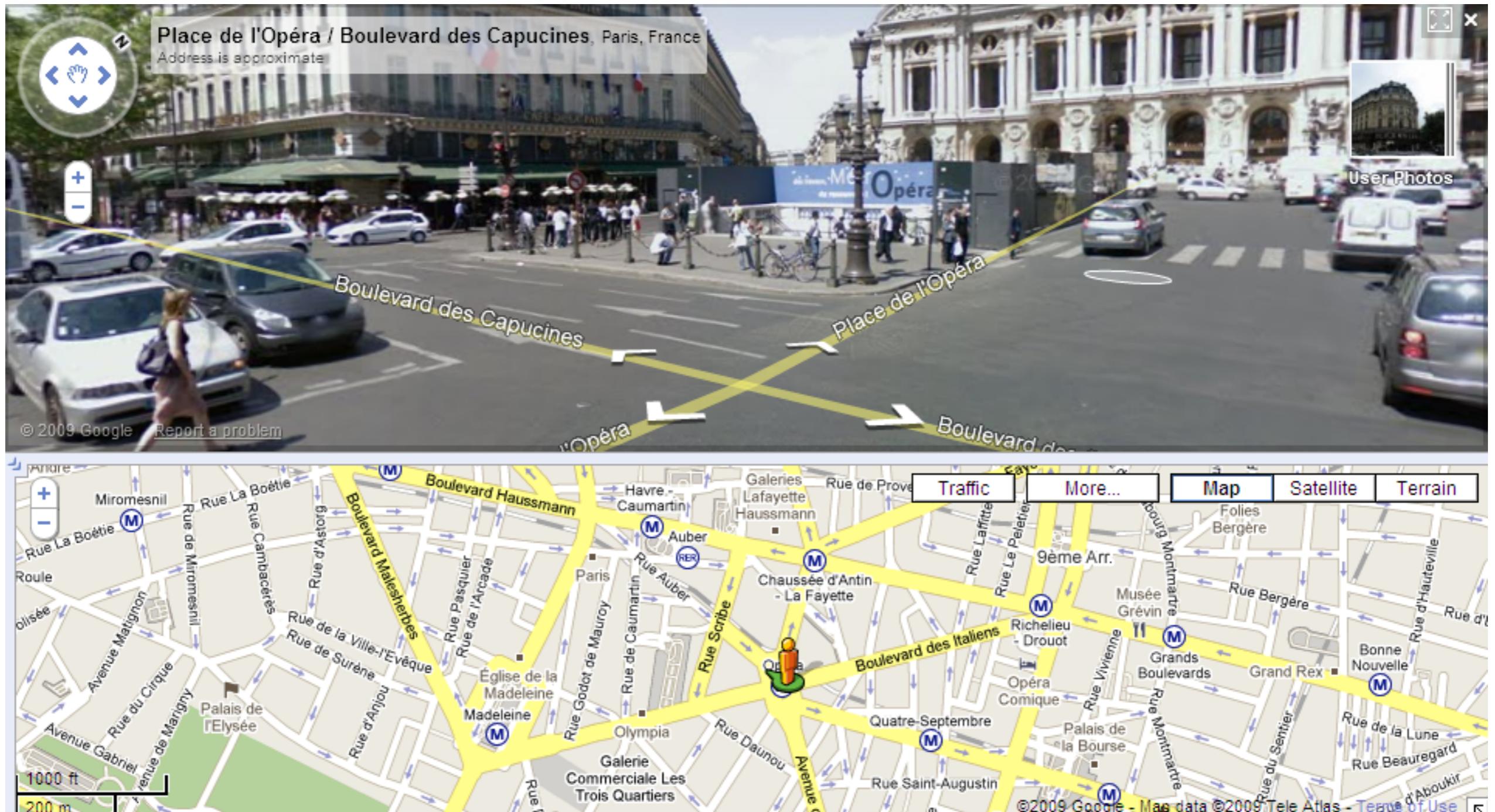


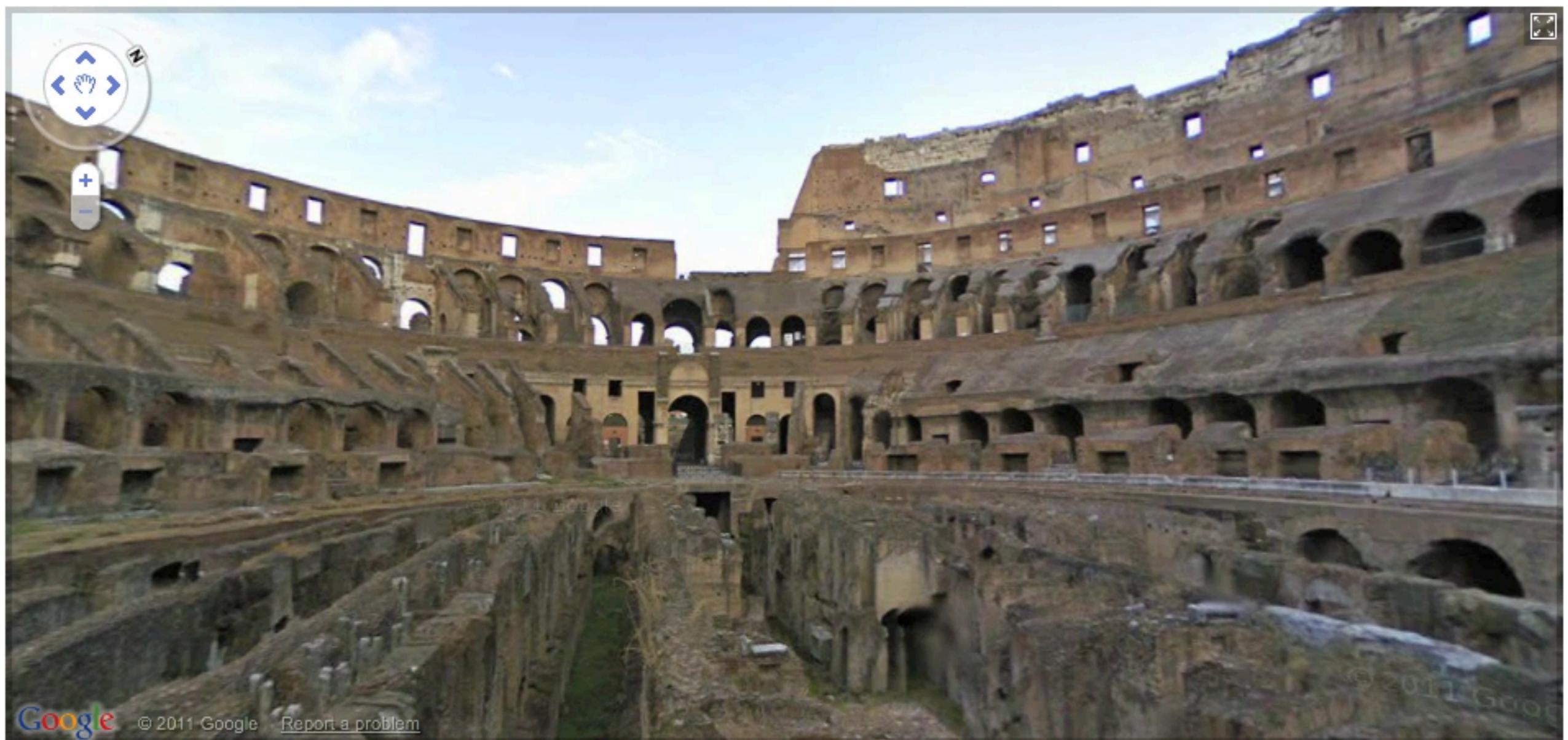
# Vision Applications: Google Street View

## CS 378 Fall 2014 Meeting 17

Bryan Klingner 23 October 2014



# Street View is **everywhere**.



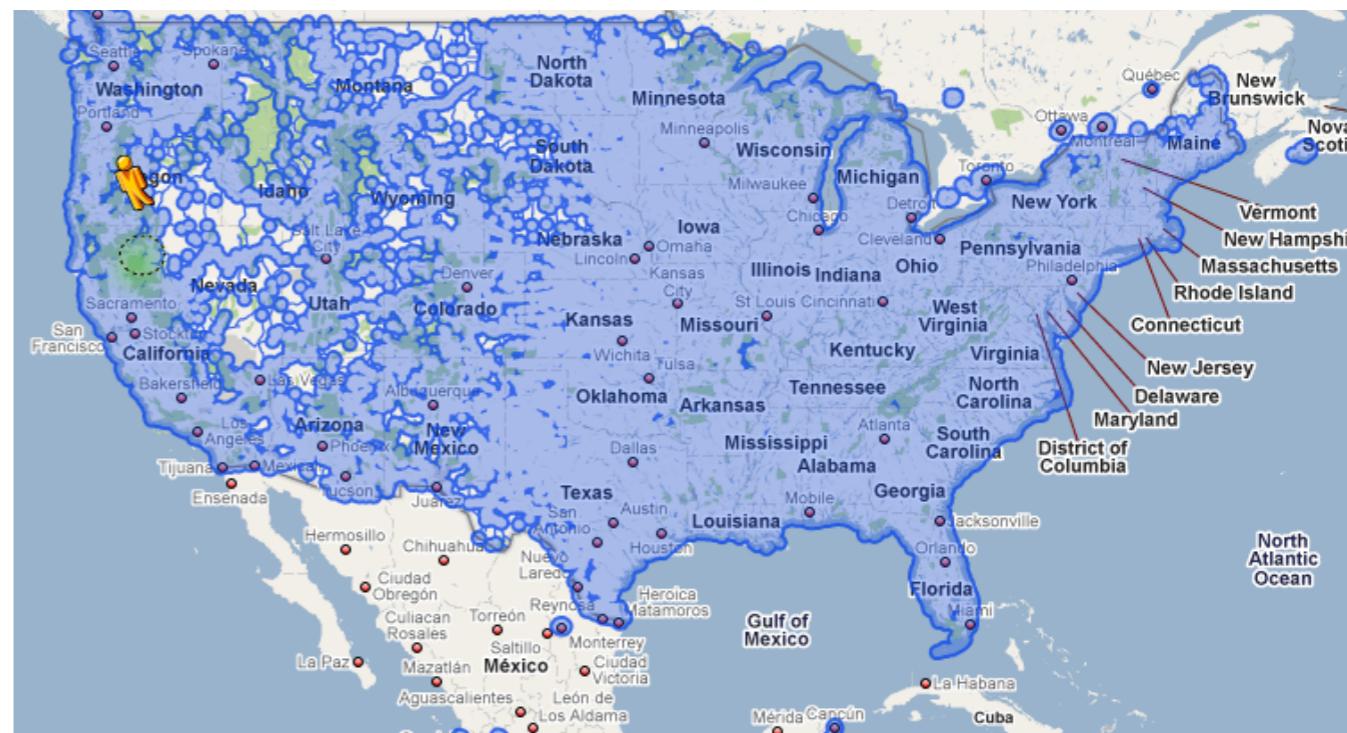
# Street View is **everywhere**.



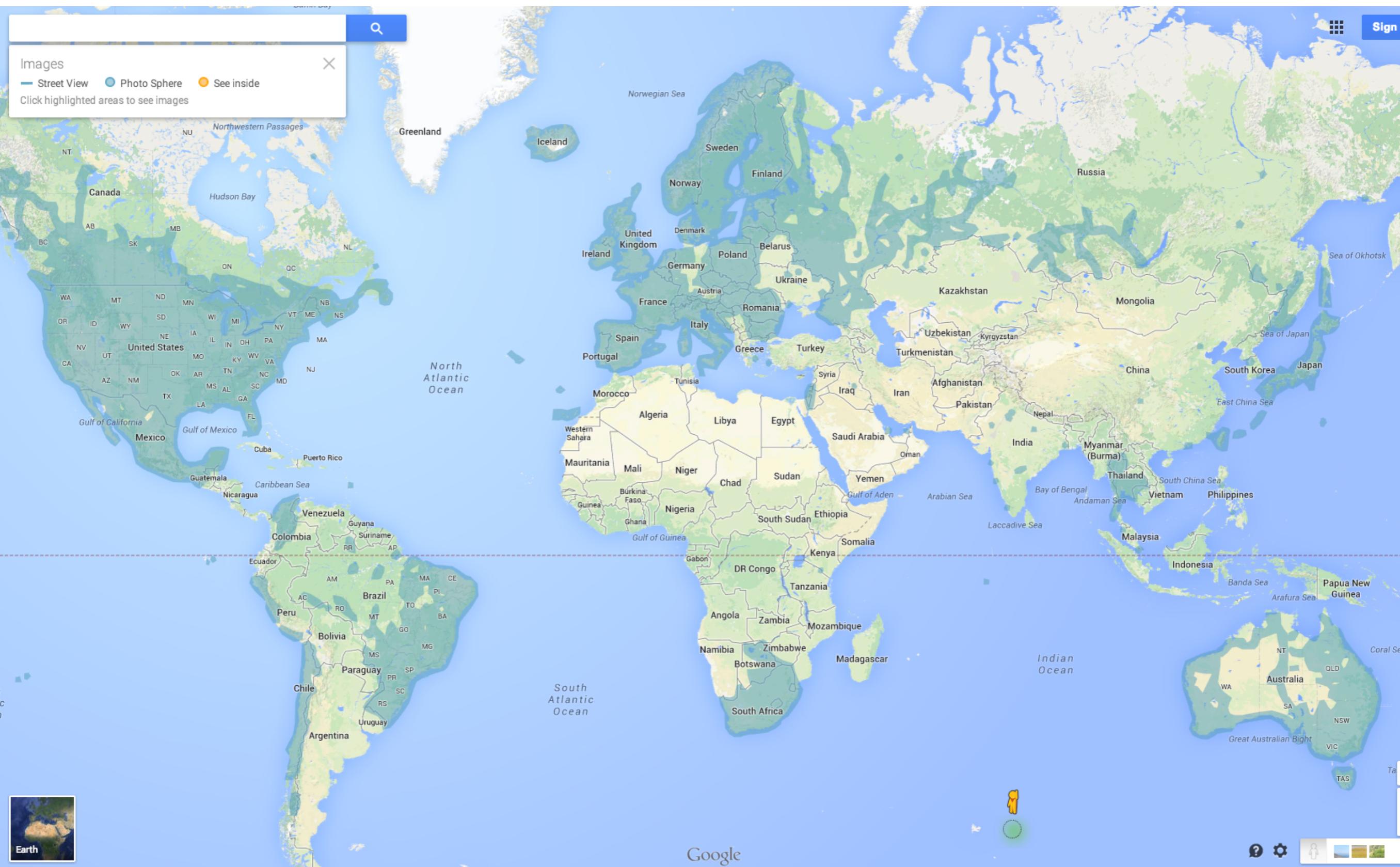
# Street View is **everywhere.**



# US Growth: 2007-2009



# Now in 50 countries, with more to come!



# We're far from finished.

We're creating an **immersive, accurate, globally-consistent** model of the world



