

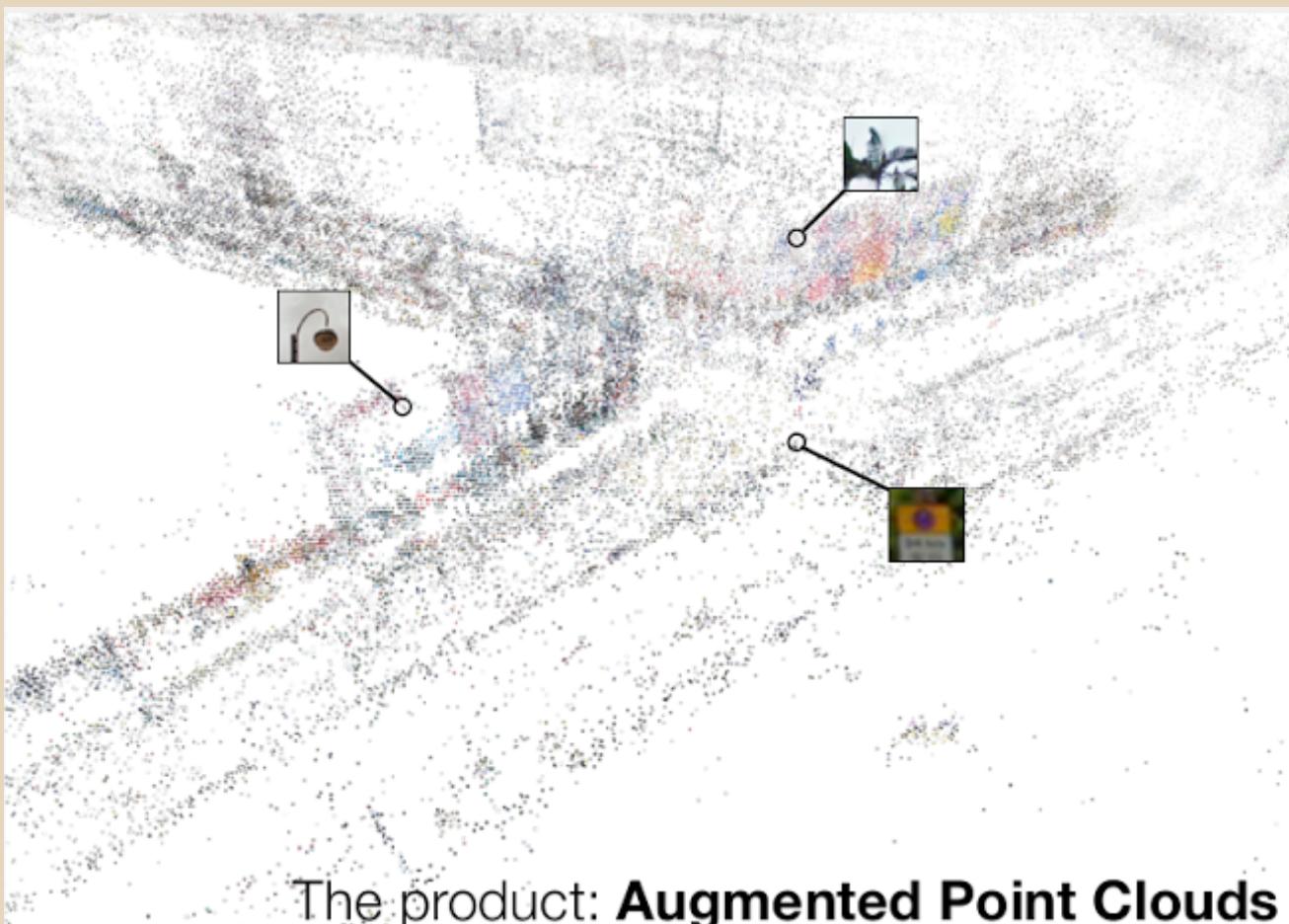
Interpreting Image Camera Location from Known 3D Models

December 2nd 2014

Nicholas Walther
Michelle Lawrence
Lyee Chong

Inspiration

Oct 2nd Structure from Motion lecture and question asked by Matt B.



