COMPUTER VISION PROJECT

3D OBJECT RECOGNITION OF MONOCULAR IMAGES

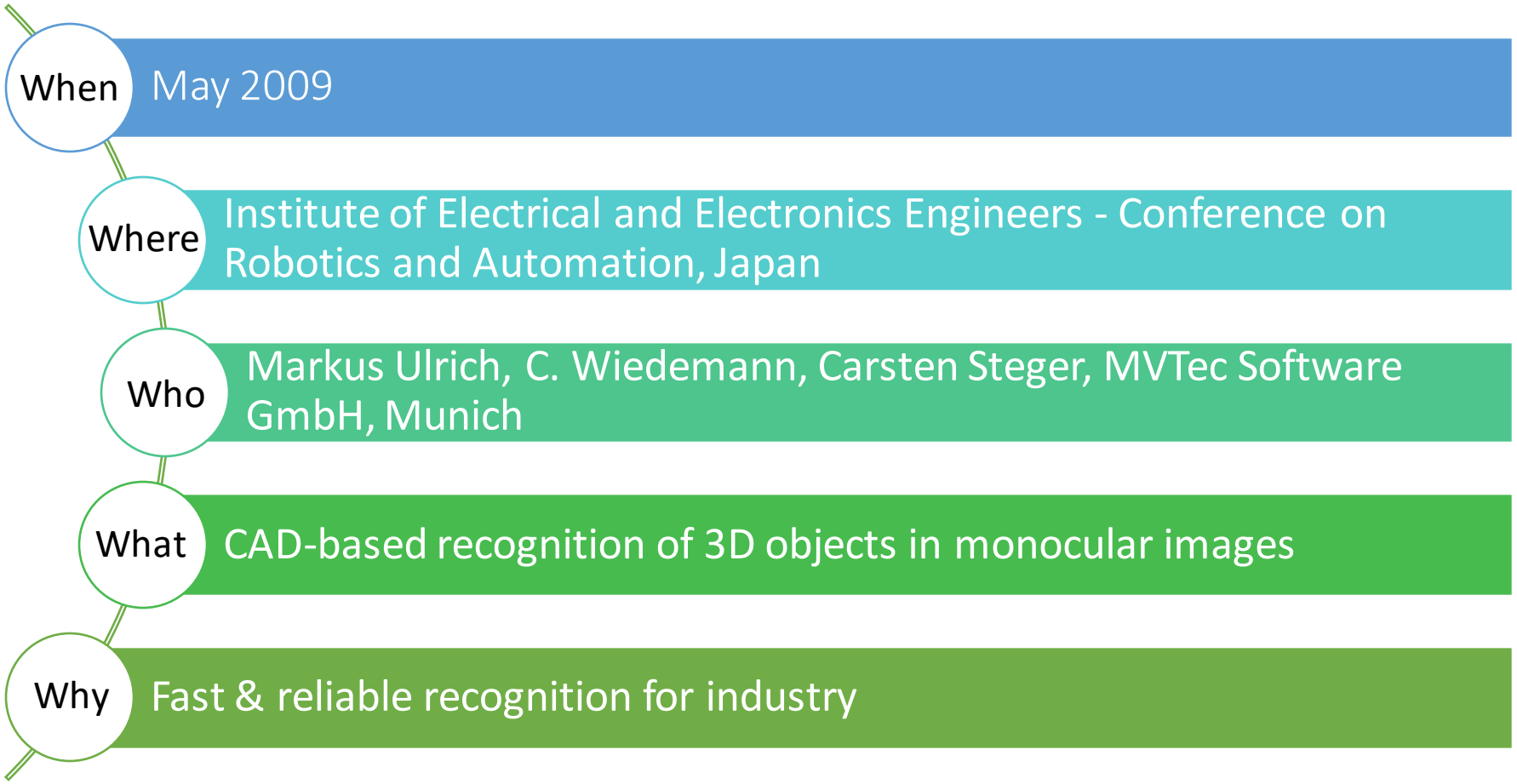
17/02/2022

Jean-Louis Materna - 1989315

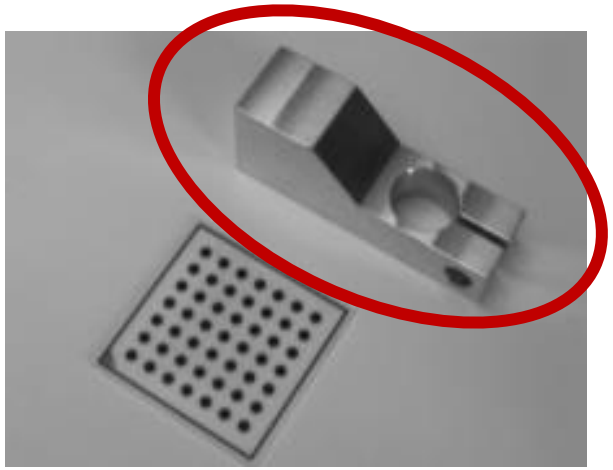
Tristan Desjardins - 1989333

1. PAPER PRESENTATION

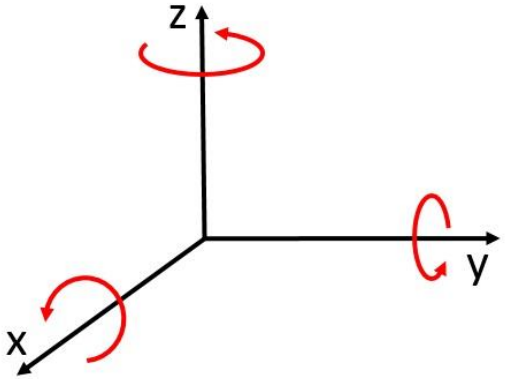
1.1. General informations



1.2. Goal



Image

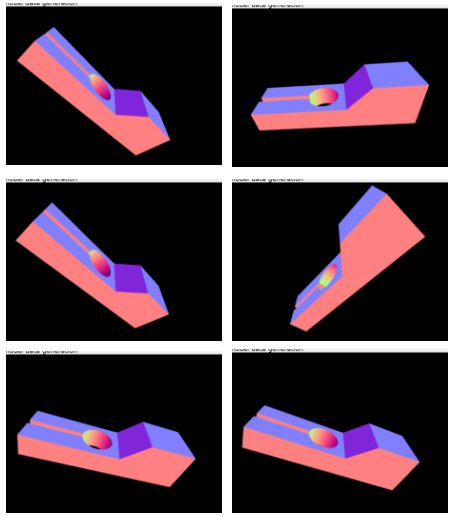


(x, y, z, rot_x, rot_y, rot_z)

