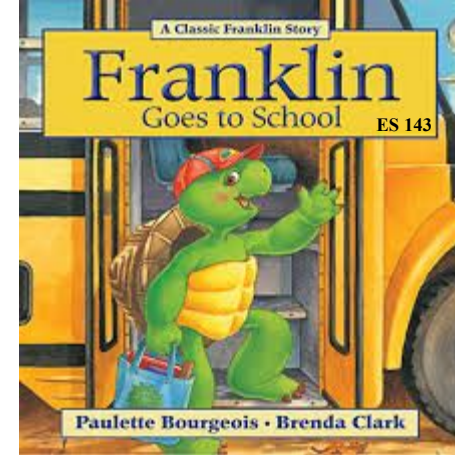


Team Turtle

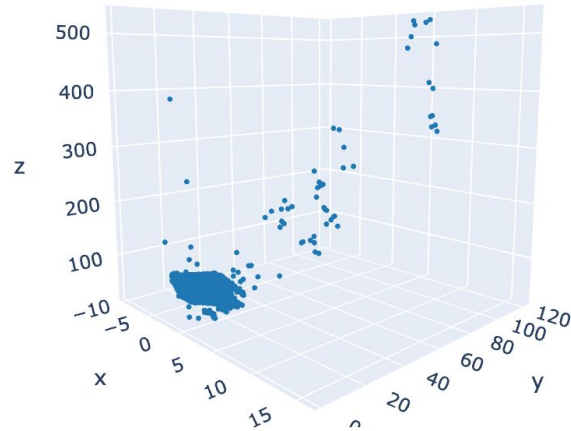


Dylan, Wisoo,
Kaihong, Arnav

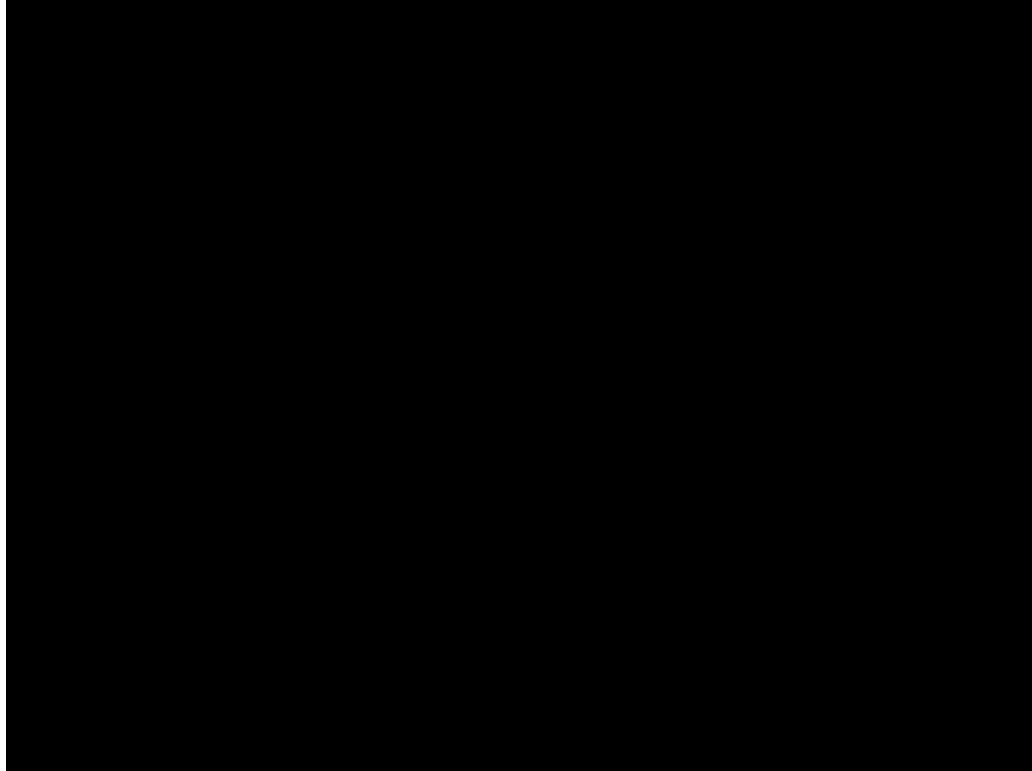
Turtle Results

Link:

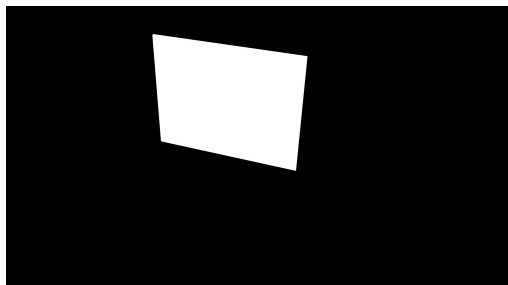
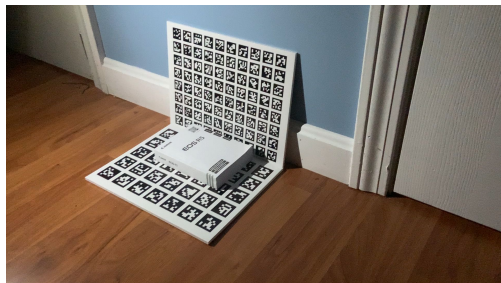
https://deepnote.com/project/6c8403a5-0093-4399-b6a6-d4b2f6663e50#%2FHarvard_3d_scanner_team_turtle%2Fswept_plane.ipynb



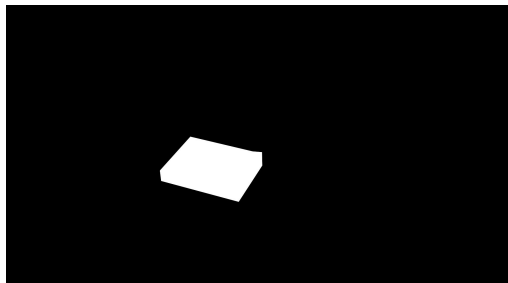
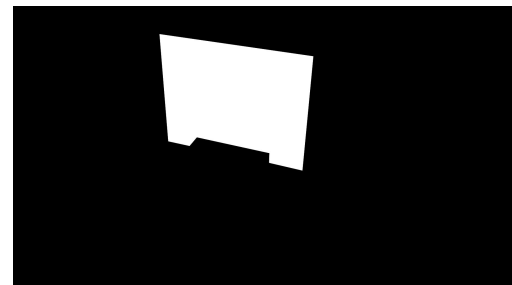
Turtle Video



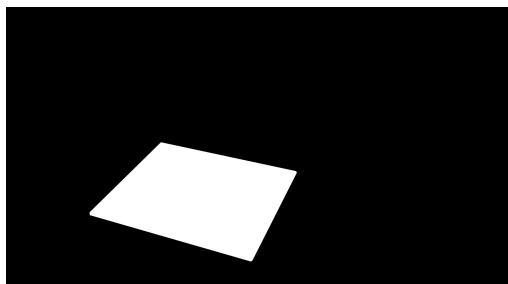
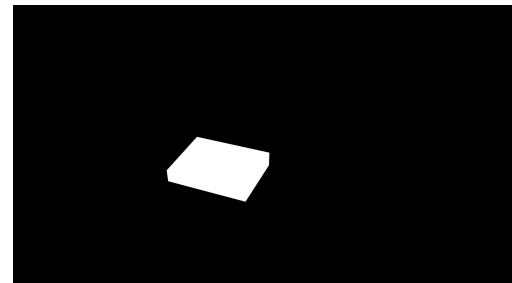
Masks



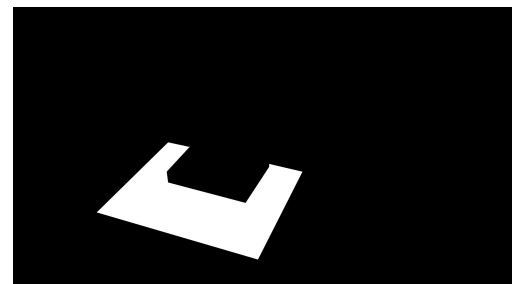
“The top board”