*Grand Theft Auto IV, Rockstar Games*

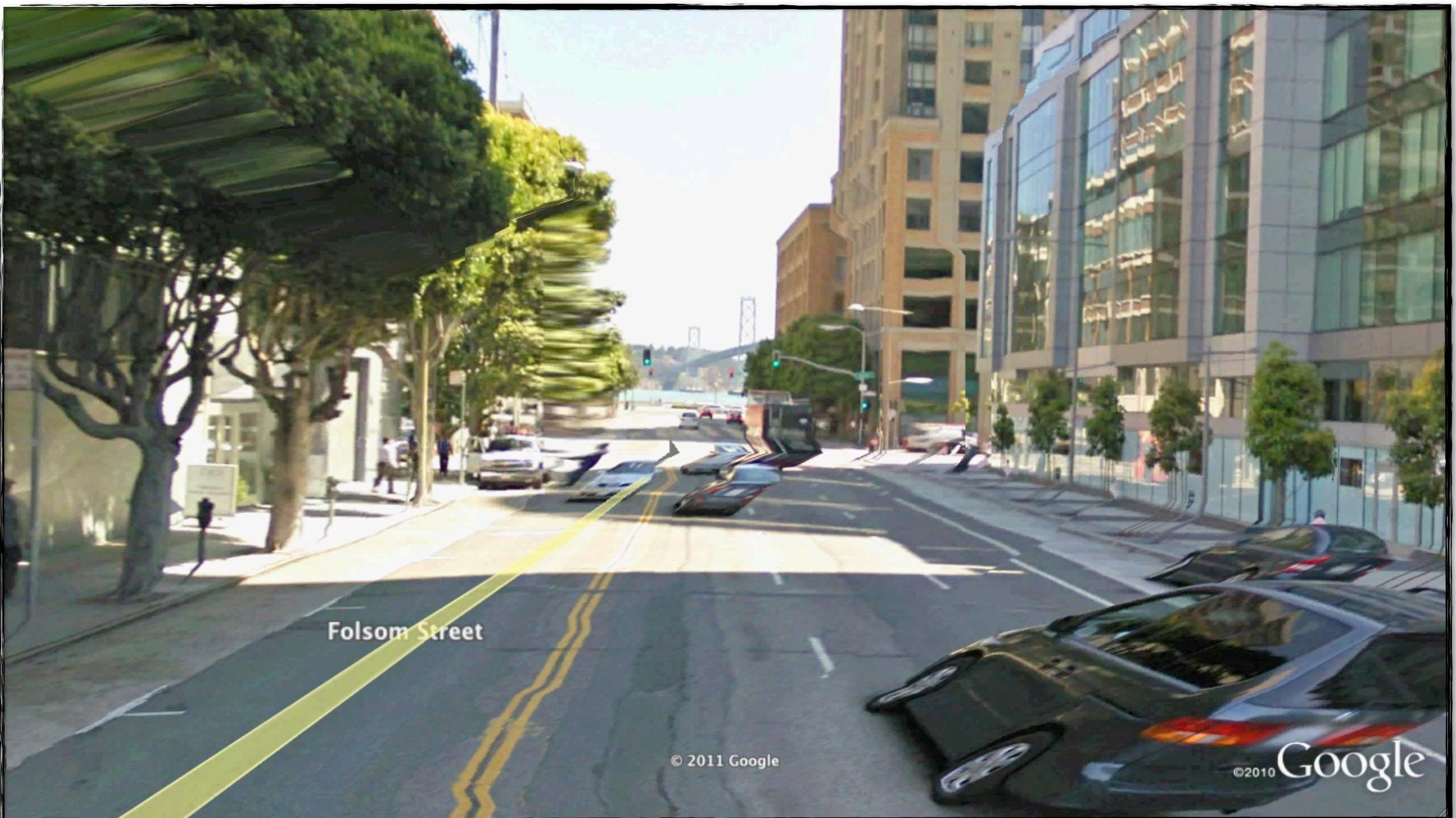
# Progress so far...

“Click2go” pancake/waffle - June 2009

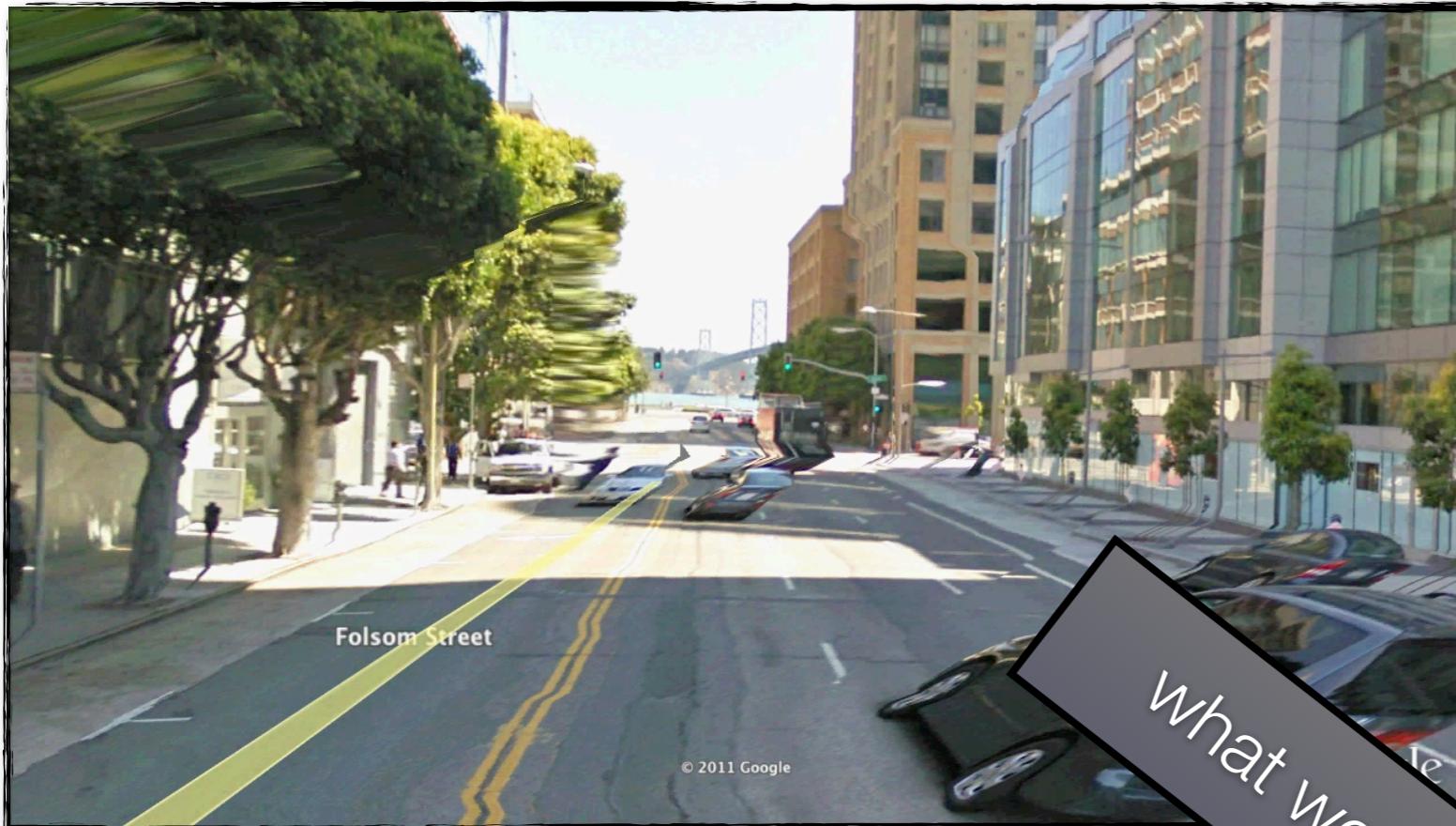


# Progress so far...

Autopia in Google Earth 6 - December 2010



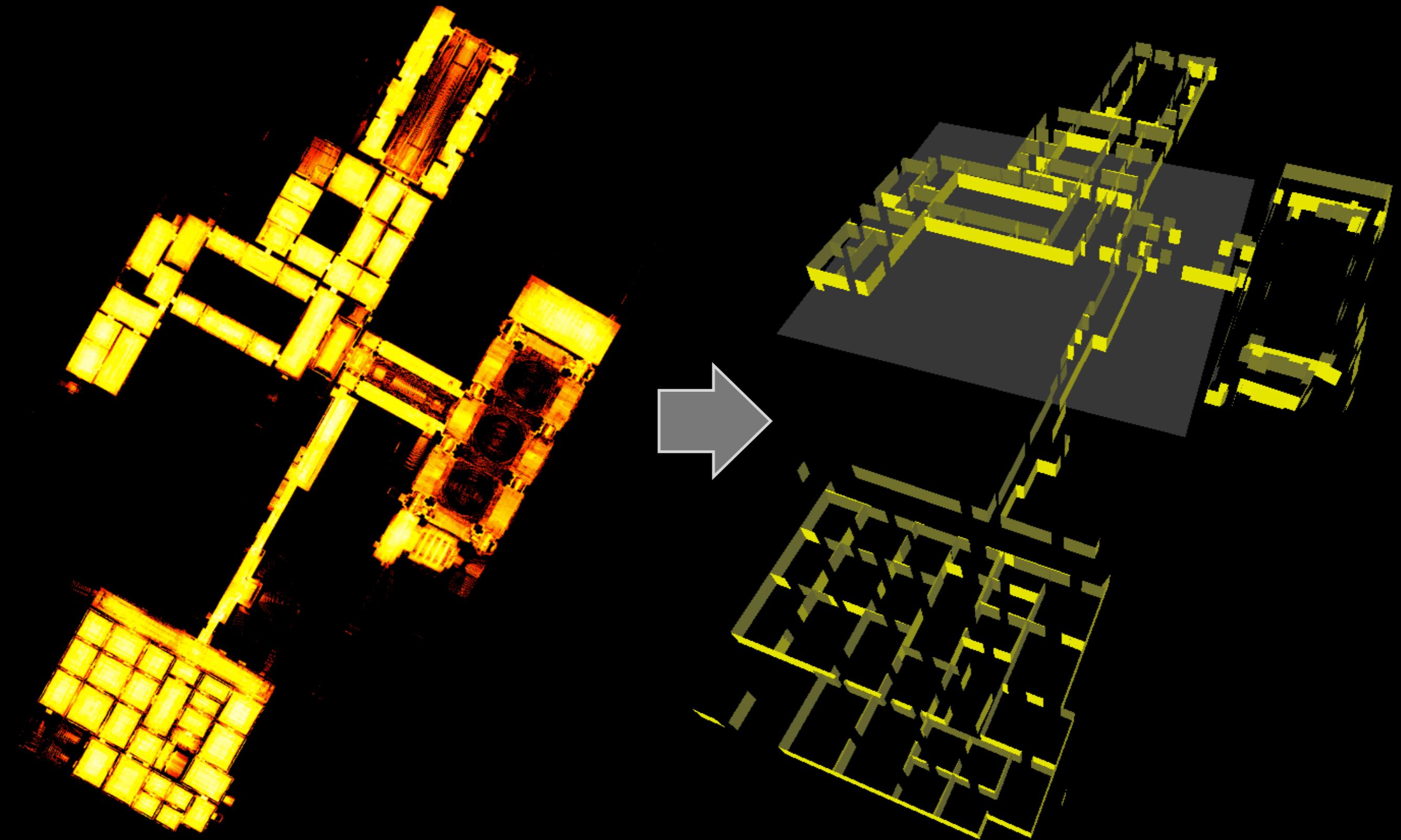
# Getting from here to there is **our job.**



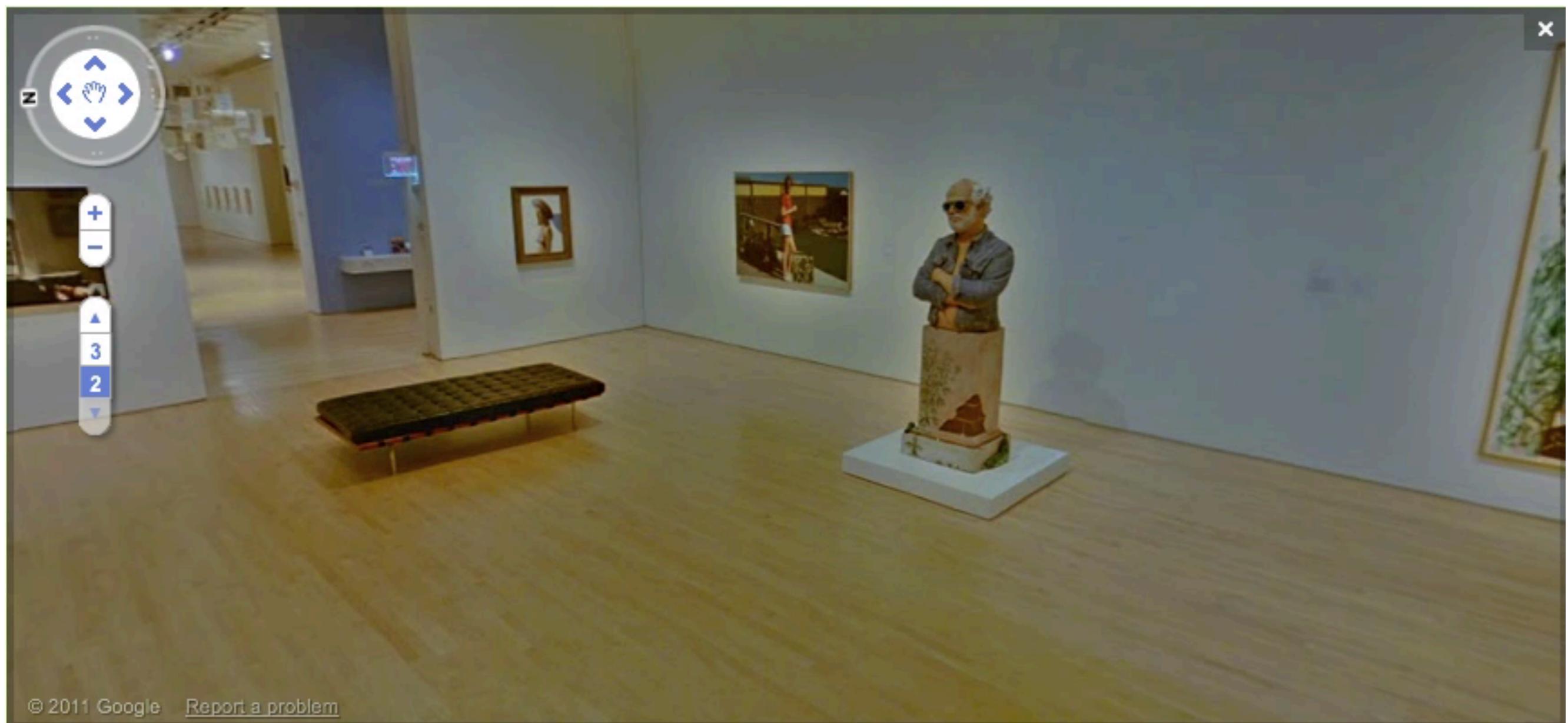
# Street View is also Indoors: The Google Art Project



