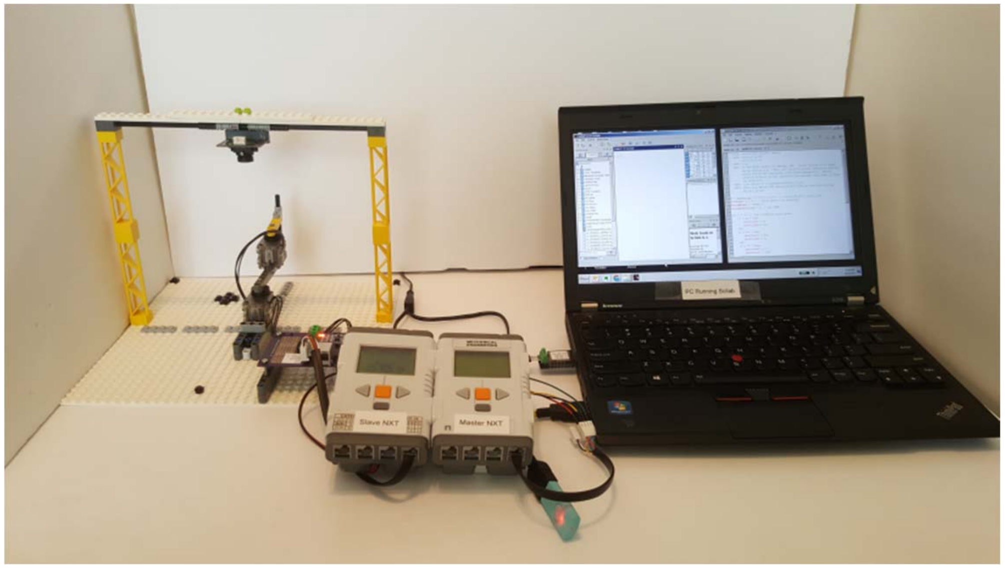
Visually Servoed Robotics

Visual-servoing uses image data to command robot behavior. This is often seen in assembly lines for pick-and-place tasks. Robots often use visual-servoing to avoid obstacles or track targets. There are two different configurations. The first is external; the camera's field-of-view (FOV) sees the robot and workspace. The second is internal and often called "eye-in-hand"; the camera is mounted on the robot and sees the workspace. Regardless of configuration, visual-servoing demands knowing how to map pixel locations to those in the world (i.e. task space or work space)

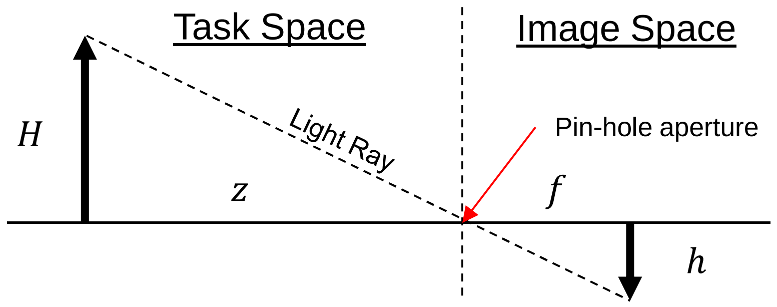
Question: Below is an external camera configuration. The camera is mounted above the robot (a 2-link planar manipulator) and its task space (the white 32 x 32 Lego base plate). What is the mapping of the camera's pixel coordinates (in image space) to the robot's (x, y) coordinates (in task space)?



© 2020, Paul Oh

Solution: To begin, a camera model is needed. A camera model describes how light rays reflect off objects and into the camera's imaging sensor. The pin-hole camera model is the simplest but assumes no lens distortions and all pixels in the imaging sensor are the same size.

With this model, the light rays are straight lines, all passing thru the lens' aperture (i.e. pin-hole). Thus similar triangles describes the relationship between sizes in task space and image space.

