Mesh Adaptation Based on the Depth Image (MABDI)

Lucas Chavez

Dr. Lumia[1] Dr. Fierro[1]
Dr. Chaimowicz[2] Dr. Campos[2] Dr. Anderson[3]

[1]University of New Mexico

[2]Universidade Federal de Minas Gerais

[3]Sandia National Laboratories

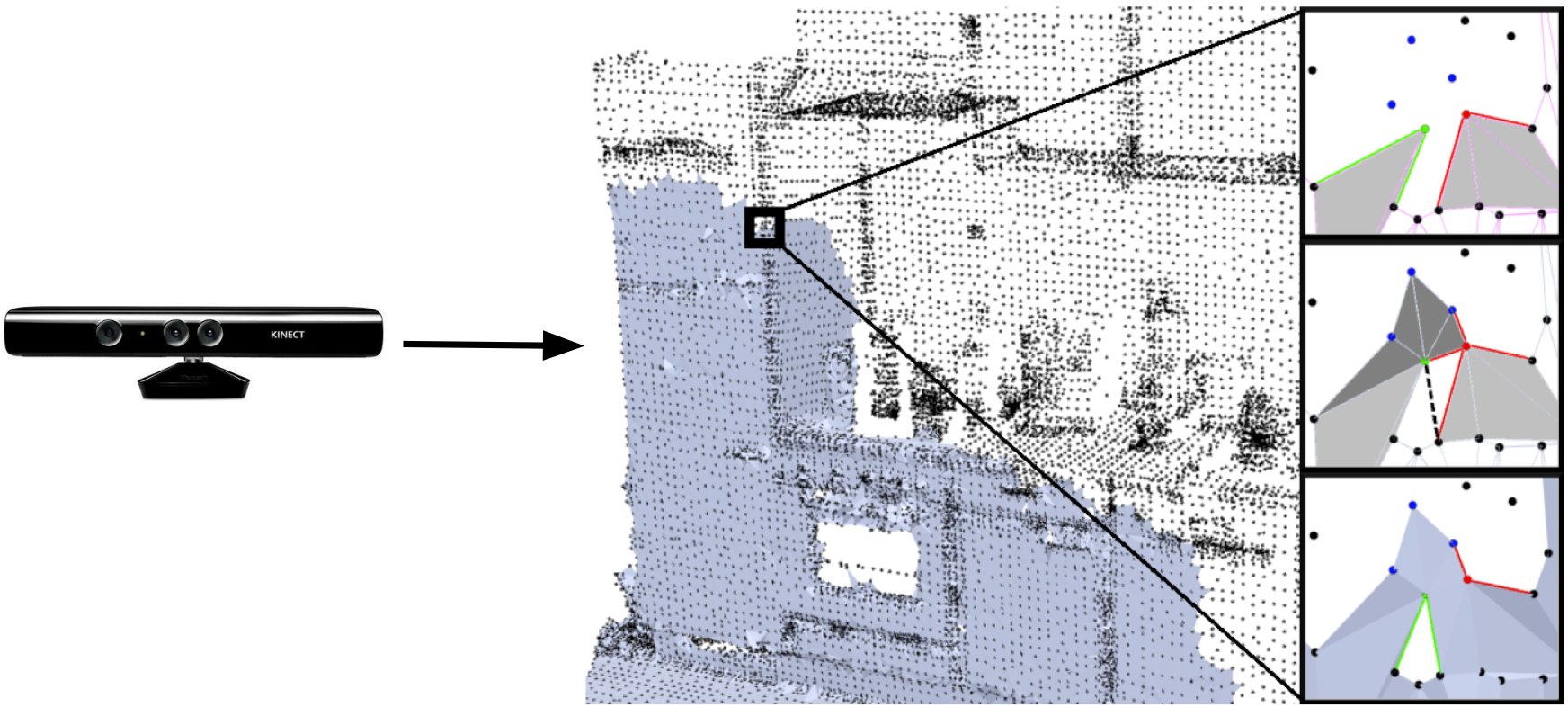
April 2013

Outline

Outline

What is the main objective?

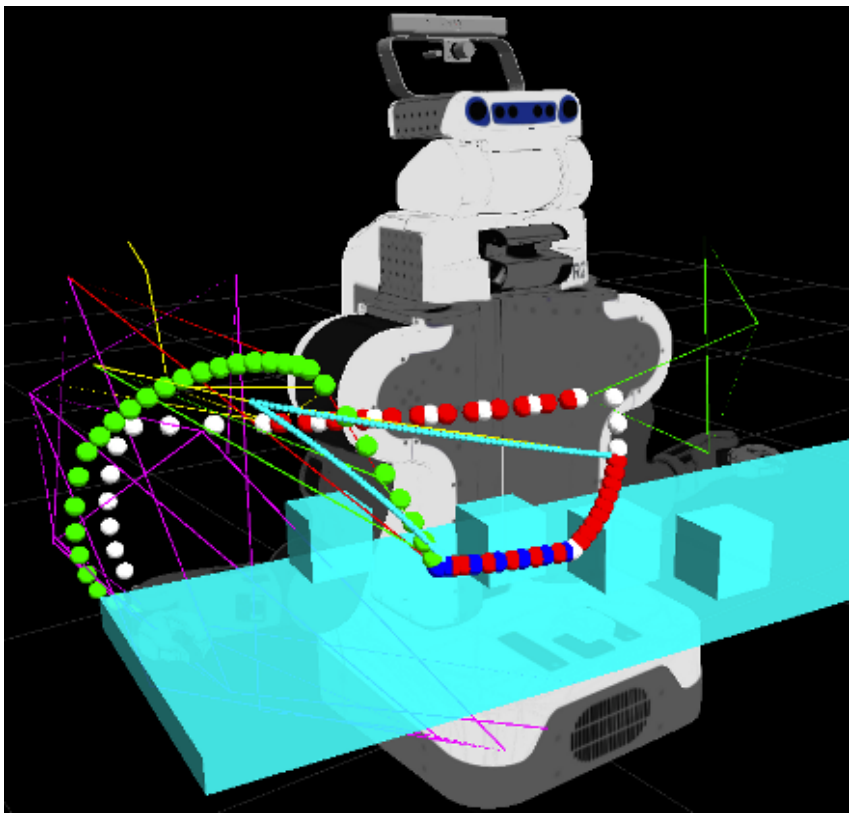
RGB-D sensor to make a useful representation of the environment



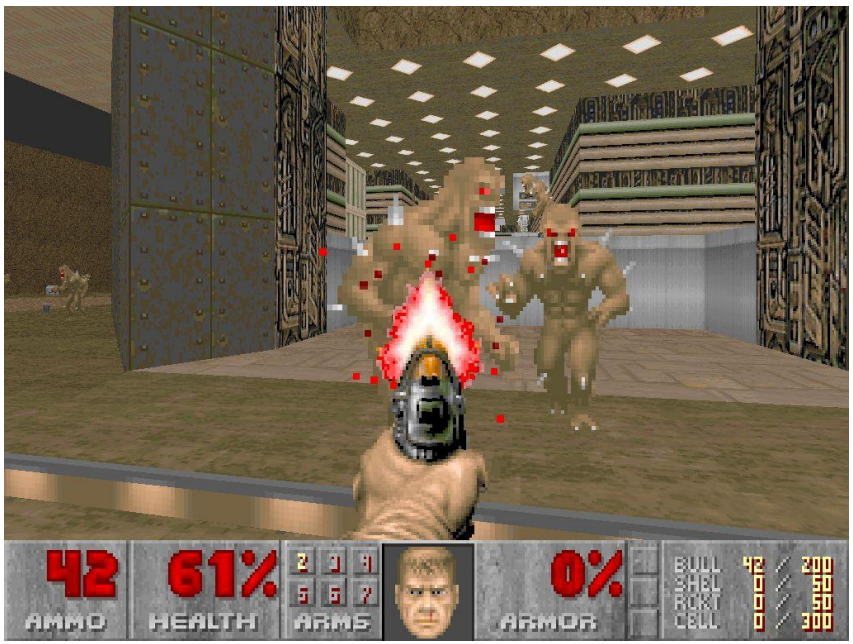
(Marton et al. 2009) On Fast Surface Reconstruction Methods for Large and Noisy Point Clouds

What can representation like this be used for?

Robotic applications



Urban search and rescue (USAR)



(Berenson et. al 2012) A robot path planning framework that learns from experience
(Kadous et. al 2006) Effective user interface design for rescue robotics

Comparing the different representations

	Adaptability	Computationally Inexpensive	Low Memory Requirement	SA: Robot	SA: Human
Landmark Locations	x	x	x	-	-
Point Clouds	-	x	-	-	-
Surfels	x	x	x	-	x
Implicit Functions	x	-	-	x	x
Static Mesh	-	x	x	x	x
Adaptive Mesh	x	o	o	x	x

Desired representation methodology characteristics:

- Adapt existing mesh with new measurements
- Quickly identify new parts of the scene
- React to new or removed objects
- Computationally and memory efficient

Related works

Surface Reconstruction - Computer Graphics & Computational Geometry (Seitz 2006; Mencl 1997)

- Volume-based
- Signed Distance Function (Ohtake 2003)
- Surface-based
- Incremental Surface Growing (Marton 2009)

Simultaneous Localization and Mapping (SLAM)

- Probabilistic Geometry Estimation (Cashier 2012)
- Kintinuous (Whelan et. al 2012)
- KinectFusion (Newcombe et. al 2011)
- RGB-D Mapping (Henry et al. 2010)

Outline

What do I intend to do different?

