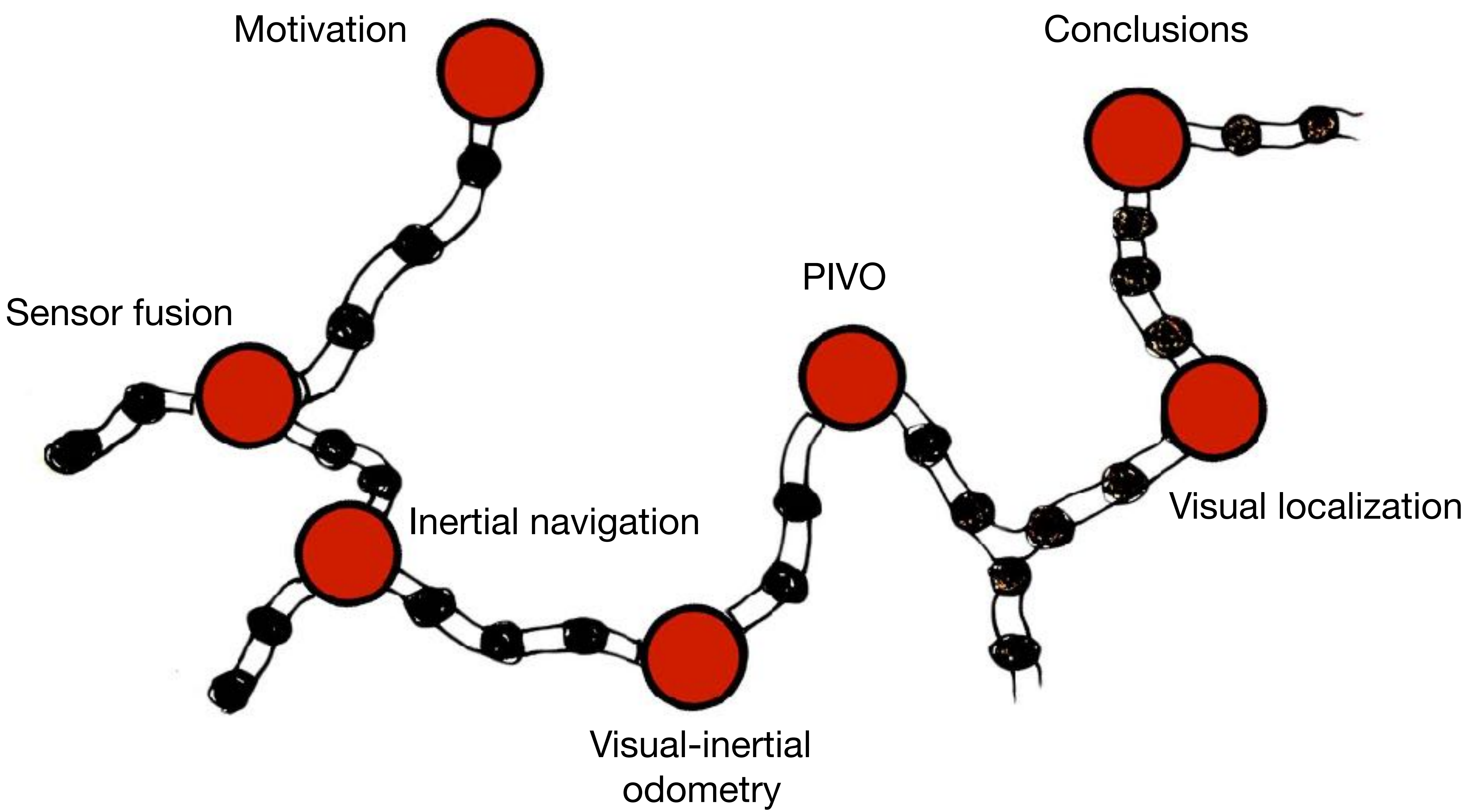
Artificial Intelligence and Vision:
Visual-Inertial Odometry and Localization for Next Generation Augmented Reality

Juho Kannala
Aalto University

Joint work with
A. Solin, S. Cortés, E. Rahtu,
X. Li, J. Ylioinas, J. Verbeek

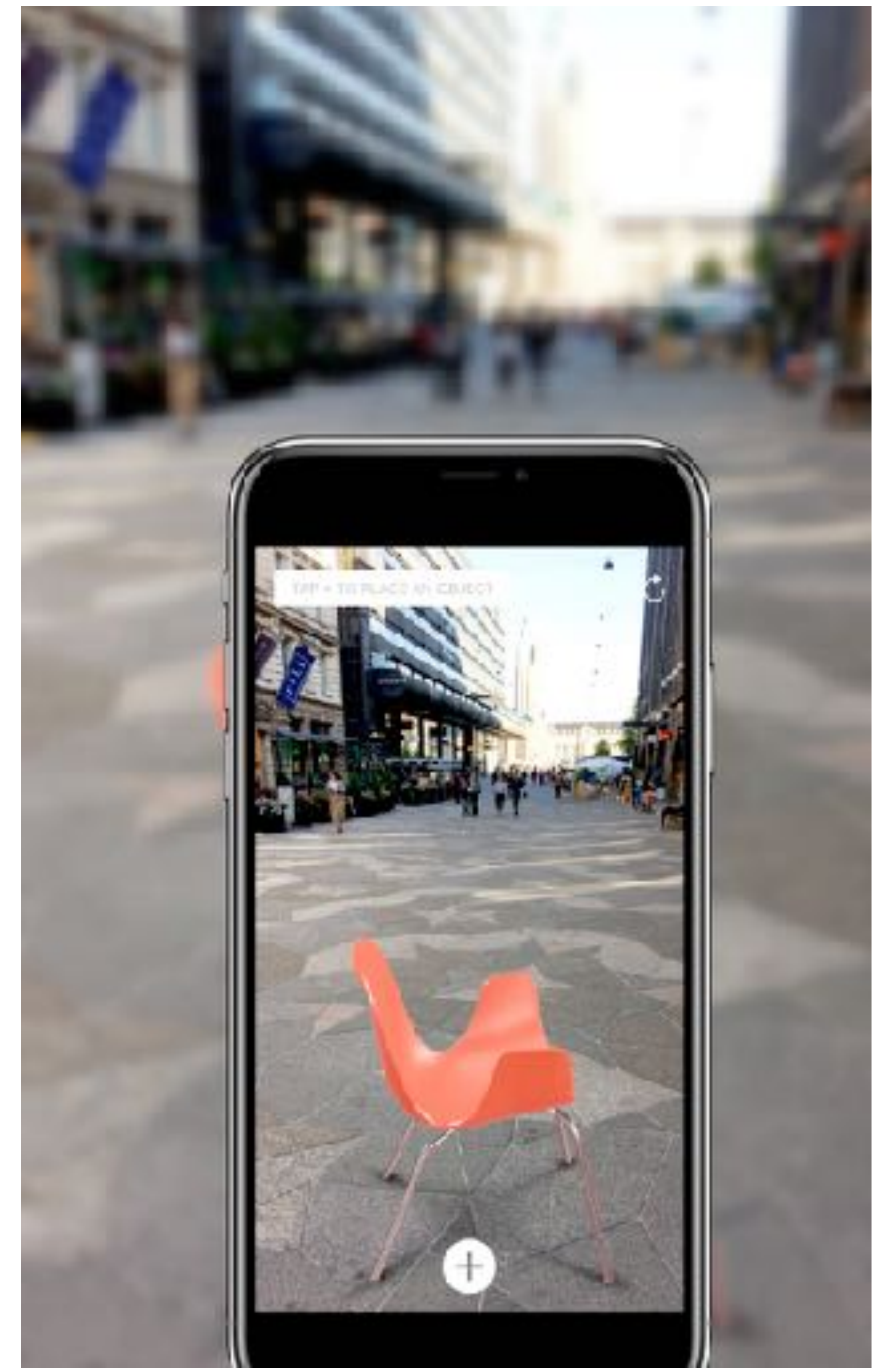
May 13th, 2019

Presentation outline



Motivation: Background

- What:
  - **Track** the motion of the device precisely in real-time
  - **Localize** the device with respect to a pre-built map/model
- Why:
  - Needed to enable augmented reality
- Why is it challenging?



