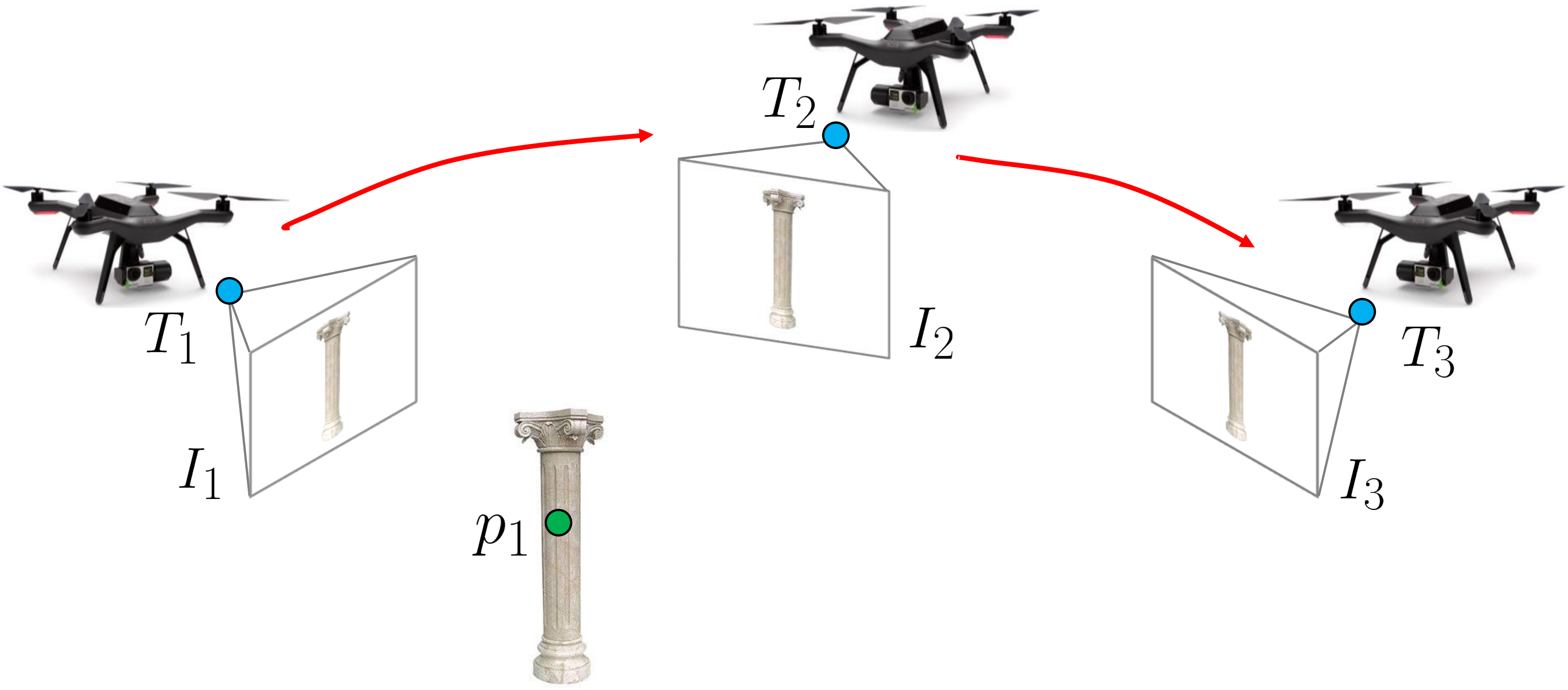
Multi-Level Mapping: Real-time Dense Monocular SLAM

