Carnegie Mellon
THE ROBOTICS INSTITUTE

CVPR 2014 Visual SLAM Tutorial Kintinuous

Michael Kaess

kaess@cmu.edu

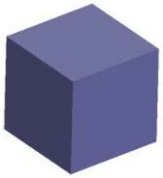
The Robotics Institute

Carnegie Mellon University

Recap: KinectFusion [Newcombe et al., ISMAR 2011]

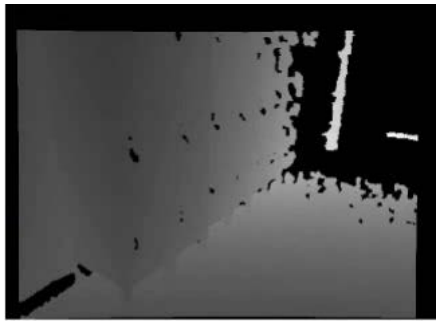
RGB-D camera

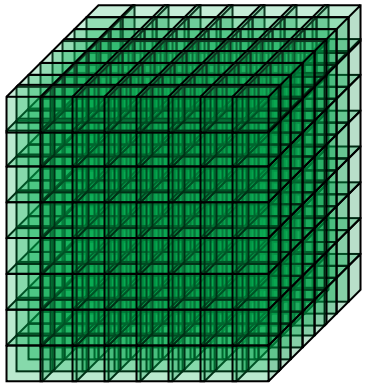
GPU



3D/color model

RGB D



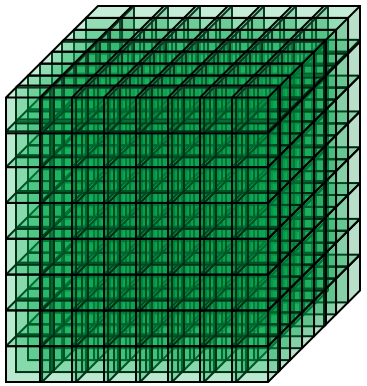


TSDF (volumetric model)

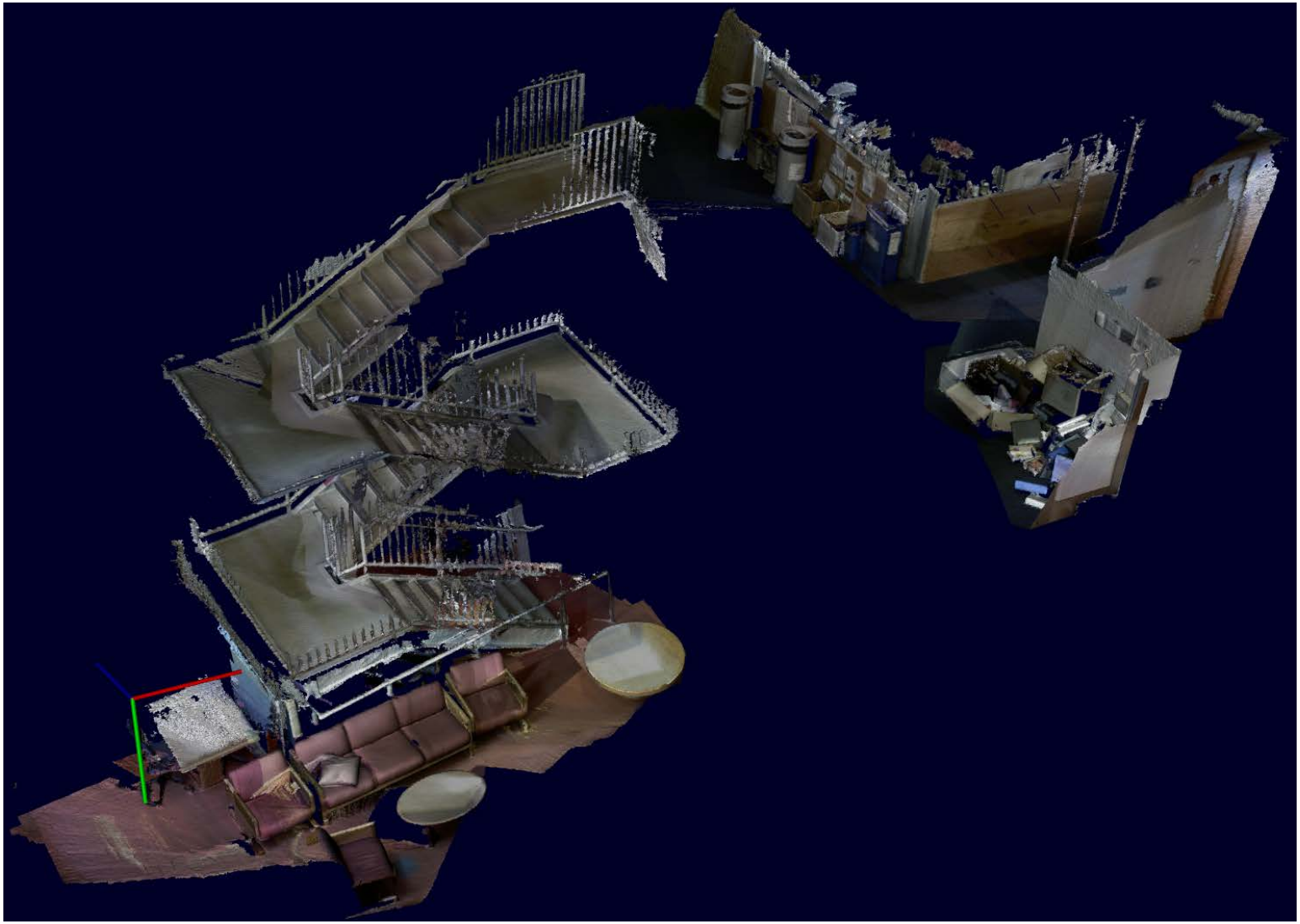
KinectFusion

- Real-time
  - GPU is the enabler
- Unprecedented quality
  - Essentially a moving average

- How much space fits into the volume?
  - Depends on the resolution you want:
    - On 2GB GPU: 512x512x512 voxels
    - At 5mm/voxel: 2.5m side length



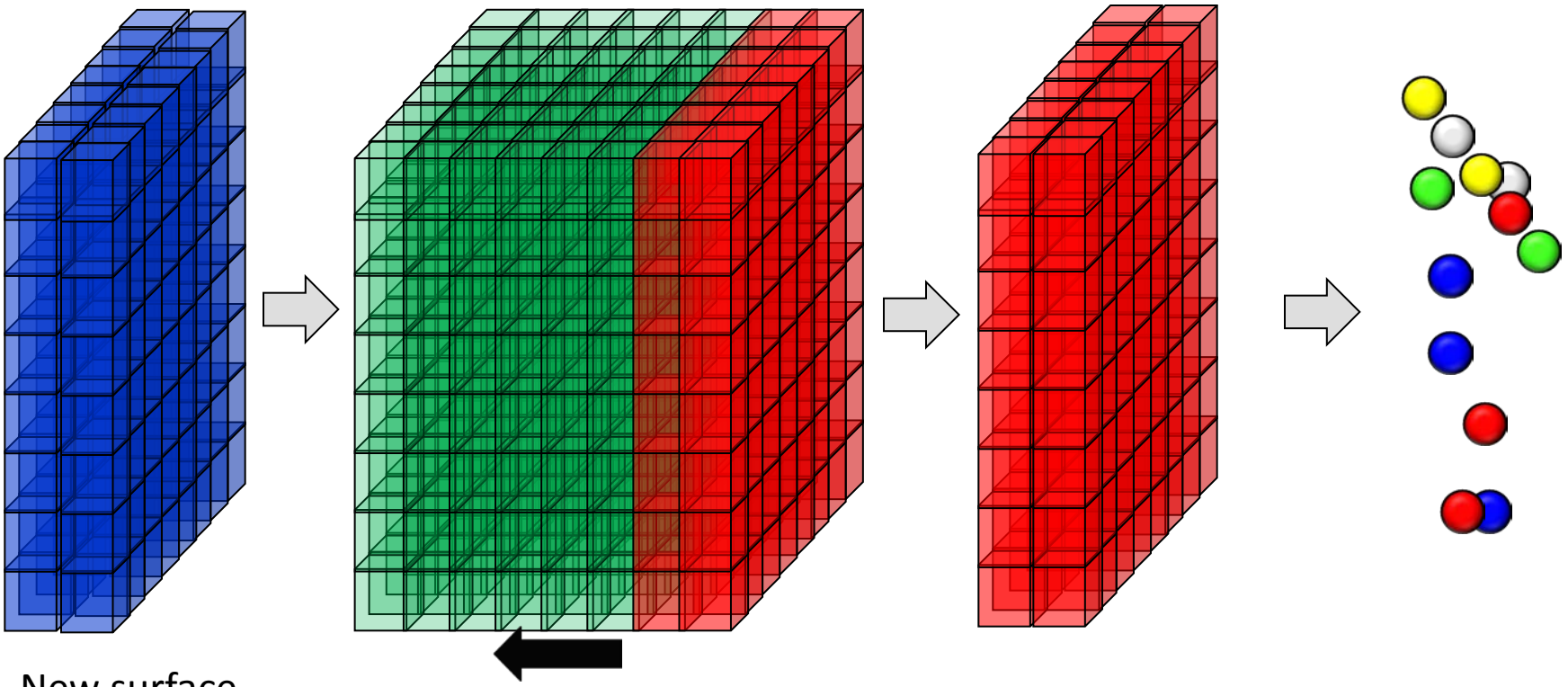
How to map large spaces?



[Whelan et al., RSS12 RGB-D workshop]

Large Spaces: Move the TSDF!