**Goal:** convert raw laser data into a **concise**,  
**globally-accurate** 3D model of building interiors



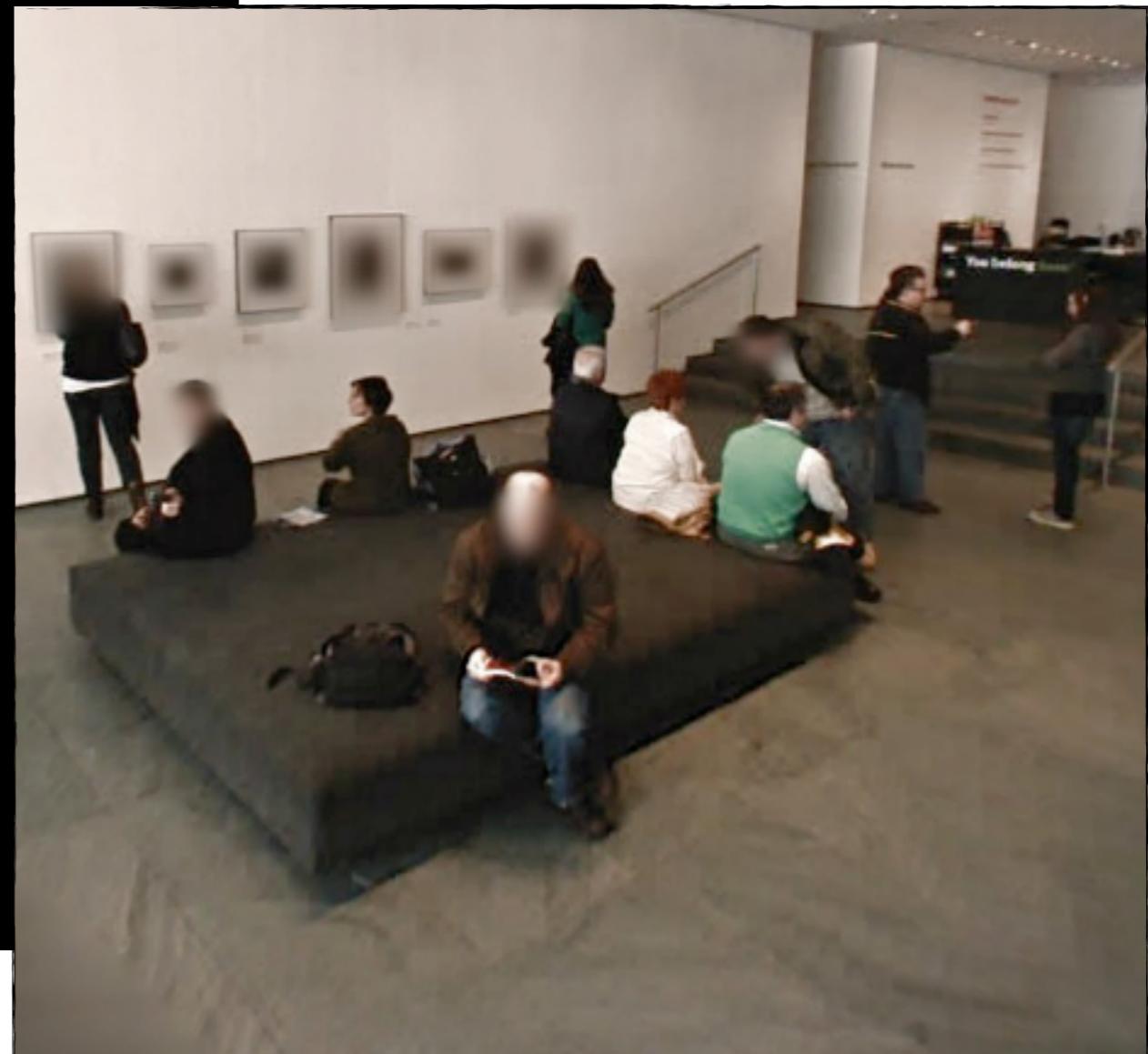
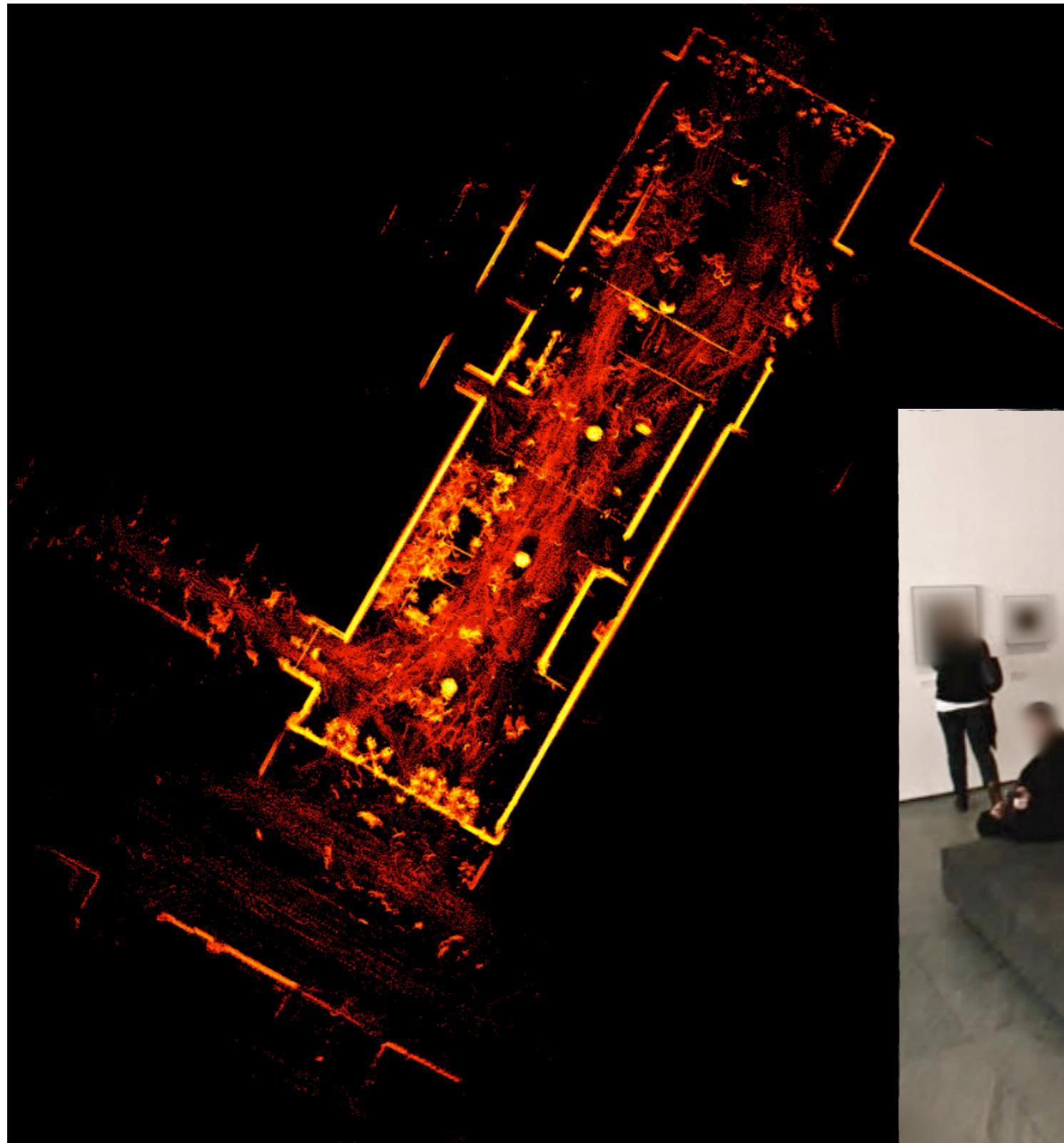
# Why? navigation.



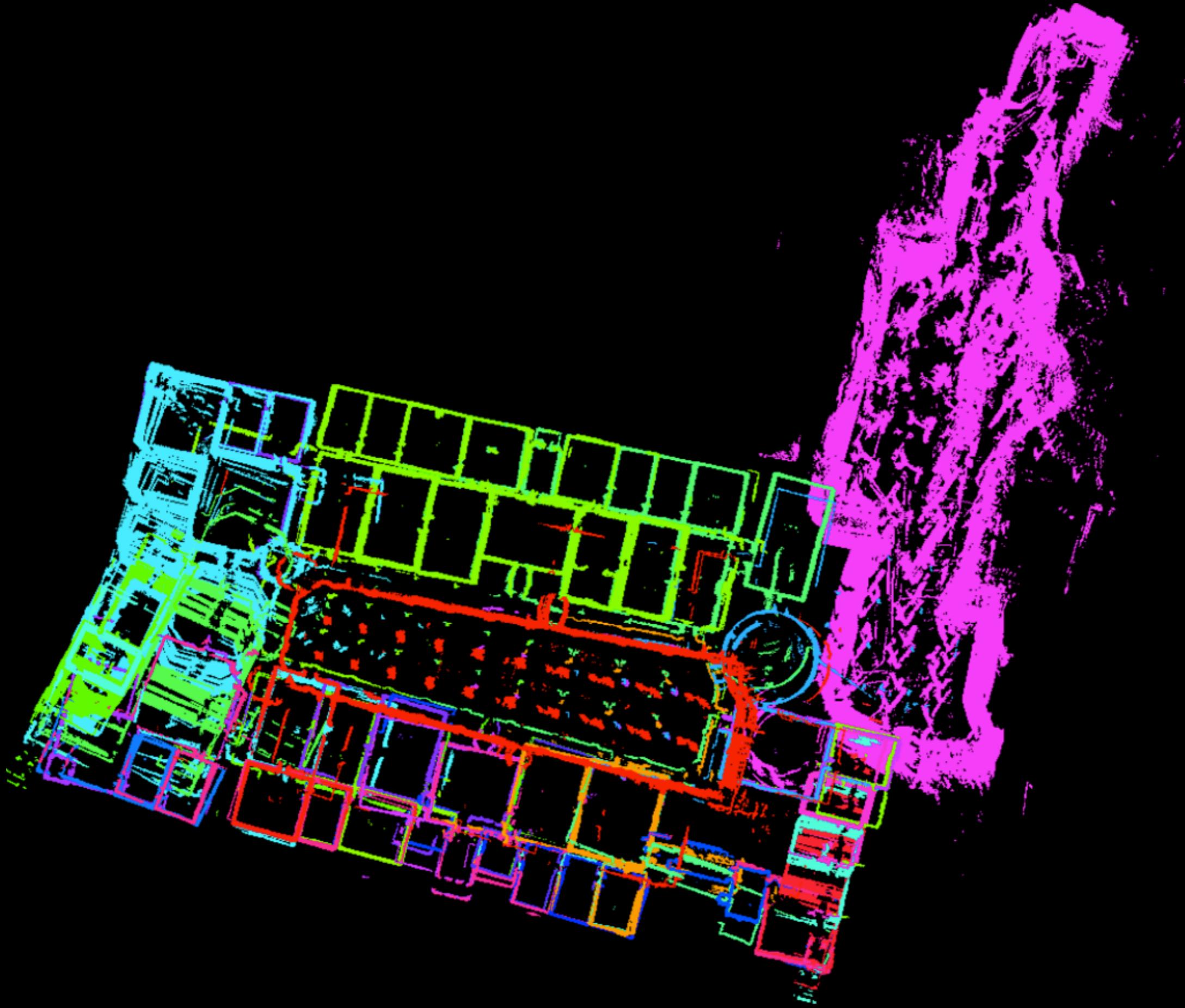
# Why? Parallax correction.