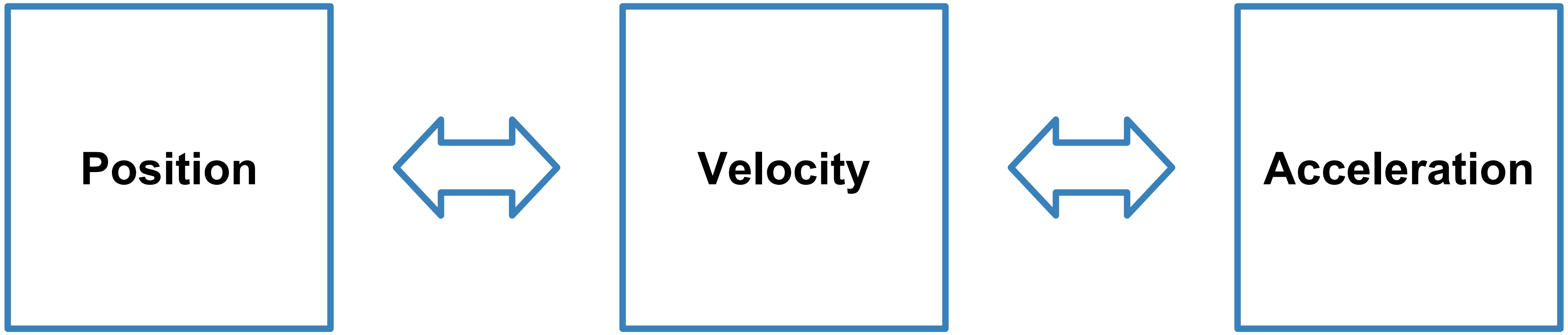
Motivation: iPhone data

Sensor fusion on smartphones

- Fusion refers to combining information from several sources
- Smartphone sensors include:
  - **Accelerometer**
  - **Gyroscope**
  - **Camera**
  - Magnetometer (compass)
  - GNSS (such as GPS)
  - Wi-Fi/BLE
  - Microphone

Inertial Navigation: How it could work

- Velocity is the integral of acceleration
- Position is the integral of velocity
- We can observe acceleration and angular velocity in the mobile phone



Inertial navigation: Why it does not work

- All inertial navigation systems suffer from integration drift
- Small errors in measurement of acceleration and angular velocity …
- Progressively larger errors in velocity…
- Even greater errors in position.

Inertial navigation: Why it does not work

- All inertial navigation systems suffer from integration drift
- Small errors in measurement of acceleration and angular velocity …
- Progressively larger errors in velocity…
- Even greater errors in position.

- The dominant component in acceleration is gravity.
- Even slight error in orientation makes the gravity ‘leak’.

- The sequential nature of the problem makes the errors accumulate.

Additional problems on smartphones

- IMUs are cheap and small
- Noisy and low-quality signals
  (biases, transients effects, alignment issues, etc.)
- Additive and multiplicative biases
  (not observing the absolute accelerations or rotations)
- Low sampling frequency
  (100 Hz vs. 2000 Hz)
- Missing data / variable sampling rate

But these are all only hardware limitations…

Inertial Navigation: How to make it work

- Input data is the accelerometer data $\mathbf{a}_k$ and gyroscope data $\boldsymbol{\omega}_k$.

[1] Solin A, Santiágo C, Rahtu E, Kannala J (FUSION 2018).
Inertial odometry in handheld smartphones.

Inertial Navigation: How to make it work

- Input data is the accelerometer data $\mathbf{a}_k$ and gyroscope data $\boldsymbol{\omega}_k$.
- Dynamical model:

$$\begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \\ \mathbf{q}_k \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{k-1} + \mathbf{v}_{k-1} \Delta t_k \\ \mathbf{v}_{k-1} + [\mathbf{q}_k (\tilde{\mathbf{a}}_k + \varepsilon_k^{\mathrm{a}}) \mathbf{q}_k^{\star} - \mathbf{g}] \Delta t_k \\ \boldsymbol{\Omega}[(\tilde{\boldsymbol{\omega}}_k + \varepsilon_k^{\omega}) \Delta t_k] \mathbf{q}_{k-1} \end{pmatrix}$$

for the position $\mathbf{p}_k$, velocity $\mathbf{v}_k$, and orientations $\mathbf{q}_k$ over time steps t_k.

[1] Solin A, Santiágo C, Rahtu E, Kannala J (FUSION 2018).
Inertial odometry in handheld smartphones.

Inertial Navigation: How to make it work

- Input data is the accelerometer data $\mathbf{a}_k$ and gyroscope data ω_k.
- Dynamical model:

$$\begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \\ \mathbf{q}_k \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{k-1} + \mathbf{v}_{k-1}\Delta t_k \\ \mathbf{v}_{k-1} + [\mathbf{q}_k(\tilde{\mathbf{a}}_k + \varepsilon_k^{\mathrm{a}})\mathbf{q}_k^\star - \mathbf{g}]\Delta t_k \\ \Omega[(\tilde{\omega}_k + \varepsilon_k^{\omega})\Delta t_k]\mathbf{q}_{k-1} \end{pmatrix}$$

  for the position $\mathbf{p}_k$, velocity $\mathbf{v}_k$, and orientations $\mathbf{q}_k$ over time steps t_k.
- A (non-linear) Kalman filter combines the model with the data in a probabilistic way.

[1] Solin A, Santiágo C, Rahtu E, Kannala J (FUSION 2018). Inertial odometry in handheld smartphones.

