ETH zürich

EMBEDDED PHOTOMETRIC VISUAL ODOMETRY

Samuel Bryner

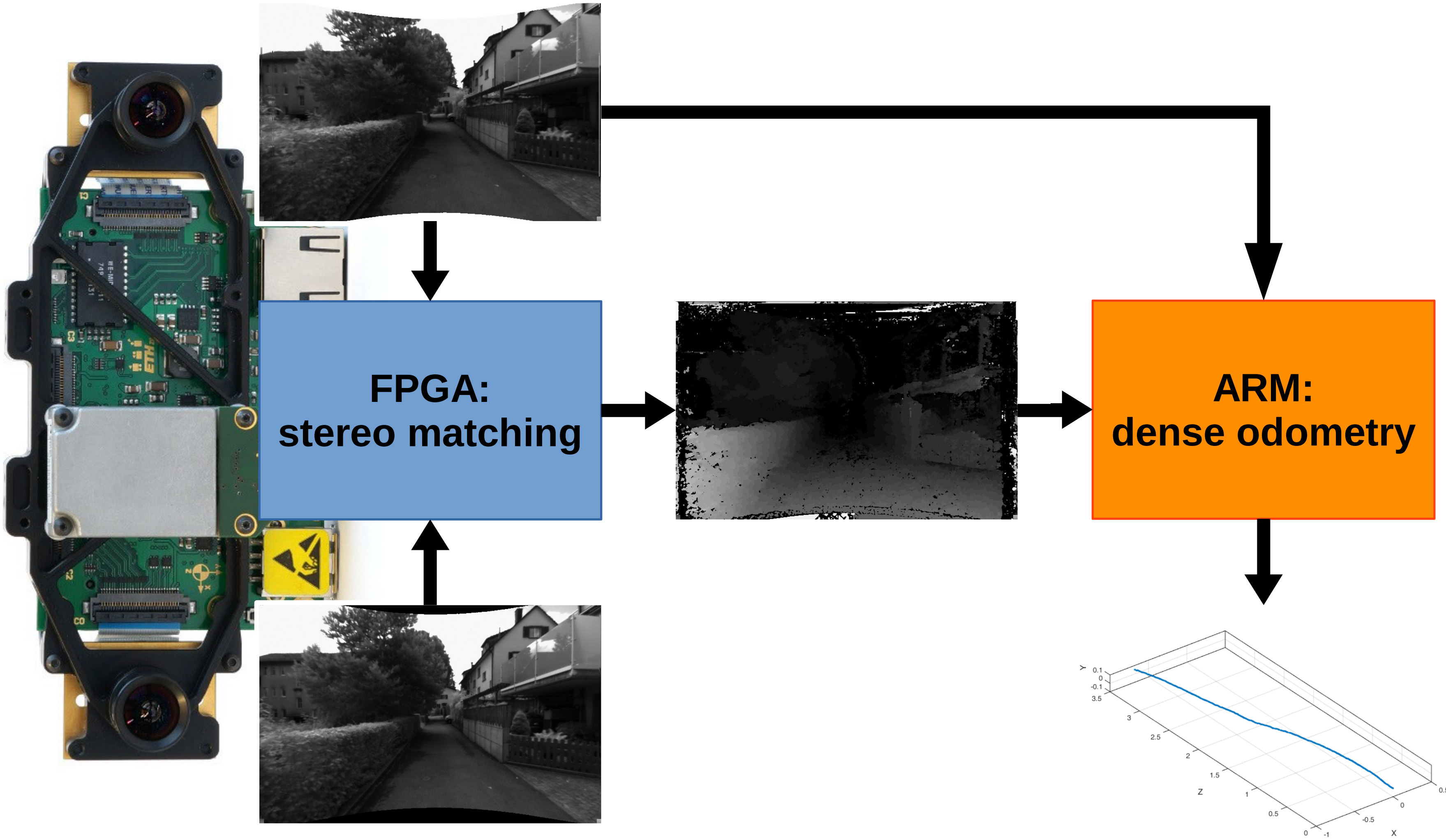
Bachelor Thesis
Supervised by Jörn Rehder and Pascal Gohl.