Treat volume as a circular buffer



4. New surface enters volume

1. Camera motion

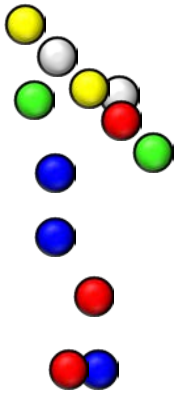
2. Raycast

3. Extracted point cloud

[Whelan et al., RSS12 RGB-D workshop]

Kintinuous 1.0

- The result is a set of point cloud "slices" of the TSDF volume



Extracted point cloud

Marton *et al.* ICRA09

Greedy mesh triangulation

Dense triangular mesh

[Whelan, Johannsson, Kaess, Leonard, McDonald, ICRA 13]

Adding Color

- Second TSDF used to stored RGB color components, only used for integration, not registration.
  - Usually has colour "bleeding" artefacts around edges of objects.
    - Coincides with depth discontinuities and poor angles of incidence
- Artefacts remedied by
  - Rejecting colour measurements on boundaries in the depth image.
    - 7x7 window is inspected around each pixel.
  - Weighting the integration by the angle of incidence.

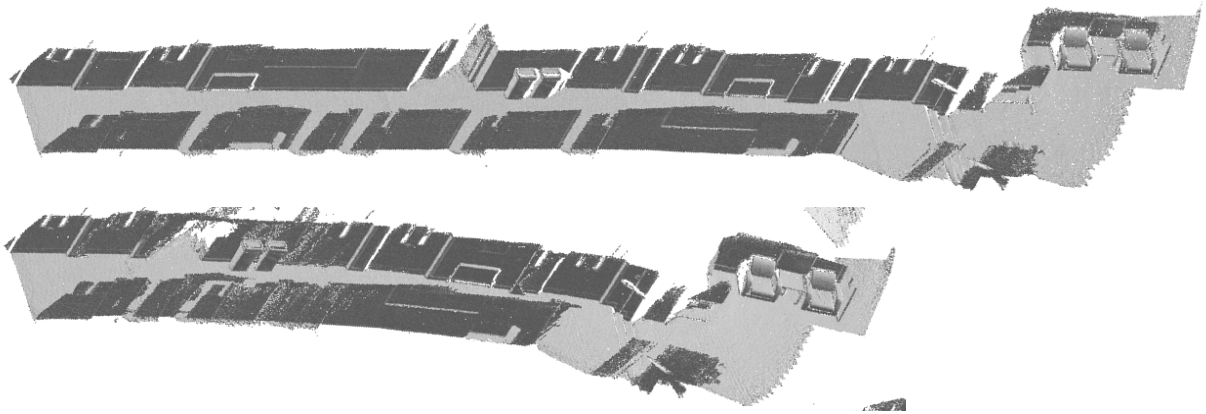
$$cos(\theta) = [0 \;\; 0 \;\; 1] \cdot \mathbf{n} = n_z$$

[Whelan, Johannsson, Kaess, Leonard, McDonald, ICRA 13]

Kintinuous: Stairs in MIT Stata Center

Are we done?

- Mapping relies on sufficient geometry
  - Fails in hallways or large open space



  - ICP fails if
    - Depth lacks strong geometric features
    - No geometric features fall within the TSDF
  - Can use color instead, but integration is poor if
    - RGB/Depth registration is inaccurate
    - Angle of incidence is extreme

[Whelan, Johannsson, Kaess, Leonard, McDonald, ICRA 13]

Kintinuous 1.5

- Merging of geometric and photometric data
  - Combines photometric warping function based on intensity correspondences of Steinbrueker et al. with ICP
  - Computationally cheap
- GPU-ification of Steinbruecker et al.
  - Products computed with same tree reduction implementation of Gauss-Newton as ICP in KinectFusion
    - Really fast
    - Cholesky done on CPU, cheap 6x6 solve

"Real-Time Visual Odometry from Dense RGB-D Images" by F. Steinbruecker, J. Sturm, D. Cremers, Workshop on Live Dense Reconstruction with Moving Cameras at the Intl. Conf. on Computer Vision (ICCV), 2011

[Whelan, Johannsson, Kaess, Leonard, McDonald, ICRA 13]

Kintinuous 1.5

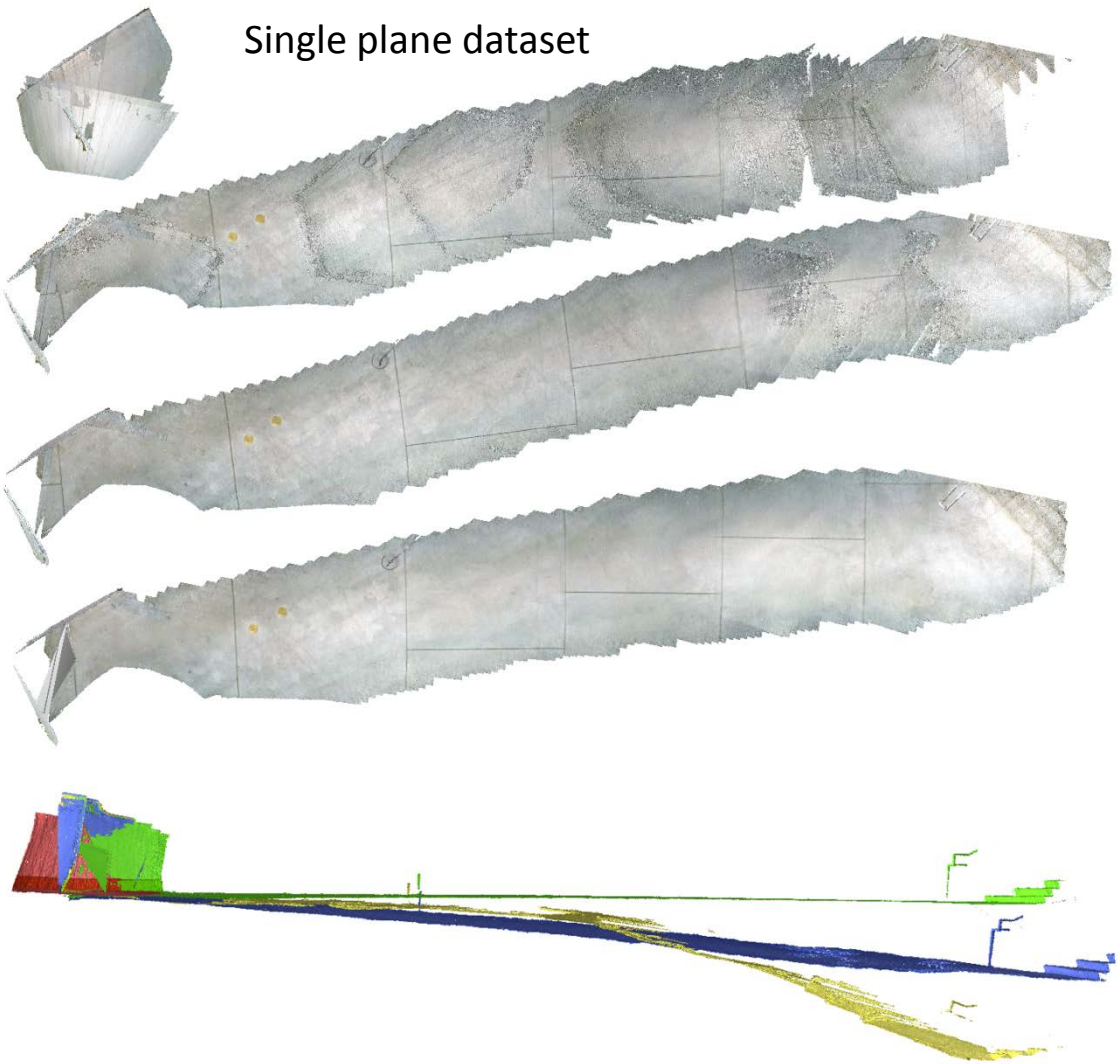
Varied lighting corridor dataset

- FOVIS
- ICP
- RGB-D
- ICP+RGB-D

[Whelan, Johannsson, Kaess, Leonard, McDonald, ICRA 13]

