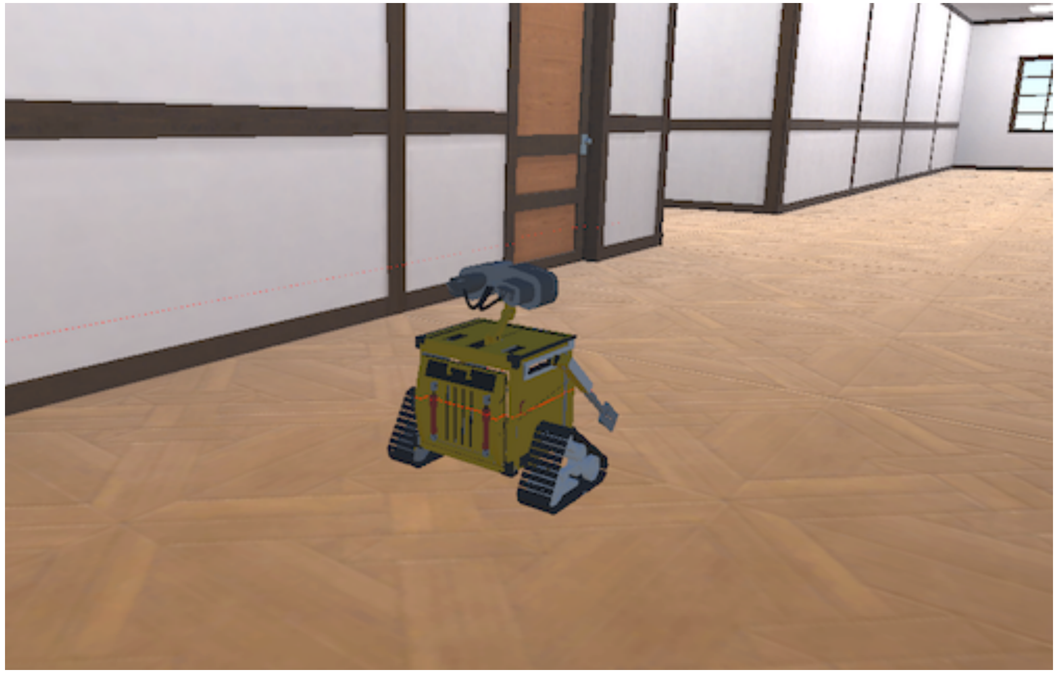
Omnidirectional Vision System

Goal of the Computer Vision?

By numbers computer has to understand the image



Human Vision

```
val(:,:,1) =

  Columns 1 through 19

   218   218   218   218   218   218   218   219   219   219   218   216   216   216   217   218   218   217   216
   217   218   218   218   218   217   217   219   219   219   219   218   217   215   215   216   216   216   215
   217   217   217   217   217   217   217   218   219   219   219   218   217   215   215   215   214   214   215
   217   217   217   217   217   217   217   218   218   219   219   218   217   215   215   216   216   215   215
   216   216   216   216   216   216   216   217   218   218   218   218   217   216   216   217   217   217   215
   216   216   216   216   216   216   216   217   218   218   218   217   217   217   217   217   217   217   216
   217   216   216   216   216   216   216   216   218   218   218   217   217   217   217   217   217   217   216
   216   216   216   216   215   215   215   216   217   217   217   217   217   217   217   217   217   217   217
   216   216   216   215   215   214   214   215   215   215   215   216   216   217   217   217   217   217   217
   216   217   216   215   214   214   214   214   214   214   215   215   216   216   217   217   217   217   217
   216   216   216   214   214   214   214   214   214   214   214   215   216   216   216   216   217   217   217
   215   216   215   214   213   212   212   213   214   214   214   215   215   216   216   216   216   216   216
   215   215   215   214   212   211   211   212   213   214   214   214   215   215   215   216   216   216   216
   214   214   214   214   214   213   212   213   214   214   214   214   214   214   215   215   215   216   216
   214   214   214   214   214   214   214   214   214   214   214   214   214   214   215   215   215   215   215
   215   214   214   214   214   214   214   214   214   214   214   214   214   214   215   215   215   215   215
   215   214   214   214   214   214   214   215   215   215   214   214   214   214   215   215   215   215   215
   215   215   214   214   214   214   215   215   215   215   214   214   214   214   214   215   215   215   215
   215   214   214   214   214   214   215   215   215   214   214   214   214   214   214   214   214   215   215
   215   214   213   213   214   215   215   214   214   213   213   213   213   213   213   214   214   214   215
   213   213   213   213   213   214   214   213   213   213   213   213   213   213   213   213   214   214   214
   212   212   212   212   212   212   213   213   213   213   213   213   213   212   213   213   213   214   214
   212   212   212   212   212   212   212   212   212   212   212   213   213   212   213   213   213   213   213
   212   212   212   212   211   212   212   212   212   212   213   213   213   213   213   213   213   213   213
   211   211   211   211   211   211   211   211   212   212   212   213   213   213   213   213   213   213   213
   211   211   211   211   211   211   211   211   211   212   212   213   213   213   213   213   213   213   213
   211   211   211   211   211   211   211   211   211   211   212   213   213   213   213   212   213   213   213
```

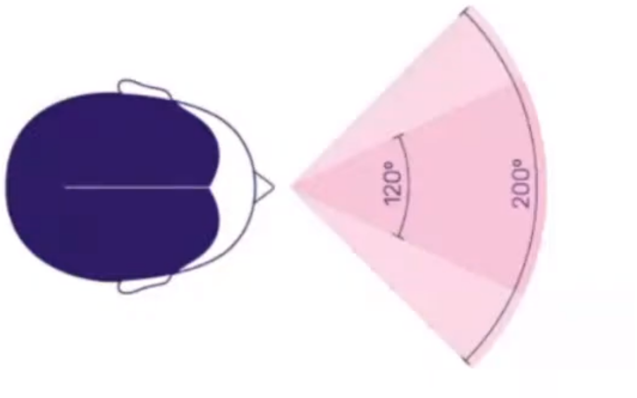
Computer Vision

Scene Understanding

The goal of the computer vision is to understands the world as humans do. The human vision perceives 120º horizontal field of view without eyes movements, with the eyes movements it's even more. Human vision is able to provide the 3d understanding of the scene as well

Human Visual Perceptions

- Field of View
- Depth Perception

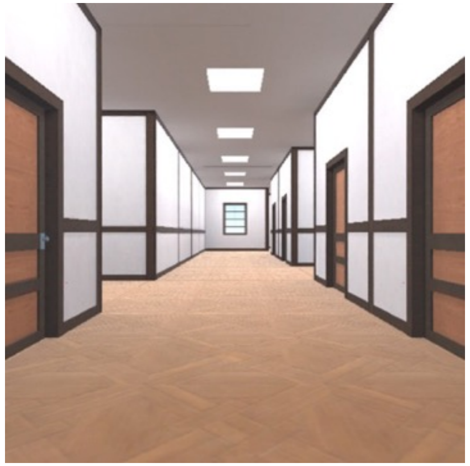


Consequently, the reduced field of view or the lack of depth information limits the goal of developing intelligent systems to match the performance and the robustness of human vision

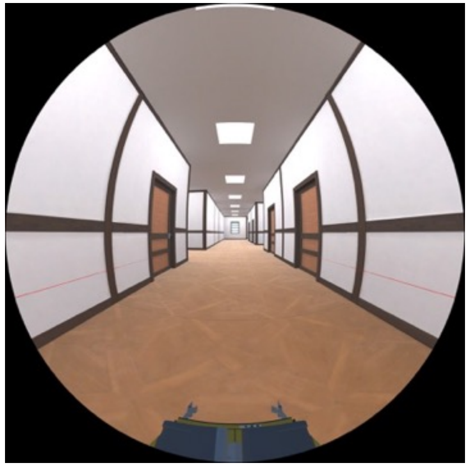
Scene Understanding

So, we observed that ultra wide angle lenses, e.g. fisheye lenses can be beneficial for different tasks. This is not surprising since the 180° field of view allows to include much more details about the scene. However, just by using a snapshot captured by a system included only one camera we cannot get the depth information.