OUTLINE

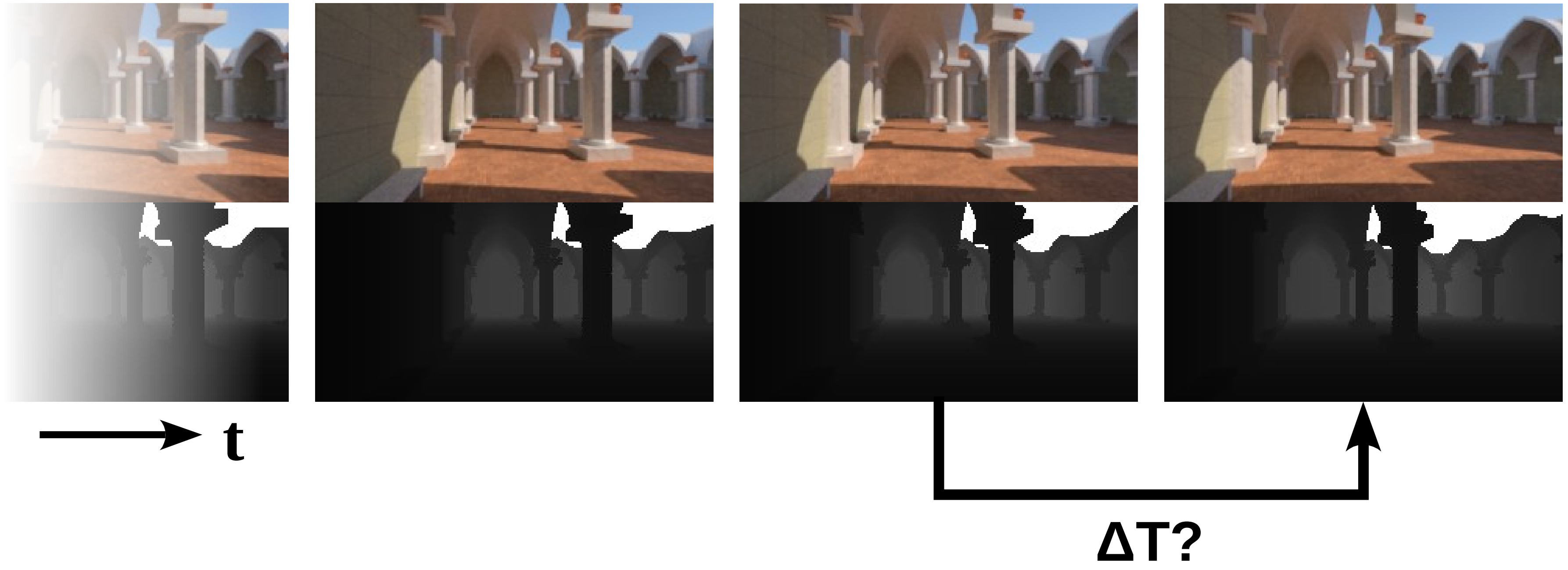
1. OUTLINE
2. MOTIVATION
3. METHOD: intuitive explanation
4. METHOD: actual math
5. RESULTS
6. CONCLUSION

