ETH zürich

EMBEDDED PHOTOMETRIC VISUAL ODOMETRY

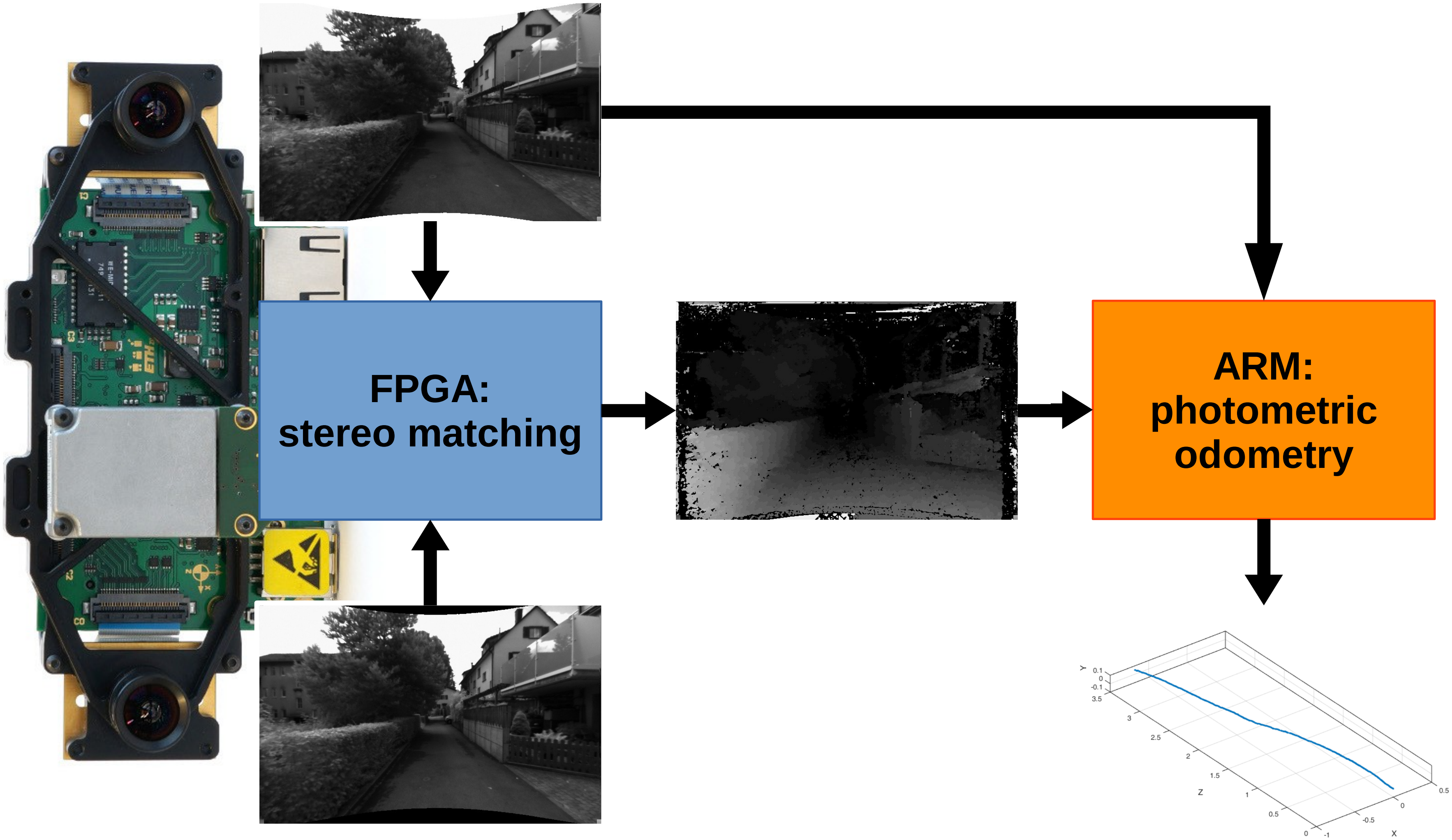
Samuel Bryner

Bachelor Thesis
Supervised by Jörn Rehder and Pascal Gohl.