Inertial Navigation: How to make it work

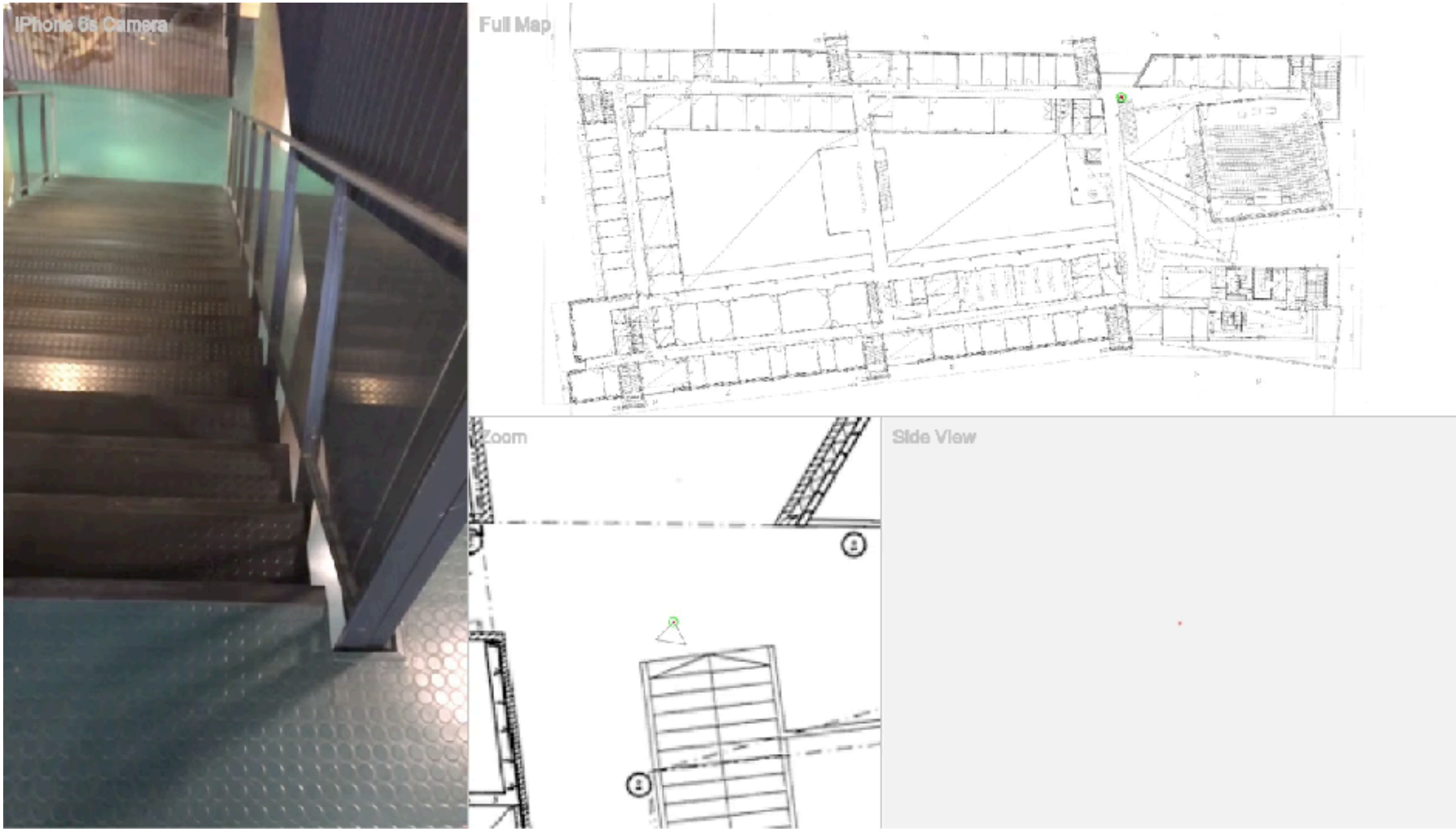
- Additional constraints are required.
- This framework can use:
  - Zero-velocity updates
  - Position fixes
  - Loop-closures
  - Barometric air pressure for relative height
  - Indirect orientation info
  - …
- A pseudo-constraint keeping the velocity component from exploding

Example studies

- Equipment used:
  - Off-the-shelf iPhone
- Sensors:
  - Gyroscope, accelerometer
  - Sampling rate: 100 Hz
- Computations:
  - Off-line

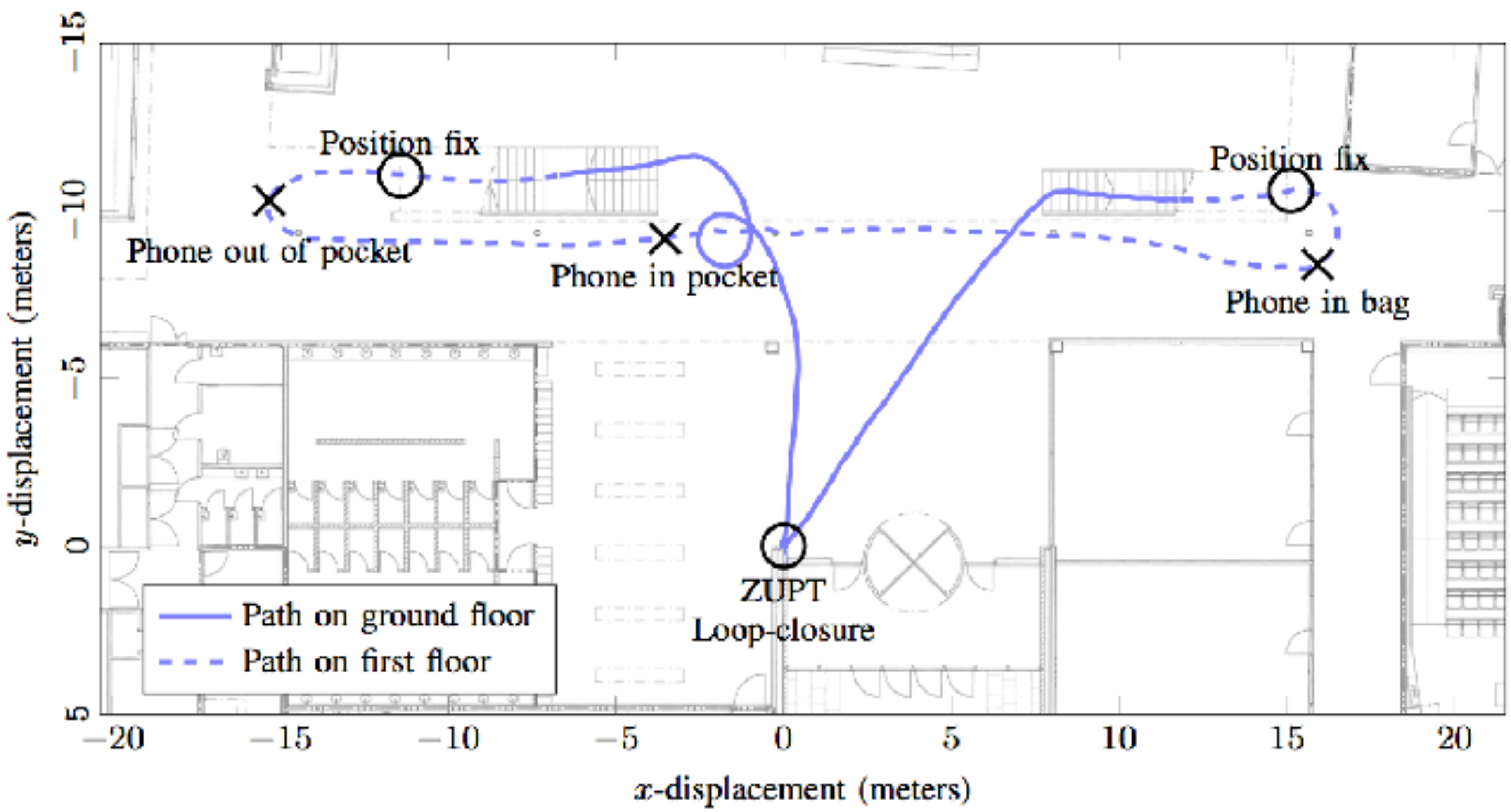
    …but can (of course) be done on the device