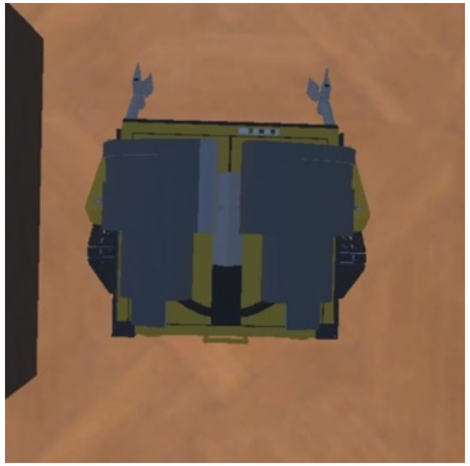
FOV: 60°



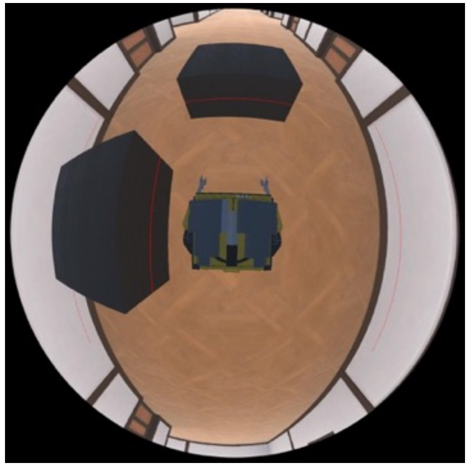
FOV: 180°



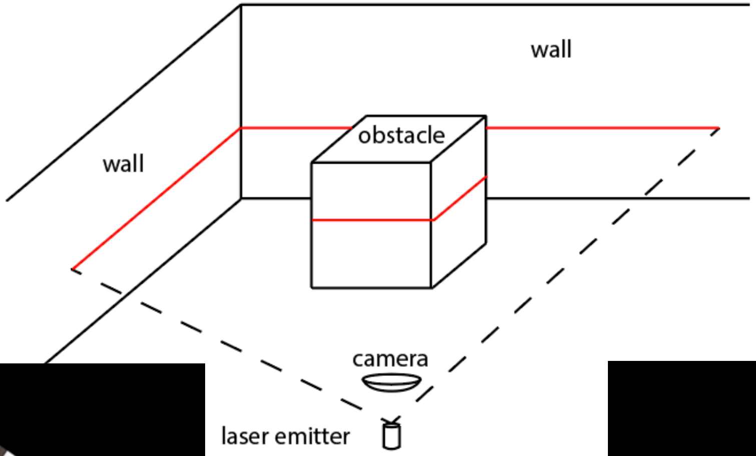
FOV: 60°

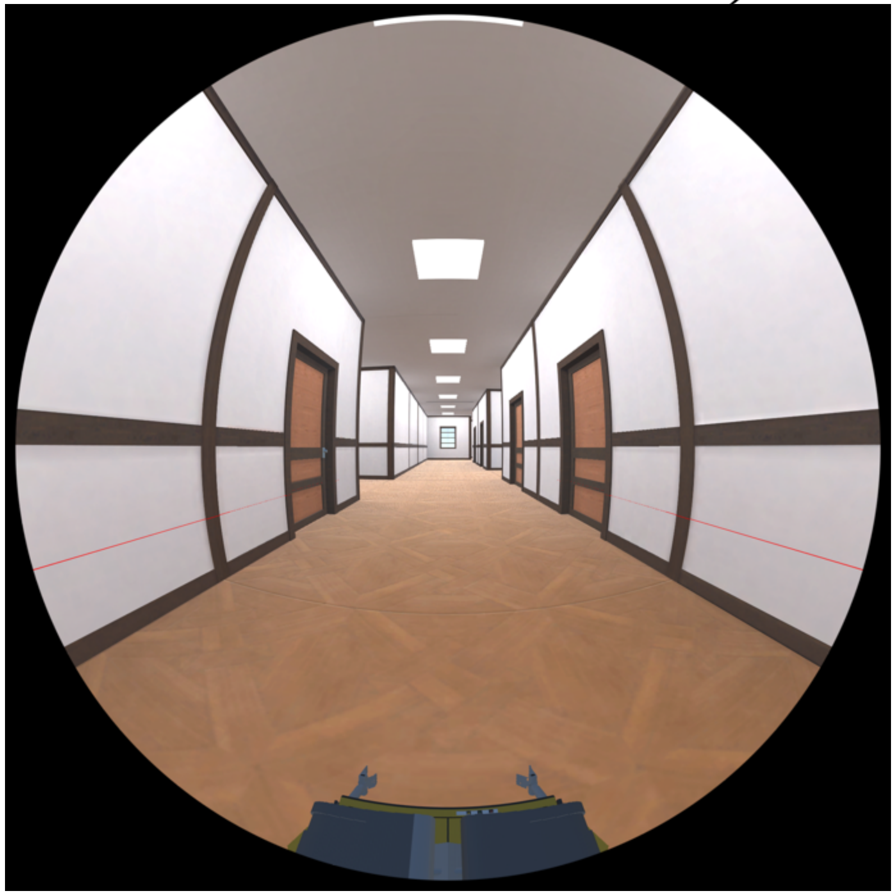


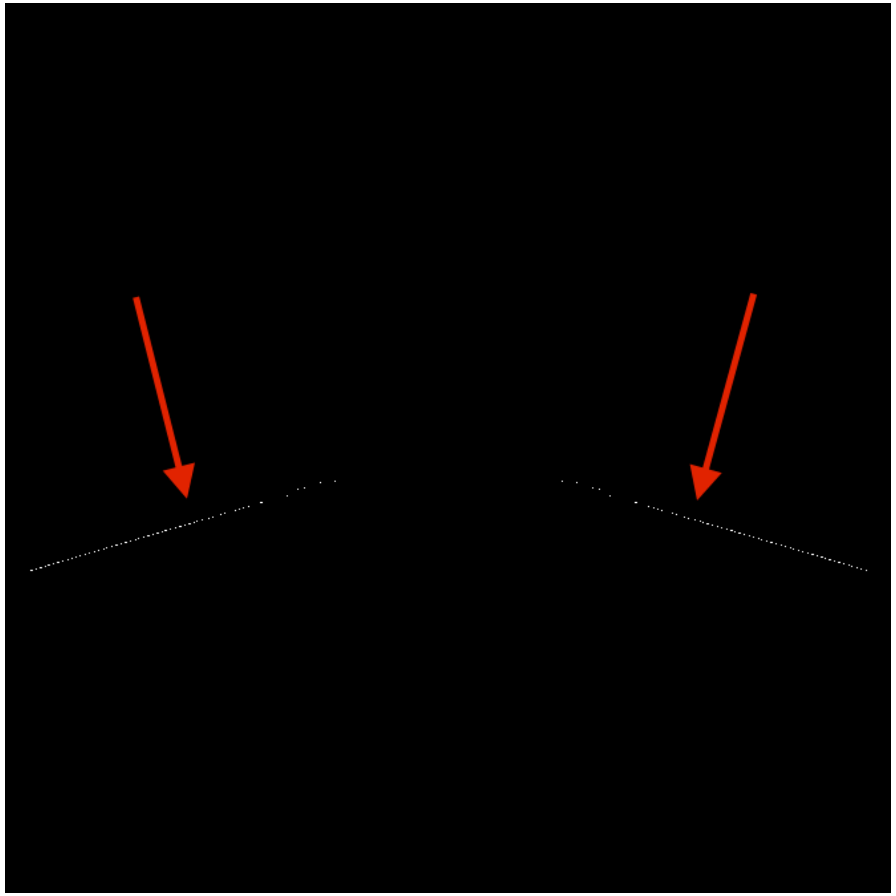
FOV: 180°



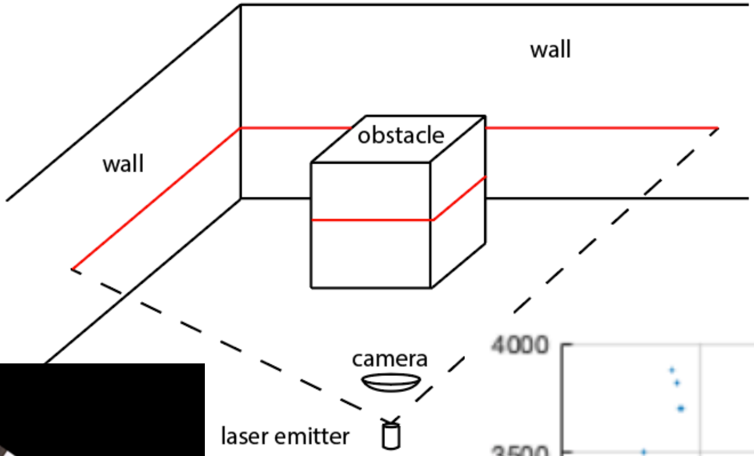
Metric information

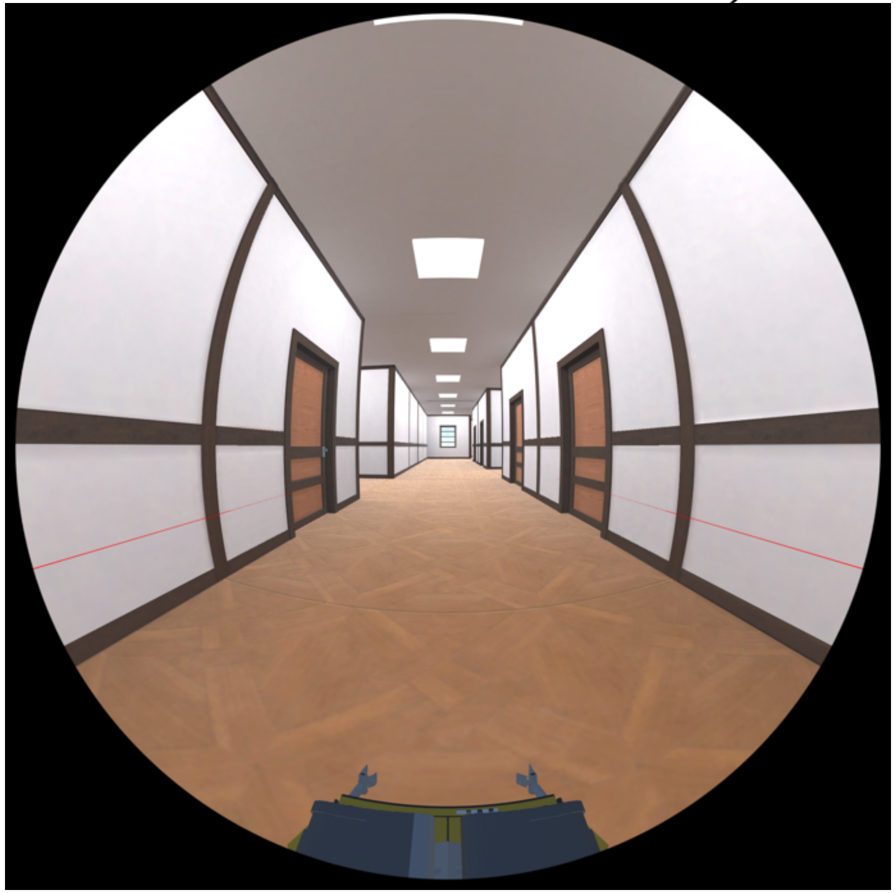


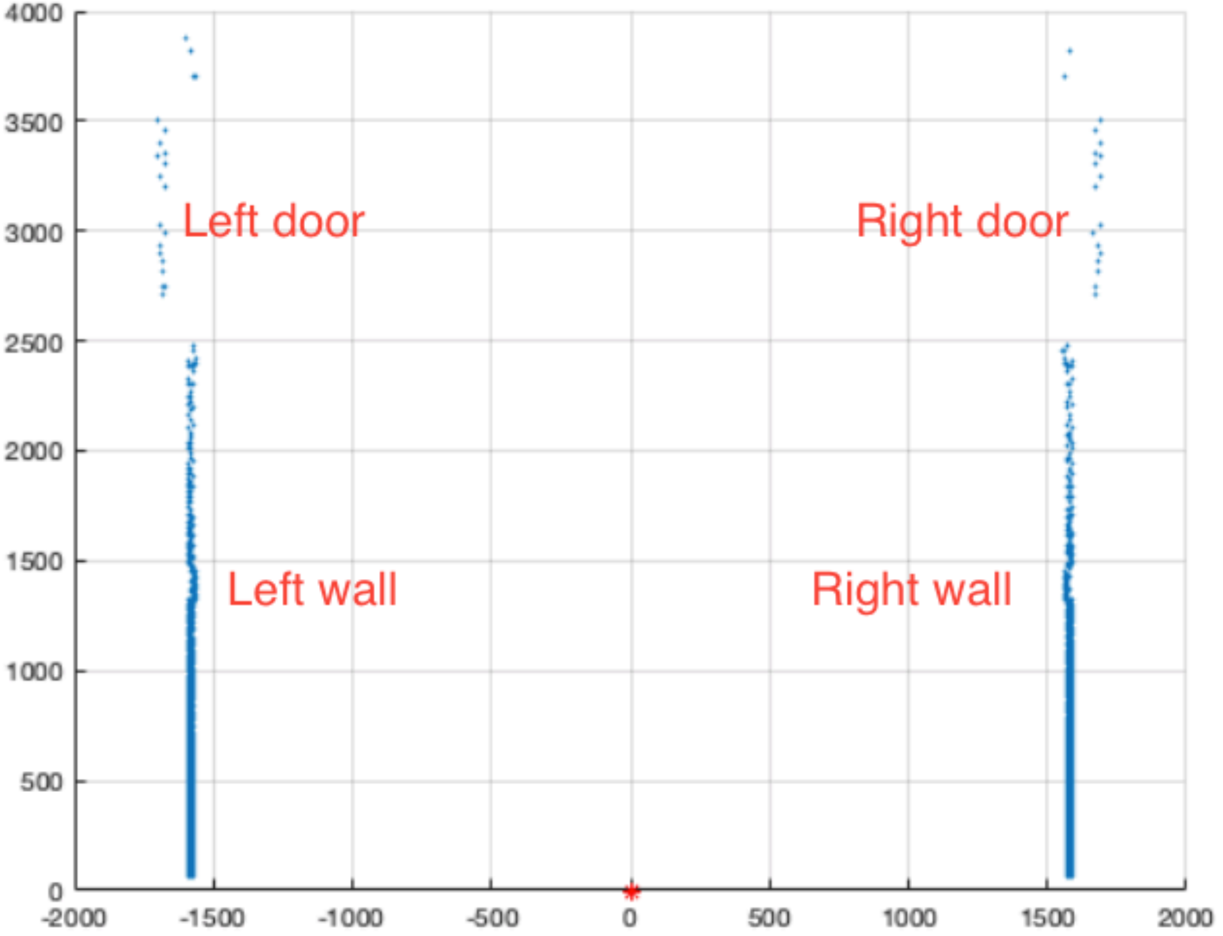


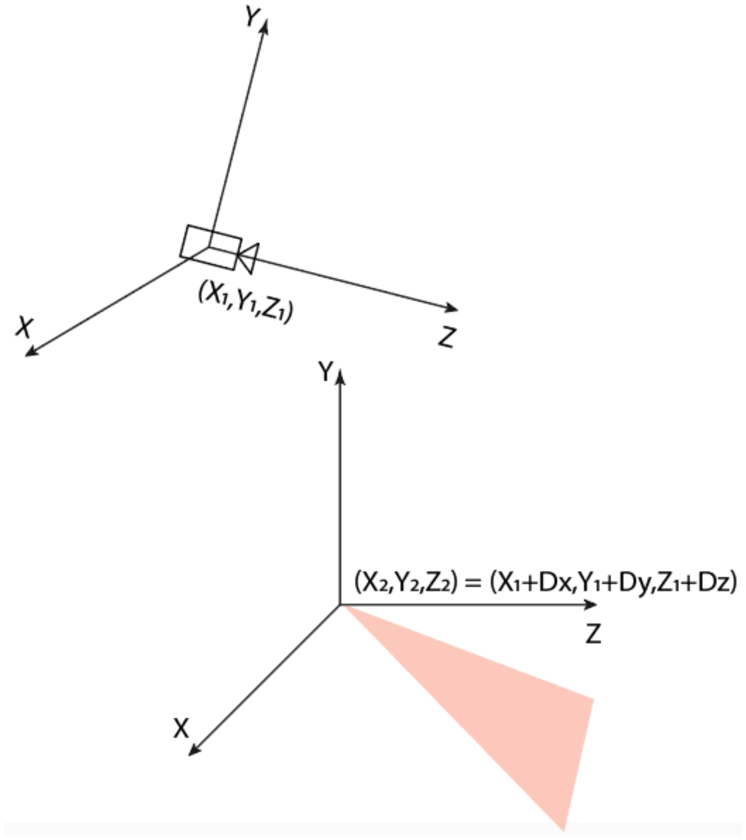


Metric information





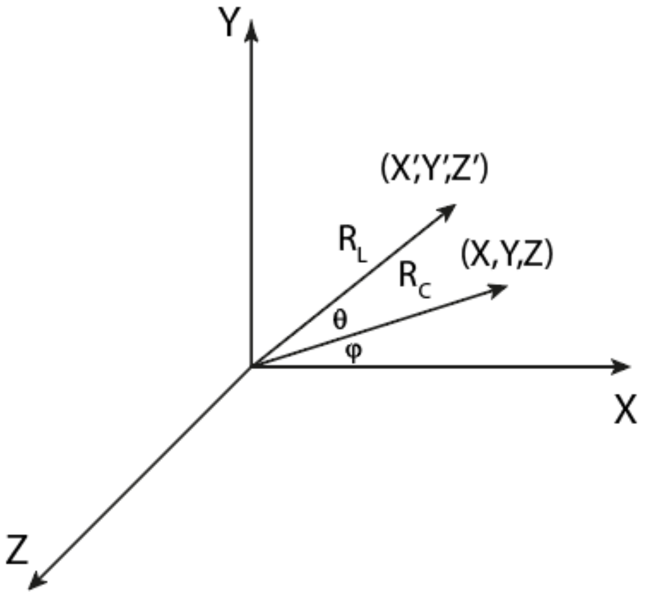