Autonomous Systems Lab

MOTIVATION

- demonstrating novel integration of FPGA and ARM trough photometric odometry
- NOT the most efficient approach to embedded visual odometry
  - M. Dymczyk, „*visual-inertial motion estimation on computationally constrained platforms*“, technical report, 2014

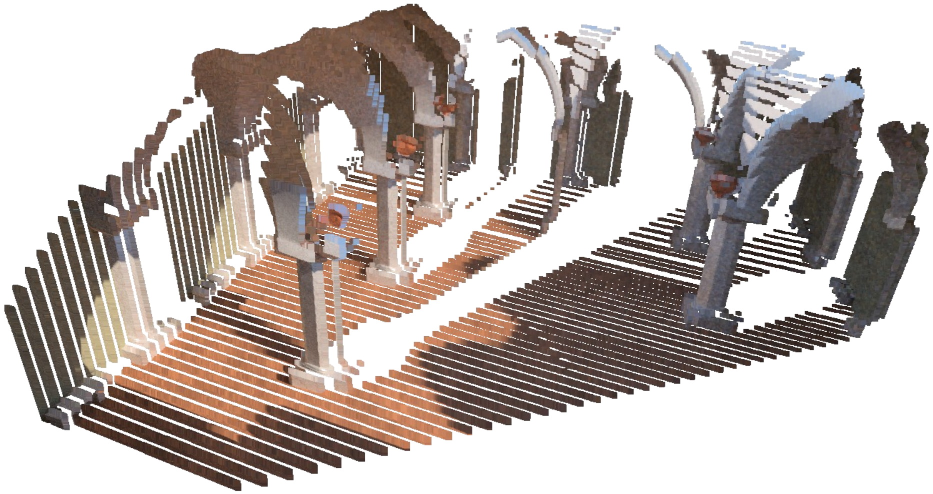


D. Honegger, H. Oleynikova, M. Pollefeys, „*Real-Time and Low Latency Embedded Computer Vision Hardware Based on a Combination of FPGA and Mobile CPU*“, IROS 2014 (IEEE/RSJ International Conference on Intelligent Robots and Systems).