The main idea is to leverage fast image processing techniques

- Works well with GPU
- Depth image contains lots of information

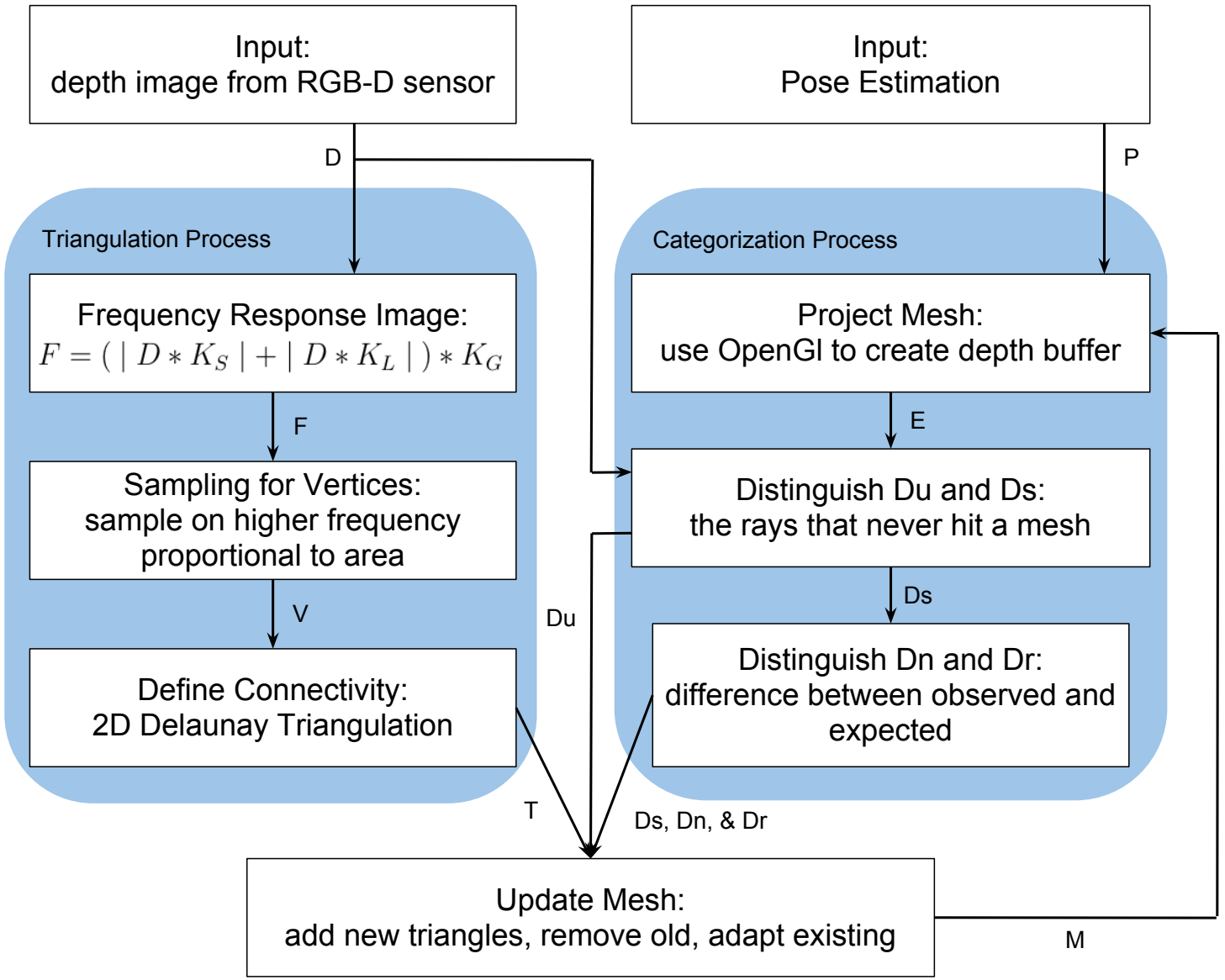
Want to categorize measurements and define connectivity

- Connectivity defined in depth image is preserved
- Measurements need to be categorized
  - D_u - unseen
  - D_s - supporting
    - D_n - new object
    - D_r - removed object

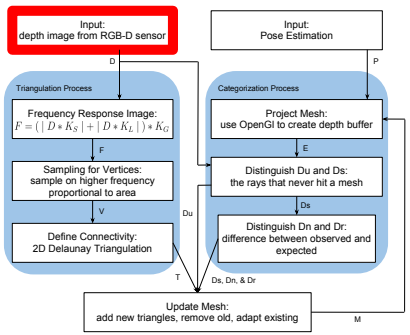
How will it work?

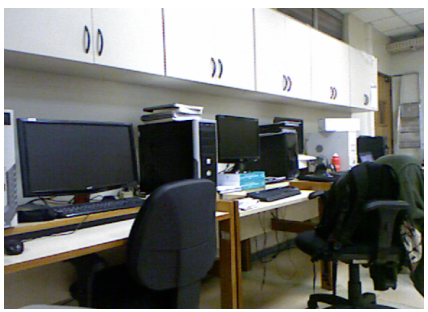
Variable Name	Description
D	Depth image from RGB-D sensor
F	Frequency response image
$K_S, K_L,$ and K_G	Image convolution operators
V and C	Mesh vertices and connectivity of the current triangulation
T	Current triangulation. Contains both V and C
P	The known pose of the sensor
E	Expected depth image
D_u	Previously *unseen* parts of D
D_s	Parts of D which *support* an existing part of M
D_n	Parts of D which correspond to a *new* object
D_r	Parts of D which correspond to a *removed* object
M	Current mesh

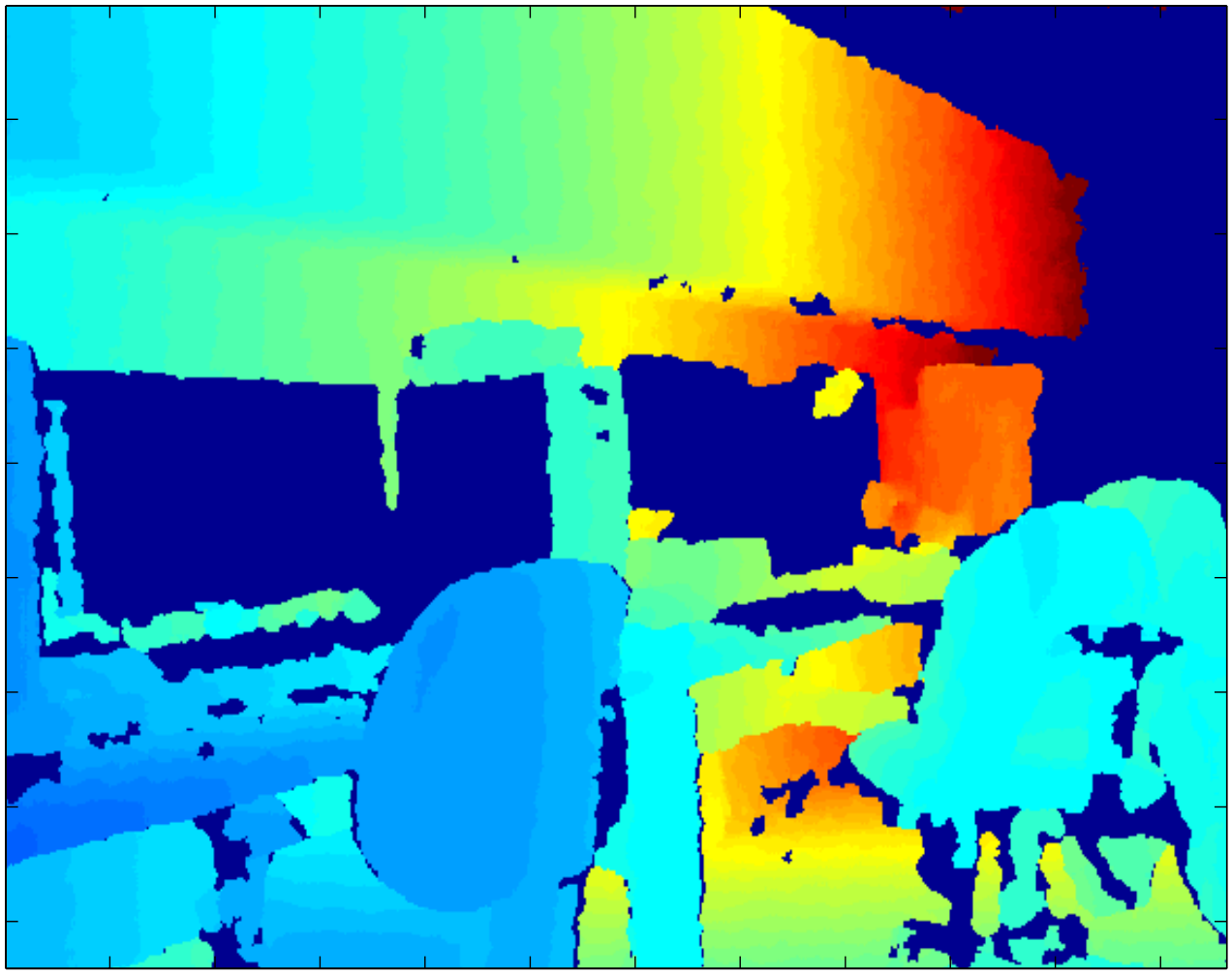
How will it work?