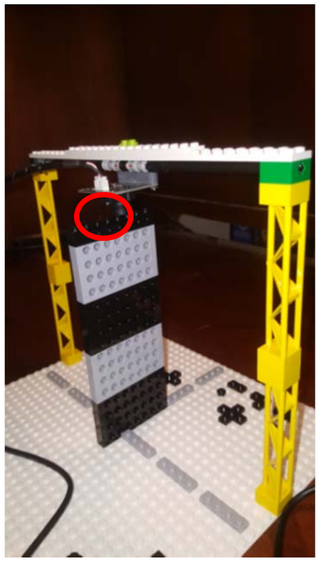


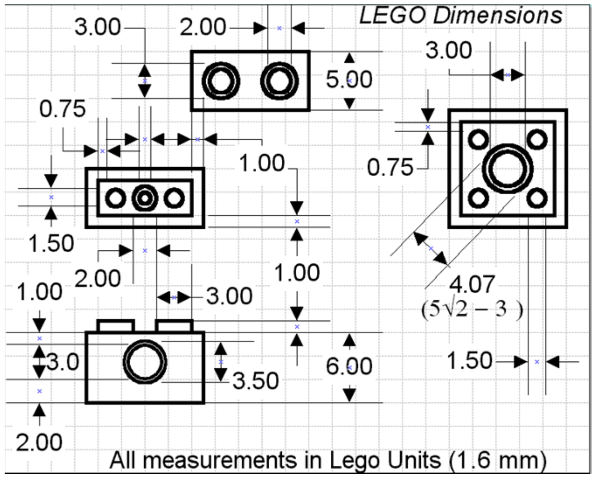
Similar triangles says: $\frac{H}{z} = \frac{h}{f}$ then $f = z\frac{h}{H}$

Where

H: object height (e.g. meters, studs)
z: target-to-lens distance (e.g. meters, studs)
h: image height (in pixels)
f: lens focal length (in pixels)

Step 1: Measure the target-to-lens distance z





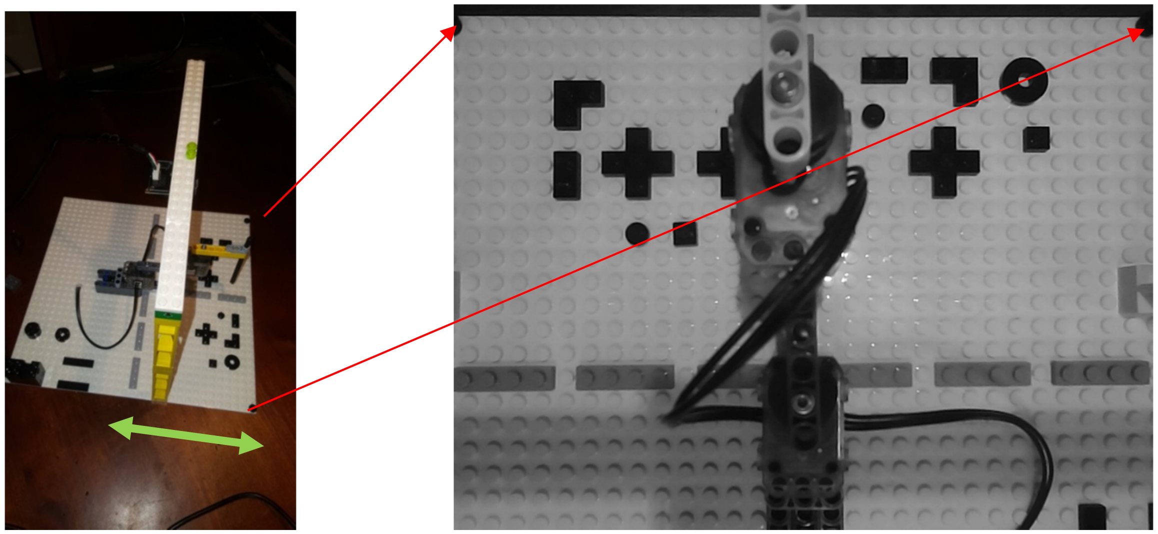
https://bricks.stackexchange.com/questions/288/what-are-the-dimensions-of-a-lego-brick

Bricks were stacked on the base plate until they touched the camera's lens. The photo shows this stack consisted of 17 Bricks. The topmost Brick's stud (red circle) touched the lens

Each Brick measures 6 mm high. Thus $z = 17 \times 6\text{ mm} + 1\text{ mm (due to last stud)} = 103\text{ mm}$