METHOD: project into space

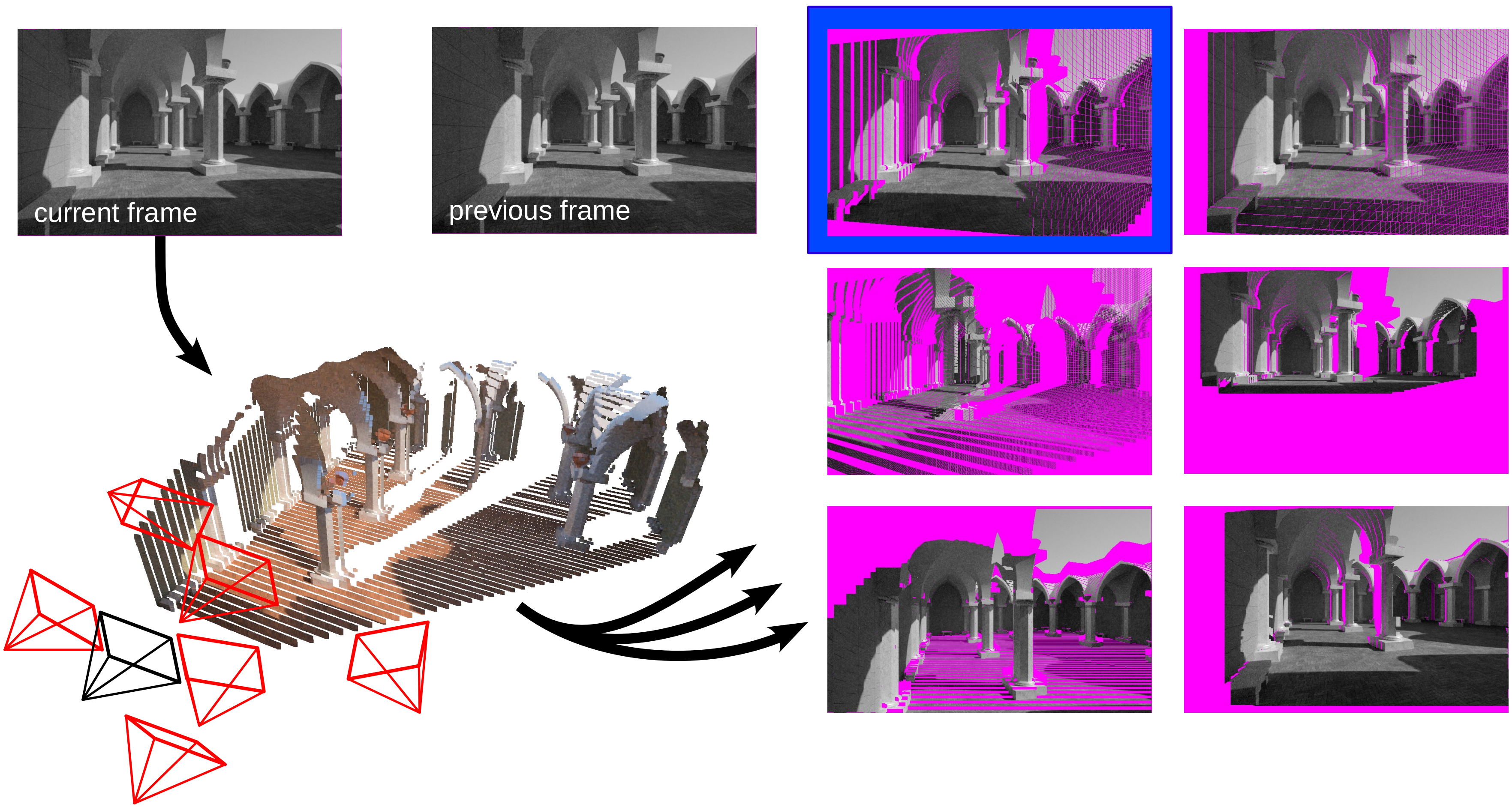
+

=



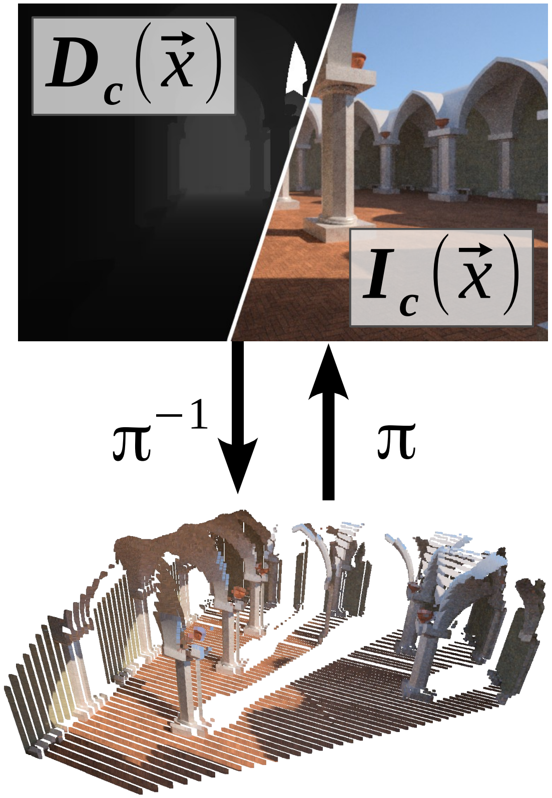
point cloud

METHOD: render from new viewpoints



METHOD: measure photometric error

METHOD: formularizing the problem



