Robust Gyroscope-Aided Camera Self-Calibration

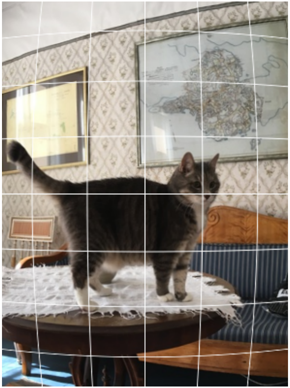
Santiago Cortés Arno Solin Juho Kannala

Aalto University

July 11, 2018

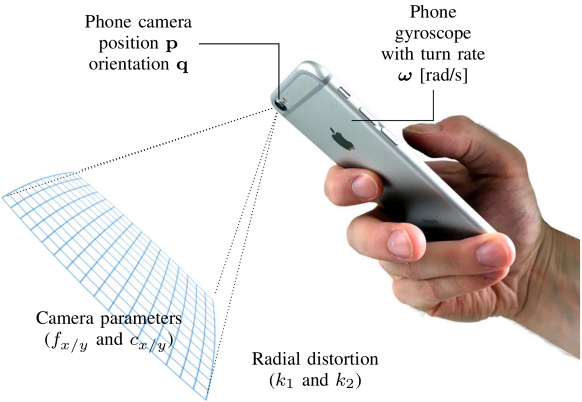
Motivation

- Camera sensors are common in smart devices
- Use cases: AR/VR, games, odometry, photography, *etc.*
- But the observed images are distorted
- The distortion can be estimated off-line or be factory-calibrated
- We want to estimate the distortion online



What the camera sees

Idea


Phone camera
position $\mathbf{p}$
orientation $\mathbf{q}$
Phone
gyroscope
with turn rate
$\boldsymbol{\omega}$ [rad/s]
Camera parameters
($f_{x/y}$ and $c_{x/y}$)
Radial distortion
(k_1 and k_2)

Camera model

- World coordinates (x, y, z) to image coordinates (u, v):
- Pinhole camera model:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K}\,\mathbf{E} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- where the intrinsic and extrinsic matrices are:

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{E} = \begin{pmatrix} \mathbf{R}^{\mathsf{T}} & -\mathbf{R}^{\mathsf{T}}\mathbf{p} \end{pmatrix}$$

- Camera pose $(\mathbf{R}, \mathbf{p})$: the camera orientation (quaternion) and position

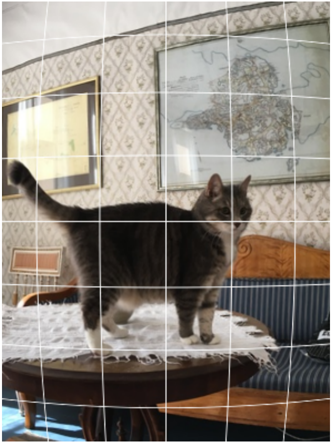
Camera model (non-linear)

- Lens distortions are typically non-linear
- Radial distortion coefficients k_1 and k_2:

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} (1 + k_1\, r^2 + k_2\, r^4)$$

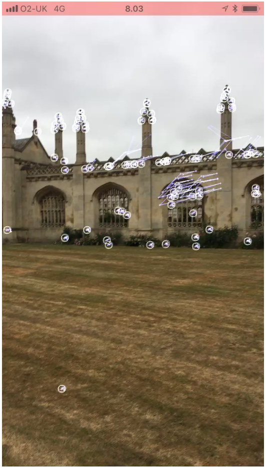
with the radial component given by

$$r = \sqrt{\left(\frac{u - c_x}{f_x}\right)^2 + \left(\frac{v - c_y}{f_y}\right)^2}$$



Feature tracking

- The dense image is not convenient to work with
- Choose sparse points by a feature detector
- Track the points over frames using a feature tracker
- Measurement data consists of tracks of points over frames



Motion model

- The gyroscope for drives the orientation dynamics:

$$\frac{\mathrm{d}\mathbf{q}(t)}{\mathrm{d}t} = \frac{1}{2}\,\boldsymbol{\Omega}(\omega)\,\mathbf{q}(t)$$

  $\mathbf{q}(t)$ is the quaternion at t and ω the angular velocity.

- The position $\mathbf{p}(t) = (p_1(t), p_2(t), p_3(t))$ is modeled as a Wiener velocity model:

$$\frac{\mathrm{d}^2 p_j(t)}{\mathrm{d}t^2} = w(t)$$

  $w(t)$ is white noise.

State estimation

- The state variables are:

$$\mathbf{x} = \begin{pmatrix} \mathbf{c}^\mathsf{T} & \mathbf{p}^\mathsf{T} & \mathbf{v}^\mathsf{T} & \mathbf{q}^\mathsf{T} & \mathbf{z}^\mathsf{T} \end{pmatrix}^\mathsf{T}$$

  $\mathbf{c} = (f_x, f_y, c_x, c_y, k_1, k_2)$ are the parameters, $\mathbf{p}$ position, $\mathbf{v}$ velocity, $\mathbf{q}$ orientation, and $\mathbf{z}$ feature world coordinates.

- State space model:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_k \, \mathbf{x}_{k-1} + \varepsilon_k, \\ \mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k) + \gamma_k, \end{aligned}$$

  where $\mathbf{A}_k$ depends on ω_k and $\mathbf{y}_k = (u_1, v_2, \ldots)$ are the feature image coordinates.

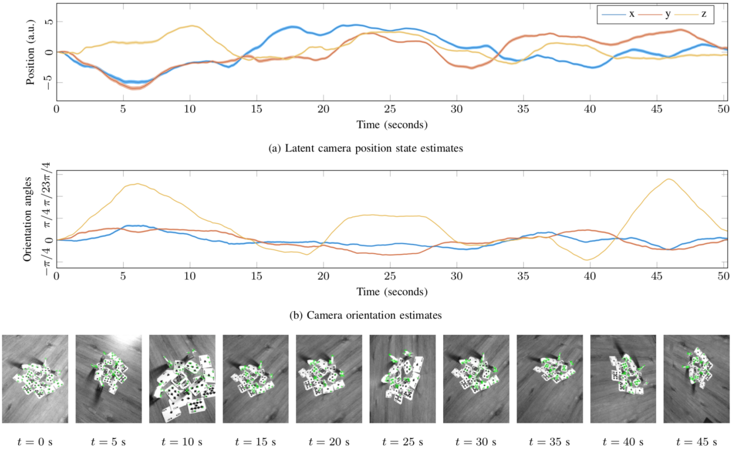
- We use an Extended Kalman filter for inference.

Experiments

The videos are available at:

https://youtu.be/ro7TeQKgfT0

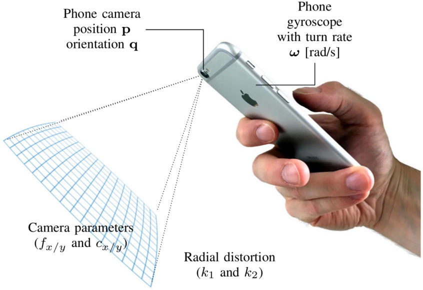
Real-world experiment



(c) Associated frames at different time points

Recap

- Online estimation of camera parameters
- Information fusion:
  - Gyroscope-driven
  - Feature-track observations
- Movement constrained by a Wiener velocity motion model
- Inference done by an EKF



- Link to codes can be found on my homepage!
- Homepage: http://arno.solin.fi
- Twitter: @arnosolin