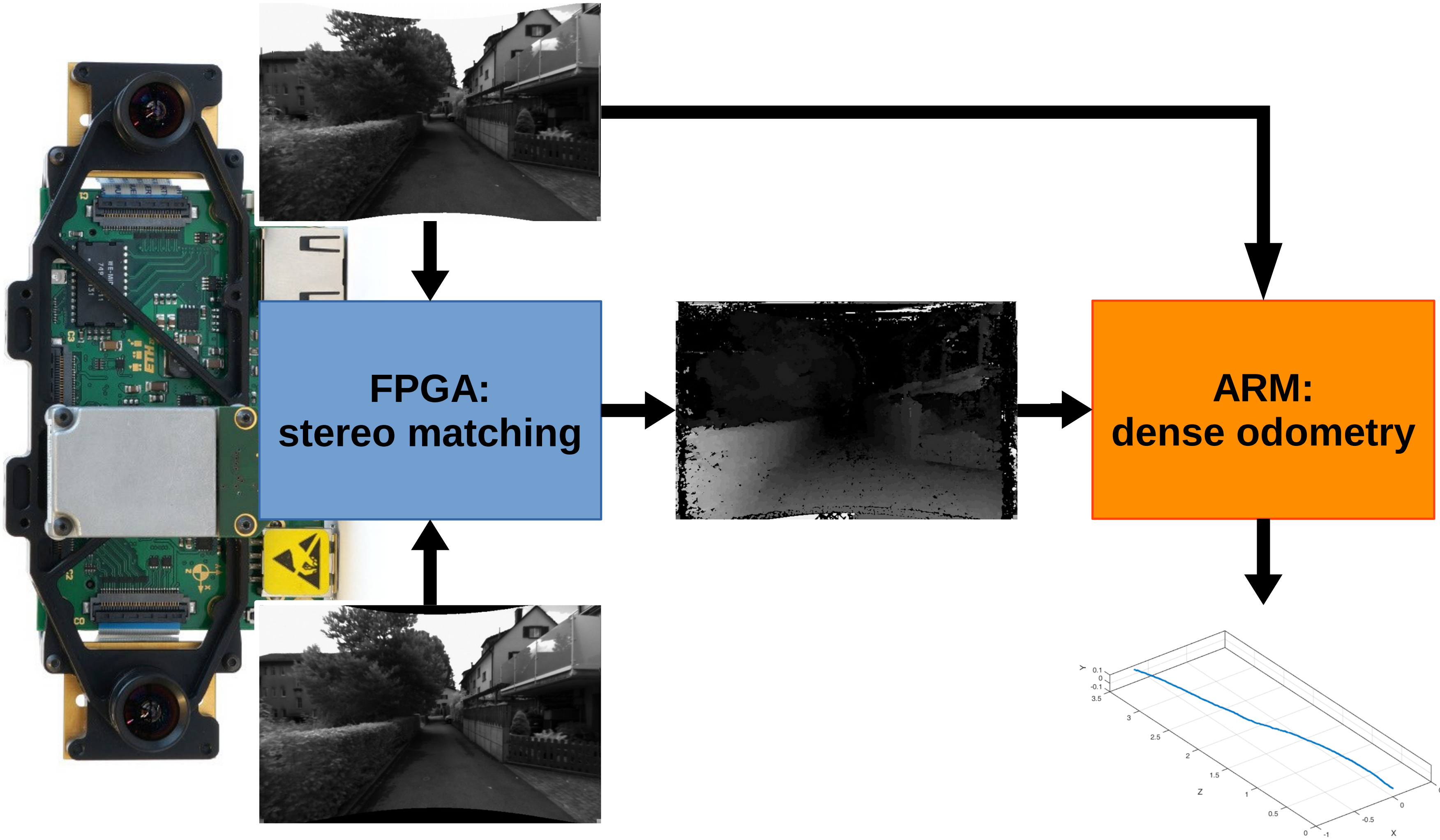
ETH zürich

EMBEDDED PHOTOMETRIC VISUAL ODOMETRY

Samuel Bryner

Bachelor Thesis
Supervised by Jörn Rehder and Pascal Gohl.