point in image: $\vec{x} := (u, v) \in \mathbb{R}^2$

intensity: $I(\vec{x}) : \mathbb{R}^2 \to \mathbb{R}$

disparity: $D(\vec{x}) : \mathbb{R}^2 \to \mathbb{R}$

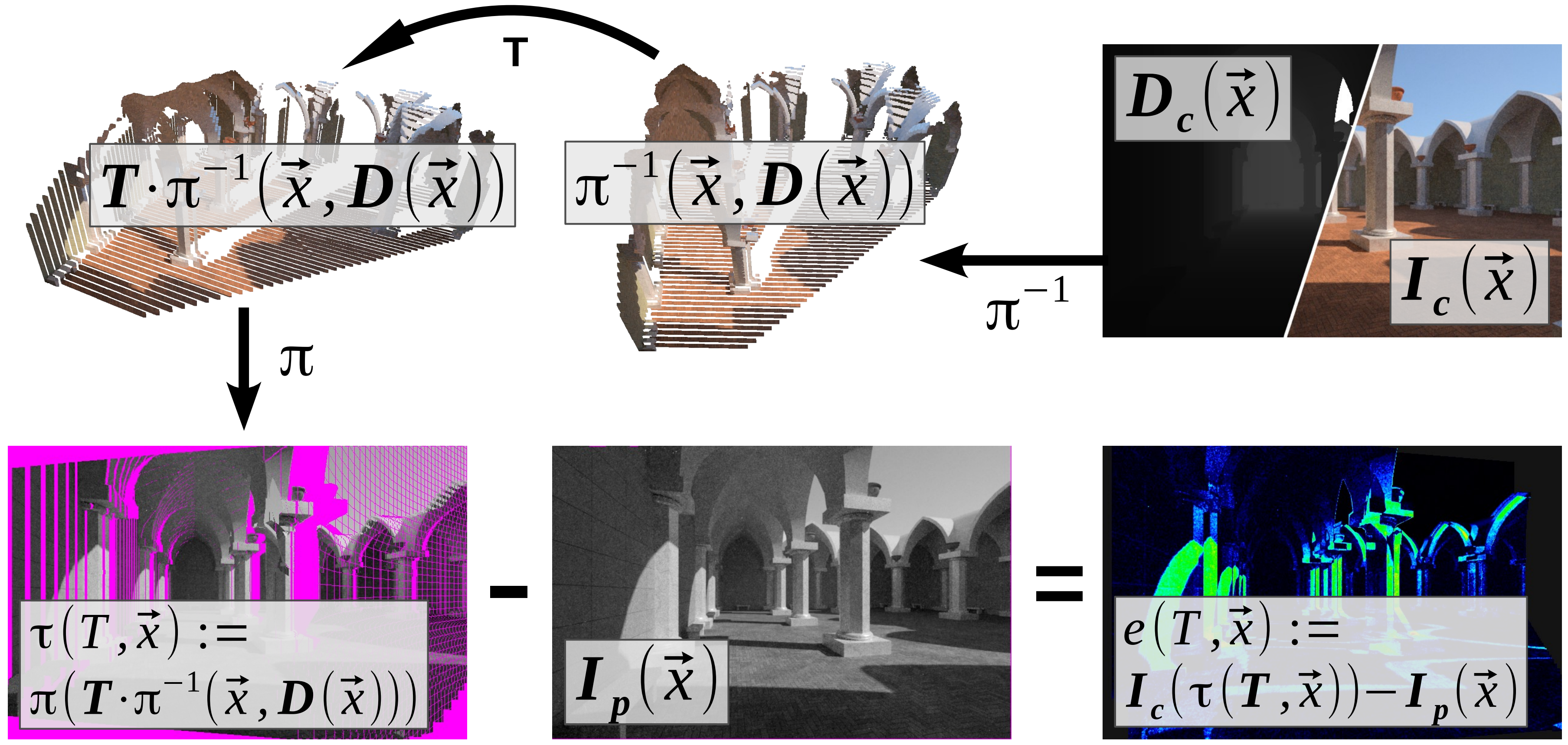
back-project:

$$\pi^{-1}(\vec{x}, D(\vec{x})) := \frac{b}{D(\vec{x})} \begin{bmatrix} u - c_u \\ v - c_v \\ f \end{bmatrix}$$

project:

$$\pi(x, y, z) := \frac{f}{z} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c_u \\ c_v \end{bmatrix}$$

METHOD: warping pipeline



A. Comport, E. Malis, and P. Rives, "*Accurate quadrifocal tracking for robust 3d visual odometry,*" in IEEE Conference on Robotics and Automation, April 2007, pp. 40–45.

therefore:

minimize

$$e(\boldsymbol{T},\vec{x}) := \boldsymbol{I}_{\boldsymbol{k}}(\vec{x}) - \boldsymbol{I}_{\boldsymbol{c}}(\tau(\boldsymbol{T},\vec{x}))$$

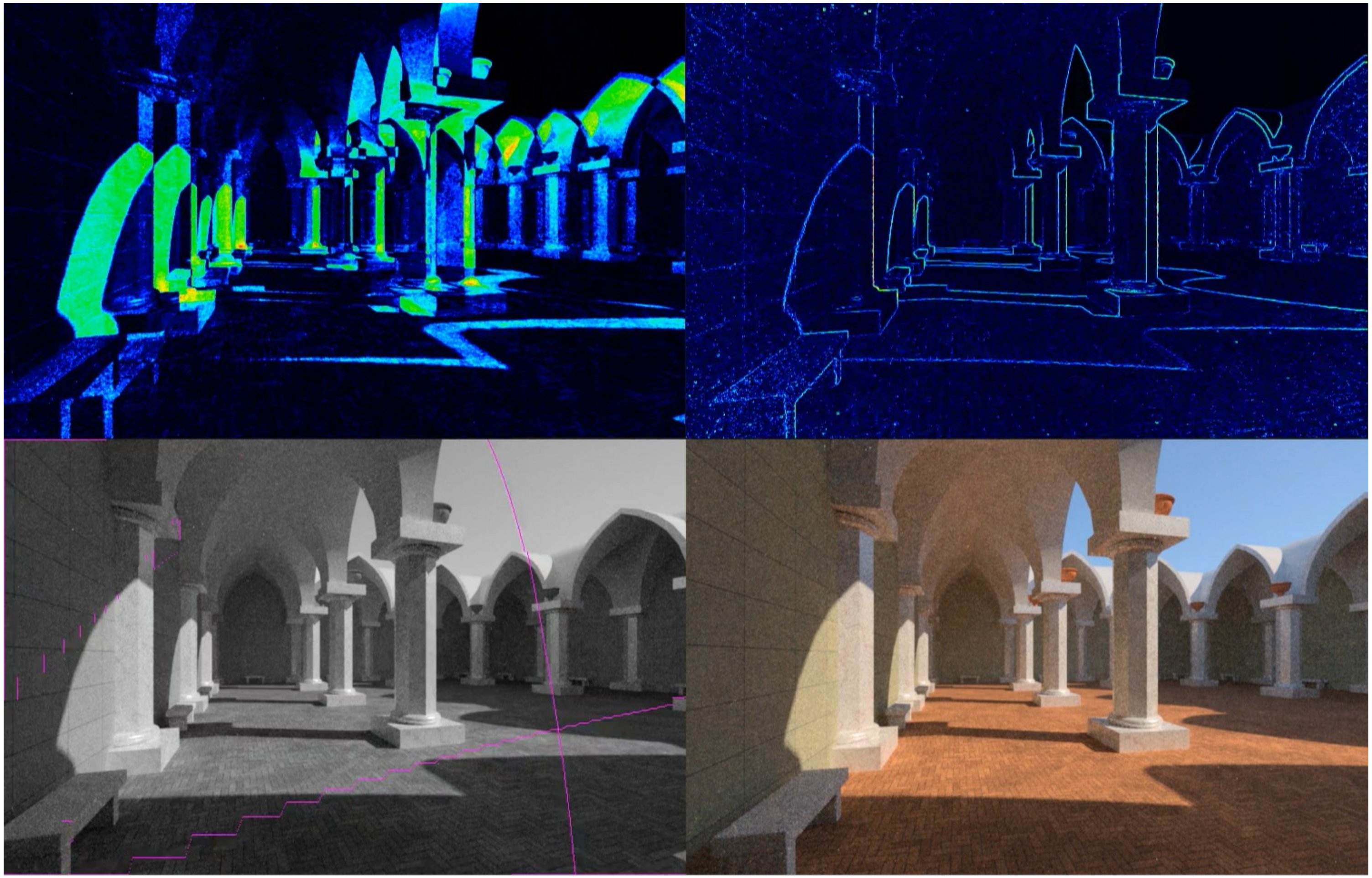
for every pixel:

$$\hat{\boldsymbol{T}} = \underset{\boldsymbol{T}}{argmin} \sum_{\vec{x}\in \boldsymbol{I}_{\boldsymbol{k}}} e(\boldsymbol{T},\vec{x})^2$$

using Gauss Newton:

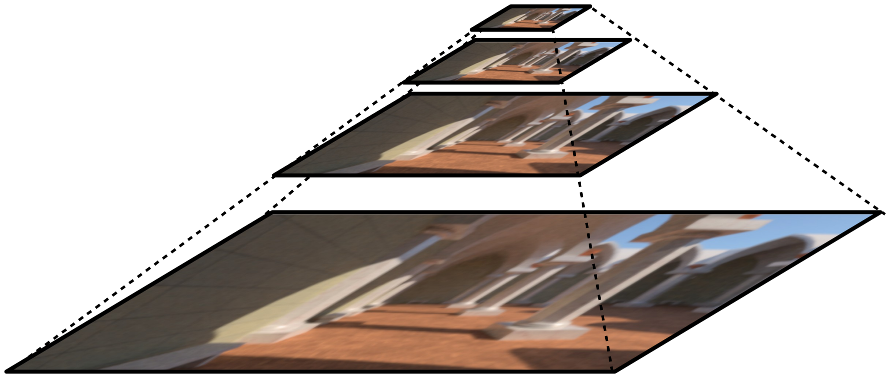
$$\boldsymbol{J}^{\boldsymbol{T}} \boldsymbol{J} \Delta \boldsymbol{T} = -\boldsymbol{J}^{\boldsymbol{T}} e(\boldsymbol{T})$$

GAUSS-NEWTON IN ACTION

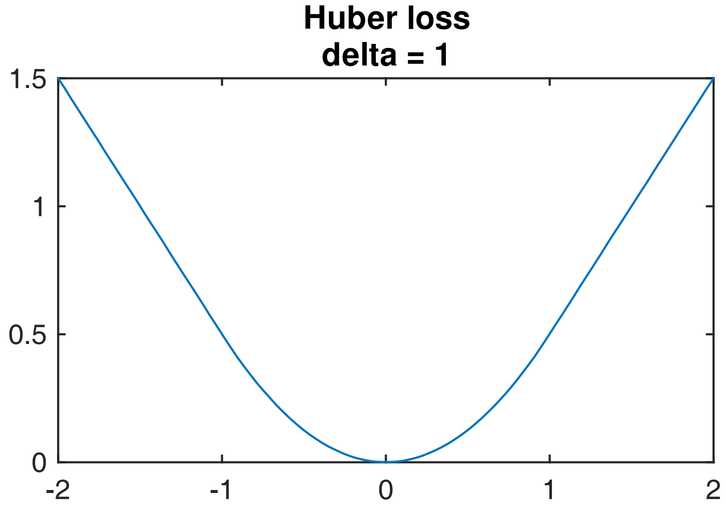


METHOD: optimizations

- use image pyramid
  - good speed / precision tradeoff
  - better region of conversion