MOTIVATION

- demonstrating 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


FPGA:
stereo matching
ARM:
dense odometry

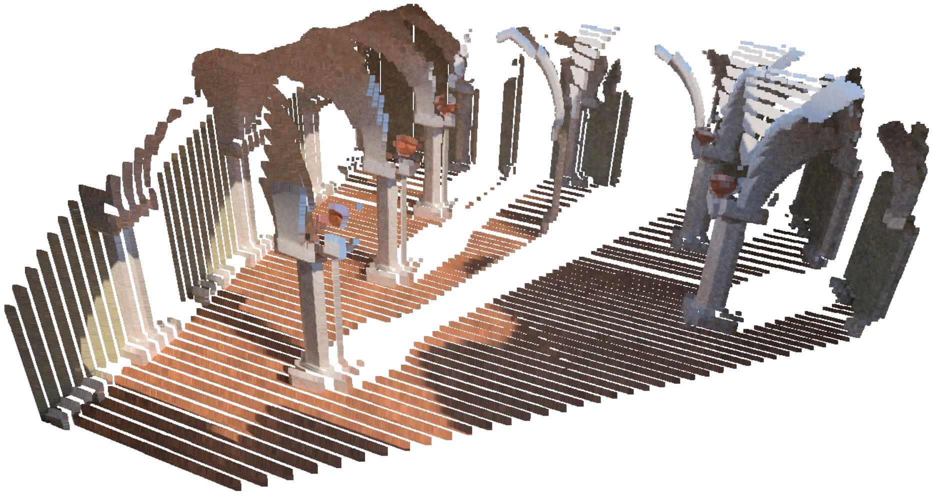
METHOD: the problem



METHOD: 1. project into space

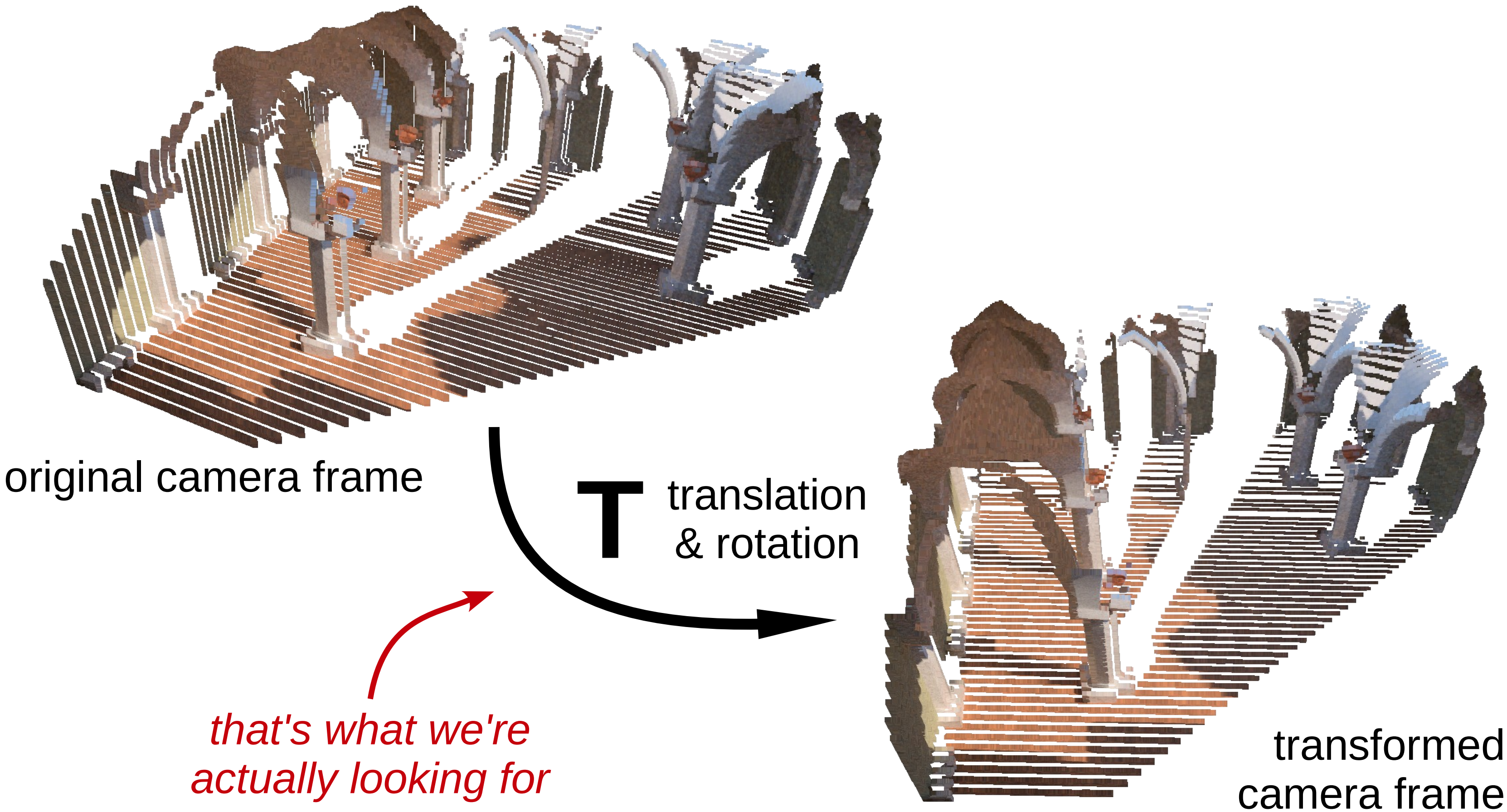