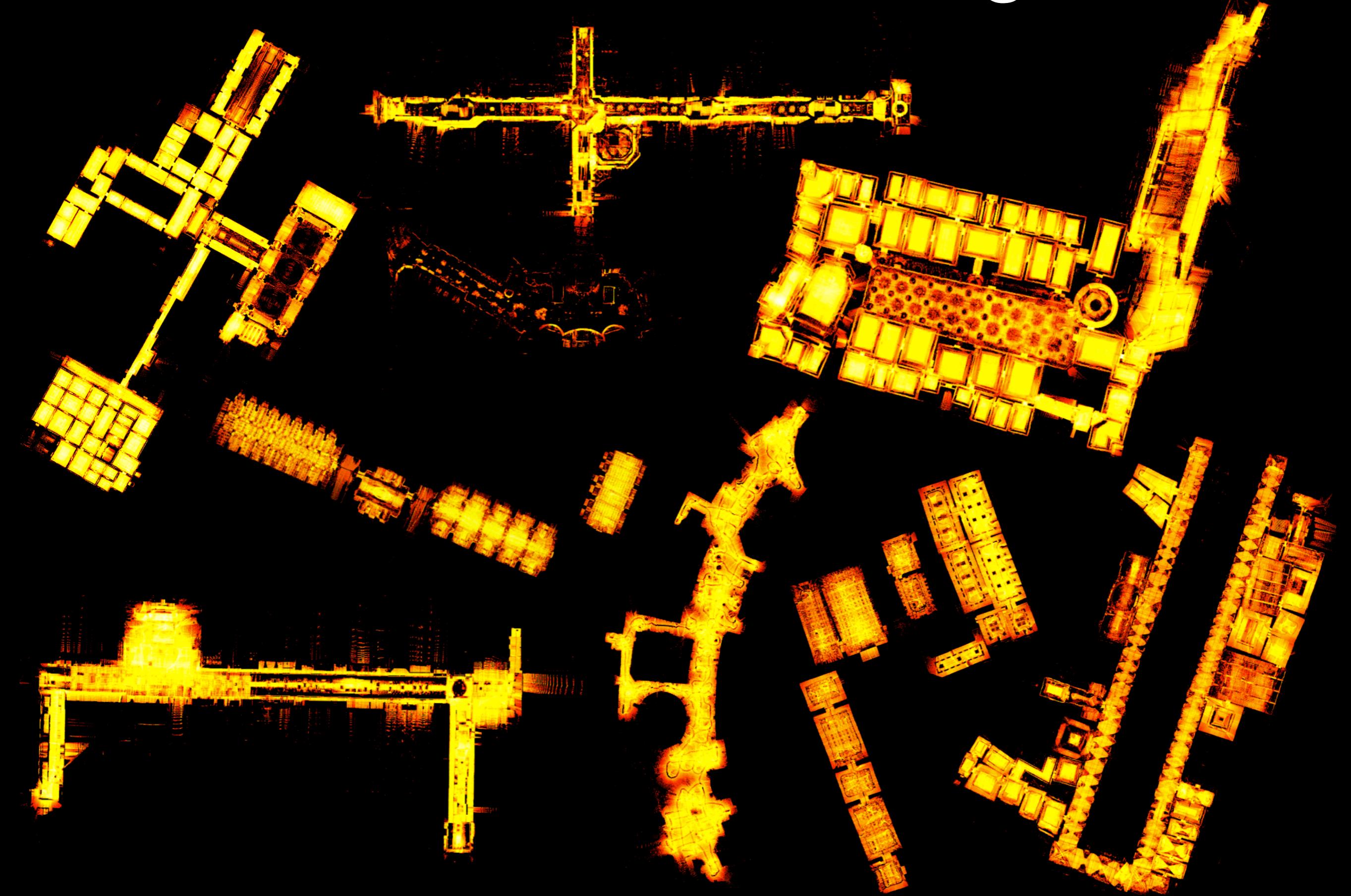
# What makes this hard? **Noise.**



What makes this hard? **Bad Pose.**



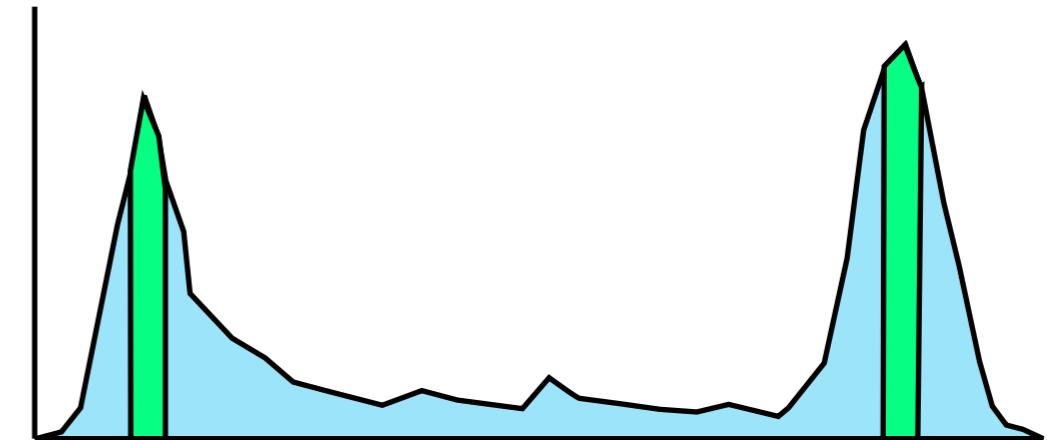
# What makes this hard? **Google Scale.**



