ODOMETRY AND LOW-COST SENSOR FUSION IN TMM DATASET

February 18th to February 20th, 2016
In memory of professor Giorgio Manzoni

Ambrogio M. Manzino **Cinzia Taglioretti (*)**

(*) Politecnico di Torino, DIATI

POLITECNICO DI TORINO

Aim of the work

- Identify the most powerful motion model and filtering technique in TMM survey
- in view of further integrations with Photogrammetry
- Tool of integration = UKF / AUKF

Case study

- TMM around Politecnico di Torino, the area is occluded by building with 2 to 5 floors
- The path is displayed in red, the results we'll see are the part in green with ±90° curves, starts and stop

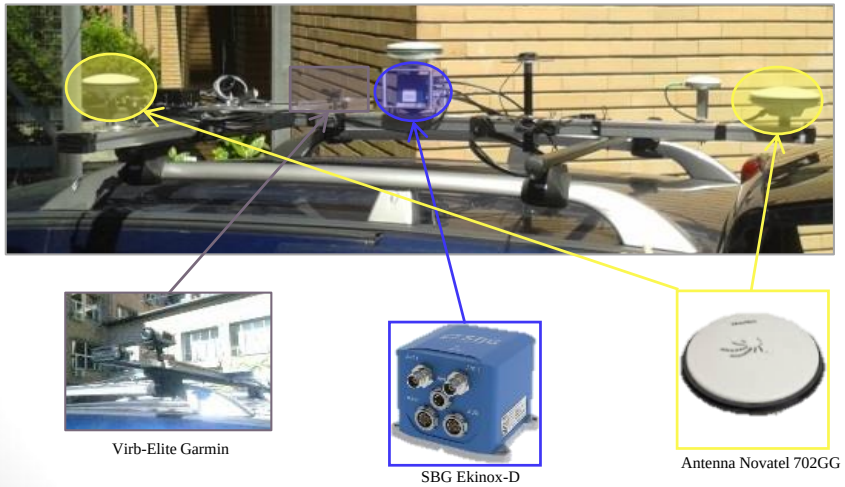
The low cost sensors used …and why?

Garmin® Action Cam Virb Elite.
GPS high sensitivity,
1920*1080 pixel, high velocity frames (25 fps)
(accelerometer, altimeter and compass are not used here)
(~300€)

↓ No raw data available: only positions without rms

↑ The images have GPS time - tag

The «cross» equipped on Fiat Doblò car



Pegasem® DMI but used as speed sensor
1024ppr (±0.1 m/s @ 10m/s)
serial interface
(~1500€)

Used sensors: the reasons

We decided to use the less precise GPS data acquired by the Action-Cam with **the aim of testing how far a motion model and a more or less refined filtering technique could bring benefits** in the determination of the vehicle trajectory …

… and in future we want to use the camera (image) orientations in a single filter to improve the accuracy of the trajectory

The DMI (VMI) permits the introduction of velocities into the analysis and also caters for the typical deficiencies of the motion models: e.g. failure to consider stops at traffic lights or zebra crossings.
The DMI permits to "adjust" these models realized for a continuous trend of the vehicle also to a discontinuous trend.

What “truth” is used as a comparison?

A simultaneous precise TMM survey with

INS SBG Ekinox D (nav grade)

- **GNSS internal receiver with two antennas**
- **3 axis accelerometer, 3 Gyro, 3 magnetometer**
- **“Data in” for external odometer, DVL, etc…**
- **Solution in real time or…**
- **row data in internal memory for post processing**

The «reference» trajectory was computed using post processing data and Inertial Explorer ® Novatel Software with mean planimetric st. dev. = 2cm

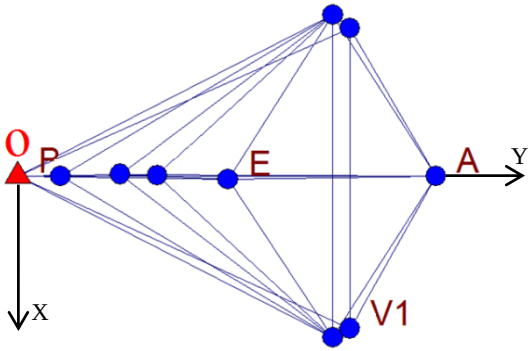
The comparison that we will do here regards only the planimetric results.