OUTLINE

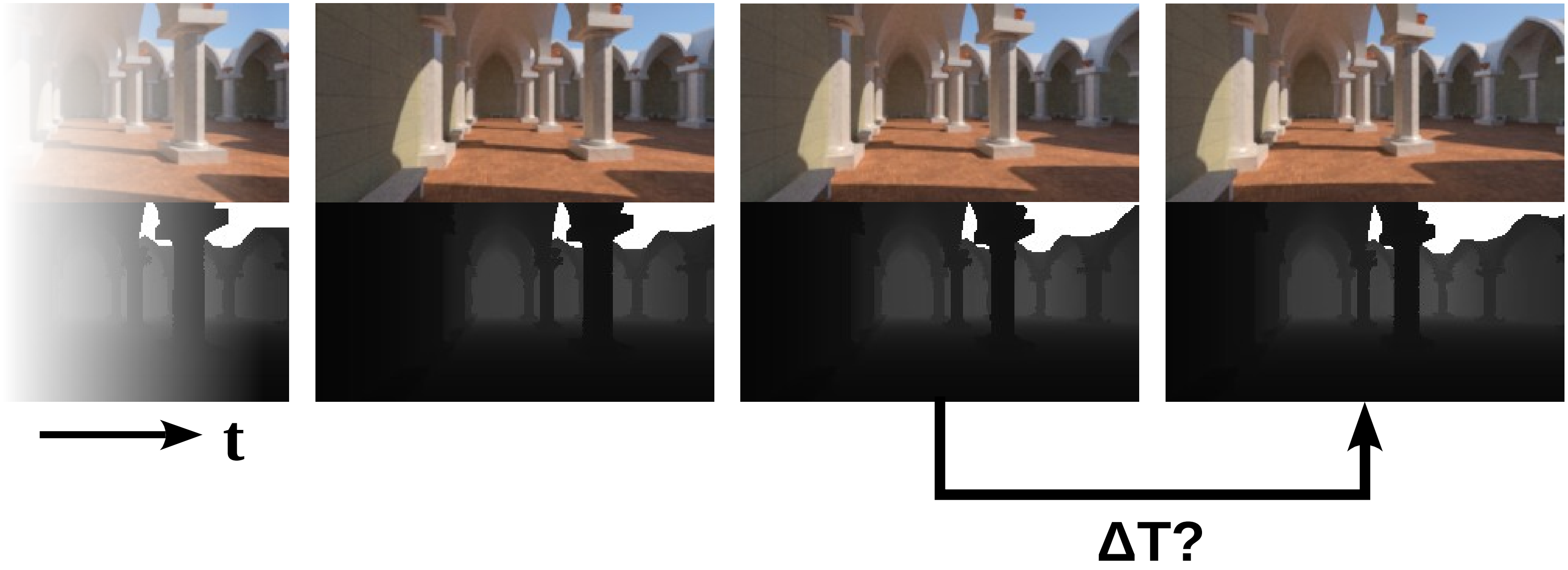


FPGA:
stereo matching
ARM:
dense odometry

MOTIVATION

- demonstration new integration of FPGA and ARM with embedded odometry
- stereo core enables photometric odometry instead of sparse odometry
- TODO: ref to Marcins Arbeit? http://students.asl.ethz.ch/upl_pdf/459-report.pdf

