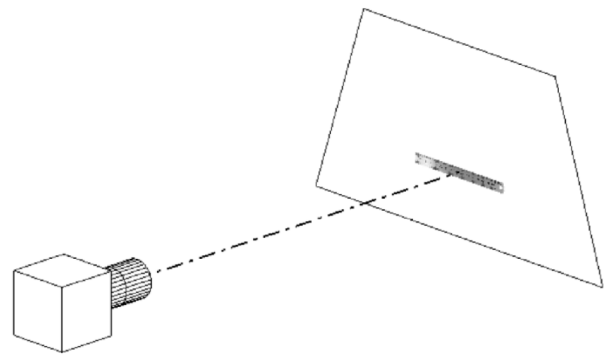
Camera calibration

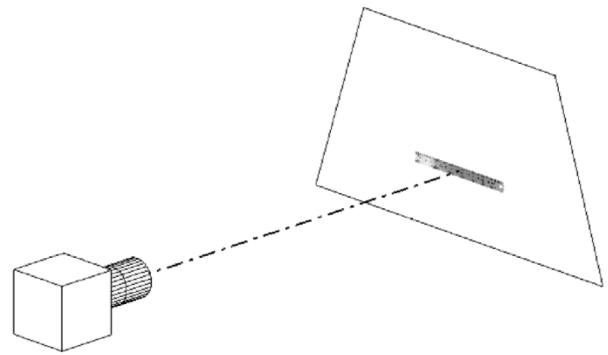
For a simple 1d system, we could just place a ruler in the field-of-view

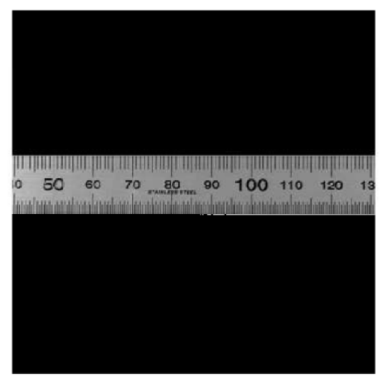


we know:
Δi and Δj (in pixels)
we want:
Δx and Δy (in m)

Camera calibration

For a simple 1d system, we could just place a ruler in the field-of-view

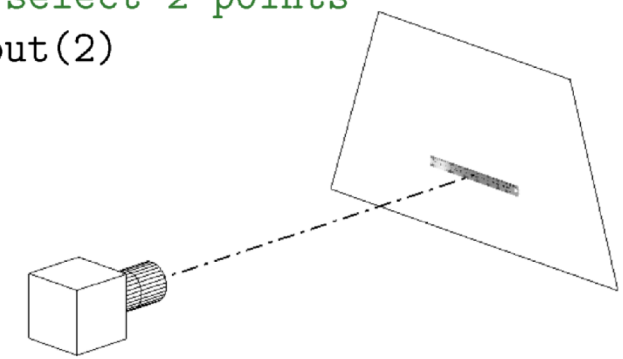


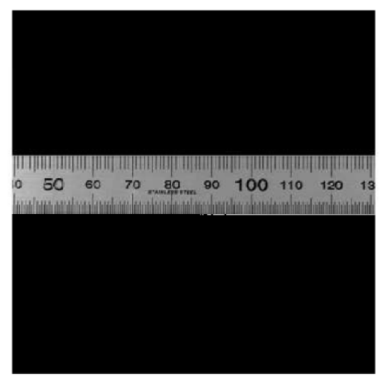


Camera calibration

If we select 2 known graduations, we can obtain the pixel to meters conversion.

```
% graphically select 2 points
[ip,jp] = ginput(2)
```

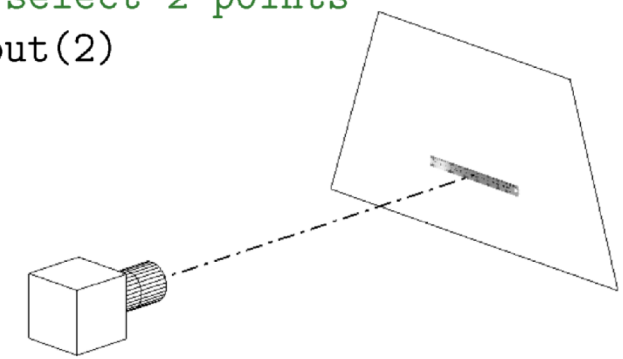


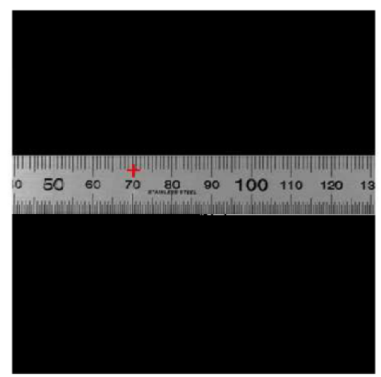


Camera calibration

If we select 2 known graduations, we can obtain the pixel to meters conversion.

```
% graphically select 2 points
[ip,jp] = ginput(2)
```



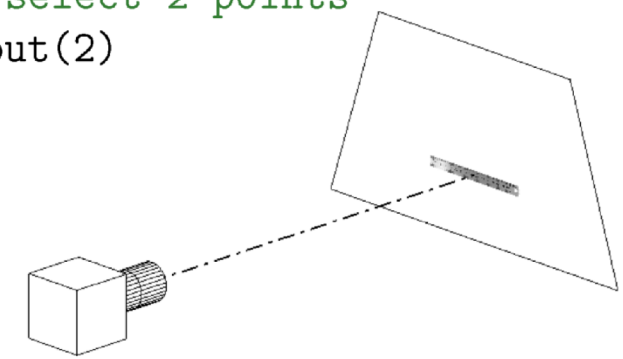


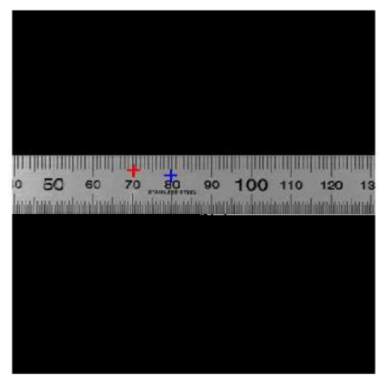
[ip(1), jp(1)]

Camera calibration

If we select 2 known graduations, we can obtain the pixel to meters conversion.

```
% graphically select 2 points
[ip,jp] = ginput(2)
```

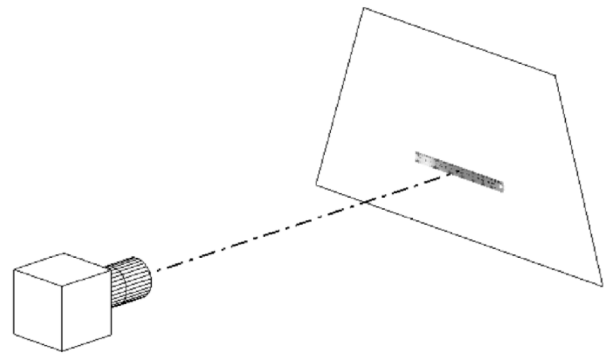


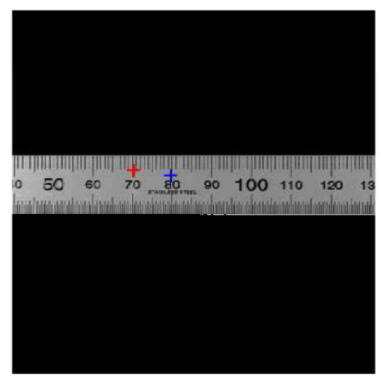


[ip(1), jp(1)]
[ip(2), jp(2)]

Camera calibration

$$\text{pixels m}^{-1} = \frac{ip(2) - ip(1)}{0.01}$$

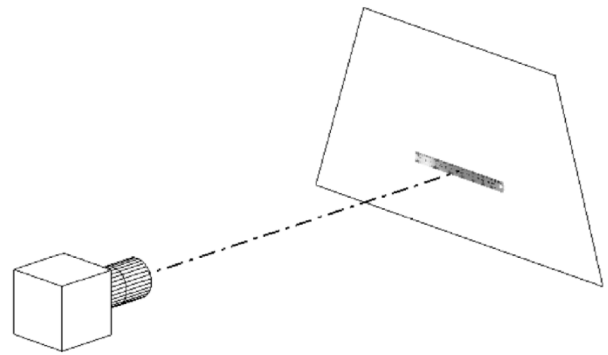


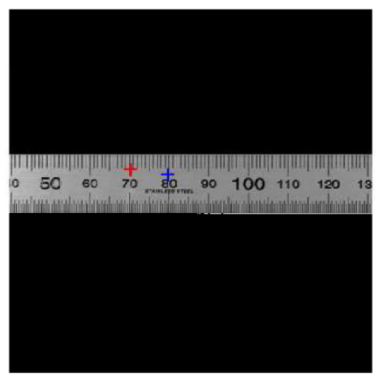


[ip(1), jp(1)]
[ip(2), jp(2)]

Camera calibration

However, this approach assumes no lens distortion, or parallax problems.

