A Complete Solution for Seamless Indoor/Outdoor Navigation

Jan Zwiener, M.Sc.; Axel Lorenz, M.Sc.; Prof. Dr.-Ing. Reiner Jäger

Tuesday 24 February

Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Content

Overview

- Indoor/Outdoor Navigation
- Georeferenced building (ETRF89)

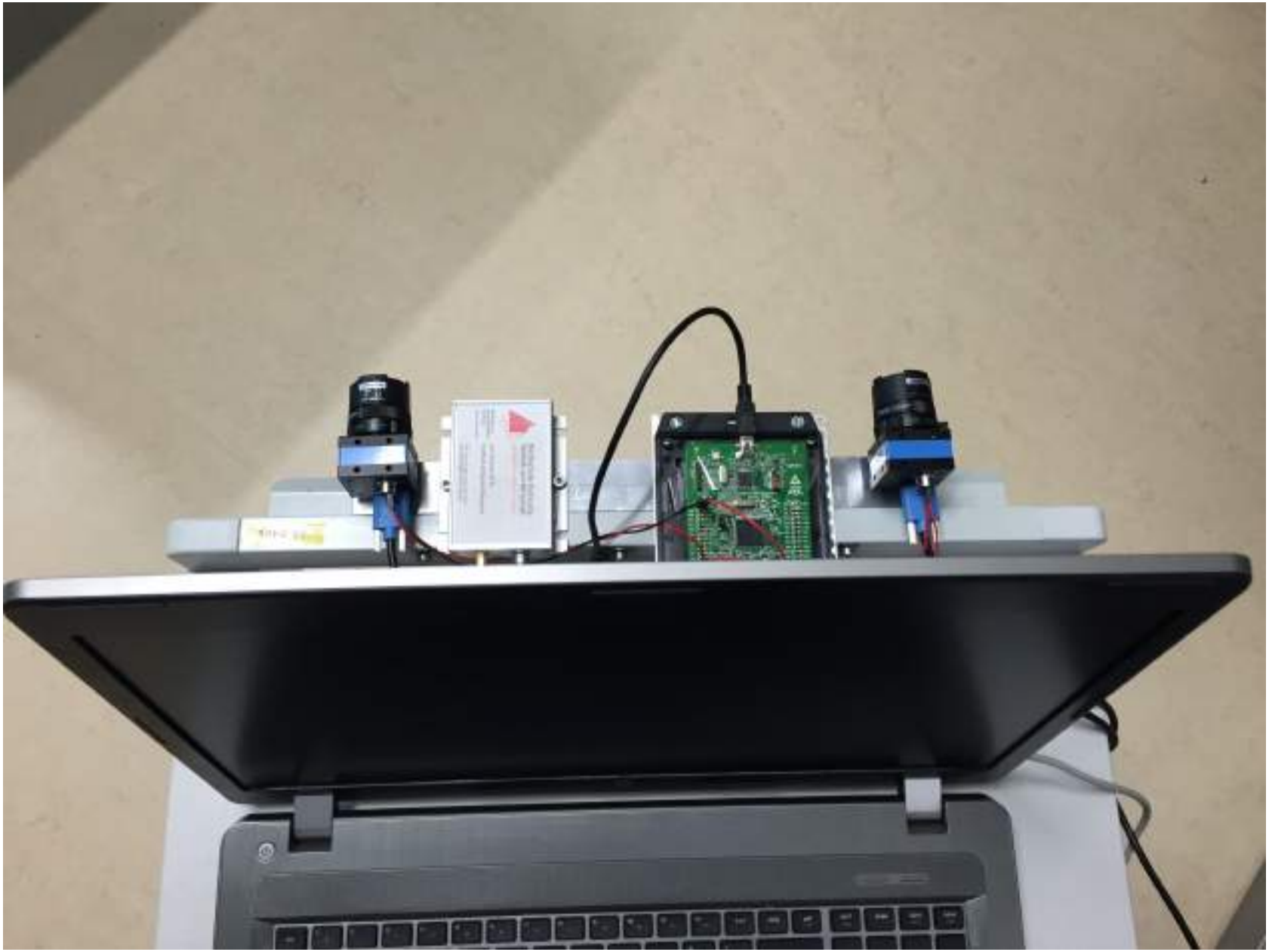
System Setup

- IMU (Accelerometer and Gyroscope)
- Pressure sensor
- Two global shutter cameras
- L1 GNSS receiver
- Synchronization controller (STM32F407 microcontroller)

Approach

- $\vec{x}$ Absolute values from GNSS and marker
- $\dot{\vec{x}}$ Relative changes by visual odometry
- $\ddot{\vec{x}}$ Acceleration by IMU

System Overview



IMU

- In-house development
- STM32F407 32bit Microcontroller ARM Cortex M4
- RTOS ChibiOS (Stable Branch 2.6.6)
- Bosch BMA280 Accelerometer
- InvenSense MPU9150 Gyroscope
- ms5611 Barometer
- MEMSIC MXR9500G/M
- u-blox receiver on-board with timepulse pin

Stereo Camera Pair

- DFK 42BUC03 (`theimagingsource.com`)
- CMOS sensor
- USB 2.0
- Fixed baseline
- Global shutter
- Baseline length 28.5 cm
- 1280x960 pixel
- 25 fps
- Trigger input (137 μs delay)
- Synchronization controller (STM32F407 microcontroller)

Georeferencing

Example KARL Station (Karlsruhe):

System	**E UTM** (m)	**N UTM** (m)	**h** (m)
`KARL ITRF_365-2014`	32456947.9928	5428873.0007	182.9012
`KARL ETRF89`	32456947.5137	5428872.5151	182.8841
Δ	0.479 m	0.485 m	0.017 m

Software Stack

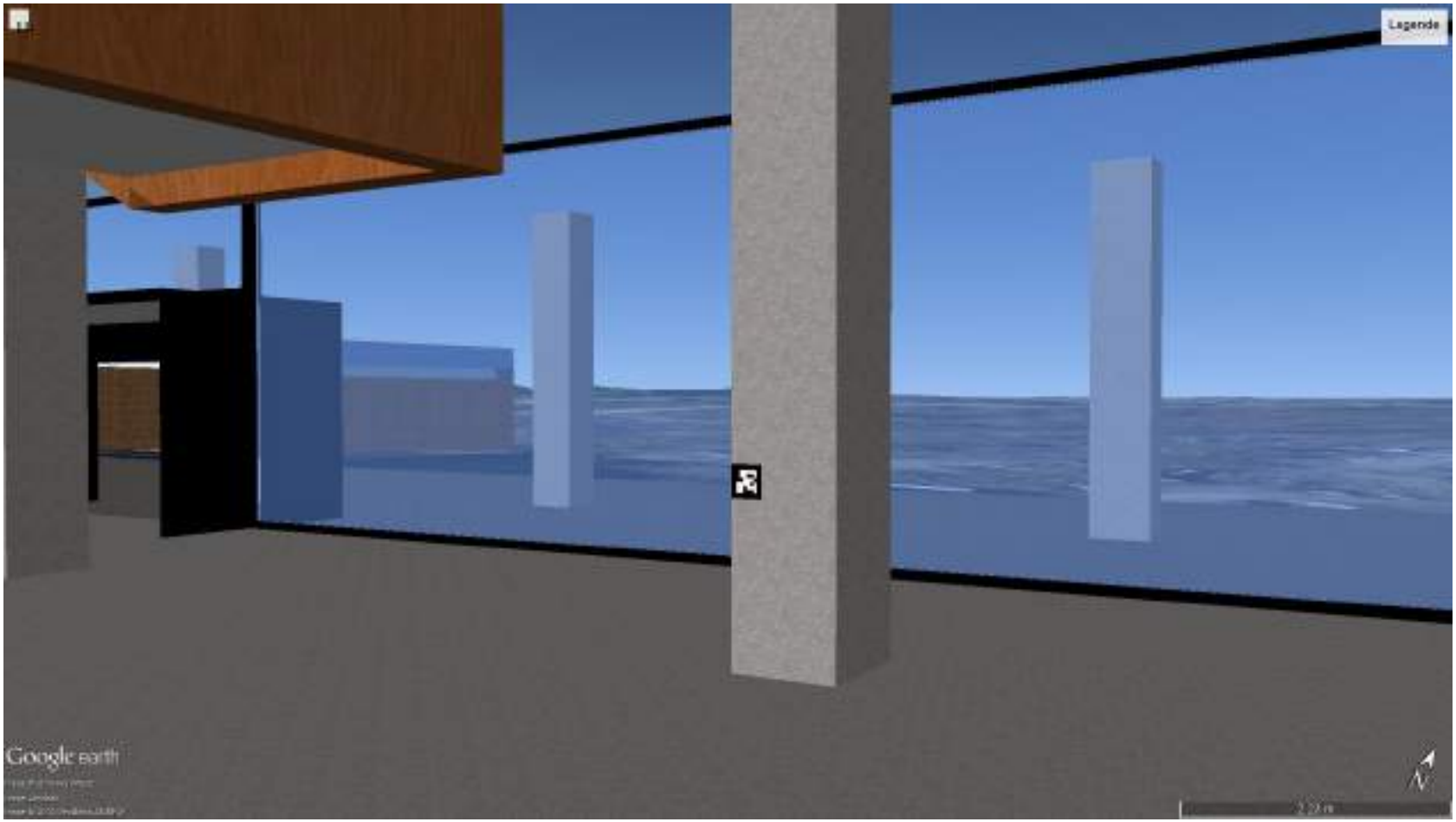
- In-house developed navigation library
- Parameter estimation package
- Runs on Windows, Linux, Mac OS X and embedded (32-bit) ARM systems
- EIGEN C++ math library
- OpenCV 2.4 for computer vision tasks
- ARToolKitPlus (Wagner et al. 2007)[1]
- Visual Odometry with libviso2 (Geiger et al.)[2]