W. Nicholas Greene[1], Kyel Ok[1], Peter Lommel[2], and Nicholas Roy[1]

[1]MIT Computer Science and Artificial Intelligence Laboratory (CSAIL)
[2]Draper

Monocular SLAM

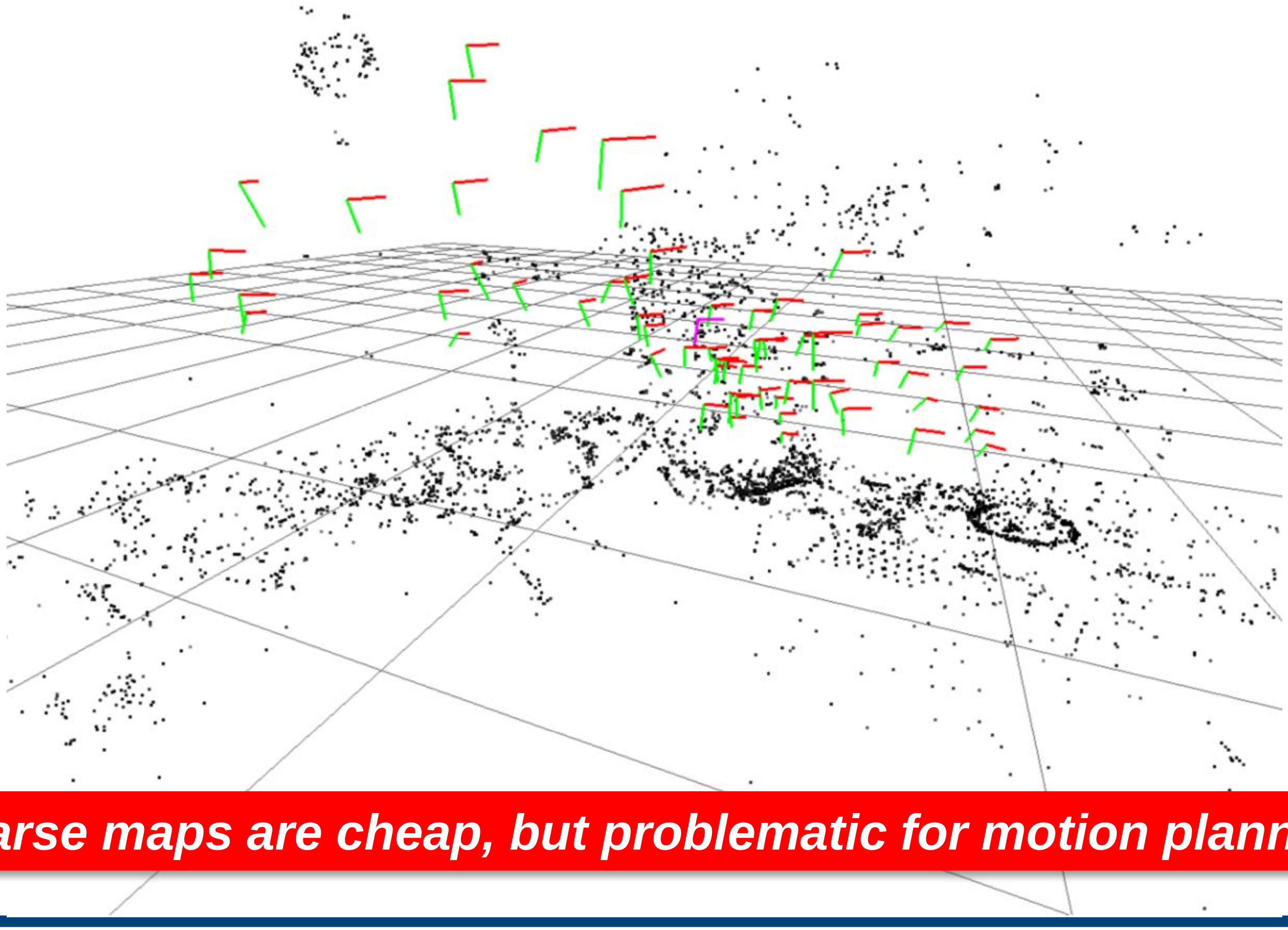
Simultaneous Localization and Mapping



We want to estimate dense 3D maps online using low-SWaP cameras to enable high-speed autonomous navigation, but…