$+$

$=$

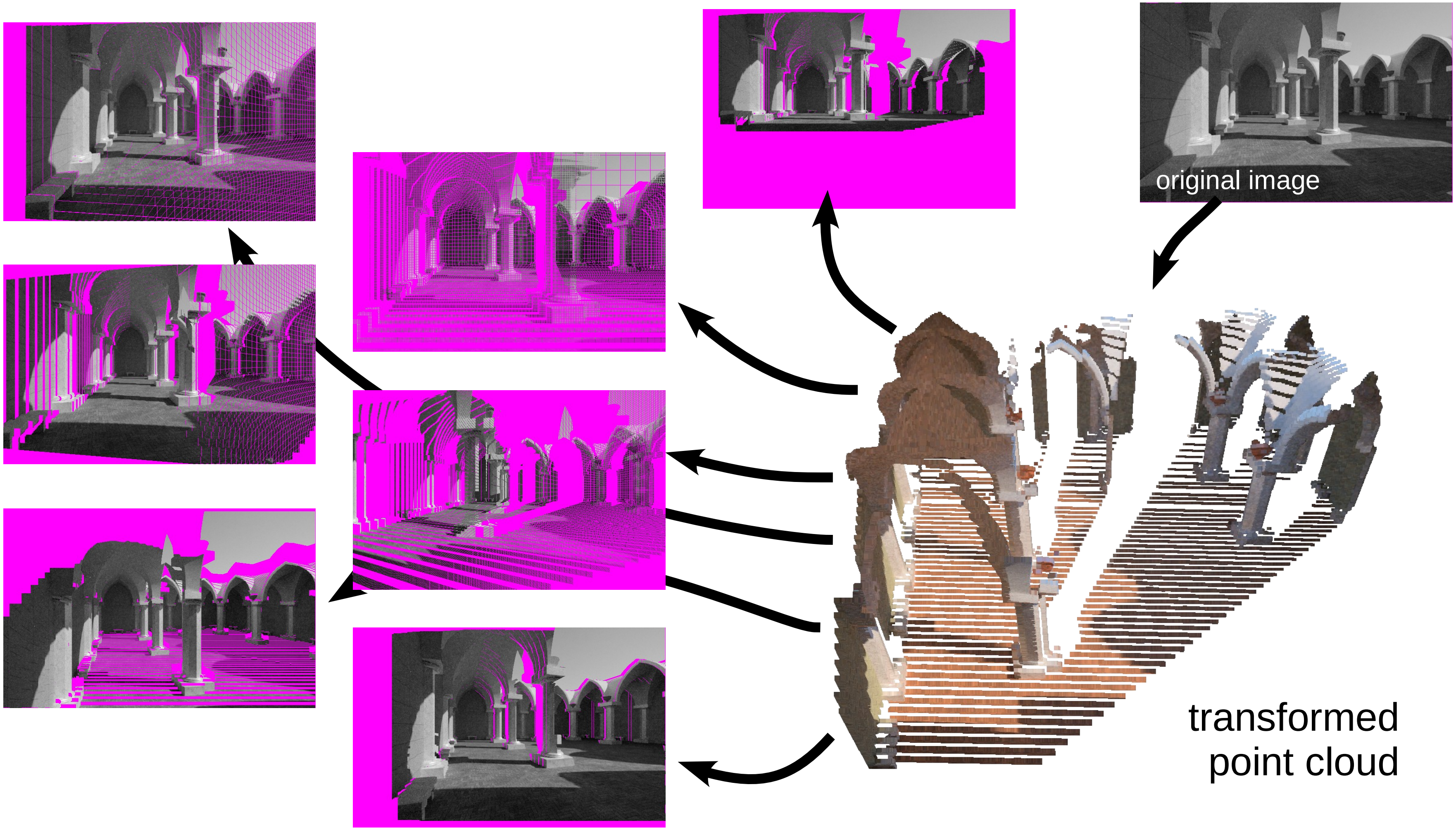


point cloud

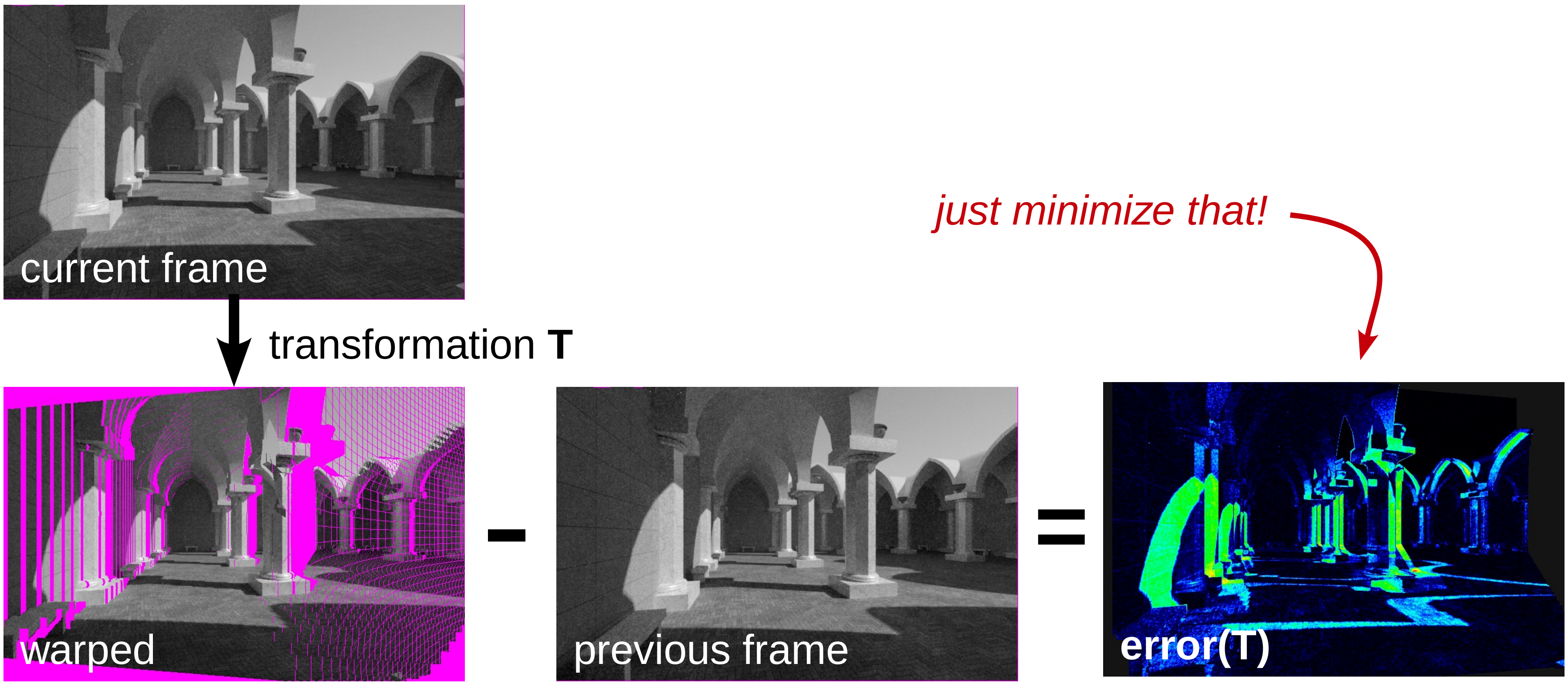
METHOD: 2. move camera trough space



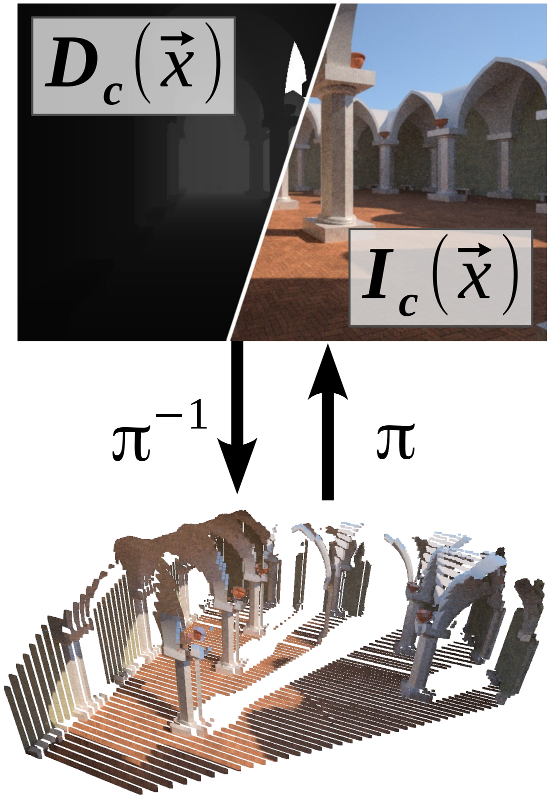
METHOD: 3. project back into camera



METHOD: 4. measure MSE