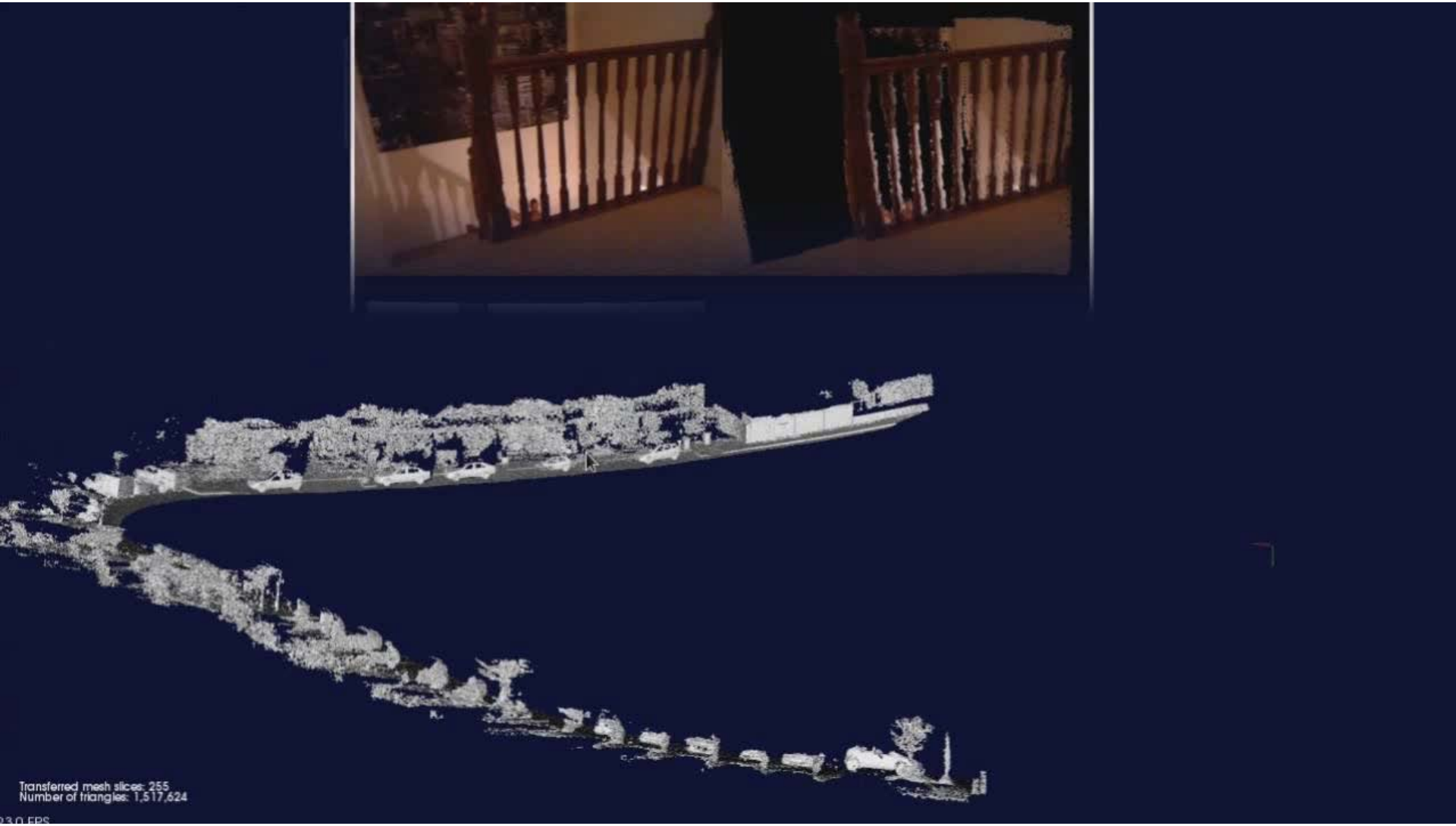
Kintinuous 1.5

- ICP
- RGB-D
- FOVIS
- ICP+RGB-D
- Side view



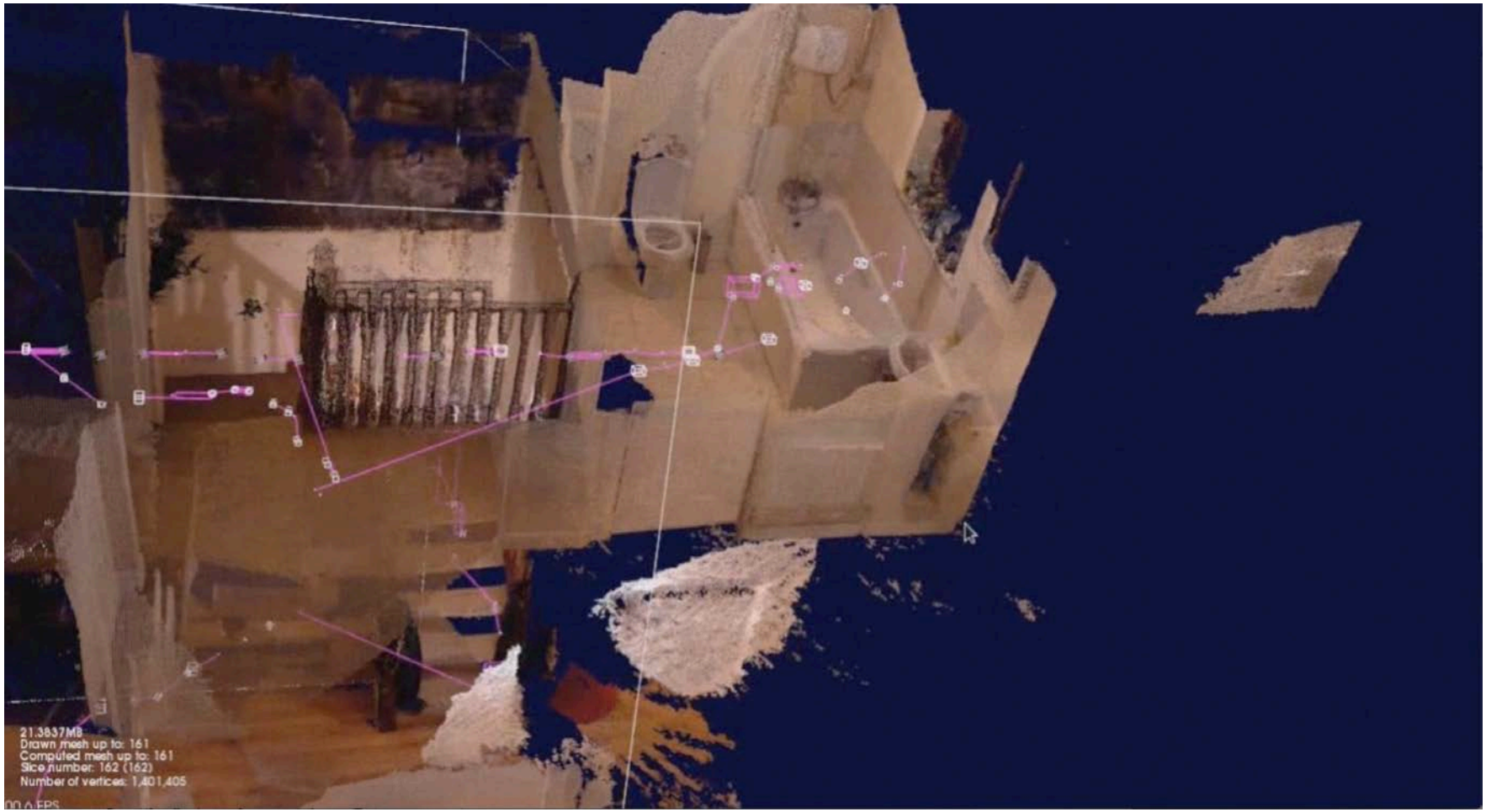
[Whelan, Johannsson, Kaess, Leonard, McDonald, ICRA 13]

Kintinuous: Reconstruction of an Apartment



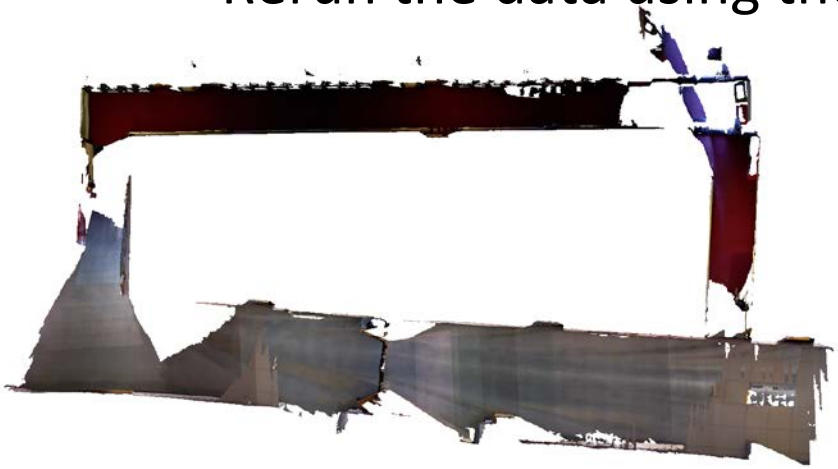
[Whelan, Johannsson, Kaess, Leonard, McDonald, ICRA 13]

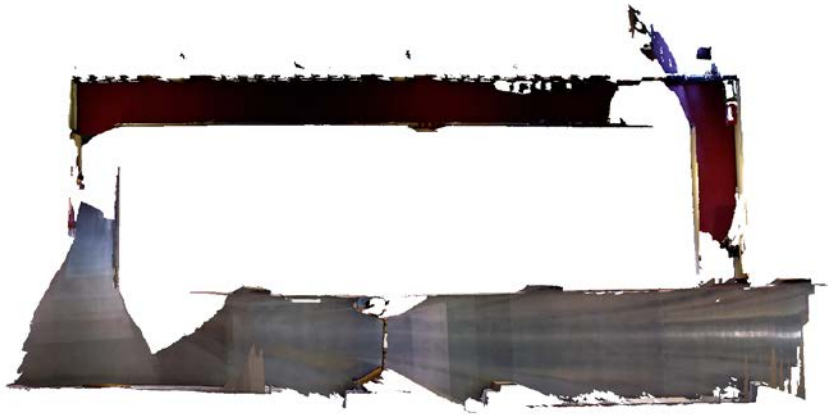
Kintinuous: Reconstruction of an Apartment



Eliminate Drift

- Easy in 2 passes:
  - Build an every-frame pose graph
  - Detect visual loop closures (e.g. DBoW)
  - Optimise the camera pose for each frame (iSAM)
  - Rerun the data using the optimised trajectory

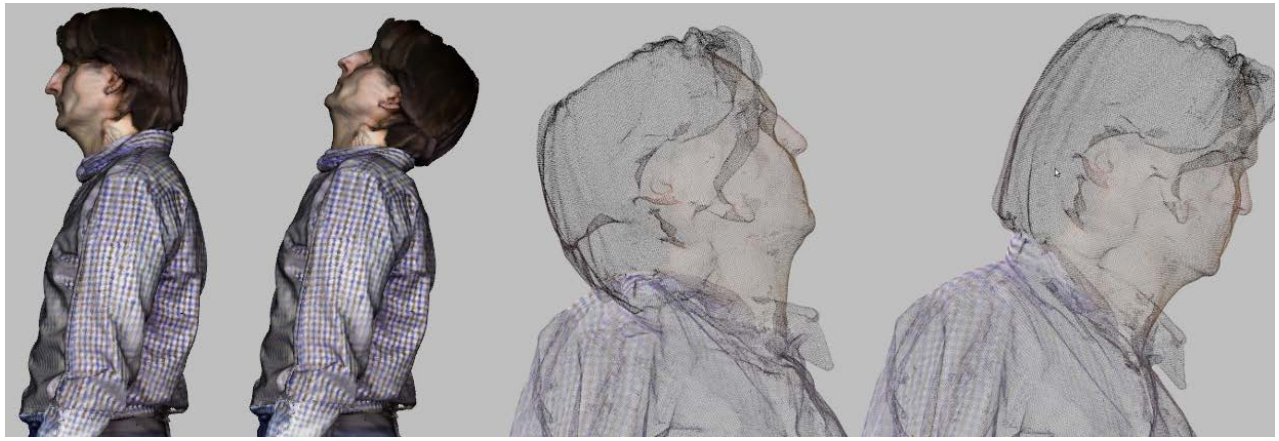




- But how to do this online?