[1] `https://handheldar.icg.tugraz.at/artoolkitplus.php` for marker detection

[2] `http://www.cvlibs.net/software/libviso/`

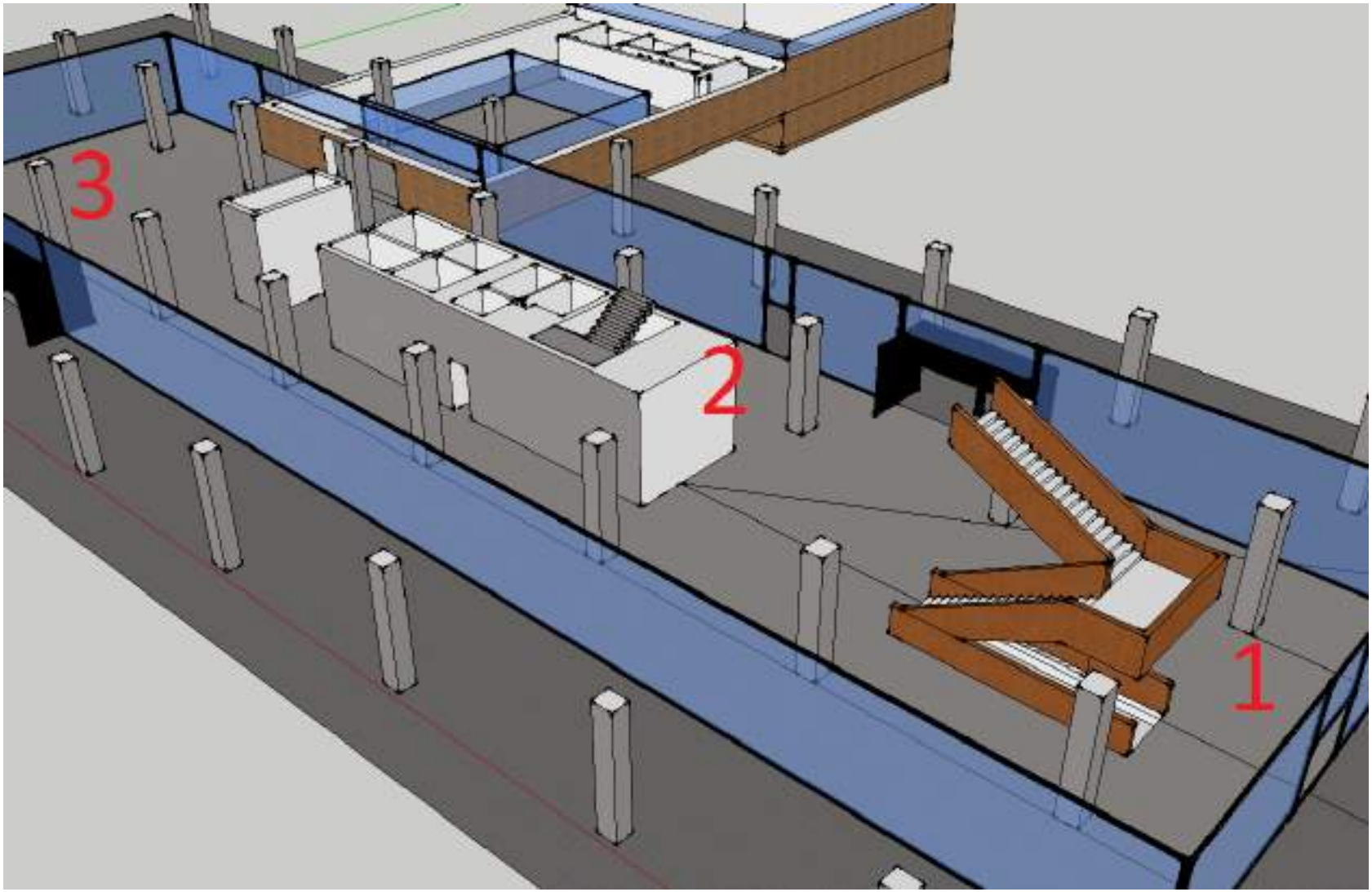
ARToolKitPlus BCH id-marker

ARToolKitPlus BCH id-marker



ARToolKitPlus BCH id-marker

ARToolKitPlus BCH id-marker



Marker database

Stored items:

- Marker id (up to 4096 ids)
- Position as ECEF coordinates (ETRF89)
- Marker size 19 cm
- Orientation of marker as matrix: $\mathbf{R}_b^n$

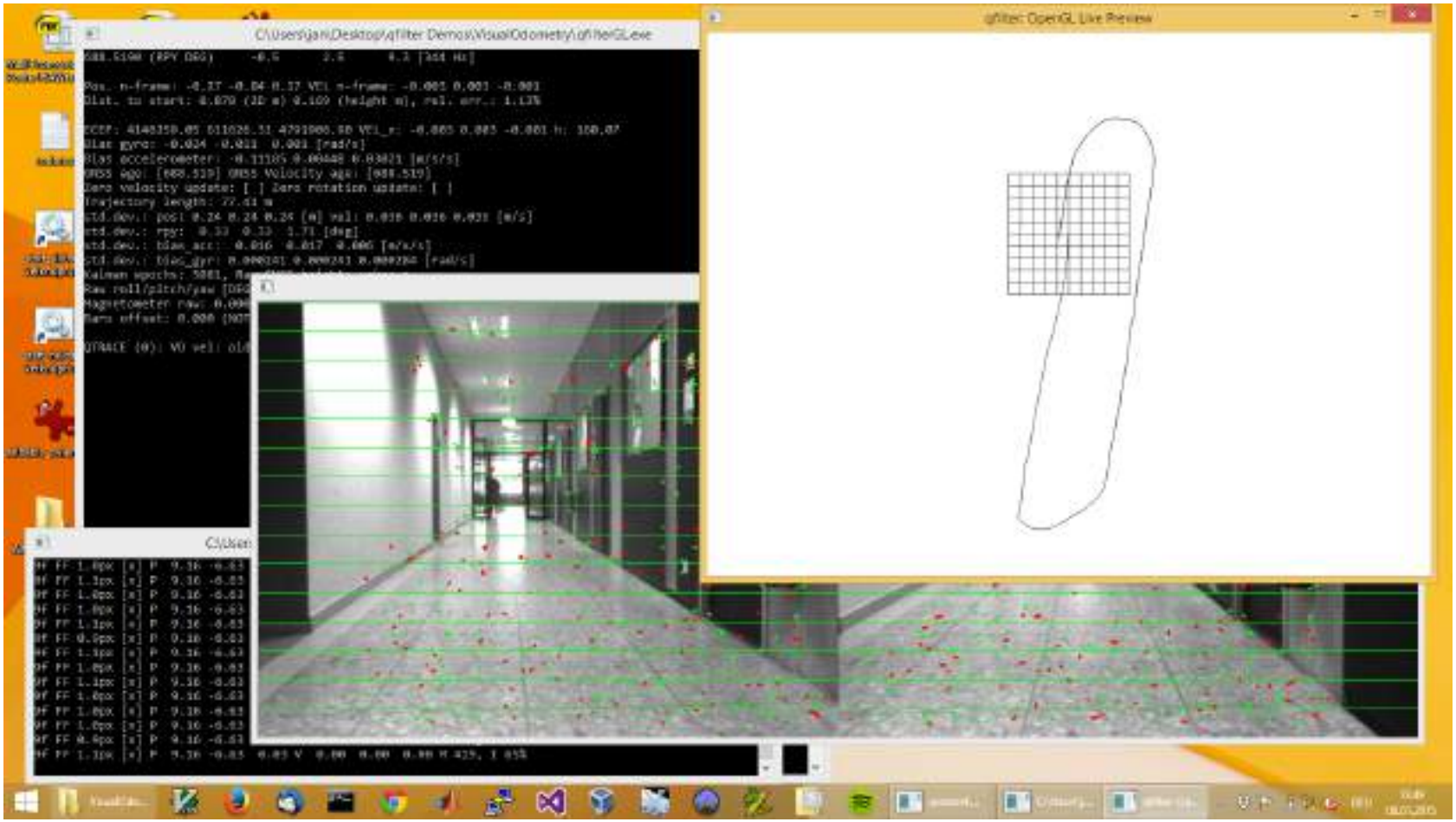
$$p_{\text{Camera}}^{\text{ETRF89}} = \mathbf{R}_n^e \cdot \mathbf{R}_b^n \cdot p_{\text{Camera}}^{\text{b}} + p_{\text{Marker}}^{\text{ETRF89}} \qquad (1)$$

b = Camera body frame
n = Navigation frame
Covariance of $p_{\text{Camera}}^{\text{b}}$ obtained by bundle adjustment.