METHOD: formularizing the problem



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

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

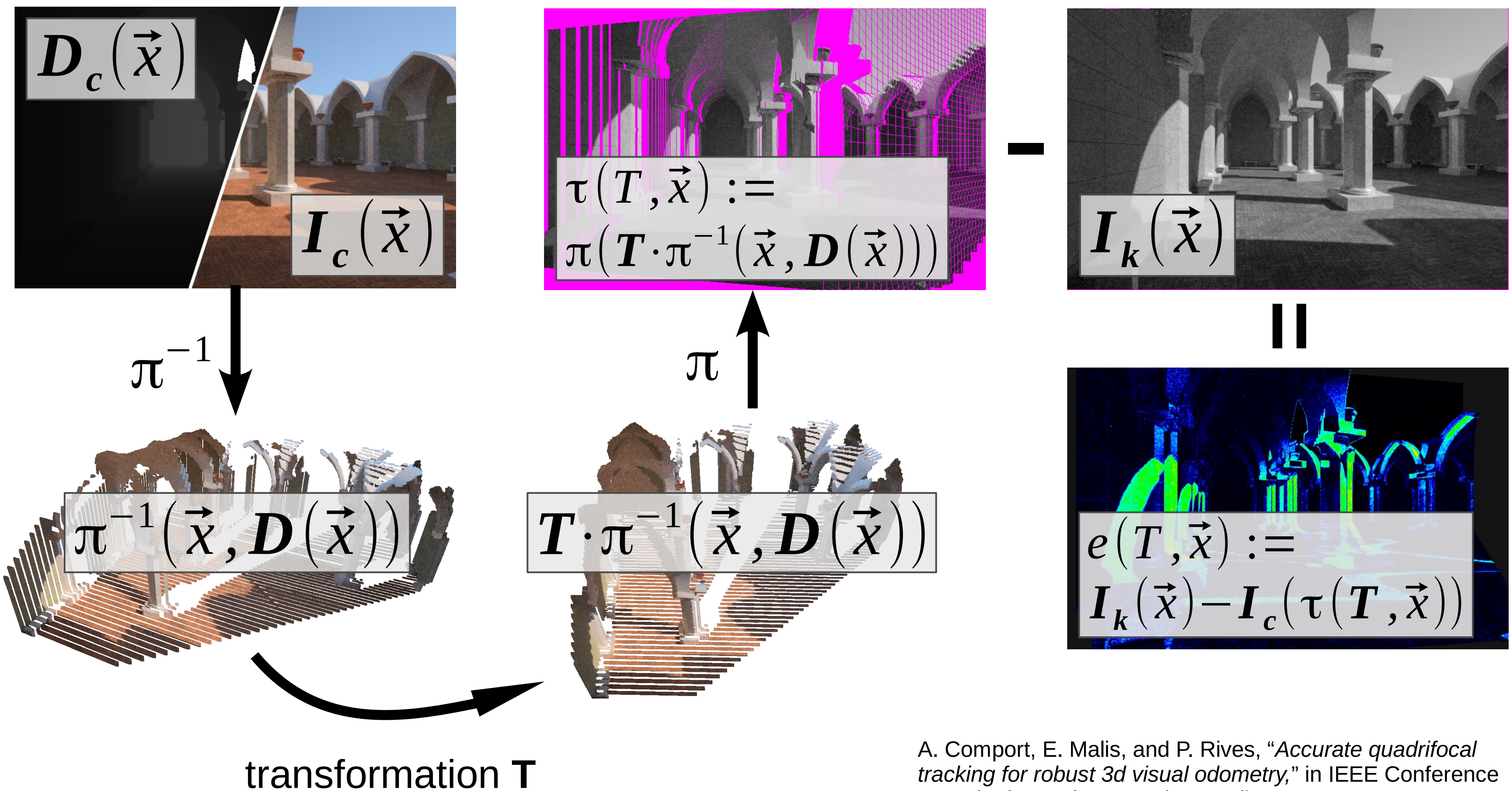
disparity: $D(\vec{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}$