# Building the model in 3 (easy?) steps:

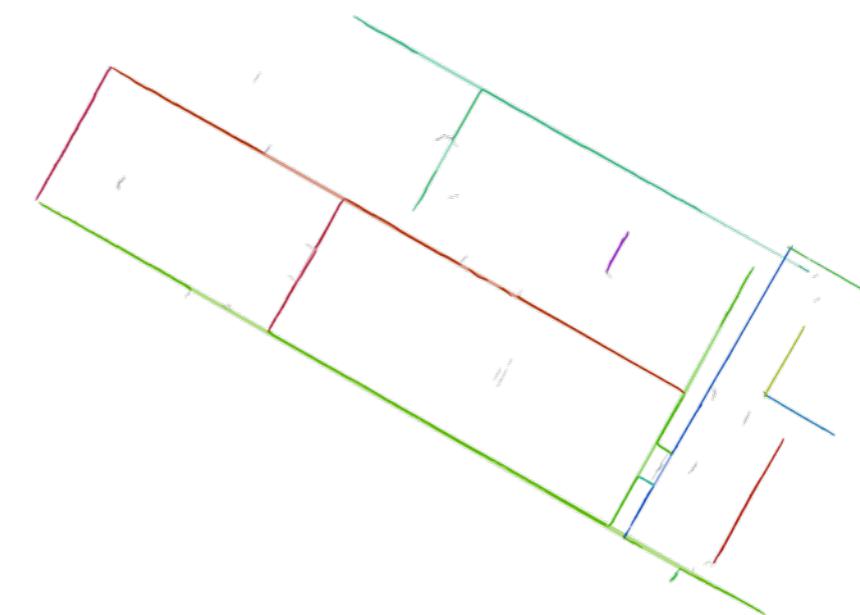
1

Laser scan  
segmentation



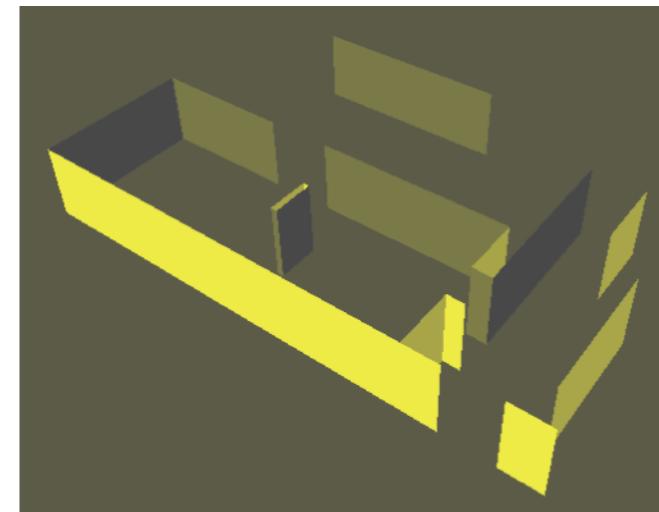
2

Wall geometry  
reconstruction



3

Post processing



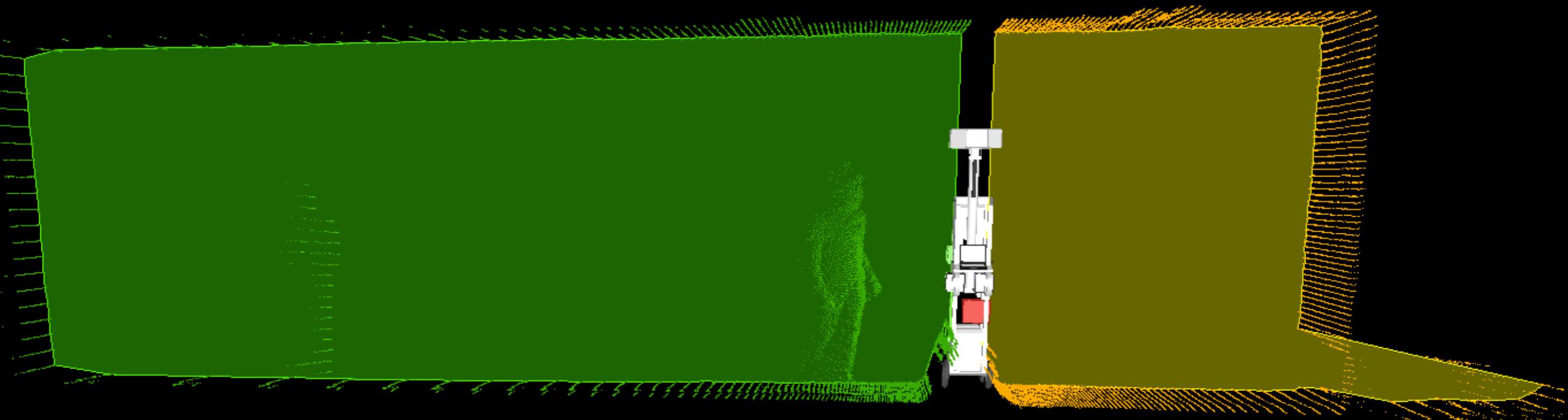
1

# Laser scan segmentation



left laser scan

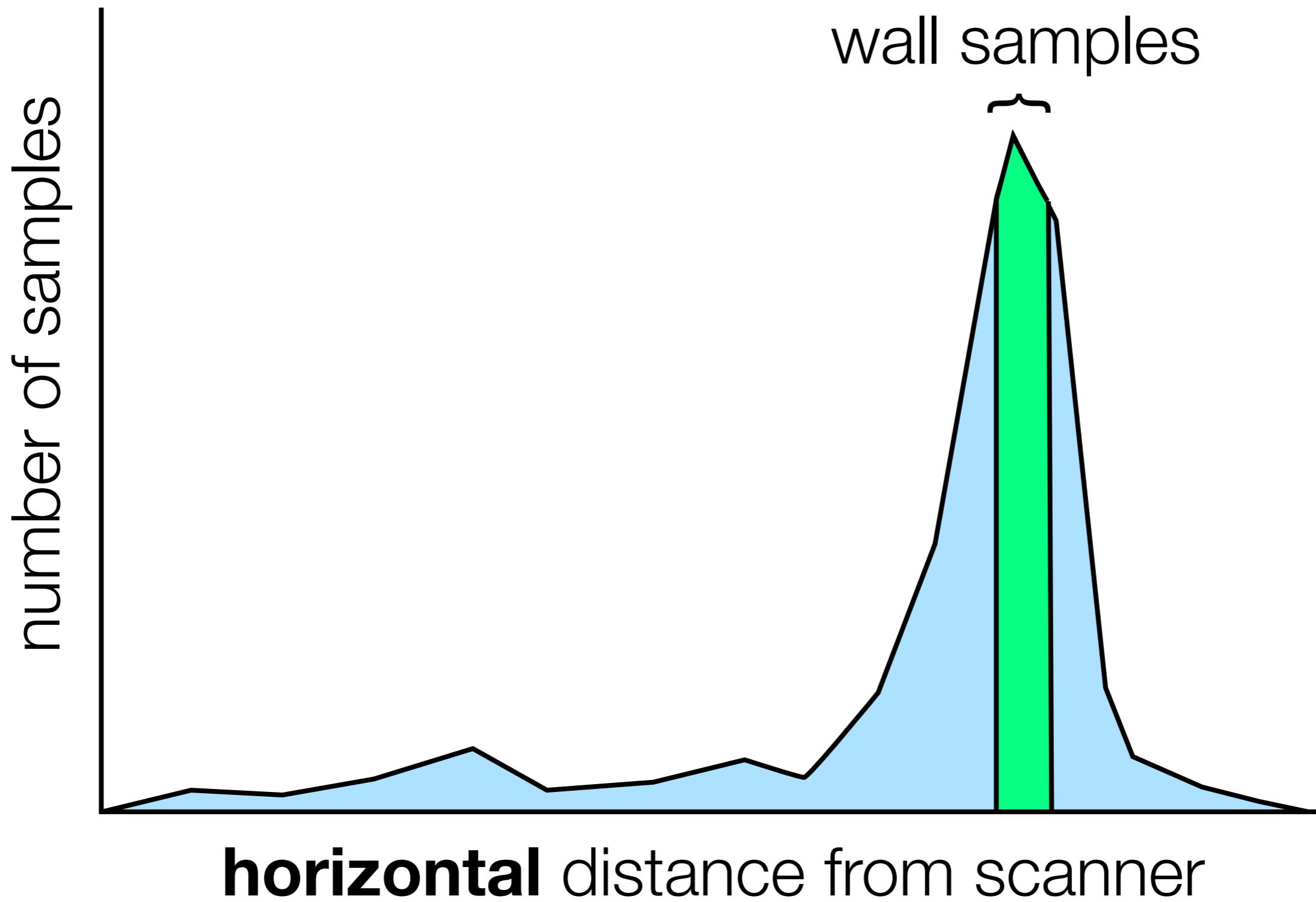
right laser scan



laser scans as viewed in **cbview** by djfilip

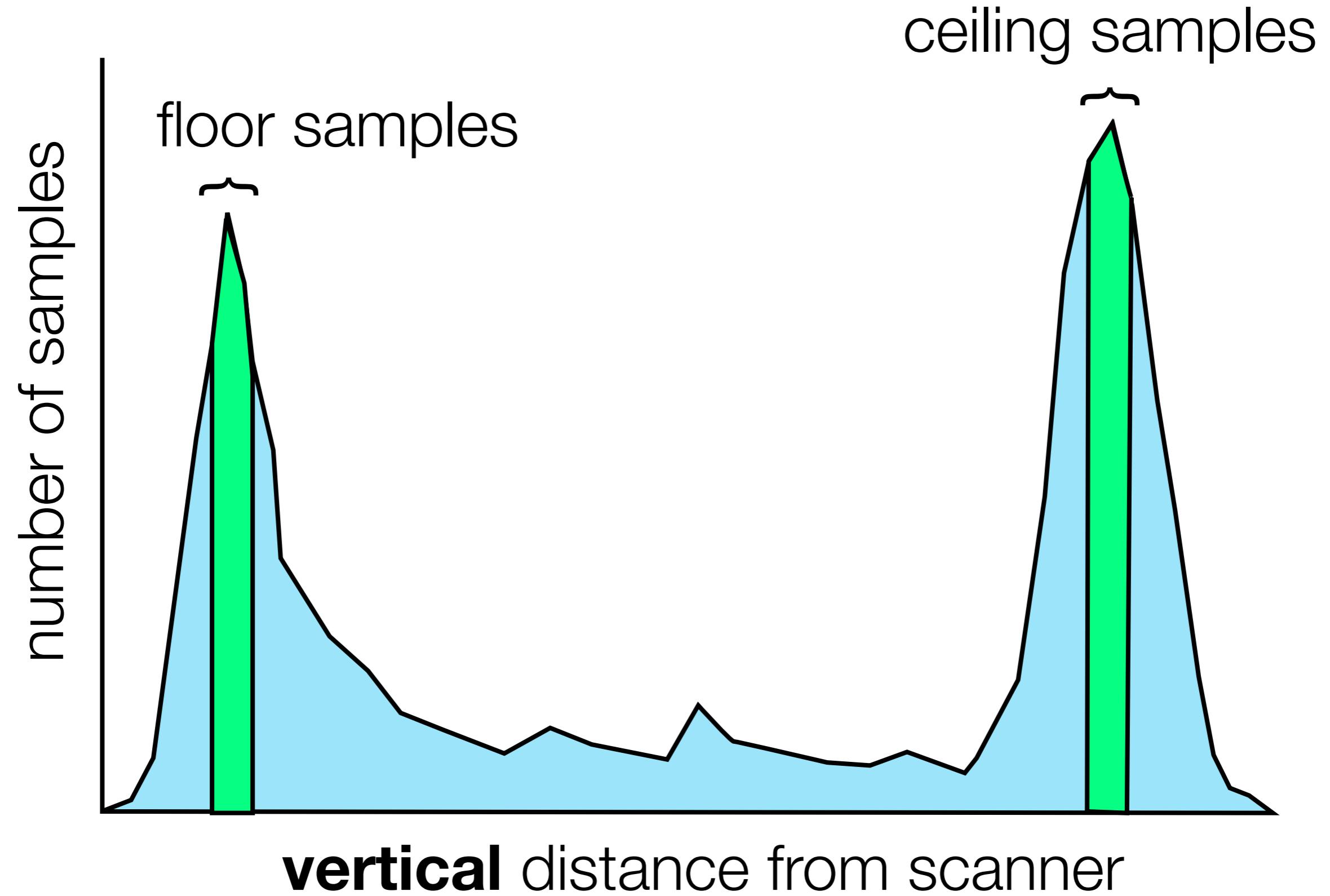
1

# Laser scan segmentation



1

# Laser scan segmentation



all samples



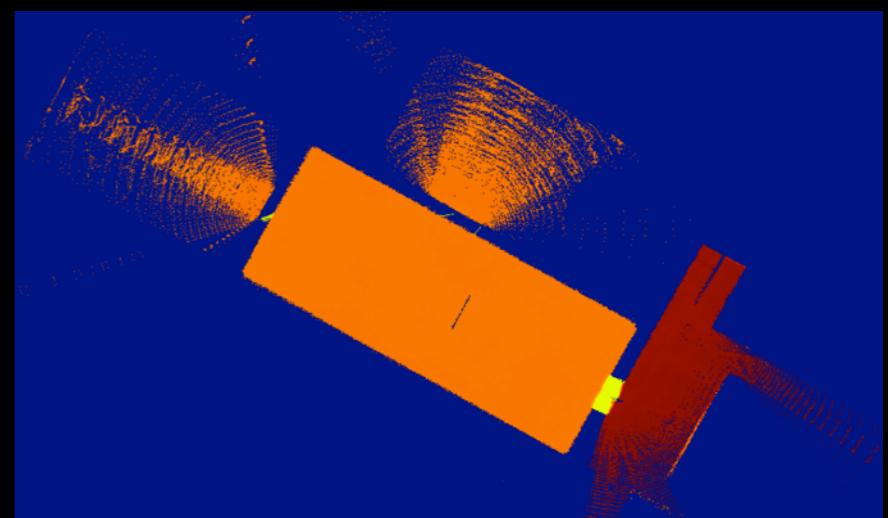
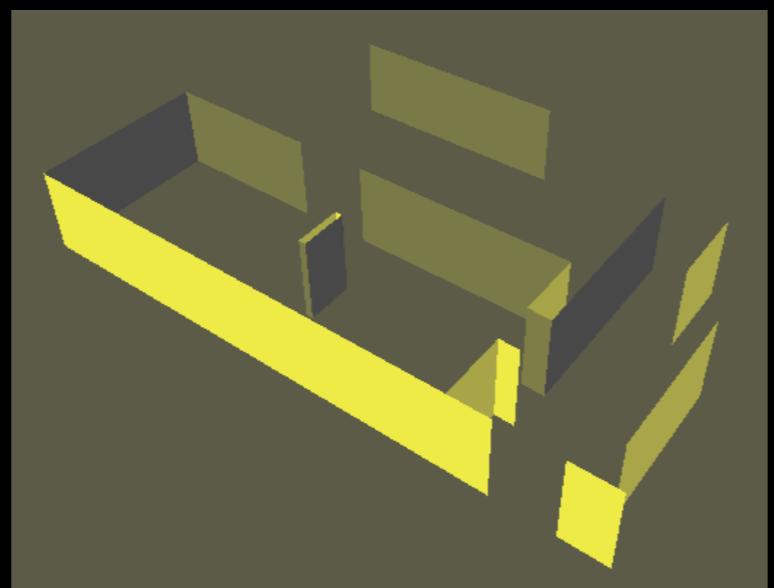
floor samples