“The book”



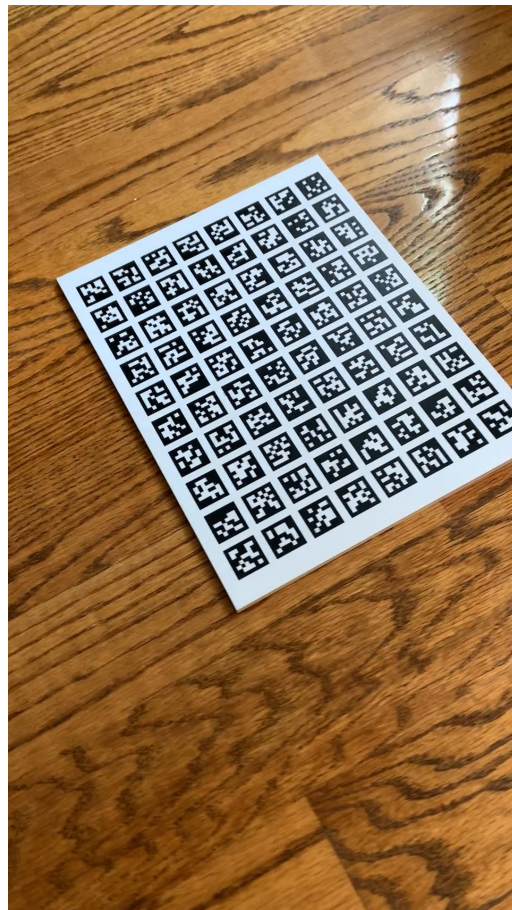
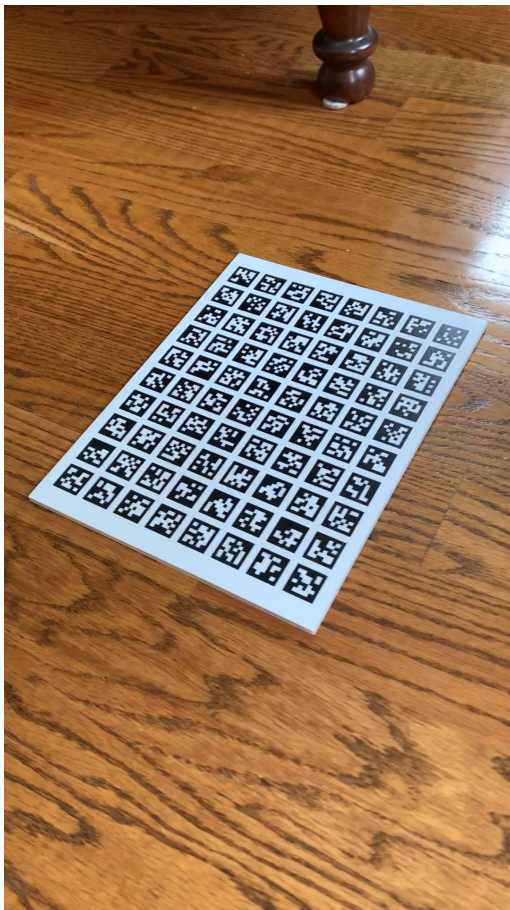
“The bottom board”



CalibrateCamera()

Connecting 3d points in world coordinate space to 2d points on the image plane

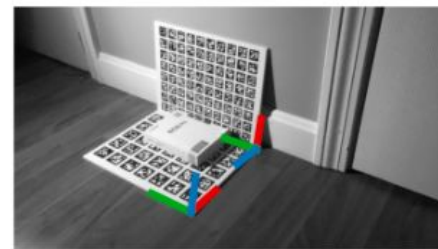
- create a linear system of boards to camera matrix K