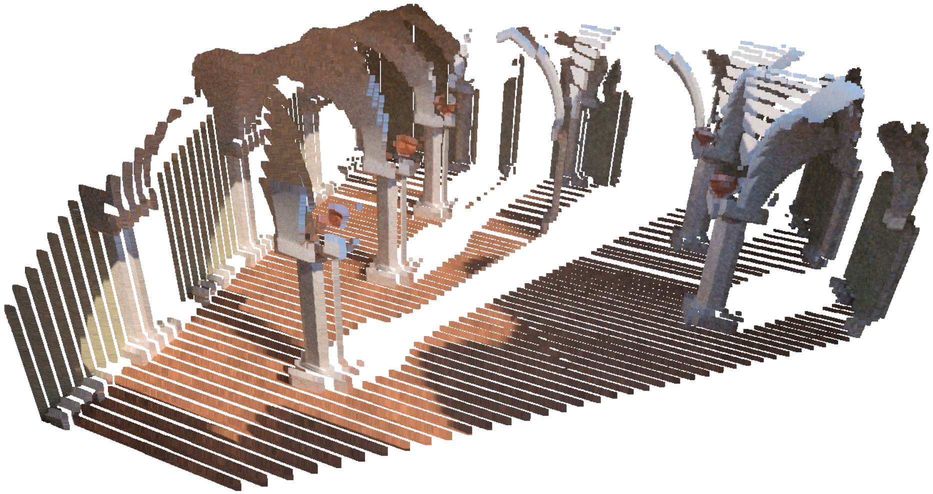
METHOD: the problem



METHOD: 1. project into space

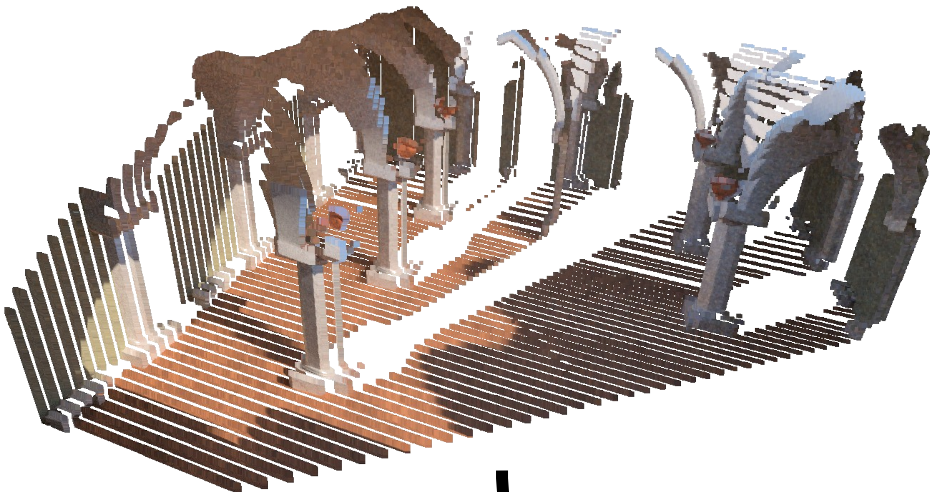
+

=



point cloud

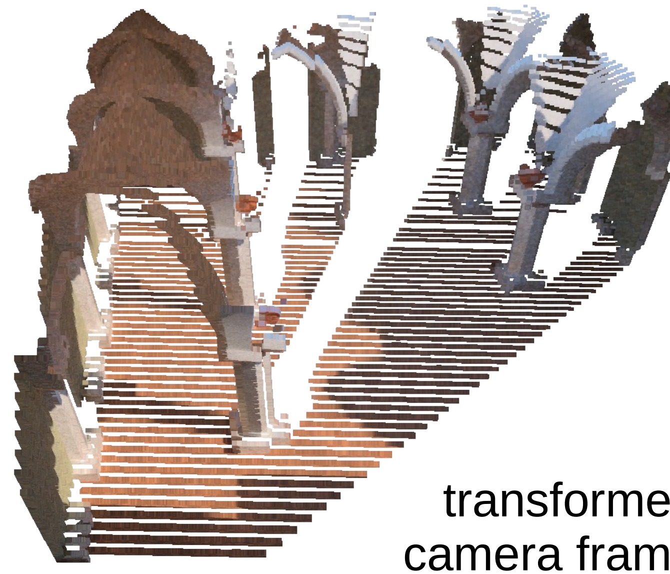
METHOD: 2. move camera trough space



original camera frame

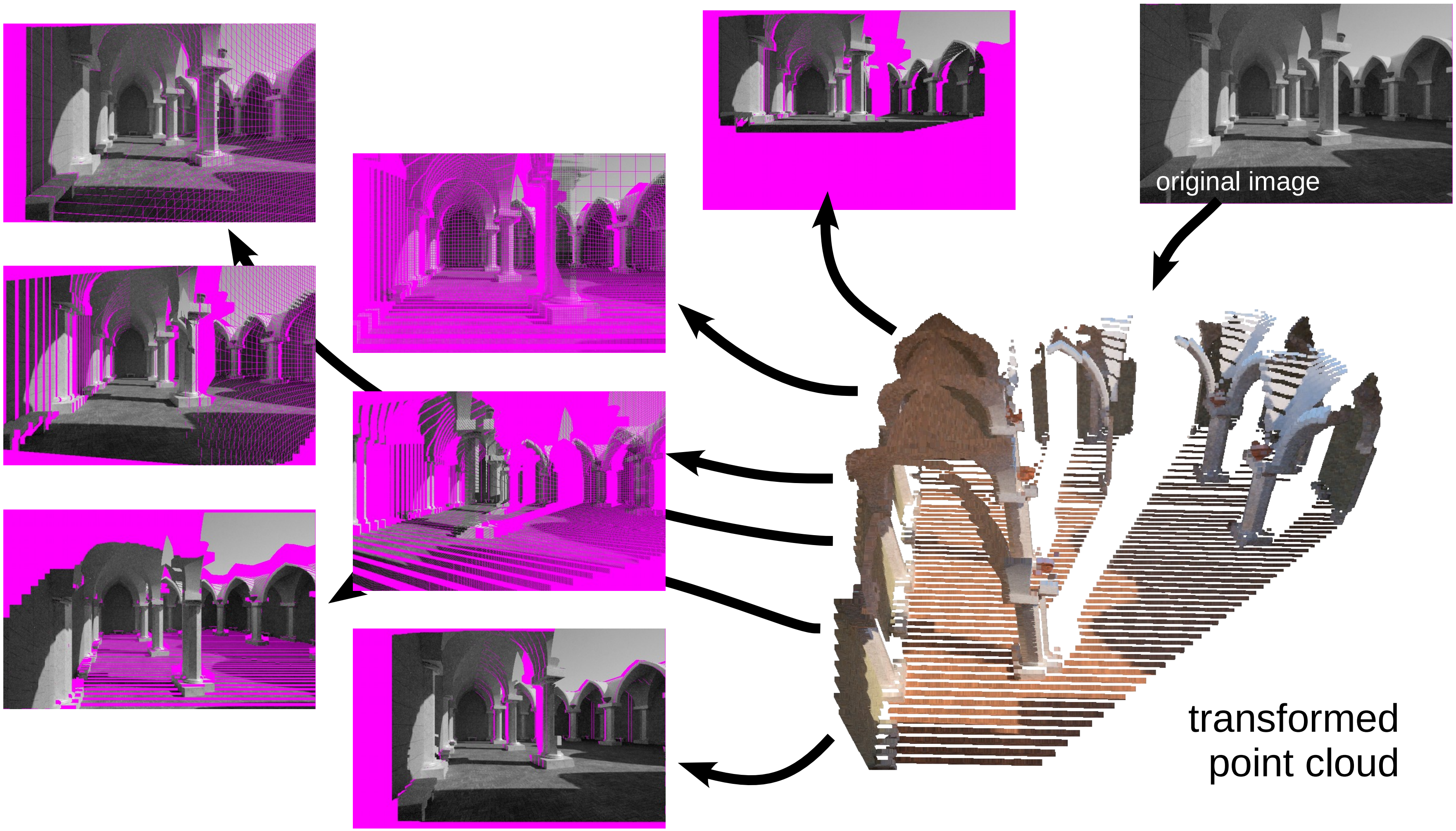