wall samples

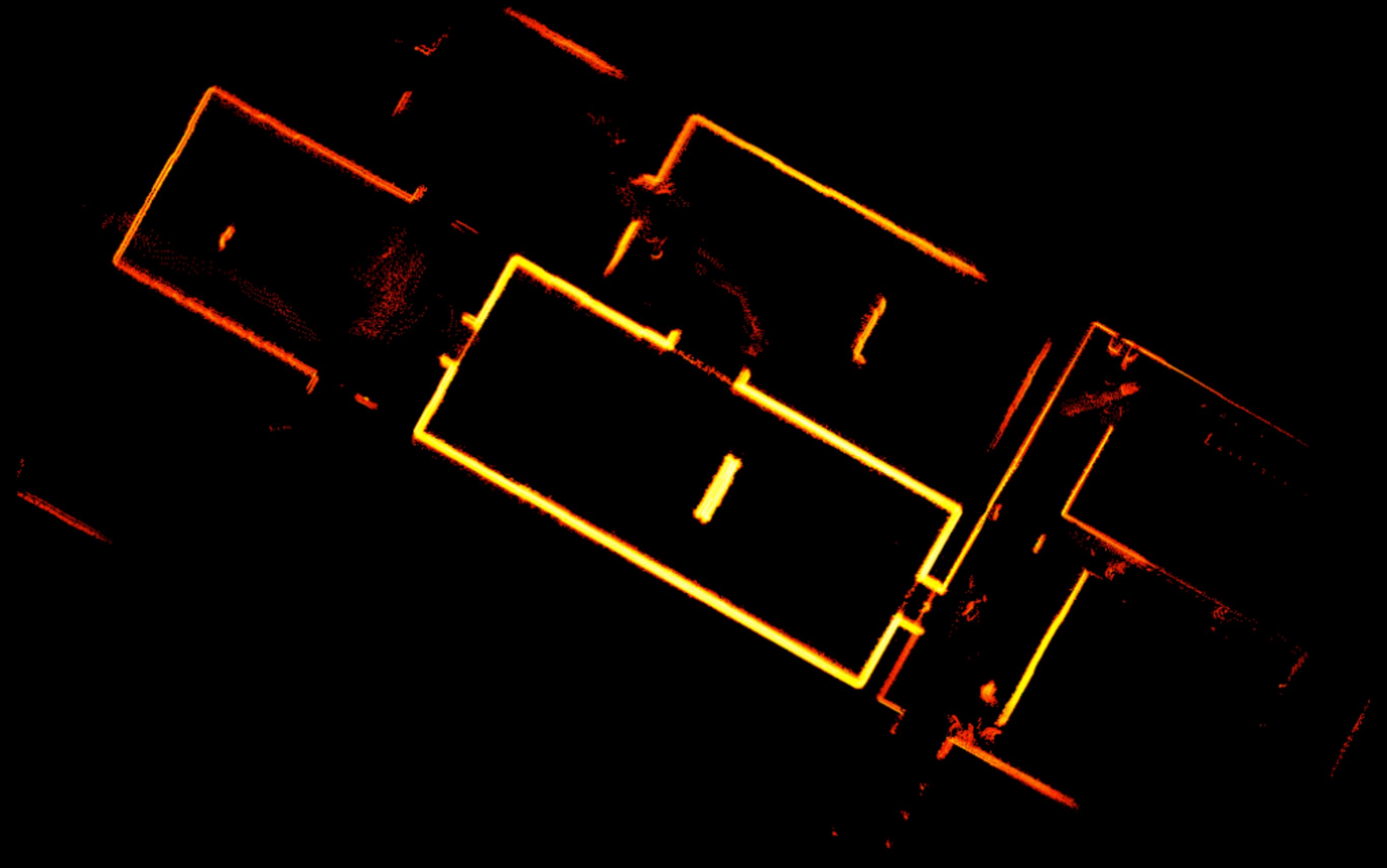
ceiling samples

wall geometry

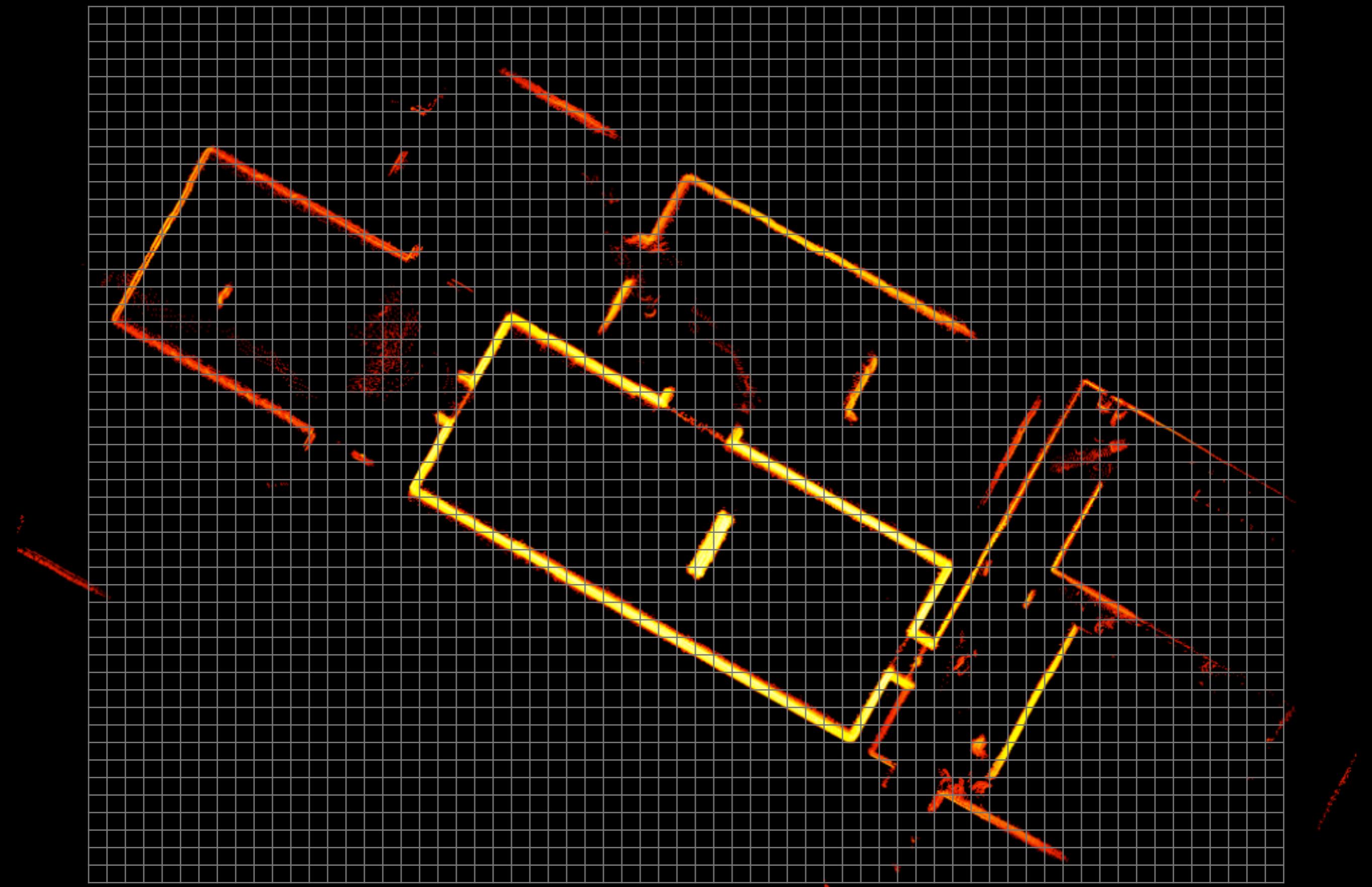
ceiling height map



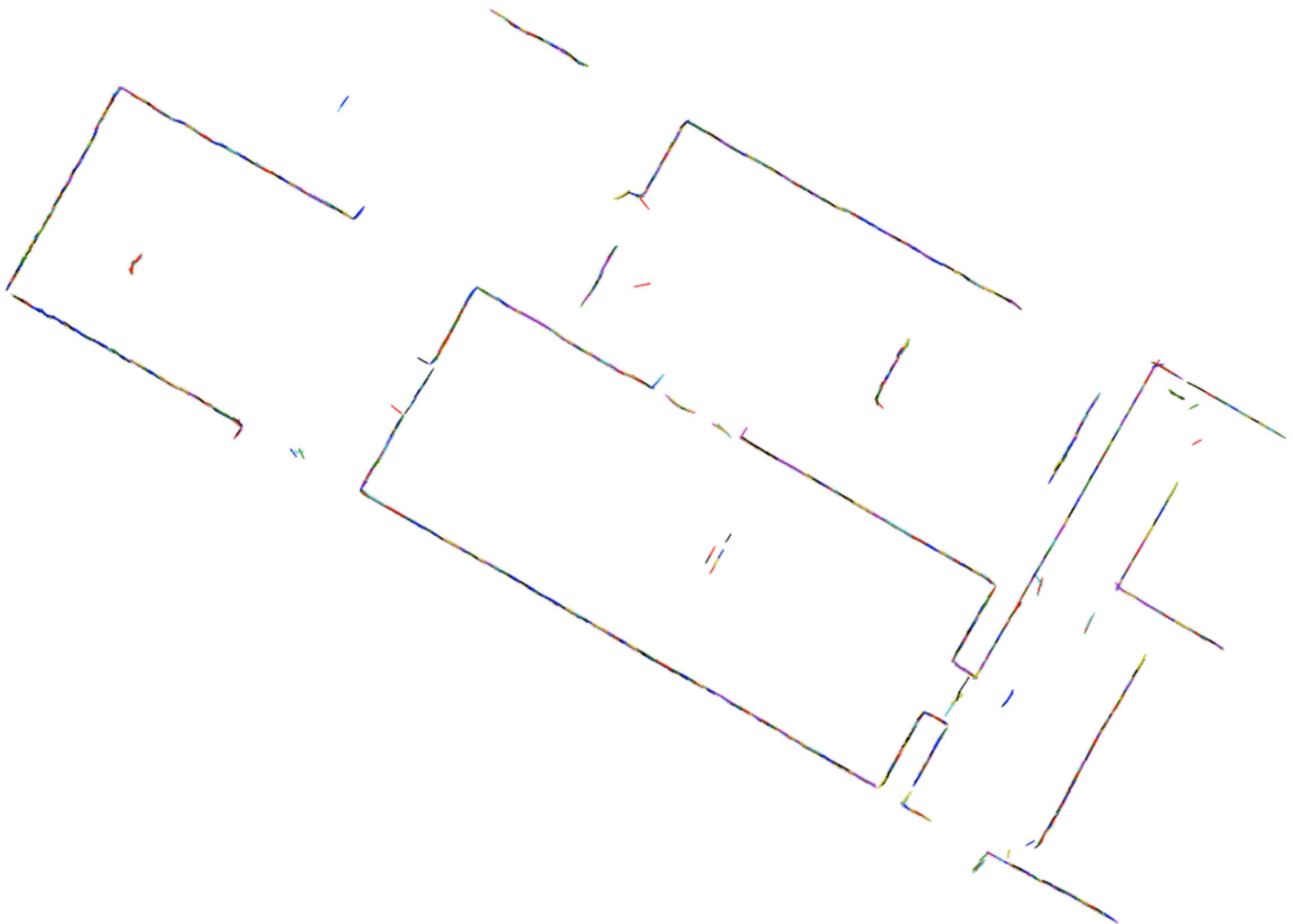
## ② Wall geometry reconstruction



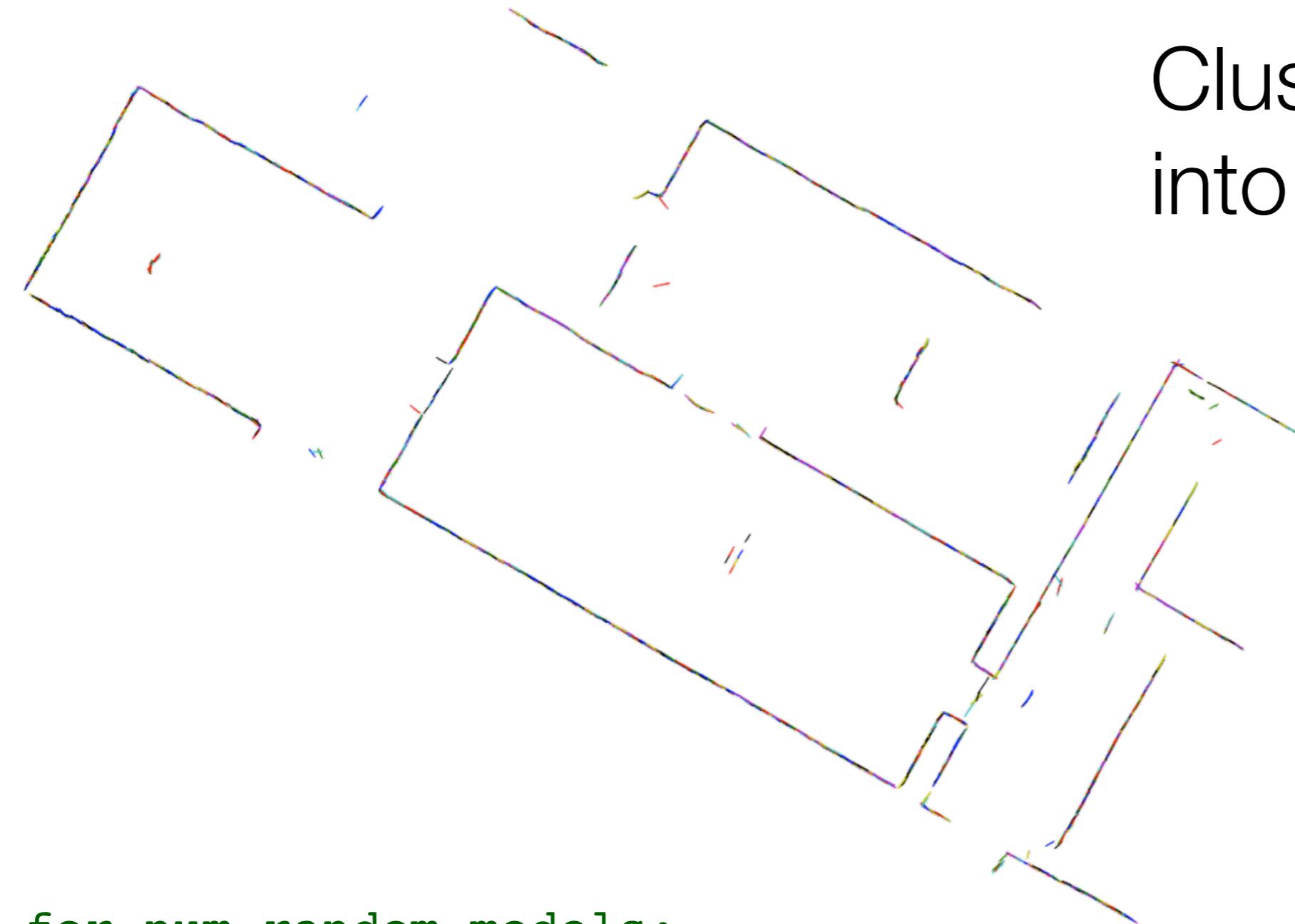
Find the best fit line model for each grid cell



# Sparse, oriented “edgels”



# Clustering edgels into wall segments



```
for num_random_models:  
    model <- line model implied by a random sparse wall edgel  
    inliers <- empty set  
    do:  
        inliers <- set of sparse wall edgels that fit this model  
        model <- best-fit line to inliers  
    while inlier size increases
```

# Clustering edgels into wall segments



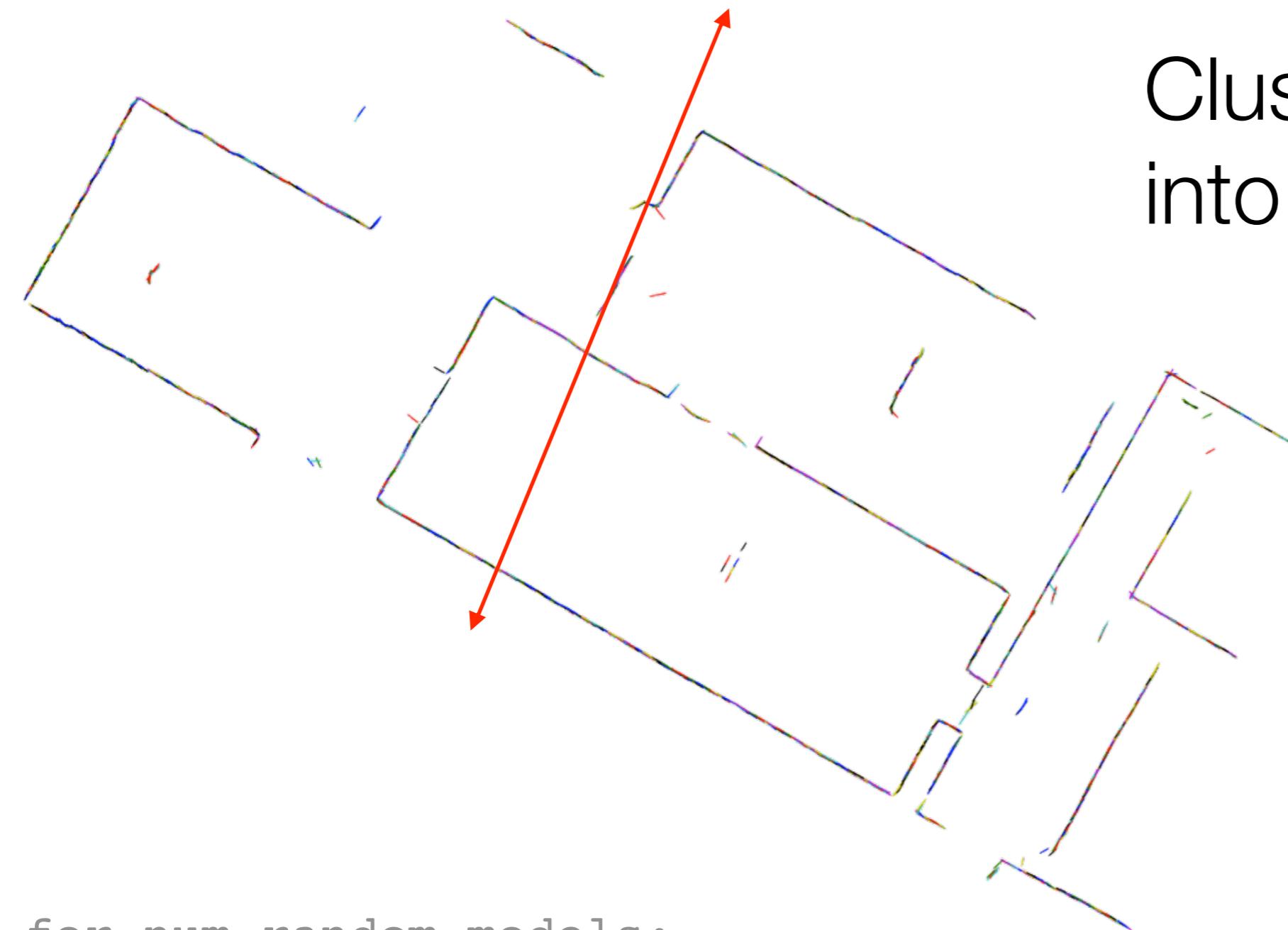
```
for num_random_models:  
    model <- line model implied by a random sparse wall edgel  
    inliers <- empty set  
    do:  
        inliers <- set of sparse wall edgels that fit this model  
        model <- best-fit line to inliers  
    while inlier size increases
```

# Clustering edgels into wall segments



```
for num_random_models:  
    model <- line model implied by a random sparse wall edgel  
    inliers <- empty set  
    do:  
        inliers <- set of sparse wall edgels that fit this model  
        model <- best-fit line to inliers  
    while inlier size increases
```

# Clustering edgels into wall segments



```
for num_random_models:  
    model <- line model implied by a random sparse wall edgel  
    inliers <- empty set  
    do:  
        inliers <- set of sparse wall edgels that fit this model  
        model <- best-fit line to inliers  
    while inlier size increases
```

# Clustering edgels into wall segments



```
for num_random_models:  
    model <- line model implied by a random sparse wall edgel  
    inliers <- empty set  
    do:  
        inliers <- set of sparse wall edgels that fit this model  
        model <- best-fit line to inliers  
    while inlier size increases
```

# Clustering edgels into wall segments



```
for num_random_models:  
    model <- line model implied by a random sparse wall edgel  
    inliers <- empty set  
do:  
    inliers <- set of sparse wall edgels that fit this model  
    model <- best-fit line to inliers  
while inlier size increases
```

# Clustering edgels into wall segments



```
for num_random_models:  
    model <- line model implied by a random sparse wall edgel  
    inliers <- empty set  
    do:  
        inliers <- set of sparse wall edgels that fit this model  
        model <- best-fit line to inliers  
    while inlier size increases
```

# Clustering edgels into wall segments

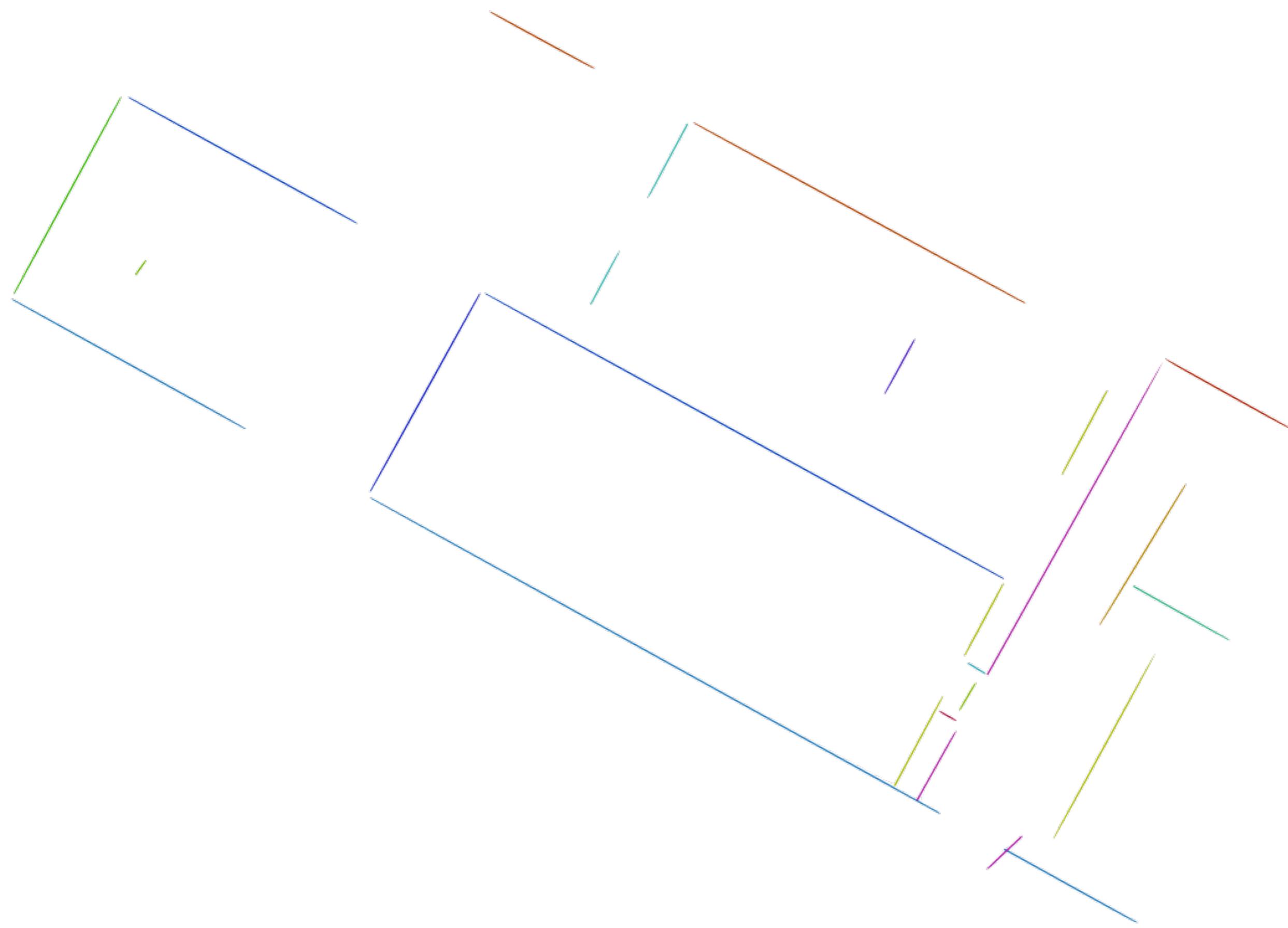


```
for num_random_models:  
    model <- line model implied by a random sparse wall edgel  
    inliers <- empty set  
    do:  
        inliers <- set of sparse wall edgels that fit this model  
        model <- best-fit line to inliers  
    while inlier size increases
```