Original Objectives

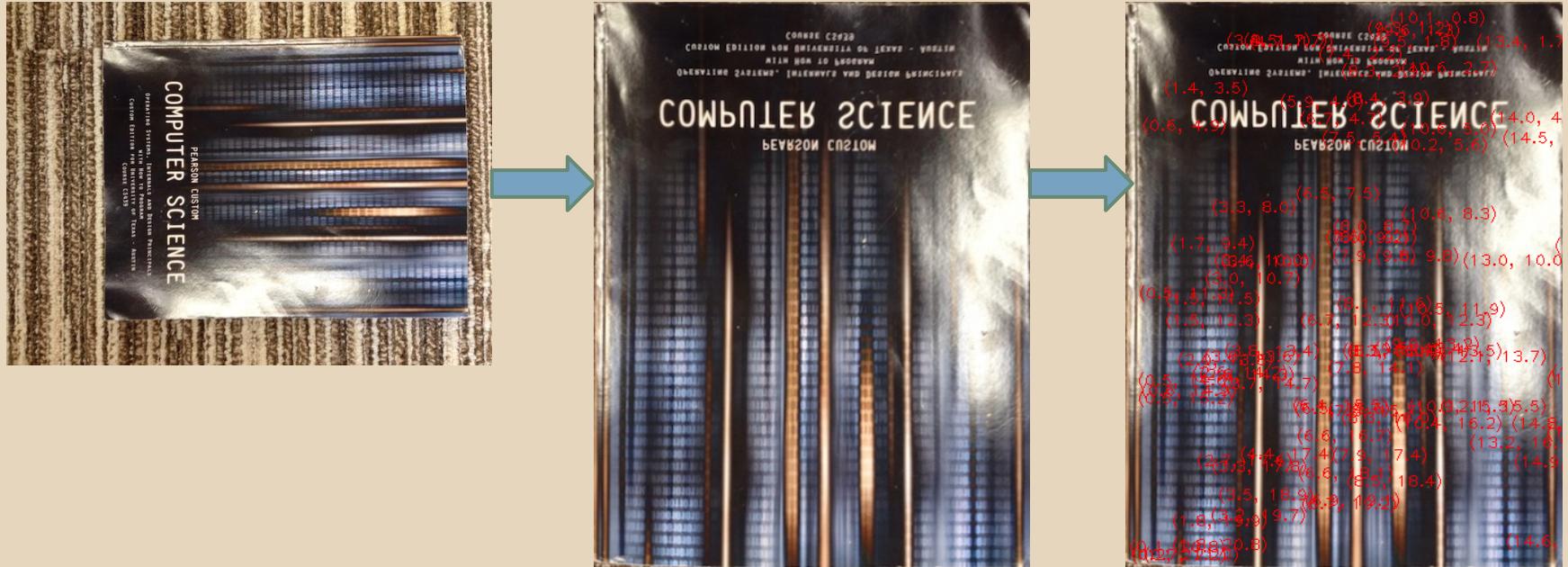
- Obtain small point cloud of Austin from Google's street view data and matching panorama images
- Match features in images to point cloud data to produce augmented point cloud
- Using images of Austin, find features with known locations and triangulate camera locations
- Real time camera triangulation

Revised Objectives

- Create two synthetic scenes (simple and complex)
- Create point clouds using measured features from both scenes
- Match features to measured locations in both scenes
- Triangulate camera location

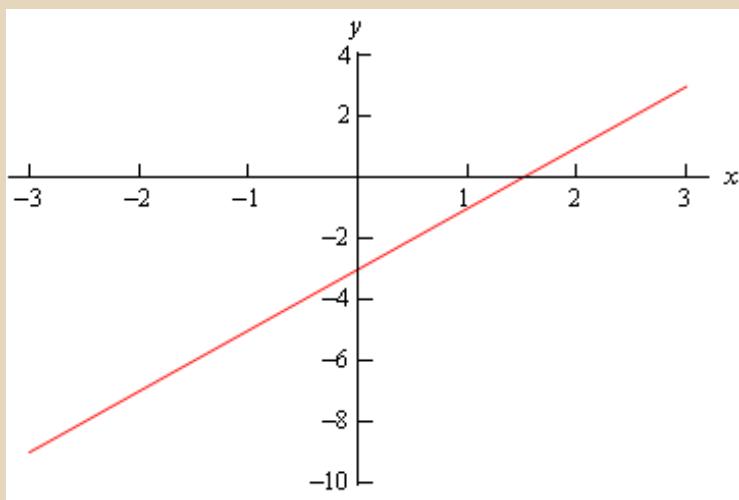
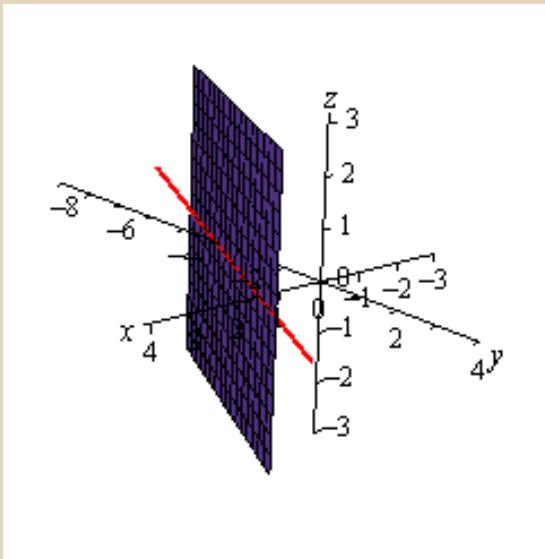