Irrespective of :

- Texture (only geometry is used)
- Scale (take deformations into account)
- Object occlusions

1.3. Solution - General idea

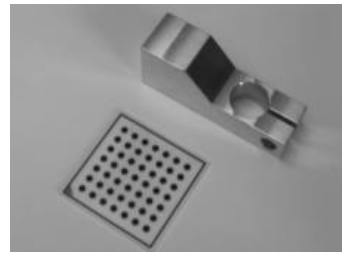


Generate images (camera views) of the object, known coordinates

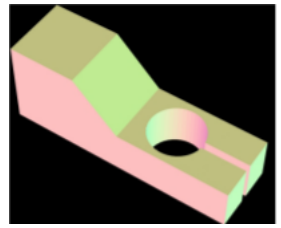
Generate a “tree” (pyramids)

Offline (training) phase

Test image

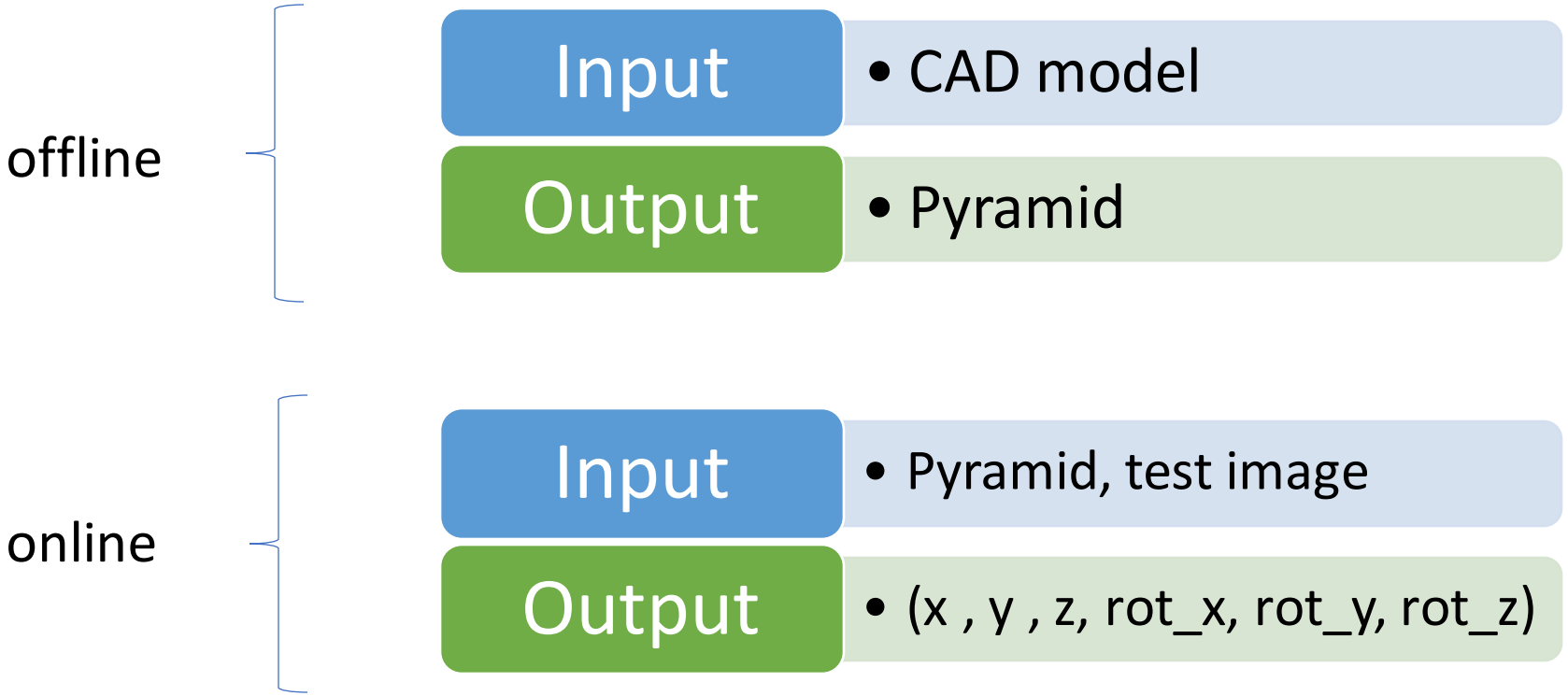


tree



Online (testing) phase

1.3. Solution - Structure

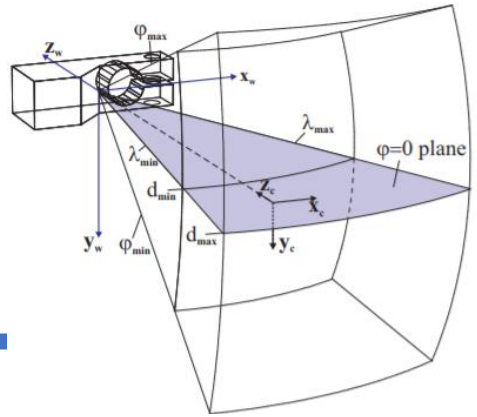


1.3. Solution - Theory

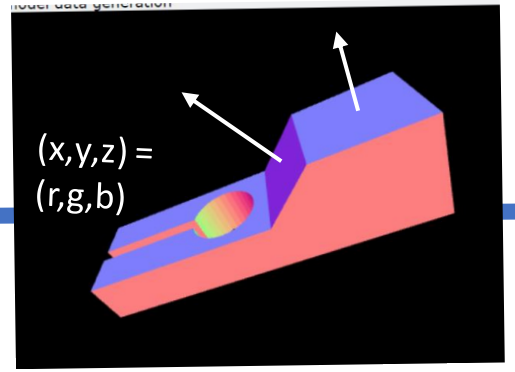
1.3.1. Generation of the camera views

CAD file

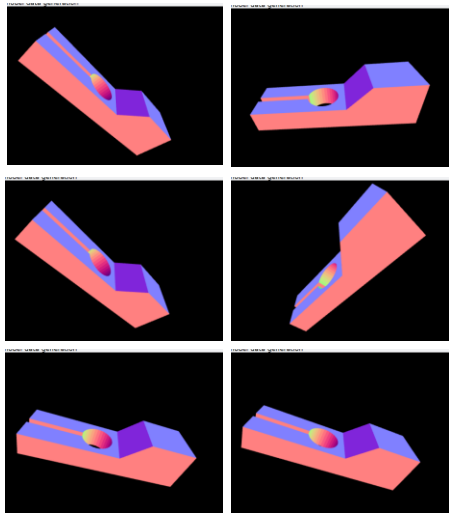
3D representation of the object



Camera position varies along (ρ, θ, ϕ)