Input from the RGB-D Sensor

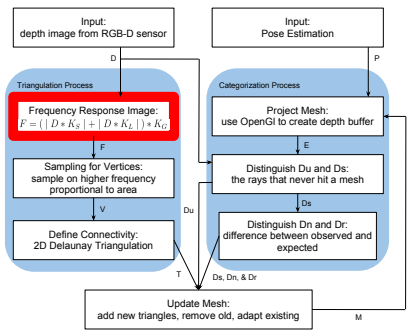


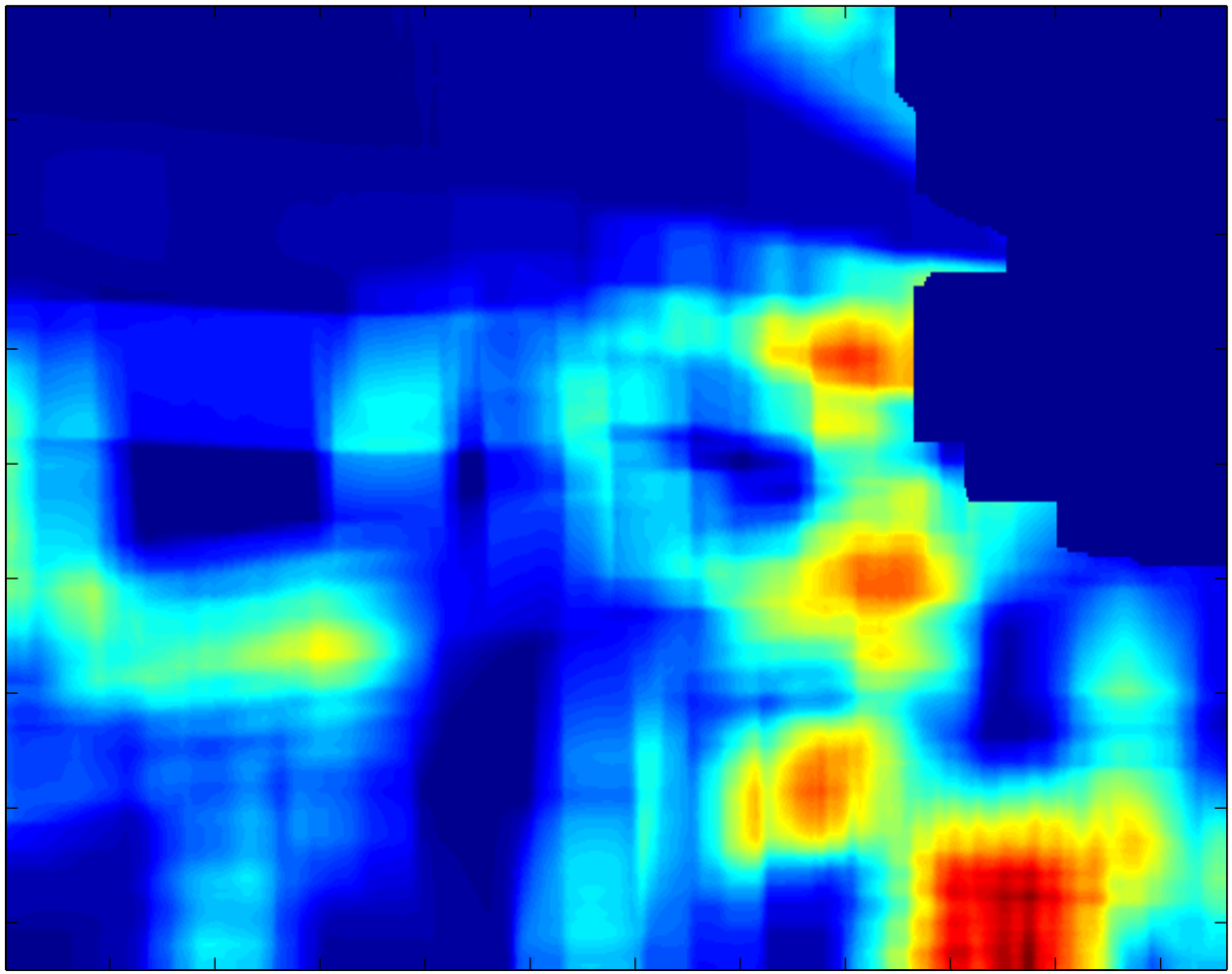




I want to know the areas that need more vertices

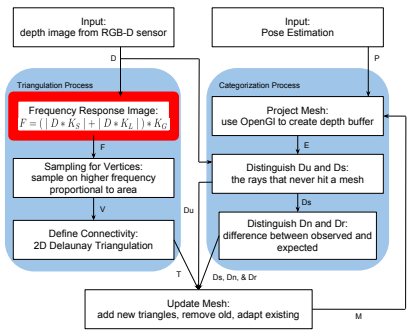
$$F = (\mid D * K_S \mid + \mid D * K_L \mid) * K_G$$

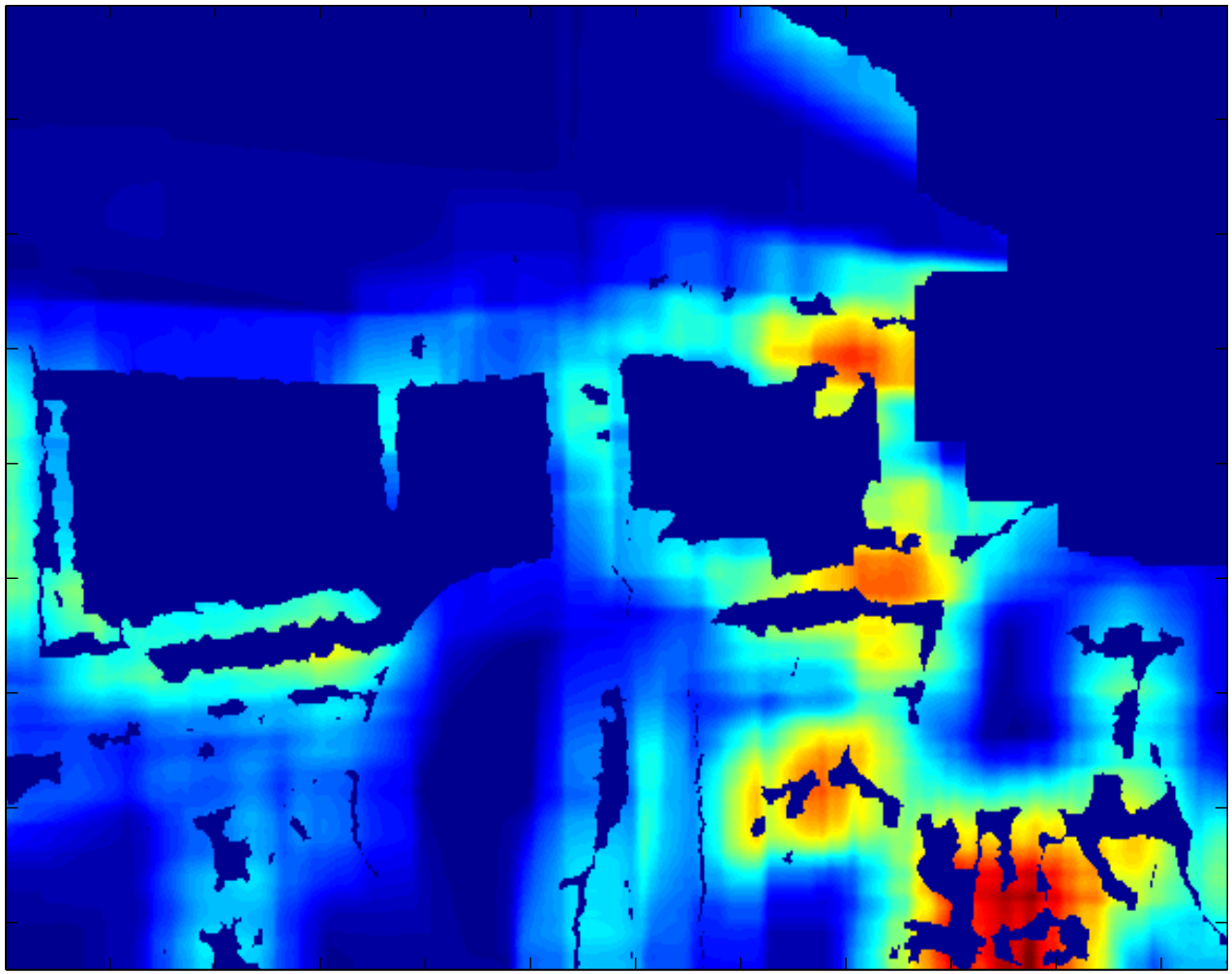




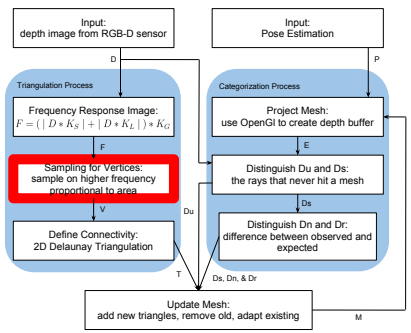
I want to know the areas that need more vertices