Creating Ground Truth Process

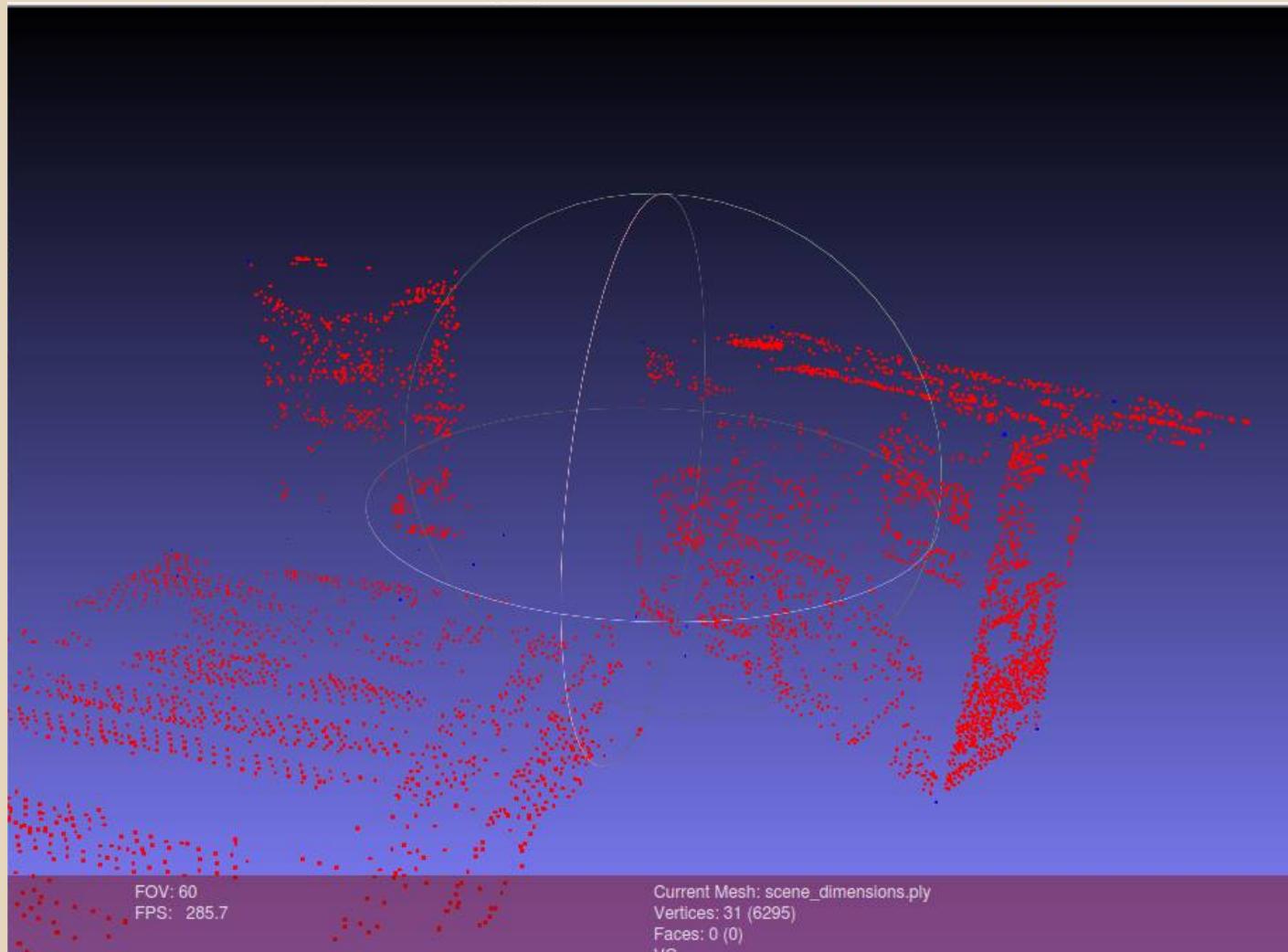


```
point_pixels = [(132, 53), (134, 437), (634, 433), (631, 46)]
point_locations = [(0, 0), (16.8, 0), (16.8, 21.4), (0, 21.4)]
map, kp_list, desc_list = cl.map_features(image, point_pixels, point_locations)
```

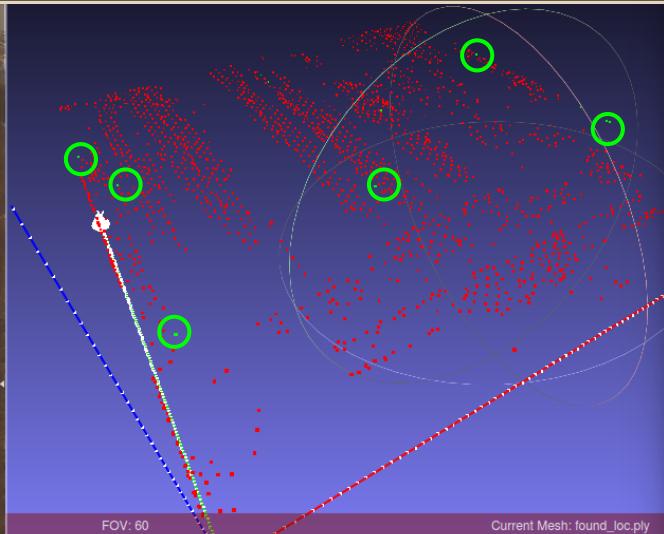
Creating Ground Truth Challenges



Creating Ground Truth Product



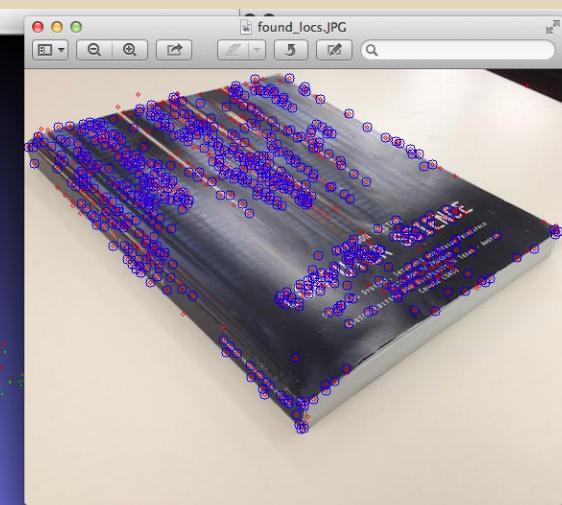
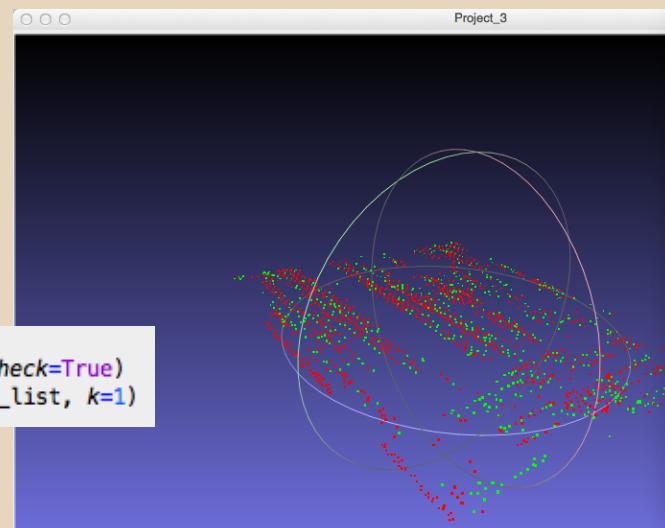
Feature Matching