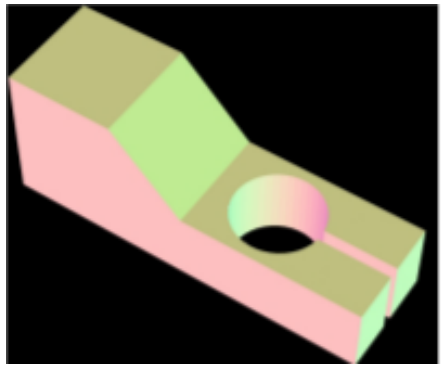


Adding colors to faces



Generated images

1.3.2. Filtering edges



Colors allow to store faces information in the image itself

$$C = \begin{pmatrix} grr & grc \\ grc & gcc \end{pmatrix}$$

$$\begin{aligned} grr &= gr_R{}^2 + gr_G{}^2 + gr_B{}^2 \\ grc &= gr_R gc_R + gr_G gc_G + gr_B gc_B \\ gcc &= gc_R{}^2 + gc_G{}^2 + gc_B{}^2 \end{aligned}$$

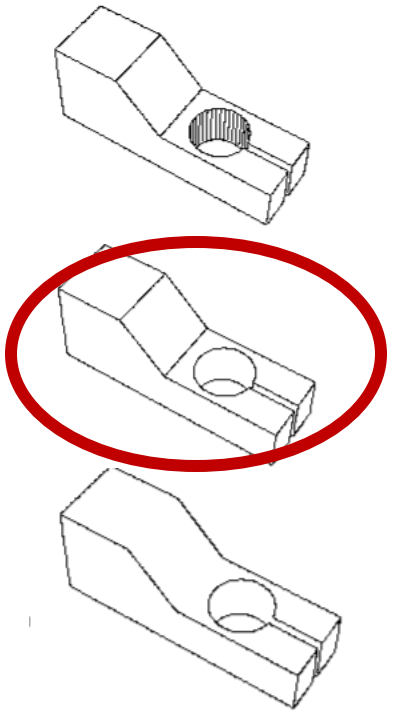
amplitude

$$A = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}$$

angle

$$\delta = 2 \arcsin(A/2)$$

Recovering edges amplitudes and angles from the colors



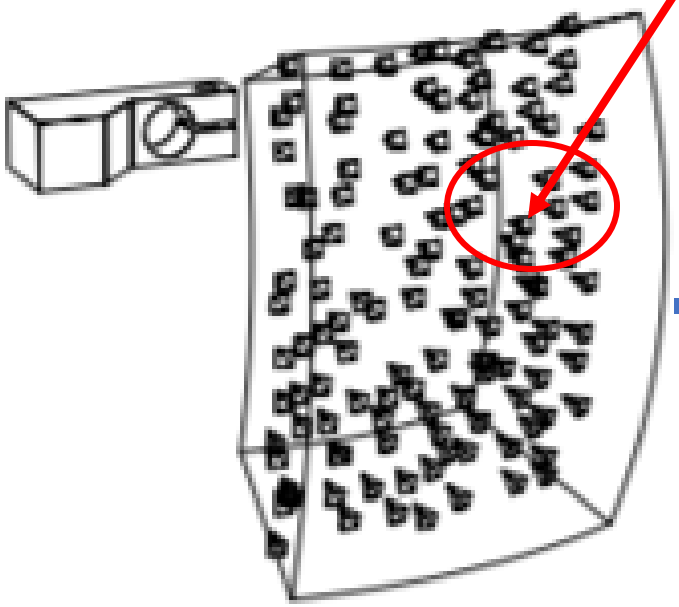
Threshold on the angle

1.3.3. Pyramid generation

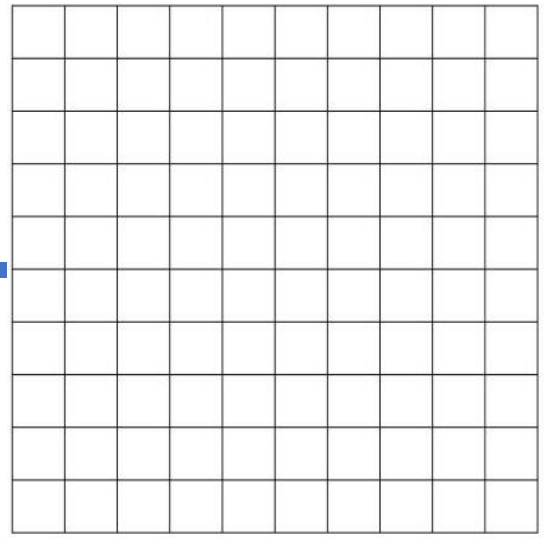
$$\text{similarity} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\langle m_i, s_i\rangle|}{\|m_i\| \cdot \|s_i\|}$$