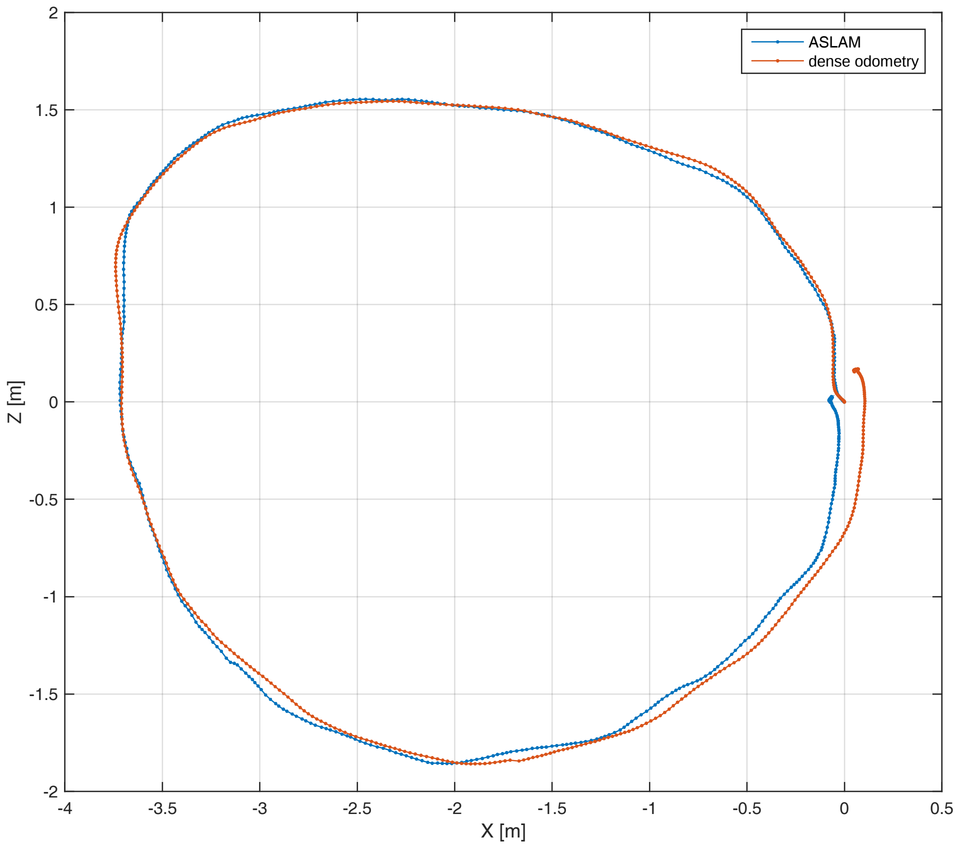


- only use pixels with strong gradient
- Huber weights



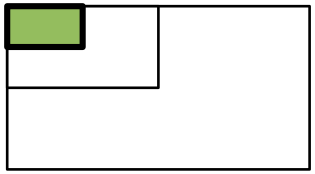
RESULTS: photometric odometry VS. ASLAM

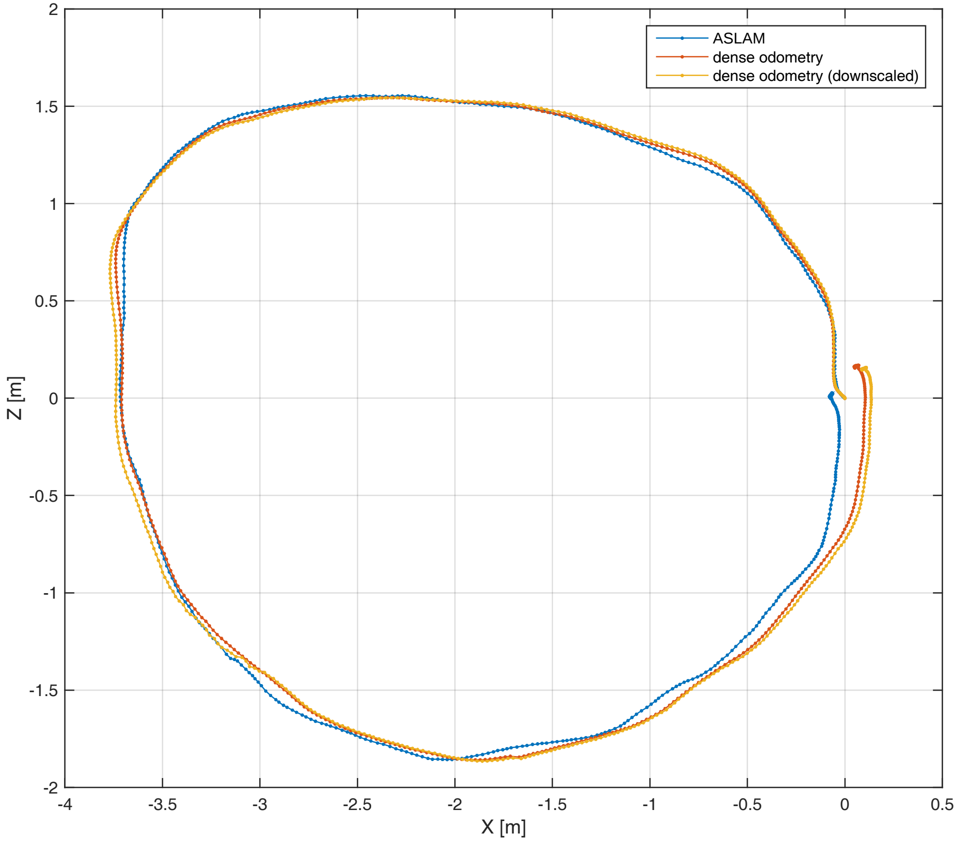
offline with OpenCV SGM for stereo



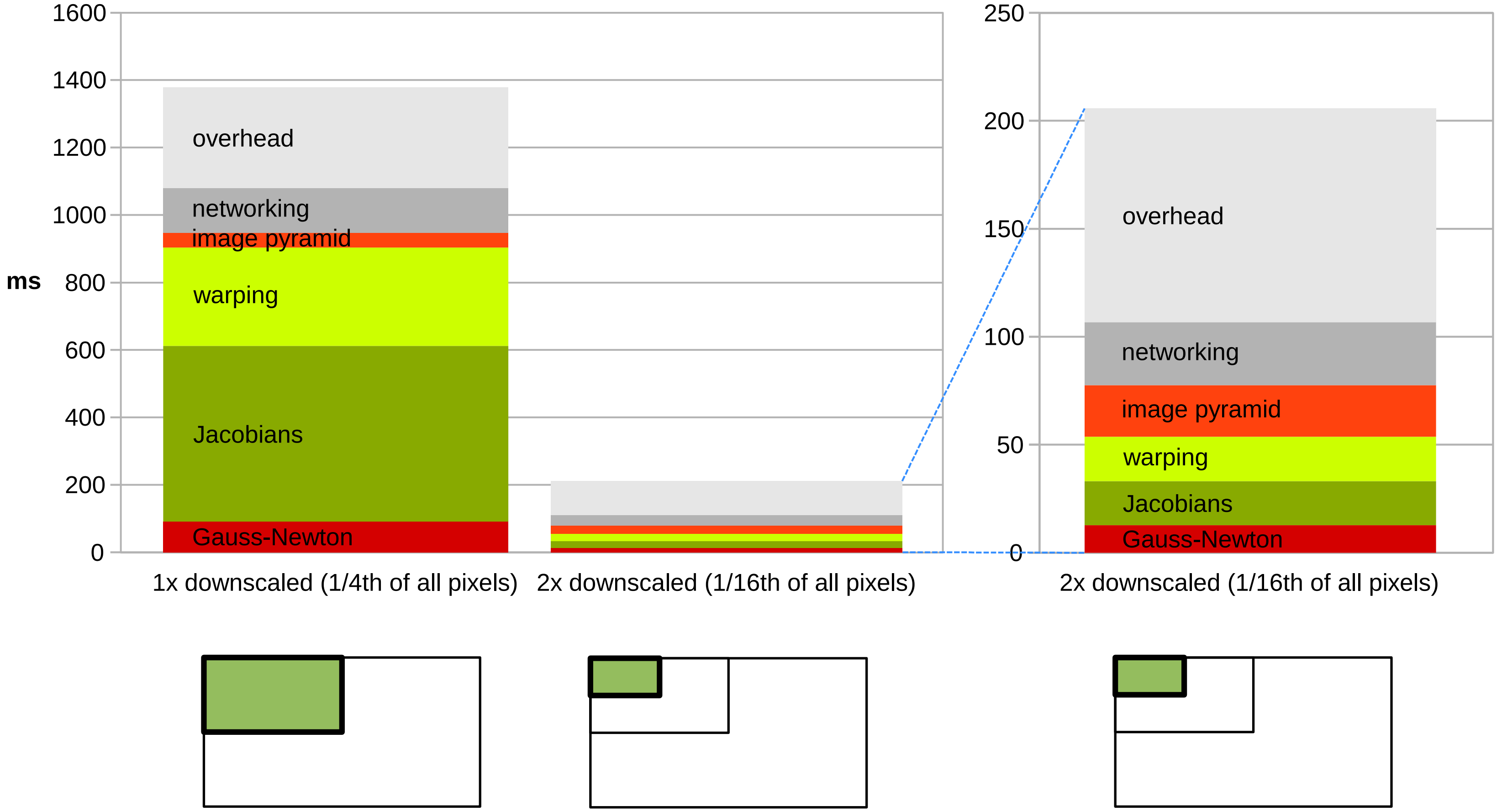
