

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

Jacques Kaiser

Supervised by Dr. Agostino Martinelli

INRIA



June 24, 2015

Micro aerial vehicles



Micro aerial vehicles



Micro aerial vehicles



Micro aerial vehicles



Micro aerial vehicles



Localization in various environments

Micro aerial vehicles



Localization in various environments

Micro aerial vehicles



A basic state vector:

$$X = \begin{bmatrix} \textit{position} \\ \textit{velocity} \\ \textit{orientation} \end{bmatrix}$$

Localization in various environments

Micro aerial vehicles



A basic state vector:

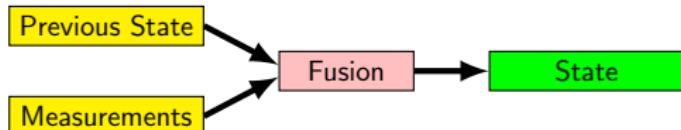
$$X = \begin{bmatrix} \textit{position} \\ \textit{velocity} \\ \textit{orientation} \end{bmatrix}$$

Localization in various environments

The goal of sensor fusion for state estimation is to **recover X**

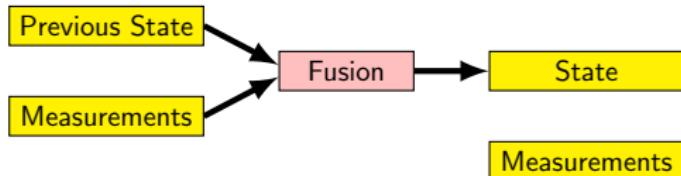
Visual-inertial sensor fusion

Filter based method



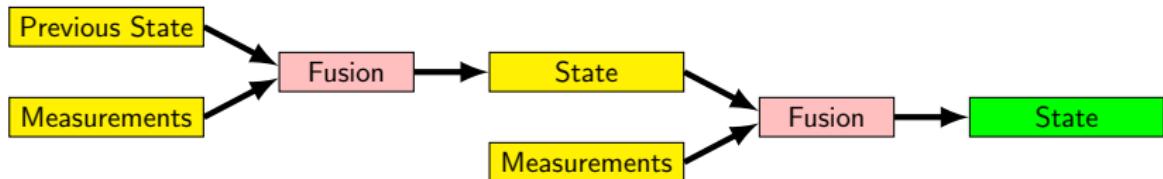
Visual-inertial sensor fusion

Filter based method



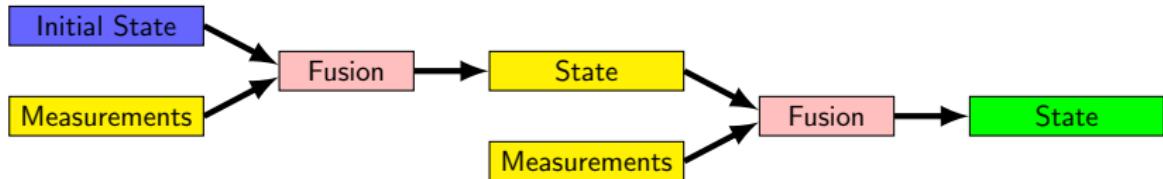
Visual-inertial sensor fusion

Filter based method



Visual-inertial sensor fusion

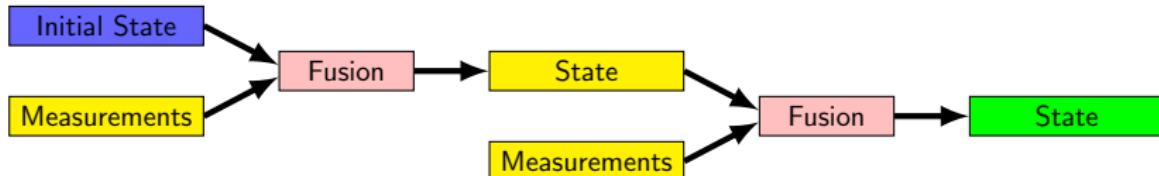
Filter based method



How to recover the **initial state**?

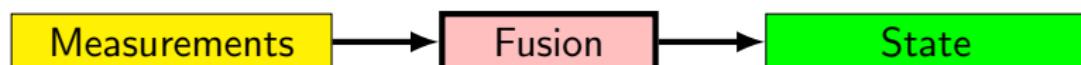
Visual-inertial sensor fusion

Filter based method

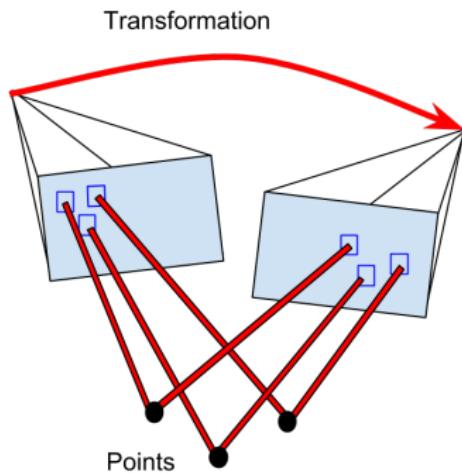


How to recover the **initial state**?