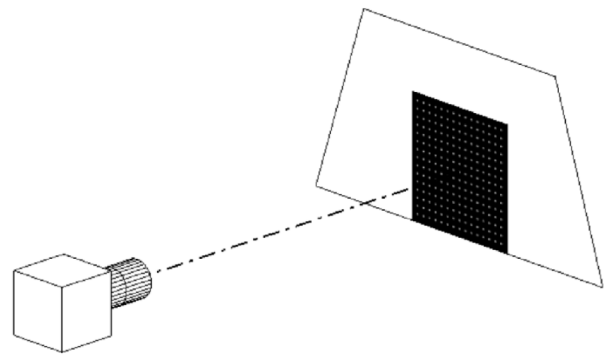




[ip(1), jp(1)]
[ip(2), jp(2)]

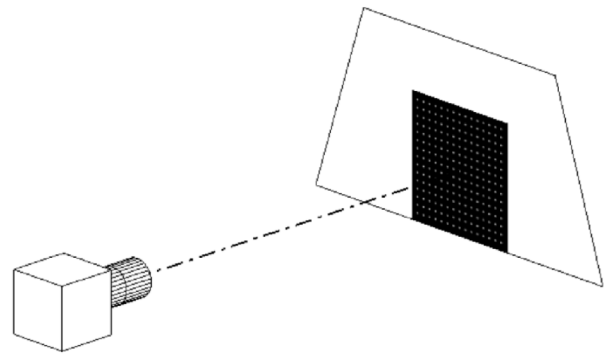
Camera calibration

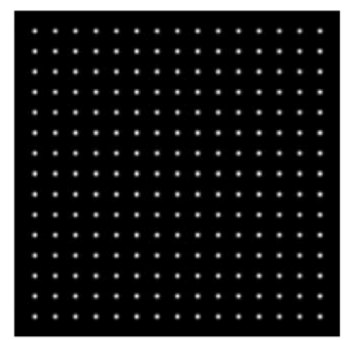
It is far preferable to use a 2D calibration target.



Camera calibration

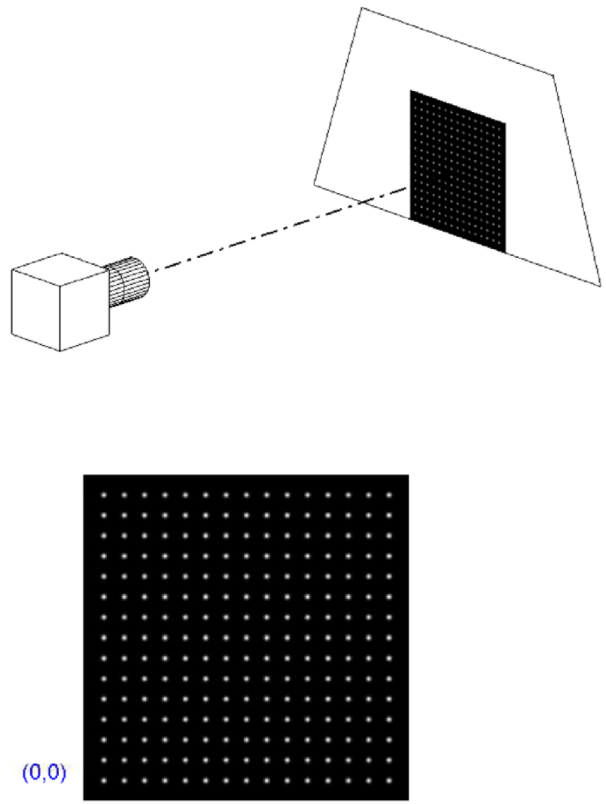
It is far preferable to use a 2D calibration target.





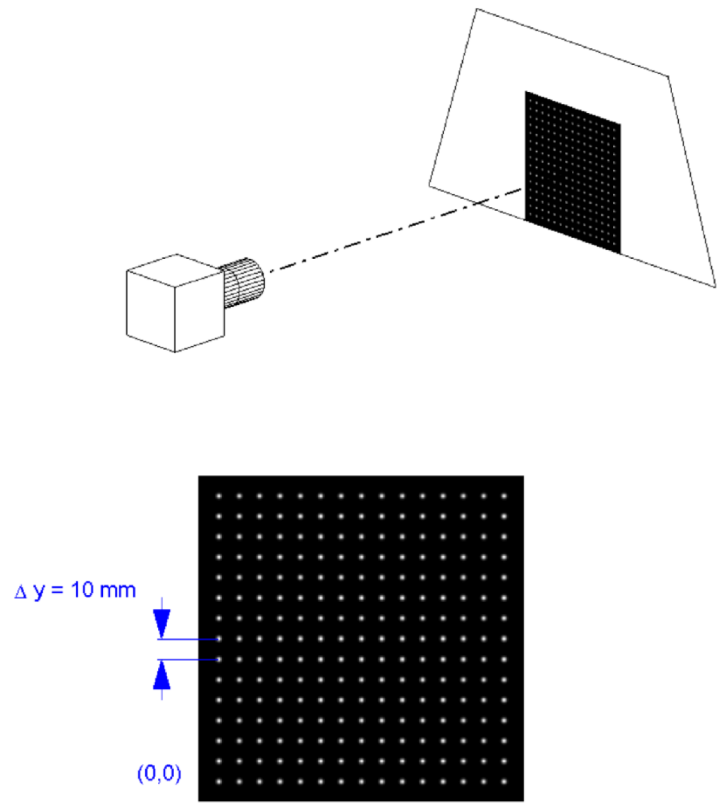
Camera calibration

We will know the real space coordinates of all the dots in the calibration image.



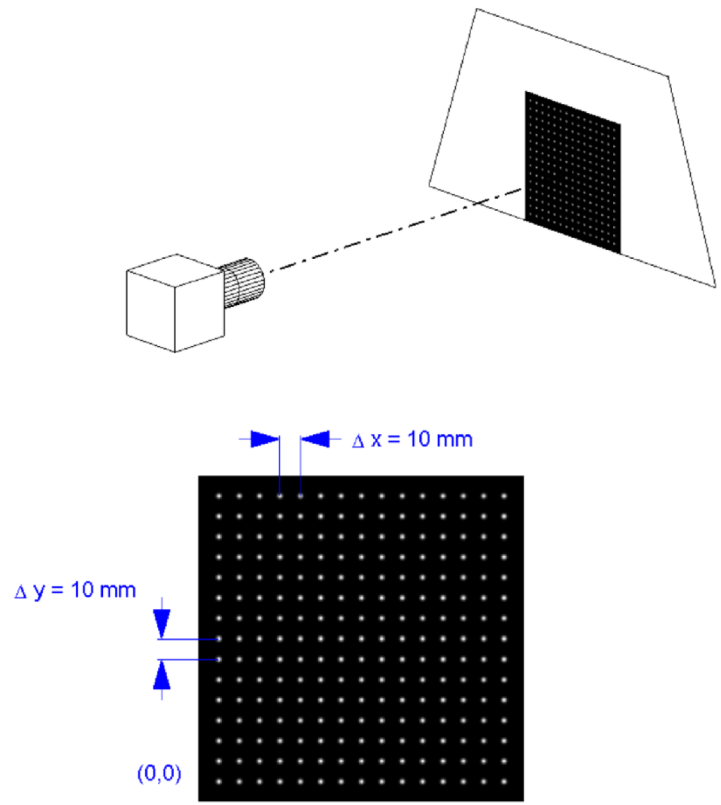
Camera calibration

We will know the real space coordinates of all the dots in the calibration image.



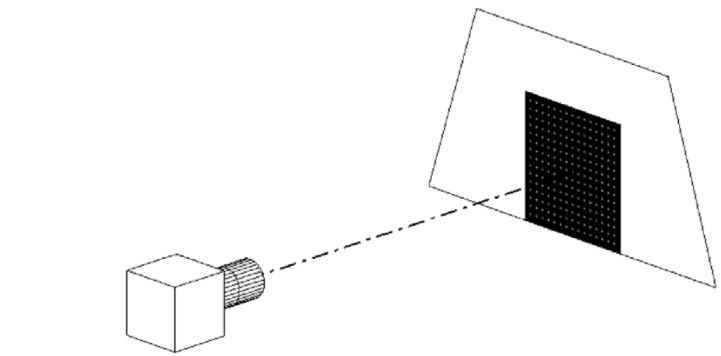
Camera calibration

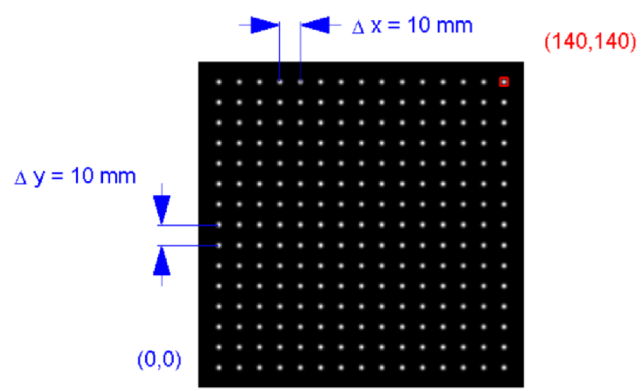
We will know the real space coordinates of all the dots in the calibration image.