© 2020, Paul Oh

Step 2: Measure object height in task space



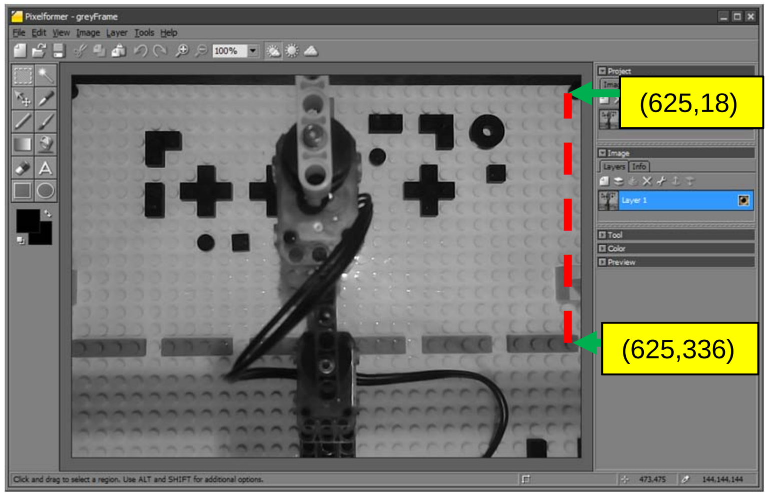
Two black 1-round studs were mounted on the top corners of the white base plate. These serve as calibration points. At this camera height and position, the camera's FOV shows the calibration points (almost) at the top corners of the image (red arrows).

A ruler measures the height (green arrow) to be 12 cm. Alternatively, one counts 15 stud spacings hence
$H = 15 \times 8 \text{ mm} = 120 \text{ mm} = 12 \text{ cm}$

© 2020, Paul Oh

Step 3: Calculate focal length f

One frame was captured in Scilab as a PNG file. This image file was then imported into Pixelformer.


Pixelformer - greyFrame
File Edit View Image Layer Tools Help
(625,18)
(625,336)
Click and drag to select a region. Use ALT and SHIFT for additional options.

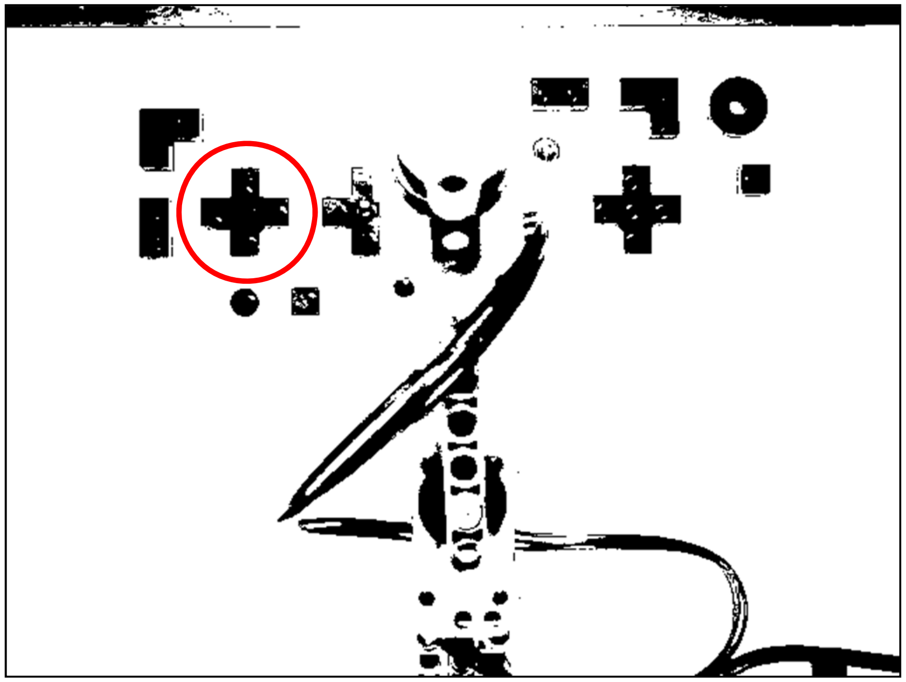
Mouse locations report pixel values. Green arrows and yellow text box show pixel values of these one of the calibration points. Hence $h = 336 - 18 = 318$ pixels

Answer: Hence $f = z\frac{h}{H} = 103 \text{ pixels} \frac{318 \text{ pixels}}{120 \text{ mm}} = 272.95 \text{ pixels}$