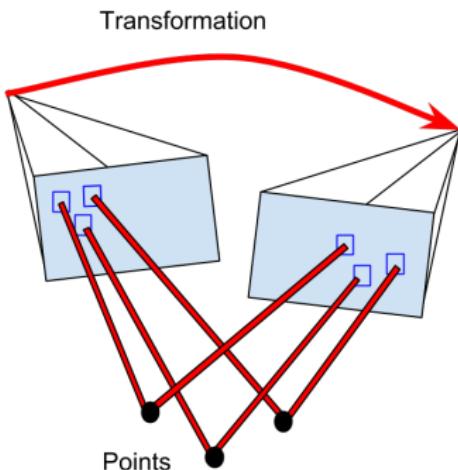
We need a **deterministic solution**



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



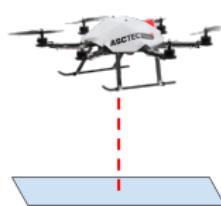
Methods to recover the absolute scale



Methods to recover the absolute scale



Not suited to
unknown
environments



Not precise, works
only in hover



Does not work in GPS
denied environments

Inertial Measurement Unit (IMU)

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

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

[Back to the title](#)

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

[Back to the title](#)

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

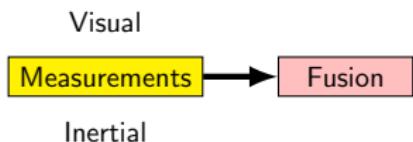
Visual

Measurements

Inertial

Back to the title

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



Back to the title

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

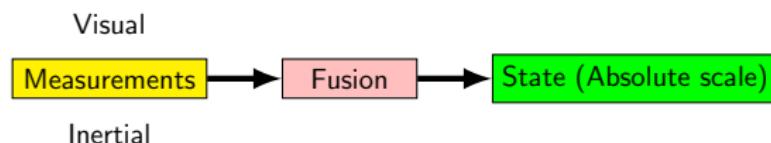


Table of Contents

Sensor fusion

- Filter based fusion

- Deterministic solution

- Absolute scale

The closed-form solution

- Expression

- Identifying performance bottlenecks

- Estimating the gyroscope bias

- Validation

Conclusion

The closed-form solution

Transactions on Robotics (T-RO) 2012

International Journal of Computer Vision (IJCV) 2014

The Closed-Form Solution

Requires:

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

Output:

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

The Closed-Form Solution

Requires:

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

Output:

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

$$\Xi X = S$$

The Closed-Form Solution

Requires:

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

Output:

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

$$\Xi X = S$$

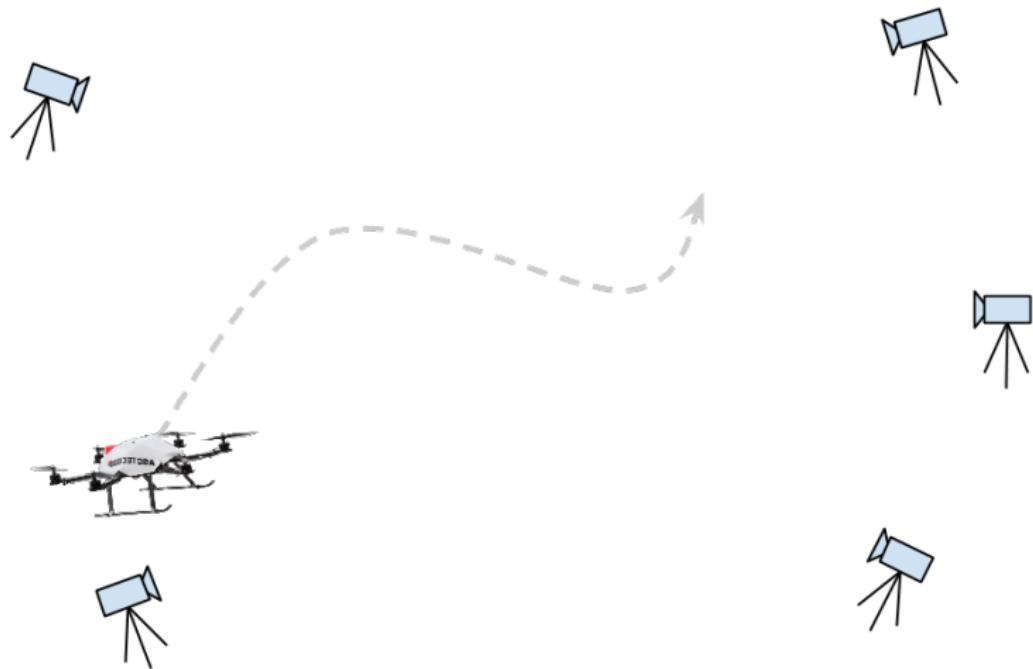
Test setup



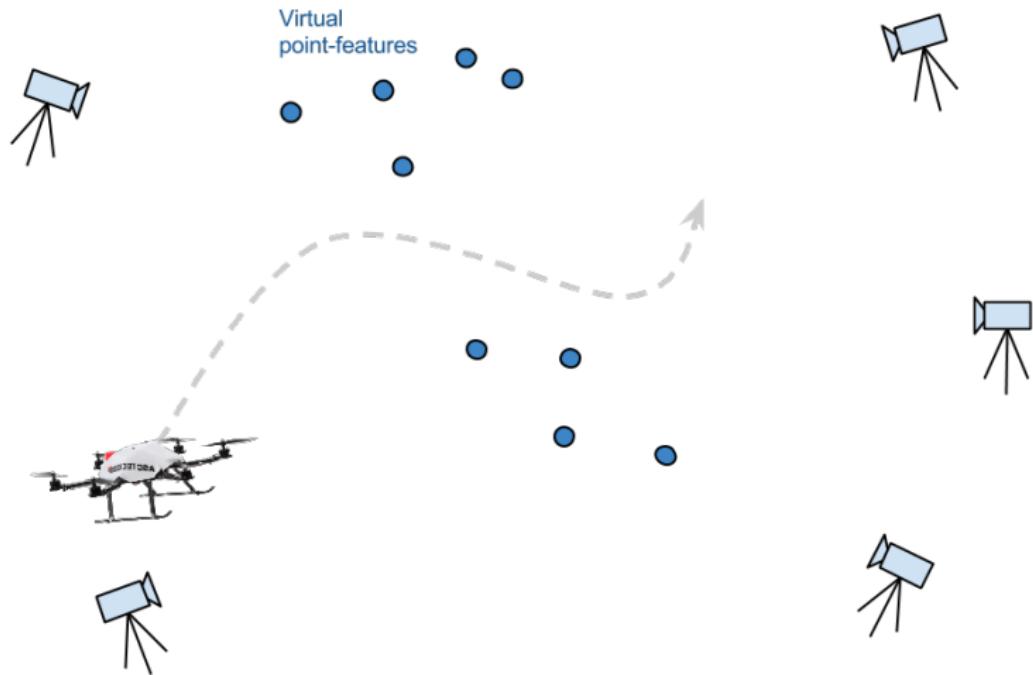
Test setup



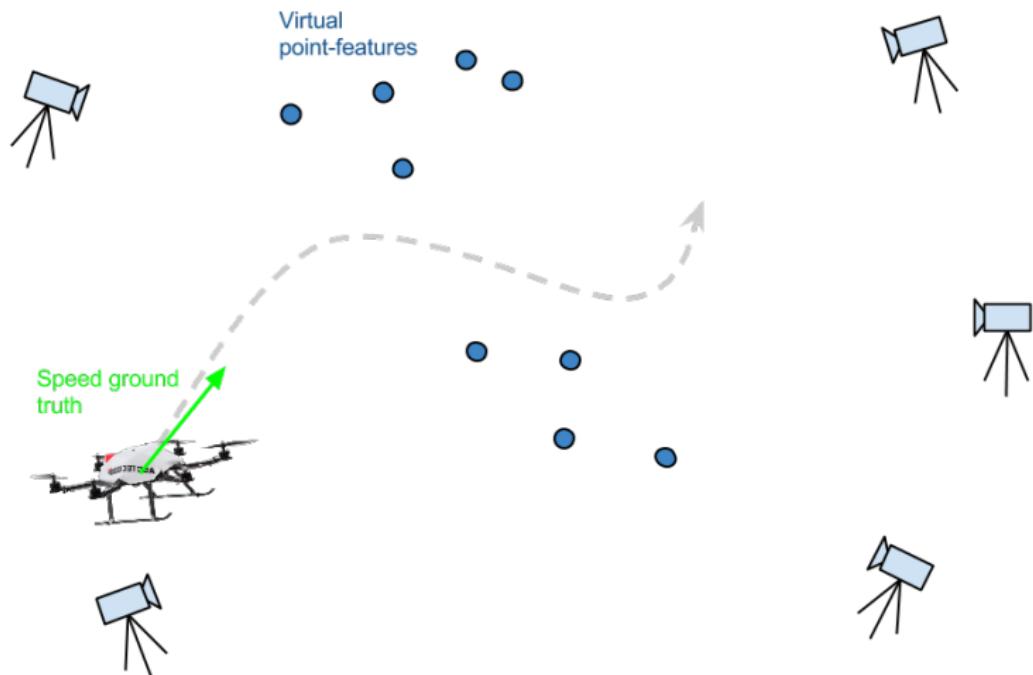
Test setup



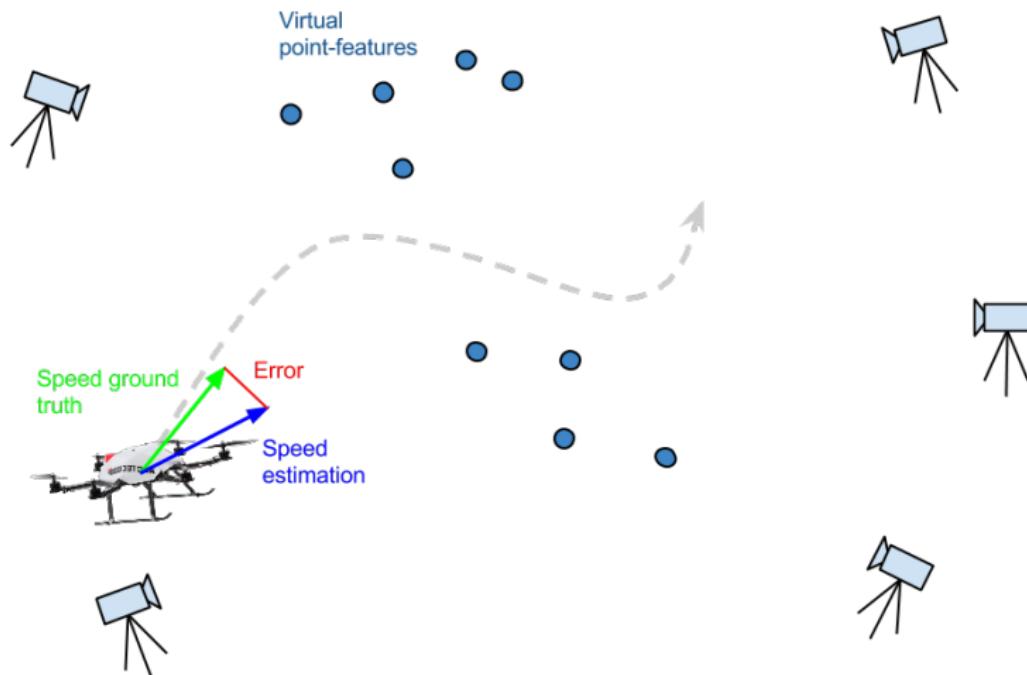
Test setup



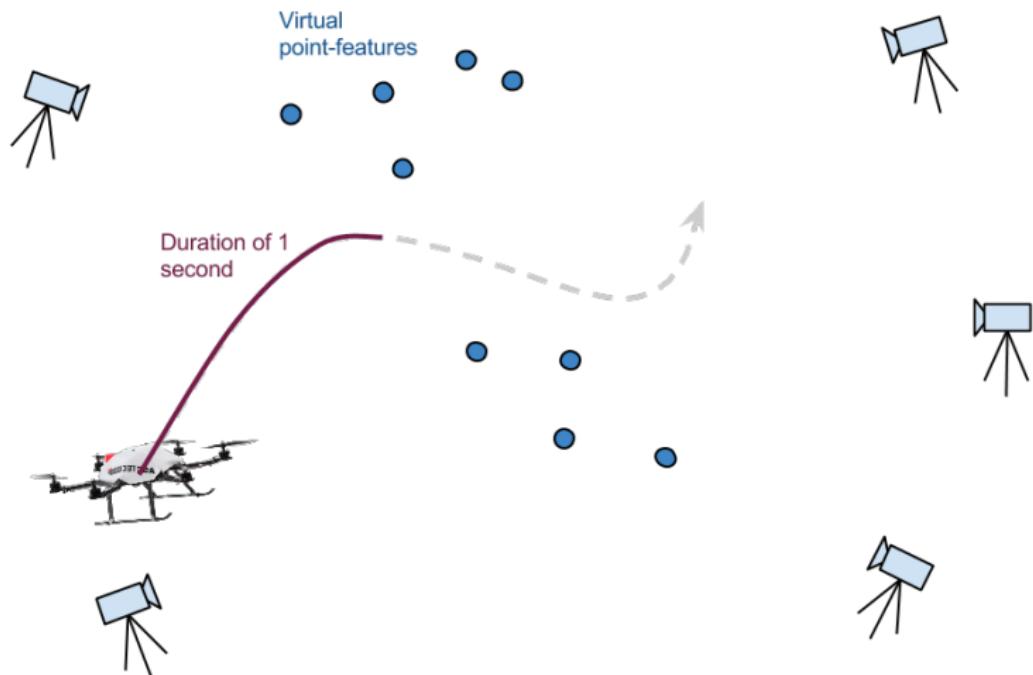
Test setup



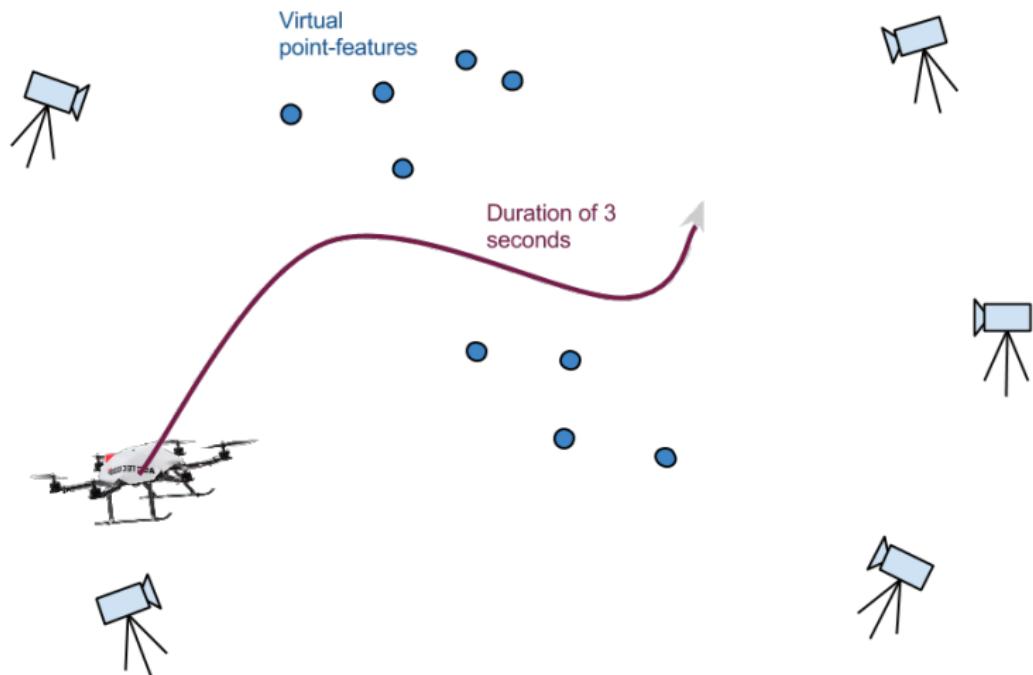