T translation & rotation

that's what we're actually looking for

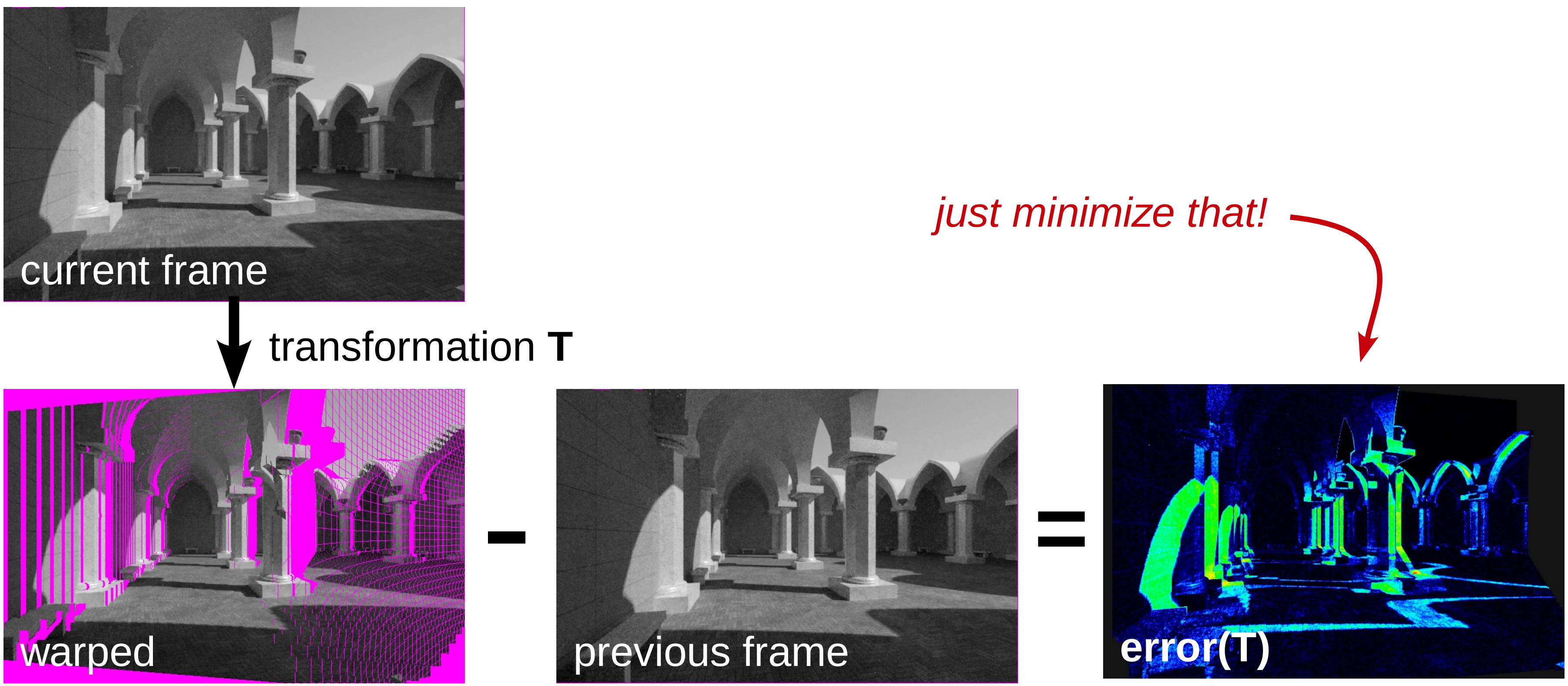


transformed camera frame

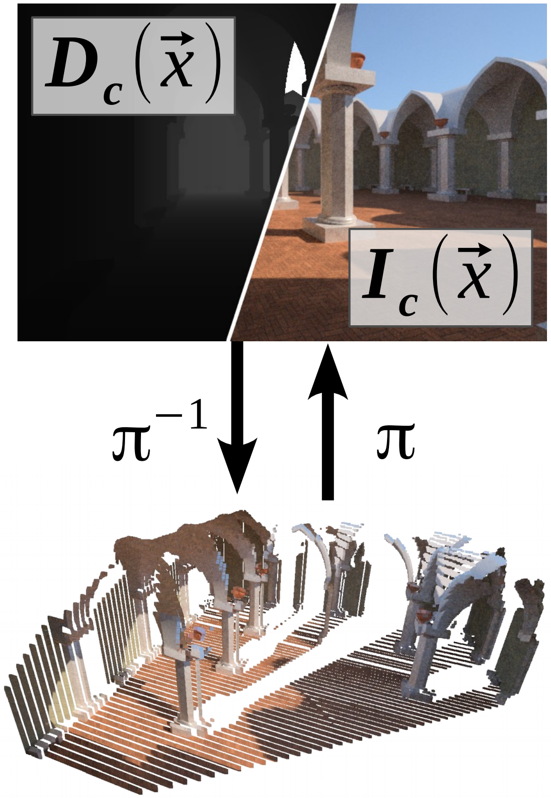
METHOD: 3. project back into camera



METHOD: 4. measure MSE



METHOD: formularizing the problem



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