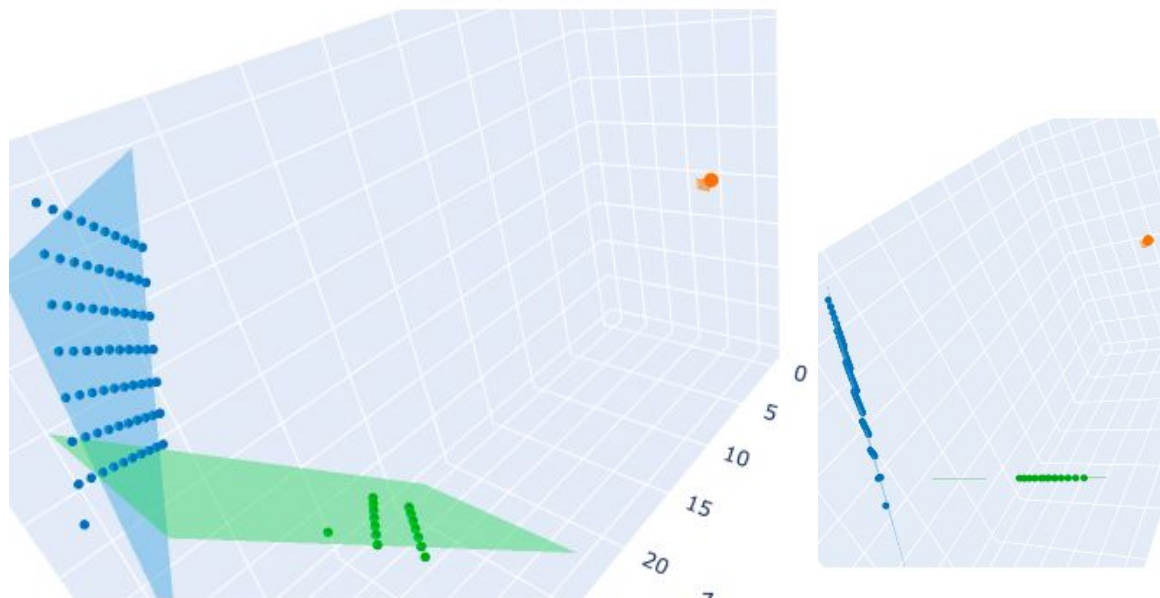
computeTwoPlanes()