Test setup



Test setup



Test setup



Problem: not robust in practice

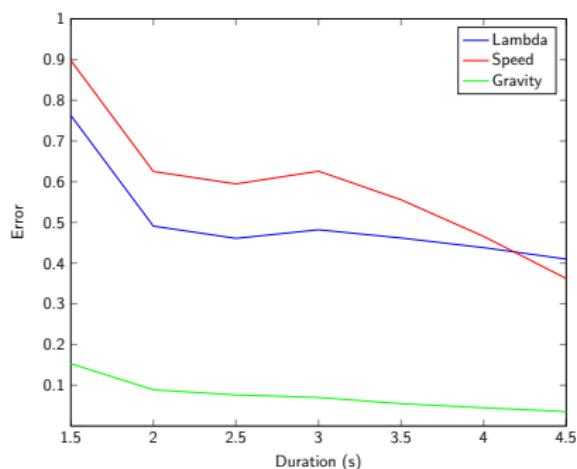
"A closed-form solution for state estimation with a visual-inertial system that does not require initialization was presented. However, this approach is not suitable for systems that rely on noisy sensor data"

— 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"

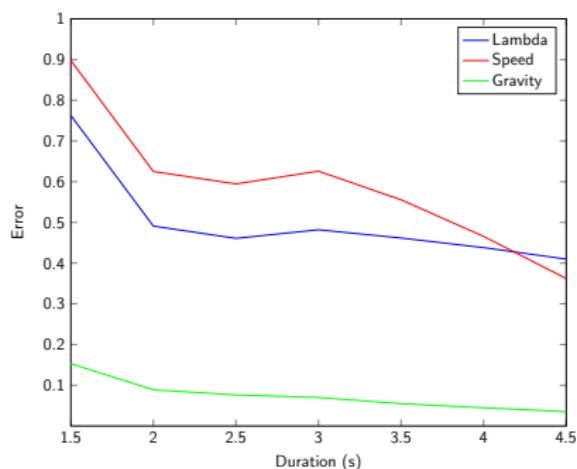
— Matthias Faessler, ICRA 2015



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:

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

Improving the performance

What makes the estimations so bad?

Possible bottlenecks

- ▶ Motion;
- ▶ Observed features;
- ▶ Noisy sensors.