Now, with the focal length f any pixel location (u, v) in the camera's image space, one can calculate the (x, y) in the task space

© 2020, Paul Oh

Problem: Use a pin-hole camera model to determine the Lego Cross' (x, y) location (red circle) in the robot's task space

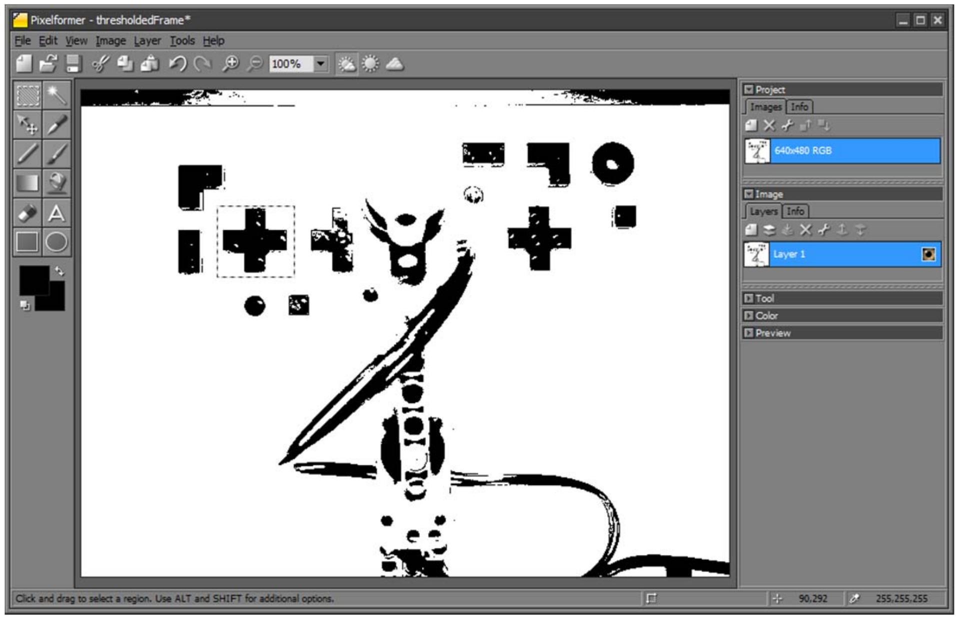


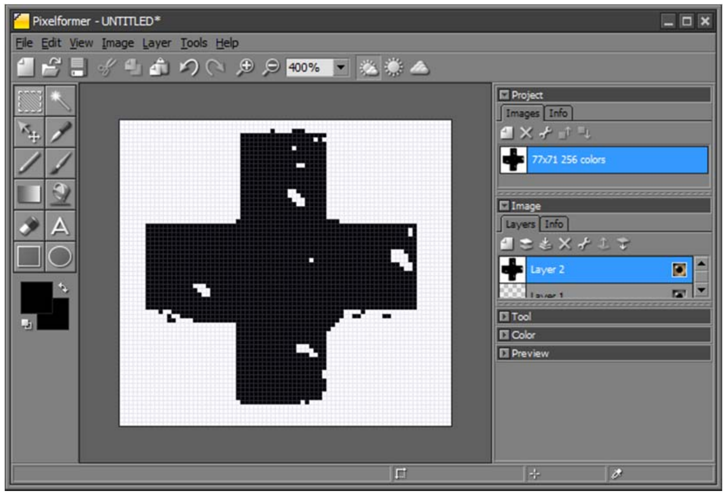
Solution: Thresholding was performed on camera video. A sample frame was captured as a PNG file above.

© 2020, Paul Oh

Step 1: Create template for object detection

The 640 x 480 PNG file was imported into Pixelformer (below left). A template (below right) was created by cropping the 640 x 480 image (faded dashed box around the Lego Cross). The resulting template measured 71 x 77 pixels.