Camera calibration

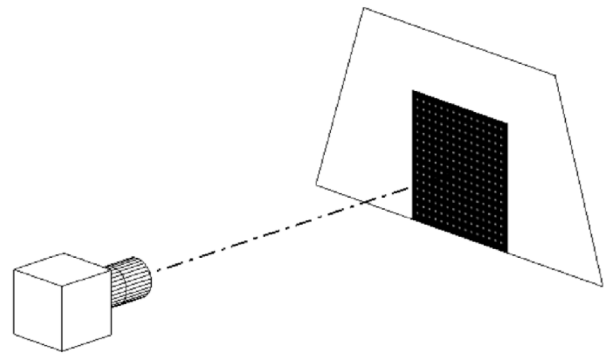
We will know the real space coordinates of all the dots in the calibration image.

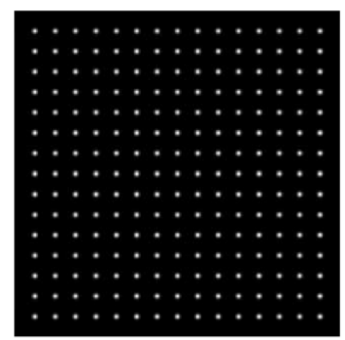




Camera calibration

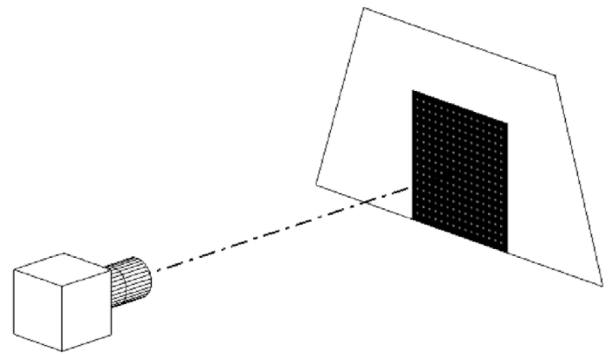
Then all we need to do is find the (i,j) pixel coordinates for all the dots in the image.

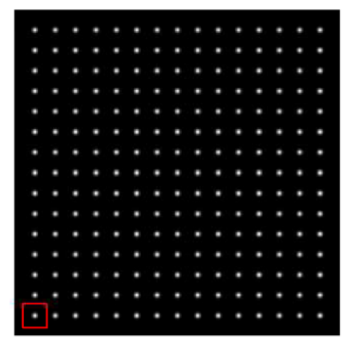




Camera calibration

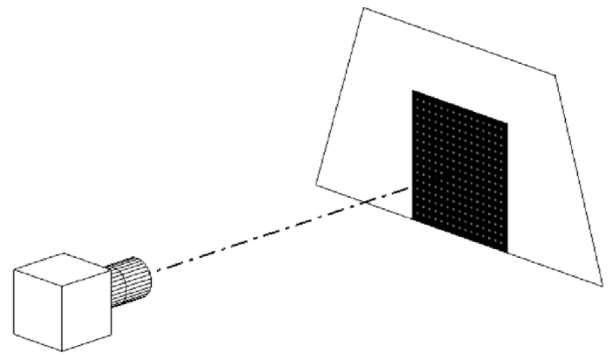
Select a template from the calibration image, and cross-correlate this template to find all the calibration dots.

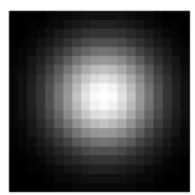


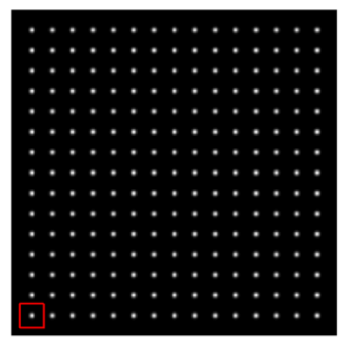


Camera calibration

Select a template from the calibration image, and cross-correlate this template to find all the calibration dots.

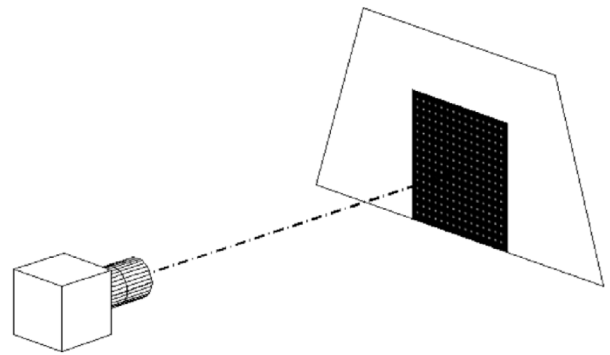


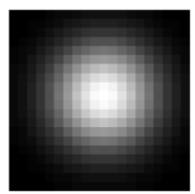


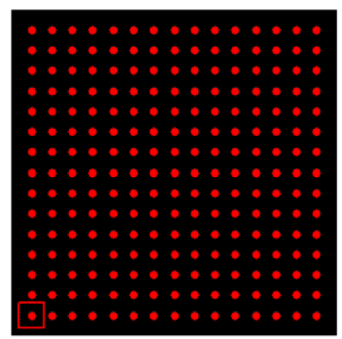


Camera calibration

Select a template from the calibration image, and cross-correlate this template to find all the calibration dots (i_c, j_c).



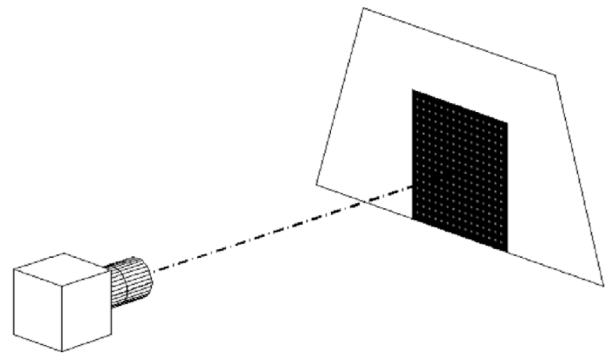


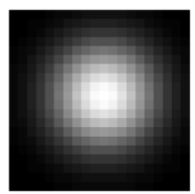


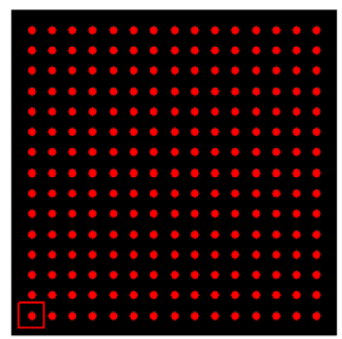
(i_c, j_c)

Camera calibration

We can now solve for the transformations f_x and f_y (typically use polynomial surfaces).



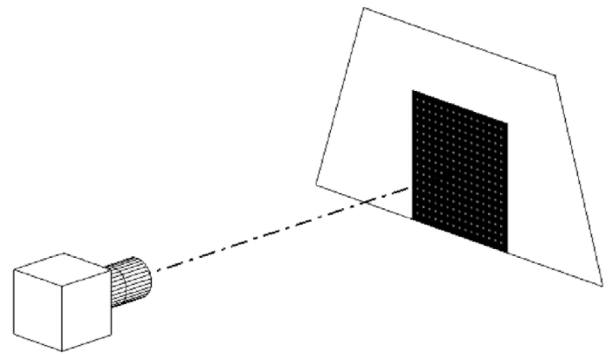


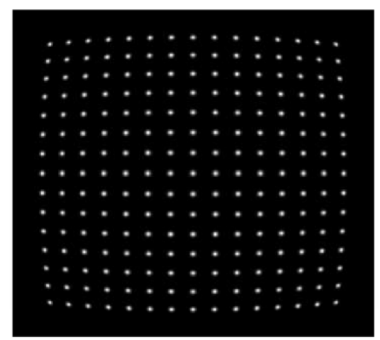


$$x = f_x(i_c, j_c)$$
$$y = f_y(i_c, j_c)$$

Camera calibration

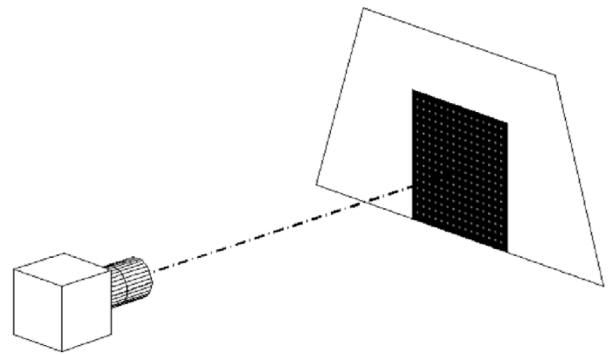
This is particularly useful where we have barrel distortion from the lens.