Embedded Deformation

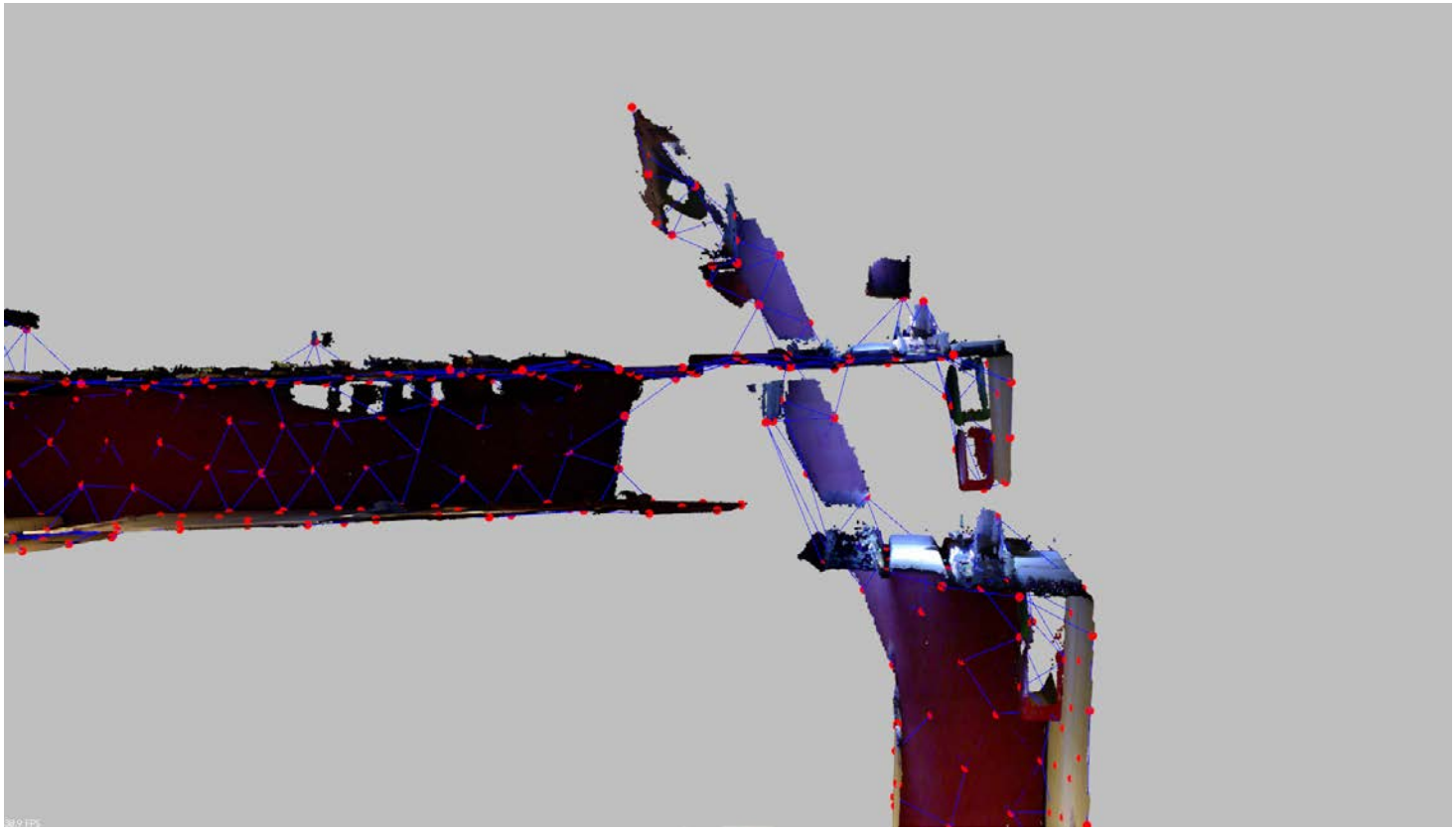
- "Embedded Deformation for Shape Manipulation" by Robert W. Sumner, Johannes Schmid, Mark Pauly in SIGGRAPH 2007
- Meant for real-time manipulation of 3D models
- Parameterised by user specified control points
- Cannot be directly applied to SLAM



[Whelan, Kaess, Leonard, McDonald, IROS 13]

Embedded Deformation in SLAM: Sampling

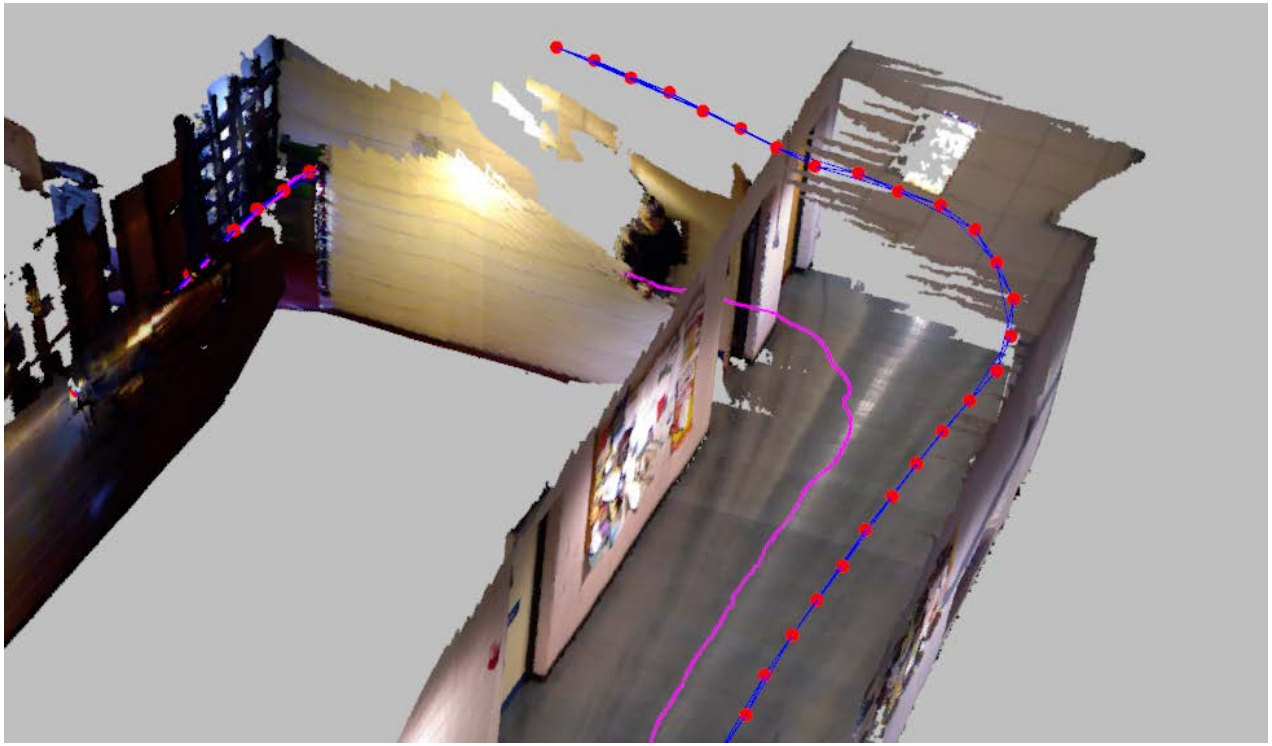
- Nearest neighbour graph sampling can connect unrelated areas of the map



[Whelan, Kaess, Leonard, McDonald, IROS 13]

Embedded Deformation in SLAM

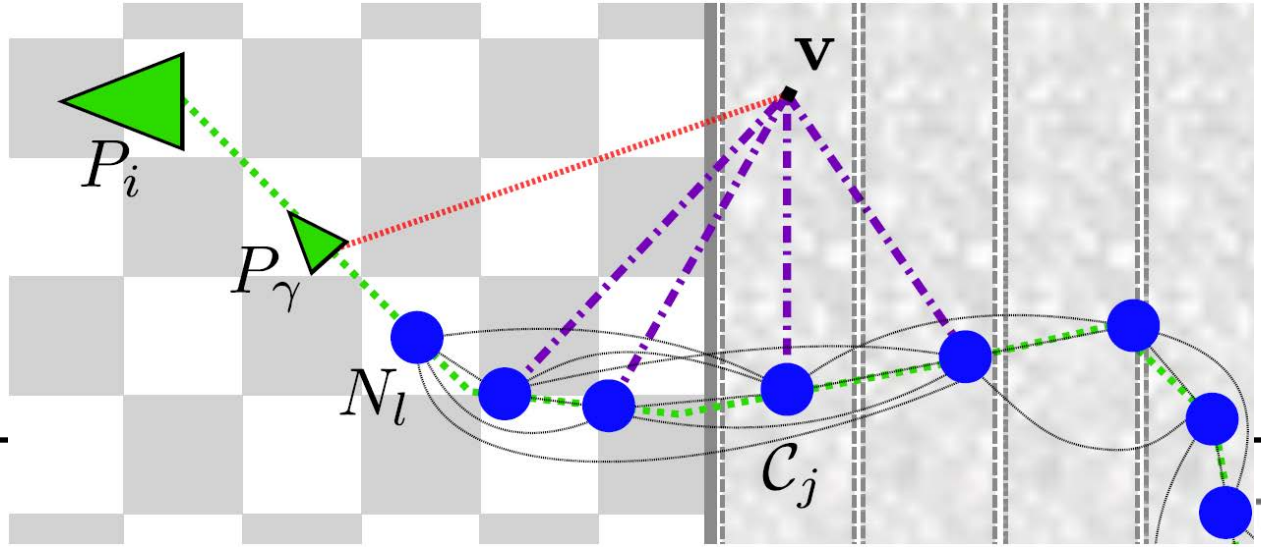
- Instead of sampling, use the pose graph as deformation graph
- Instead of user input, use the optimized pose graph



[Whelan, Kaess, Leonard, McDonald, IROS 13]