Real time performances

0.05° Roll, Pitch
0.05° True Heading
2 cm RTK GNSS Position @ 200 Hz
5 cm Real-time Heave
2.5 cm Delayed Heave
48h Internal Data Logger

Calibrations and synchronizations

Both camera and DMI were calibrated before the survey: the first using the calibration tool of the commercial software Matlab® (Zhang, 1999), instead the second in the Laboratory with a lathe, a speedometer and a chronometer (to compute the scale value).



Since we used many sensors, we computed all lever arm in the frame system of the aluminum "cross" placed on the baggage rack of the car.

The synchronization between DMI (speedometer) and the camera was made using the same GPS time scale of the camera and the DMI connected to INS Ekinox.
(The photo frames are in GPS time)

We could have other low-cost solutions:
e. g .modelling PC time error with GPS time

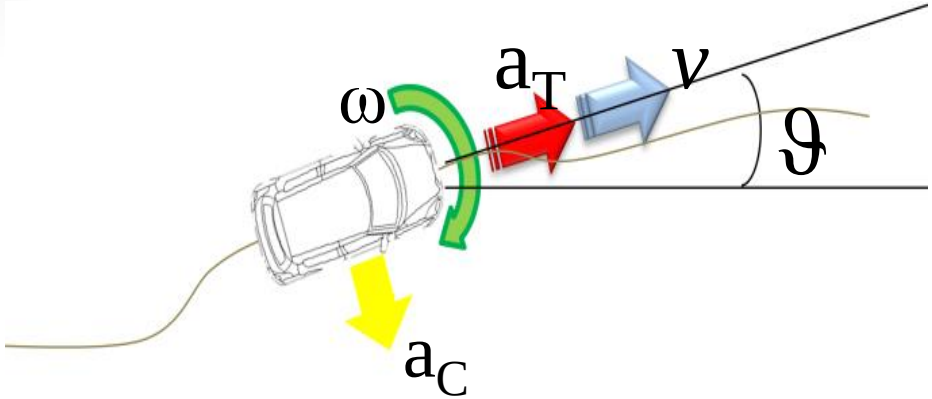
Before the comparison: we compute

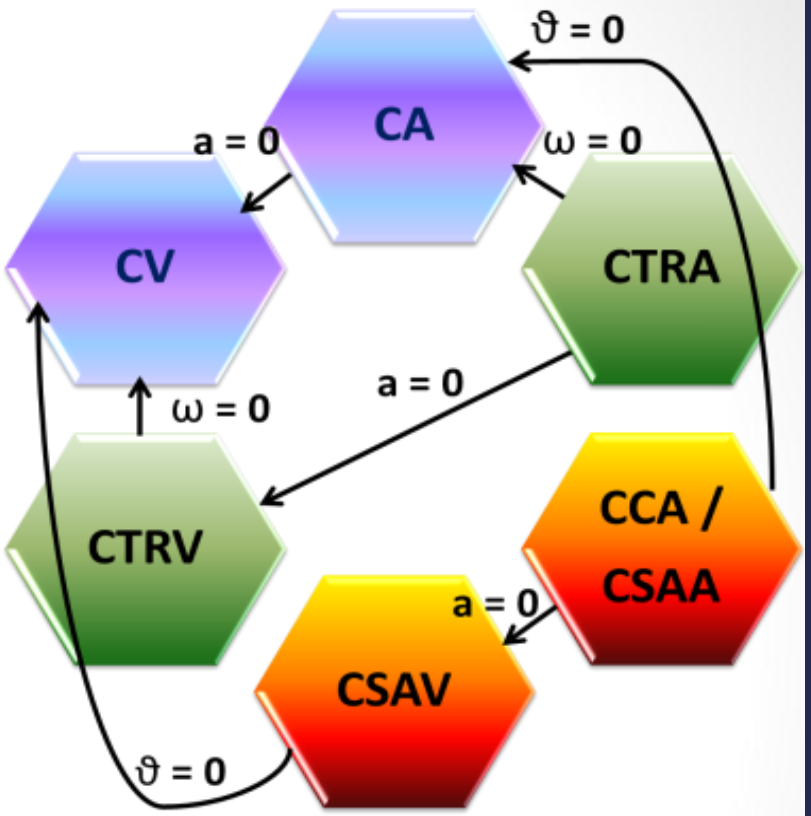
<u>Spatial - time lever arm:</u>

- Camera lever arm
- Odometer (speedometer): data in GPS time, random time more or less @100 Hz
- Lever arm from Ekinox SBG to Garmin camera and speedometer