RESULTS: 1/16th of pixels

offline with OpenCV SGM for stereo



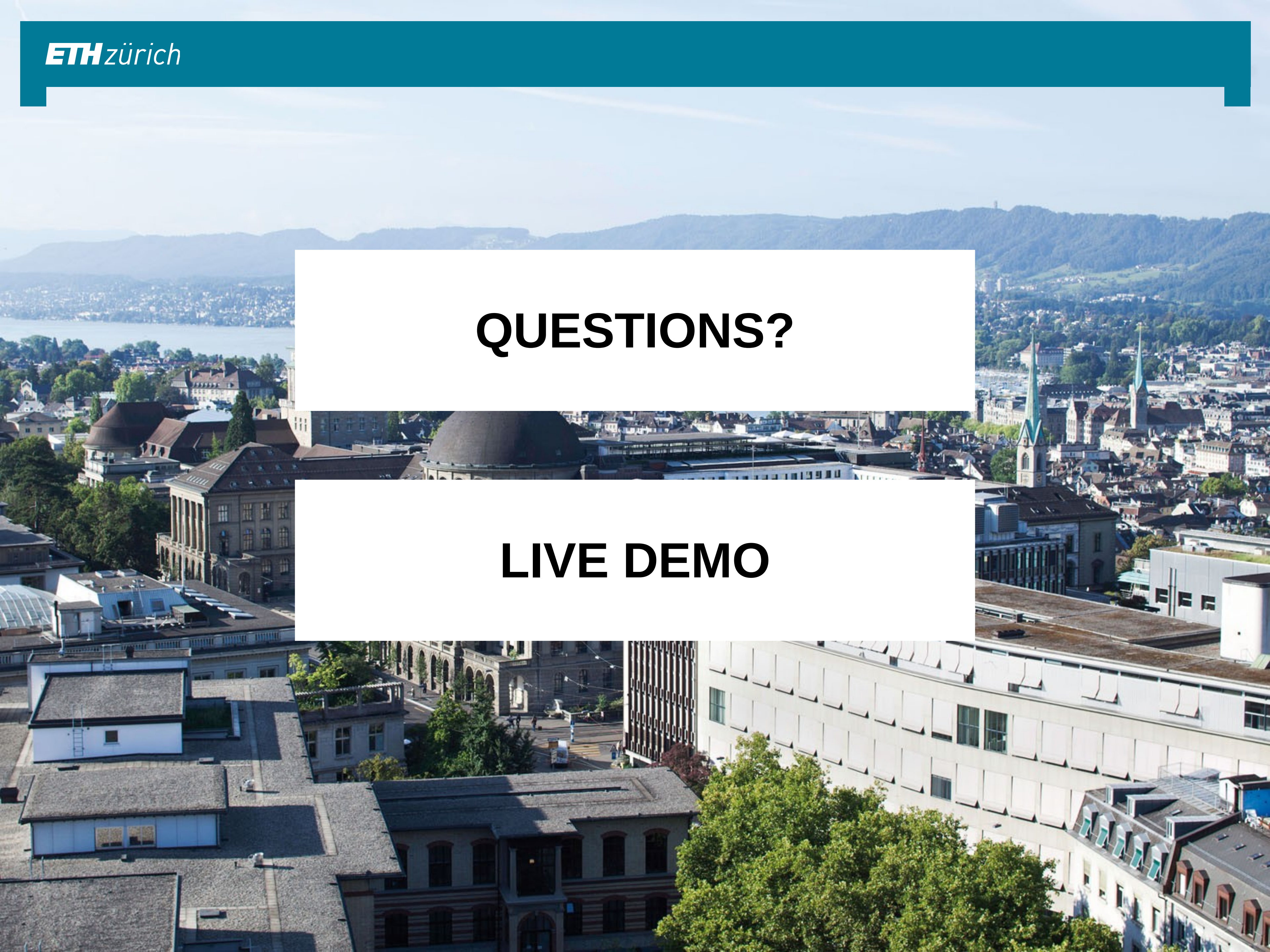


RESULTS: timing



CONCLUSION

- 5 Hz photometric odometry
  - leverage FPGA with stereo core
  - early abort of image pyramid
- further speedup trough FPGA possible



QUESTIONS?

LIVE DEMO