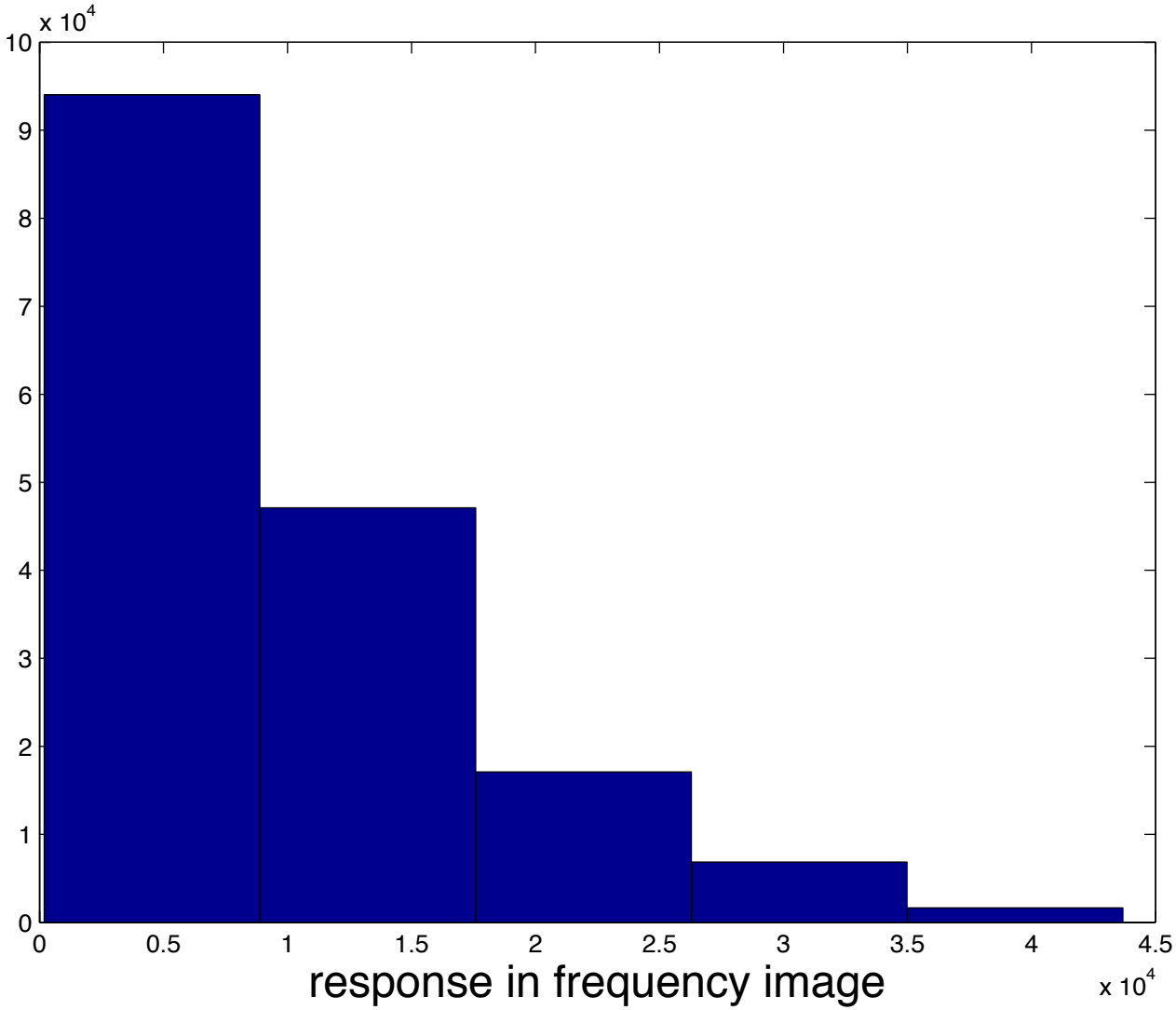
$$F = (\mid D * K_S \mid + \mid D * K_L \mid) * K_G$$



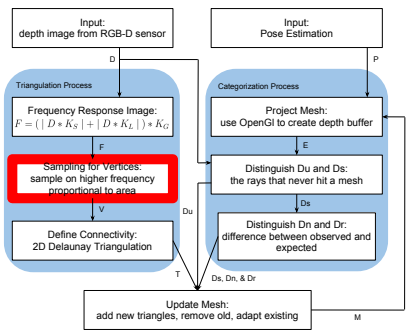


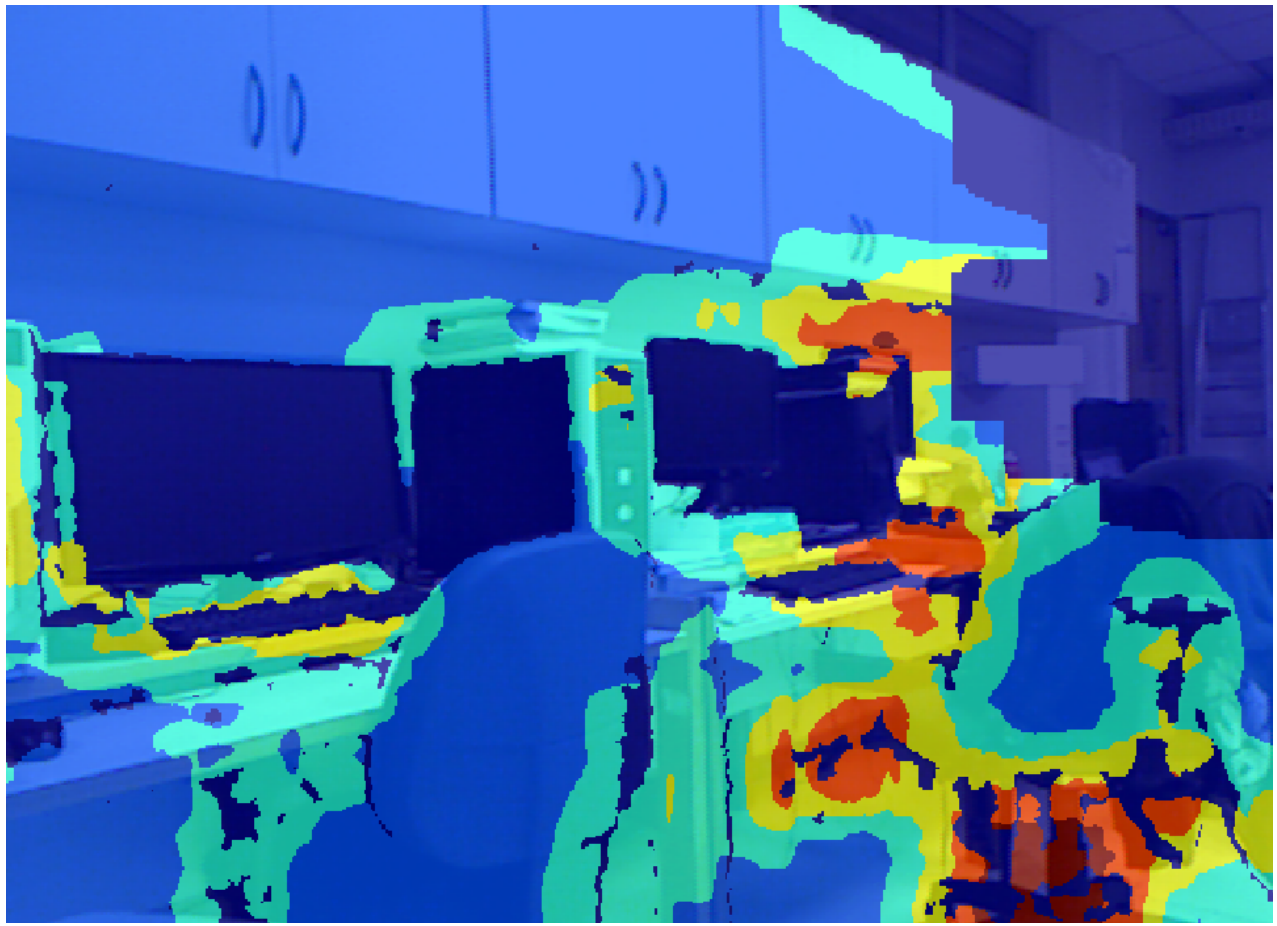
Determine the actual vertices by probabilistic sampling