# Clustered edgels

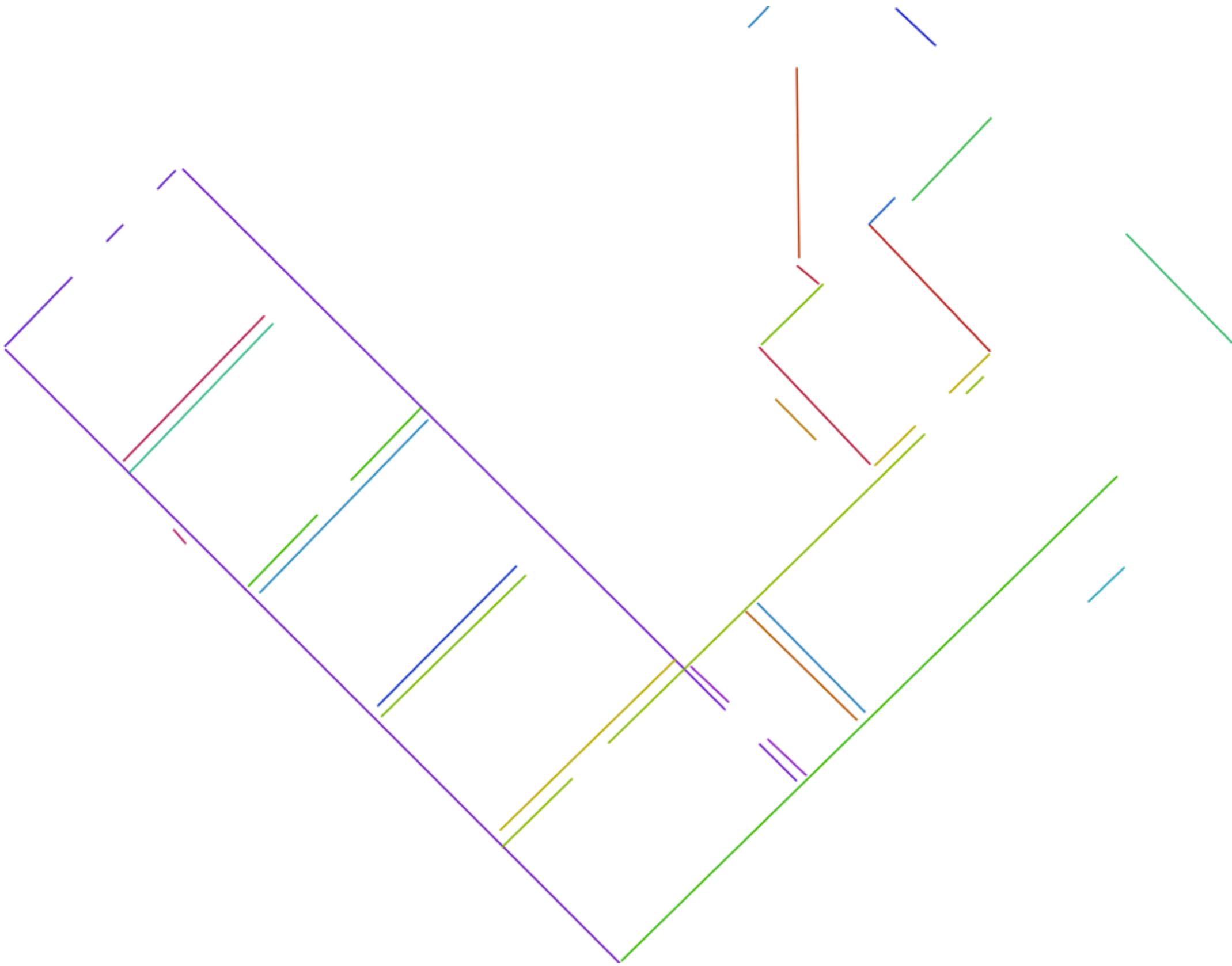


# Wall segments colored by cluster

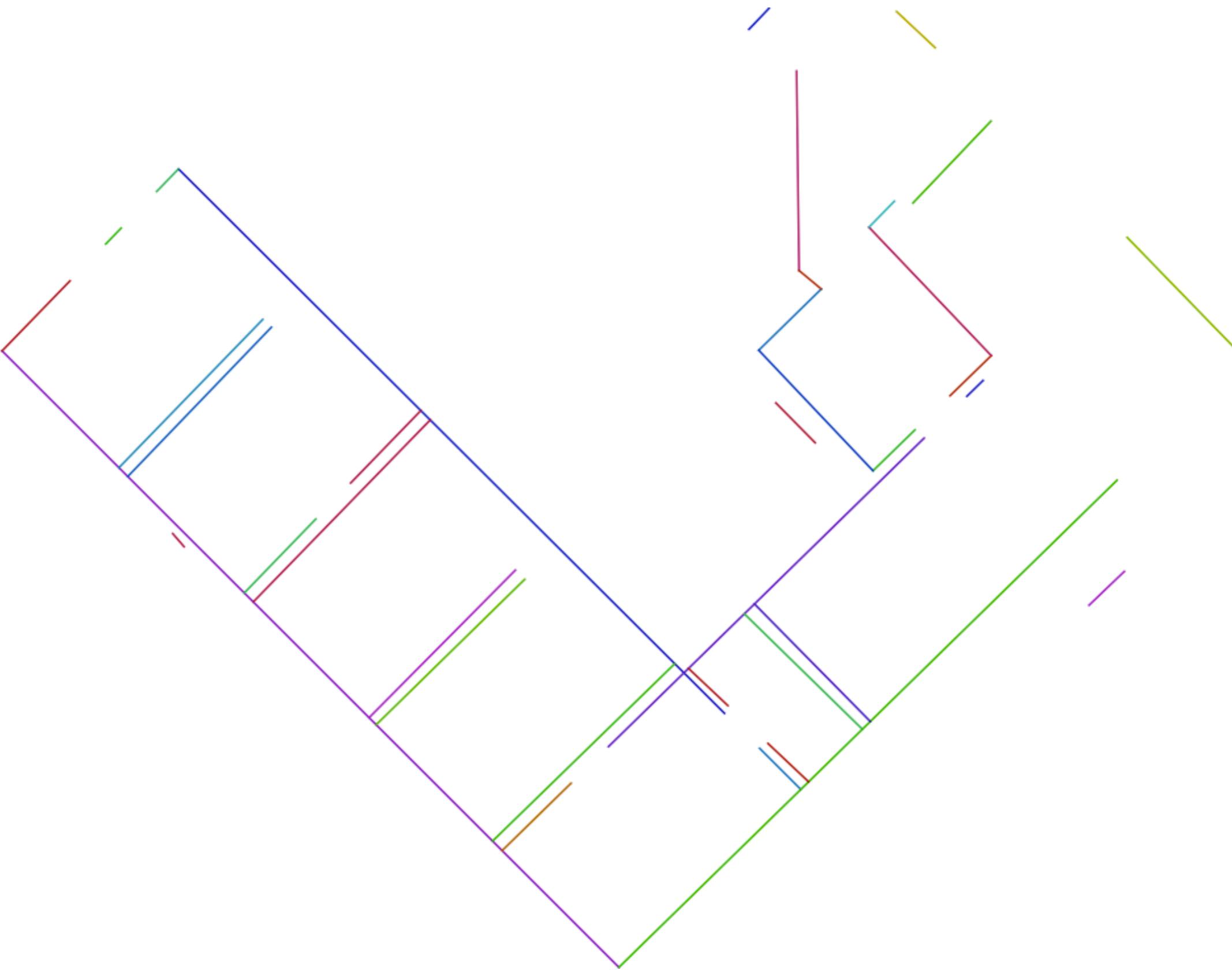


③

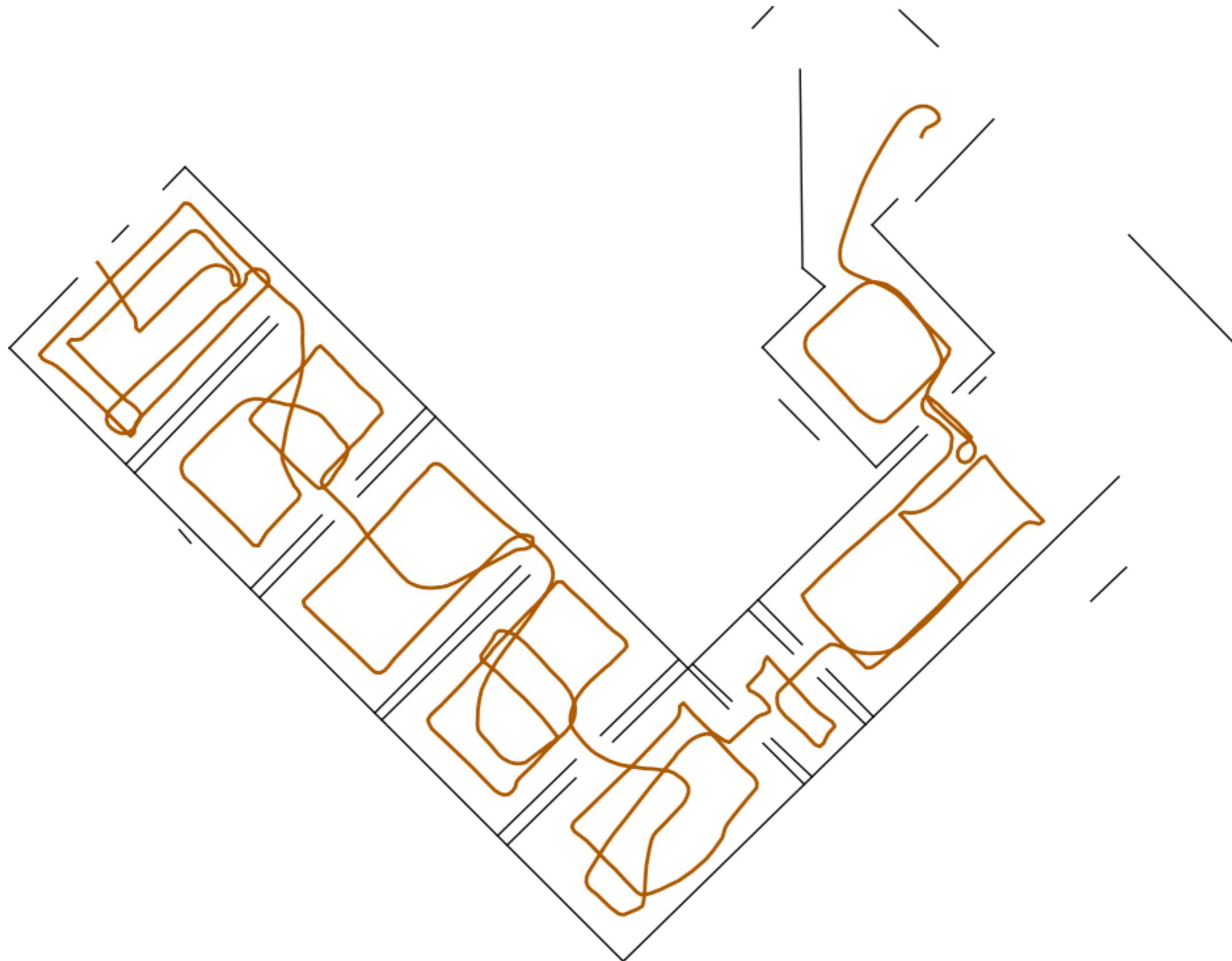
# Post processing



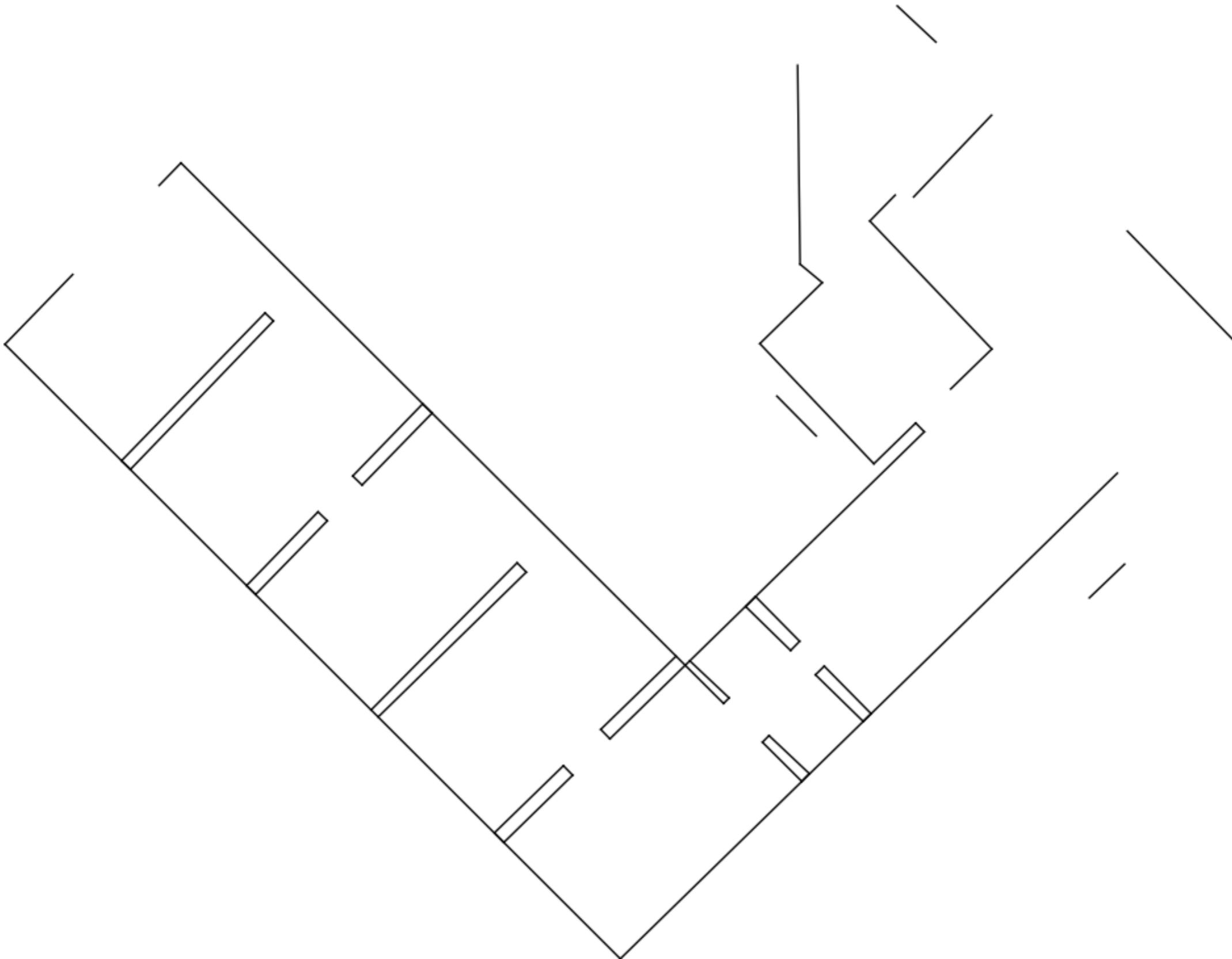
# Close gaps



# Clip against trolley path

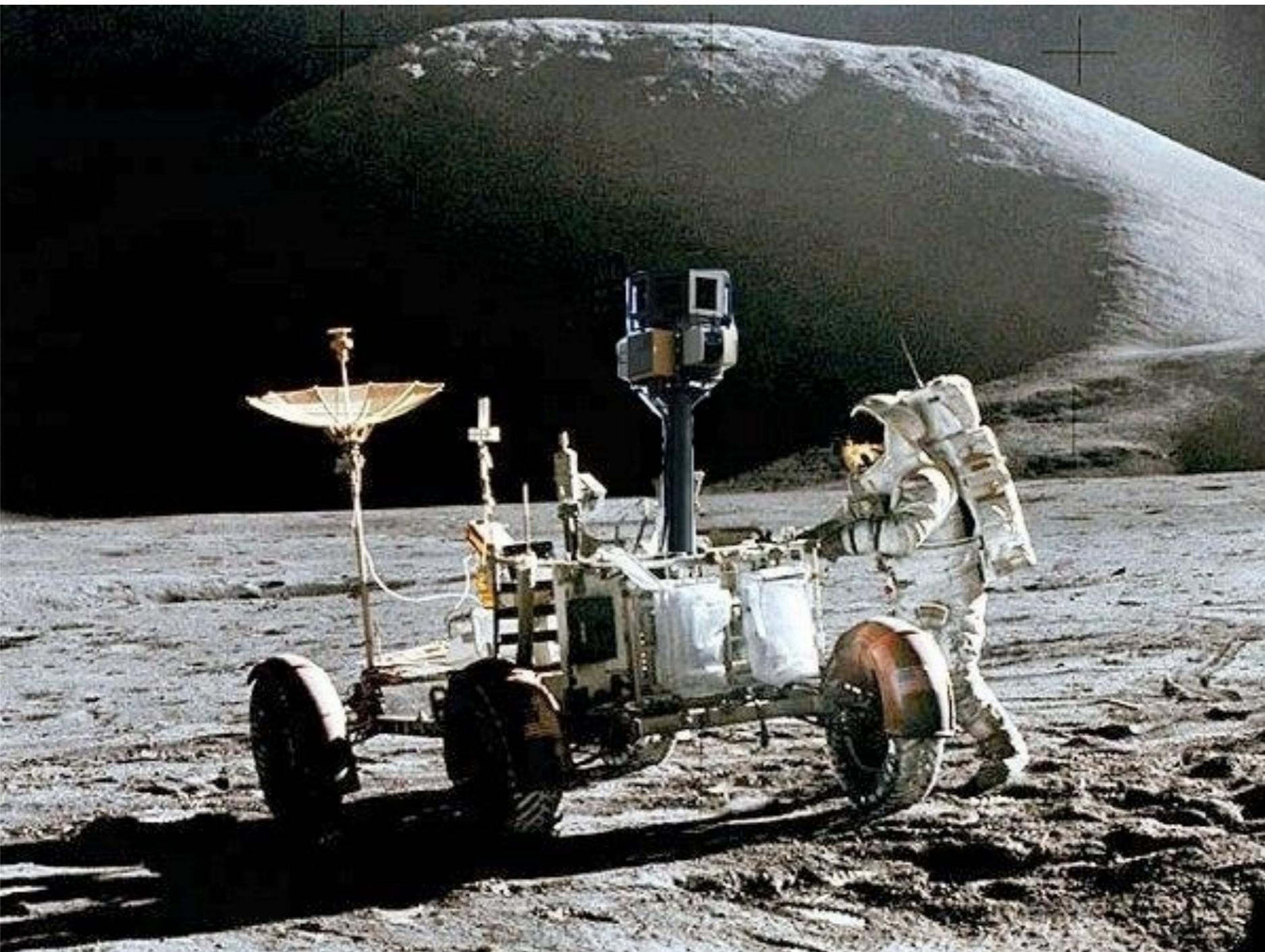