← BFMatcher +
distance filter

BFMatcher + →
cross validation +
k = 1

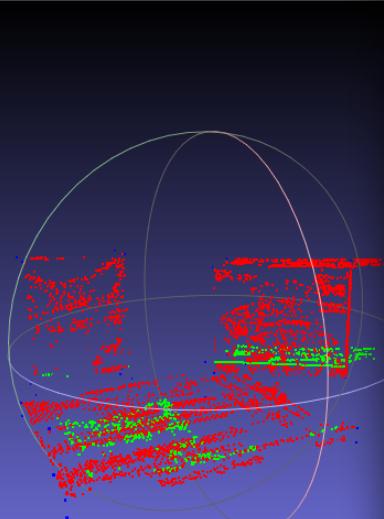
```
# Match descriptors.  
bf = cv2.BFMatcher(normType=cv2.NORM_L1, crossCheck=True)  
matches = bf.knnMatch(img_desc_list, known_desc_list, k=1)
```



```
/Users/Michelle/comp_vision_project_4/project_4/solve  
|   book_binding.ply In 6 msec  
all images loaded in 6 msec  
Opened mesh  
/Users/Michelle/comp_vision_project_4/project_4/solve
```

Feature Matching

Matches for the poptart box →



Current Mesh: found_loc_poptart.ply
Vertices: 1650 (9509)
Faces: 0 (0)
VC

```
open3d match
Users/Michelle/comp_vision_project_4/project_4/foun
d_loc_cars_box.ply in 24 msec
All files opened in 24 msec
Open3d match
Users/Michelle/comp_vision_project_4/project_4/foun
d_loc_poptart.ply in 316 msec
All files opened in 317 msec
```



← Feature matching in complex scene

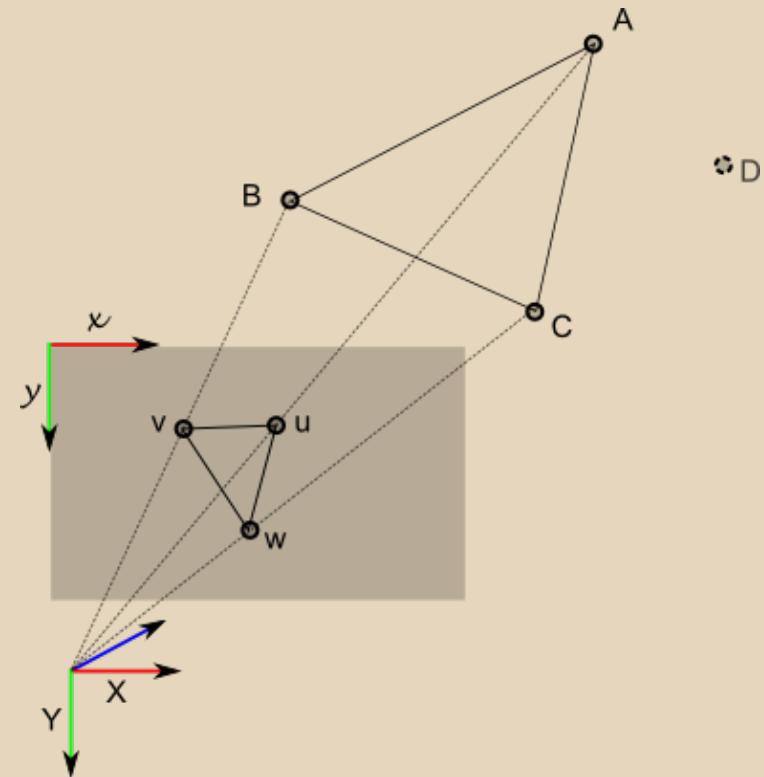
So much noise....