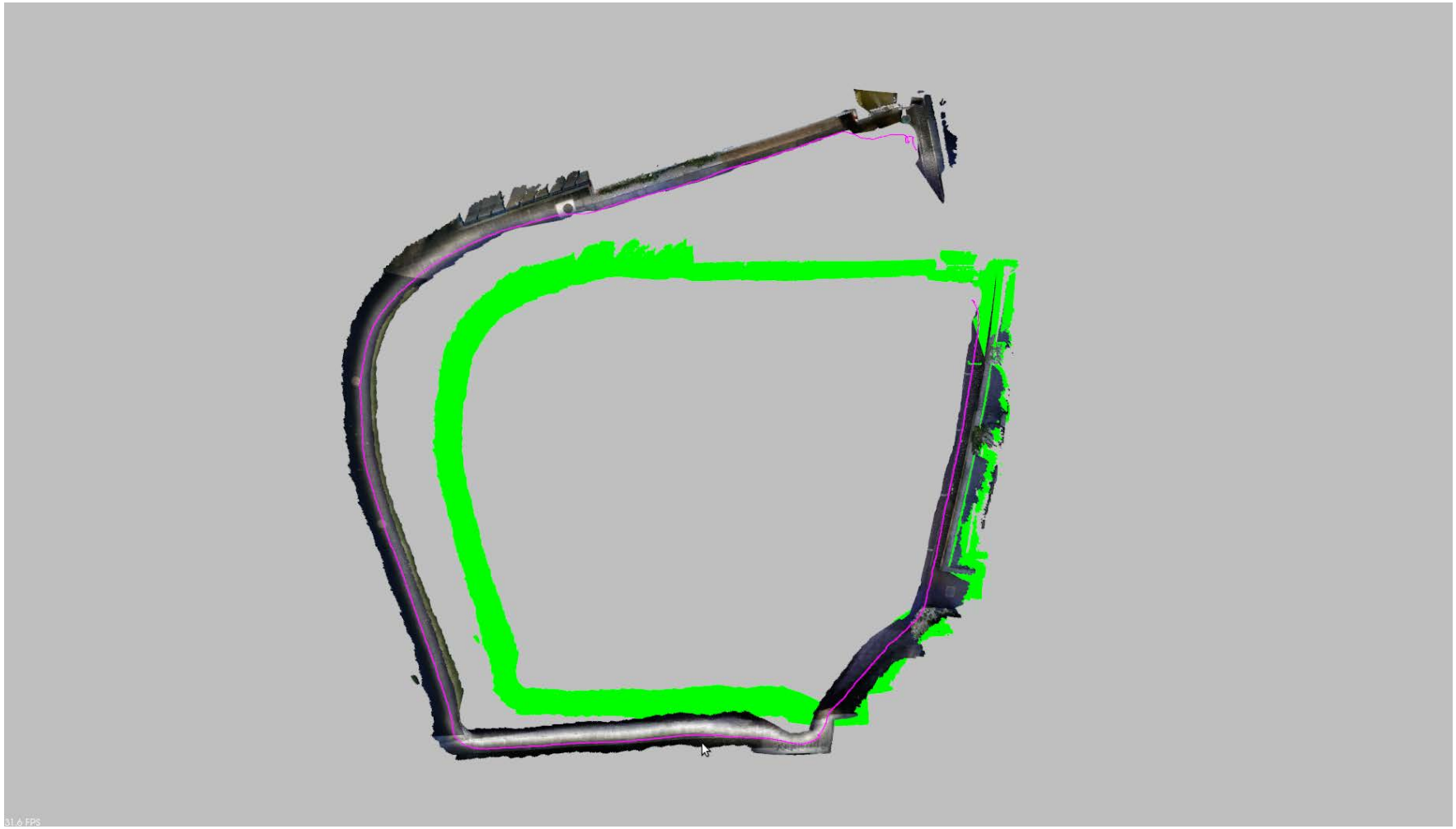
Embedded Deformation in SLAM: Sampling

- What about vertices?
  - If we just take the k-nearest nodes to each vertex, we may select nodes which came from much older poses in the graph.
  - We can restrict the set of nearest nodes to each vertex according to the pose which created a given vertex
    - Kintinuous incrementally produces “cloud slices” from the TSDF, each with an associated camera pose.



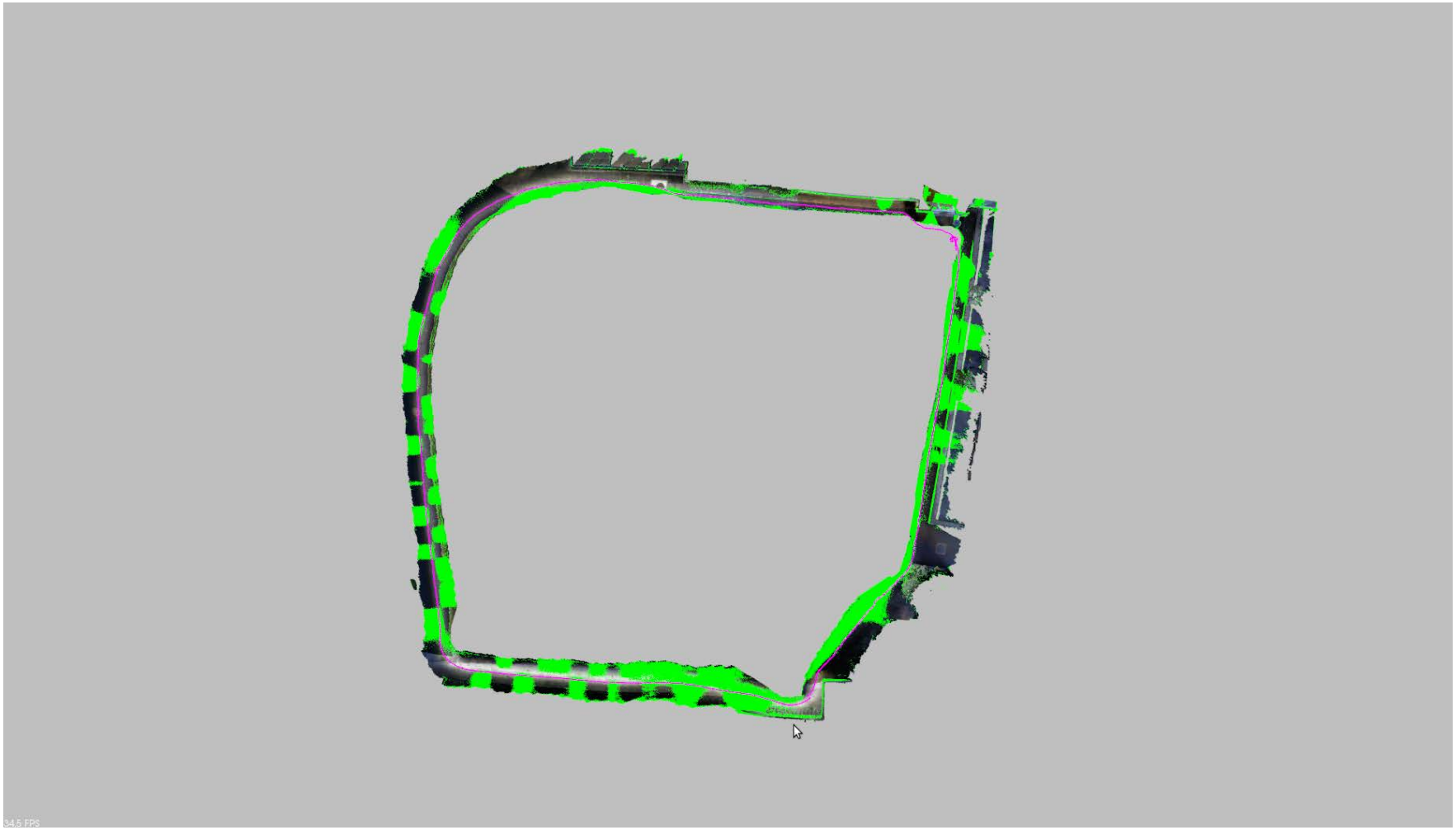
[Whelan, Kaess, Leonard, McDonald, IROS 13]

Results: 1-pass vs. 2-pass



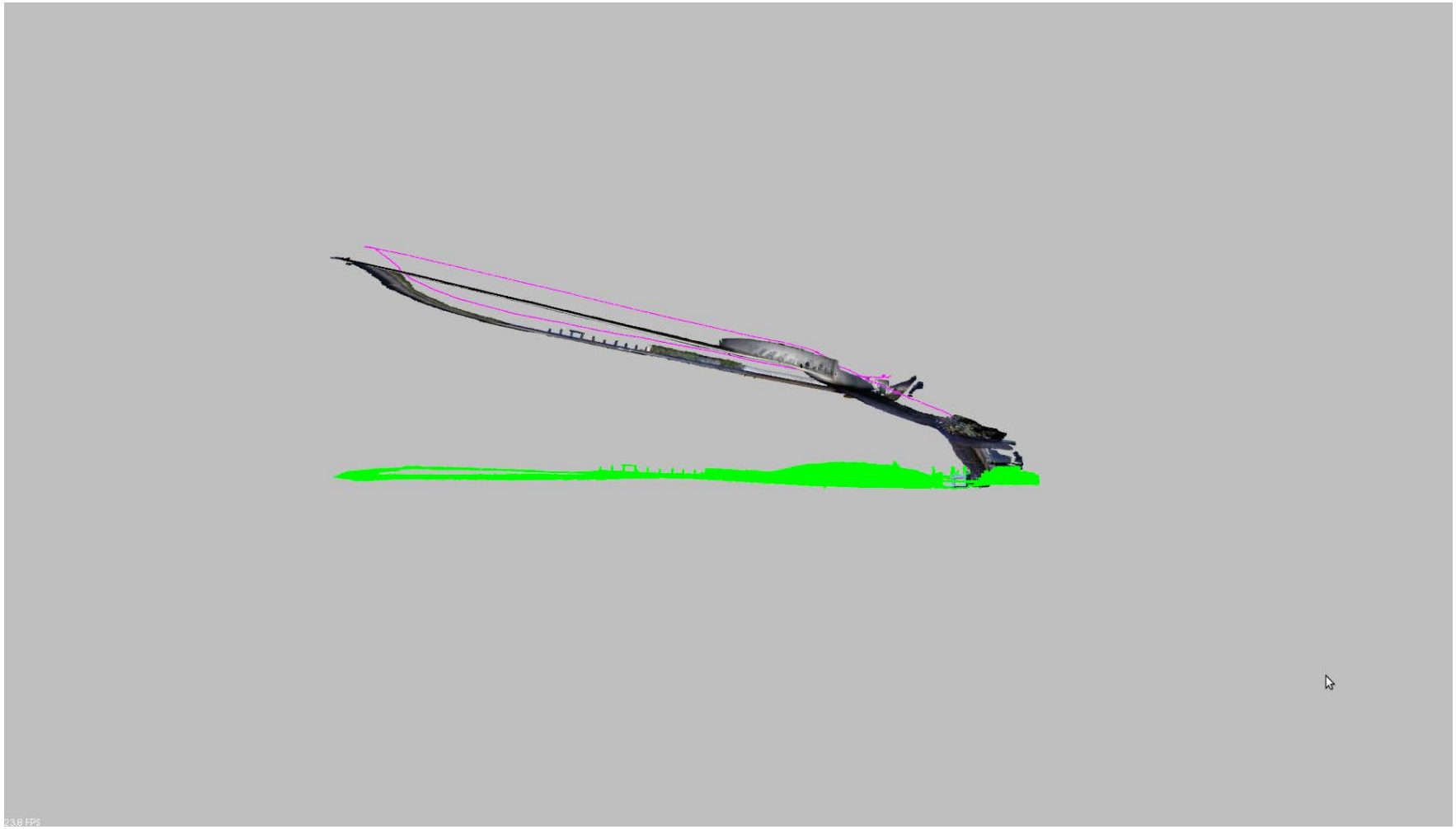
[Whelan, Kaess, Leonard, McDonald, IROS 13]