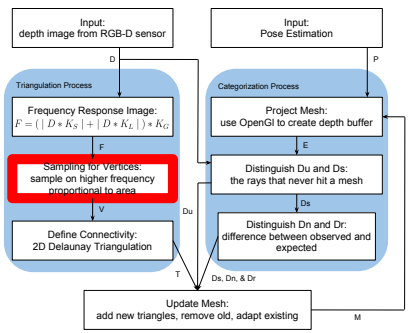


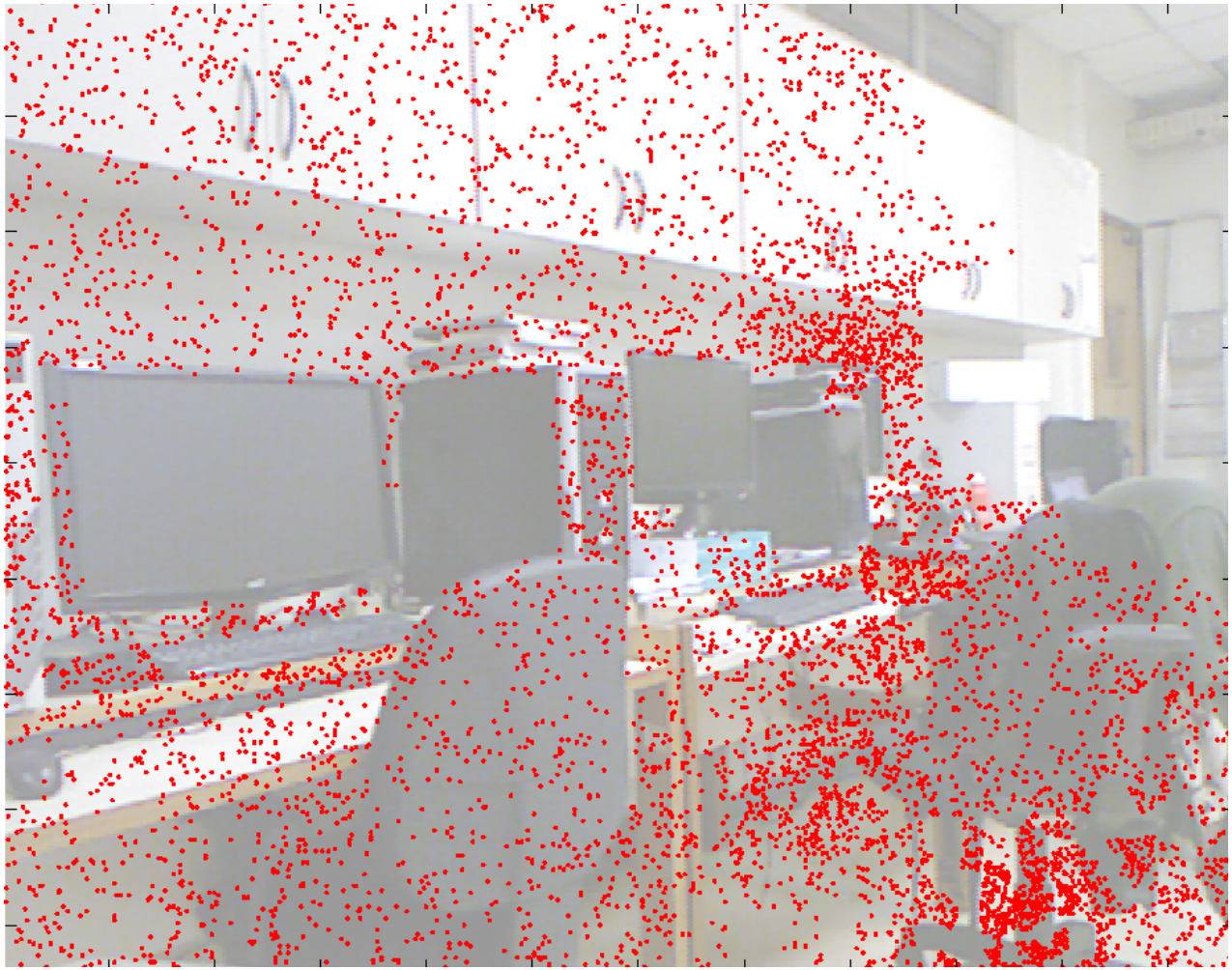
Determine the actual vertices by probabilistic sampling



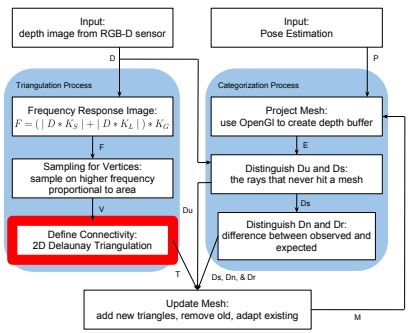


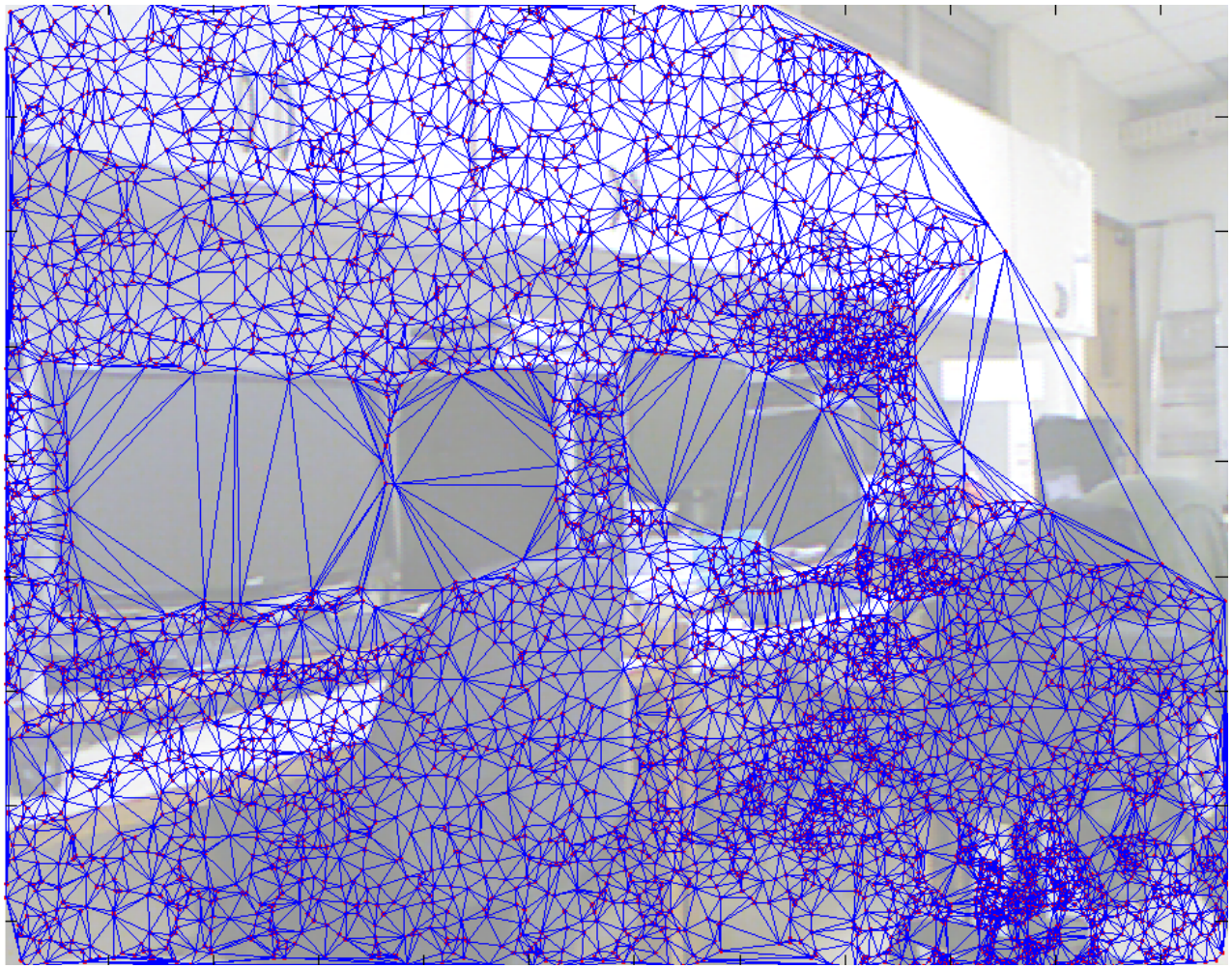
Determine the actual vertices by probabilistic sampling