$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & D_x \\ 0 & 1 & 0 & D_y \\ 0 & 0 & 1 & D_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$$

If we multiply these matrixes, we will get exactly the same equations:

$$X_2 = X_1 + D_x$$
$$Y_2 = Y_1 + D_y$$
$$Z_2 = Z_1 + D_z$$



Let's consider rotation around Z-axis.

$$X = R_c cos(\varphi)$$
$$Y = R_c sin(\varphi)$$

$$X' = R_c cos(\theta+\varphi) = R_c cos(\theta)cos(\varphi) - R_c sin(\theta)sin(\varphi)$$
$$Y' = R_c sin(\theta+\varphi) = R_c sin(\theta)cos(\varphi) + R_c cos(\theta)sin(\varphi)$$

$$X' = Xcos(\theta) - Ysin(\theta)$$
$$Y' = Xsin(\theta) + Ycos(\theta)$$

In a matrix form we can write it as:

Z-axis: *X*-axis: *Y*-axis:

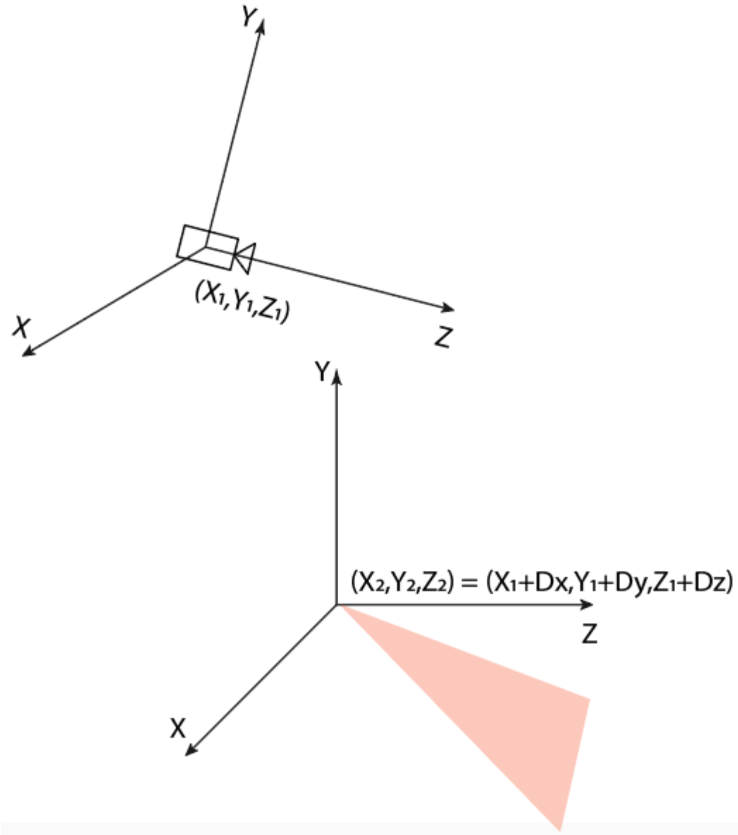
$$R_Z = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad R_X = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\gamma) & -sin(\gamma) \\ 0 & sin(\gamma) & cos(\gamma) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad R_Y = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Order matters:

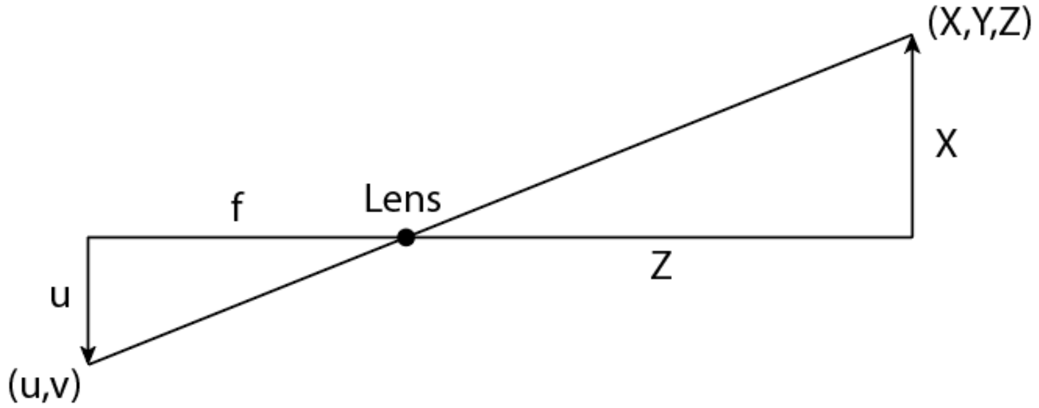
$$R = R_Z R_Y R_X = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\gamma) & -sin(\gamma) \\ 0 & sin(\gamma) & cos(\gamma) \end{bmatrix}$$

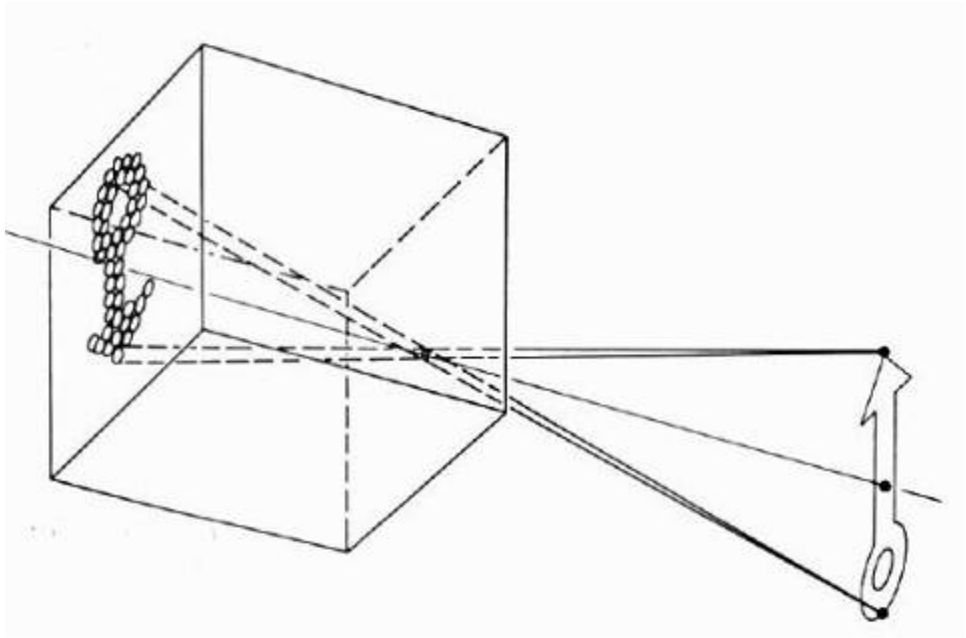
After multiplication:

$$R = \begin{bmatrix} cos(\theta)cos(\beta) & cos(\theta)sin(\beta)sin(\gamma) - sin(\theta)cos(\gamma) & cos(\theta)sin(\beta)cos(\gamma) + sin(\theta)sin(\gamma) \\ sin(\theta)cos(\beta) & sin(\theta)sin(\beta)sin(\gamma) + cos(\theta)cos(\gamma) & sin(\theta)sin(\beta)cos(\gamma) - cos(\theta)sin(\gamma) \\ -sin(\beta) & cos(\beta)sin(\gamma) & cos(\beta)cos(\gamma) \end{bmatrix}$$



$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} R_{12} R_{13} D_x \\ R_{21} R_{22} R_{23} D_y \\ R_{31} R_{32} R_{33} D_z \\ 0 \quad 0 \quad 0 \quad 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$$



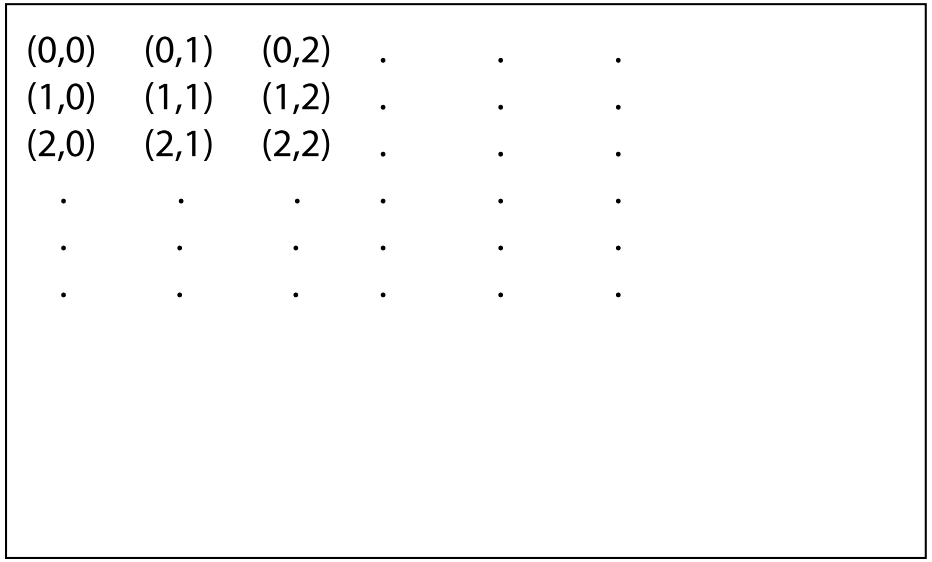


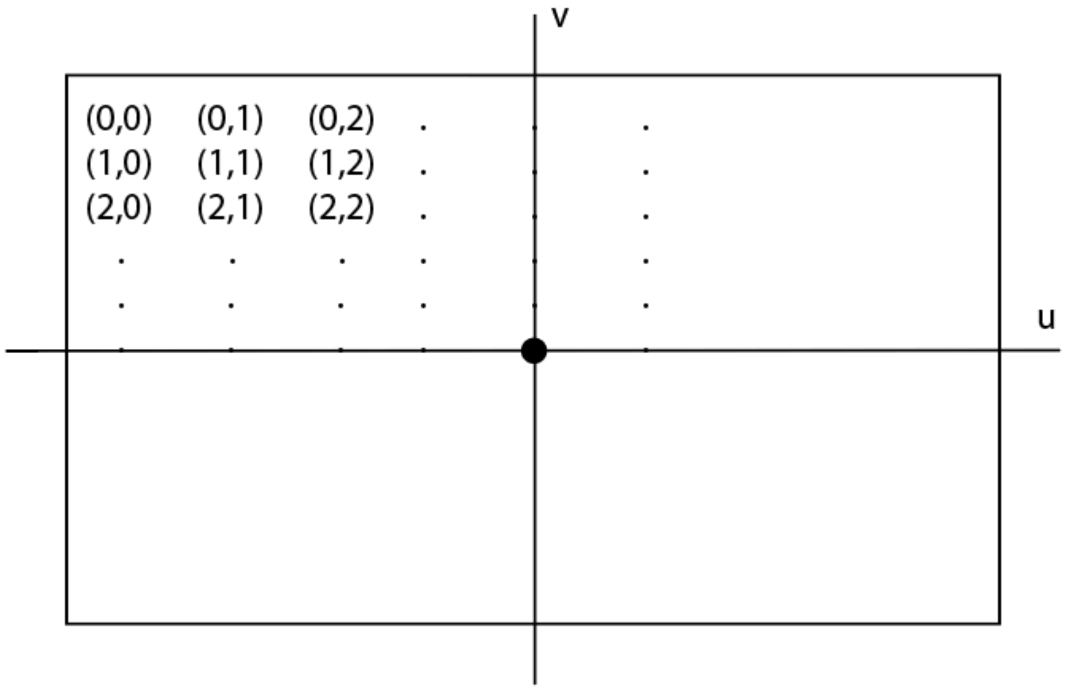
World coordinated are defined as (X,Y,Z), pixel coordinates as (u,v) and f – is the focal length. From the similar triangles we have:

$$u/_X = f/_Z$$
$$u = fX/_Z$$
$$v = fY/_Z$$

The relationship between image and world points also can be written in a matrix form:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/_Z \\ Y/_Z \\ 1 \end{bmatrix}$$