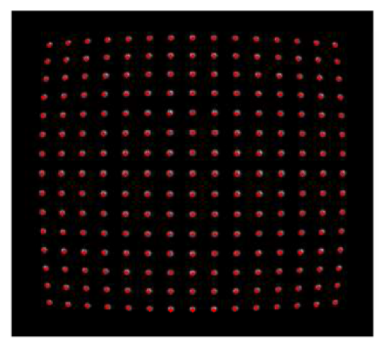




Camera calibration

This is particularly useful where we have barrel distortion from the lens!.

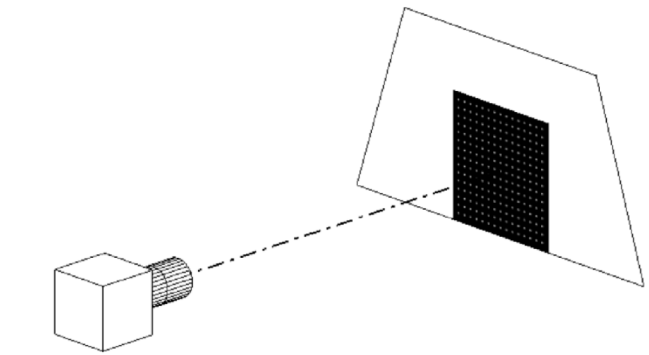




$$x = f_x(i_c, j_c)$$
$$y = f_y(i_c, j_c)$$

Camera calibration

This is particularly useful where we have barrel distortion from the lens.



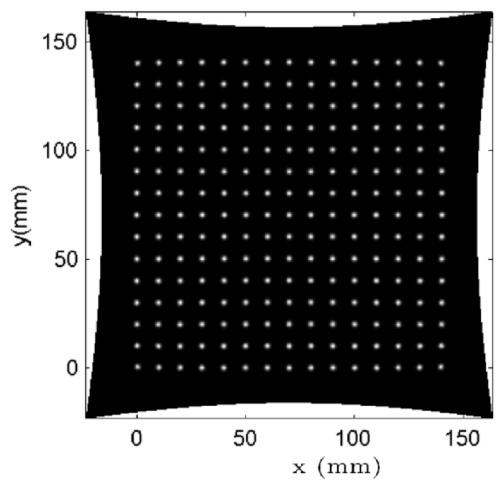
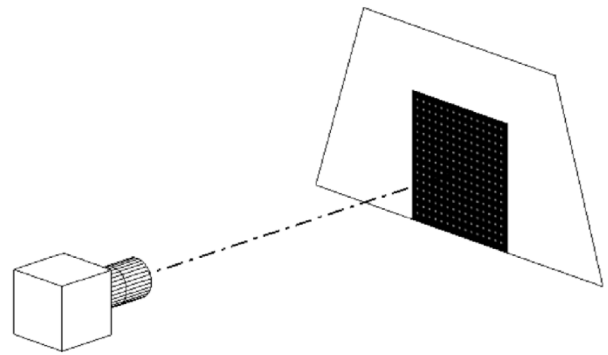
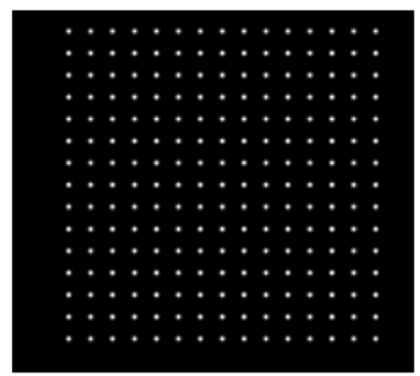


image transformed into real space

Camera calibration

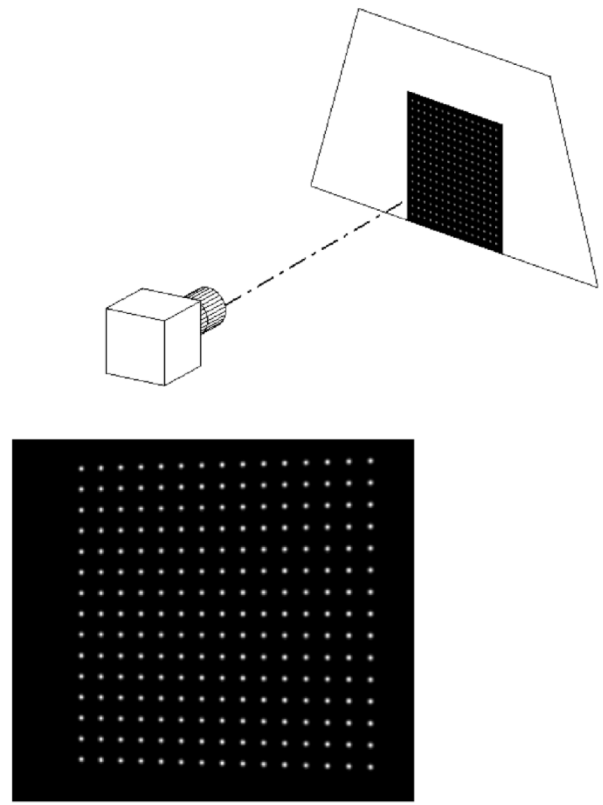
and also where we have parallax distortion due to camera axis not being orthogonal to the laser sheet.





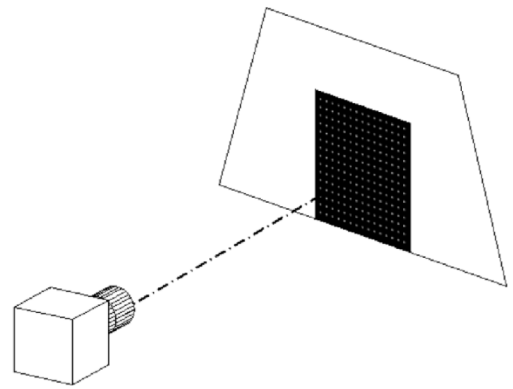
Camera calibration

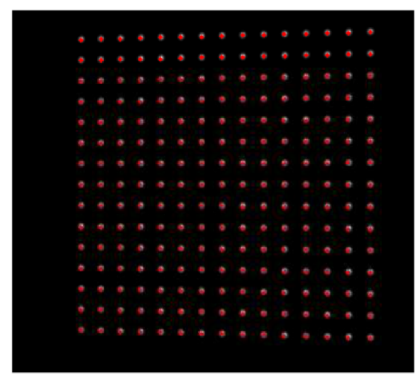
and also where we have parallax distortion due to camera axis not being orthogonal to the laser sheet.



Camera calibration

and also where we have parallax distortion due to camera axis not being orthogonal to the laser sheet.

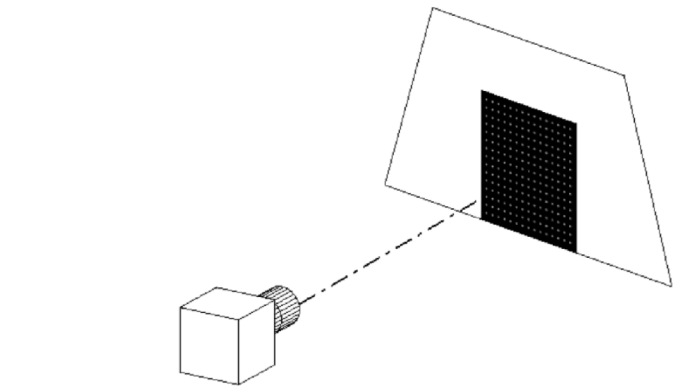




$$x = f_x(i_c, j_c)$$
$$y = f_y(i_c, j_c)$$

Camera calibration

and also where we have parallax distortion due to camera axis not being orthogonal to the laser sheet.