Results: 1-pass vs. 2-pass



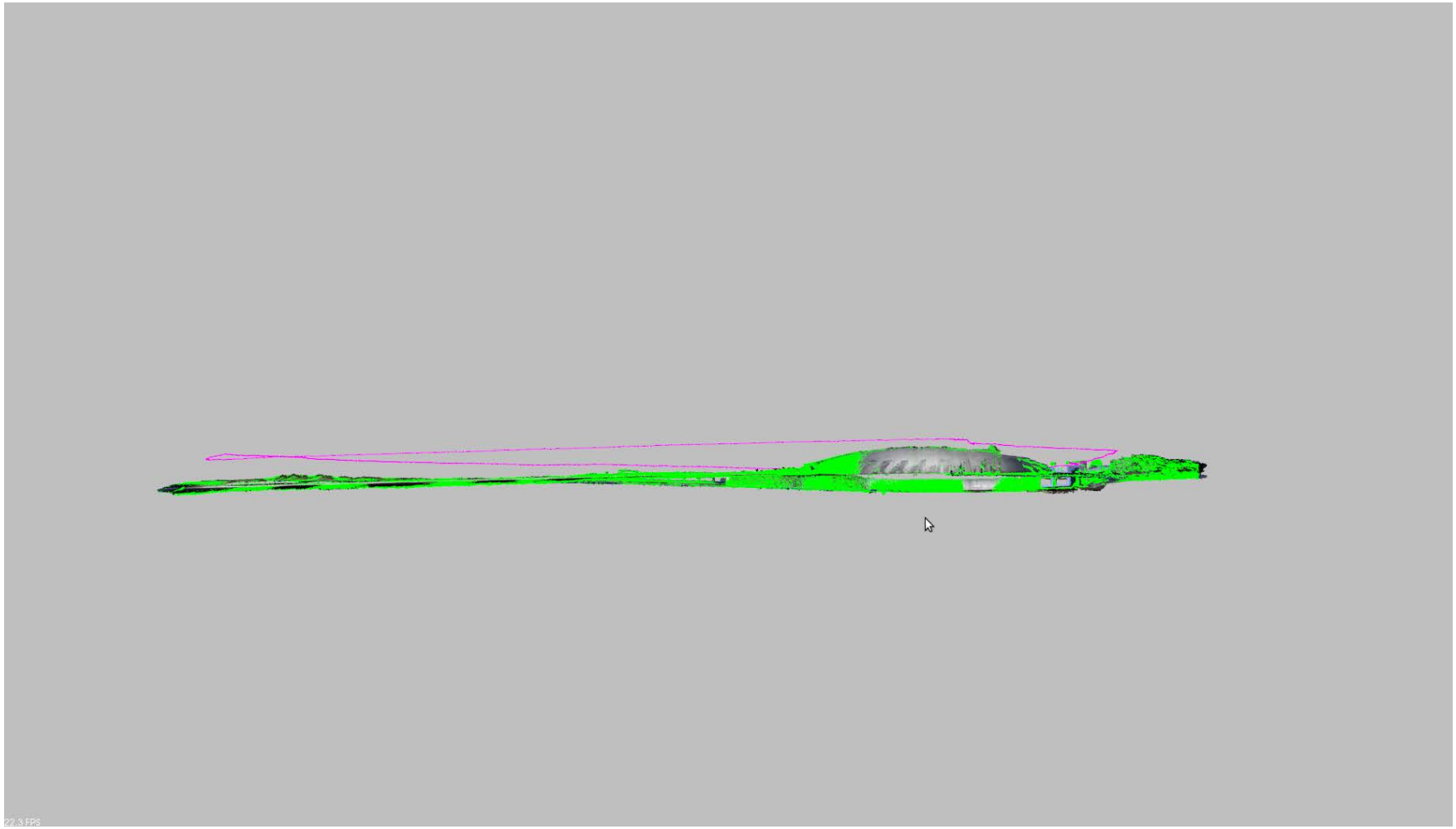
[Whelan, Kaess, Leonard, McDonald, IROS 13]

Results: 1-pass vs. 2-pass



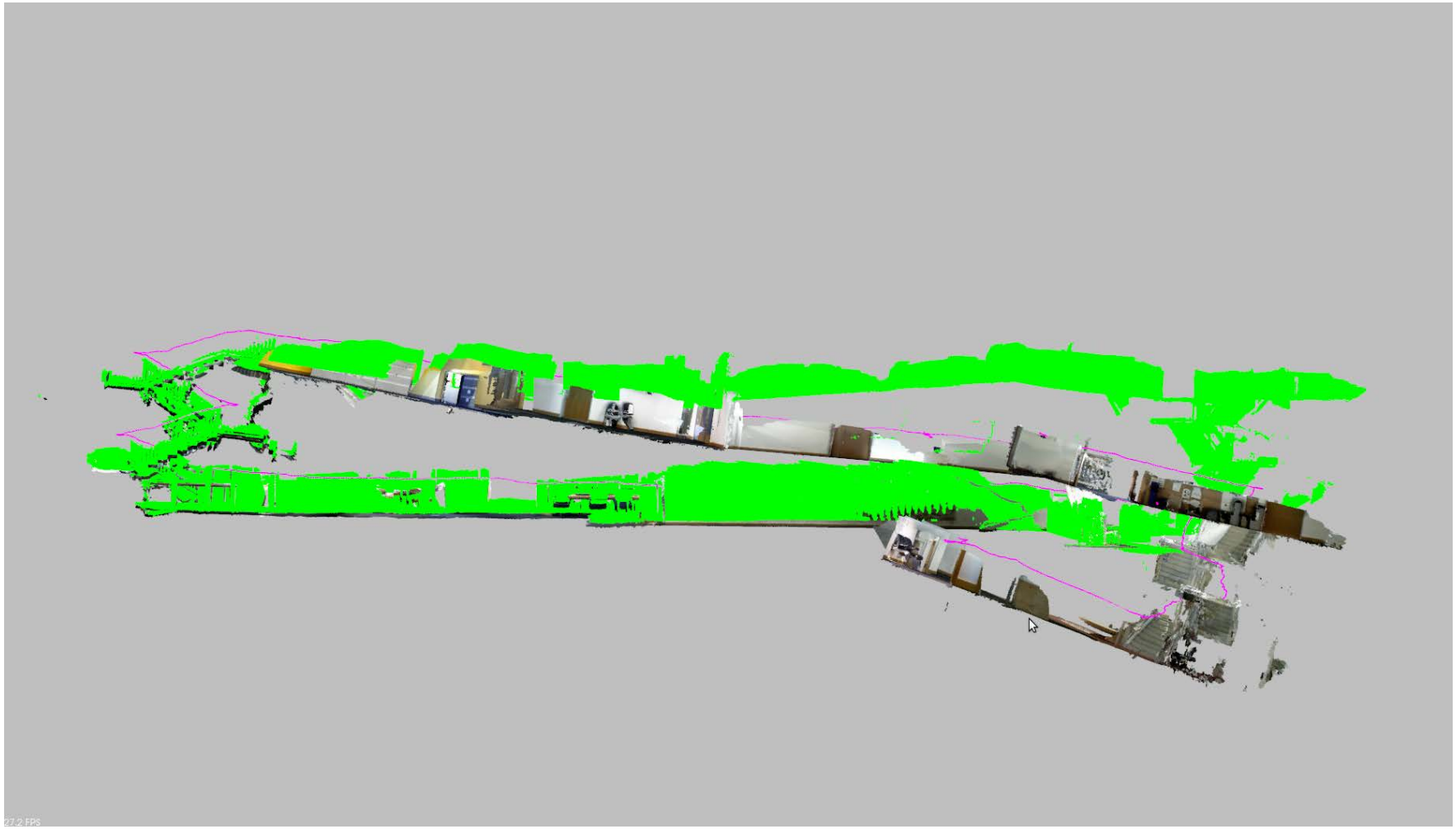
[Whelan, Kaess, Leonard, McDonald, IROS 13]