Camera to IMU offset from pre-calibration in IMU body frame.

Navigation module: marker detector

Navigation module: visual odometry



Navigation module: visual odometry

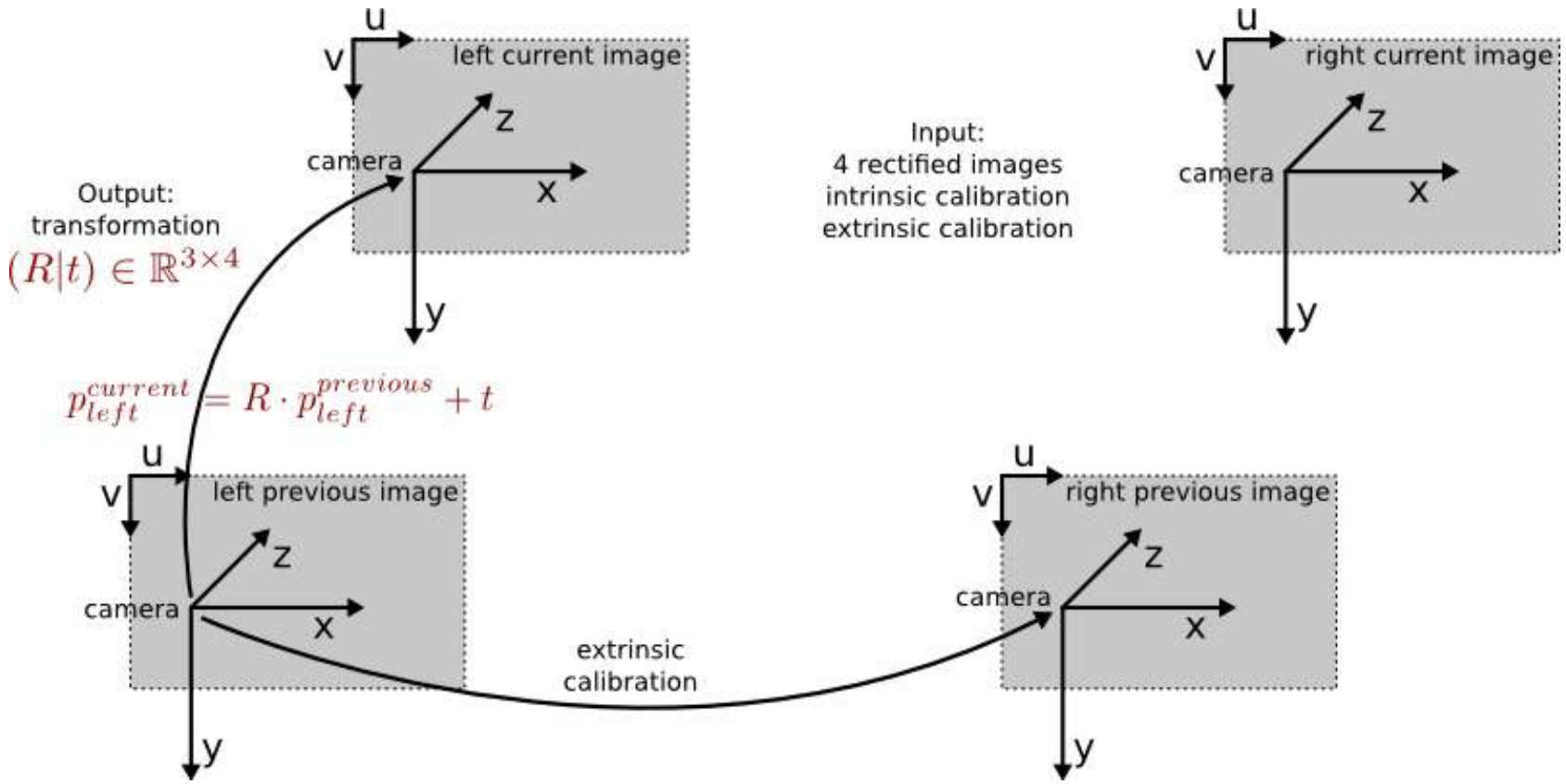


Image: Andreas Geiger `http://www.cvlibs.net/software/libviso/`

Visual odometry: relative state updates

- Translation from epoch n to $n+1$: $\vec{t}_n^{n+1}$
- Rotation from epoch n to $n+1$: $\mathbf{R}_n^{n+1}$

$$\vec{X} = \left[\vec{x}_n^T \, \vec{x}_{n+1}^T\right]^T \tag{2}$$

$$\mathbf{Q}_{XX} = \begin{pmatrix} \mathbf{Q}_n & \mathbf{Q}_1 \cdot \mathbf{F}^T \\ \mathbf{F} \cdot \mathbf{Q}_1 & \mathbf{Q}_{n+1} \end{pmatrix} \tag{3}$$

where **F** is the system dynamic relationship between the two epochs[3].

[3] see LYNEN et al. (2013): A Robust and Modular Multi-Sensor Fusion Approach Applied to MAV Navigation. Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS).

Navigation module: System Dynamics

$$\frac{d}{dt}\begin{pmatrix} \Delta\vec{\mathrm{p}} \\ \Delta\vec{\mathrm{v}} \\ \Delta\vec{\psi} \\ \Delta\vec{\mathrm{b}}_{\mathrm{a}} \\ \Delta\vec{\mathrm{b}}_{\omega} \end{pmatrix} = \begin{pmatrix} 0 & \mathrm{I} & 0 & 0 & 0 \\ 0 & 0 & [f^n \times] & -R_b^n & 0 \\ 0 & 0 & 0 & 0 & -R_b^n \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta\vec{\mathrm{p}} \\ \Delta\vec{\mathrm{v}} \\ \Delta\vec{\psi} \\ \Delta\vec{\mathrm{b}}_{\mathrm{a}} \\ \Delta\vec{\mathrm{b}}_{\omega} \end{pmatrix} + \mathrm{B} \cdot \vec{w}$$

$$\mathrm{B} \cdot \vec{w} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -R_b^n & 0 & 0 & 0 \\ 0 & -R_b^n & 0 & 0 \\ 0 & 0 & \mathrm{I} & 0 \\ 0 & 0 & 0 & \mathrm{I} \end{pmatrix} \cdot \begin{pmatrix} \vec{\mathrm{n}}_{\mathrm{a}} \\ \vec{\mathrm{n}}_{\omega} \\ \vec{\mathrm{n}}_{\mathrm{b}_{\mathrm{a}}} \\ \vec{\mathrm{n}}_{\mathrm{b}_{\omega}} \end{pmatrix}$$

Additional simple 1D Kalman filter to estimate the barometric height offset.

