Multi-Level Mapping: Real-time Dense Monocular SLAM

W. Nicholas Greene, Kyel Ok, Peter Lommel, and Nicholas Roy

DRAPER · MIT AEROASTRO · CSAIL

1. Motivation

- Would like to estimate ***dense 3D maps online*** for ***high-speed autonomous navigation***
- ***Cameras*** are information ***dense, low SWaP, inexpensive,*** but
  - ***Sparse SLAM:*** Cheap, but ***low map fidelity***
  - ***Dense SLAM:*** High map fidelity, but ***expensive***
  - ***Semi-dense SLAM:*** Balance, but maps still ***noisy with holes***

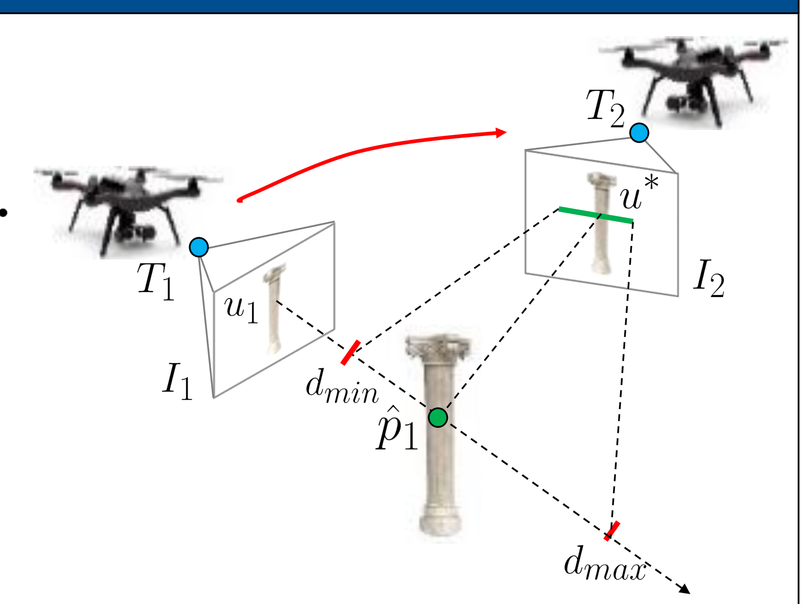
Can we compute dense 3D maps with the efficiency of semi-dense methods?

2. Monocular SLAM

Given images $I_1, I_2, \ldots$

Localize: estimate camera pose $T_1, T_2, \ldots$

Map: estimate 3D points $\{p_i\}$



Which pixels should be mapped?

Corners? Gradients? Low-texture regions?

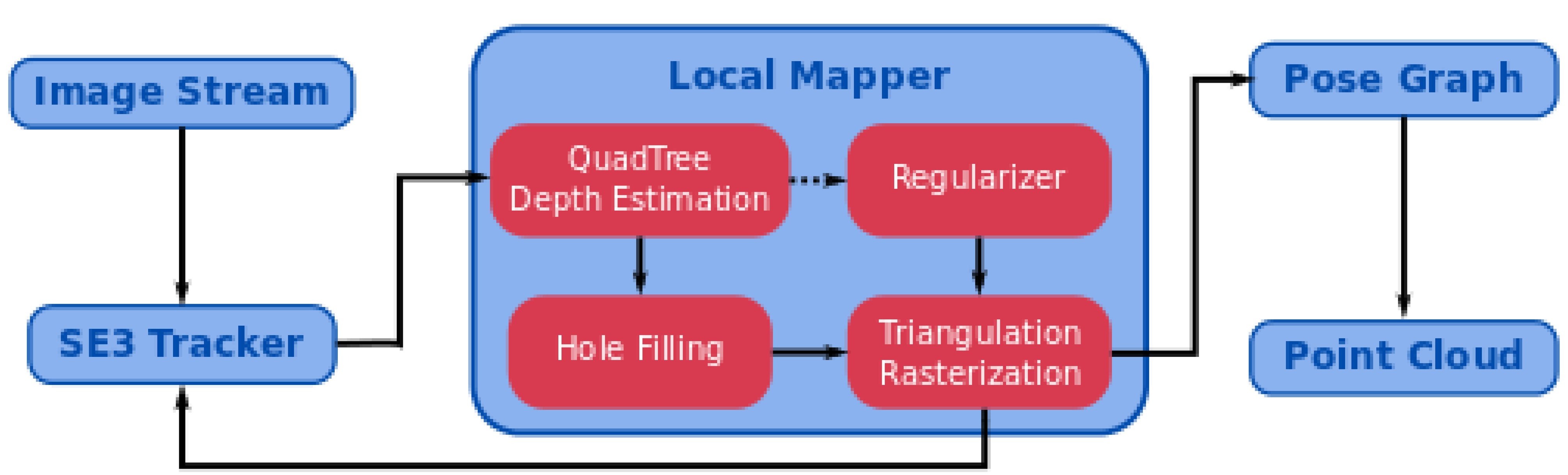
Every pixel?