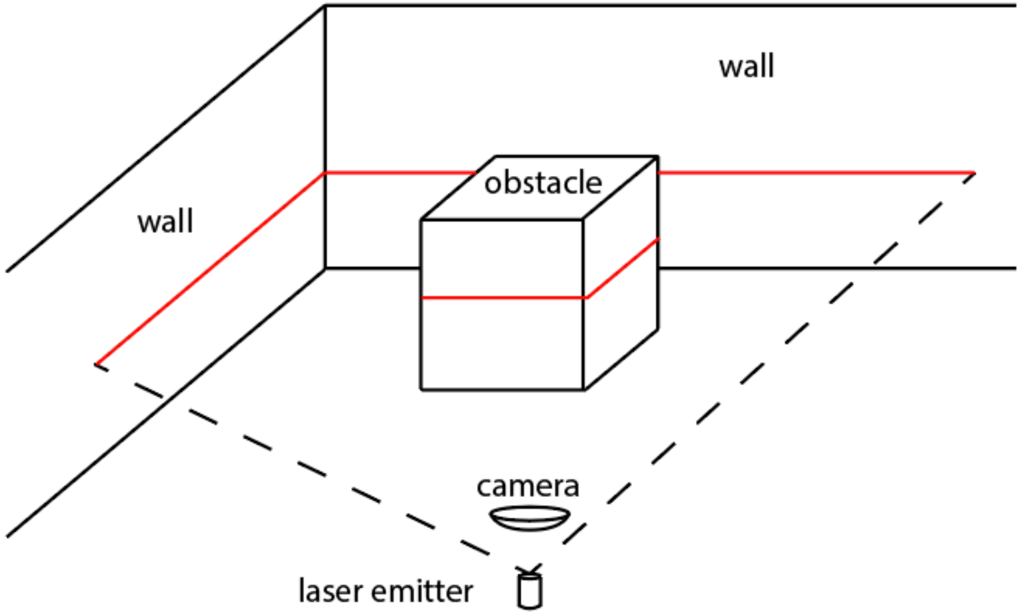


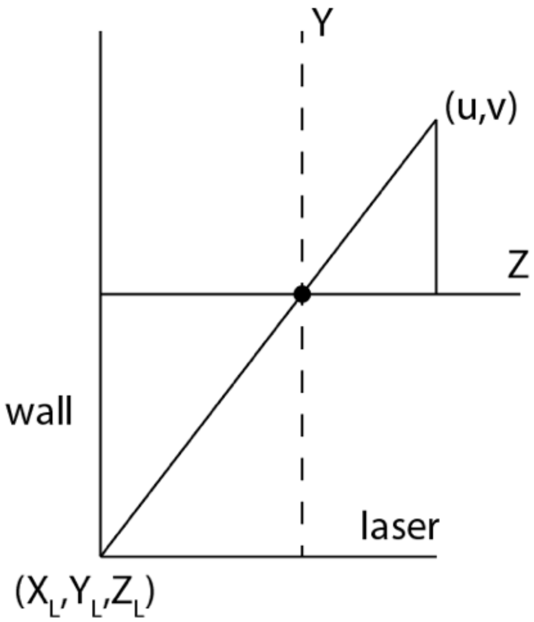


$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/_Z \\ Y/_Z \\ 1 \end{bmatrix}$$

$$u = {}^{fX}/_Z + u_0$$

$$v = {}^{fY}/_Z + v_0$$





$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_X & 0 & u_0 \\ 0 & f_Y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_L/Z_L \\ Y_L/Z_L \\ 1 \end{bmatrix}$$

$$u = {f_X X_L}/{Z_L} + u_0$$

$$v = {f_Y Y_L}/{Z_L} + v_0$$

From the second equation we can find Z_L as all of the rest parameters are known:

$$Z_L = \frac{f_Y Y_L}{v - v_0}$$

Perspective Image

Conventional sensors such as normal cameras have relatively modest FOVs which complicates the reconstruction of the whole surroundings

Fisheye Image

More recent research direction looks to improve the situation by extending the FOV by deploying omnidirectional cameras

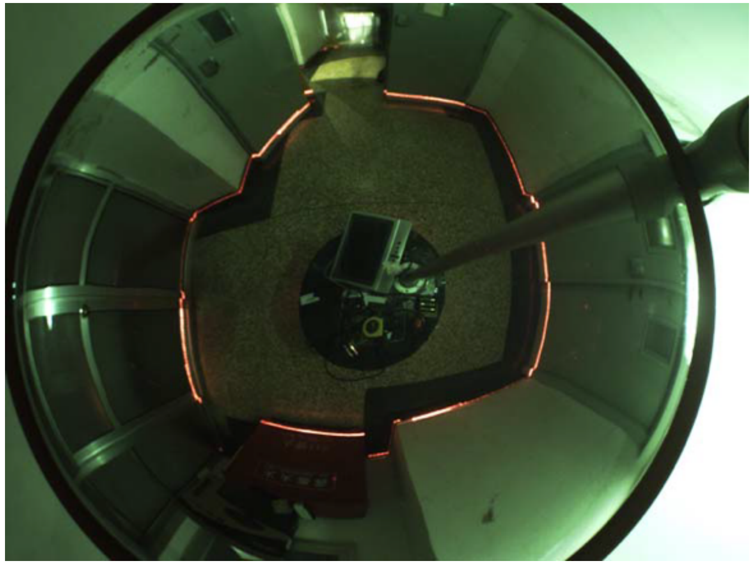
In order to carried out 2D mapping or 3D reconstruction of the indoor environment, a vision system must be calibrated.

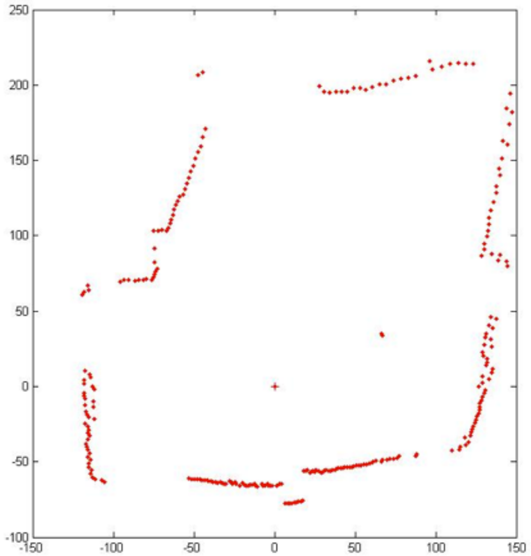
Uncalibrated case

$$\begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \times \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \boldsymbol{t_Z} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