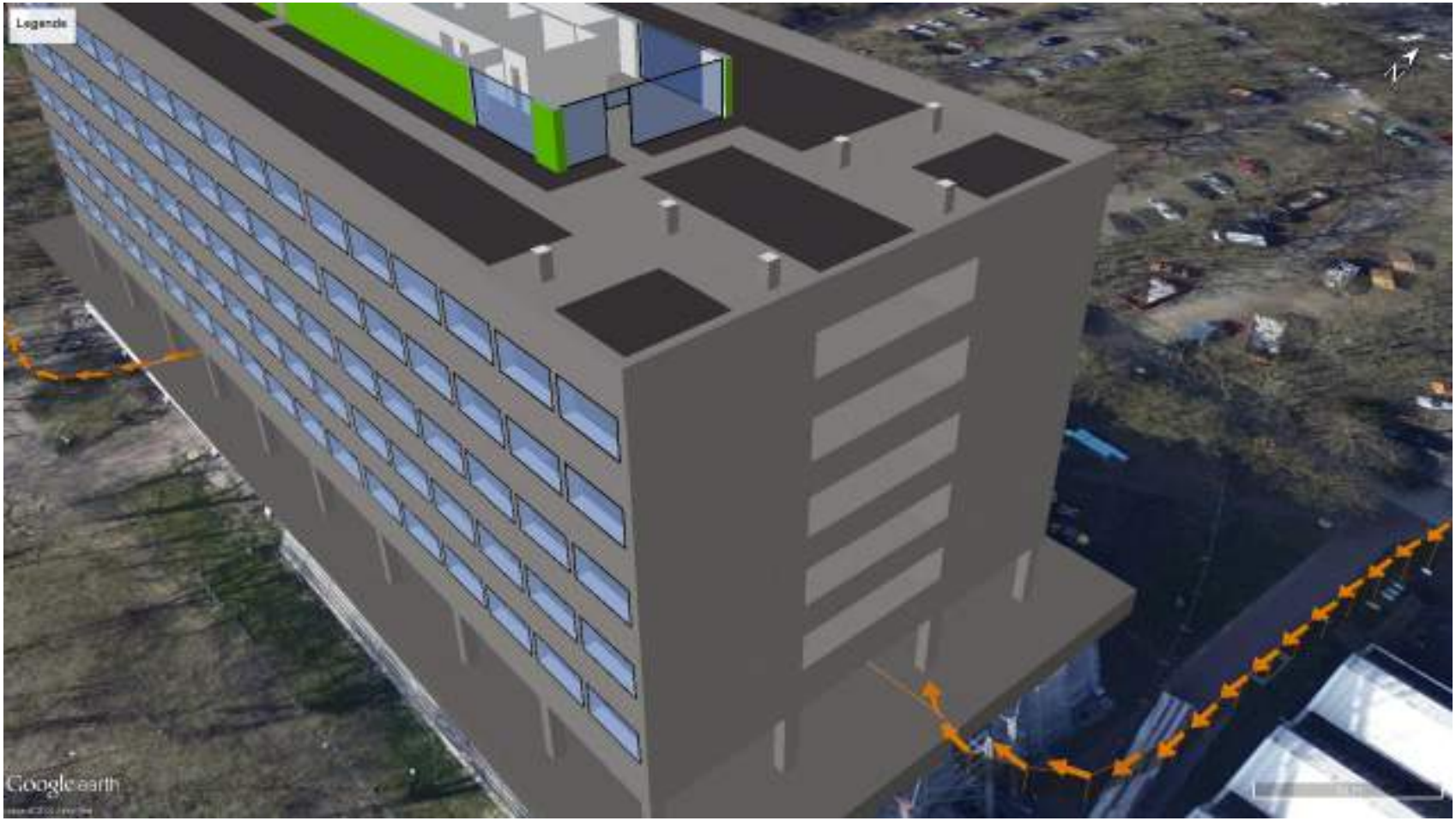
Time Synchronization

- Microcontroller is connected to IMU and u-blox.
- u-blox timepulse is connected to Microcontroller GPIO.
- Camera trigger is connected to GPIO.
- Microcontroller clock is synchronized to GPS time.
- Every measurement has a GPS timestamp (GPS week, TOW).

Example run

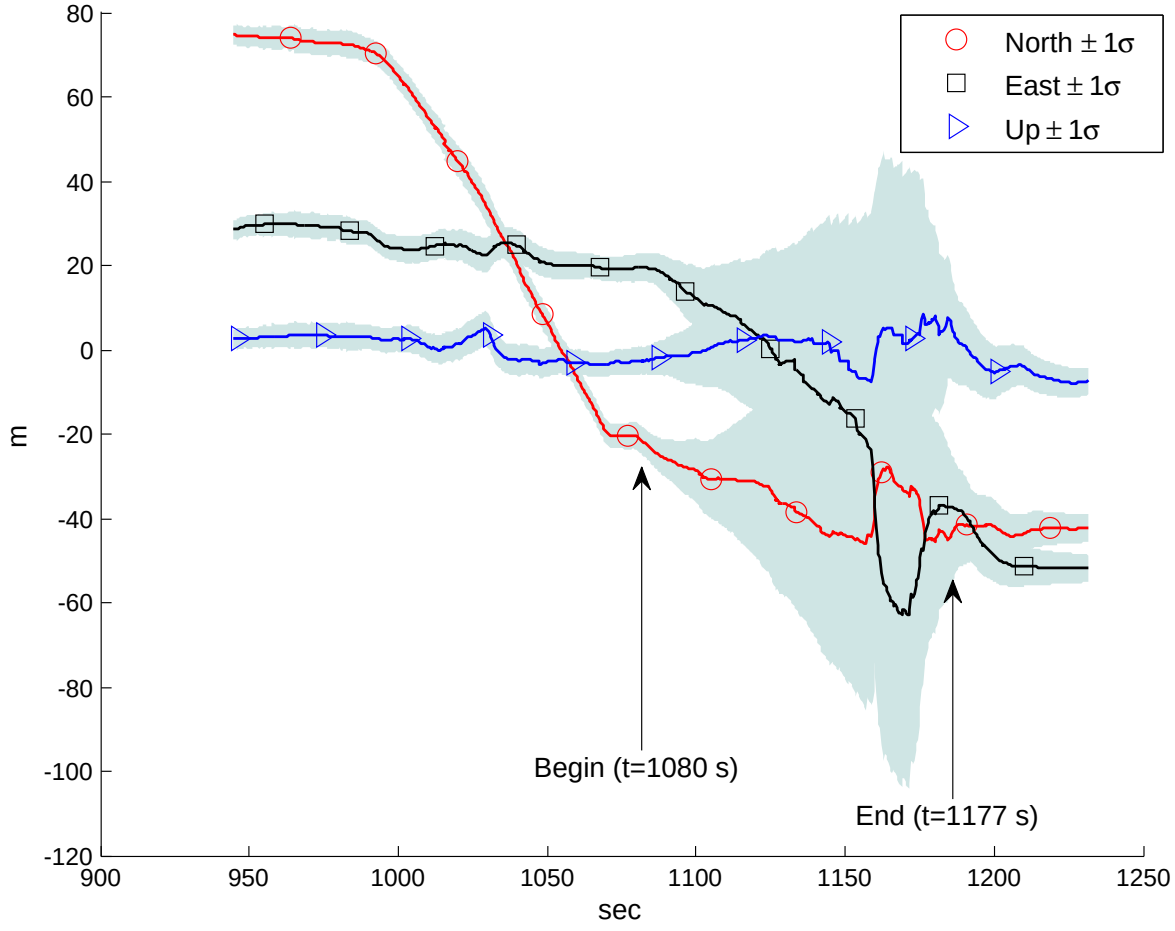
Example run

Example run



Example run

u-blox M8N receiver reported accuracy



We enter the building at approx. t=1080 s.