Sensors provide measurements affected by a Gaussian noise:

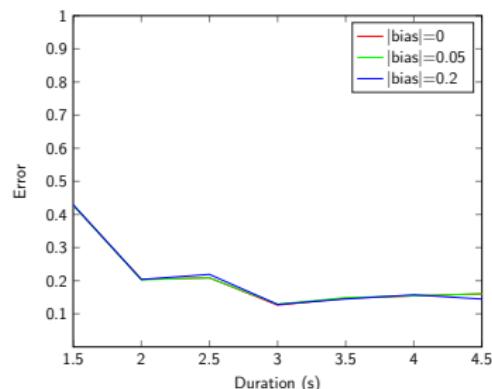
$$N(\mu + B, \sigma^2)$$

We considered:

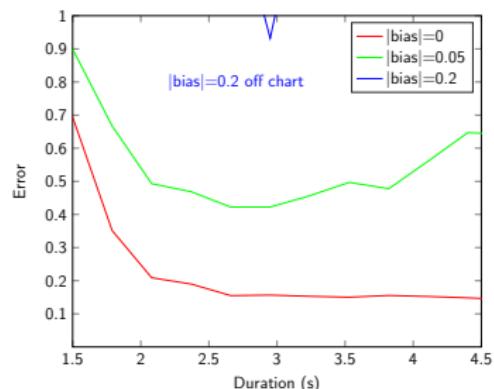
- ▶ Biased accelerometer;
- ▶ Biased gyroscope.

Biased inertial measurements

Speed estimation error



Varying accelerometer bias



Varying gyroscope bias

Estimating the gyroscope bias

Reminder: $\Xi X = S$

Estimating the gyroscope bias

Reminder: $\Xi X = S$

- ▶ Insert the gyro bias B in X ;

Estimating the gyroscope bias

Reminder: $\Xi X = S$

- ▶ Insert the gyro bias B in X ;

Estimating the gyroscope bias

Reminder: $\Xi X = S$

- ▶ Insert the gyro bias B in X ;
- ▶ Alternative: non-linear minimization of the residual

$$cost(B) = \underset{X}{\operatorname{argmin}} ||\Xi X - S||^2$$

With:

B the gyroscope bias,

Ξ and S computed with respect to B

Estimating the gyroscope bias

Reminder: $\Xi X = S$

- ▶ Insert the gyro bias B in X ;
- ▶ Alternative: non-linear minimization of the residual

$$cost(B) = \underset{X}{\operatorname{argmin}} ||\Xi X - S||^2$$

With:

B the gyroscope bias,

Ξ and S computed with respect to B

Estimating the gyroscope bias

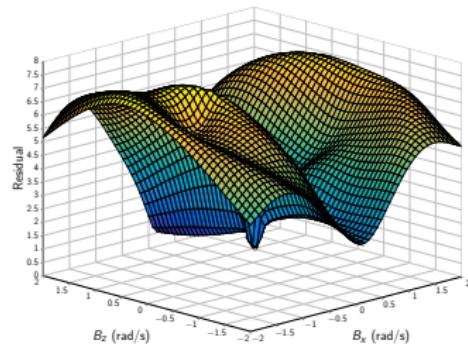
Reminder: $\Xi X = S$

- ▶ Insert the gyro bias B in X ;
- ▶ Alternative: non-linear minimization of the residual

$$cost(B) = \underset{X}{\operatorname{argmin}} ||\Xi X - S||^2$$

With:
 B the gyroscope bias,

Ξ and S computed with respect to B



Cost function: residual with respect to gyroscope bias

Estimating the gyroscope bias

Reminder: $\Xi X = S$

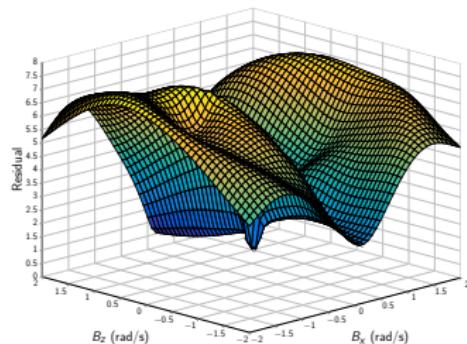
- ▶ Insert the gyro bias B in X ;
- ▶ Alternative: non-linear minimization of the residual

$$cost(B) = \underset{X}{\operatorname{argmin}} ||\Xi X - S||^2$$

With:

B the gyroscope bias,

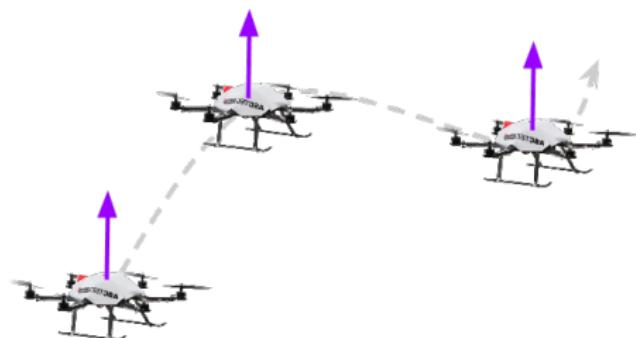
Ξ and S computed with respect to B



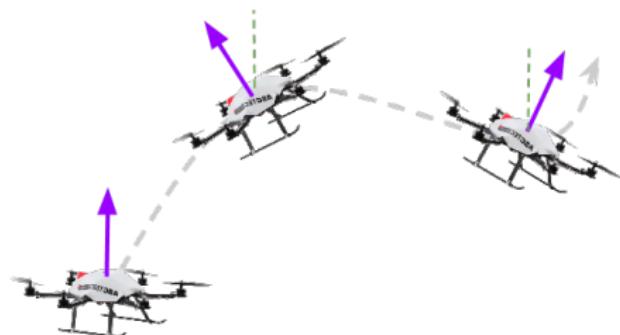
Cost function: residual with respect to gyroscope bias