Prior Work: Sparse Methods

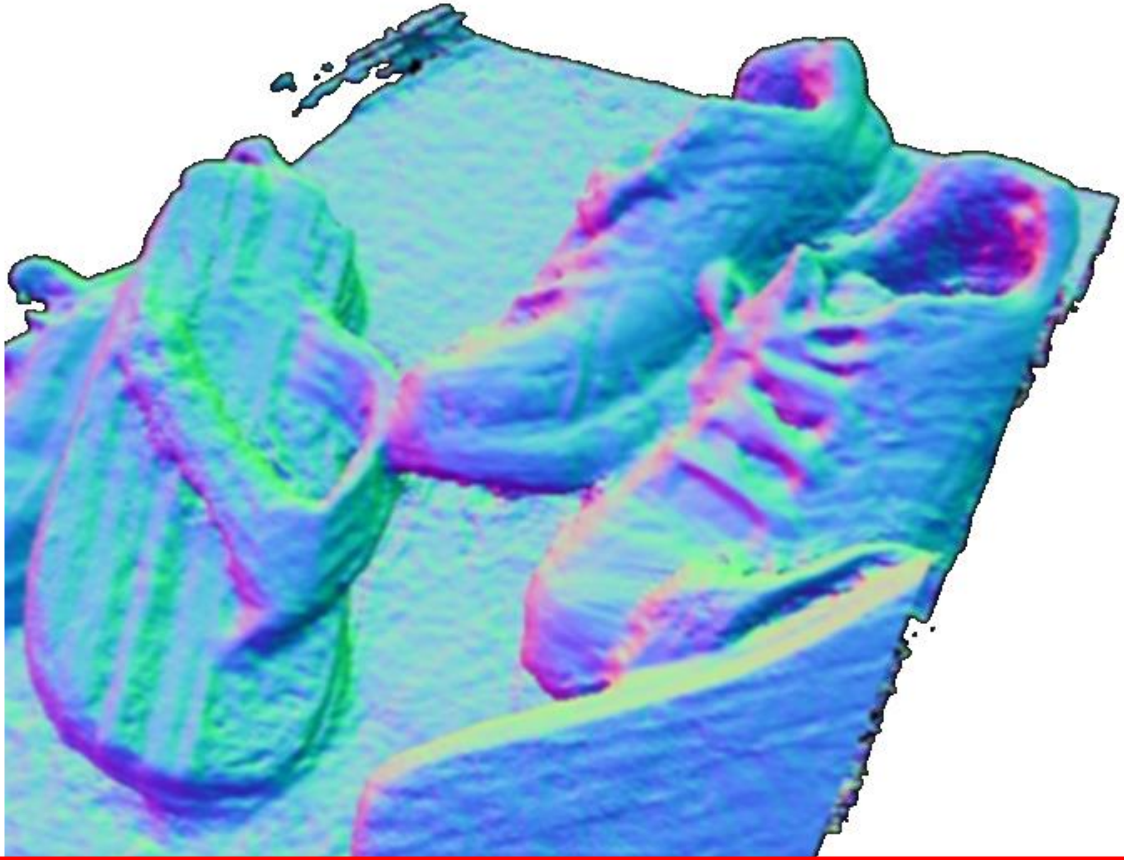
Use ***features*** (FAST, SIFT, etc.) extracted from images to estimate ***sparse point cloud***
MonoSLAM (Davison, ICCV 2003, 2007), PTAM (Klein and Murray, ISMAR 2007)



Sparse maps are cheap, but problematic for motion planning

Prior Work: Dense Methods

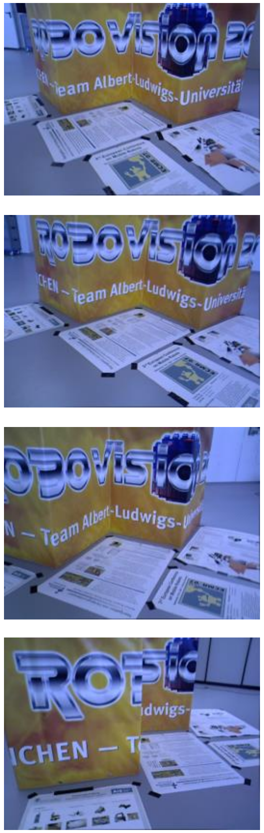
Use ***raw pixel intensities*** and ***GPU acceleration*** to estimate ***dense mesh.***
DTAM (Newcombe et al., ICCV 2011), MonoFusion (ISMAR 2013)