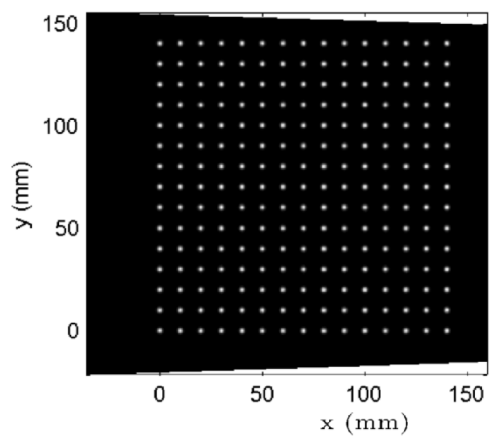
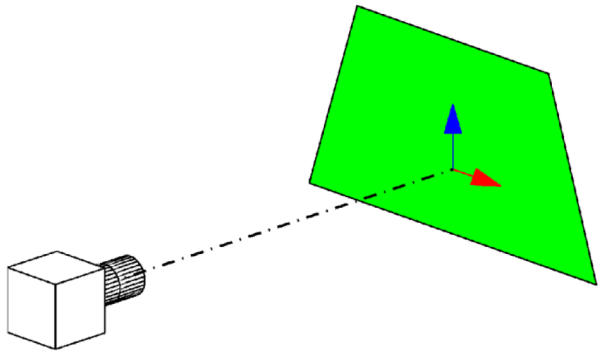


image transformed into real space

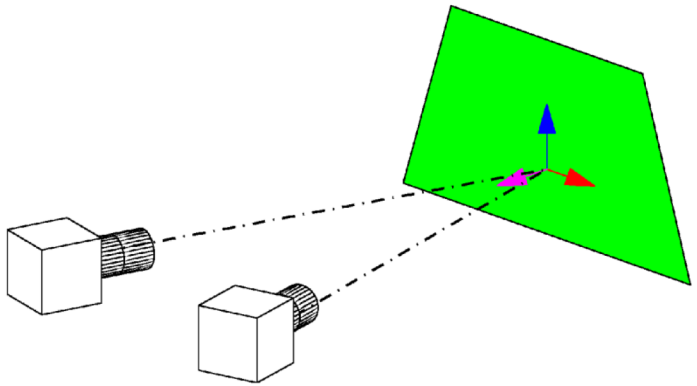
Camera calibration

With one camera, we can only measure the location of objects in x and y on a plane.



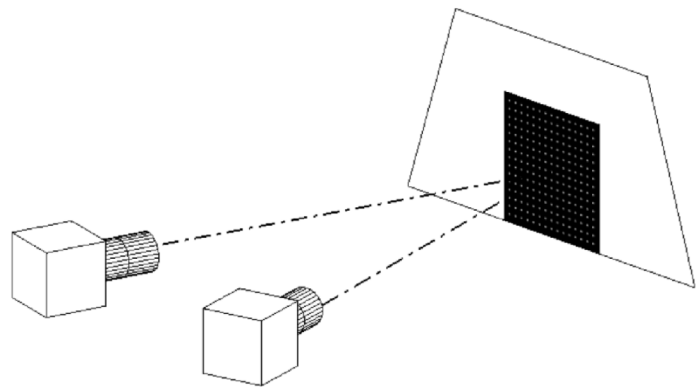
Camera calibration

By adding a second camera, we can determine the x, y and z location of objects.



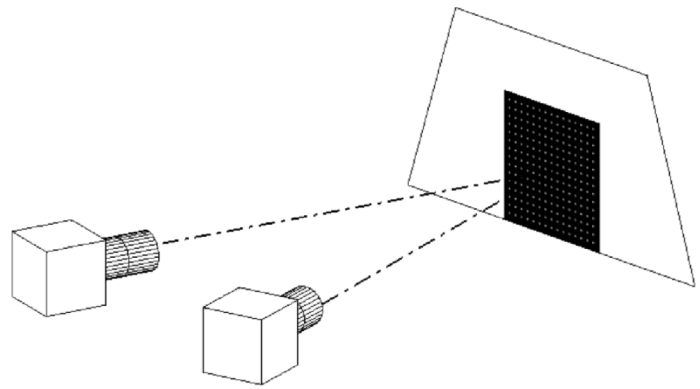
Camera calibration

We calibrate in a similar fashion, but we must now shift the calibration target in the out-of-plane direction.

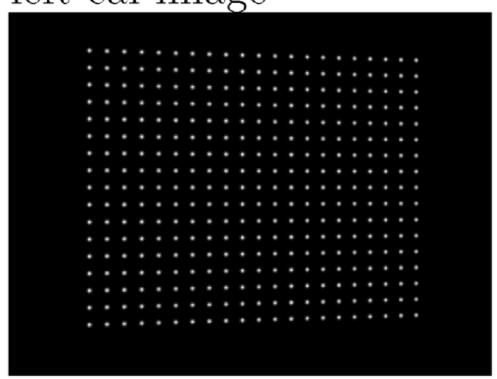


Camera calibration

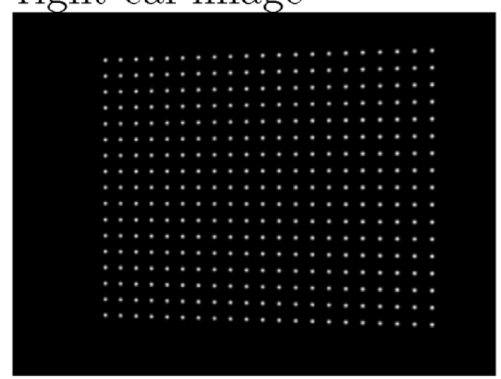
We calibrate in a similar fashion, but we must now shift the calibration target in the out-of-plane direction.



left cal image

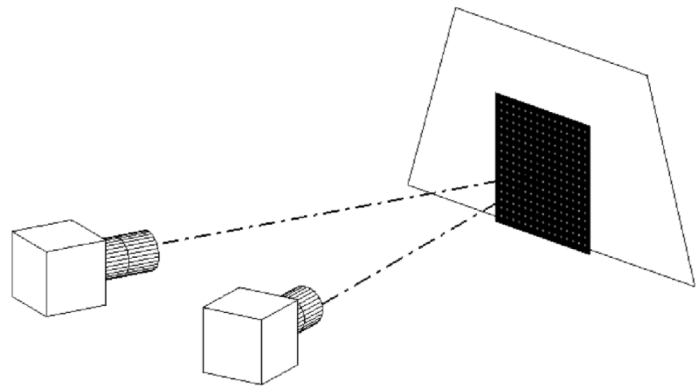


right cal image

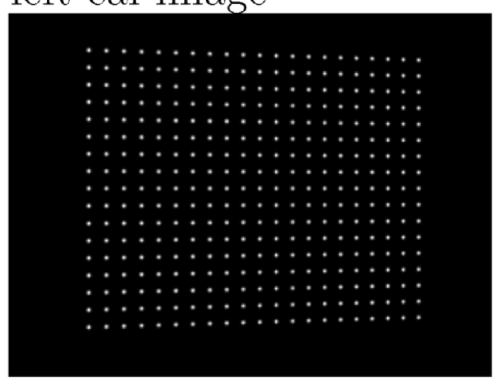


Camera calibration

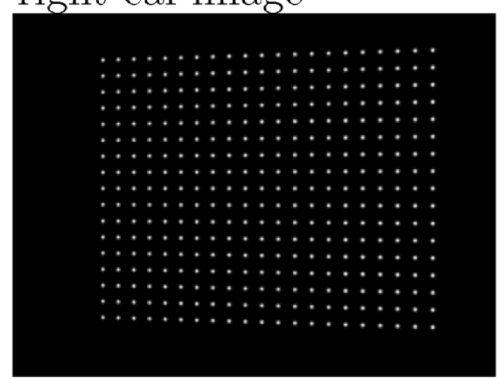
We calibrate in a similar fashion, but we must now shift the calibration target in the out-of-plane direction.



left cal image

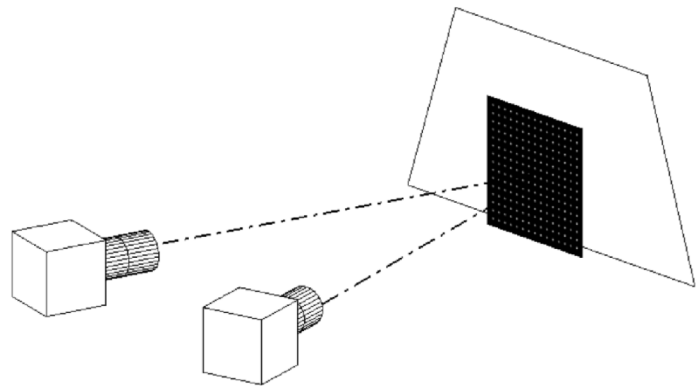


right cal image

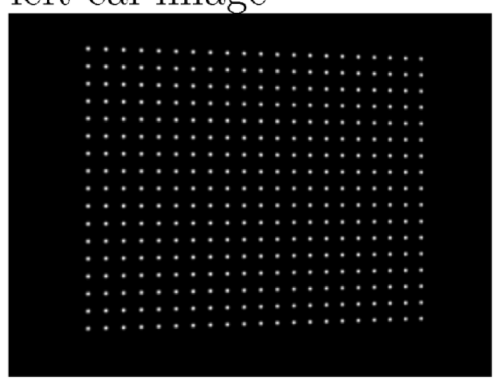


Camera calibration

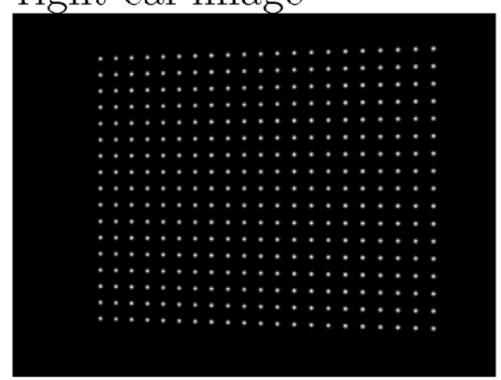
We calibrate in a similar fashion, but we must now shift the calibration target in the out-of-plane direction.