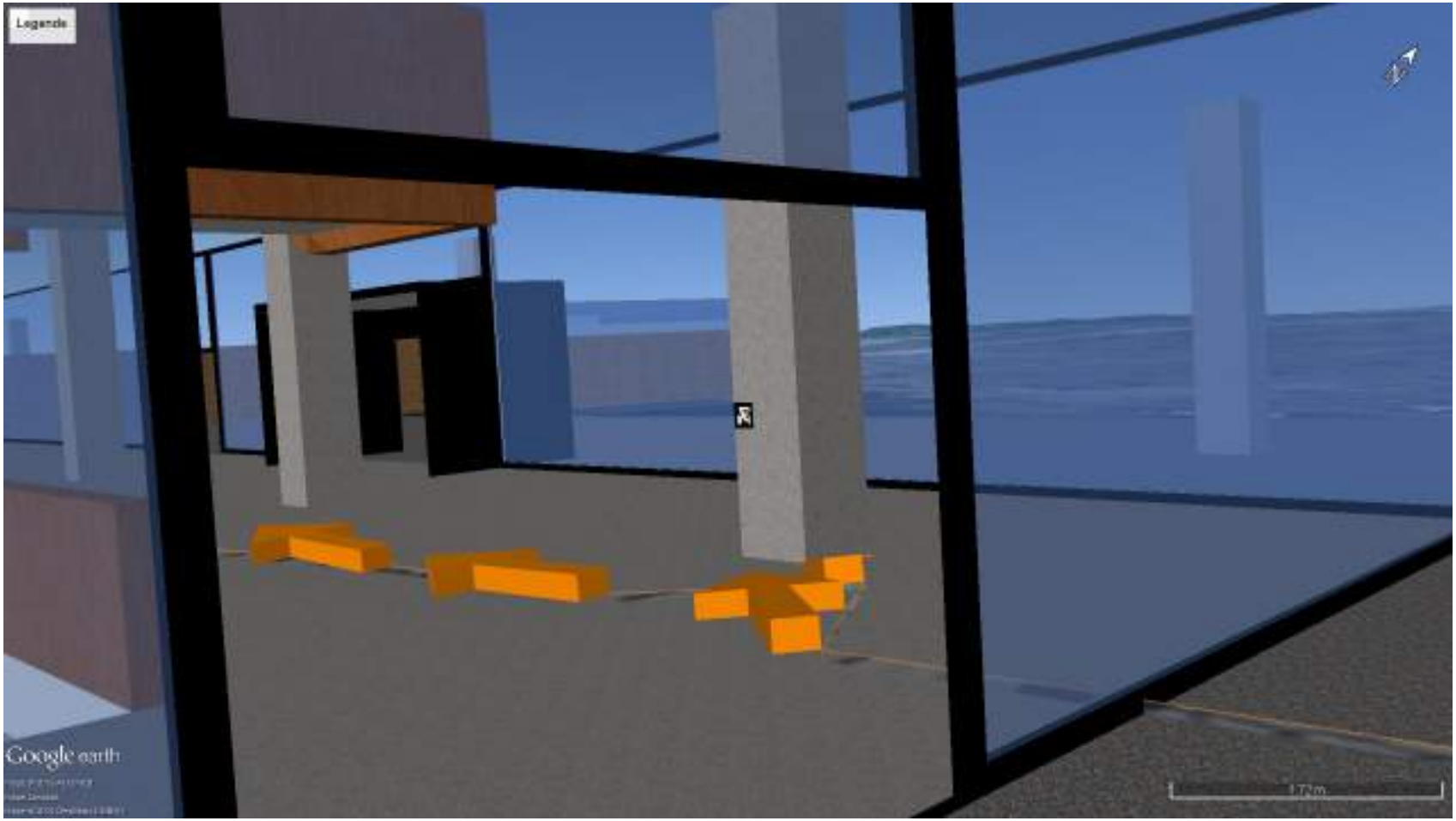
u-blox M8N input

Filtered u-blox M8N input:

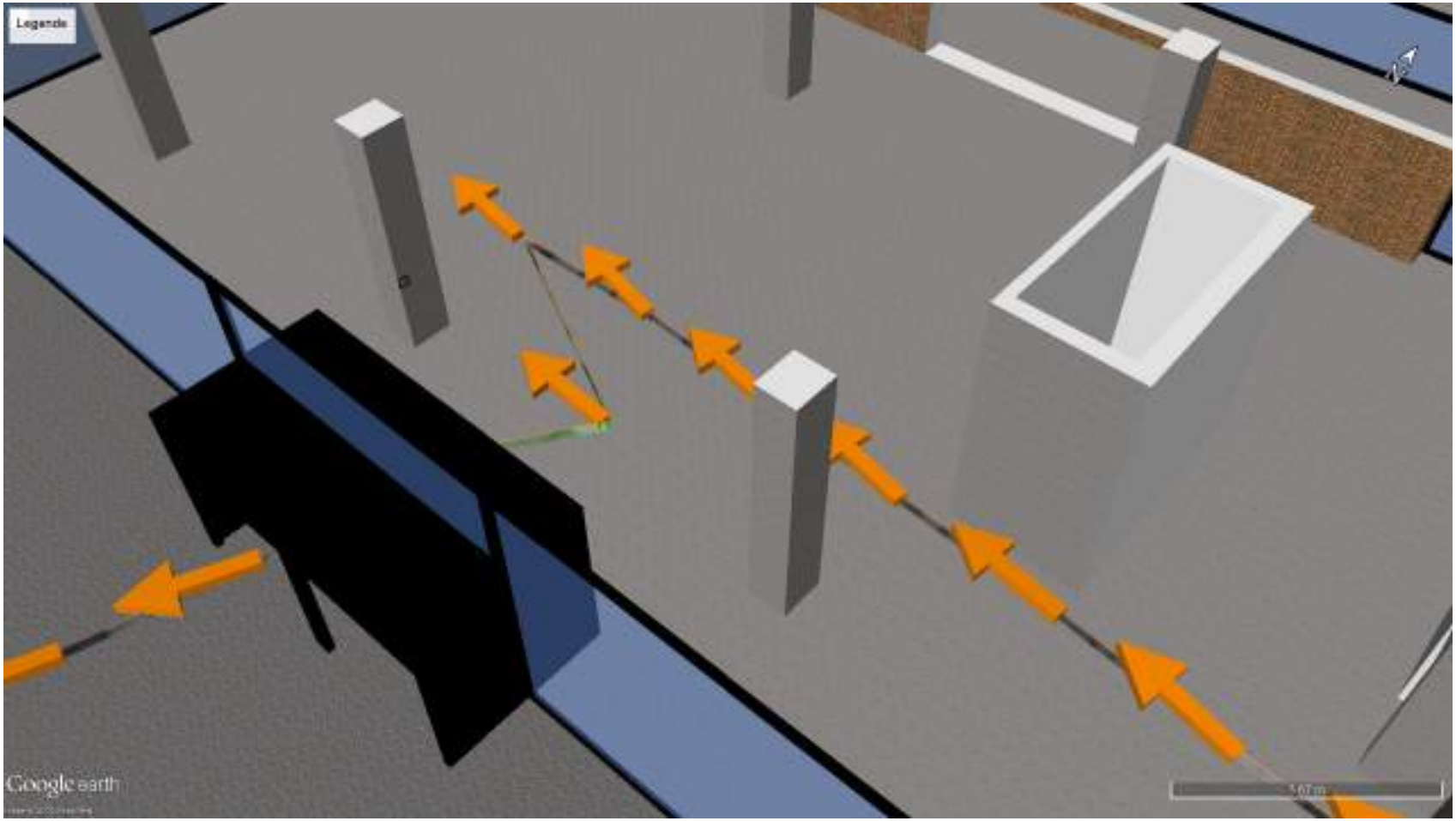
- Approx. transformation to ETRF89
- Filtered if reported speed accuracy is not good enough (0.3 m/s)
- Filtered if reported position accuracy is not good enough (8.0 m)