For each board,

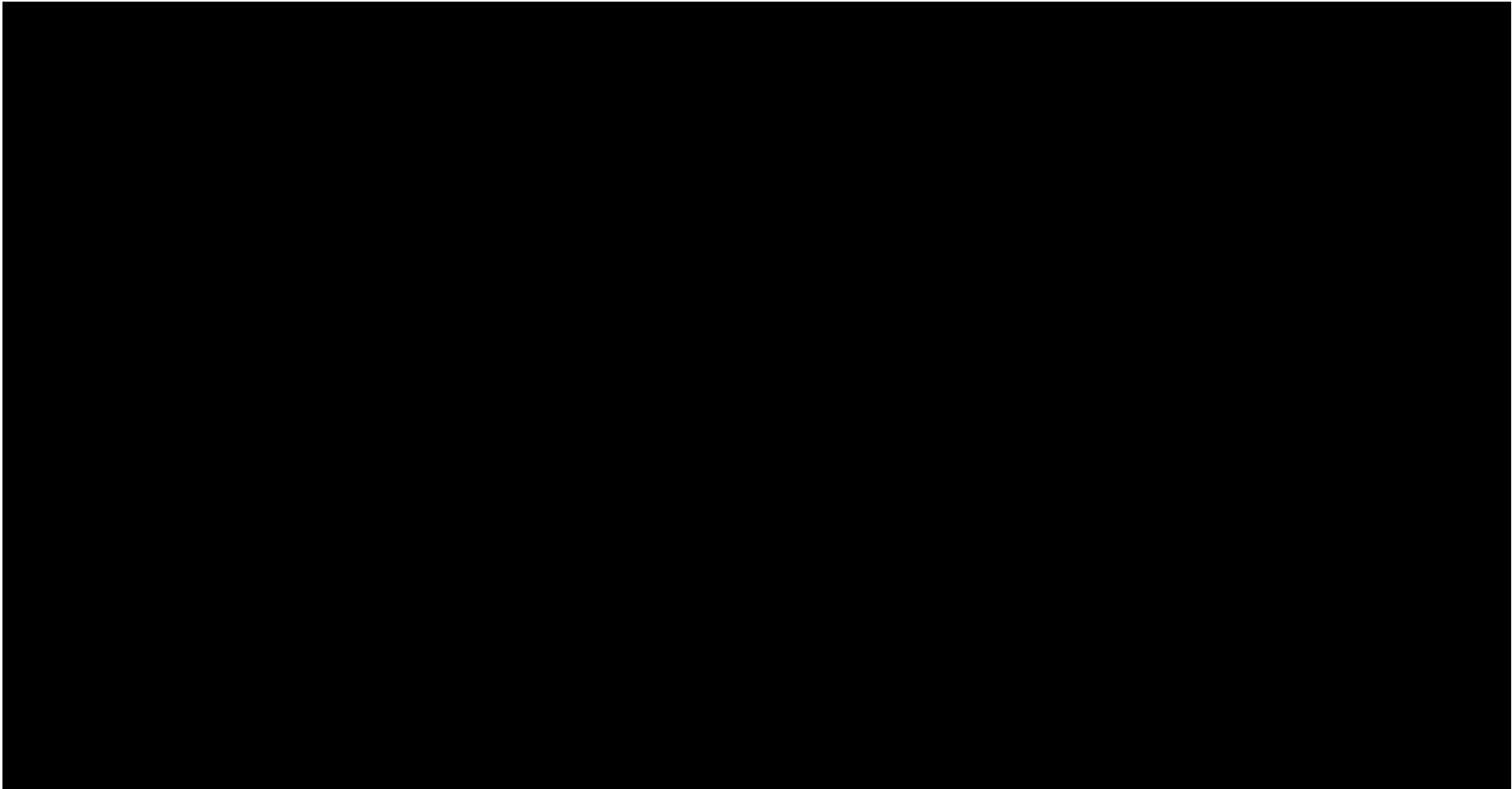
1. Extract image points
2. Normalize the image points
3. Compute the homography
4. Extract R and T from H
5. $\Pi^c = \mathbf{G}^{-T} \Pi^p$
 - a. Π^p is the plane normal (0, 0, 1, 0)
 - b. Simplified: $\Pi^c = (r3, -r3^T t)$
6. Visualize the planes



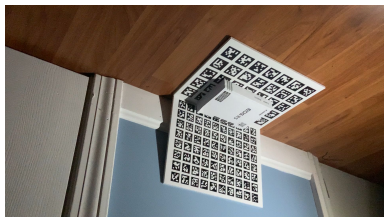
now visl
creating figure!



end of fig.show()



processVideo()



Video



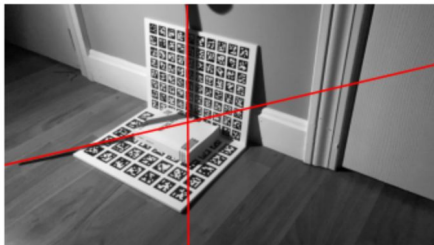
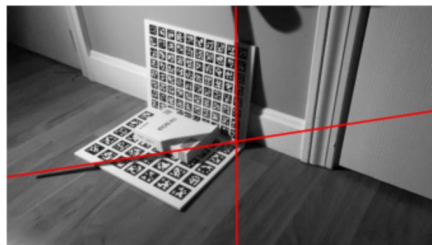
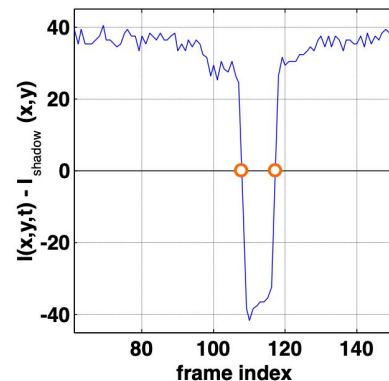
Compress, rotate,
Split into frames,
grayscale



frames

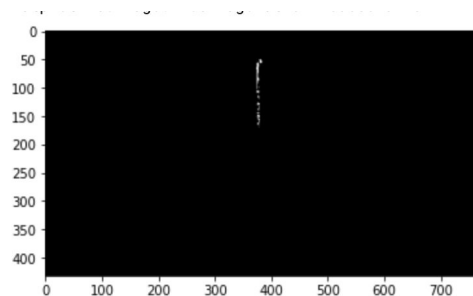


Find time that
shadow crosses each
pixel
(front_edge_times)