left cal image

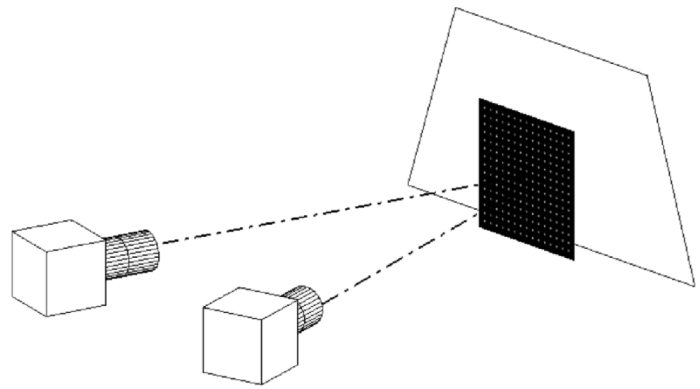


right cal image

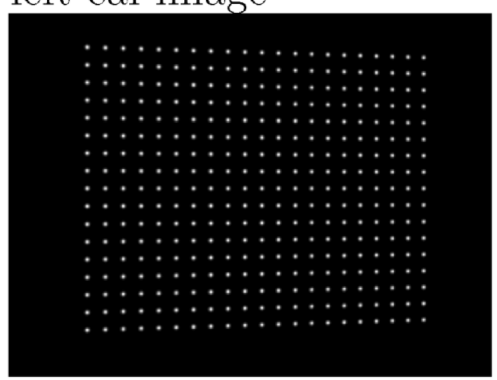


Camera calibration

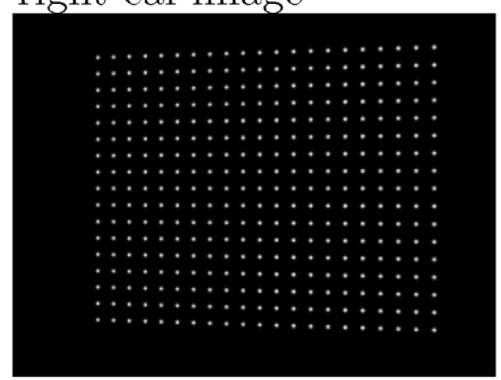
We calibrate in a similar fashion, but we must now shift the calibration target in the out-of-plane direction.



left cal image

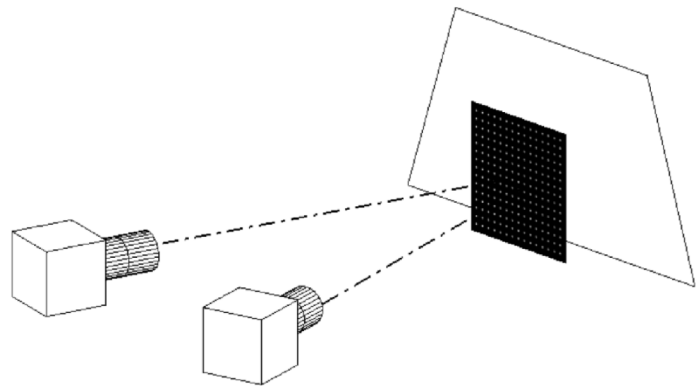


right cal image

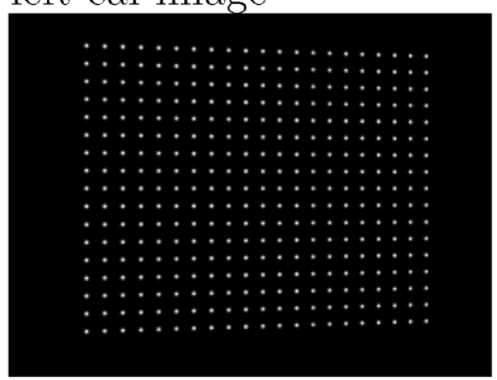


Camera calibration

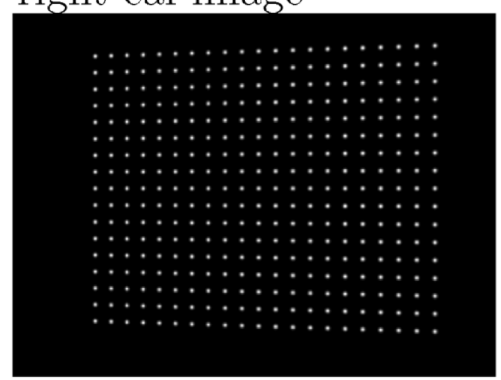
We calibrate in a similar fashion, but we must now shift the calibration target in the out-of-plane direction.



left cal image

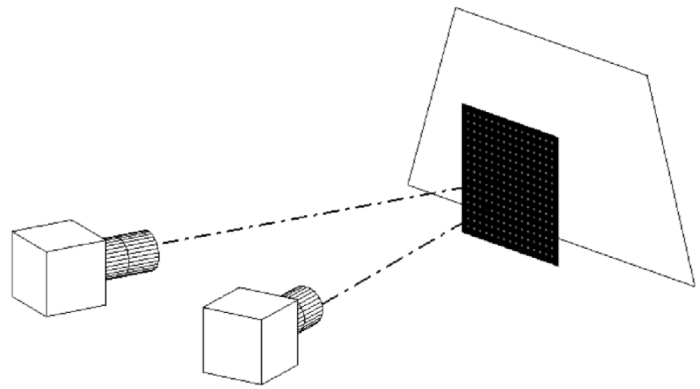


right cal image

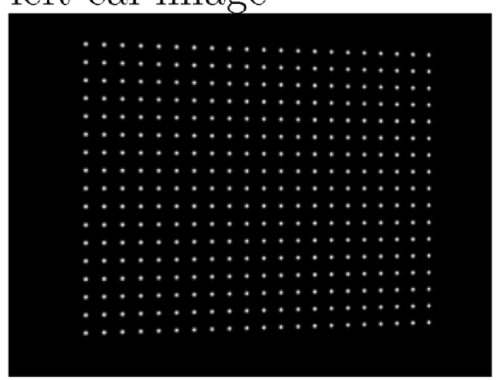


Camera calibration

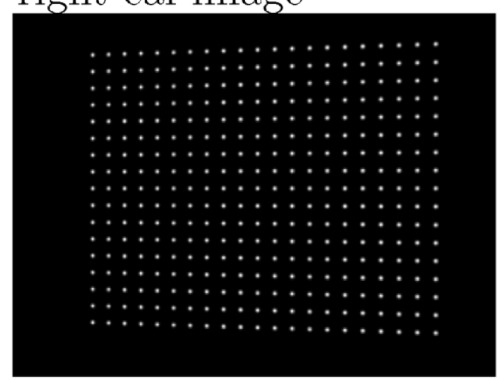
We calibrate in a similar fashion, but we must now shift the calibration target in the out-of-plane direction.



left cal image

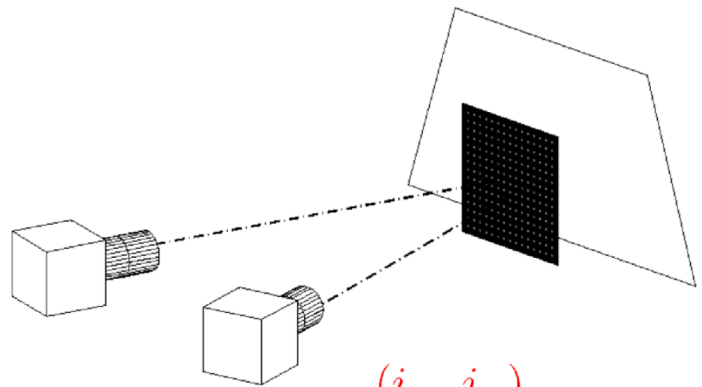


right cal image

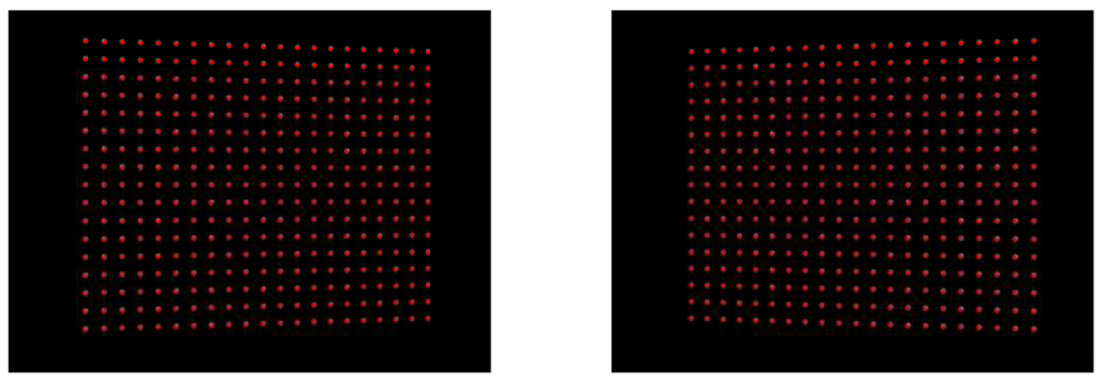


Camera calibration

We can use cross-correlation to find all of the dots in the left and right calibration images.