Example: With position fixes

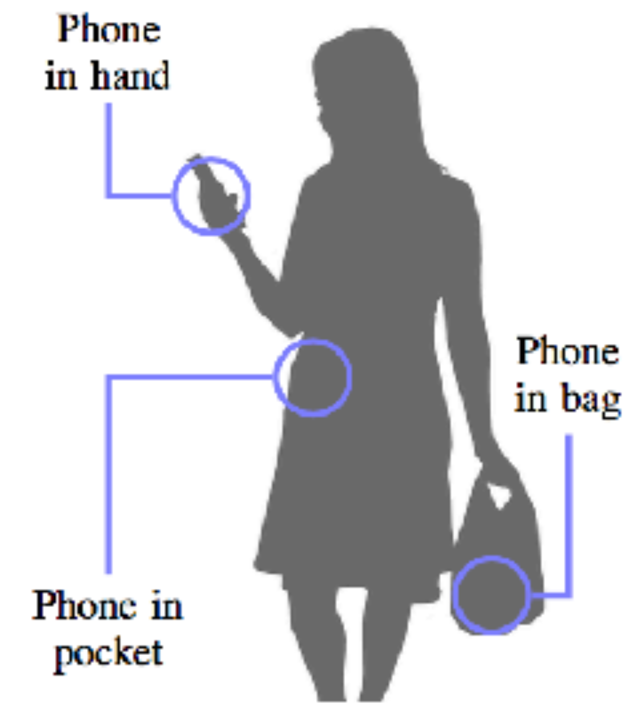


Note: The camera is *not* used at all.

Example

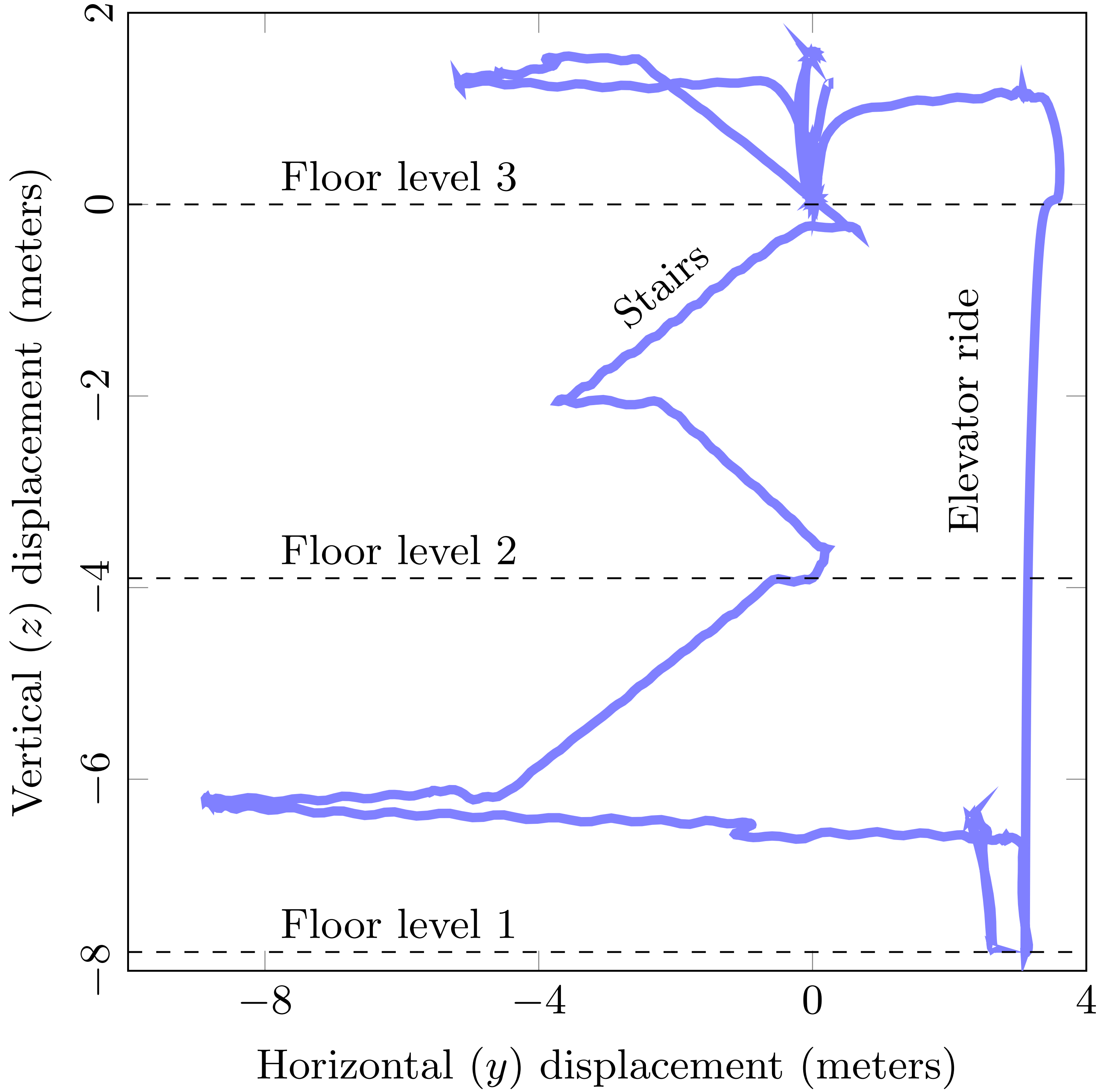


Position fix
Position fix
Phone out of pocket
Phone in pocket
Phone in bag
ZUPT
Loop-closure
Path on ground floor
Path on first floor
y-displacement (meters)
x-displacement (meters)
−15
−10
−5
0
5
−20
−15
−10
−5
0
5
10
15
20



Phone in hand
Phone in pocket
Phone in bag

Example



Floor level 3
Floor level 2
Floor level 1
Stairs
Elevator ride
Vertical (z) displacement (meters)
Horizontal (y) displacement (meters)
2
0
−2
−4
−6
−8
−8
−4
0
4

Visual-inertial odometry

- Combining visual and inertial data for odometry
- Constraints from visual features seen in consecutive frames
- Strengths over visual-only:
  - Infer the true scale
  - Survive from occlusions

Problems on smartphones

- Small field-of-view (monocular camera)

Google Tango FOV

Problems on smartphones

- Small field-of-view
  (monocular camera)

iPhone FOV

Problems on smartphones

- Small field-of-view
  (monocular camera)
- Rolling-shutter camera
  (not optimised for VIO)
- Limited processing power
  (maybe not that limited…)
- Handheld movement
  (different from a drone/robot)
- Full occlusions
  (the camera might be covered)
- No control of environment
  (moving objects, feature-poor)

PIVO

PROBABILISTIC INERTIAL-VISUAL ODOMETRY

- Previous methods tend to be developed visual-first
  (and in this case visual information is bad)
- Treats visual information as a signal of opportunity
- Information hidden within the noise

- The camera provides bursts of high-quality odometry
  (recognise those bursts!)
- A calibrated IMU can provide good long-range results
  (learn the calibration online!)