Pixelformer - thresholdedFrame*
File Edit View Image Layer Tools Help
100%
Project
Images Info
640x480 RGB
Image
Layers Info
Layer 1
Tool
Color
Preview
Click and drag to select a region. Use ALT and SHIFT for additional options.
90,292
255,255,255



crossTemplate.png: (71 x 77) row x column template

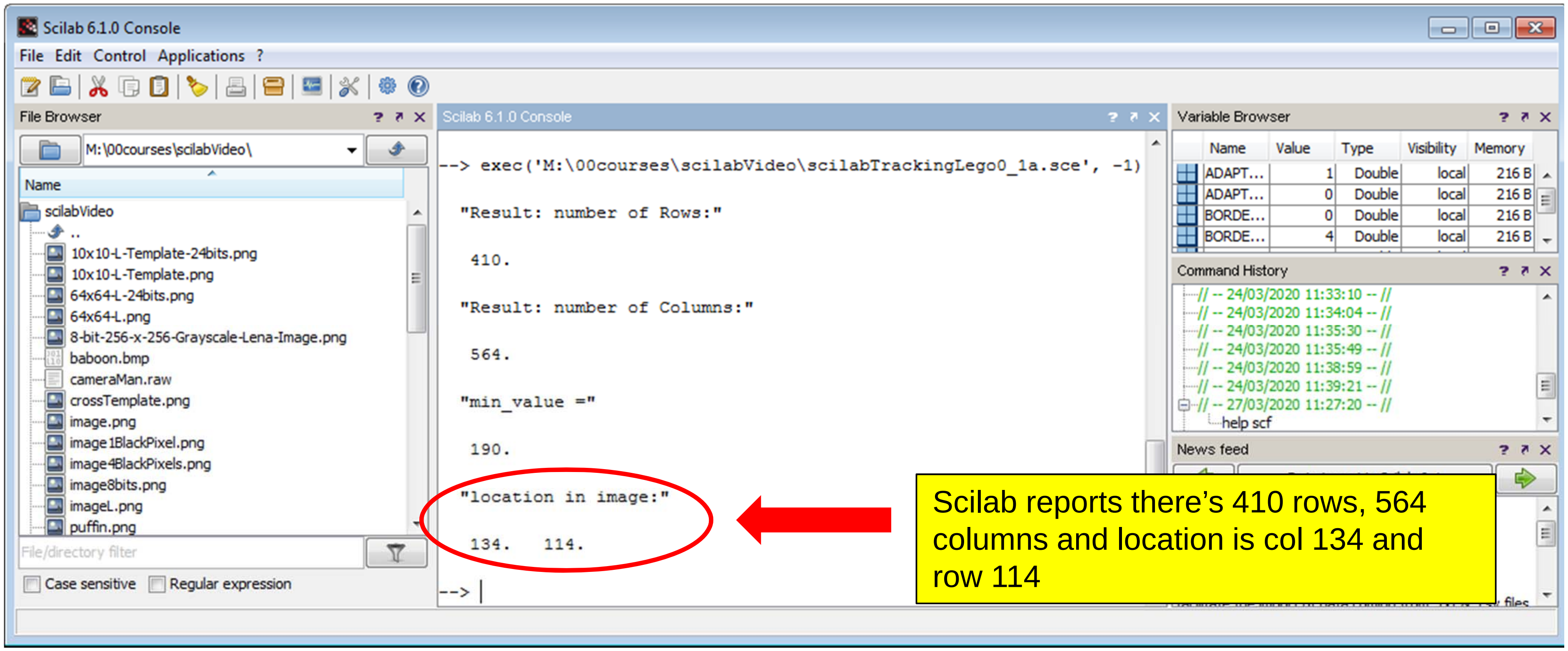
NB: Pixelformer also shows that the center of the Lego cross is **(149, 172)**. However, we want object tracking to yield this pixel location.

© 2020, Paul Oh

Added on 03/30/20

Step 2: Use object tracking to determine pixel location of Lego Cross

The thresholded 640x480 PNG file and 71x77 template PNG file were fed into a Scilab SSD program



```
--> exec('M:\00courses\scilabVideo\scilabTrackingLego0_1a.sce', -1)

 "Result: number of Rows:"

  410.

 "Result: number of Columns:"

  564.

 "min_value ="

  190.

 "location in image:"

  134.   114.

-->
```

Name	Value	Type	Visibility	Memory
ADAPT...	1	Double	local	216 B
ADAPT...	0	Double	local	216 B
BORDE...	0	Double	local	216 B
BORDE...	4	Double	local	216 B

```
// -- 24/03/2020 11:33:10 -- //
// -- 24/03/2020 11:34:04 -- //
// -- 24/03/2020 11:35:30 -- //
// -- 24/03/2020 11:35:49 -- //
// -- 24/03/2020 11:38:59 -- //
// -- 24/03/2020 11:39:21 -- //
// -- 27/03/2020 11:27:20 -- //
   help scf
```