$$f(\rho) = a_0 + a_2\rho^2 + \cdots + a_N\rho^N$$

$$\rho = \sqrt{(u - u_c)^2 + (v - v_c)^2}$$





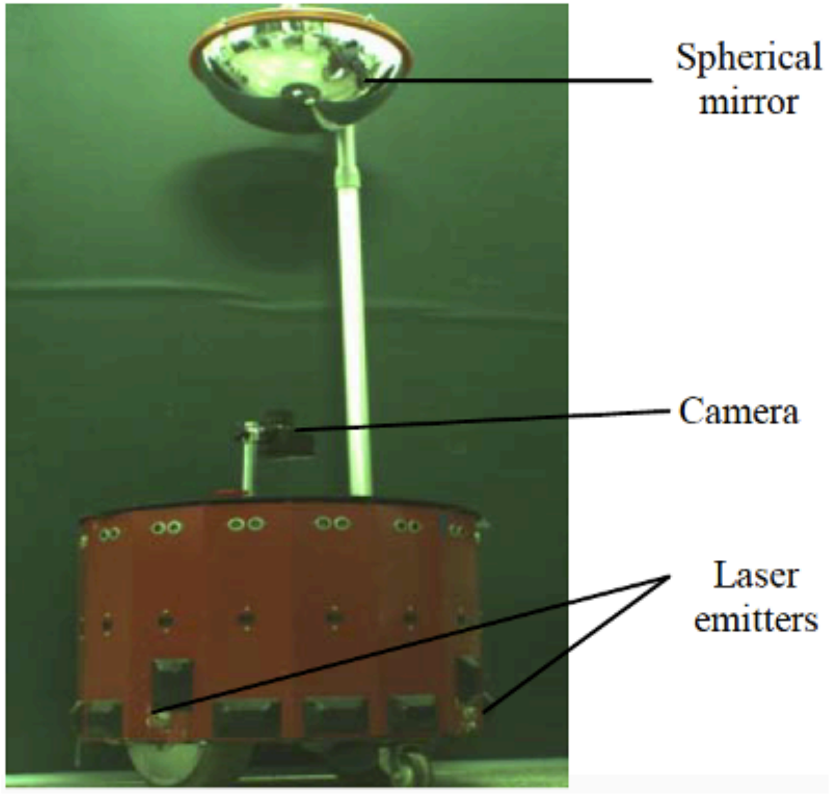
Calibrated case

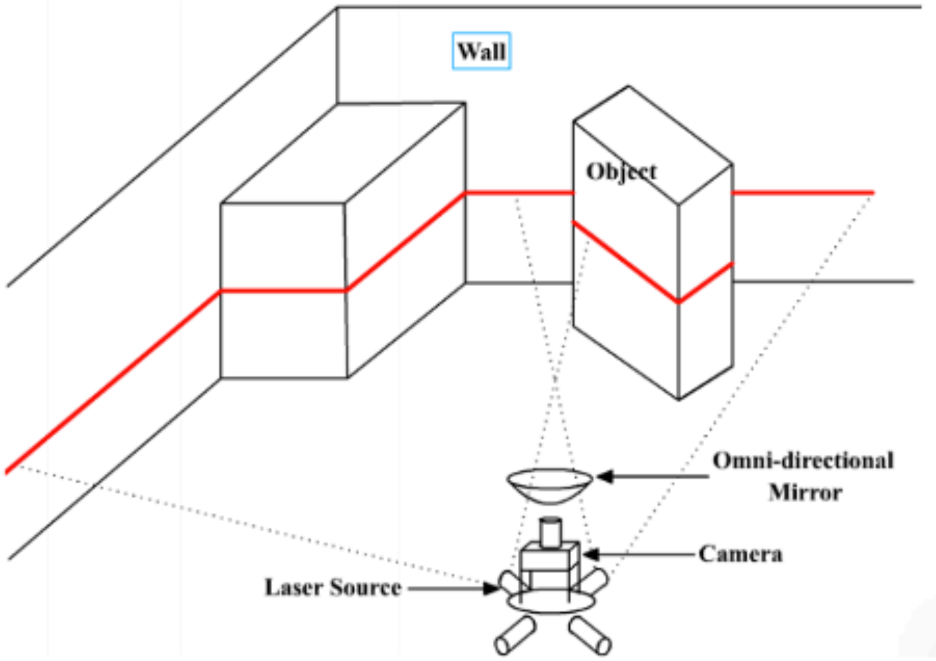
$$\begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \times [\boldsymbol{r}_1^c \quad \boldsymbol{r}_2^c \quad \boldsymbol{r}_3^c][\boldsymbol{r}_1^l \quad \boldsymbol{r}_2^l \quad \boldsymbol{r}_3^l \quad \boldsymbol{t}^l] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

$$f(\rho) = a_0 + a_2\rho^2 + \cdots + a_N\rho^N$$

$$\rho = \sqrt{(u - u_c)^2 + (v - v_c)^2}$$

In order to calibrate the Vision System, it also must not include several Laser Planes with the same emitting color (e.g. Red Color). We can only calibrate one Laser Plane emitted by one Laser Emitter.





In general form the equation of the laser plane projection can be written in the following way:

$$\begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \times [R_C] \begin{bmatrix} R_L & | & T_L \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \times [r_1^c \quad r_2^c \quad r_3^c][r_1^l \quad r_2^l \quad r_3^l \quad t^l] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

where u, v are the pixel coordinates of the image point; r_1^c,r_2^c,r_3^c are the column vectors of the camera rotation matrix; parameters r_1^l ,r_2^l ,r_3^l, t^l represent column vectors of the laser plane transformation matrix; *X*, *Y*, *Z* – world coordinates of the laser projection. The polynomial $f(\rho)$ can be overwritten as:

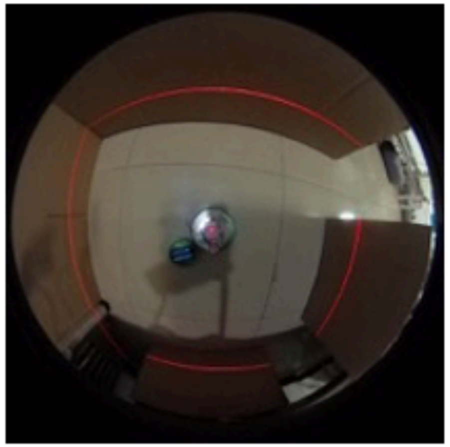
$$f(\rho) = a_0 + a_2\rho^2 + \cdots + a_N\rho^N$$

$$\rho = \sqrt{(u - u_c)^2 + (v - v_c)^2}$$

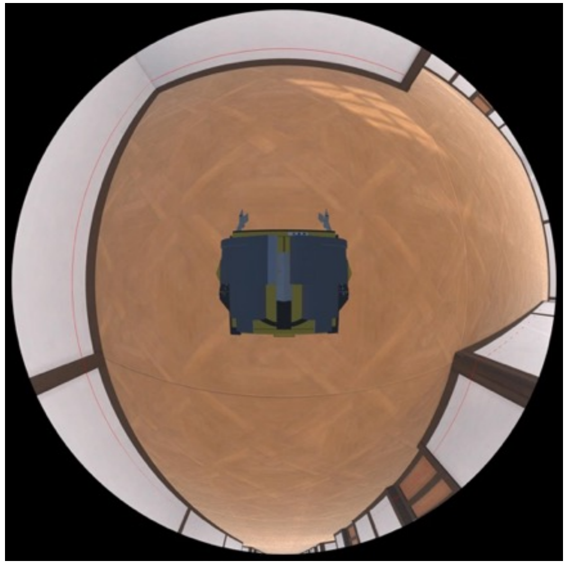
where a_i – coefficients; *N* – degree of the polynomial; u_c and v_c represent the coordinates center of an omnidirectional image.