[2] Solin A, Santiágo C, Rahtu E, Kannala J (WACV 2018).
PIVO: Probabilistic Inertial-Visual Odometry for Occlusion-Robust Navigation

PIVO

- State space model (solvable by EKF)
- Dynamics driven by the IMU
  (alike the inertial odometry)
- Pose augmentation on every new frame
- Visual update performed per feature track
- Suspicious visual updates rejected
  (if not agreeing with the uncertainties)

City-wide example

City-wide example

Recap - inertial-visual odometry

- Principled approach for fusing inertial and visual information
- Robustness to occlusion and dynamic objects in the scene
- Comparable with state-of-the-art in ideal scenes
- Improved performance in challenging conditions

Visual localization

- Tracking provides relative motion of the device

Visual localization

- Tracking provides relative motion of the device
- Track must be aligned with a map to obtain global coordinates

Visual localization

- Tracking provides relative motion of the device
- Track must be aligned with a map to obtain global coordinates
- This can be solved using deep learning

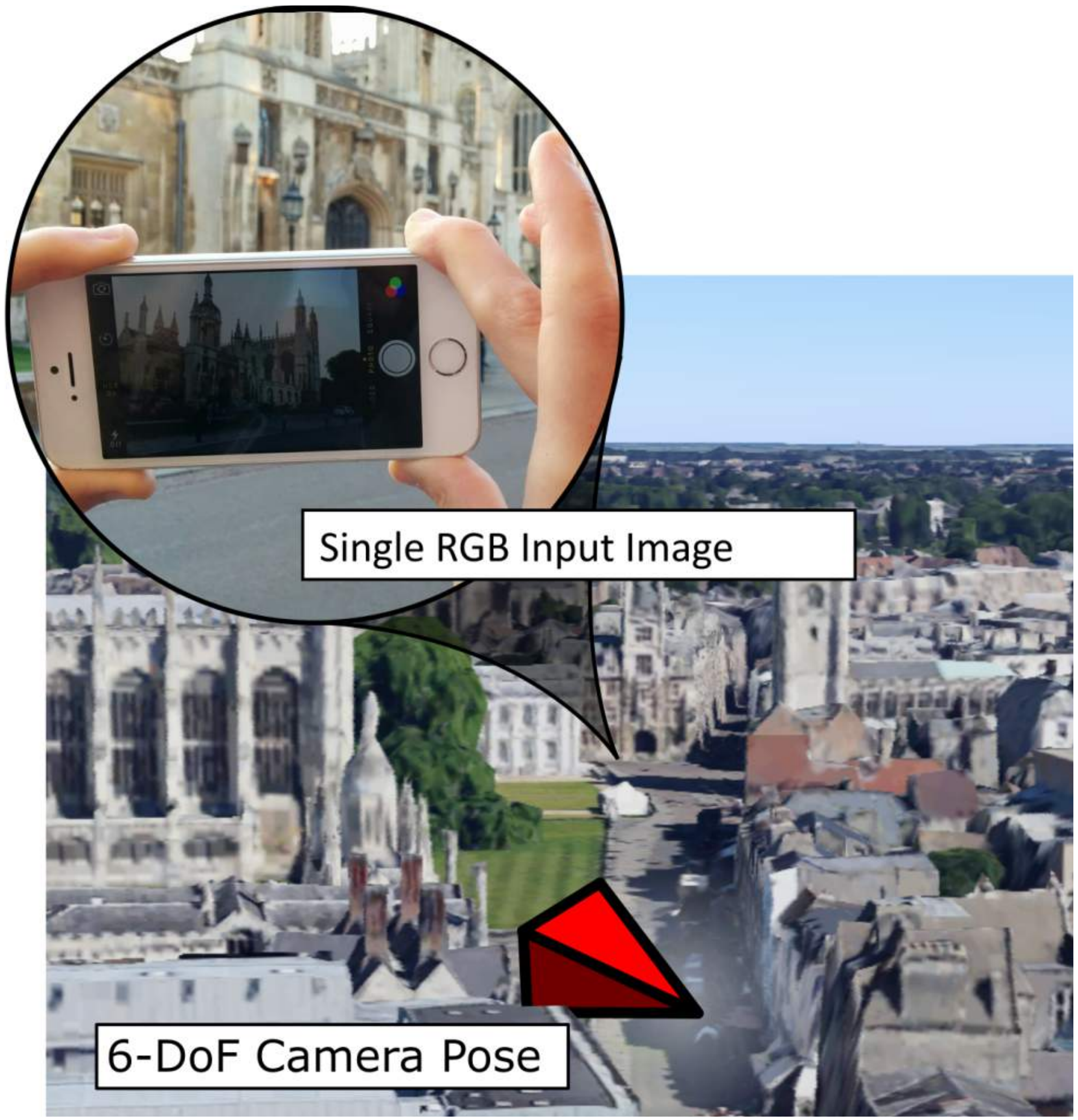


Image courtesy of Kendall et al.

Kendall, Grimes & Cipolla: PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. ICCV 2015

Visual localization

- Given training images, we compute the corresponding camera poses and a point cloud representing the 3D scene structure (= visual map)
  - This is called structure-from-motion (cf. VisualSfM, COLMAP)