If similarity > threshold,
-> merge the two views

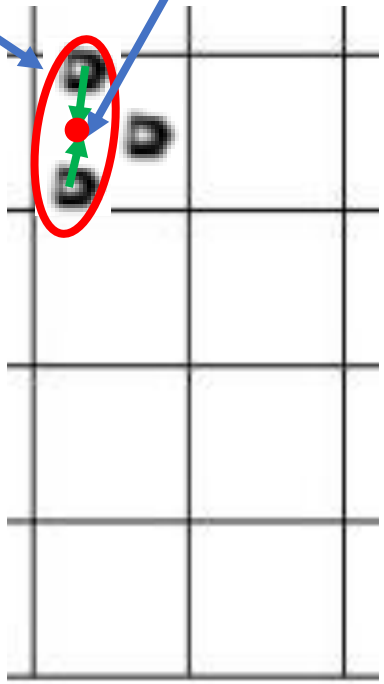
camera view + neighbours



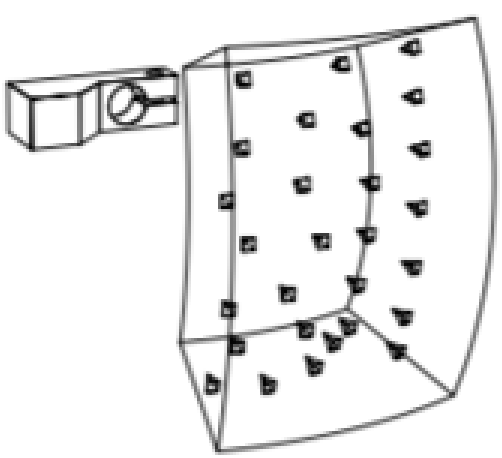
Pyramid – level 1



Grid that separates oversampled in subspaces

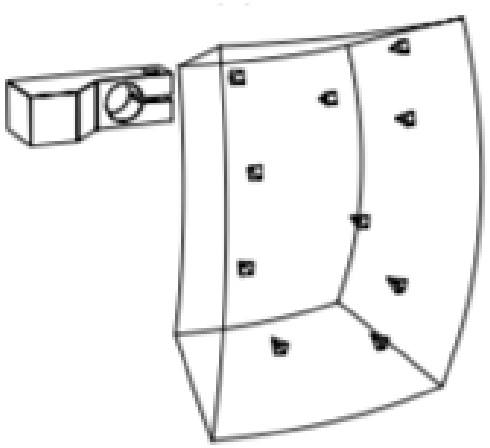


Repeat the merging procedure to build the different pyramid levels



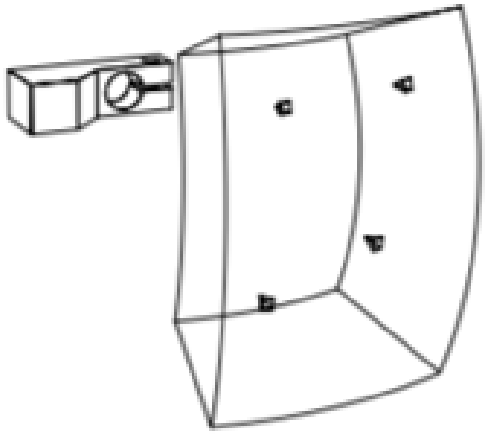
Pyramid – level 2
1st merge

merge



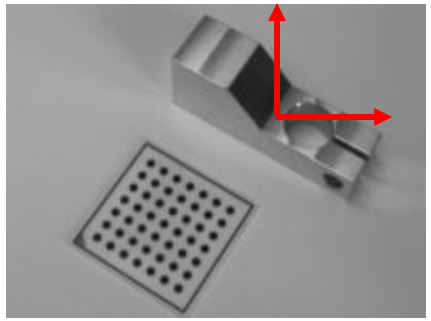
Pyramid – level 3
2nd merge

merge



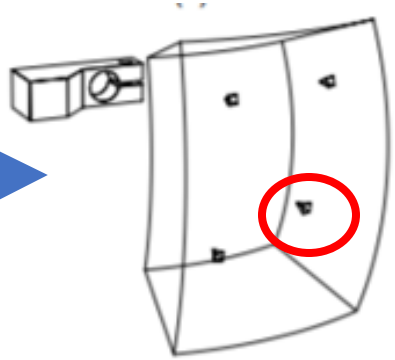
Pyramid – level 4
3rd merge

1.3.4. Testing

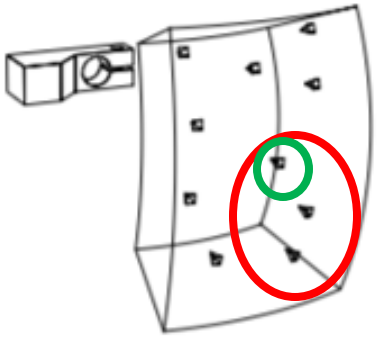


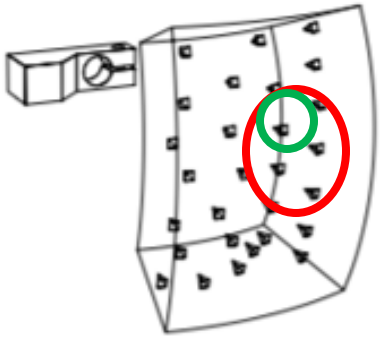
Test image

Level 4

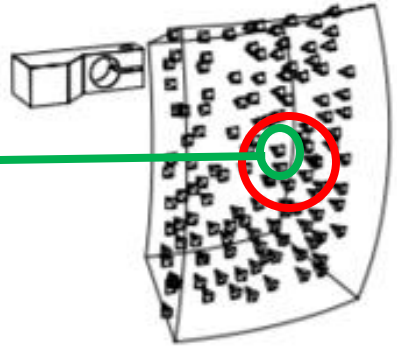


Level 3





Level 2



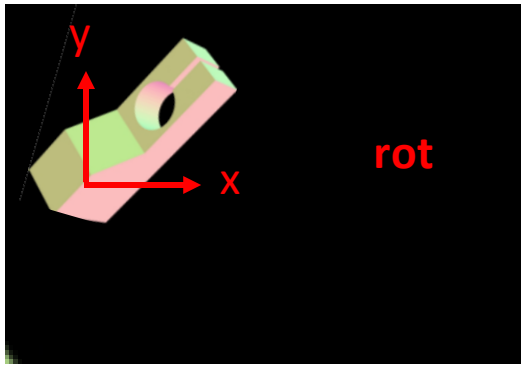
Level 1

Most similar camera view

1.3.5. Conversion

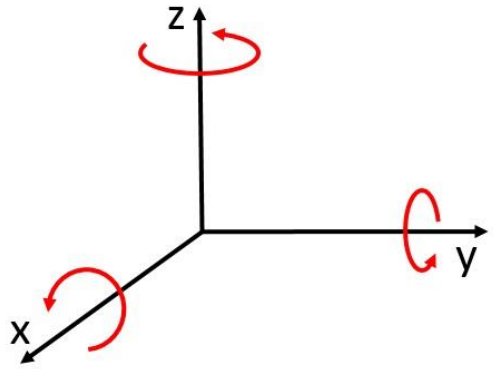
$(\rho, \theta, \phi, x, y, rot)$

of the most similar camera view



convert

(x, y, z, rot_x, rot_y, rot_z)