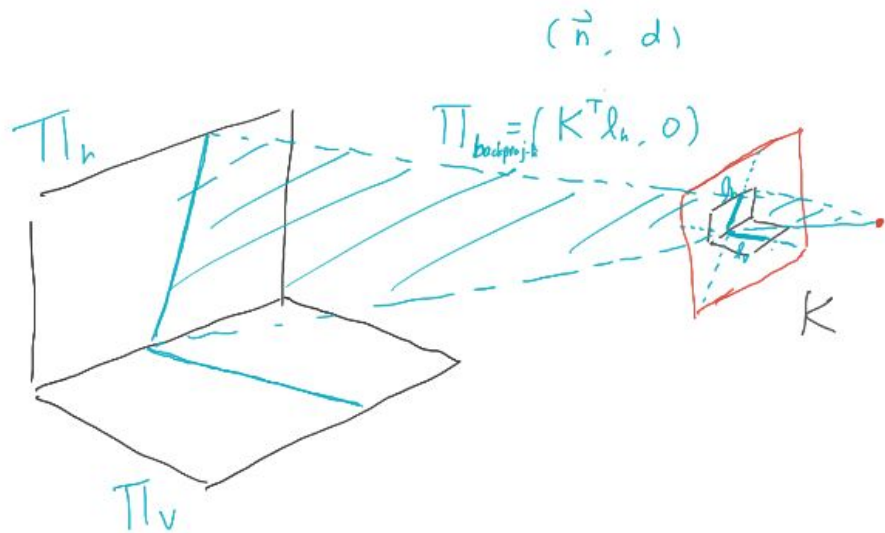
image_lines

SVD



computeShadowPlane()

- shadow_planes: dictates shadow plane at i^{th} video frame
 1. Backproject 2 points from image plane shadow line onto Π_h and Π_v
 2. Use SVD to best-fit shadow plane to 4 points

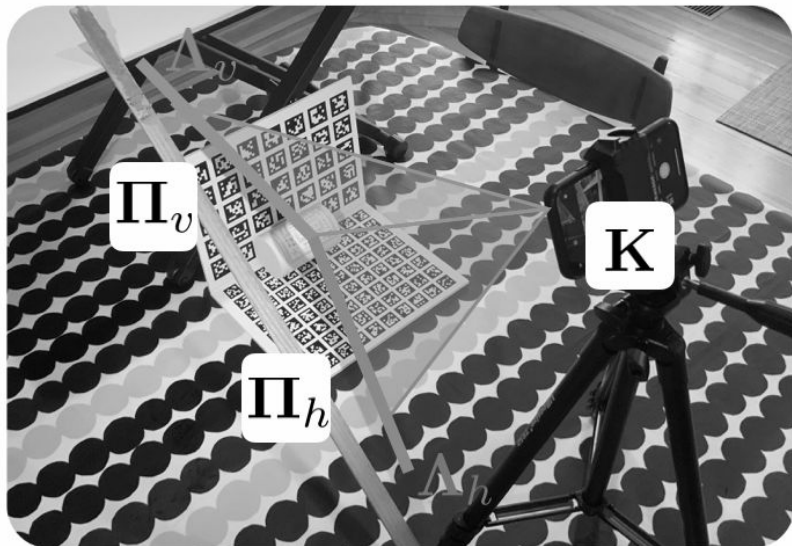


Pablo Qicasso

A.k.a. Claude Guonet

A.k.a. Vincent Van Qi

computeObjectPoints()



- point_cloud: contains 3D points in world coordinate system
 1. Backproject every points on object in image plane
 2. Find intersection of each point with correct shadow plane

Sources of Error

- Video compression?
- Video rotation?
- Angle of the light source?
- Programming errors (most likely)!