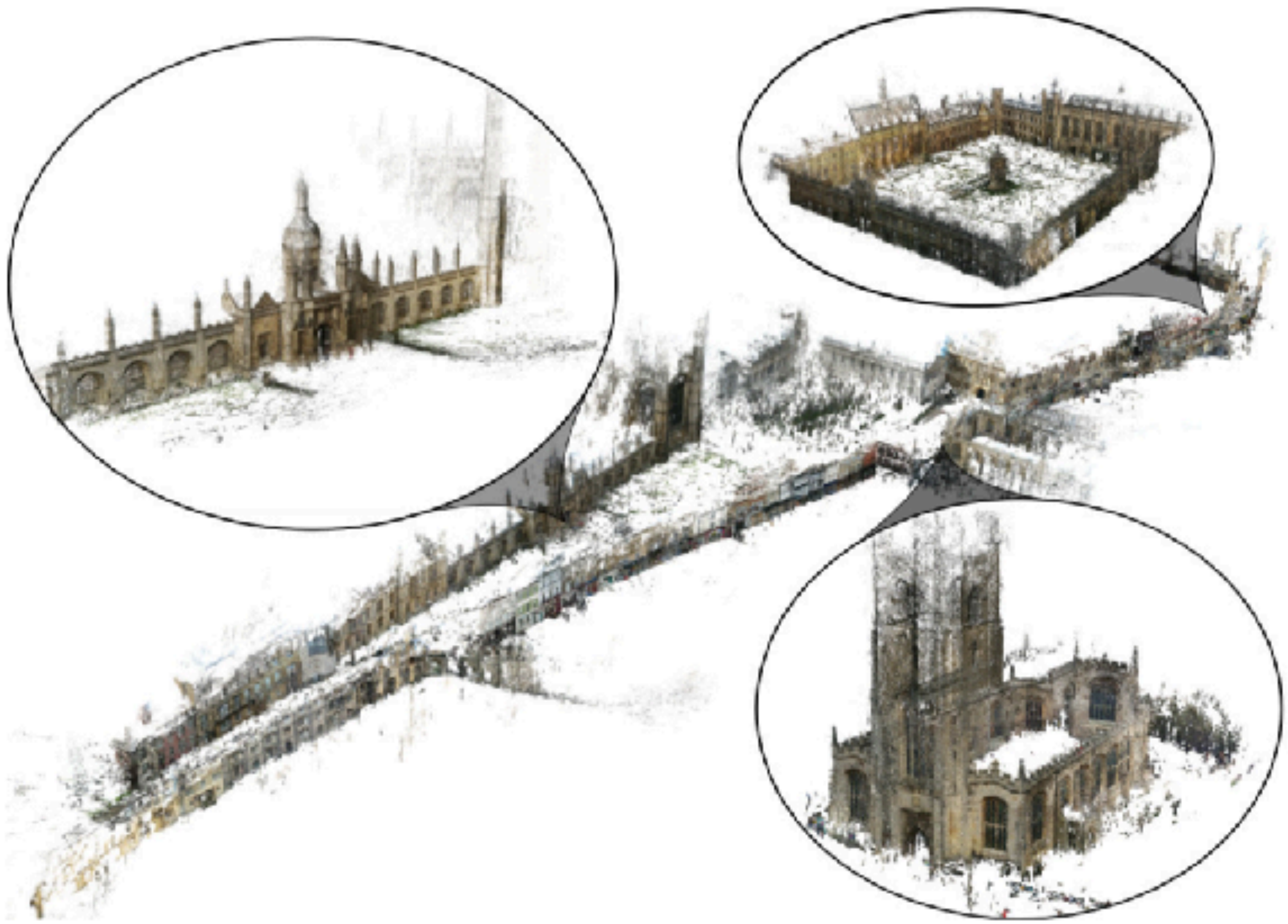


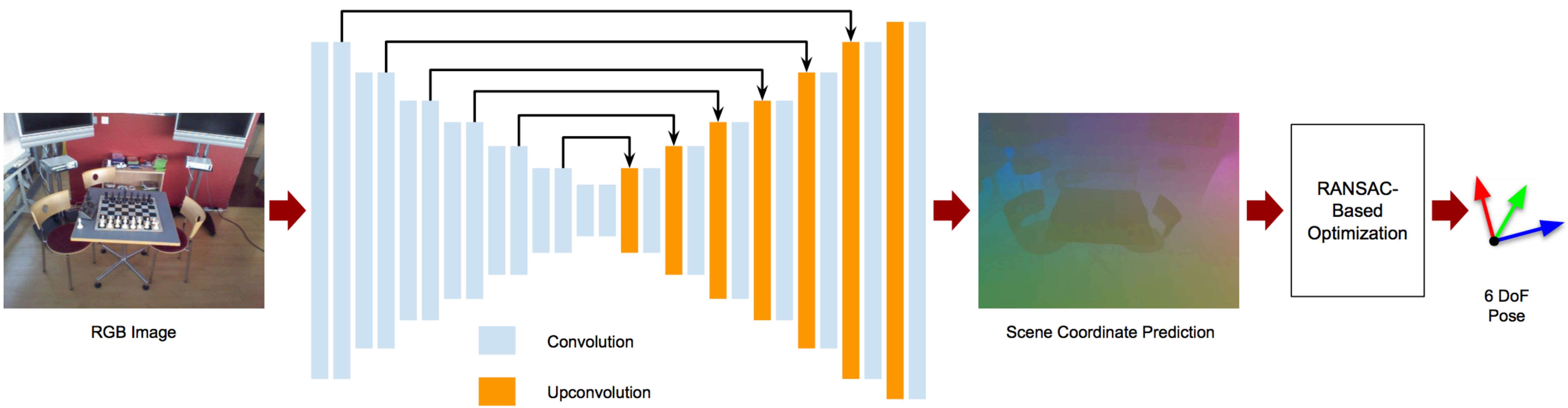
Image courtesy of Kendall et al.

Visual localization

- Given training images, we compute the corresponding camera poses and a point cloud representing the 3D scene structure (= visual map)
    - This is called structure-from-motion (cf. VisualSfM, COLMAP)
- At test time, the task is to estimate the camera pose (3D location + 3D orientation) for a query image with respect to the visual map

Scene coordinate regression

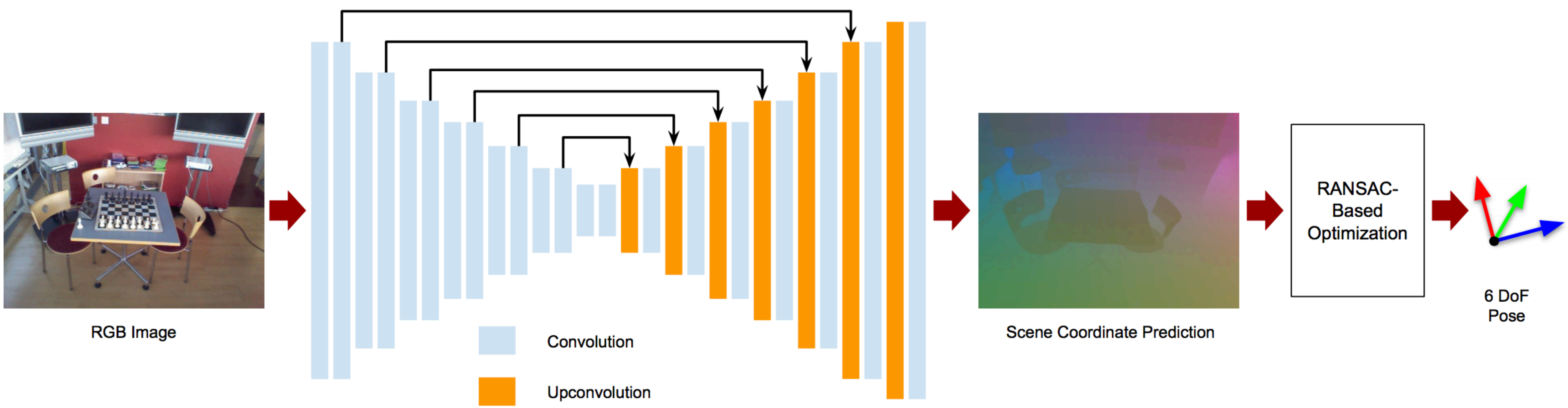
- We train a fully convolutional neural network (CNN) for regressing the scene coordinates (X,Y,Z) for all pixels