1.3.6. Refinement

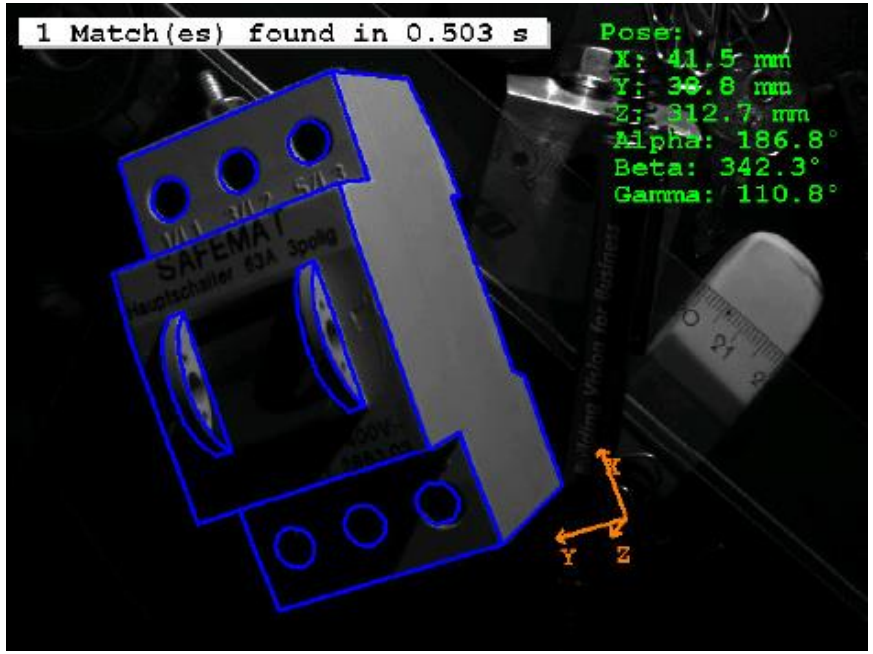
Pyramid sampling accuracy insufficient (limited by similarity threshold)

<u>Solutions:</u>

- Least Square Algorithm refinement
- Minimization over 6 degrees of freedom
- Subpixel accuracy
- Various optimizations

<u>Final results:</u>

- < 1mm standard deviation for (x, y, z)
- < 0.5° standard deviation for (rot_x, rot_y, rot_z)
- < 1s computation time



Practical object detection

2. PROJECT IMPLEMENTATION

2.1. Generating camera views

Blender 3D (clamp model) → Wavefront .obj file → Lists of points, edges and faces

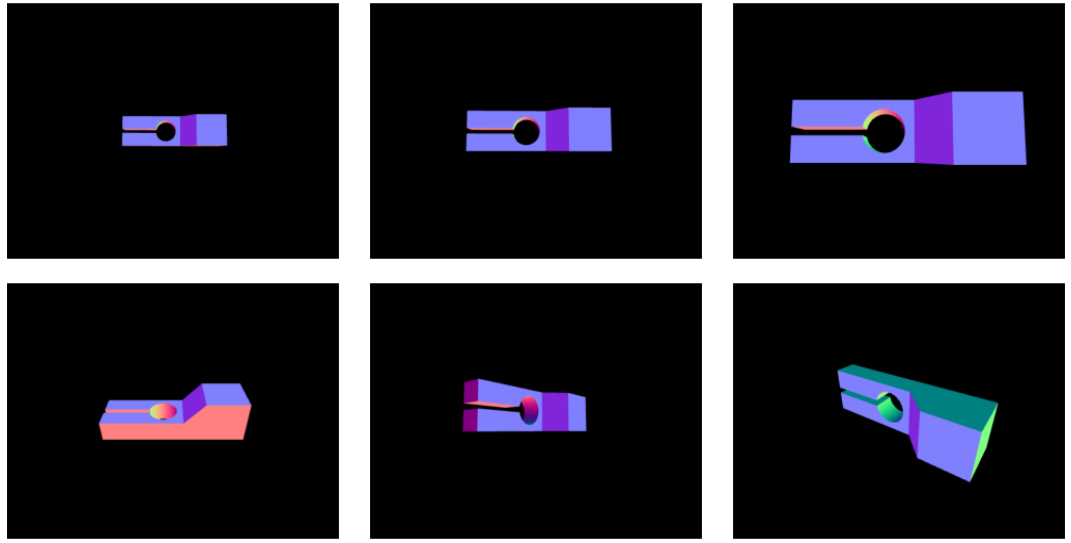
↓

Visual Python

↓

- range = (7,15), step = 2 for ρ (cm)
- range = (-50, 50), step = 2 for θ, ϕ (degrees)

←

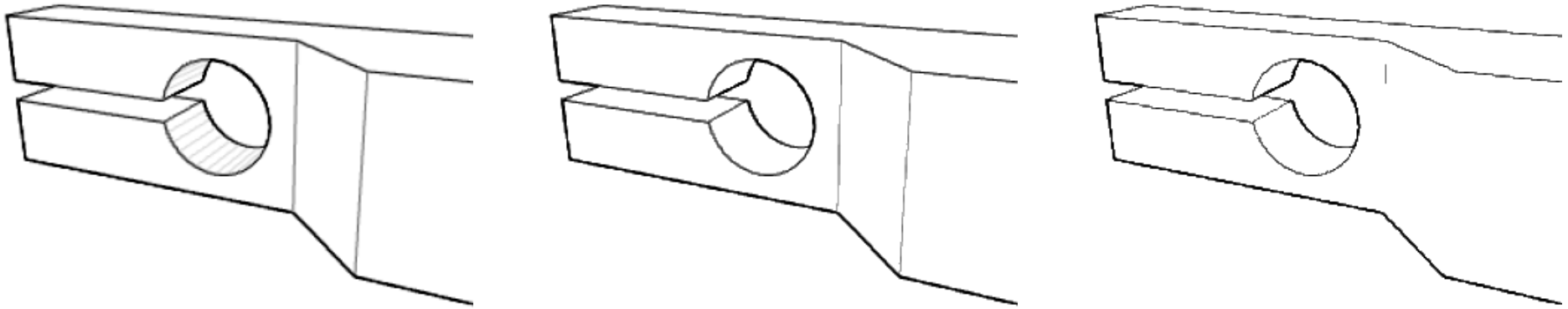


Subset of our generated images

2.2 Filtering

We manage to filter edges but did not use it for two reasons :