Results: 1-pass vs. 2-pass



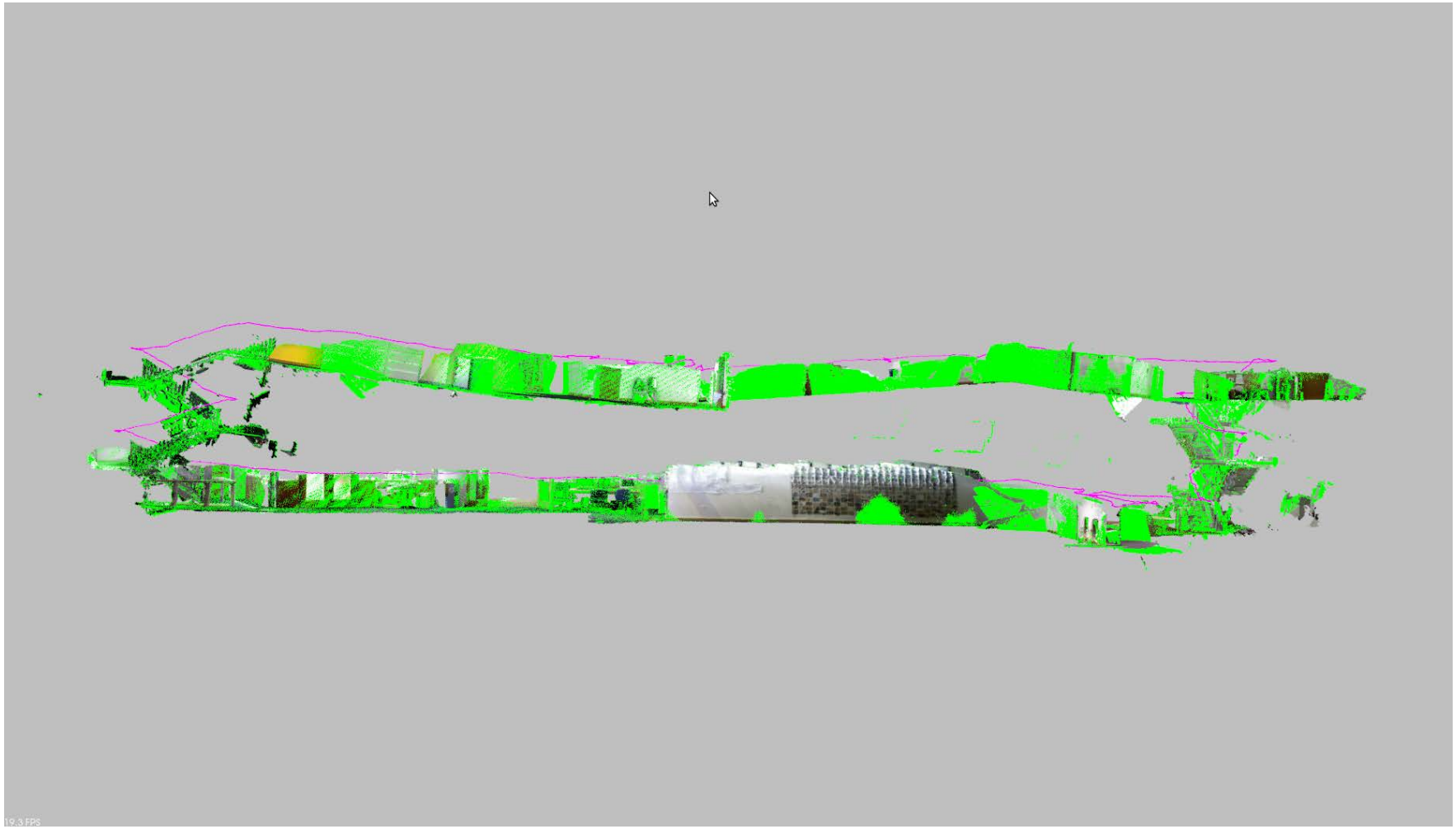
[Whelan, Kaess, Leonard, McDonald, IROS 13]

Results: 1-pass vs. 2-pass



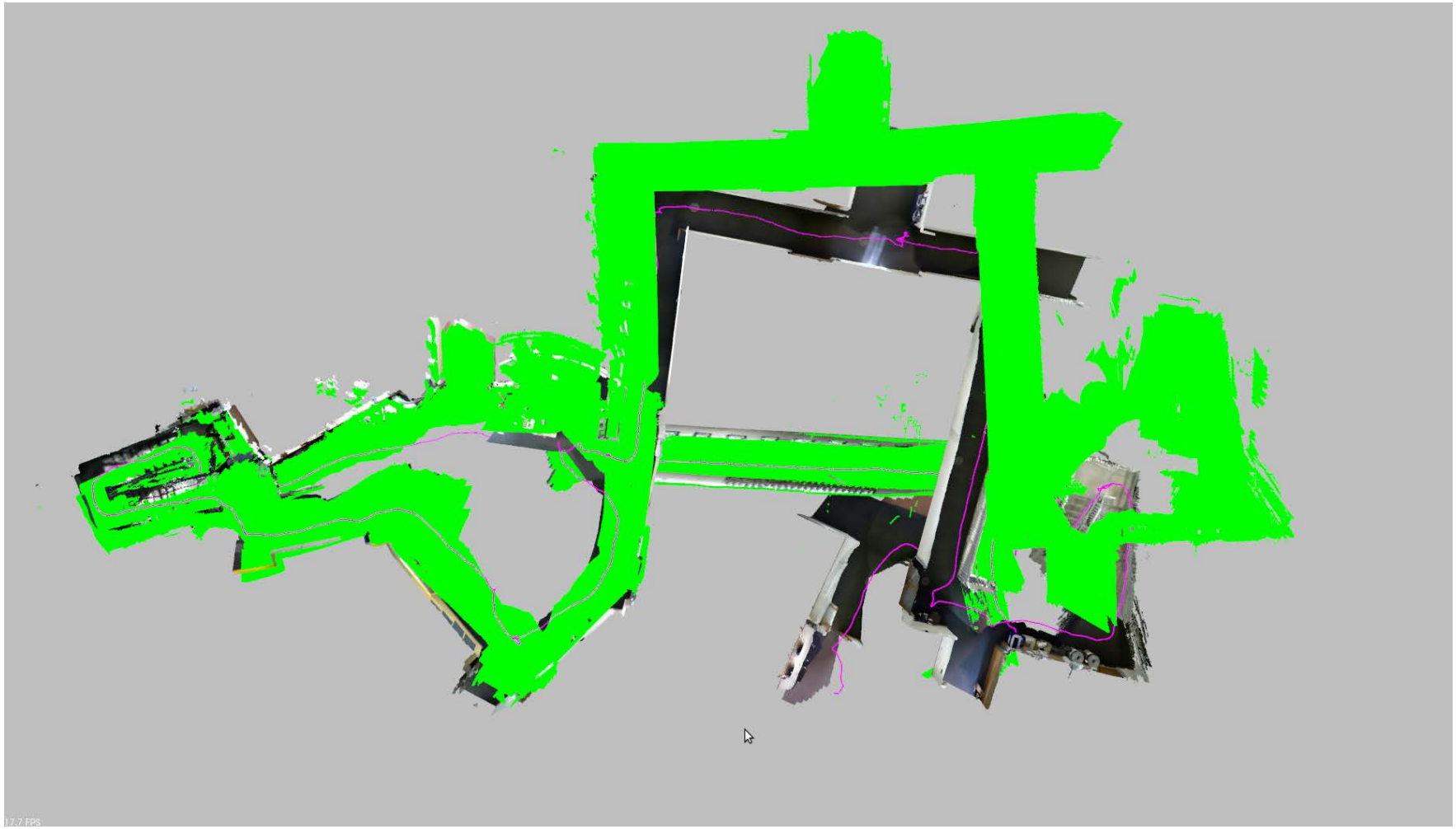
[Whelan, Kaess, Leonard, McDonald, IROS 13]

Results: 1-pass vs. 2-pass



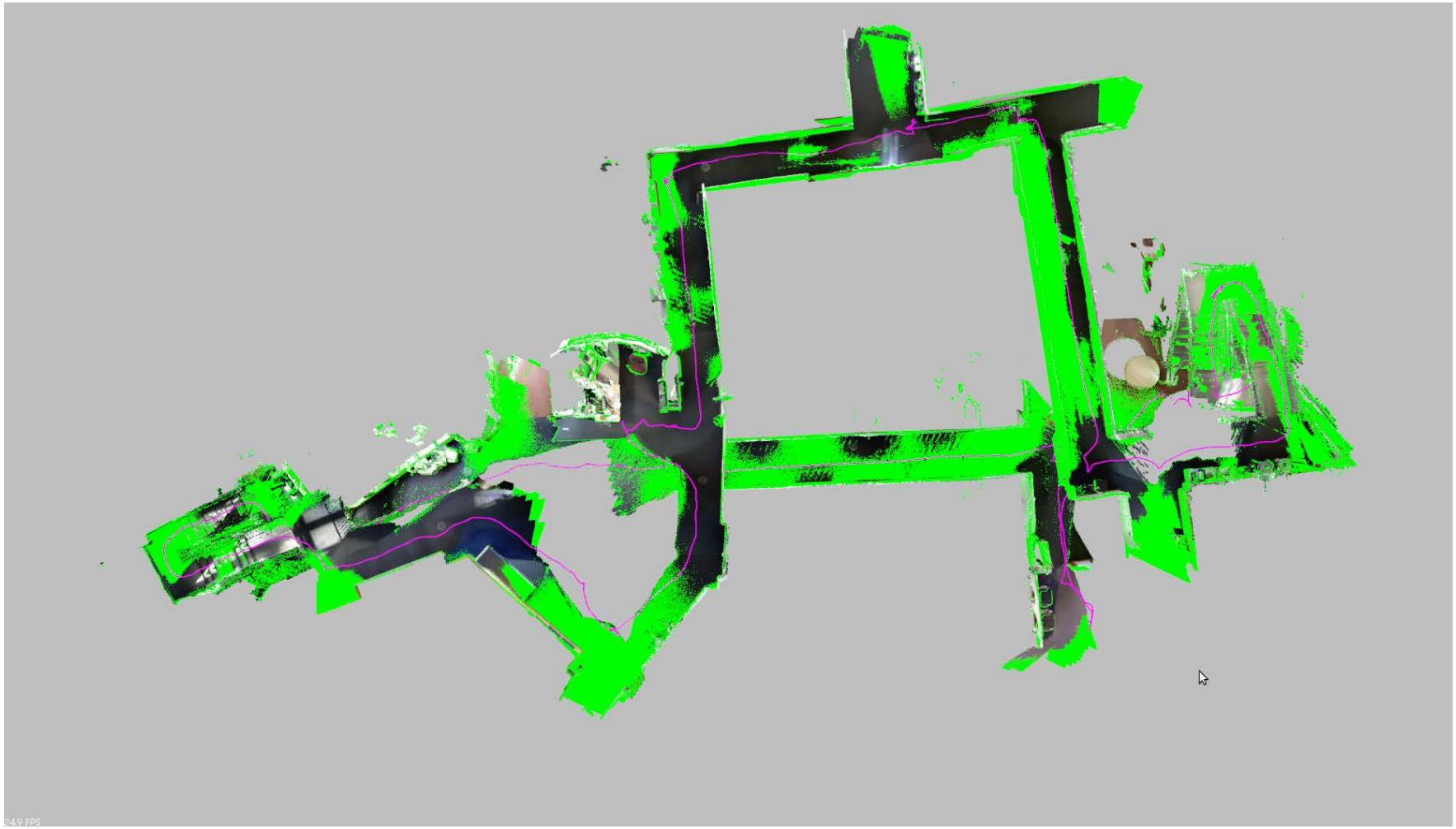
[Whelan, Kaess, Leonard, McDonald, IROS 13]