Scene coordinate regression

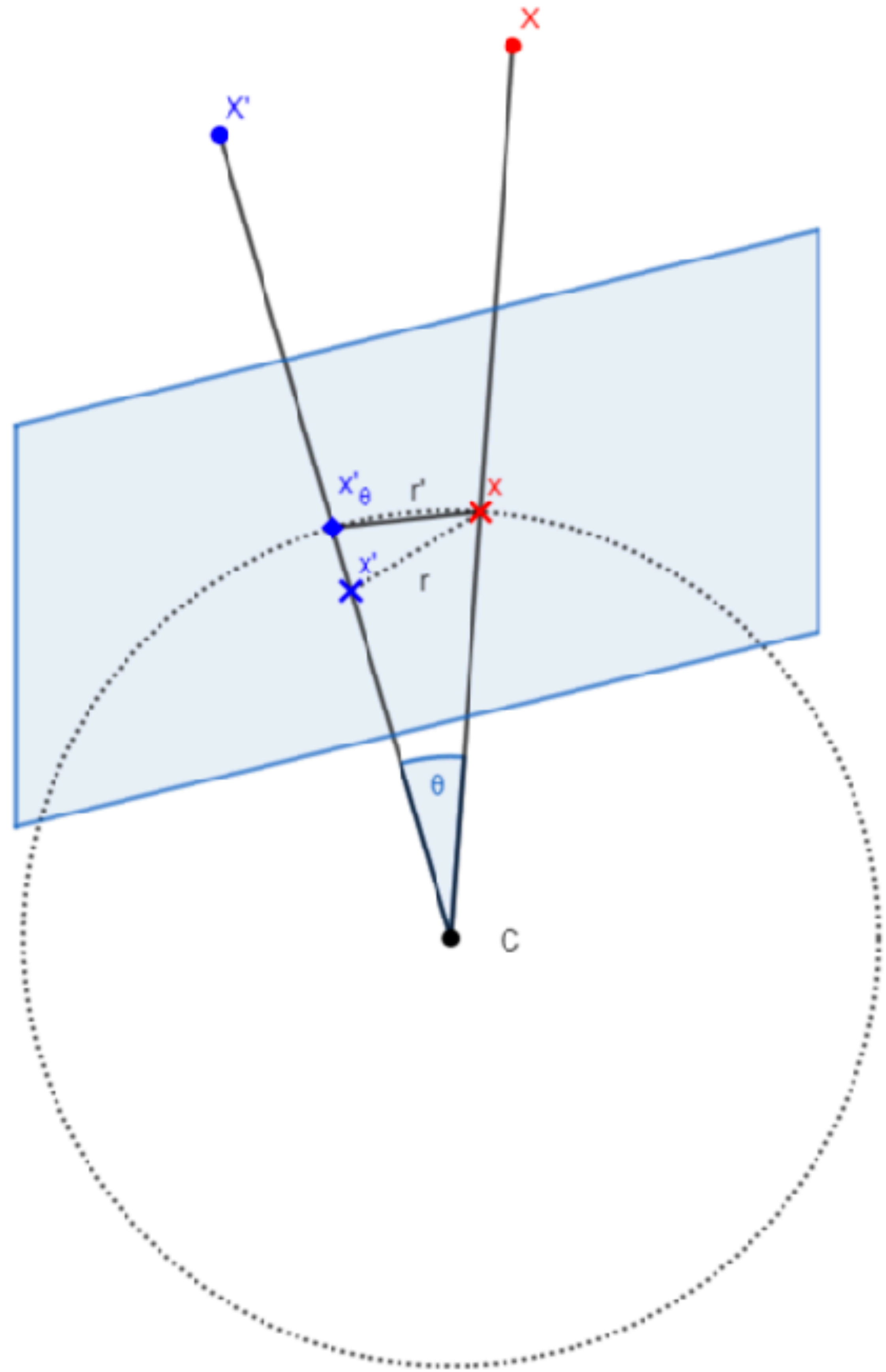
- We train a fully convolutional neural network (CNN) for regressing the scene coordinates (X,Y,Z) for all pixels

- We compute the camera pose by solving the perspective-n-point problem from the resulting 2D-to-3D matches using RANSAC (i.e. CNN maps the 2D pixel coordinates to 3D scene coordinates)

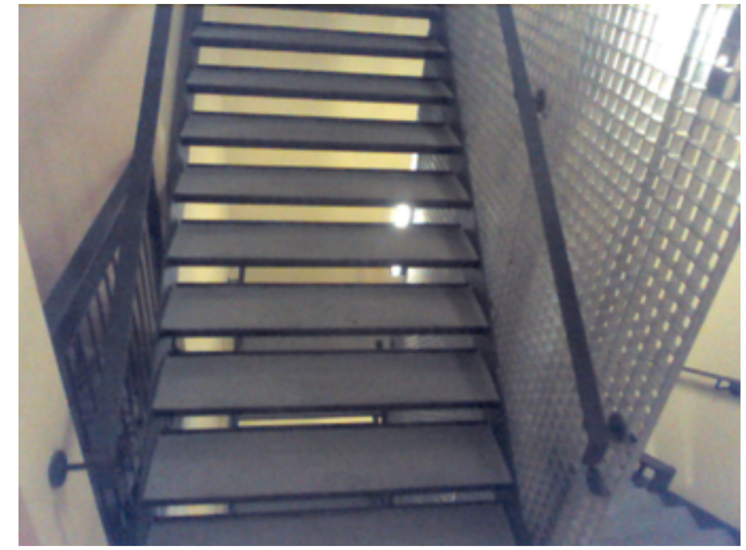


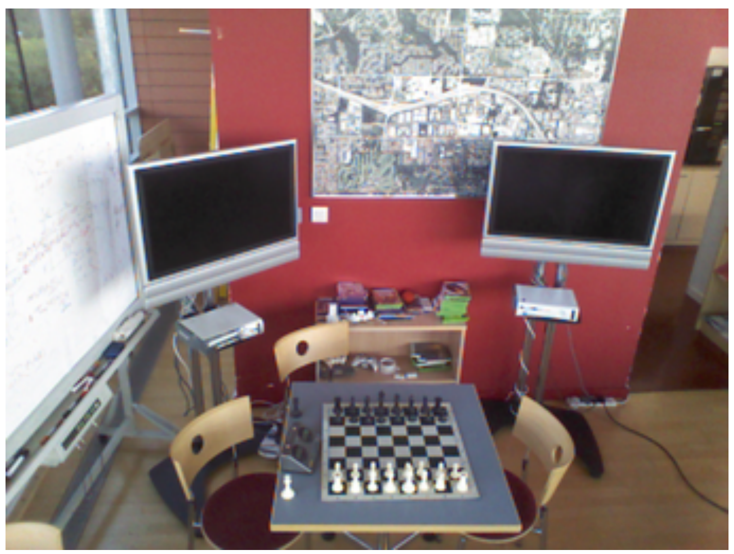
Angle-based single-view reprojection loss

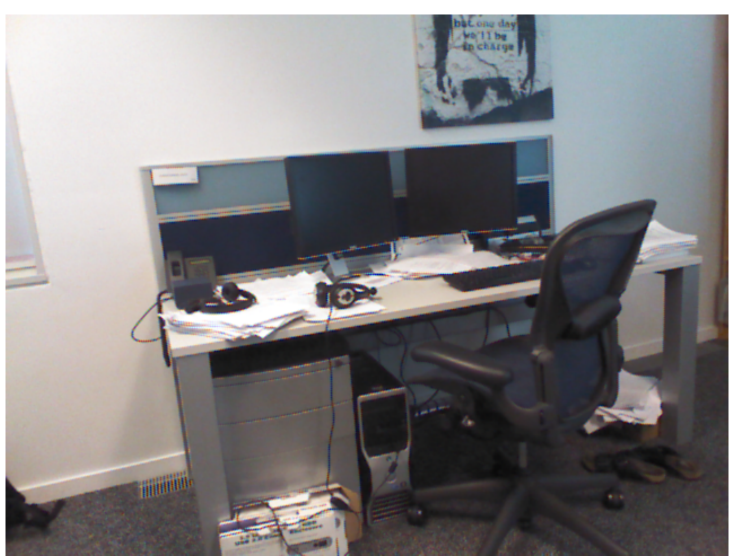
- We propose angle-based reprojection loss for optimizing the CNN
- The angle between the rays corresponding to true (X’) and predicted (X) scene coordinates is minimized for all pixels in all training images
- This formulation does not require a 3D scene model, training images with poses are sufficient!