Symmetry induced by the strong weight of the gravity

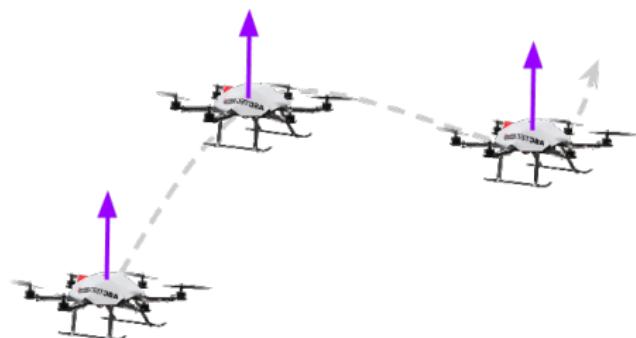
Dealing with the symmetry



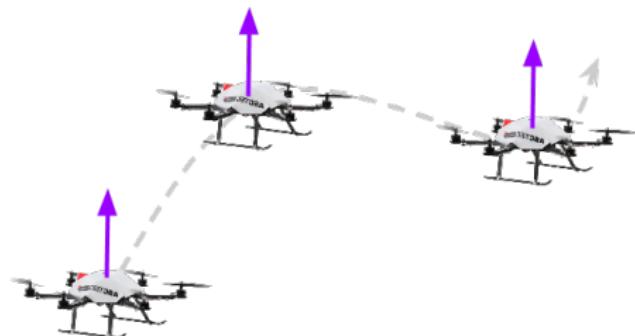
Dealing with the symmetry



Dealing with the symmetry



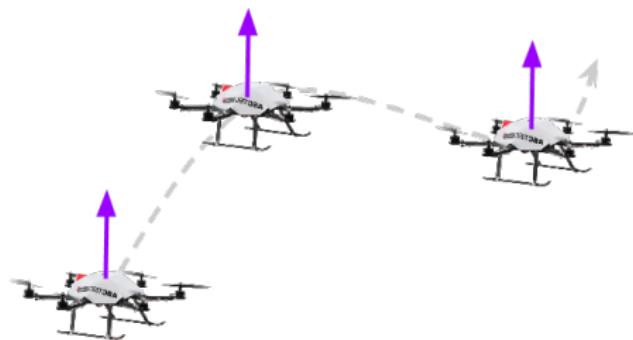
Dealing with the symmetry



We tweak the cost function:

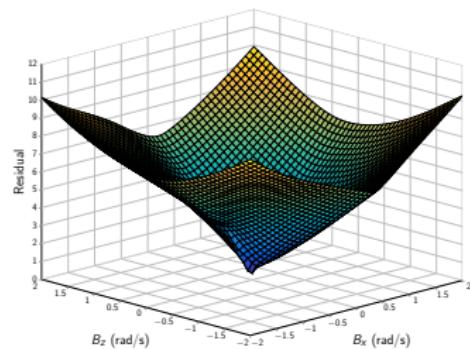
$$cost(B) = \underset{X}{\operatorname{argmin}} ||\Xi X - S||^2 + \lambda \times |B|$$

Dealing with the symmetry



We tweak the cost function:

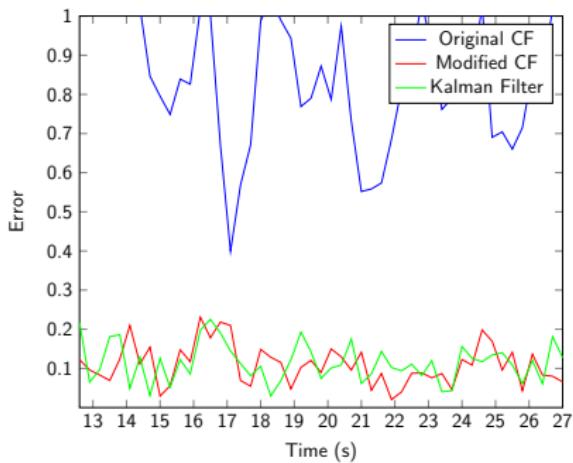
$$cost(B) = \underset{X}{\operatorname{argmin}} ||\Xi X - S||^2 + \lambda \times |B|$$