3. Key Insights

1. Low-texture regions ***correlated with planar structure*** - don't need to reason about every pixel to reconstruct scene, but don't need to completely discard data. Can estimate depth at ***multiple image scales*** to take advantage of all available texture.
2. World is locally ***smooth***. Expensive global regularization techniques can be accelerated by applying at ***multiple image scales.***

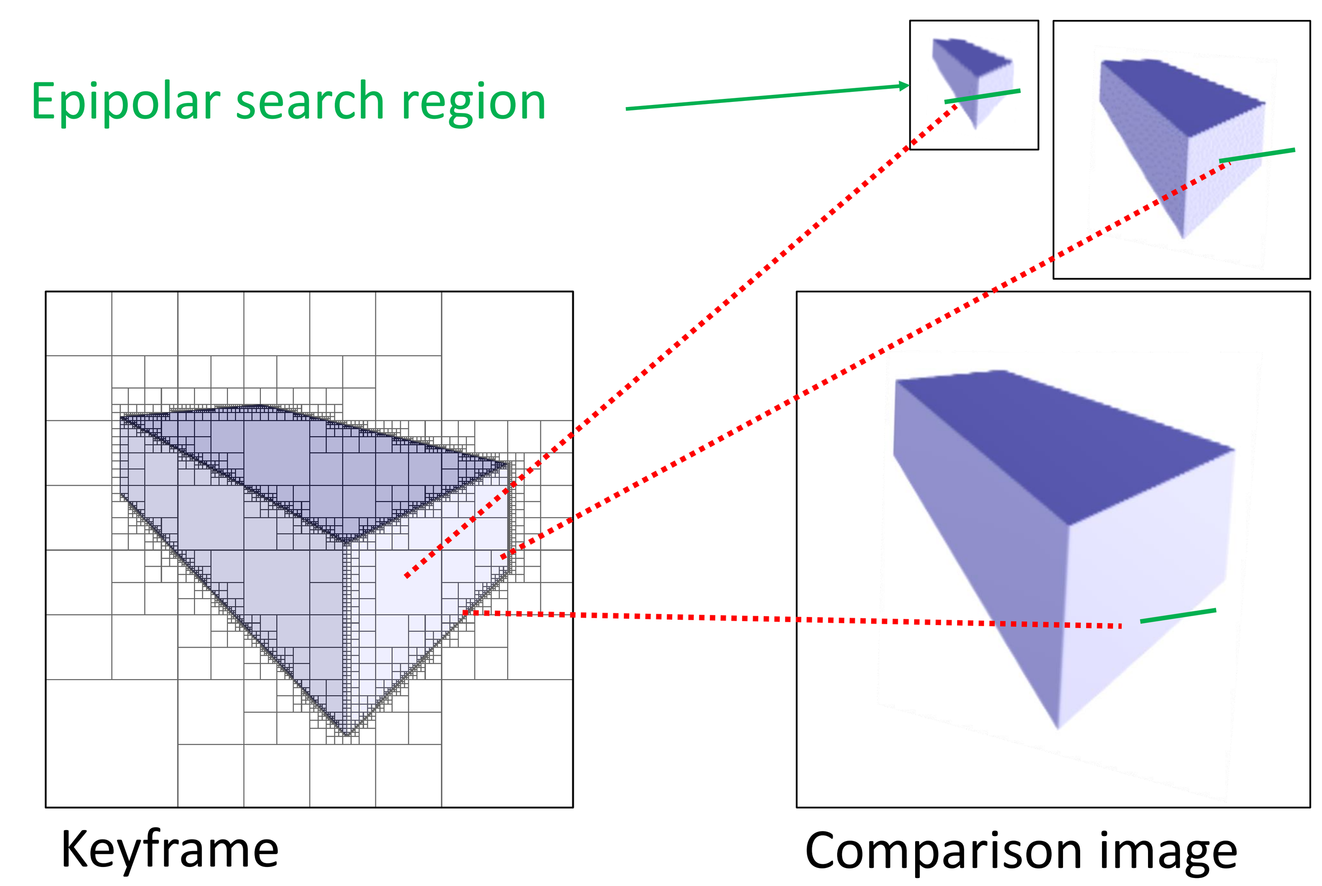
Yes! – By estimating depth on a quadtree to leverage all available texture and smoothing using a variational regularizer

4. Pipeline



5. Quadtree Depthmaps

1. Convert keyframe image to ***quadtree*** based on texture
2. Perform epipolar search for every node at ***corresponding image scale***



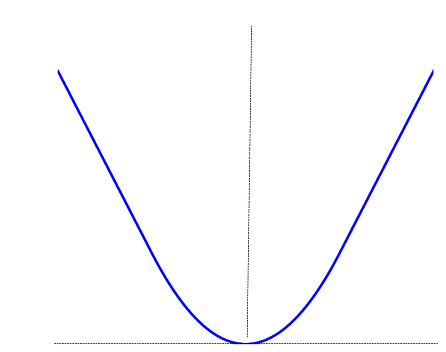
6. Smoothing

Estimate smoothed inverse depthmap ξ from noisy inverse depths $\mathbf{z}$ by minimizing ***variational functional***