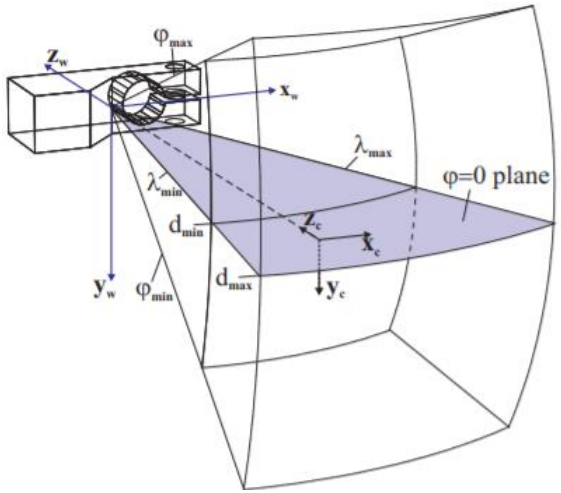
- Very costly in terms of time (computing gradients etc)
- We use a generated image as test image

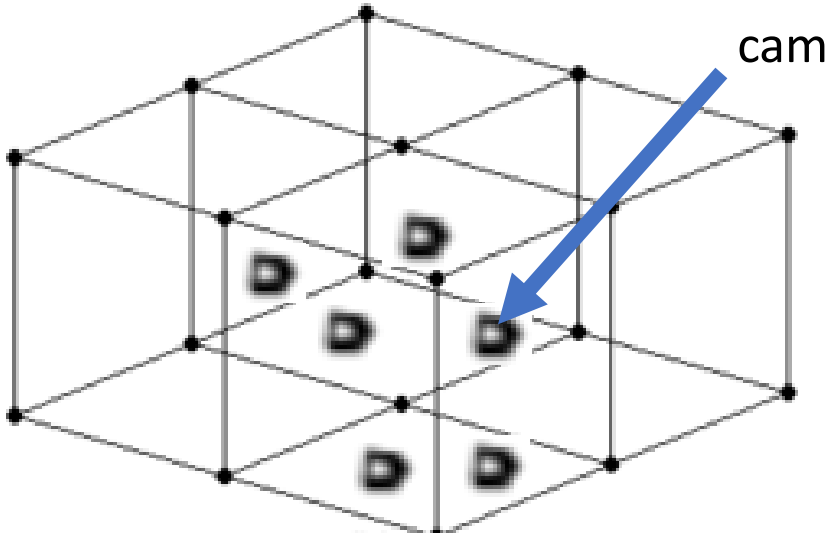


Edges remaining for threshold angle = 0°, 8°, 15°

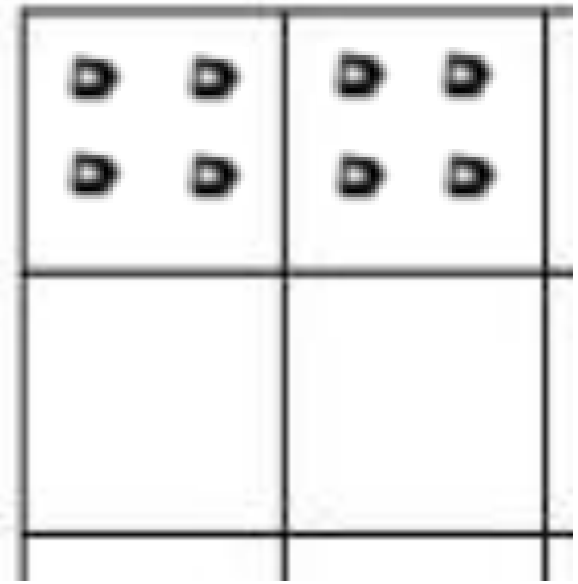
2.3. Pyramid generation

- Before generating pyramid, choose of ρ (range = (7,15), step = 2)
- Once ρ chosen, pyramid is build





In the paper (3D + random oversampling)



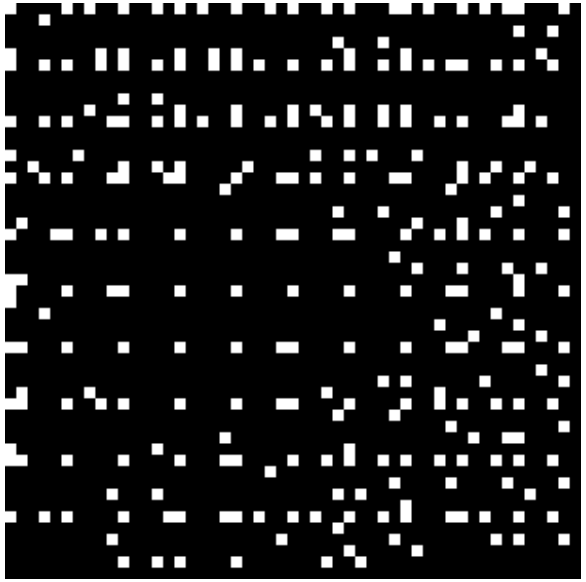
Our model (2D + ranges)

Optimization :

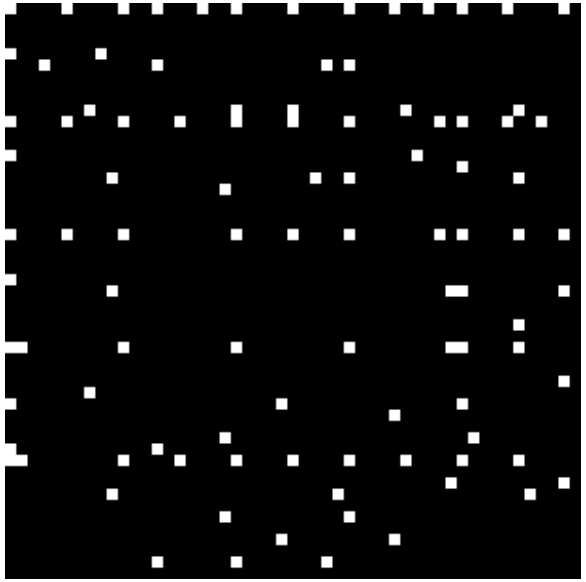
- **Using grayscale images instead of colored one**
  - ~3 times faster but loss of information
- **Optimization on the similarity**
  - Skip computation when color is constant
- **Subsampling of the images**
  - Makes the algorithm much faster (while still being reliable)
- **Saving the gradients in files to avoid recalculating them**