Lever arms are computed taking in account the use of velocity vector (output of the filter)

Motion Model





USED IN THIS TEST

4 states: **CV**
(constant velocity)

$$\vec{x}(t) = (x, v_x, y, v_y)$$

5 states: **CTRV**
(constant turn rate & velocity)

$$\vec{x}(t) = (x, y, \theta, v, \omega)$$

6 states: **CTRA**
(constant turn rate & aceleration)

$$\vec{x}(t) = (x, y, \theta, v, a, \omega)$$

Motion Model: 4 & 5 states equations

4 states: CV

$$\vec{x}(t) = (x, v_x, y, v_y)$$

$$\vec{x}(t+T) = \begin{pmatrix} x(t) + T \cdot v_x \\ v_x \\ y(t) + T \cdot v_y \\ v_y \end{pmatrix}$$

5 states: CTRV

$$\vec{x}(t) = (x, y, \theta, v, \omega)$$

$$\vec{x}(t+T) = \begin{pmatrix} \frac{v}{\omega} \cdot sin(\omega \cdot T + \theta) - \frac{v}{\omega} \cdot sin(\theta) + x(t) \\ -\frac{v}{\omega} \cdot cos(\omega \cdot T + \theta) + \frac{v}{\omega} \cdot cos(\theta) + y(t) \\ v \\ \omega \cdot T + \theta \\ \omega \end{pmatrix}$$

Measures used: GPS position and tangential velocity data (odometer)

As we can see the state eq. but also the measurement eq. are not linear and linearizable with low accuracy for ω near to zero

Motion Model: 6 state equations

6 states: CTRA $\vec{x}(t) = (x, y, \theta, v, a, \omega)$

Measures used: position and tangential velocity data

$$\vec{x}(t+T) = \begin{matrix} (\; x(t) + \frac{1}{\omega^2}[(v(t)\omega + a\omega T) \cdot sin\big(\omega T + \theta(t)\big) + \ldots \\ +a\, cos\big(\omega T + \theta(t)\big) - v(t)\omega\, sin\big(\theta(t)\big) - a\, cos\big(\theta(t)\big)] \\ y(t) + \frac{1}{\omega^2}[(-v(t)\omega - a\omega T) \cdot cos\big(\omega T + \theta(t)\big) + \cdots \\ \ldots +a\, sin\big(\omega T + \theta(t)\big) + v(t)\omega\, cos\big(\theta(t)\big) - a\, sin(\theta(t))] \\ \omega \cdot T + \theta \\ a \cdot T \\ 0 \\ 0 \end{matrix}$$

For brevity we not report here the stochastic assumptions used to calibrate the filter

The Filters: KF, EKF**, UKF, (A)UKF**

- KF, EKF: As you know, the principal limitations are the necessary linearization and the normality of the measures
- UKF: it is possible to remove the first hypothesis and maintain the second, with minor error consequences

The Unscented Kalman Filter is based on the unscented transformation, that is a mechanism for the propagation of the mean and the covariance, using non-linear transformations and additional states… suitably distributed around the average values

- (A)UKF (Augmented) use a greater number of states. This greater number of states is due to the fact that the state and measure equations are considered as not linear respect to the noise