Scilab reports there's 410 rows, 564 columns and location is col 134 and row 114

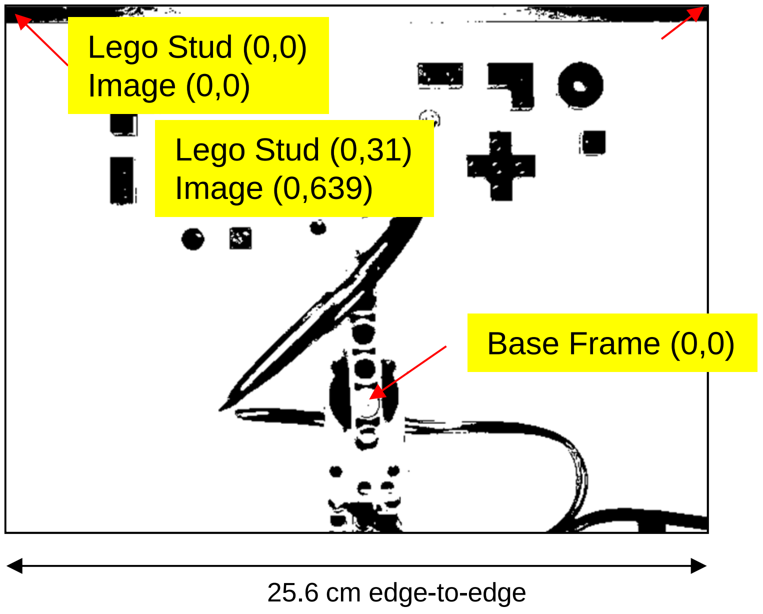
SSD reports Row 114. Template # Rows = 71. Thus 114 + 71/2 = 149.5
SSD reports Col 134. Template # Cols = 77. Thus 134 + 77/2 = 172.5
Thus the image's pixel location is (row, col) = **(149, 172)** which matches previous slide

lecture-visualServoing-041620b.pptx

© 2020, Paul Oh

Added on 03/30/20

Step 3. Determine the mapping between thresholded camera view (image space) and Lego 32 x 32 baseplate (task space)