Triangulation

Locating the camera position from a set of known feature locations.

perspective-3-point-problem
“P3P”

perspective-n-point-problem
“PnP”



Triangulation



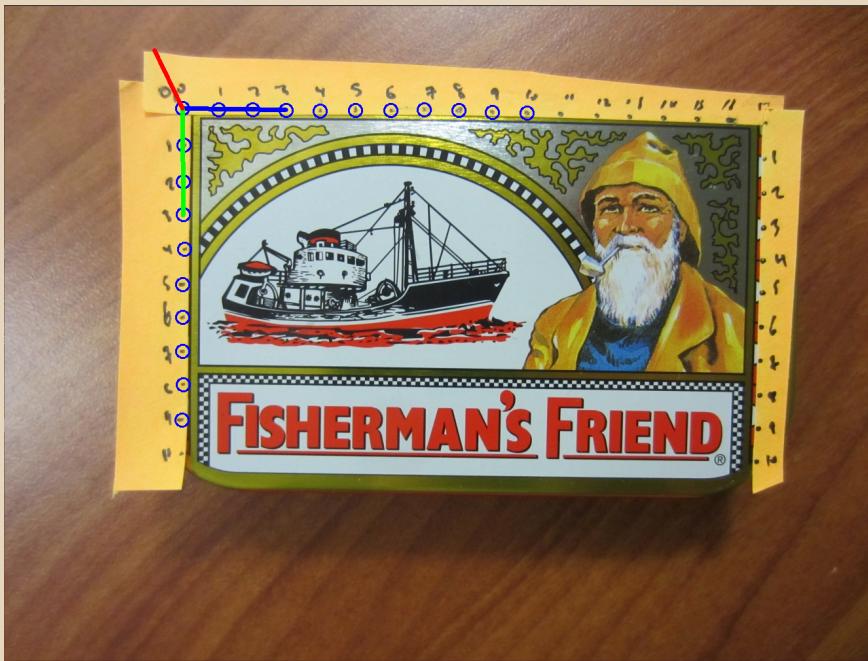
Triangulation



Triangulation



Triangulation

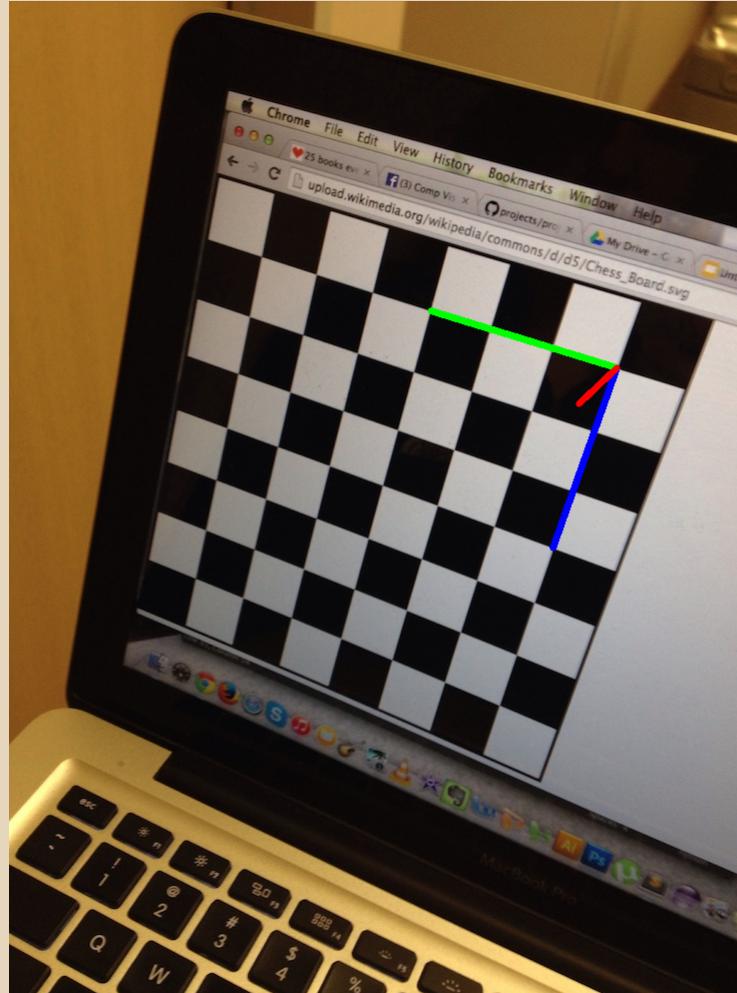


Rodrigues' rotation formula

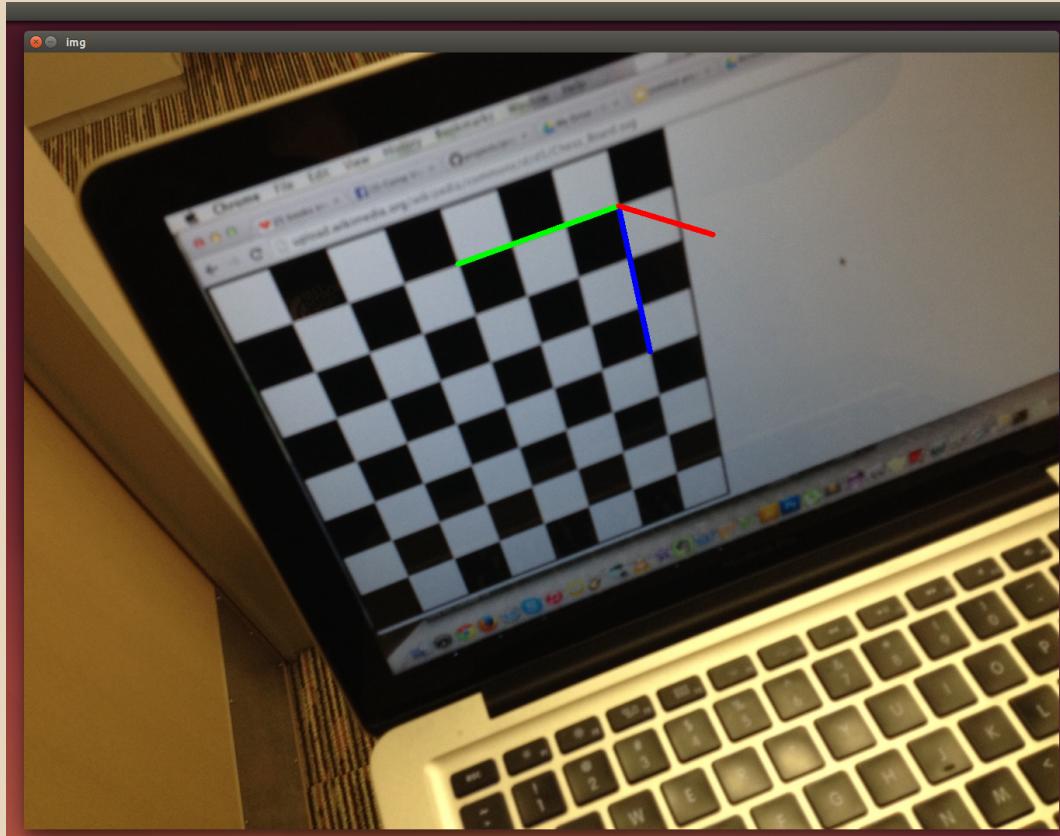
$$R = [[0.03137613], [0.0409094], [0.01422577]]$$