2.4. Testing

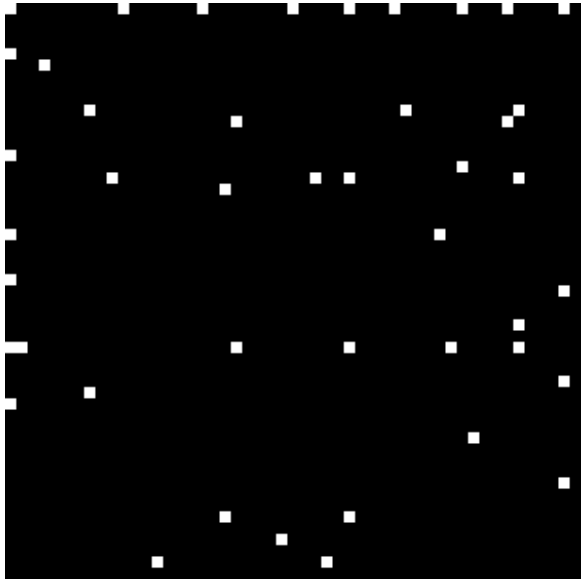
Pyramid obtained for ρ = 15



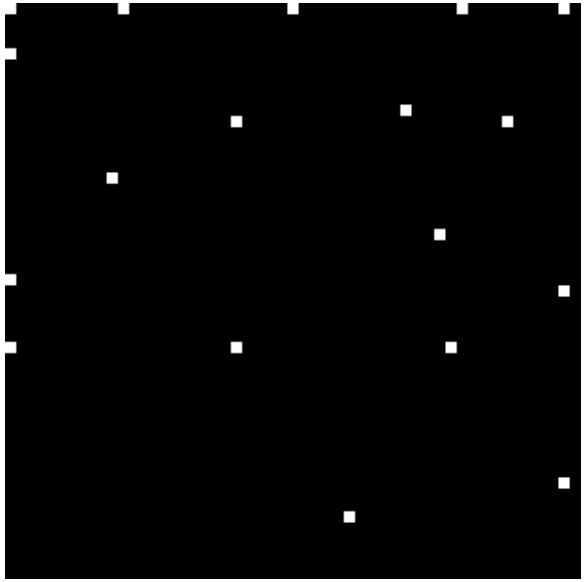
Pyramid – level 1 (1356)



Pyramid – level 2 (487)

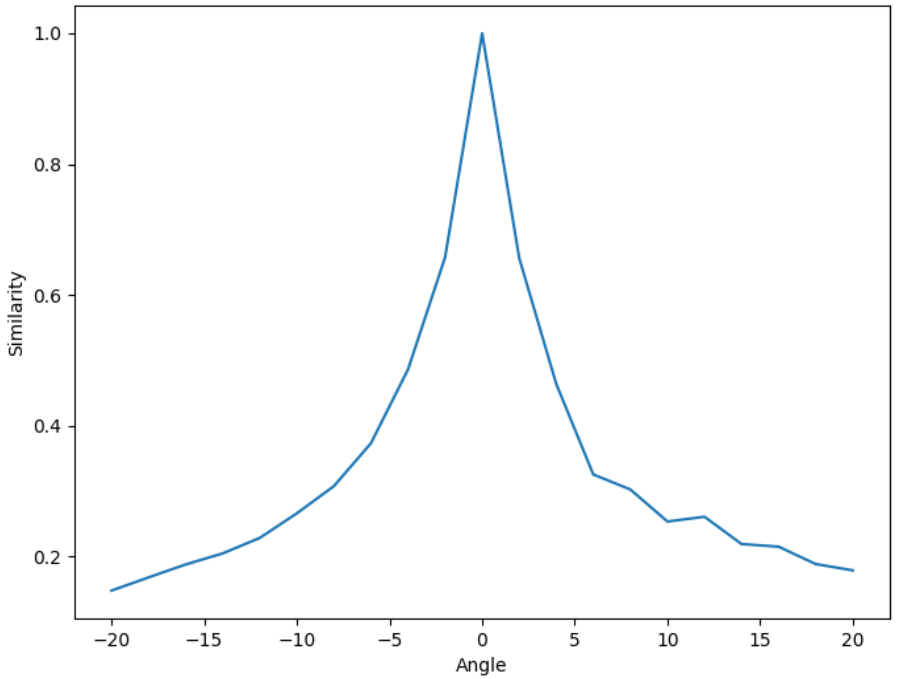


Pyramid – level 3 (227)