Overview
Legende
Start
Google earth
20 m

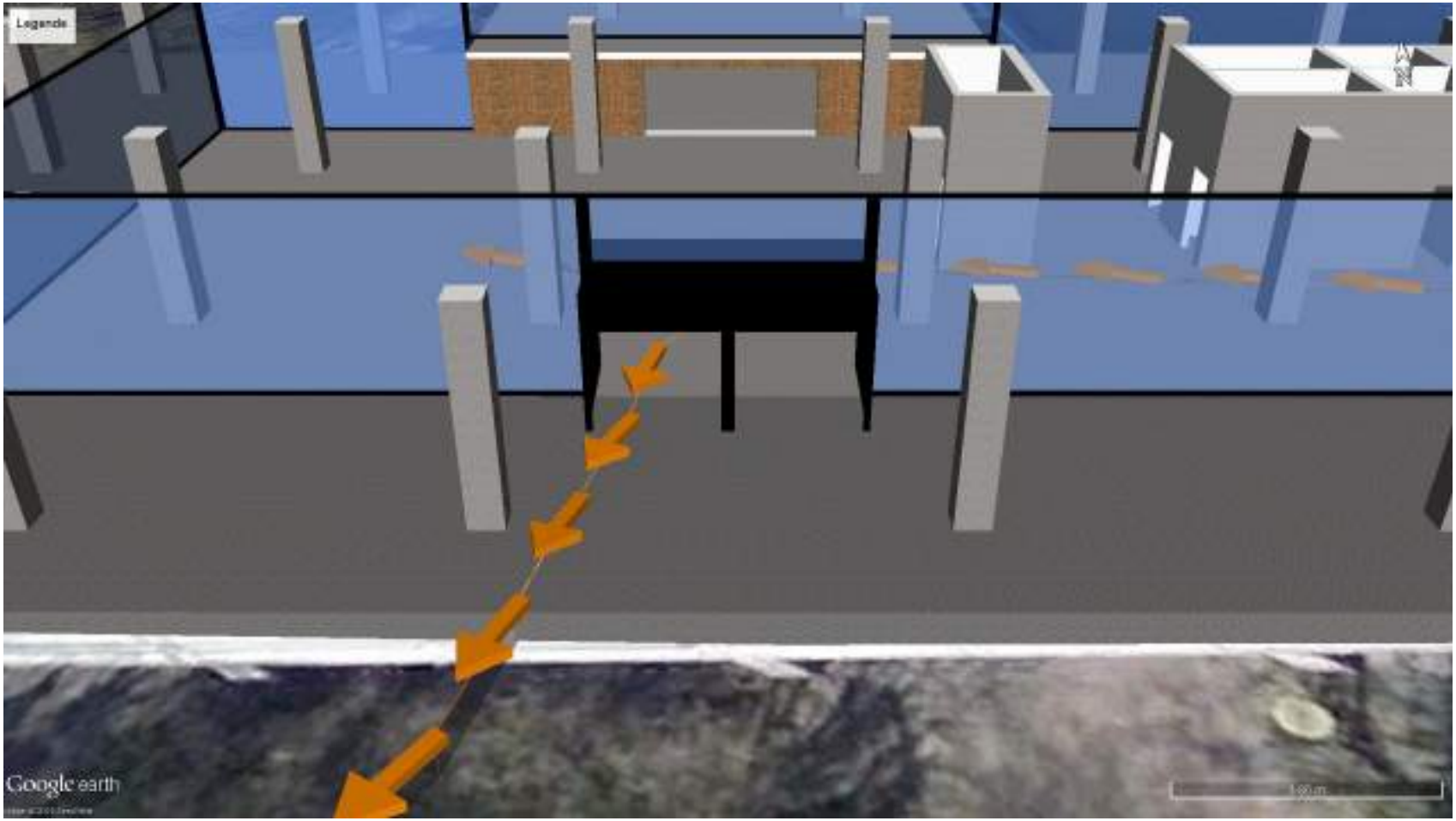
Entering the building



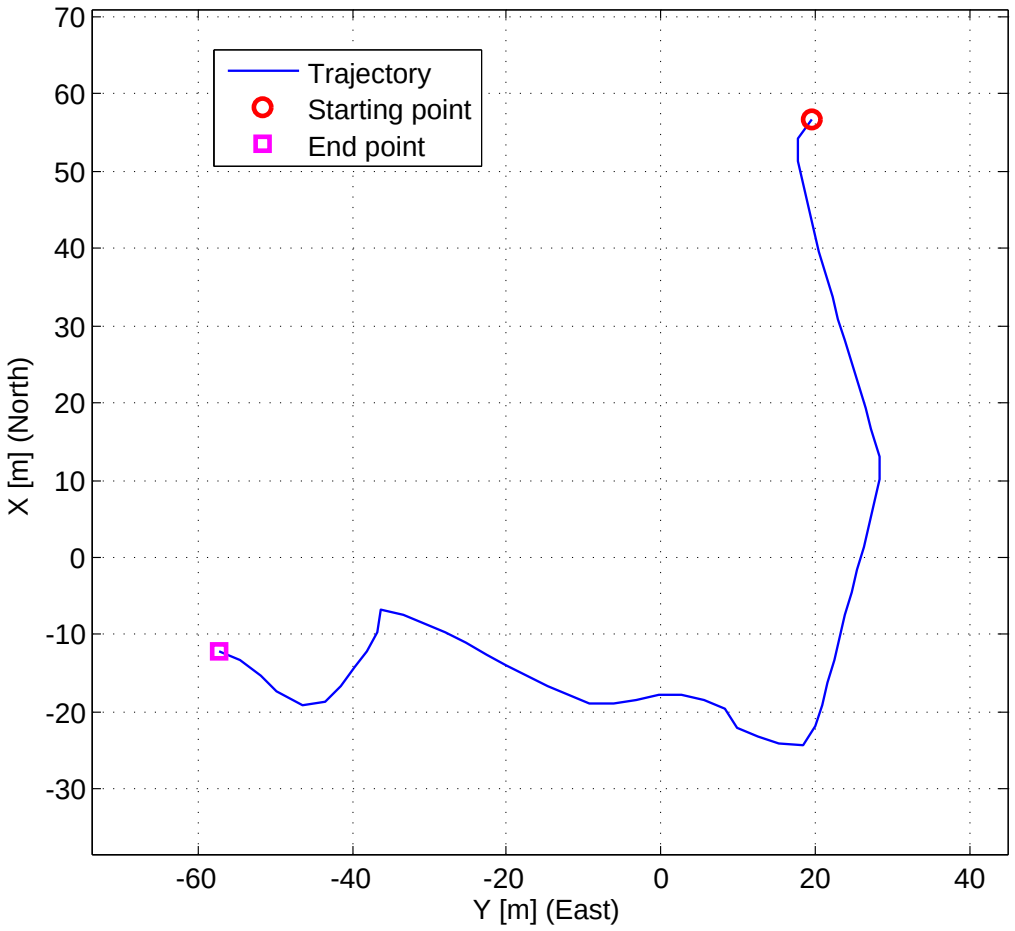
Indoor drift by visual odometry and IMU



Leaving the building



Visual Odometry and IMU only

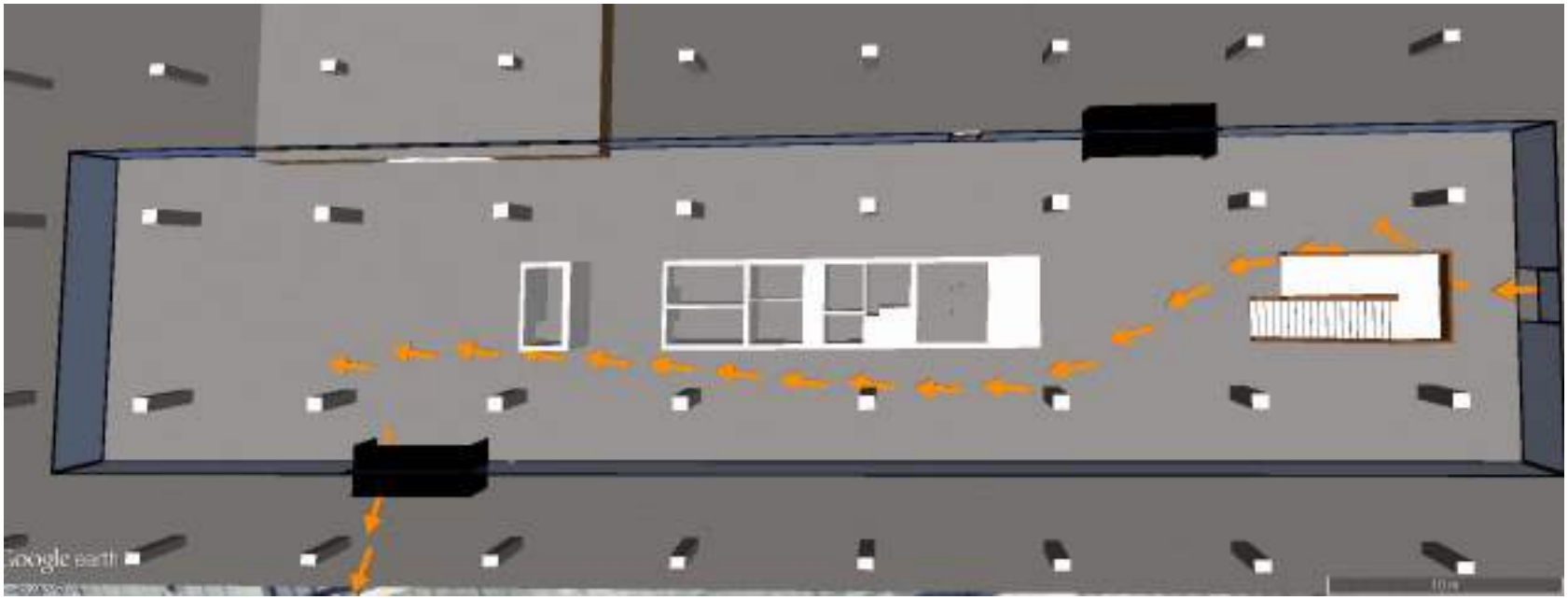


No GNSS (approximated initial position) and no marker correction.