$$t = [[-1.21201553], [-3.18248486], [8.26400771]]$$

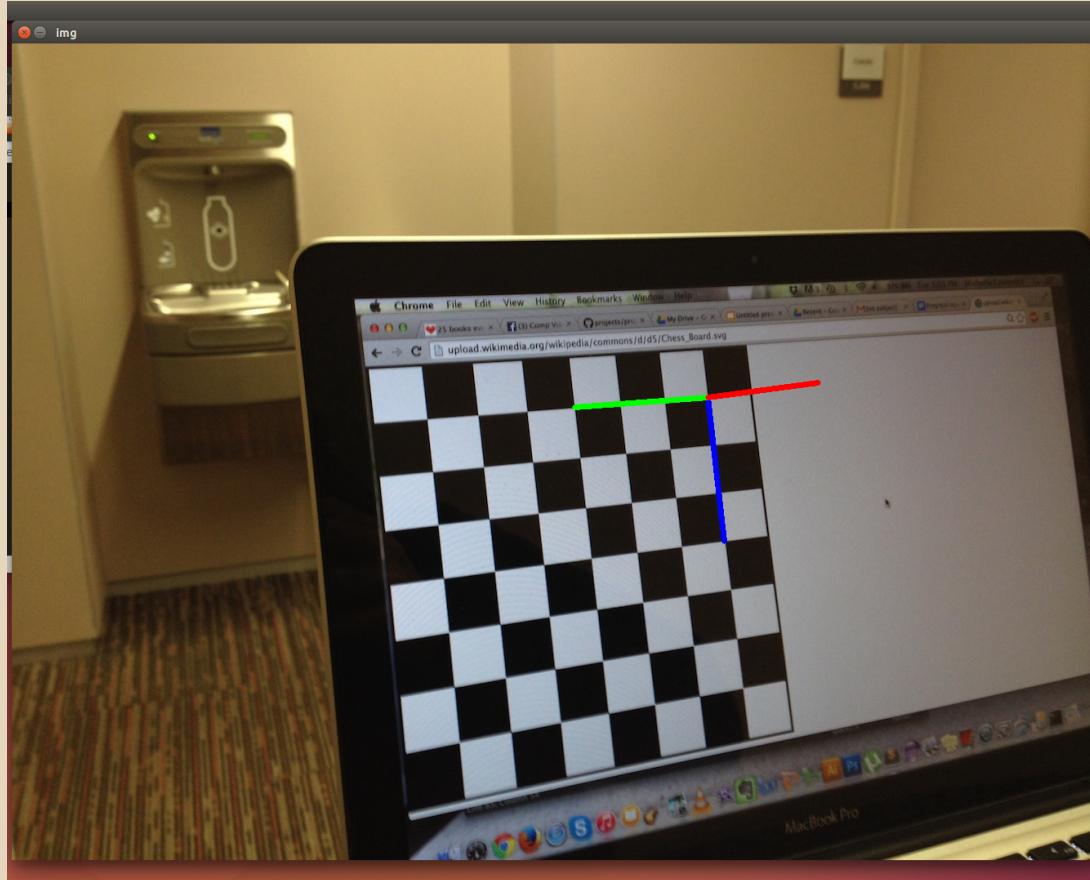
Triangulation



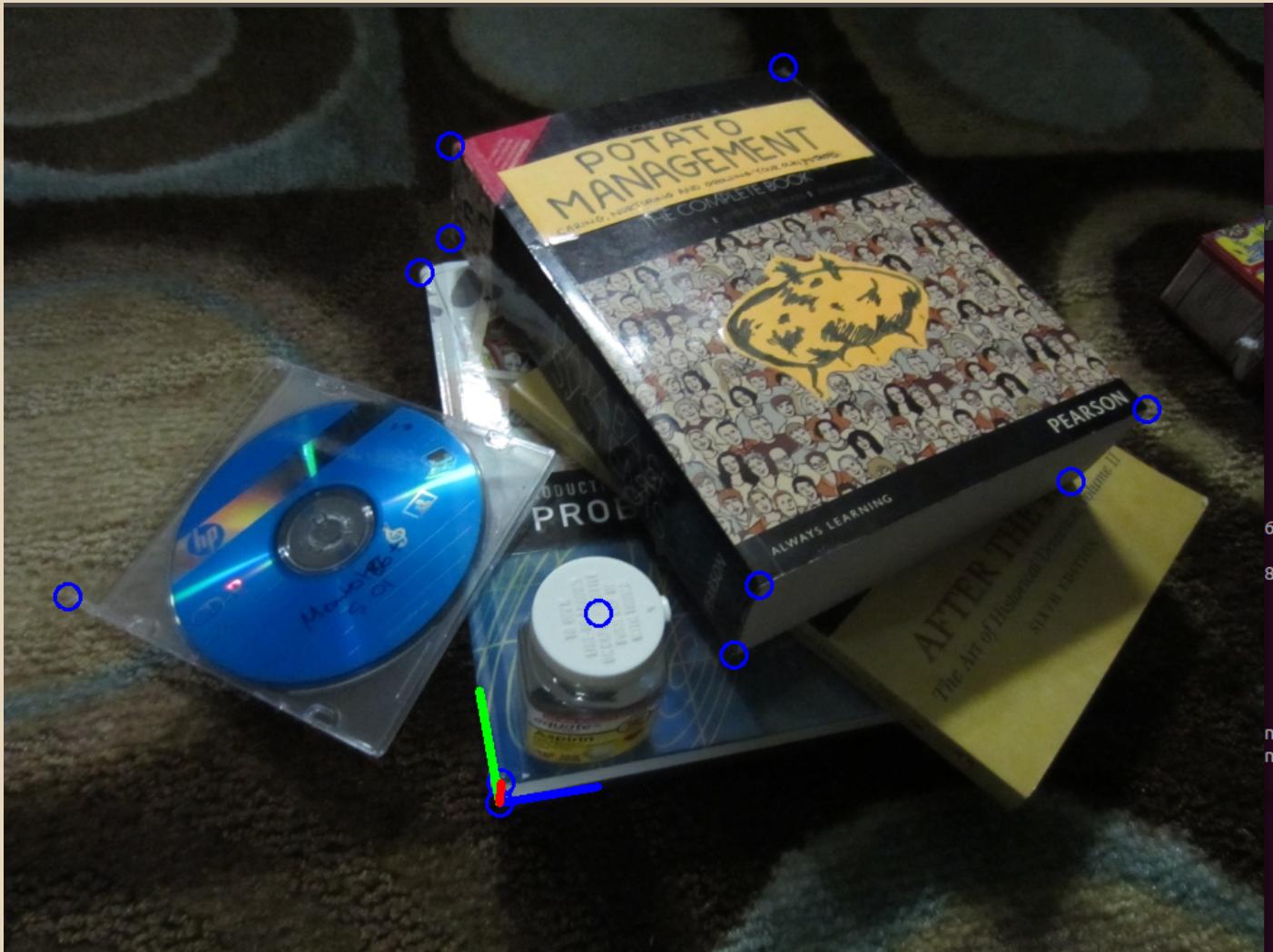