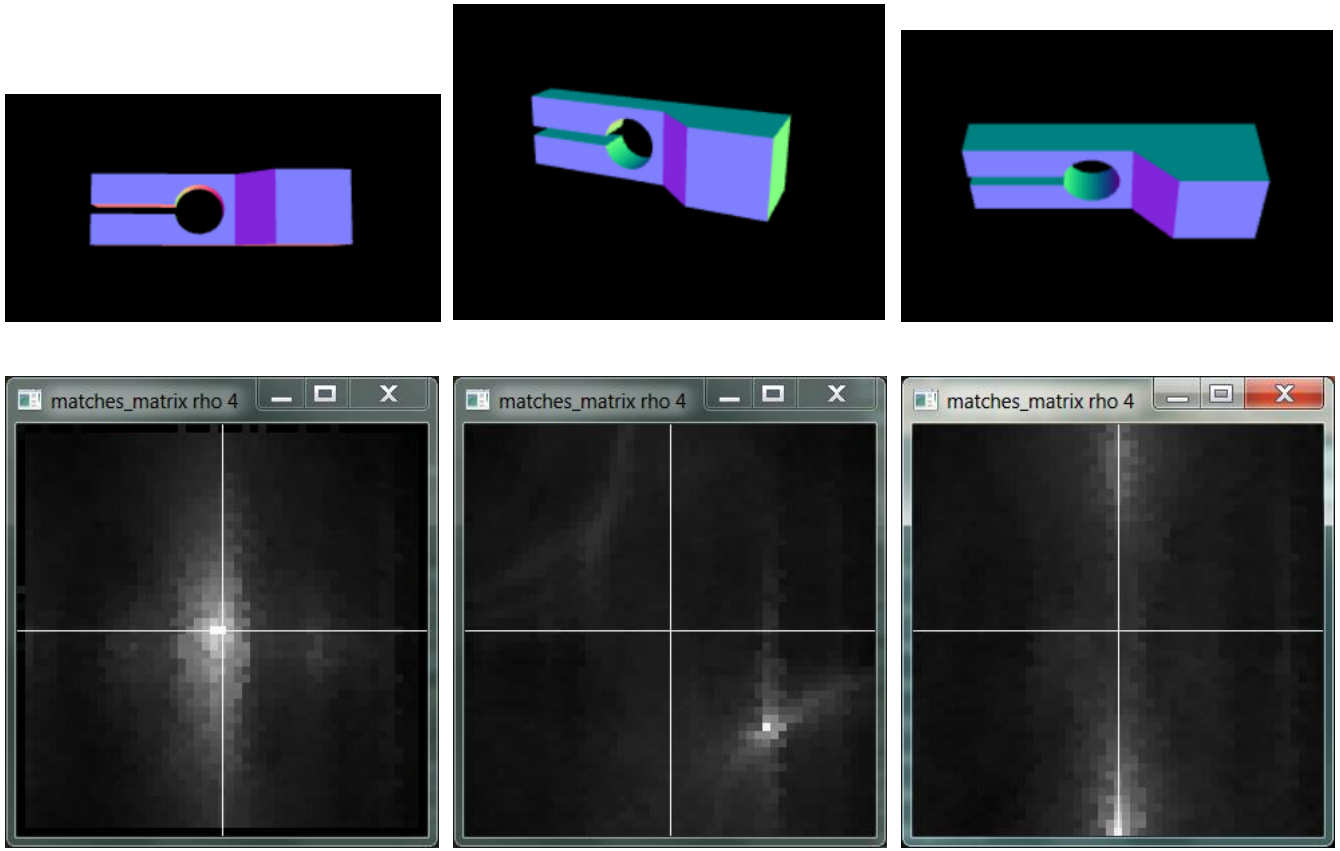


Pyramid – level 4 (87)

2.5. Similarity metric



Similarity plot (7, theta, 0) for theta in [-20 ; 20]



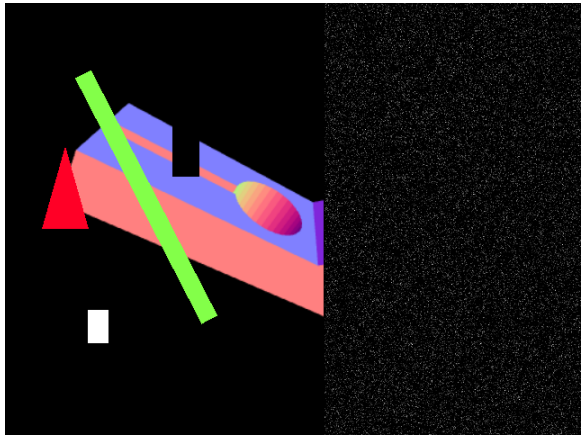
Similarity scans (rho = 15, theta = 0, 24, 50, phi = 0, 24, 0)

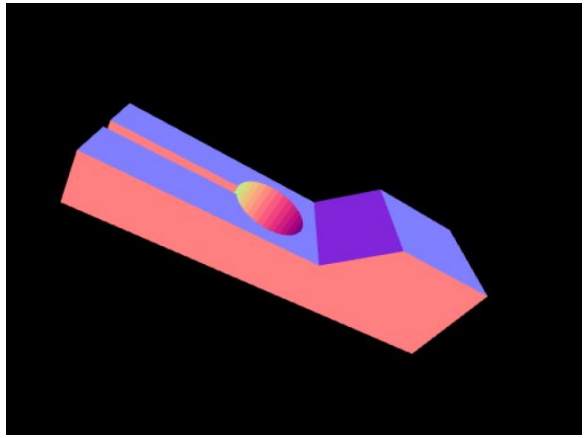
2.6. Results & performance

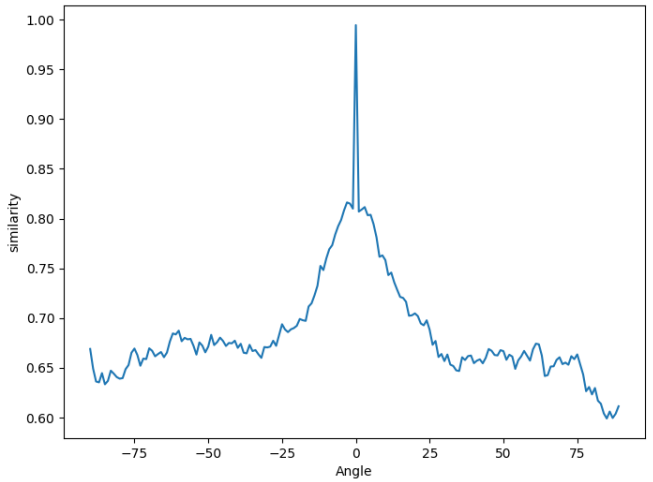
Found pose: most cases 100% accurate (5 range refinement).
Some errors at edges.

Computation time: 20s average, 6s min, 30s max.

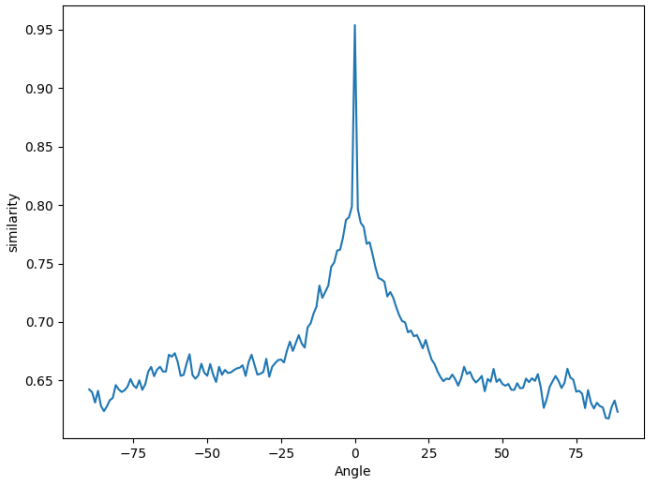
Robust with occlusions and light deformations (random noise).







Baseline rotation plot



Occlusion rotation plot

3. CONCLUSION