Visual Odometry and IMU only

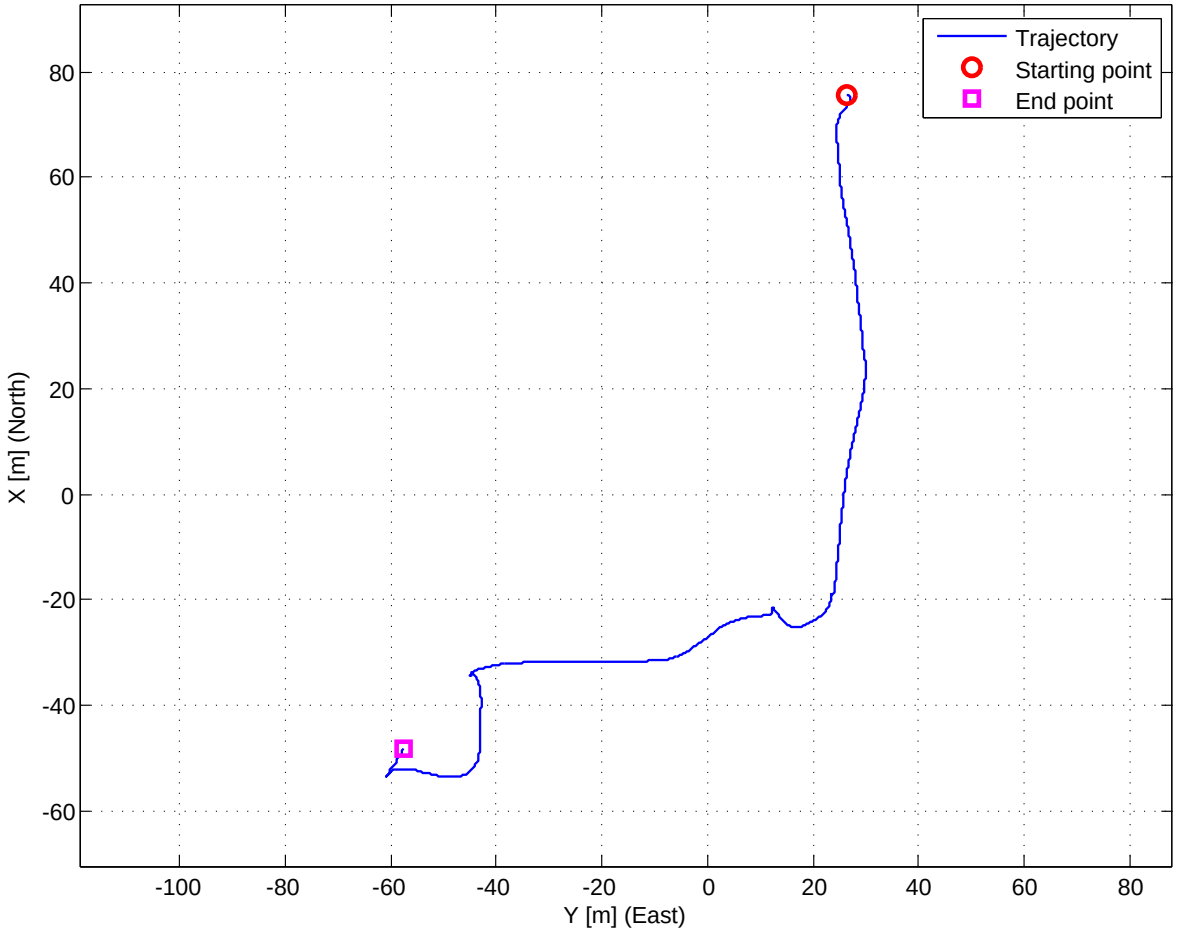
The same trajectory in Google Earth.

Visual Odometry and IMU only



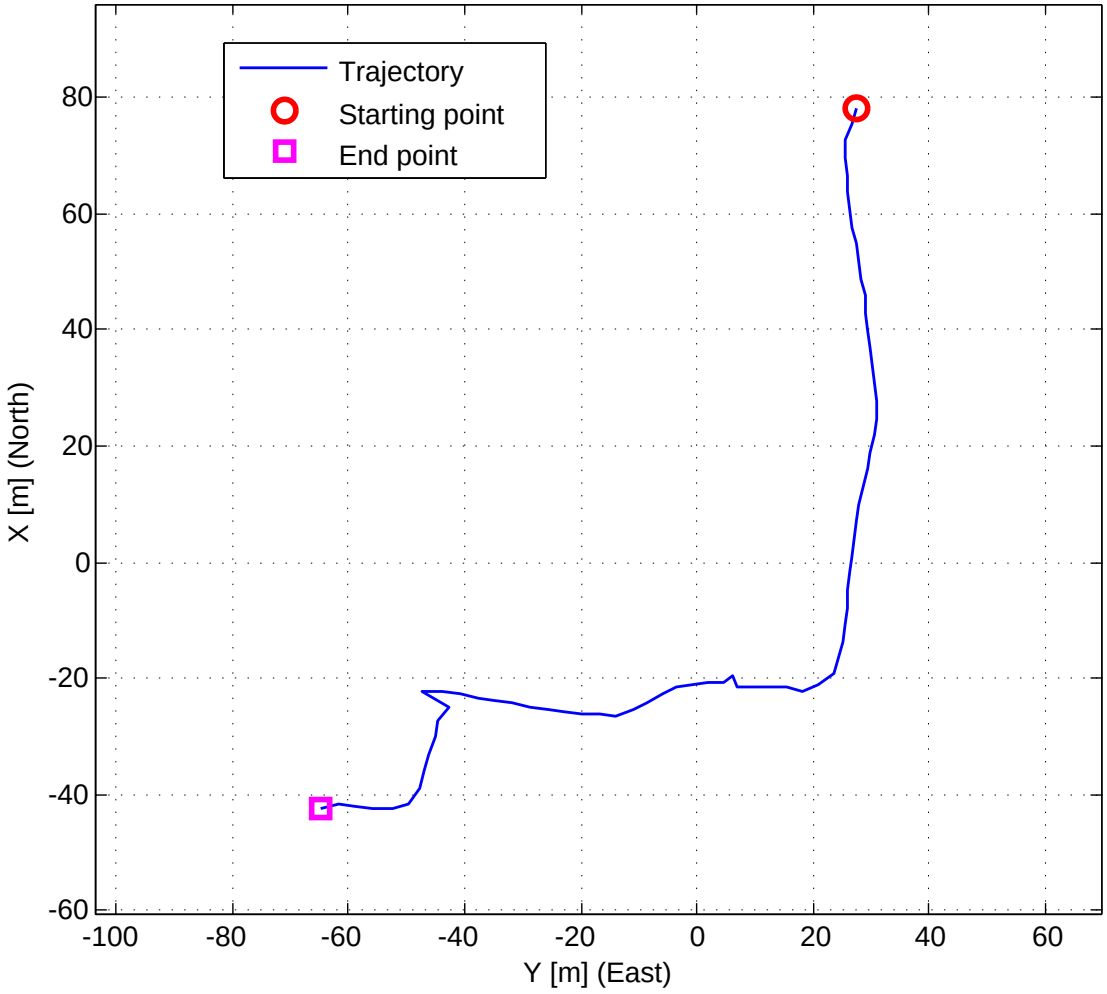
The same trajectory with a known start and orientation.

Visual Odometry, GNSS and IMU only



With GNSS, Visual Odometry and IMU but no marker.

VO, GNSS, IMU and Marker



With GNSS, Visual Odometry, Marker and IMU.

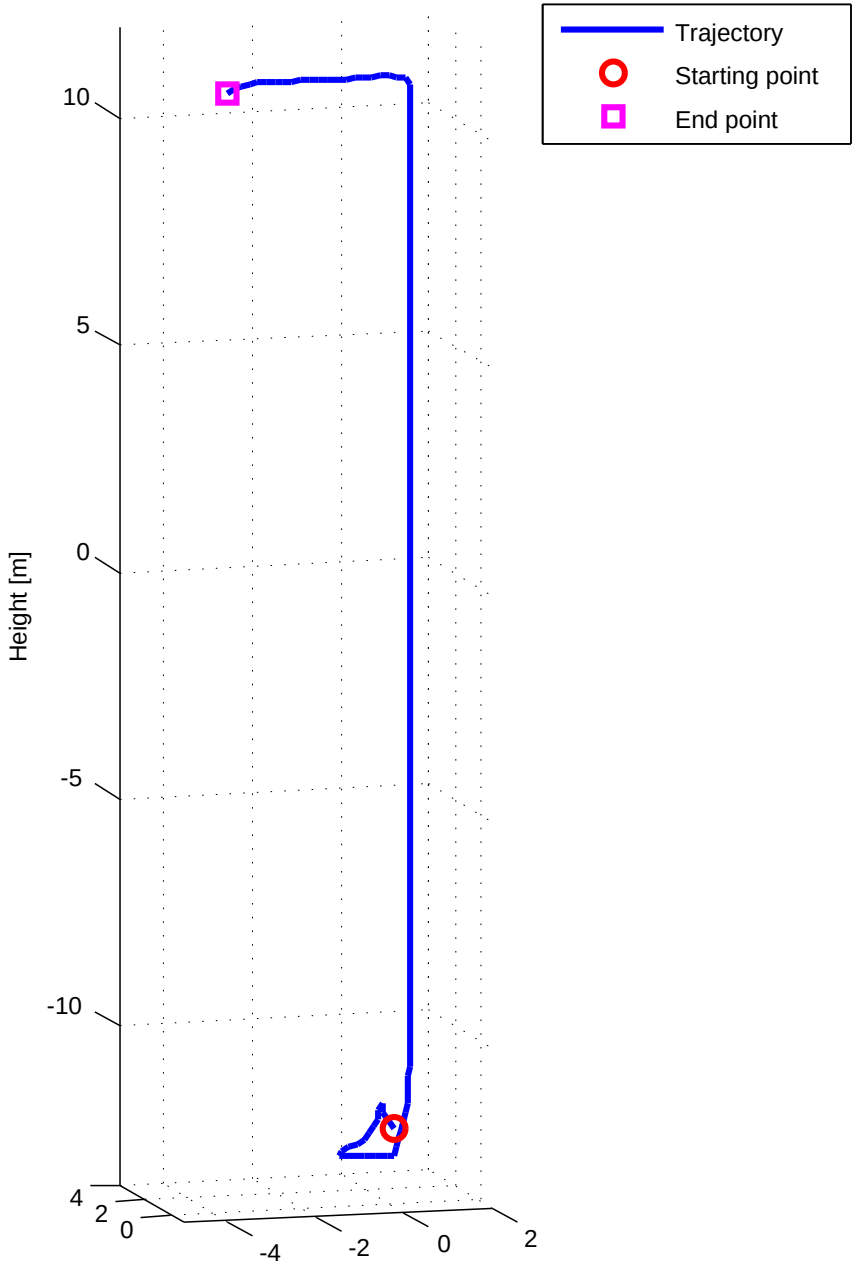
Elevators

- Visual Odometry reported speed vertical speed is wrong
- Large mirrors in elevators are a problem for the Visual Odometry
- IMU and barometer are working fine in the elevator