Triangulation



Triangulation



Triangulation



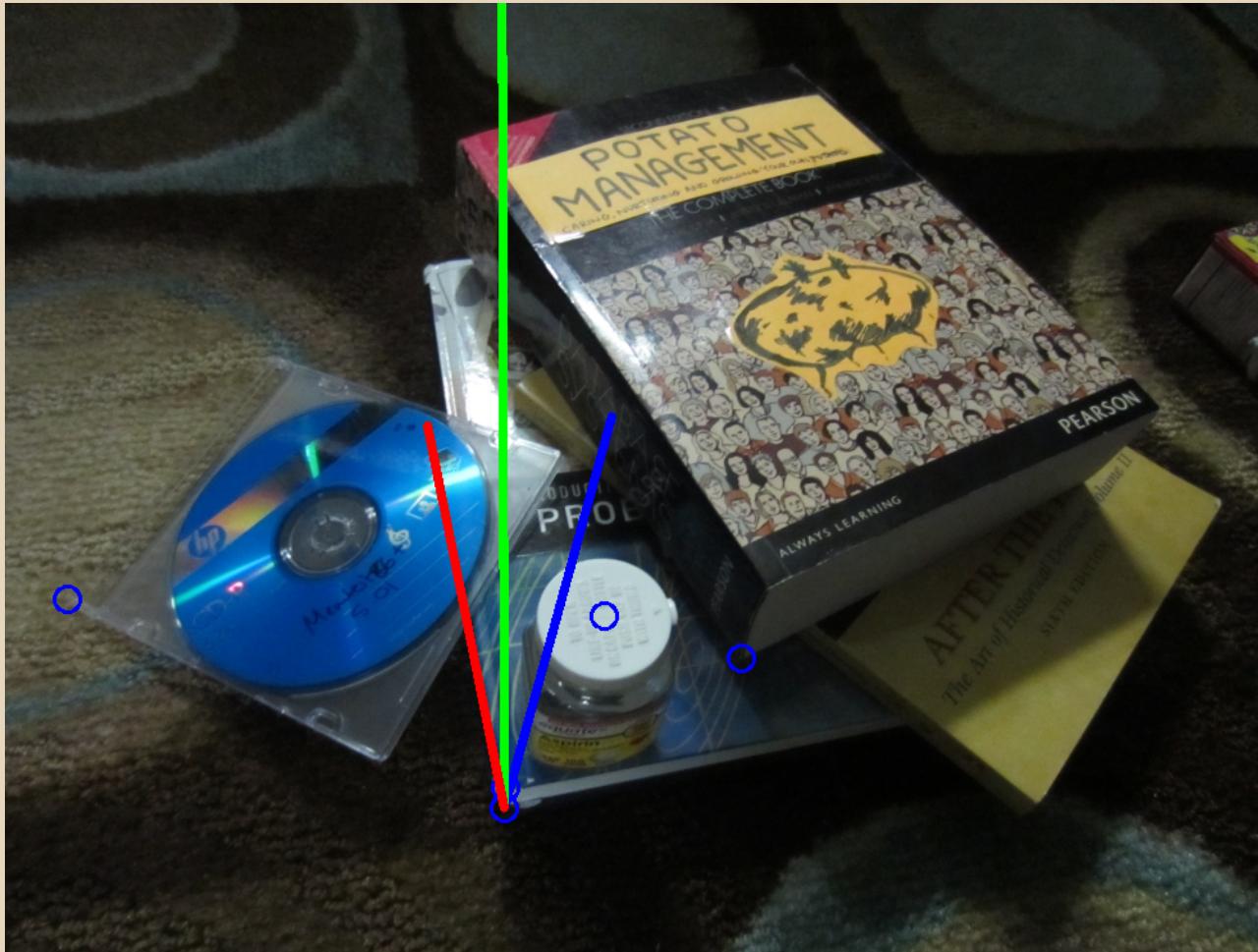
Triangulation Challenges

Camera calibration

