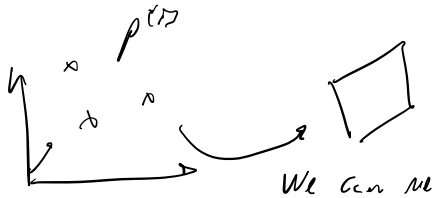
projective registration ≡ **PROJECTIVE ICP** → focus in how to develop this type of application

in this case, deal with measurements we take from a camera



We can see the points projected into the camera.

→ we will minimize the distance over the device plane (reprojection error)

3 components { camera model (map 3D points into img coordinate) / data association / least squares on manifolds

⇒ How to proceed? ≈ divide-and-conquer

1. Simulation Environment → test each module on its own (not put together least squares and data association machinery)

↓

Need be easy to debug models and know what is wrong

if I can work w/ camera, I can reprogram very easily a shape! (INSTEAD OF RANDOM POINTS)

⇓

set of segments → create random points over lines

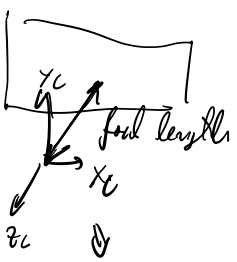
↓

if we put noisy weird shapes or distortion, our camera module is wrong

⇒ PIN HOLE CAMERA

Camera model ≡ mapping 3D pts to 2D image pts

PIN HOLE → most used one



CAMERA FRAME

principal point / center of projection

⇓

point of intersection between point and image plane

intrinsic parameters / camera matrix

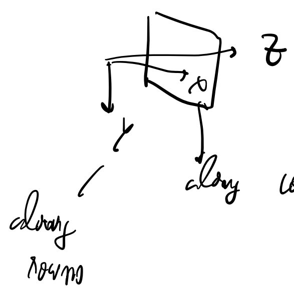
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & c_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \Big/ p_w$$

$\frac{X_{cam}}{Z_{cam}}$ = homogeneous division

3D → 2D ⇒ EK

3D pt → K camera matrix → homogeneous division

2D → 3D ⇒ only ray / direction from the center that pass through camera center ≡ unprojection



z ⇒ goes out of the camera

rows
cols → to check if pt is inside img plane or not

z ≤ 0 ⇒ cannot project points that are behind of us !

./camera_test

W → go forward
S
...

3) Data Association → nearest neighbor strategy (if space ⊕ reasonable given of camera)

↓

O(n²) ⇒ very slow.....

distance map → measurement not changes during ICP
- construct when receiving measurement
- query distance map in each iteration

4) Least Squares → exactly the same steps in ICP

exp measurements $z^{[m]} \in \mathbb{R}^2$

chain rule for derivation

z are real ⇒ belong to euclidean space !

world_pts
img_pts
| INPUT DATA

———— initialize (≈ reset)

Calibrate estimate camera pose

single iteration method
(given the correspondences)

Kernel_threshold → select kernel

provide perfect correspondences
to the robot to test

damping
the problem is not always
well conditioned → damping prevent that

LDLT ⇒ cholesky decomposing w/ diagonal

error at 0 should be 0
compute first numeric Jacobian (w/ auto diff)
⇓
Only then analytic jacobians

or the inverse
when problem is simple

ICP
- find correspondence
- optimize
- repeat