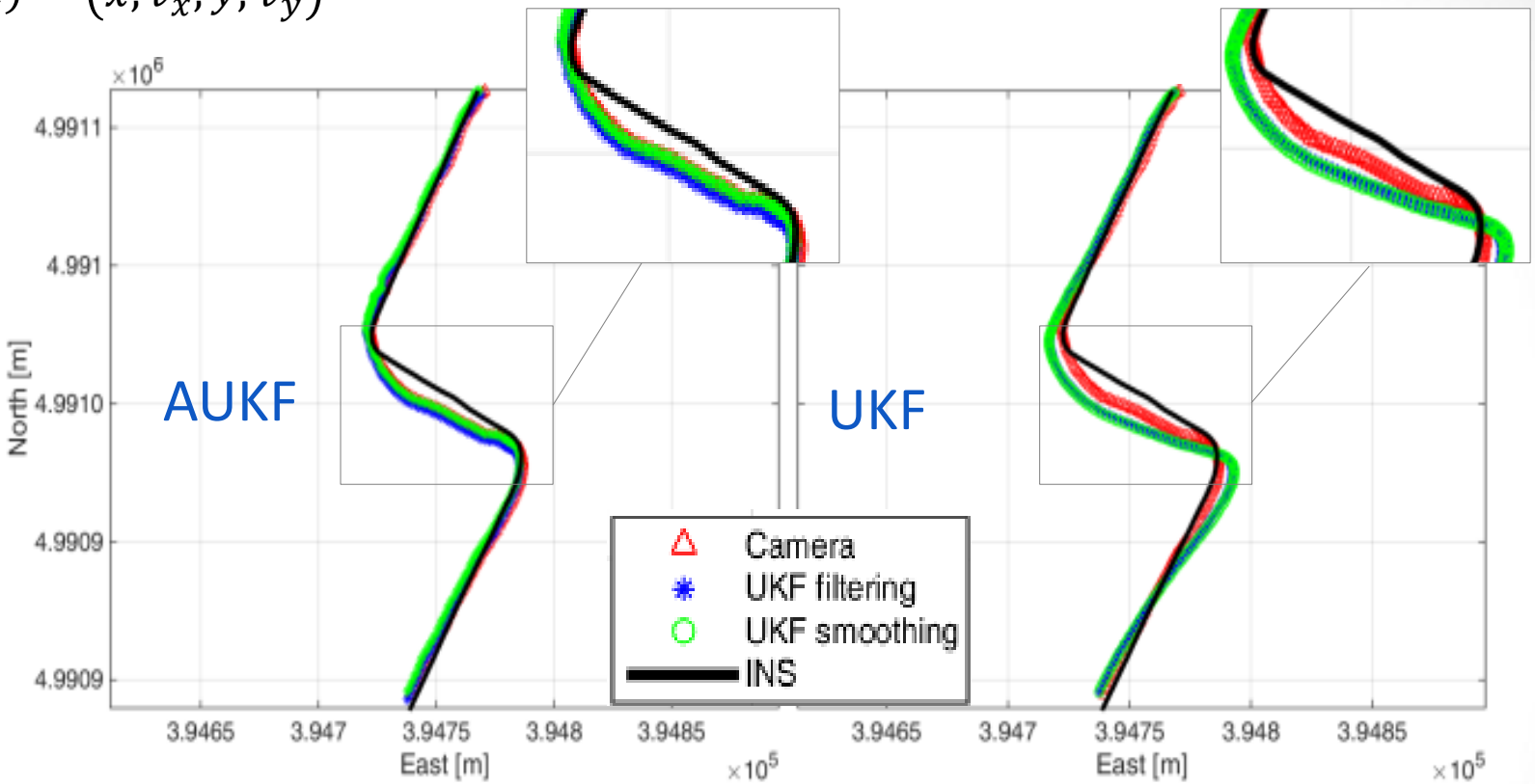
The software built and the tests run

We built a software in Matlab® using:

- The motion model with 4, 5, 6 states
- UKF and AUKF (results with EKF not listed here)
- A part of trajectory (in green)
- The comparisons took into account the lever arm before a match to a reference trajectory