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

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

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

METHOD: warping pipeline



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

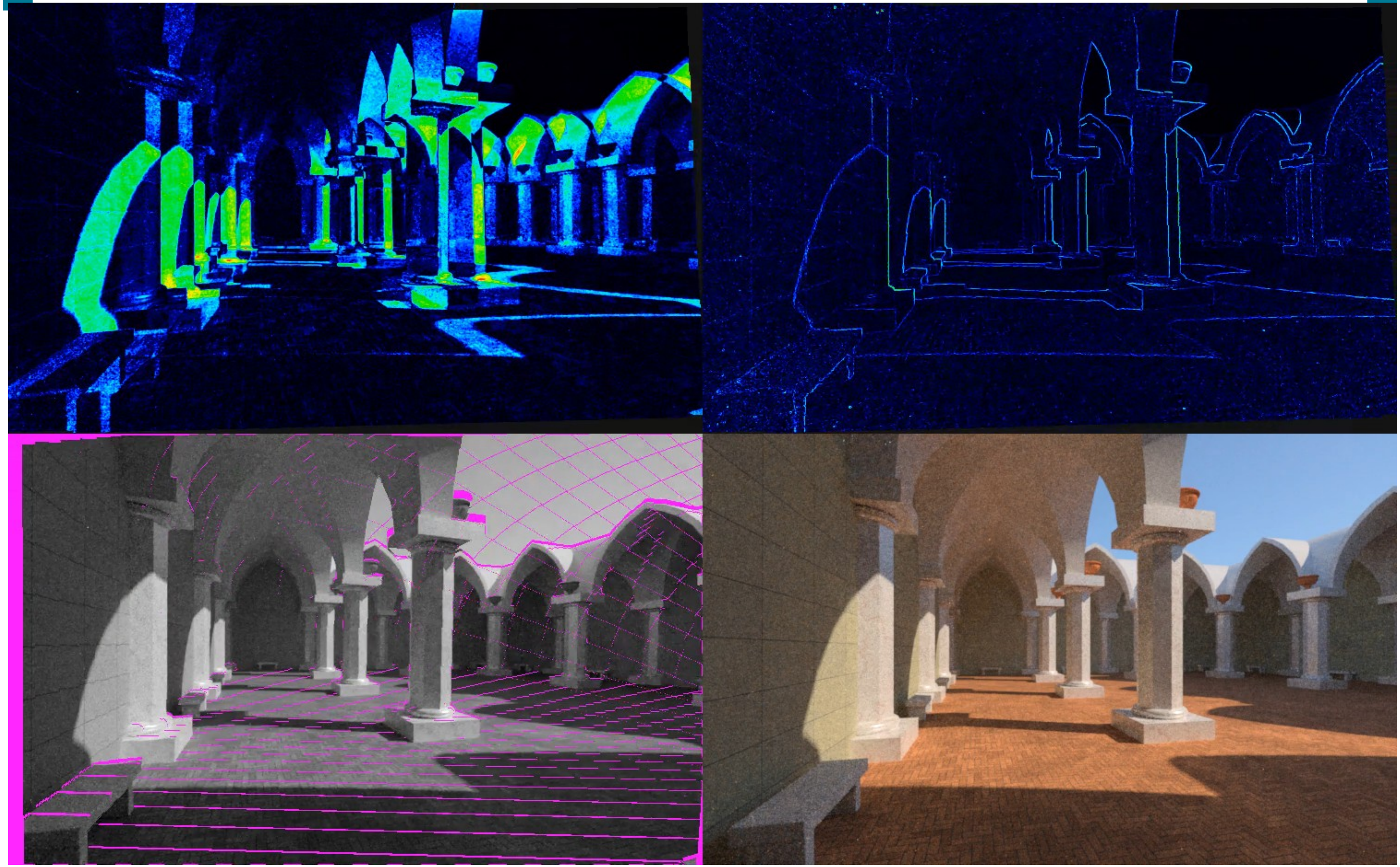
$$e(T,\vec{x}) := \boldsymbol{I}_{\boldsymbol{k}}(\vec{x}) - \boldsymbol{I}_{\boldsymbol{c}}(\tau(\boldsymbol{T},\vec{x}))$$

for every pixel:

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

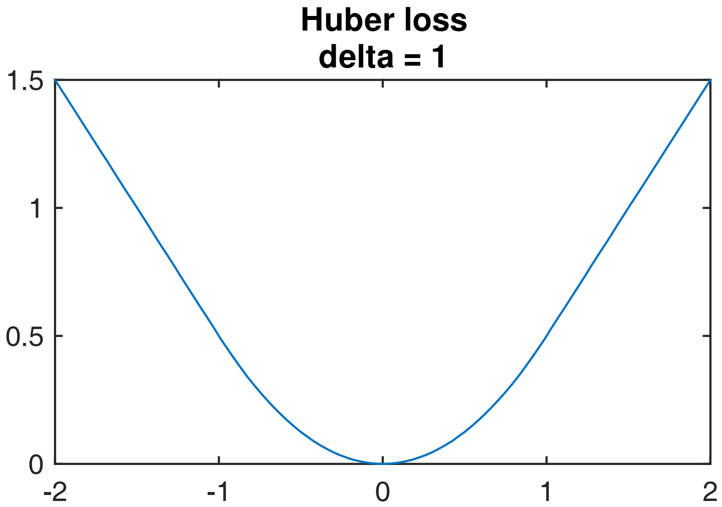
using Gauss Newton:

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



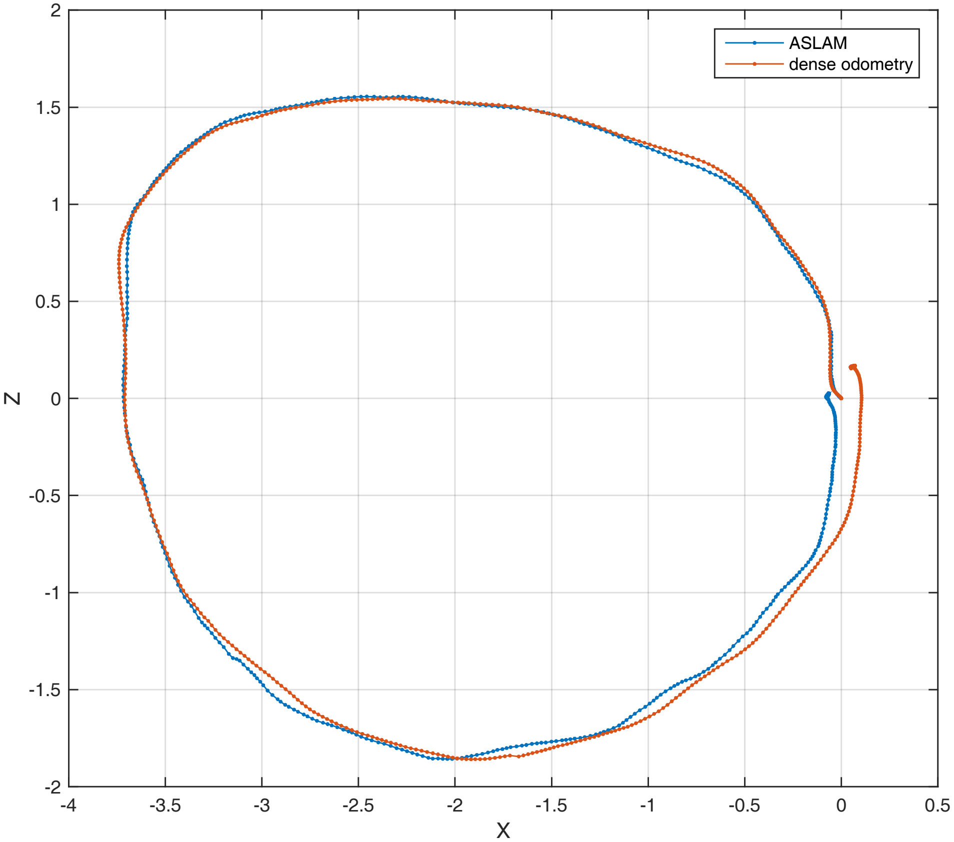
METHOD: optimizations

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



RESULTS: photometric odometry VS. ASLAM

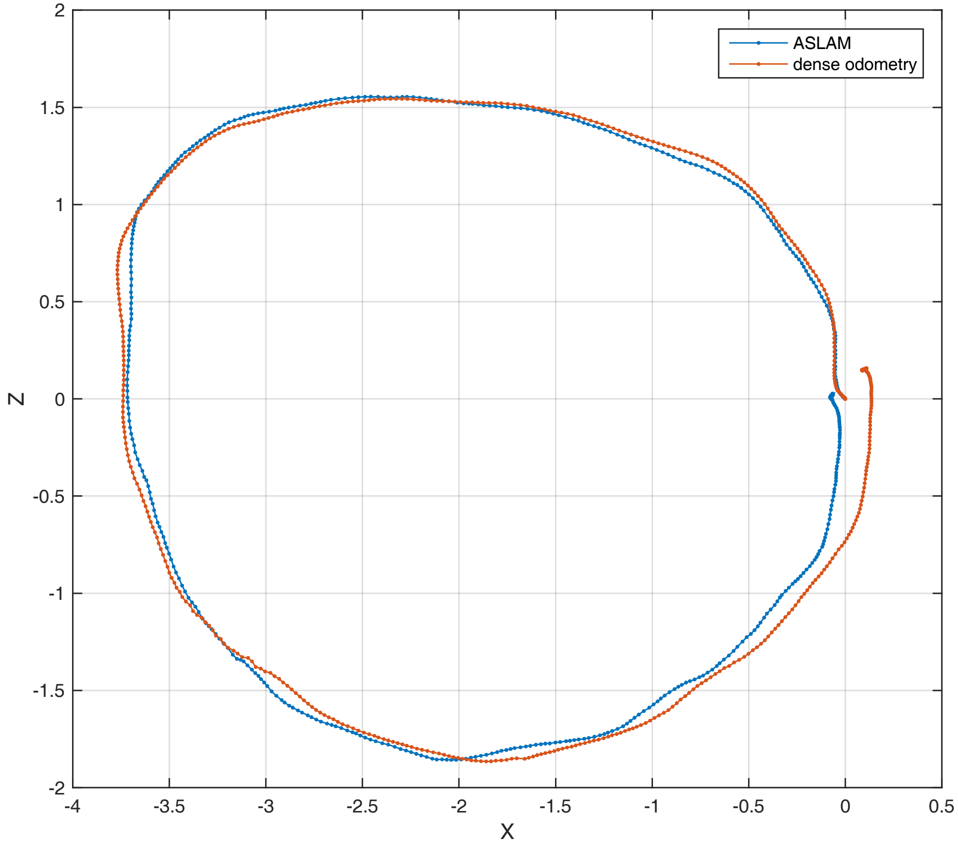
offline with OpenCV block matcher for stereo



units: meter