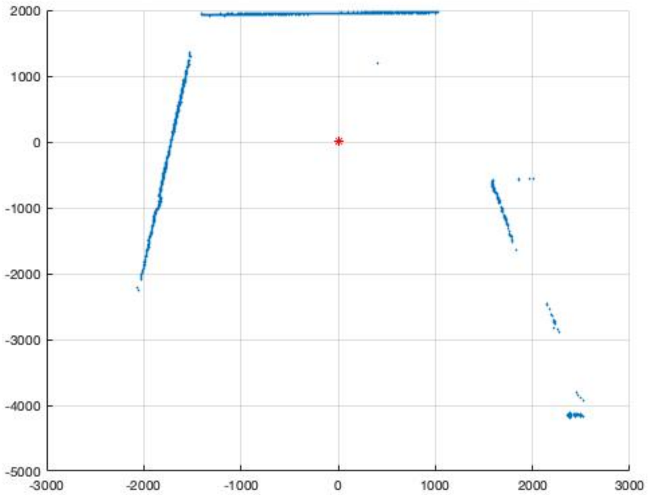
Fisheye camera

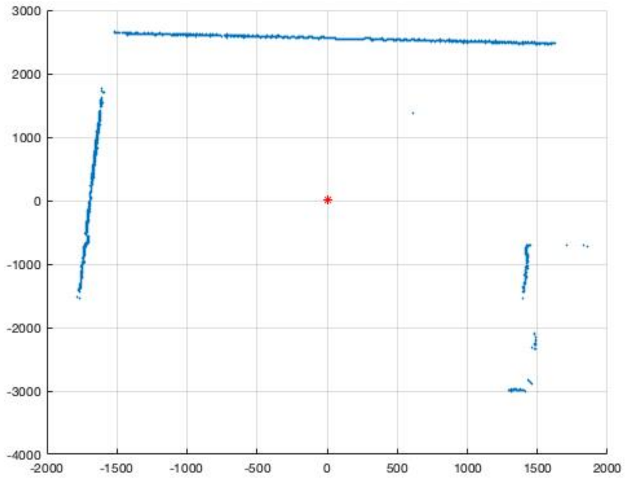
Omnidirectional laser emitter

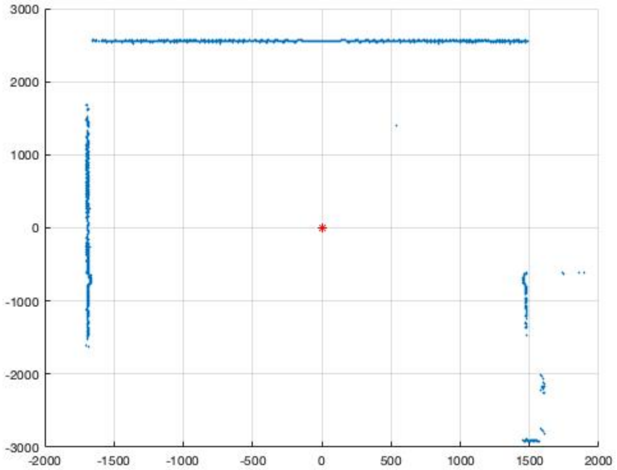


Snapshot captured by fisheye camera




2000
1000
0
-1000
-2000
-3000
-4000
-5000
-3000
-2000
-1000
0
1000
2000
3000


3000
2000
1000
0
-1000
-2000
-3000
-4000
-2000
-1500
-1000
-500
0
500
1000
1500
2000


3000
2000
1000
0
-1000
-2000
-3000
-2000
-1500
-1000
-500
0
500
1000
1500
2000

$$\begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \times [R_C] \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

The laser plane is located on the constant from the camera optical center distance – distance t_3. Therefore, in world coordinates Z is equal to zero as only X and Y are changeable.

$$\begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \times [R_C] \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = 0$$

Consequently, after multiplying transformation matrix of the laser plane by world coordinates, namely parameters of the 3rd column after multiplication will be equal to zeros:

$$r_{13} \cdot 0 = 0;\ r_{23} \cdot 0 = 0; r_{33} \cdot 0 = 0;$$

Moreover, in laser translation parameters we are only interested in the distance between the camera and laser plane. Therefore, t_1 and t_2 are equal to zeros as this offset makes no sense in our case. The value of this offset is significant for example for reconstruction tasks, where laser plane should be rotated per each scanning frame.

In contrast, for 2D mapping, laser plane has a fixed configuration and only its orientation and distance to the camera origin along the Z-axis are needed to be known. So, we significantly simplified our equation:

$$\begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \times [R_C] \begin{bmatrix} r_{11} & r_{12} & 0 & 0 \\ r_{21} & r_{22} & 0 & 0 \\ r_{31} & r_{32} & 0 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \times [R_C] \begin{bmatrix} r_{11} & r_{12} & 0 & 0 \\ r_{21} & r_{22} & 0 & 0 \\ r_{31} & r_{32} & 0 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = 0 \rightarrow \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \times \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = 0 \rightarrow \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \times \begin{bmatrix} h_{11} \cdot X + h_{12} \cdot Y + h_{13} \\ h_{21} \cdot X + h_{22} \cdot Y + h_{23} \\ h_{31} \cdot X + h_{32} \cdot Y + h_{33} \end{bmatrix} = 0$$

Cross product can be found as follows:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

So, let’s do the same with our equation:

$$\begin{cases} v(h_{31}X + h_{32}Y + h_{33}) - f(\rho)(h_{21}X + h_{22}Y + h_{23}) = 0 \\ f(\rho)(h_{11}X + h_{12}Y + h_{13}) - u(h_{31}X + h_{32}Y + h_{33}) = 0 \\ u(h_{21}X + h_{22}Y + h_{23}) - v(h_{11}X + h_{12}Y + h_{13}) = 0 \end{cases}$$

Now we consider the last two equations and open brackets:

$$\begin{cases} f(\rho)h_{11}X + f(\rho)h_{12}Y + f(\rho)h_{13} - uh_{31}X - uh_{32}Y - uh_{33} = 0 \\ uh_{21}X + uh_{22}Y + uh_{23} - vh_{11}X - vh_{12}Y - vh_{13} = 0 \end{cases}$$

And let’s separate X and Y

$$\begin{cases} X(f(\rho)h_{11} - uh_{31}) + Y(f(\rho)h_{12} - uh_{32}) + f(\rho)h_{13} - uh_{33} = 0 \\ X(uh_{21} - vh_{11}) + Y(uh_{22} - vh_{12}) + uh_{23} - vh_{13} = 0 \end{cases}$$

In order to do not deal with huge equations let’s define new variables:

$a_1 = f(\rho)h_{11} - uh_{31}$; $b_1 = f(\rho)h_{12} - uh_{32}$; $c_1 = f(\rho)h_{13} - uh_{33}$; $a_2 = uh_{21} - vh_{11}$; $b_2 = uh_{22} - vh_{12}$; $c_2 = uh_{23} - vh_{13}$.

$$\begin{cases} a_1X + b_1Y + c_1 = 0 \\ a_2X + b_2Y + \ c_2 = 0 \end{cases}$$

From the first equation we have:

$$\boldsymbol{X = (-c_1 - b_1Y)/\ a_1}$$

Congratulations! We've just found the world coordinate X! By replacing X in the 2nd equation we have and multiplying everything by a_1 we have:

$$-a_2c_1 - a_2b_1Y + a_1b_2Y + a_1c_2 = 0$$

Separate *Y*:

$$Y(-a_2b_1 + a_1b_2) - a_2c_1 + a_1c_2 = 0$$

Finally, we have:

$$\boldsymbol{Y = (a_2c_1 - a_1c_2)/(\ a_1b_2 - a_2b_1)}$$

And congratulations! Now the coordinate Y is also known to us:

Thus, each world coordinate of the laser's projection is represented by the distance from the camera to the laser plane (Z-coordinate) and coordinates X, Y what we've just found.