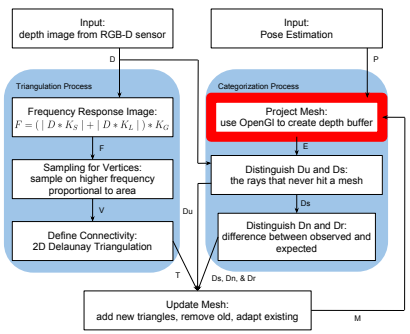


Connectivity of triangles (T) with 2D Delaunay

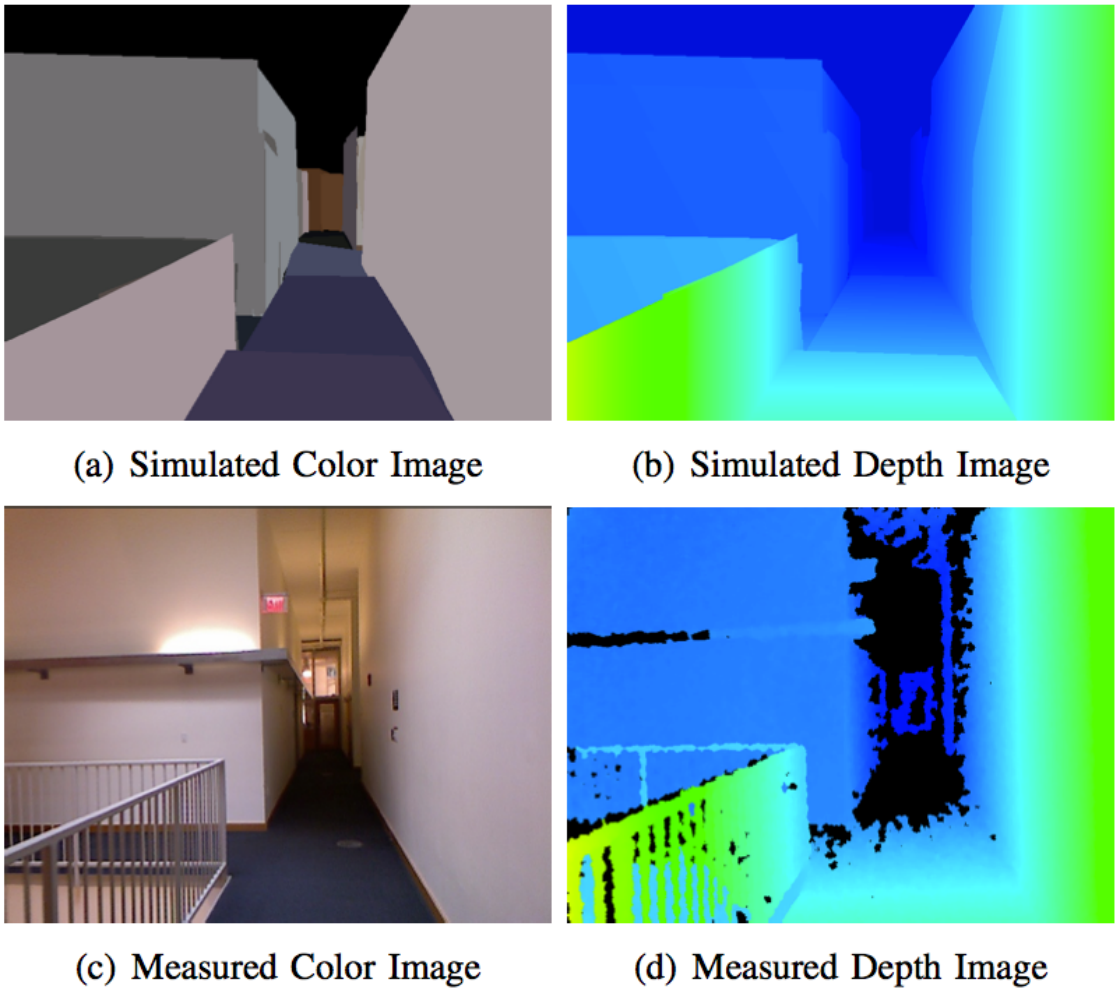




Project existing mesh onto depth image plane



This step gives the expected measurements E



(a) Simulated Color Image

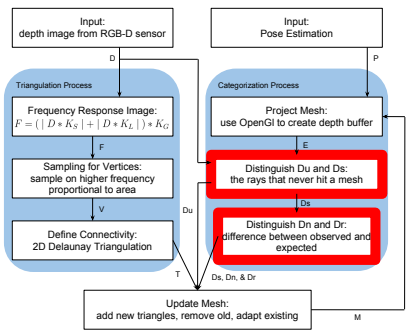
(b) Simulated Depth Image

(c) Measured Color Image

(d) Measured Depth Image

(Fallon et al. 2012) Efficient scene simulation for robust monte carlo localization using an RGB-D camera

Categorize measurements into D_u, D_s, D_n, & D_r

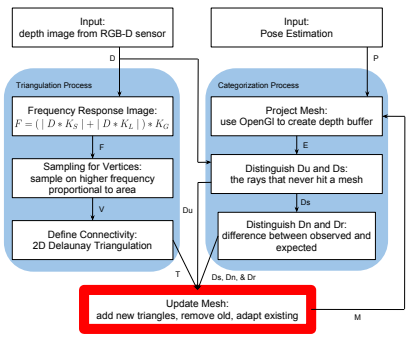


D_u - region where rays never hit the mesh
D_s - all valid measurements not containing D_u

Define the change as $C = E - D_s$
Perform blob analysis on C (twice)

D_n - blobs with $C > \epsilon$; add triangles using T
D_r - blobs with $C < \epsilon$; remove triangles