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

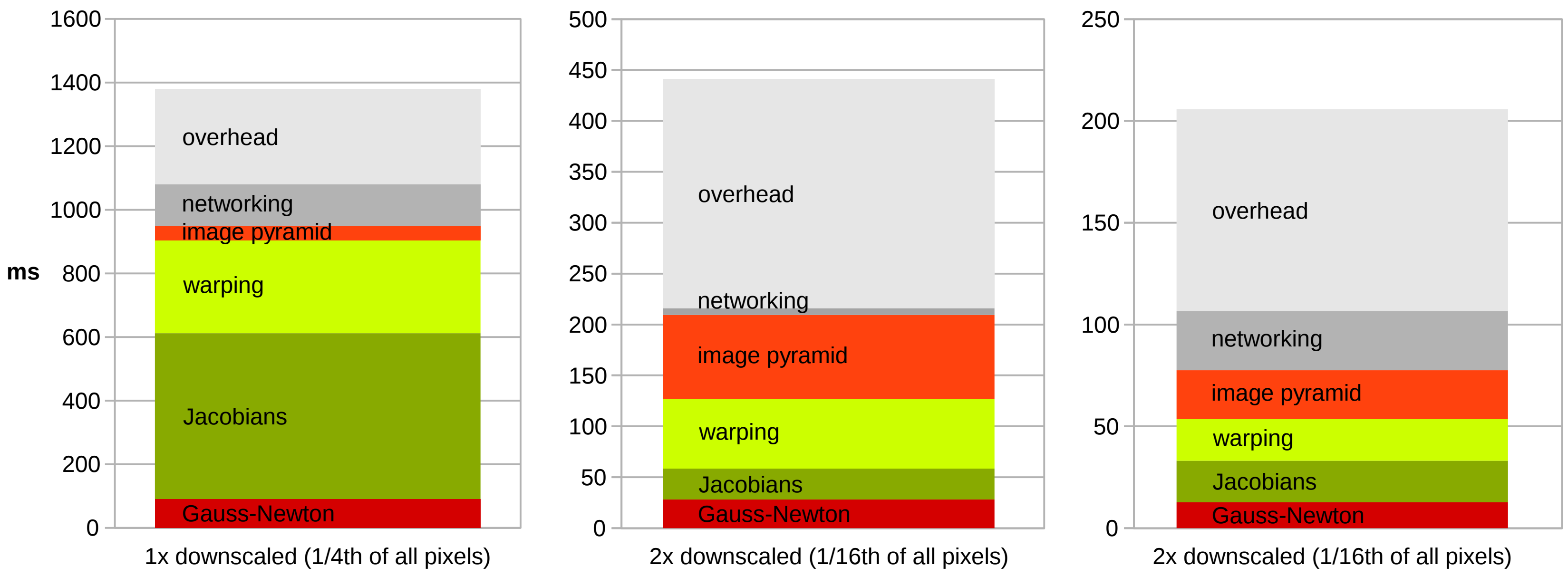
offline with OpenCV block matcher for stereo



units: meter

RESULTS: timing

TODO: throw out one of these last two graphs:



TODO: MORE RESULTS

- Error vs. time from previous plots
- circular trajectory with FPGA's stereo data

TODO: CONCLUSION

- existing: dense matching on FPGA
- here: using dense for odometry
- possible future: odometry out-of-the-box by moving more towards FPGA