Results: 1-pass vs. 2-pass



[Whelan, Kaess, Leonard, McDonald, IROS 13]

Results: 1-pass vs. 2-pass



[Whelan, Kaess, Leonard, McDonald, IROS 13]

Results: Timing

- We measure computational performance in terms of latency

Quantities	Datasets			
	Indoors	Outdoors	Two floors	In/outdoors
DBoW images	311	1181	1603	2625
Poses	313	1182	1605	2627
Nodes	89	168	176	338
Vertices	1,598,253	3,190,669	4,296,449	6,909,051
Process	Timings (ms)			
Frontend	578	528	663	973
iSAM	50	239	392	986
Deformation	115	377	461	974
Total latency	743	1144	1516	2933

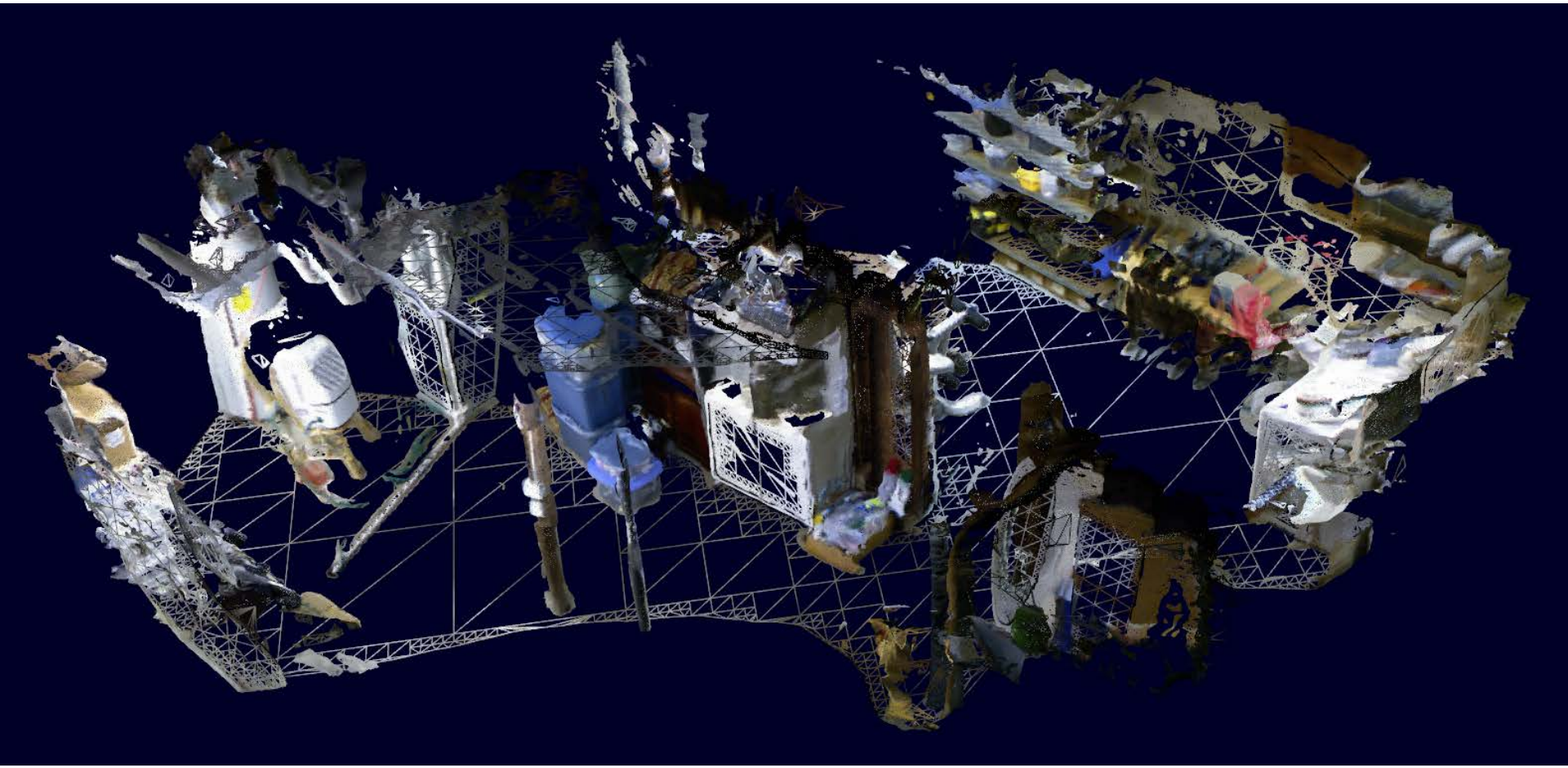
Simplification & Higher level information

- Dense maps are great but
  - Using them to plan is expensive/slow
  - Many parts are over represented

- Let's start with planes
  - Easy to detect given fidelity of Kintinuous output

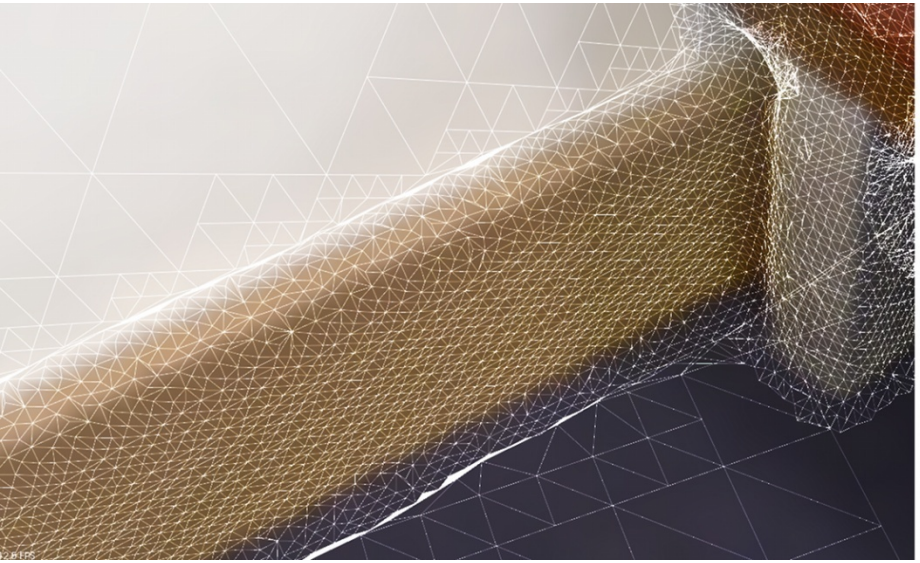
"Planar Simplification and Texturing of Dense Point Cloud Maps" by L. Ma, T. Whelan, E. Bondarev, P. H. N. de With, and J.B. McDonald. In European Conference on Mobile Robotics, ECMR, (Barcelona, Spain), September 2013

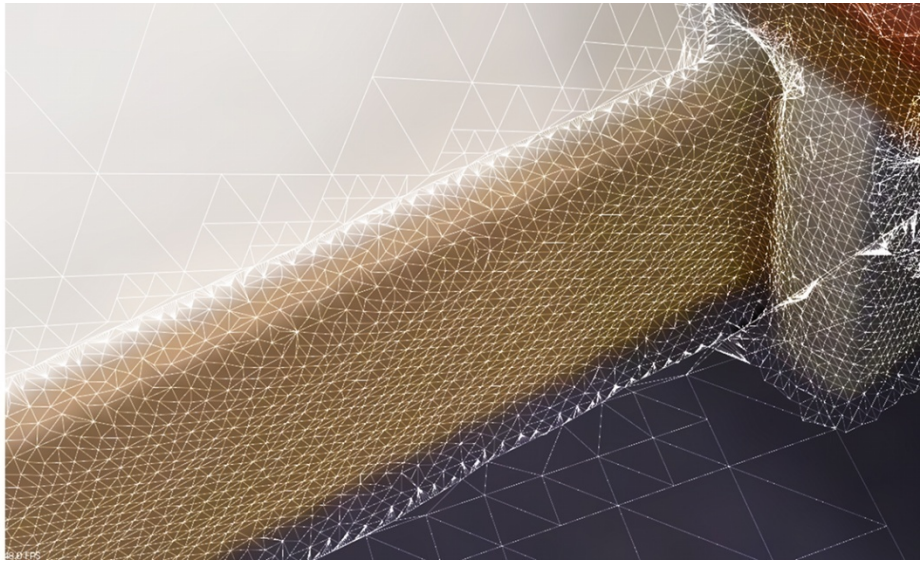
Planar Simplification



Combined planar and dense triangulation

Planar Simplification: Joining dense and planar





Planar Simplification: Object positions