2D errors w.r. to the reference trajectory

$\vec{x}(t) = (x, v_x, y, v_y)$

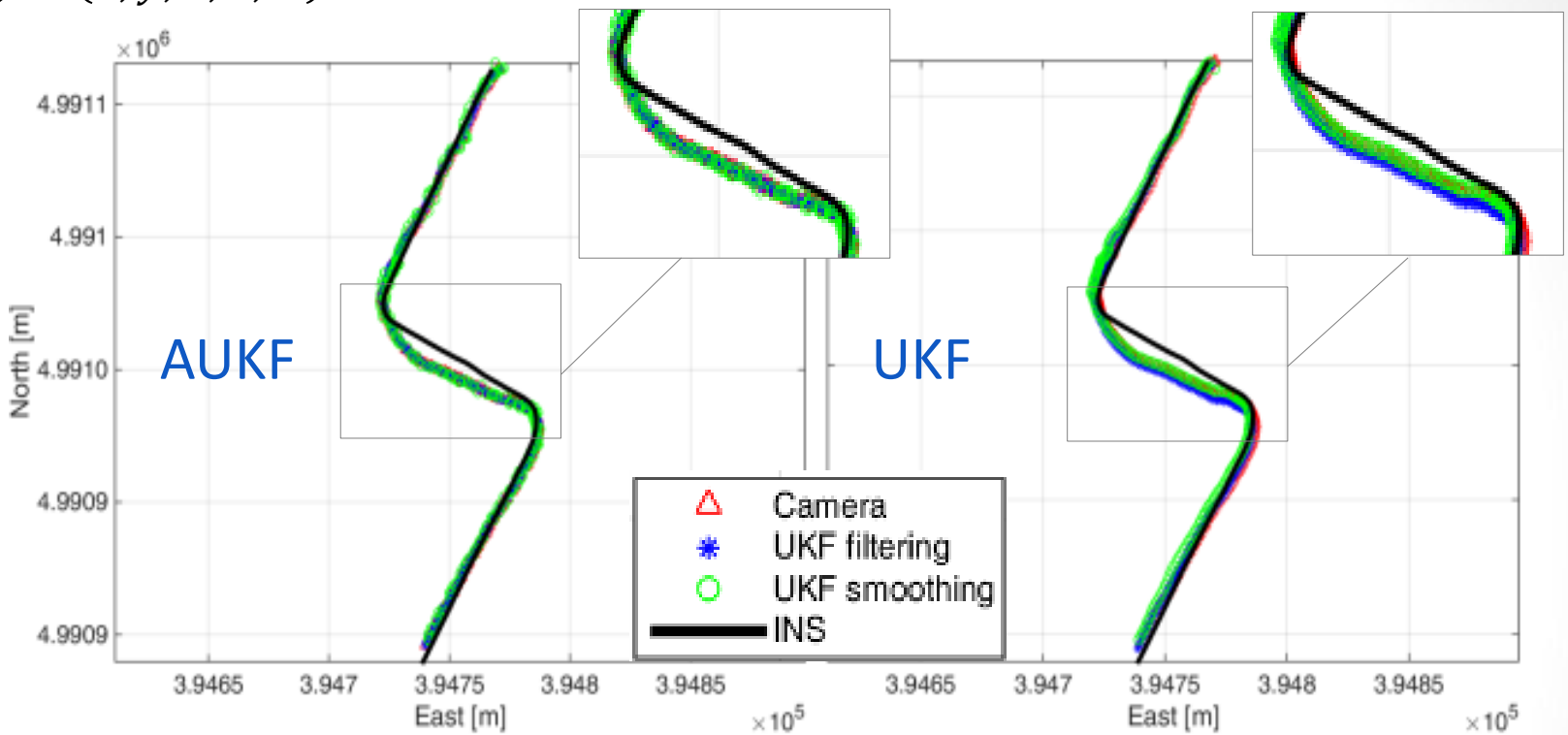


Without filter in this obstructed path the mean errors are about 10 m

4 states	AUKF (smooth)		UKF (smooth)	
2D Error(m)	Δ_{MEAN}	σ	Δ_{MEAN}	σ
	4.5	1.8	8.0	4.2