# “Heal” end caps



# Final product

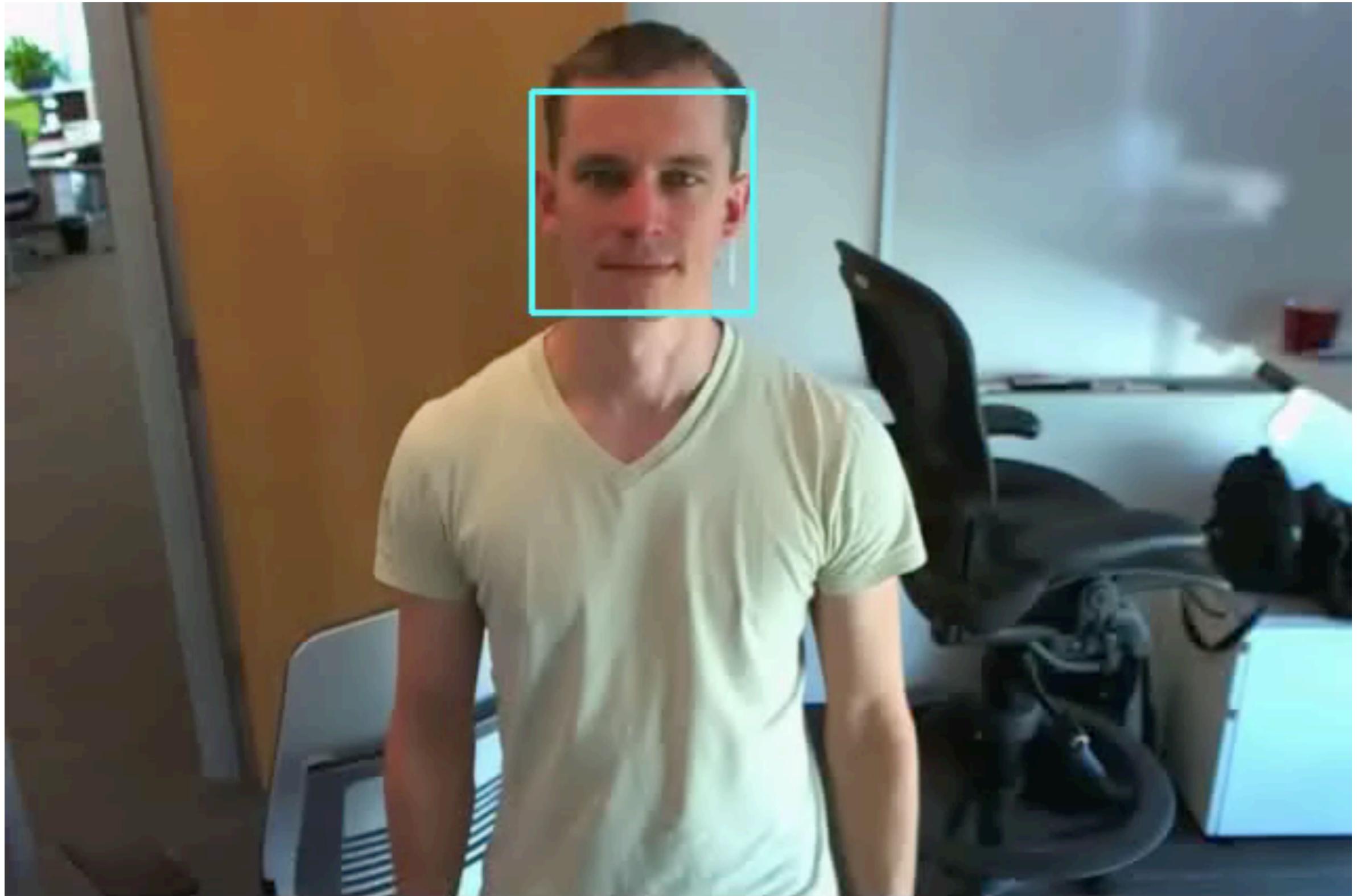
# Still lots to do! Where next?



Next Thursday: CEO of Lynx Labs  
Chris Slaughter

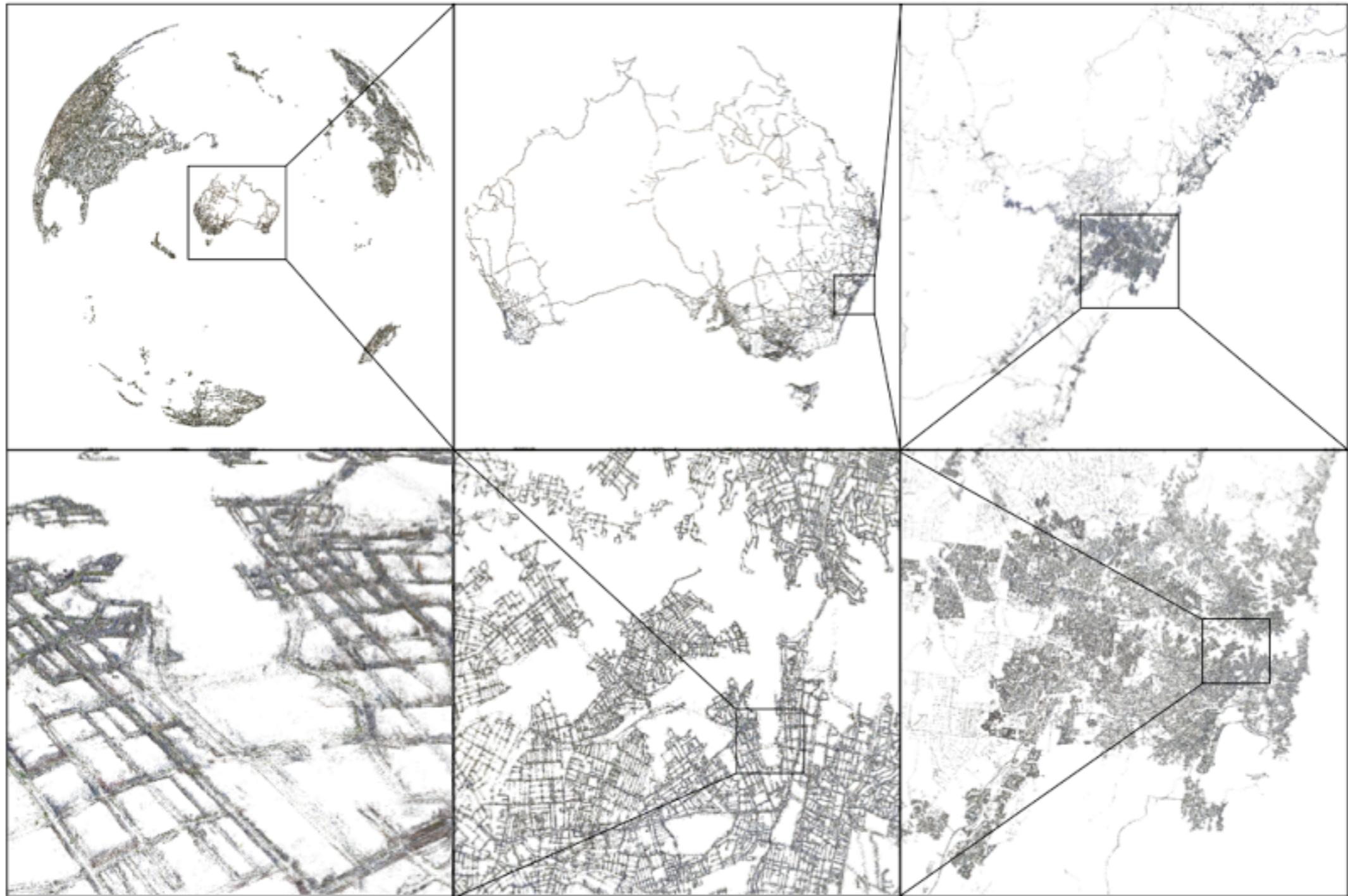


# Project 3: Tracking



- **Due 4 November, 11:59pm**
- **Next week office hours: Monday, 27 Oct 5-6pm**

# Final Project



- **Groups and proposals due Next Tuesday 28 Oct**