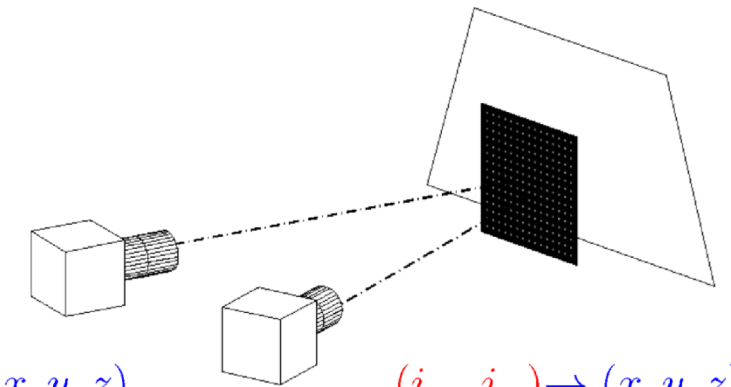


(i_{c_l}, j_{c_l}) (i_{c_r}, j_{c_r})

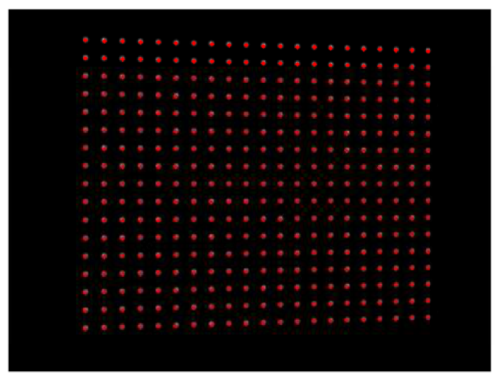


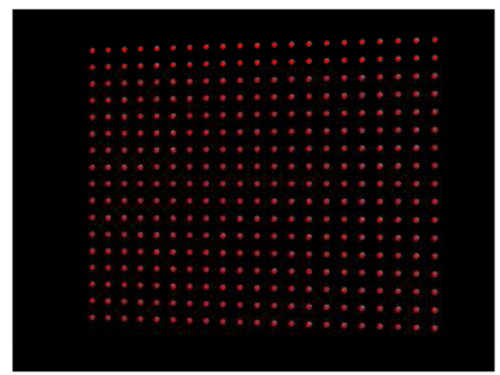
Camera calibration

And we know that each of those points corresponds to a particular (x,y,z) point in real space (x and y is the dot spacing on the target, and we shift the target to different z).



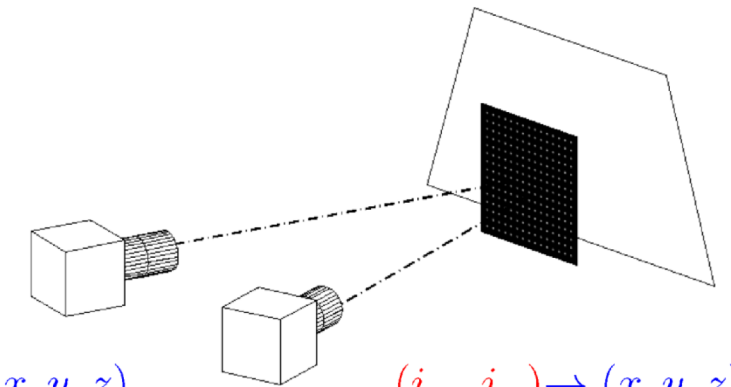
$(i_{c_l}, j_{c_l}) \rightarrow (x, y, z)$ $(i_{c_r}, j_{c_r}) \rightarrow (x, y, z)$



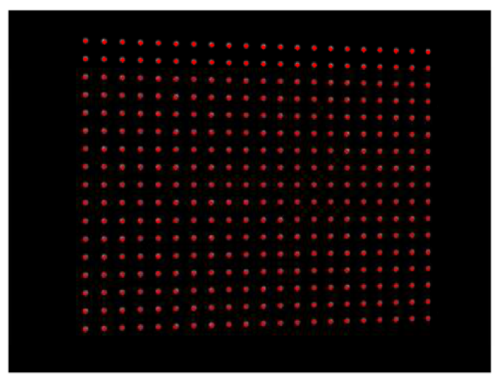


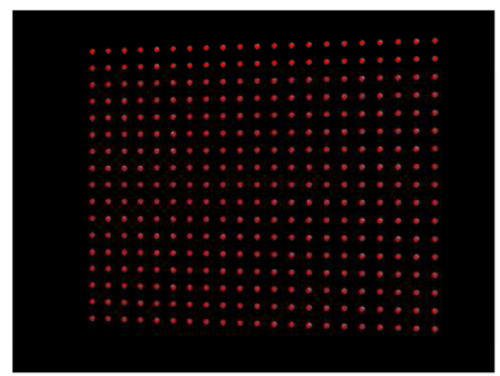
Camera calibration

Hence, each pixel pair in the left and right image (i_{c_l}, j_{c_l}) and (i_{c_r}, j_{c_r}) uniquely defines a position in 3D real space (x, y, z).



$(i_{c_l}, j_{c_l}) \rightarrow (x, y, z)$ $(i_{c_r}, j_{c_r}) \rightarrow (x, y, z)$

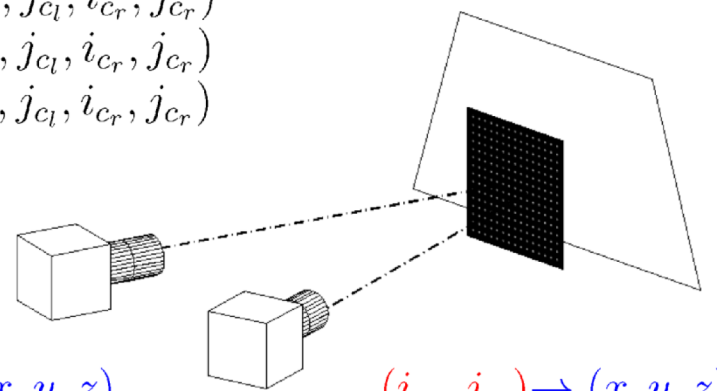




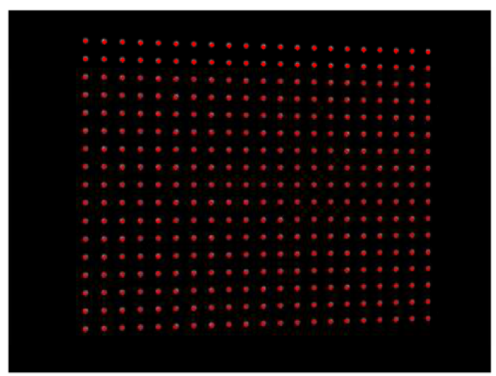
Camera calibration

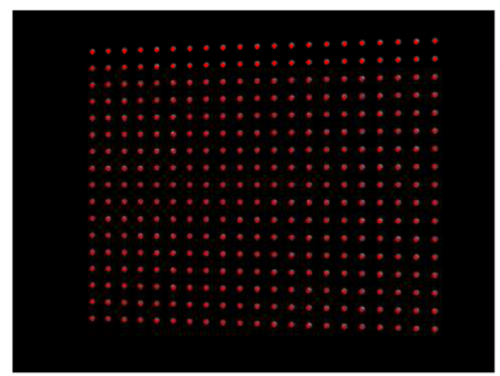
We produce a calibration fit of the following form (a four-dimensional surface fit) .

$$x = f_x(i_{c_l}, j_{c_l}, i_{c_r}, j_{c_r})$$
$$y = f_y(i_{c_l}, j_{c_l}, i_{c_r}, j_{c_r})$$
$$z = f_z(i_{c_l}, j_{c_l}, i_{c_r}, j_{c_r})$$



$(i_{c_l}, j_{c_l}) \rightarrow (x, y, z)$ $(i_{c_r}, j_{c_r}) \rightarrow (x, y, z)$

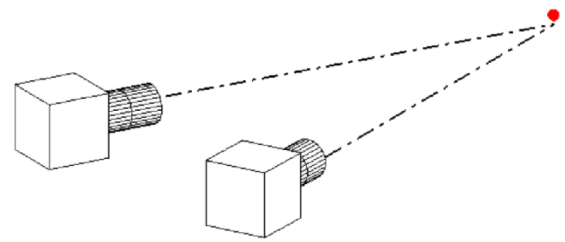




$$\begin{aligned} x \quad = \quad & A_1 + A_2 i_l + A_3 j_l + A_4 i_r + A_5 j_r + A_6 i_l j_l + A_7 i_l i_r + A_8 i_l j_r + A_9 j_l i_r \ldots \\ & + A_{10} j_l j_r + A_{11} i_r j_r + A_{12} i_l^2 + A_{13} j_l^2 + A_{14} i_r^2 + A_{15} j_r^2 \end{aligned}$$