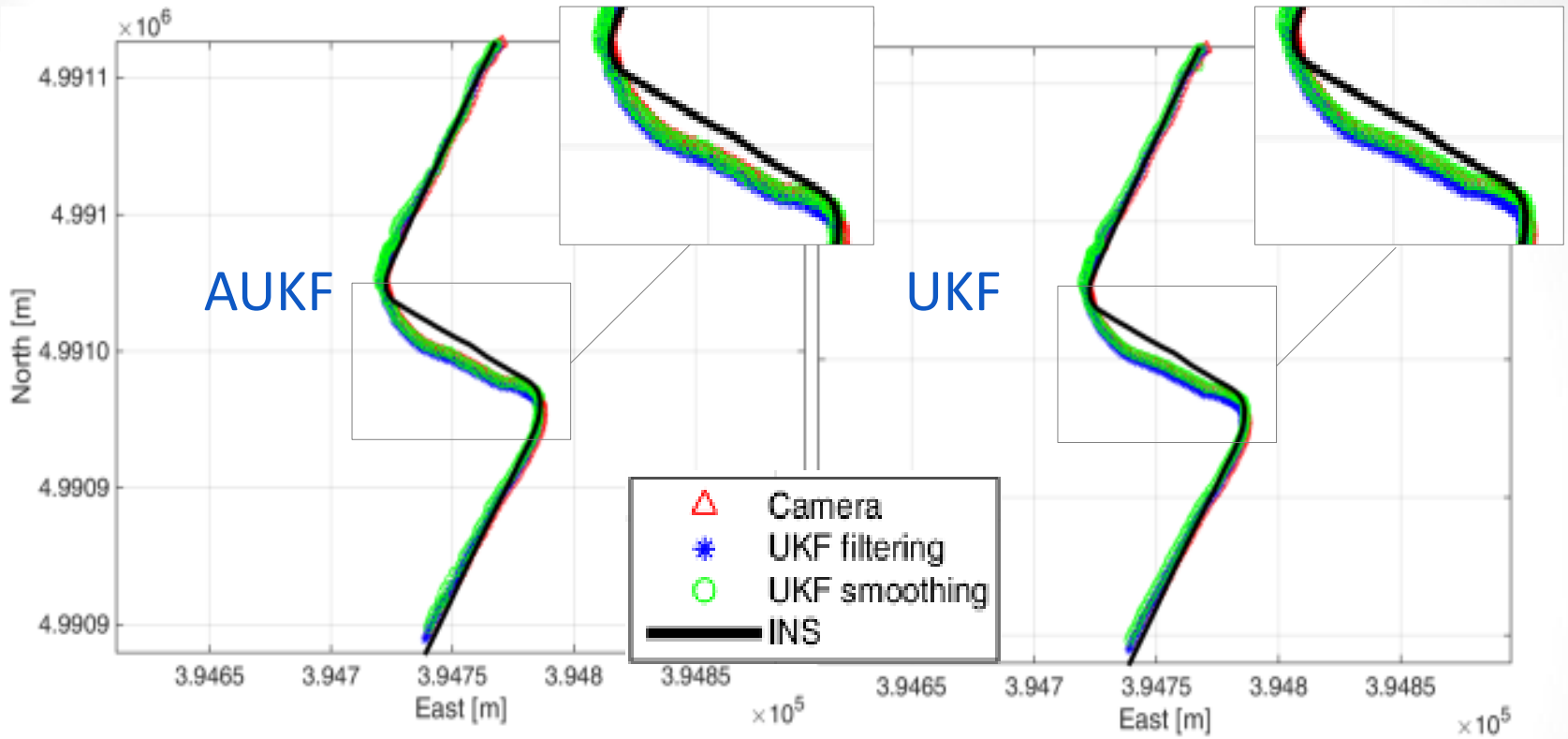
2D errors w.r. to the reference trajectory

$\vec{x}(t) = (x, y, \theta, v, \omega)$



5 states	AUKF (smooth)		UKF (smooth)	
2D Error(m)	Δ_{MEAN}	σ	Δ_{MEAN}	σ
	4.5	2.2	5.3	1.6

2D errors w.r. to the reference trajectory



The best results, especially regarding the sigma

6 states	AUKF (smooth)		UKF (smooth)	
2D Error(m)	Δ_{MEAN}	σ	Δ_{MEAN}	σ
	4.1	1.4	4.6	1.7

First loose integration with photogrammetry

Currently we have introduced in the commercial software (Photoscan) the real values of position and accuracy obtained by the UKF 5 states filter. (using the photos after camera calibration)

The aim of first approach is only to see how much this poor integration is useful to improve the our solution.

… but we are currently in a phase of building of a software able to integrate GPS data with the orientation parameters of two seq. models.

The «loose» results are quite the same of using know camera position with fixed accuracy of 2m.
These are the residual errors w.r. of the reference trajectory.

3D apriori Camera accuracy [m]	$\Delta 2D_{MEAN}$ INS-CASE [m]	$\Delta 2D\ \sigma$ INS-CASE [m]
1	2.44	1.42
2	2.57	2.11
3	5.63	3.20

Conclusions and perspectives

- The best results are obtained using 6 states with AUKF
- Speed data (entering in 5 & 6 states) improve results especially for UKF (not augmented)
- By tighter approach with photogrammetry can be expected in future to reduce the planimetric error at least to 50%

FUTURE WORK AND OUTLOOK

1. Use uBlox to put raw data (e.g. Doppler) in the filter
2. Use AUKF and photogrammetric relative orientation of two contiguous model to improve the results (heading)
3. Use of bases of two contiguous model in the filter as quasi-distance information (distance with unkn. scale factor)
4. Use ancillary data (accelerometer) in ZUPT points to define Z axis in photogrammetry.

… but I undertand

Q: Why do not use a tool similar to a «ball mouse» in 2D odometry?

Thank You for your attention

Thank You Giorgio !