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

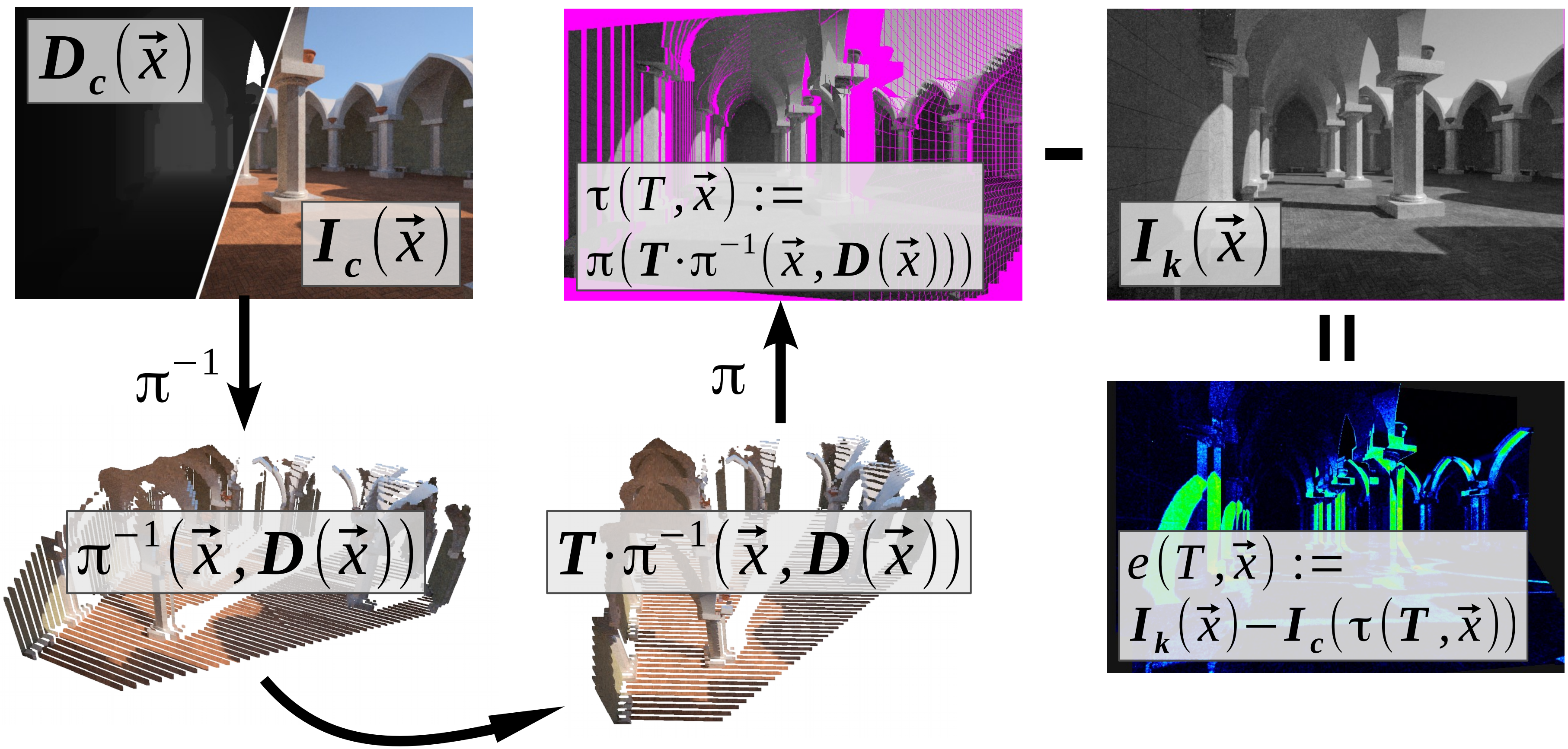
disparity: $\boldsymbol{D}(\vec{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}$

transformation: $\boldsymbol{T} \in \mathbb{R}^6$

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

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

METHOD: warping pipeline



transformation **T**

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

$$e\left(T,\vec{x}\right) := \boldsymbol{I_k}\left(\vec{x}\right) - \boldsymbol{I_c}\left(\tau\left(\boldsymbol{T},\vec{x}\right)\right)$$

for every pixel:

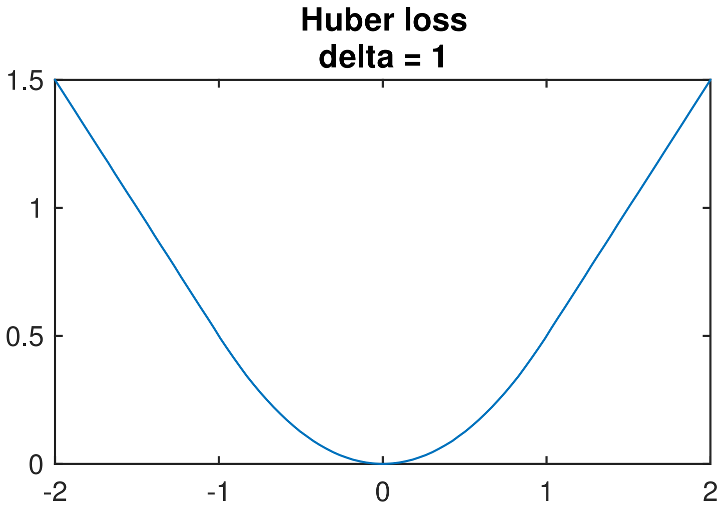
$$\hat{\boldsymbol{T}} = \underset{\boldsymbol{T}}{argmin} \sum_{\vec{x} \in \boldsymbol{I_k}} e\left(T,\vec{x}\right)^2$$

using Gauss Newton:

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

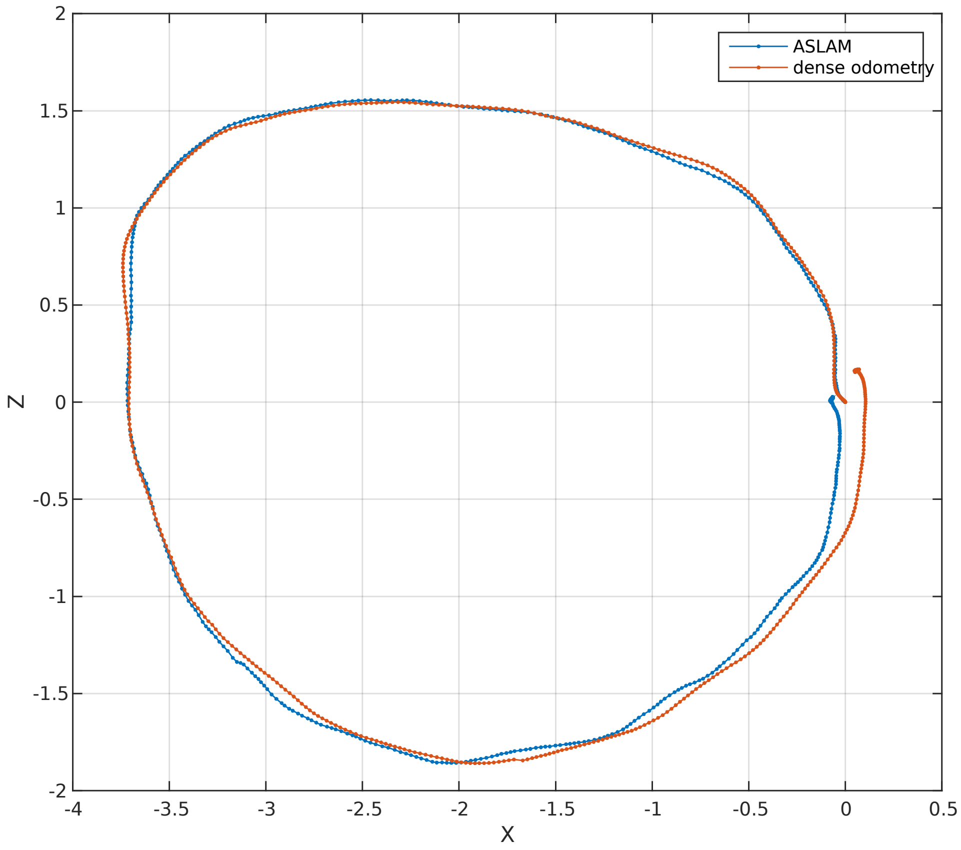
METHOD: optimizitations

- use image pyramid
- only use pixels with strong gradient
- Huber weights



RESULTS: photometric odometry VS. ASLAM

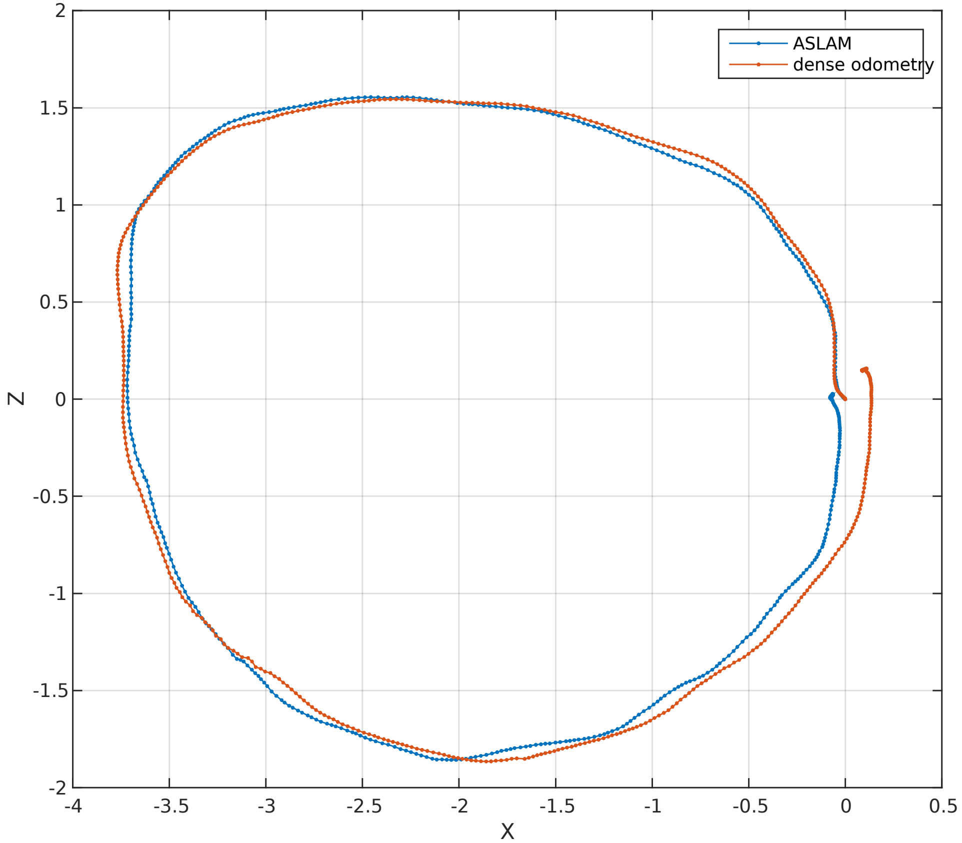
offline with OpenCV block matcher for stereo



units: meter

RESULTS: 1/16 of pixels (2x downscaled)

offline with OpenCV block matcher for stereo



units: meter

TODO: MORE RESULTS

- Error vs. time from previous plots
- circular trajectory with FPGA's stereo data
- timing data on visensor
- embedded photometric odometry with ~5 Hz

TODO: CONCLUSION

- FPGA is powerful co-processor
- embedded odometry is feasible, needs more optimization
- Was soll hier genau hin? Seh grad den Wald vor lauter Bäumen nicht mehr ;)
- Soll noch was zur 'photometric odometry' hin? „precise, easily parallelizable (->FPGA), not the most efficient way of doing odometry“