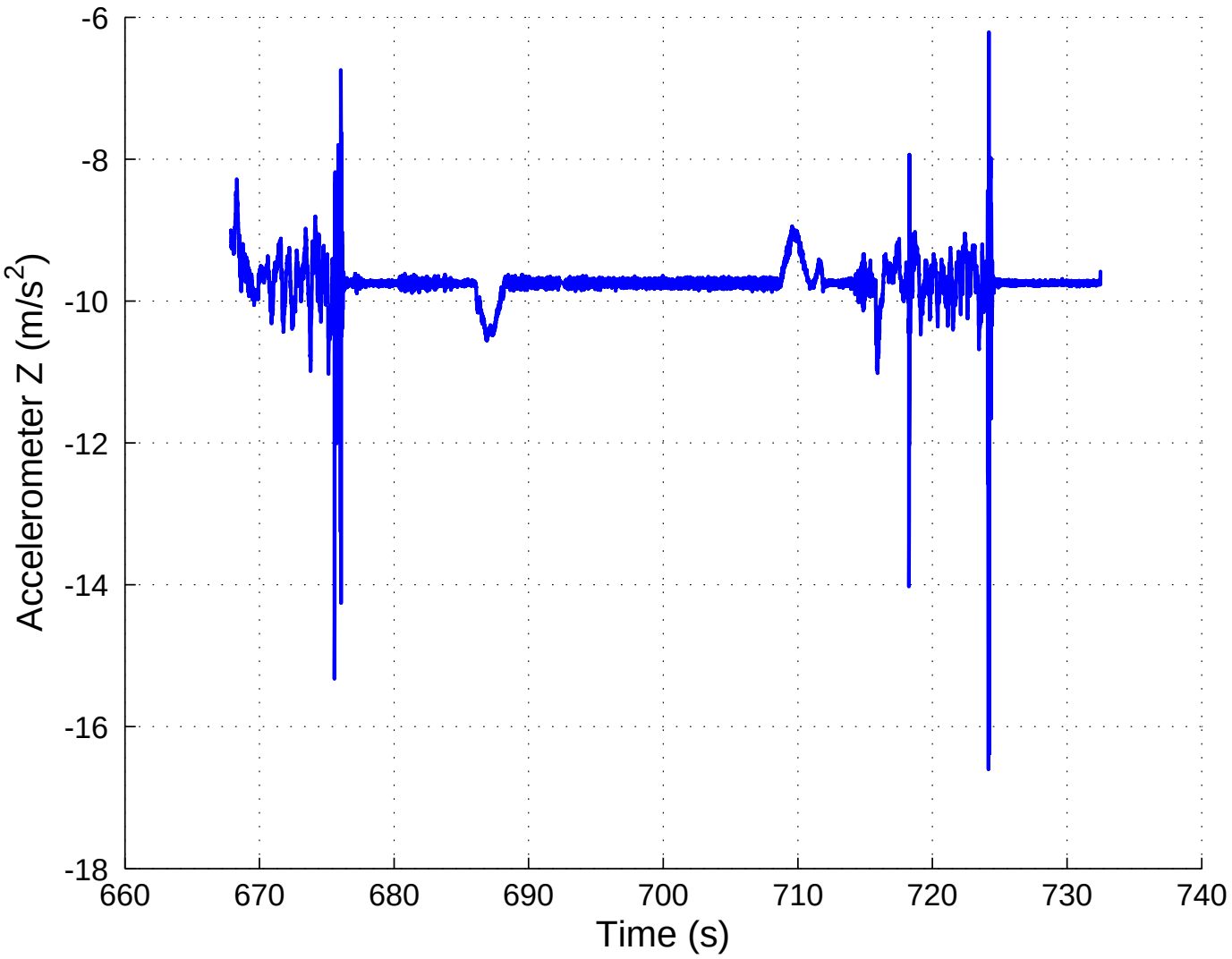
Possible solution:

- Only use horizontal movement from Visual Odometry while in an elevator
- Special marker to indicate elevators
- Visual Odometry limits horizontal drift
- Vertical component estimated by IMU and barometer

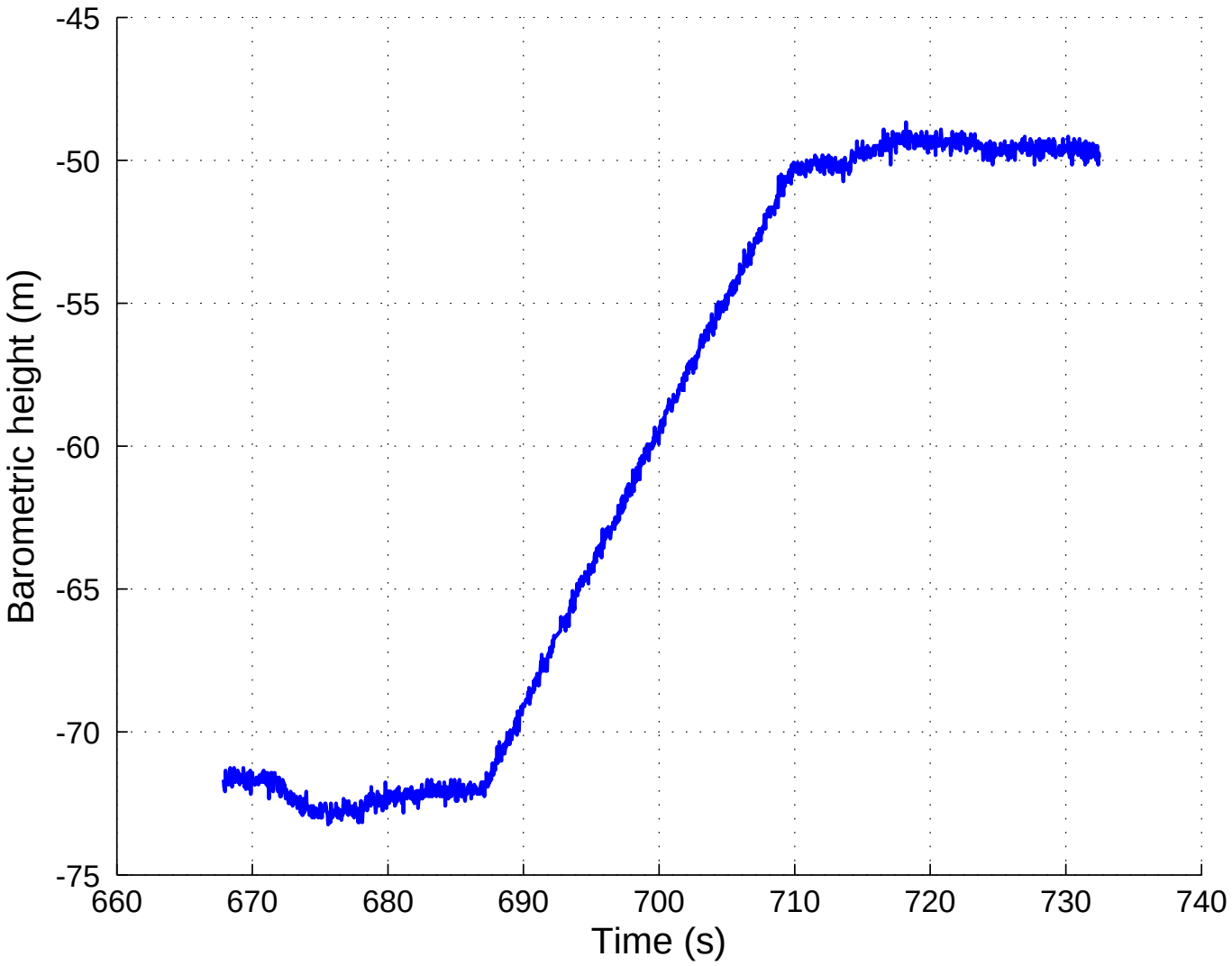
Elevator special marker


Trajectory
Starting point
End point
Height [m]
10
5
0
-5
-10
4
2
0
-4
-2
0
2

Accelerometer Z-Axis



Raw barometric height



Outlook

- Navigation of pedestrians, vehicles and goods.
- Indoor 3D mapping with DGNSS, INS and VO.
- Plenoptic cameras.
- Virtual markers.