Update mesh



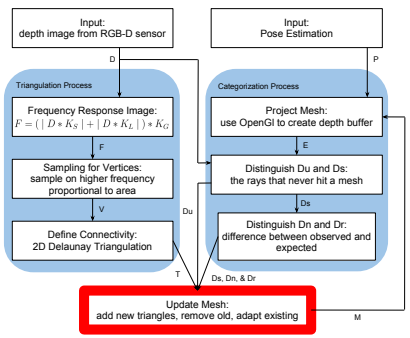
D_u - add triangles using T

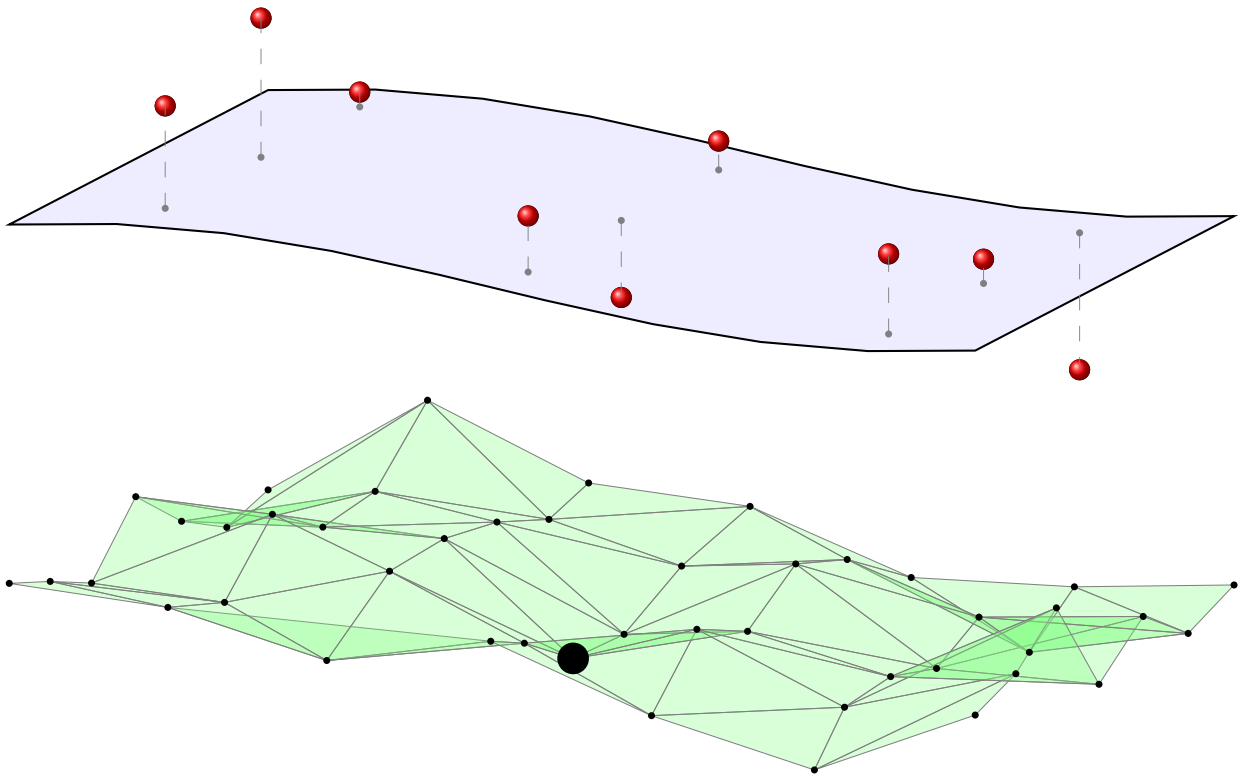
D_s - adapt

D_n - add triangles using T

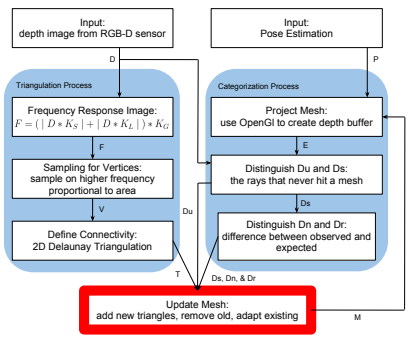
D_r - remove triangles

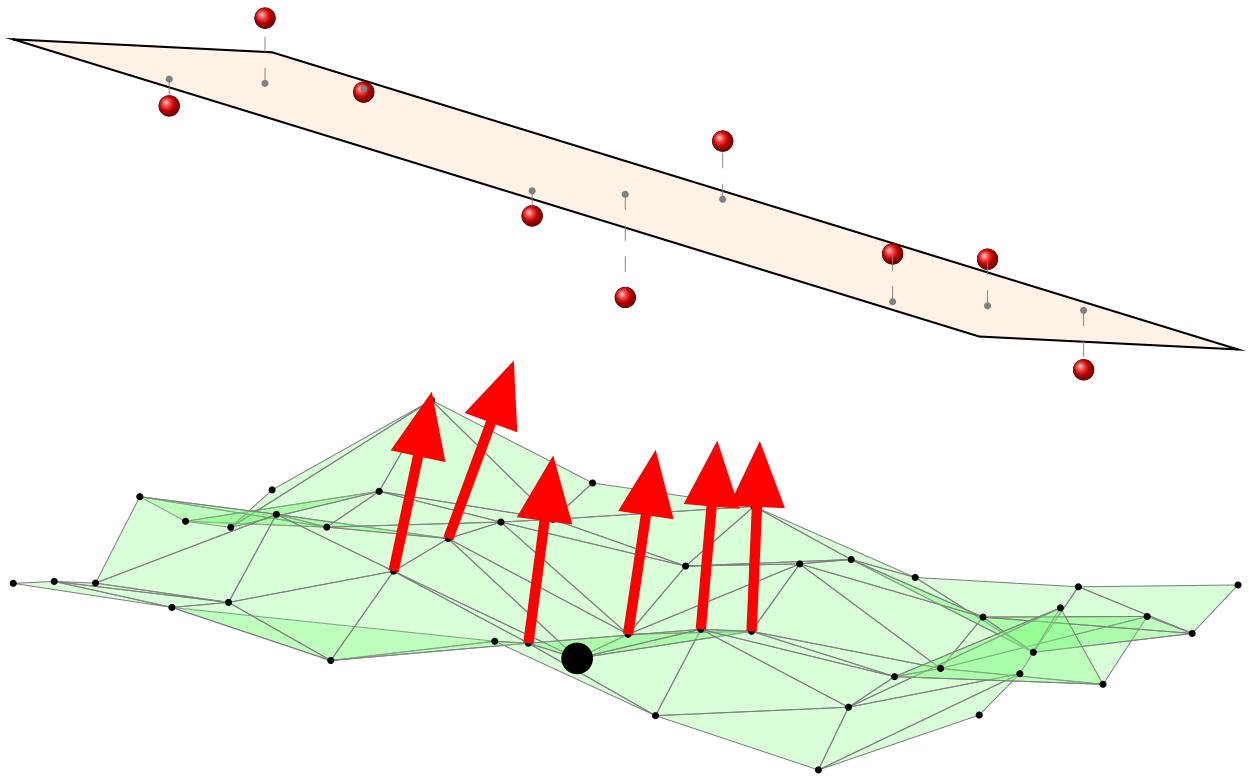
Adapt position of each vertex with QEM



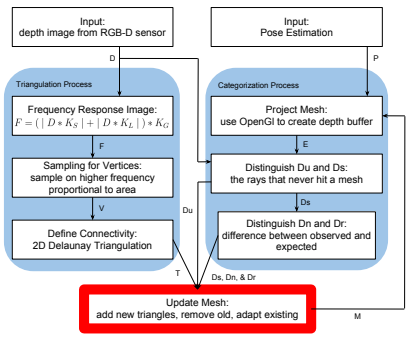


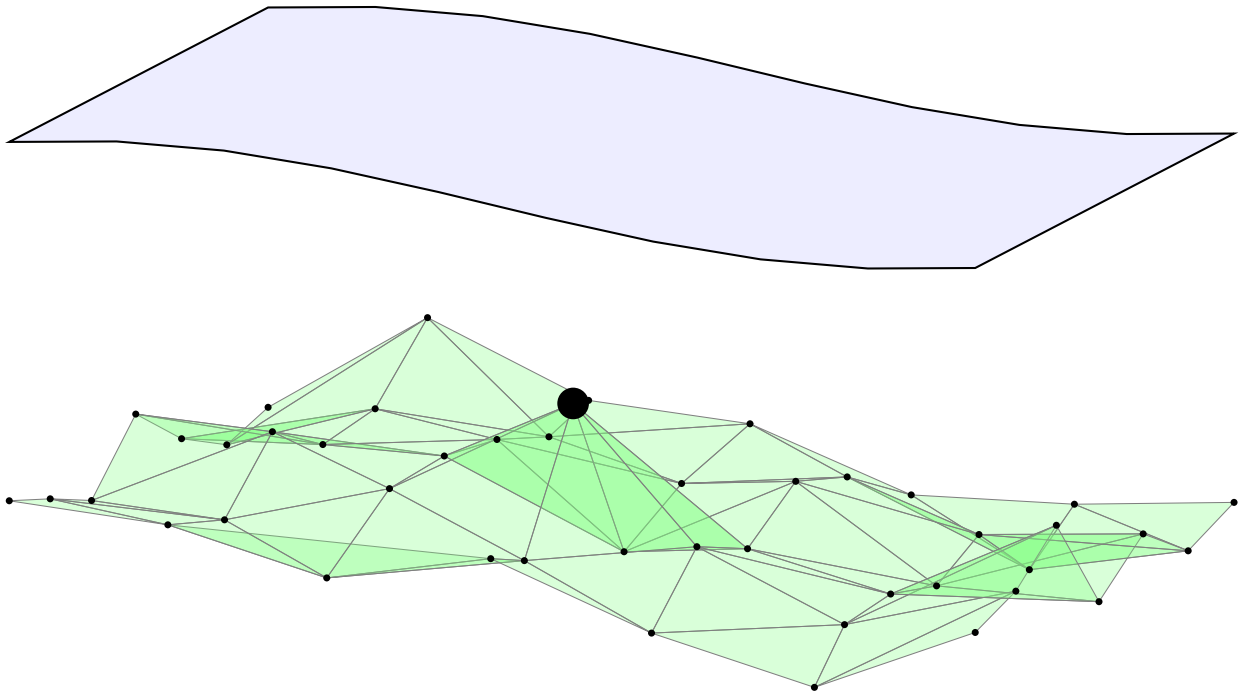
Adapt position of each vertex with QEM





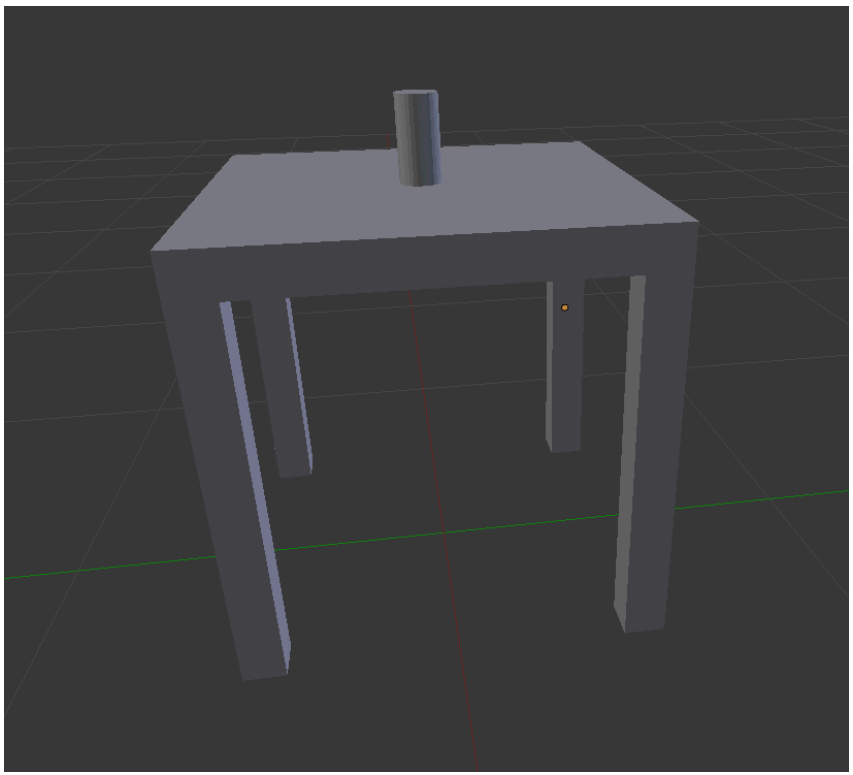
Adapt position of each vertex with QEM





Outline

A simulation pipeline to test in



Blender to define environment

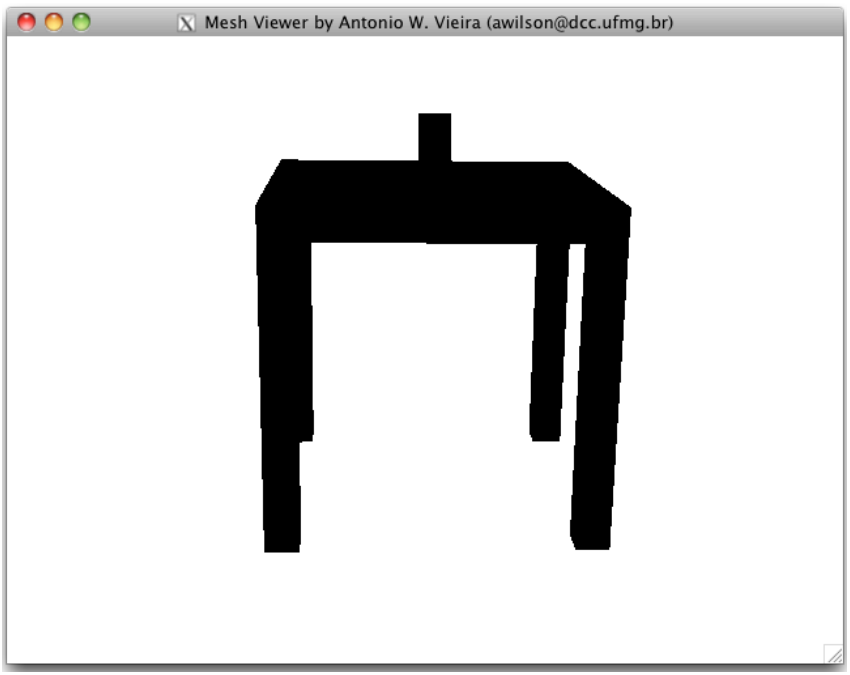
OpenGl to generate z-buffer

Open in Matlab and add noise

Can project to get point cloud

Can apply a mesh

A simulation pipeline to test in



Blender to define environment

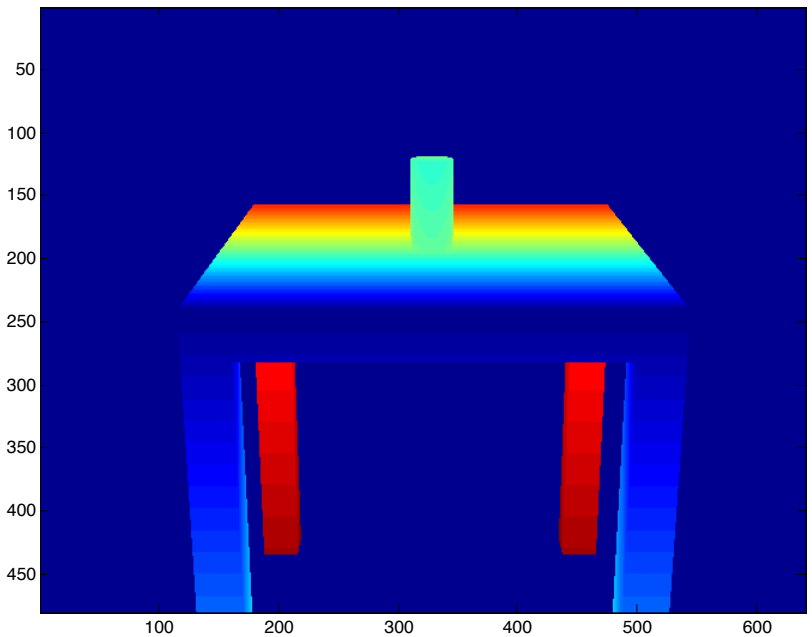
OpenGl to generate z-buffer

Open in Matlab and add noise

Can project to get point cloud

Can apply a mesh

A simulation pipeline to test in



Blender to define environment

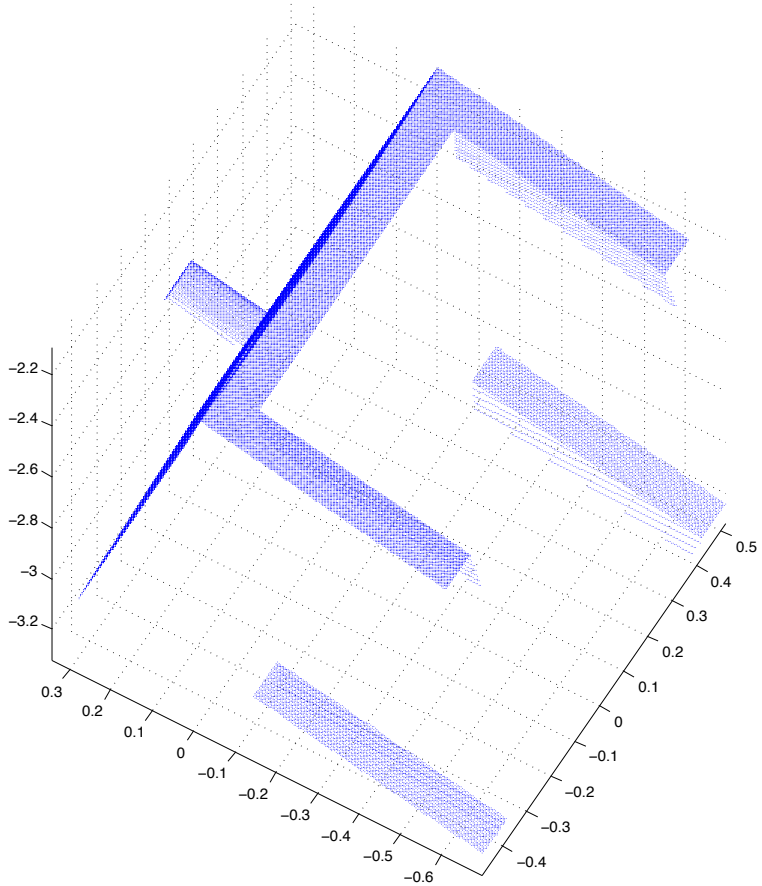
OpenGl to generate z-buffer

Open in Matlab and add noise

Can project to get point cloud

Can apply a mesh

A simulation pipeline to test in



Blender to define environment