Regularized cost function with $\lambda = 3$
18 / 23

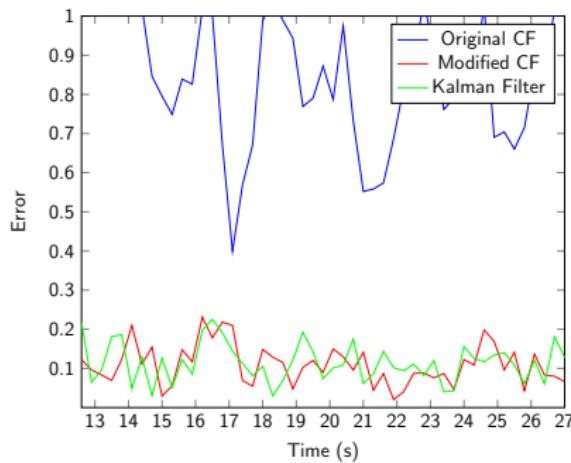
Results

Speed estimation error



Results

Speed estimation error



Moreover, our technique:

- ▶ Does not require initialization;
- ▶ Provides the gyroscope bias.

Conclusion

Conclusion

1. Closed-form solution for visual-inertial fusion;
 - ▶ No initialization required;
 - ▶ Recovers the absolute scale.

Conclusion

1. Closed-form solution for visual-inertial fusion;
 - ▶ No initialization required;
 - ▶ Recovers the absolute scale.
2. Does not work well in practice;

Conclusion

1. Closed-form solution for visual-inertial fusion;
 - ▶ No initialization required;
 - ▶ Recovers the absolute scale.
2. Does not work well in practice;
3. Gyroscope bias is a performance bottleneck;

Conclusion

1. Closed-form solution for visual-inertial fusion;
 - ▶ No initialization required;
 - ▶ Recovers the absolute scale.
2. Does not work well in practice;
3. Gyroscope bias is a performance bottleneck;
4. **Method to estimate this bias;**

Conclusion

1. Closed-form solution for visual-inertial fusion;
 - ▶ No initialization required;
 - ▶ Recovers the absolute scale.
2. Does not work well in practice;
3. Gyroscope bias is a performance bottleneck;
4. **Method to estimate this bias;**
5. Works well in practice;

Conclusion

1. Closed-form solution for visual-inertial fusion;
 - ▶ No initialization required;
 - ▶ Recovers the absolute scale.
2. Does not work well in practice;
3. Gyroscope bias is a performance bottleneck;
4. **Method to estimate this bias;**
5. Works well in practice;
6. Also provides the gyroscope bias.

Future work

Implementation on a real platform



Thank you for your attention,
Any question?

The Closed-Form Solution

Requires:

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

Output:

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

The Closed-Form Solution

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

Requires:

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

Output:

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

$$S_j = \lambda_1^j \mu_1^j - \textcolor{red}{V} t_j - \textcolor{blue}{G} \frac{t_j^2}{2} - \lambda_j^j \mu_j^j$$

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-feature i at any time t_j :

The Overconstrained Linear System

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

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

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

The Overconstrained Linear System

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

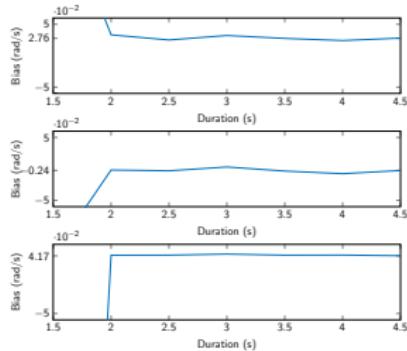
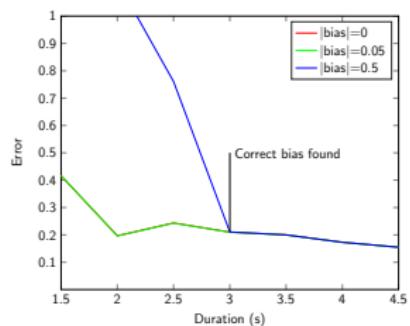
Valid for every point-feature i at any time t_j :

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

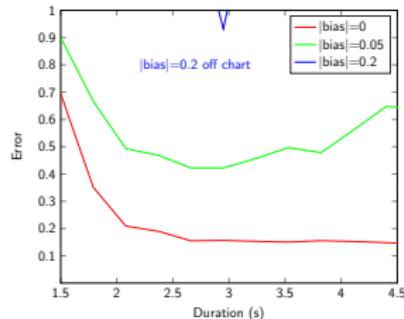
$$\Xi X = S$$

Gyroscope bias estimation

Optimized Closed-form solution



Original Closed-form solution



Numerical stability

$$\left[\begin{array}{lcl} S_j & = & \lambda_1^1 \mu_1^1 - Vt_j - G \frac{t_j^2}{2} - \lambda_j^1 \mu_j^1 \\ 0_3 & = & \lambda_1^1 \mu_1^1 - \lambda_j^1 \mu_j^1 - \lambda_1^i \mu_1^i + \lambda_j^i \mu_j^i \end{array} \right] \quad \left| \quad S_j = \lambda_1^i \mu_1^i - Vt_j - G \frac{t_j^2}{2} - \lambda_j^i \mu_j^i \right.$$