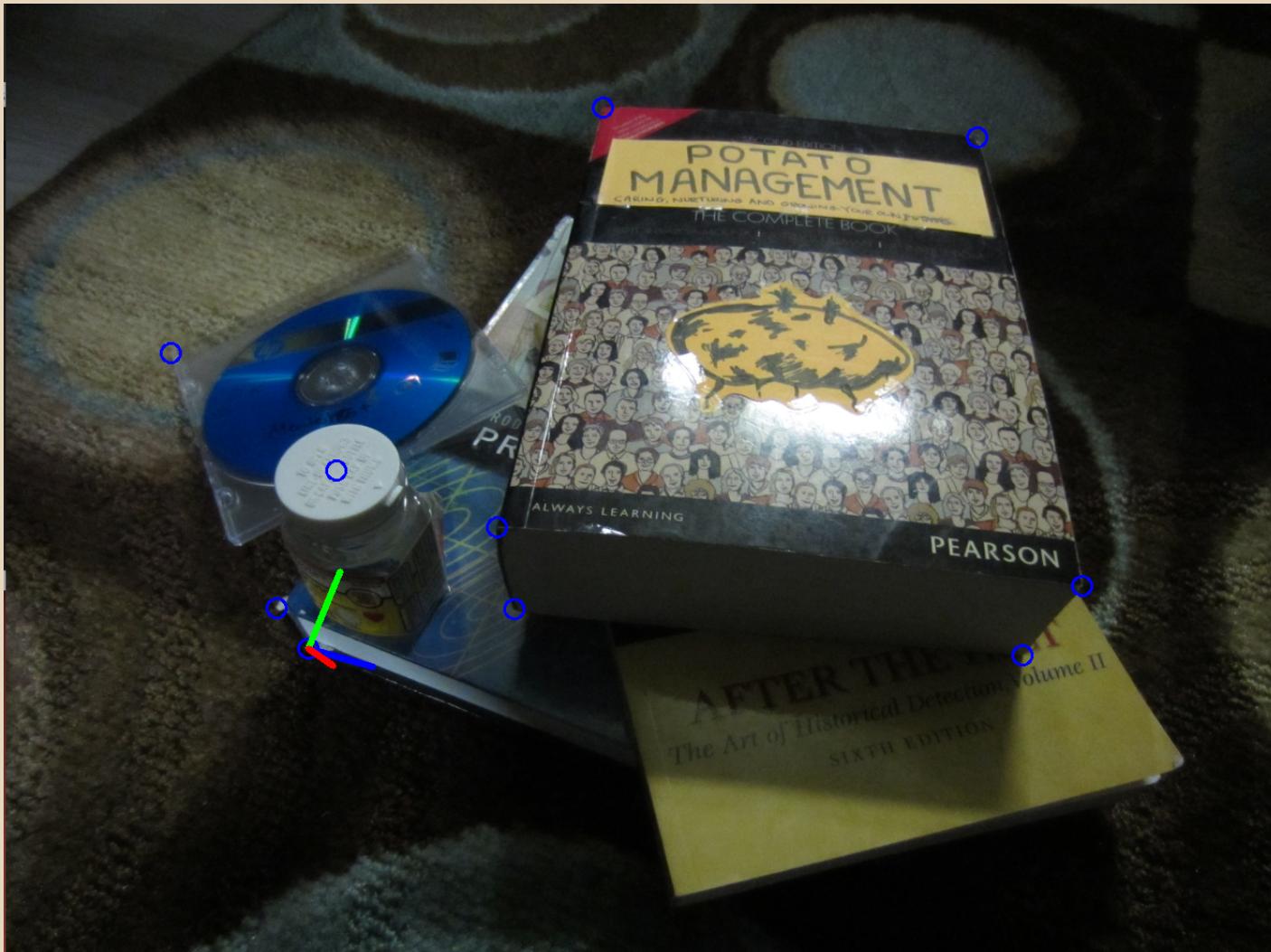
- distortion

Reliance on many features

Triangulation Challenges



Triangulation Challenges



Questions?