Dense maps are accurate, but expensive to compute and small-scale

Prior Work: Semi-dense Methods

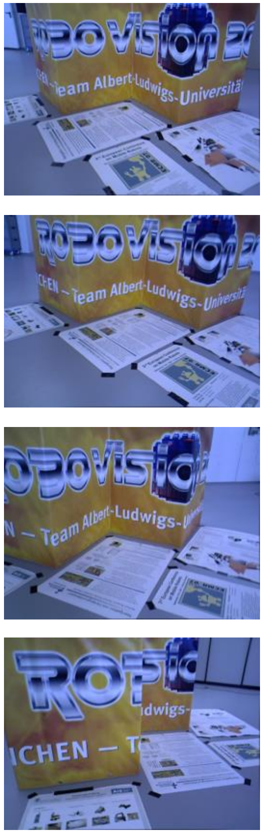
Use only pixels with ***gradient*** to estimate ***semi-dense*** point cloud
LSD-SLAM (Engel et al. ICCV 2013, ECCV 2014)



Semi-dense maps have holes in low-texture regions

Prior Work: Semi-dense Methods

Use only pixels with ***gradient*** to estimate ***semi-dense*** point cloud
LSD-SLAM (Engel et al. ICCV 2013, ECCV 2014)



Can we estimate dense geometry without sacrificing speed?

Multi-Level Mapping (MLM)

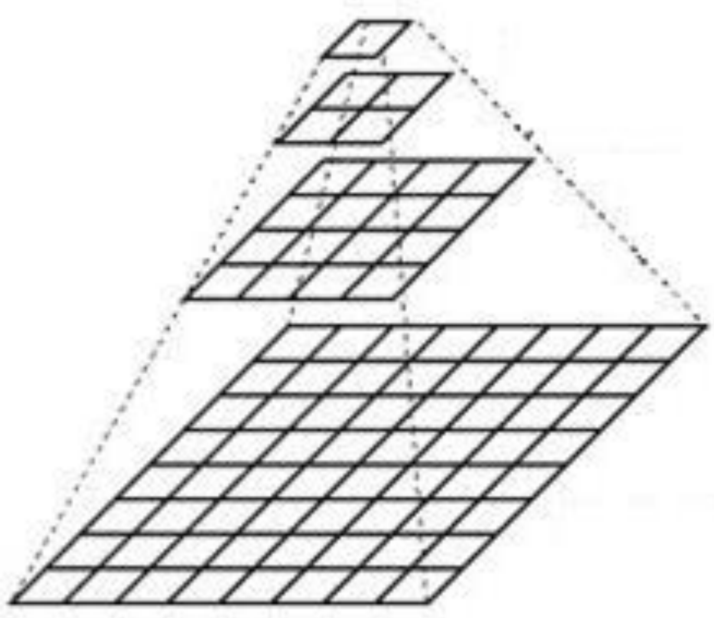
Key Insights

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.
2. World is locally ***smooth***.