25.6 cm edge-to-edge
24.8 cm from left-most stud to right-most stud

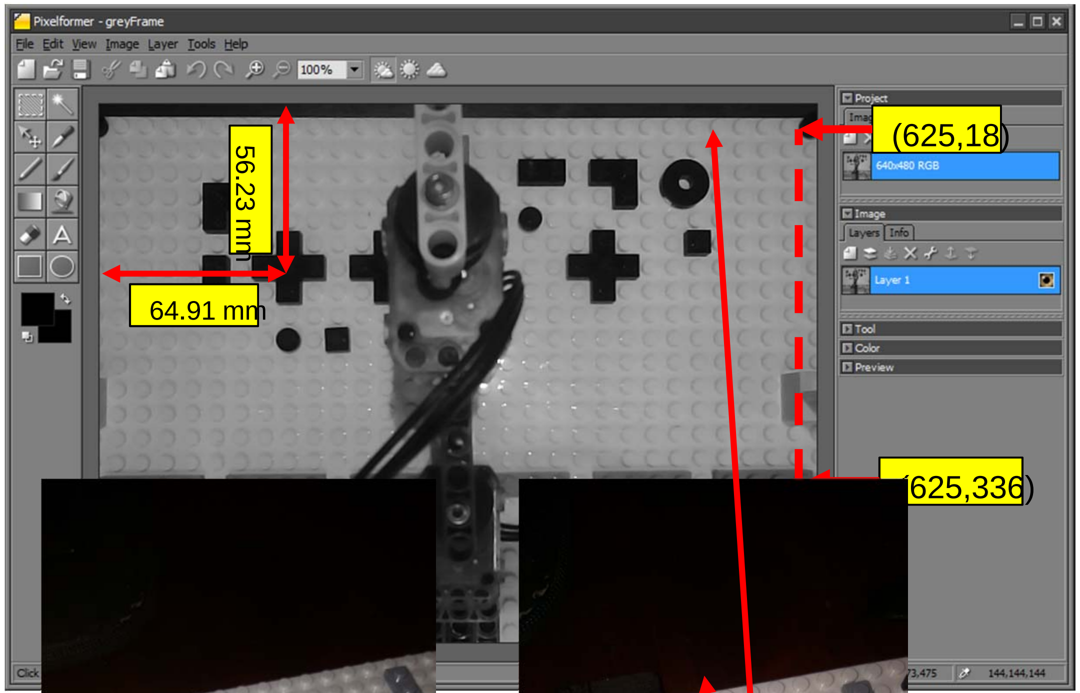
This is a 640 x 480 PNG file

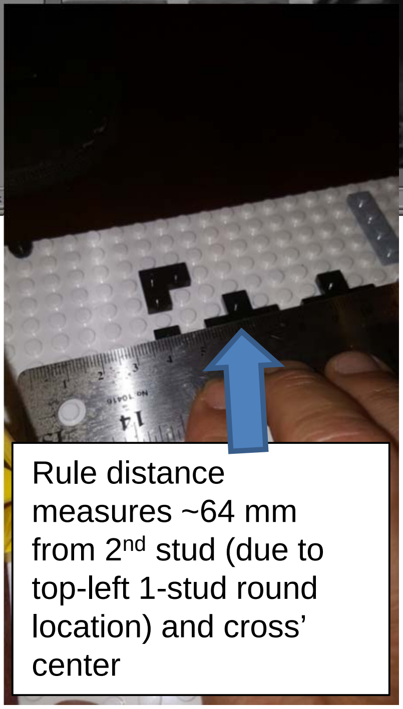
Alternatively one counts 31 stud spacings (8 mm) between left-most and right-most studs. Hence 31 * 8 mm = **248 mm**. Hence, consistent.

© 2020, Paul Oh

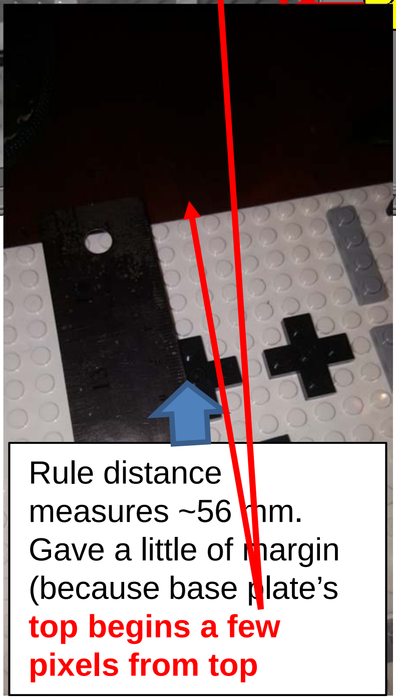
Added on 03/31/20

Step 4. Apply pin-hole camera model to find Lego Cross measurements in task space





Rule distance measures ~64 mm from 2nd stud (due to top-left 1-stud round location) and cross' center



Rule distance measures ~56 mm. Gave a little of margin (because base plate's **top begins a few pixels from top**

SSD reported the template (Lego Cross) at (149, 172) pixels in the thresholded image.

Hence in the task space and using the previously calculated focal length f:

Answer:

$$R = z\frac{r}{f} = 103\frac{149}{272.95} = 56.23 \text{ mm}$$

$$C = z\frac{c}{f} = 103\frac{172}{272.95} = 64.91 \text{ mm}$$

These computations are confirmed with a ruler

© 2020, Paul Oh

Closing Remarks

- External visual-servoing configuration enables robot to know object locations in its task space
- Lens focal length can be calculated using a pin-hole camera model
- Focal length and pixel locations (e.g. from SSD) yield's object location in task space
- Caveats
  - There is always lens distortion; pin-hole model somewhat overly simplistic
  - Glare and shadows can throw off thresholding
  - SSD only as good as the thresholded image
  - SSD does not account for any warping, rotation and perspective (i.e. zoom) effects
  - SSD reports top corner of match. For larger objects, must account for image size to determine the center of the object

Why important

The Lego Camera Tower and Scilab SSD can report the pixel location of an object. The pin-hole camera model can yield the object's location in the task space. Inverse kinematics uses these task space locations to yield robot joint positions. In the XL-320 2-link planar manipulator, the end-effector can then hover over the object's location

© 2020, Paul Oh