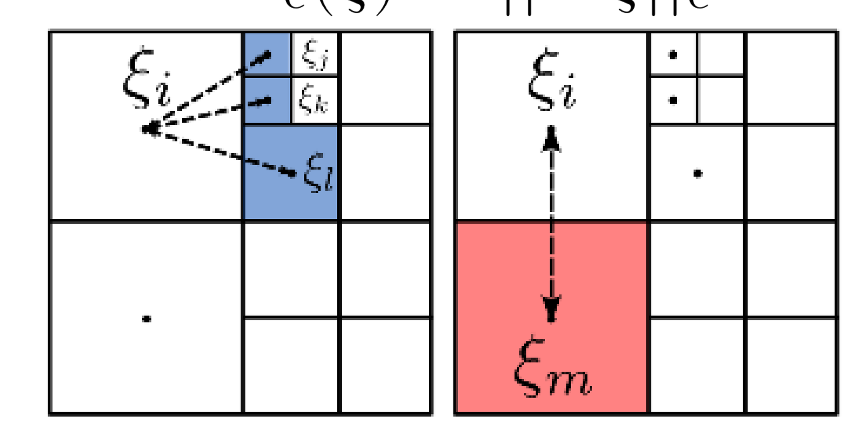
$$\min_{\xi} \; TV_{\epsilon}(\xi) + \lambda ||\mathbf{W}(\xi - \mathbf{z})||_1$$

Regularizer term · Data term · Scaling factor

Total Variation-Huber $TV_{\epsilon}(\xi) = ||\mathbf{D}\xi||_{\epsilon}$



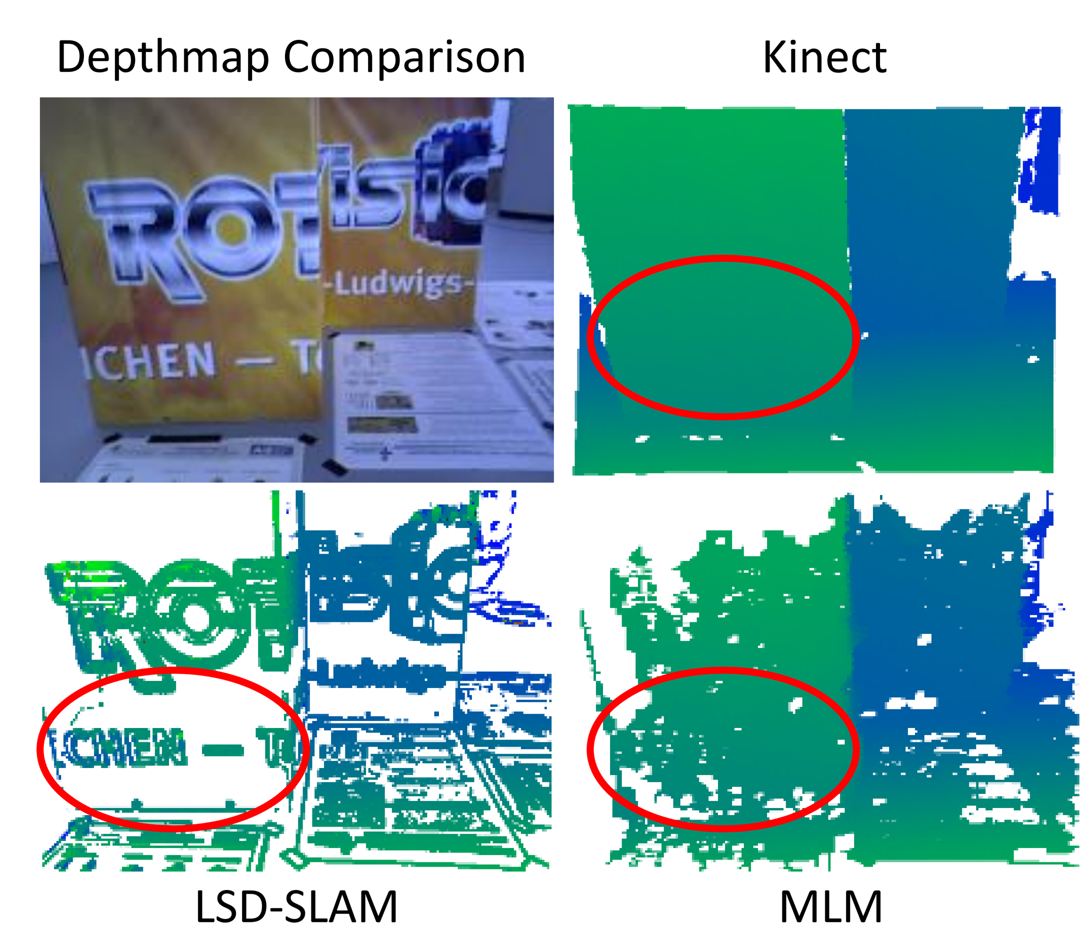
Huber Norm



Quadtree Discrete Derivative

7. Results

Evaluated against LSD-SLAM (Engel et al. ECCV 2014) using TUM RGB-D SLAM Benchmarks



Lower inverse depth error, denser keyframes, same running time, CPU-only