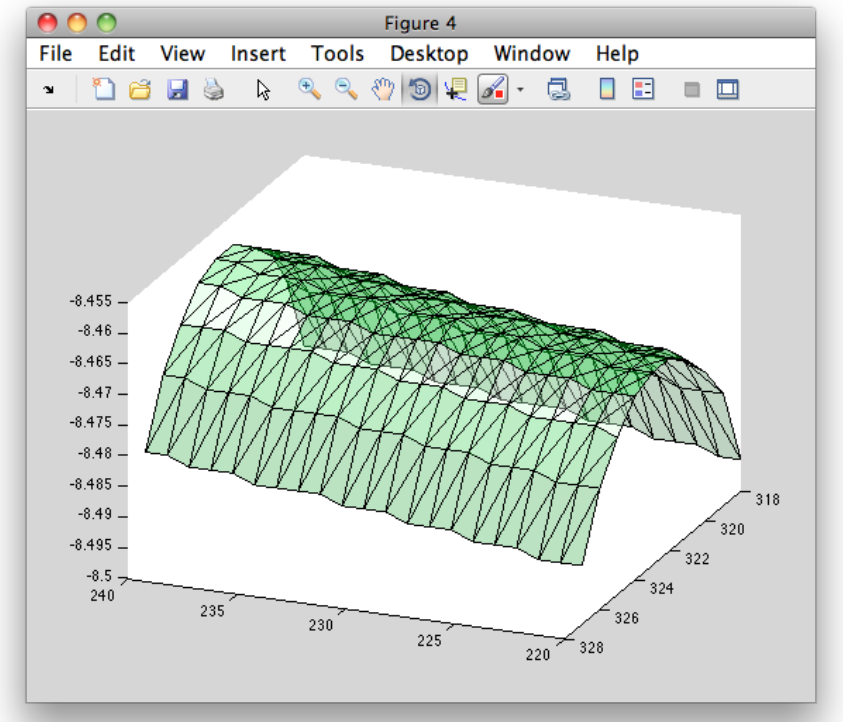
OpenGl to generate z-buffer

Open in Matlab and add noise

Can project to get point cloud

Can apply a mesh

A simulation pipeline to test in



Blender to define environment

OpenGl to generate z-buffer

Open in Matlab and add noise

Can project to get point cloud

Can apply a mesh

What experiments will I do to validate my method?

- Static scene; static object; static sensor

  Simple scene such as a cup on a table. Want to see convergence.
- Static scene; static object; dynamic sensor

  Want to test ability to add triangles from unseen parts of environment.
- Dynamic scene; static object; static sensor

  Detect new and removed objects from scene. Second cup and its removal.
- Dynamic scene; dynamic object; static sensor

  A much more thorough test of removing and adding mesh.
- Dynamic scene; dynamic object; dynamic sensor

  Complete system working in simulation.
- Repeat with real data

Summary

- Importance of a mesh-based representation.
- Main problem is measurement categorization.
- Can be solved with fast image based techniques.

- Things I'm scared of
  - Merging boundaries of areas.
  - Incremental adaptation with QEM.
  - Pose estimation will never be that good.

Questions or Comments?