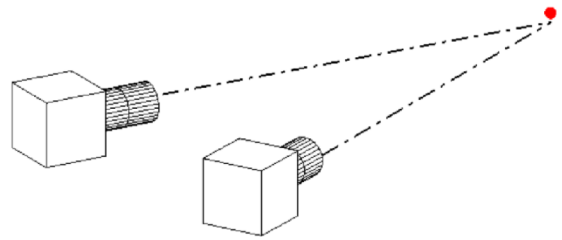
Camera calibration

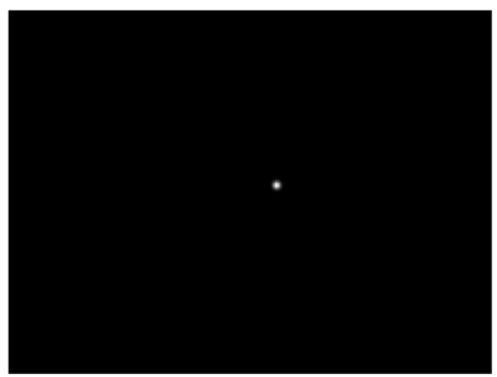
so if we have a common feature in the left and right image (the red dot) .

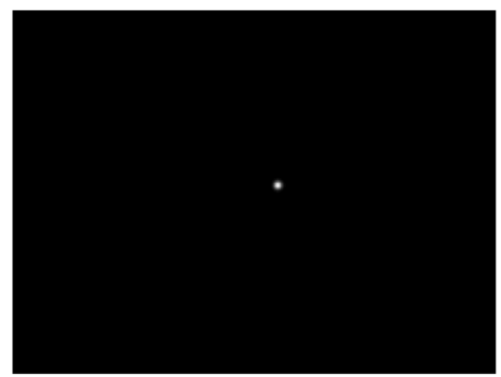


Camera calibration

we can back out its true location in real space from where it appears in pixel space within the left and right images.

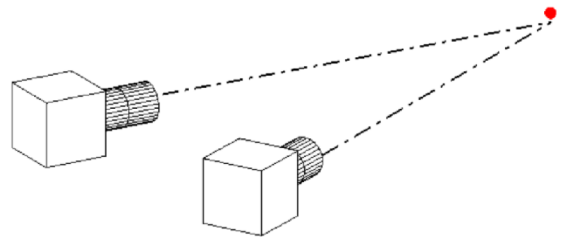


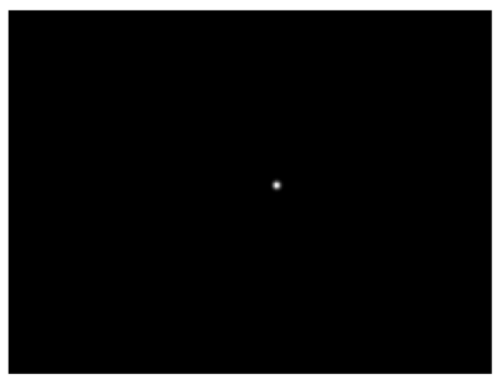


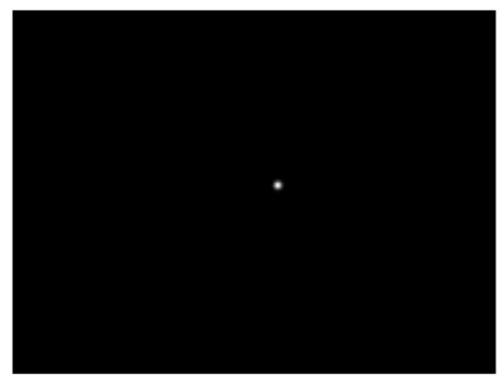


Camera calibration

we can back out its true location in real space from where it appears in pixel space within the left and right images.

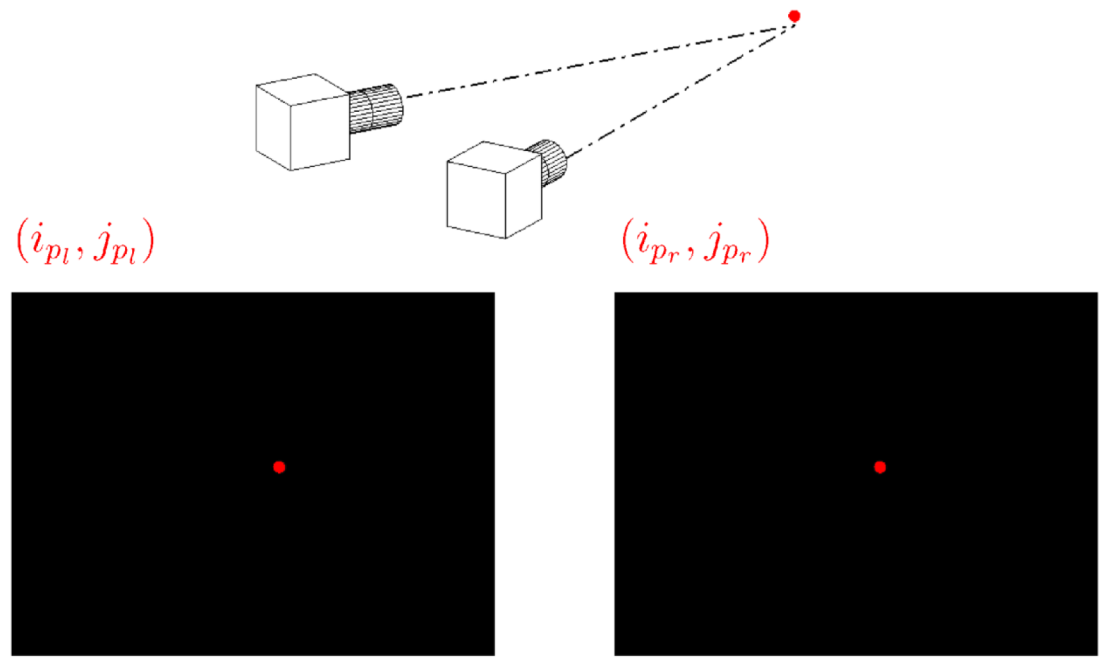






Camera calibration

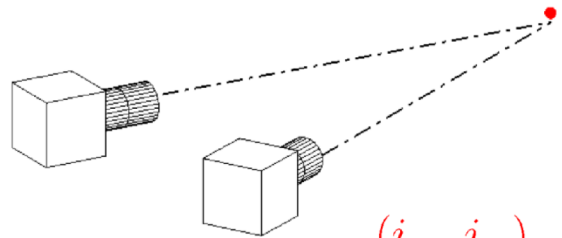
Find the location of the common feature in pixel space in the right and left image.



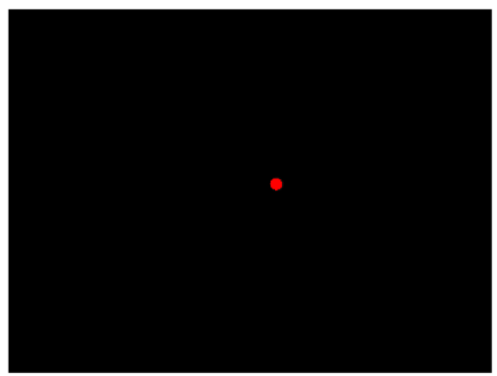
Camera calibration

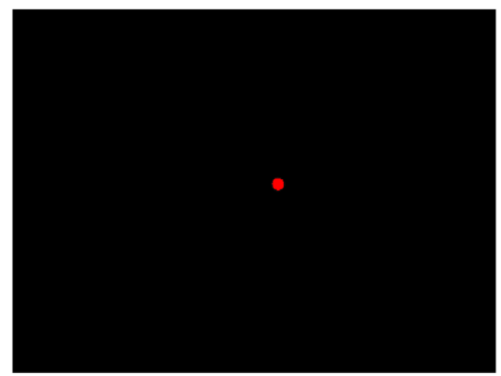
Find the location of the common feature in pixel space in the right and left image. And then plug into:

$$x = f_x(i_{p_l}, j_{p_l}, i_{p_r}, j_{p_r})$$
$$y = f_y(i_{p_l}, j_{p_l}, i_{p_r}, j_{p_r})$$
$$z = f_z(i_{p_l}, j_{p_l}, i_{p_r}, j_{p_r})$$



(i_{p_l}, j_{p_l}) (i_{p_r}, j_{p_r})





$$x = A_1 + A_2 i_l + A_3 j_l + A_4 i_r + A_5 j_r + A_6 i_l j_l + A_7 i_l i_r + A_8 i_l j_r + A_9 j_l i_r \ldots$$
$$+ A_{10} j_l j_r + A_{11} i_r j_r + A_{12} i_l^2 + A_{13} j_l^2 + A_{14} i_r^2 + A_{15} j_r^2$$