Approach

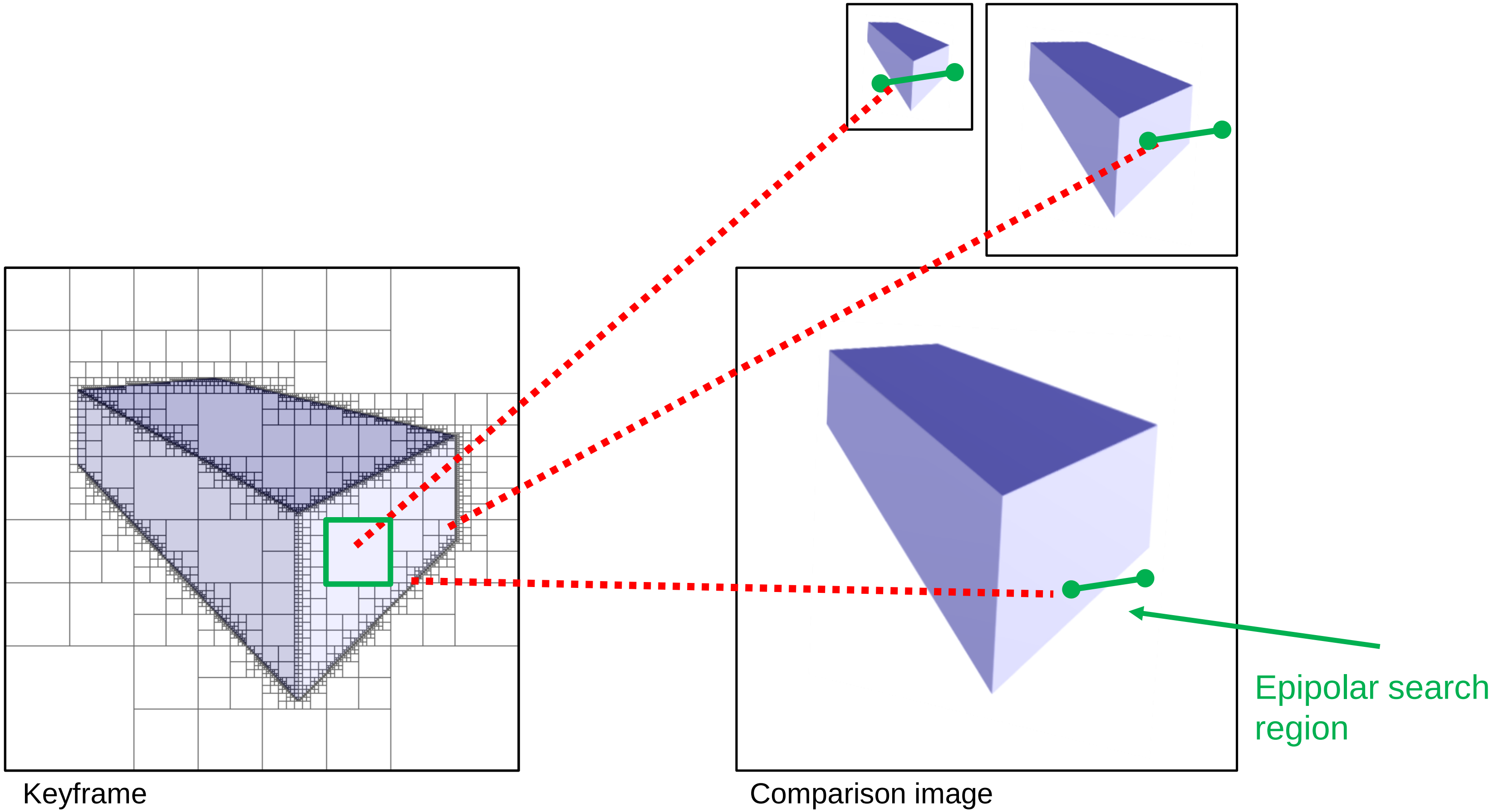
1. Estimate depth at image scale appropriate to texture
   - ***More gradient → finer resolution***
   - ***Less gradient → coarser resolution***

2. Apply smarter ***spatial regularization*** to multi-level map



Estimate depth on a quadtree to leverage all available texture and smooth using a variational regularizer

Example Keyframe: MLM



MLM estimates depth at quadtree leaves at corresponding image scale

Smoothing

The keyframe ***inverse*** depthmap $\mathbf{z}$ is likely corrupted by noise and outliers.

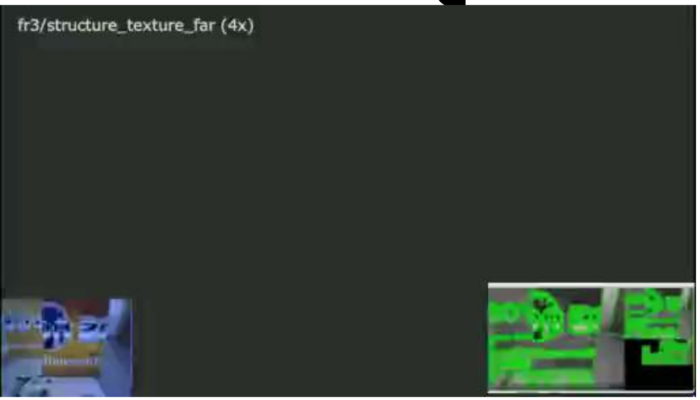
Before finalizing keyframe, apply ***variational smoothing*** technique from Chambolle and Pock 2011.

Let ξ denote the ***smoothed inverse*** depthmap. We perform the following optimization:

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

Quadtree data structure allows fast optimization without GPU acceleration

Qualitative Evaluation



OLD SLIDES

Thank You!