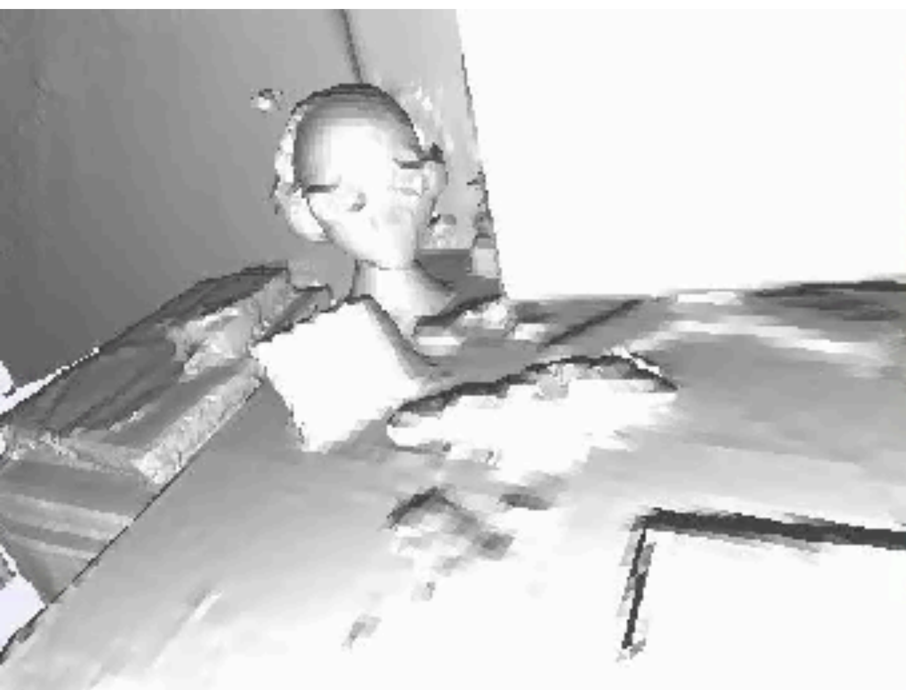
State-of-the-art results for 7-Scenes dataset



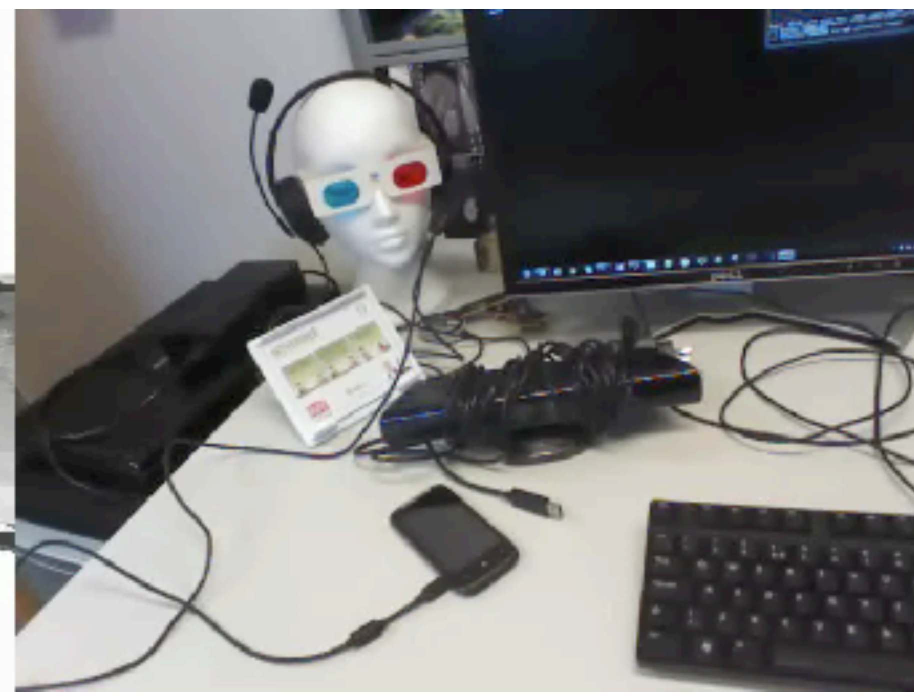




Localization of each frame in a test video (no tracking)



Brachmann & Rother (CVPR 2018)



Ours

Li et al.: Scene coordinate regression with angle-based reprojection loss for camera relocalization. ECCVW 2018

Conclusion

- Contributions for both tracking and localization:
  - Probabilistic inertial-visual odometry for occlusion-robust navigation
  - Scene coordinate regression with angle-based reprojection loss

- Ultimately, tracking and localization should be integrated

Conclusion

- Contributions for both tracking and localization:
  - Probabilistic inertial-visual odometry for occlusion-robust navigation
  - Scene coordinate regression with angle-based reprojection loss
- Ultimately, tracking and localization should be integrated
- Potential for impact in various areas:
  - More robust and precise navigation for autonomous machines (drones, robots, vehicles)
  - Improved inside-out tracking for virtual reality glasses
  - 3D-aware mobile applications (e.g. for measurement purposes)
  - Immersive augmented reality applications for smartphones

Bibliography

[1] Solin A, Cortes S, Rahtu E, Kannala J (2018).
Inertial odometry in handheld smartphones.
In International Conference on Information Fusion (FUSION)

[2] Solin A, Cortes S, Rahtu E, Kannala J (2018).
PIVO: Probabilistic inertial-visual odometry for occlusion-robust navigation.
In IEEE Winter Conference on Application of Computer Vision (WACV)

[3] Cortes S, Solin A, Rahtu E, Kannala J (2018).
ADVIO: An authentic dataset for visual-inertial odometry.
In European Conference on Computer Vision (ECCV)

[4] Li X, Ylioinas J, Kannala J (2018).
Full-frame scene coordinate regression for image-based localization.
In Robotics: Science and Systems (RSS)

[5] Li X, Ylioinas J, Verbeek J, Kannala J (2018).
Scene coordinate regression with angle-based reprojection loss for camera relocalization.
In Geometry Meets Deep Learning ECCV Workshop

Thank you!

https://users.aalto.fi/~kannalj1/