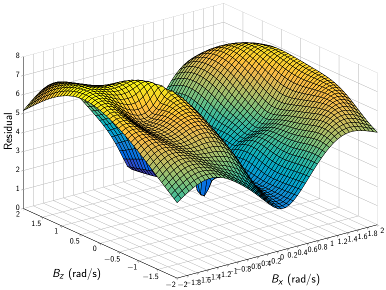
- When solving $\Xi X = S$, we are solving $\mathrm{argmin}_X \, ||\Xi X - S||^2$;
- Unfortunately, we can not express the gyroscope bias linearly;

Estimating the gyroscope bias

- When solving $\Xi X = S$, we are solving $\text{argmin}_X \, ||\Xi X - S||^2$;
- Unfortunately, we can not express the gyroscope bias linearly;
- Alternative: non-linear minimization

$$\underset{B,X}{\text{argmin}} \, ||\Xi X - S||^2$$

With B the gyroscope bias, Ξ and S computed with respect to B

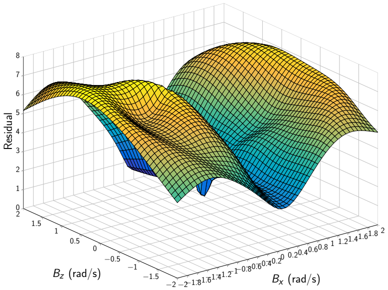


Estimating the gyroscope bias

- When solving $\Xi X = S$, we are solving $\text{argmin}_X \, ||\Xi X - S||^2$;
- Unfortunately, we can not express the gyroscope bias linearly;
- Alternative: non-linear minimization

$$\underset{B,X}{\text{argmin}} \, ||\Xi X - S||^2$$

With B the gyroscope bias, Ξ and S computed with respect to B



Symmetry induced by the strong weight of the gravity

Getting rid of the symmetry

We introduce a regularization paremeter λ:

$$\underset{B,X}{\operatorname{argmin}} ||\Xi X - S||^2 + \lambda \times |B|$$

Getting rid of the symmetry

We introduce a regularization paremeter λ:

$$\underset{B,X}{\operatorname{argmin}} ||\Xi X - S||^2 + \lambda \times |B|$$

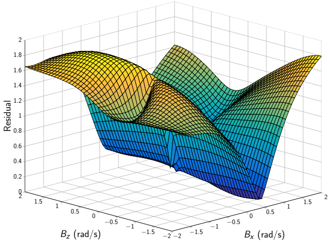


Figure : No regularization

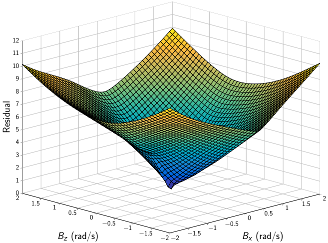


Figure : With regularization $\lambda = 3$

Results

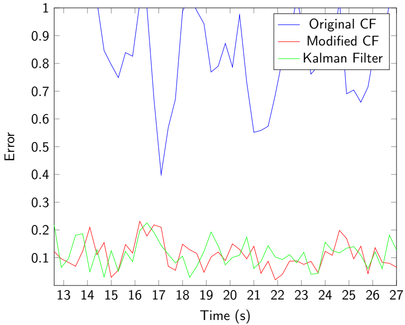


Figure : Velocity estimation error
Original closed-form solution
Regularized optimized closed-form solution
Initialized Kalman-Filter

Moreover, our technique also provides the gyroscope bias

Section 3

Conclusion

Conclusion

Potential PhD