Robot Vision

jan.lemeire

November 2021

1 The perspective projection

Depending on the position and orientation of the robot, the 4 corners of the beacon will be projected on different locations of the camera image. The transformation of a point into a pixel in an image happens through the following chain of frames:

$$global\ frame \leftrightarrow camera\ frame \leftrightarrow projection\ frame \leftrightarrow pixel\ frame \leftrightarrow image\ frame \tag{1}$$

A point in space is first expressed in the frame attached to the camera. It is then projected on the camera's image plane. Then the projection is discretized into pixels and forming an image.

Fig. 1 shows how a point P in space is projected on the projection plane (point p, called image plane in the figure). The figure is taken from Corke2011, 'Robotics, Vision and Control', chapter 11. A coordinate system (x_C, y_C, z_C), which we call the camera frame, is defined in the camera's origin and has a distance f with the projection frame, called the *focal distance*.

From Fig. 1 we can deduce the basic equations expressed in the camera's frame:

$$\frac{Y}{Z} = \frac{y}{z} = \tan \alpha_y \tag{2}$$

$$y = y_{pix}.\rho_h \tag{3}$$

$$\Rightarrow y_{pix} = \tan \alpha_y . \frac{f}{\rho_h} \tag{4}$$

with ρ_h the height (size) of a pixel. The same equations apply for the x-coordinates. We assume that a pixel has the same width as height, thus $\rho_h = \rho_w = \rho$.

Each pixel is created by a point in space that lies on the ray that starts in the camera origin and goes through the pixel (red line in Fig. 1). A ray is defined by the angles (α_x, α_y) or by what we call the $ray\ ratios$ $(\frac{x}{z}, \frac{y}{z})$.

It follows that we only need one camera parameter: $frho\frac{f}{\rho}$. This is the *intrinsic parameter* of the camera. The meaning of $frho$: the number of pixels in view between 0 and 45 degrees (a ray at 45 degrees has a ray ratio of 1).

Finally, in image processing, a pixel is most often represented in the $imageframe$ with the origin at the top left (Fig. 1). The first index of a pixel refers to the row and the second its column.

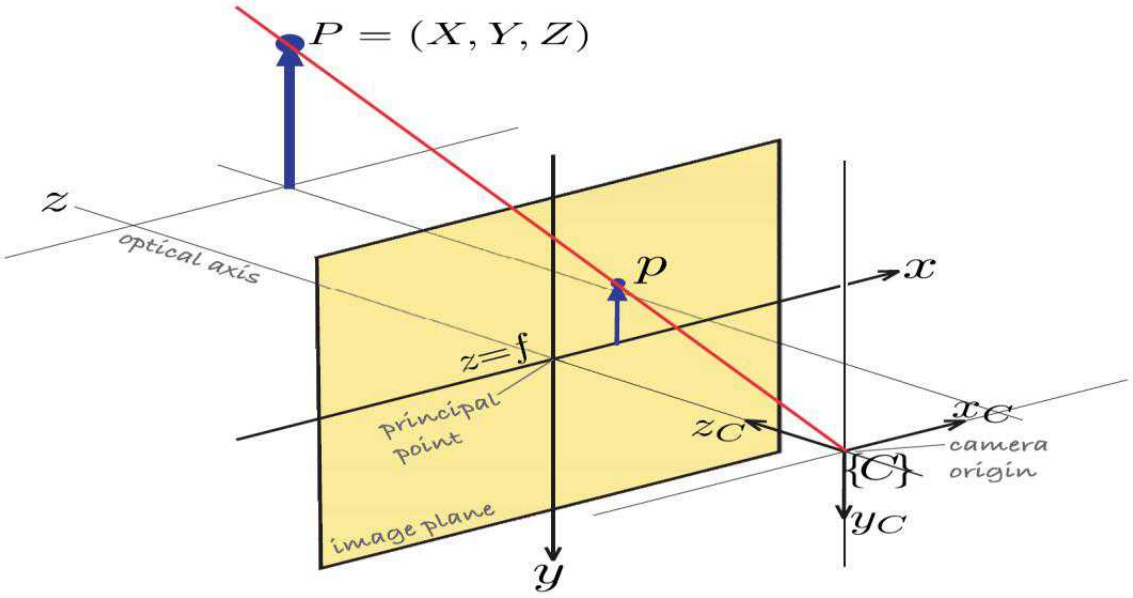

Figure 1: The perspective projection, taken from Corke2011. The image plane contains our projection frame

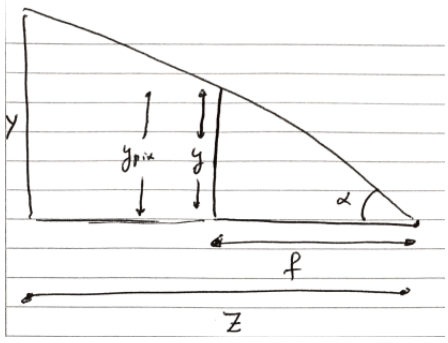

Figure 2: The perspective projection in the YZ-plane.

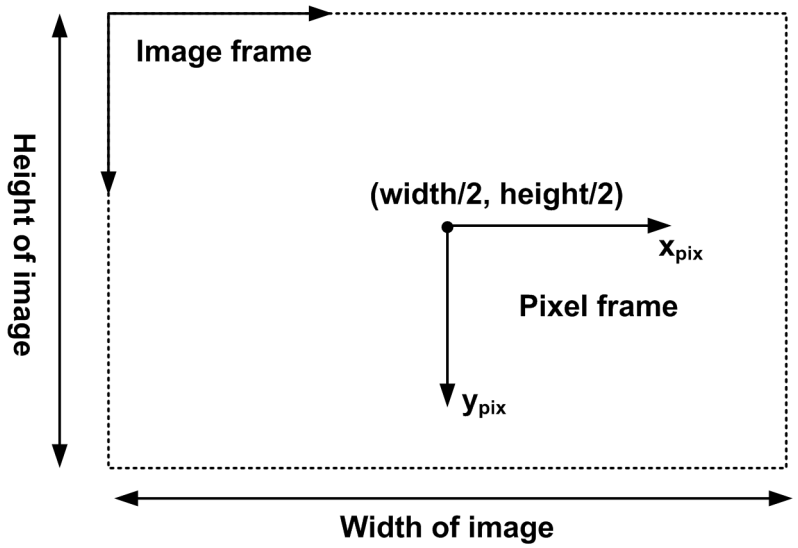

Figure 3: Pixel and image frame.

The perspective transformation is defined in the frame attached to the camera. In a global frame, a transformation is necessary based on the *pose* of the camera which consists of its location and the orientation. The first results in a translation, the second a set of rotations, most often expressed by the roll, pitch and yaw (Fig. 1). We call this the $global2camera$ transformation function. The pose parameters are the *extrinsic* parameters of the camera.

Conclusion, we switch between the following frames:

$$global\ frame \leftrightarrow camera\ frame \leftrightarrow projection\ frame \leftrightarrow pixel\ frame \leftrightarrow image\ frame \tag{5}$$

2 Robot tracking

To know the true position of the robot, we will track him with a camera. For this we put a clear figure, which we call the beacon, on his back: Fig. 2.

The position of the robot is estimated by identifying the beacon in the image (it has a distinct color) and solving the inverse problem: Fig. 2. Wikipedia: 'Inverse problems are some of the most important mathematical problems in science and mathematics because they tell us about parameters that we cannot directly observe.' Here: from a set of observations (the corners of the beacon) determine the camera's parameters and the position of the robot which caused the image.

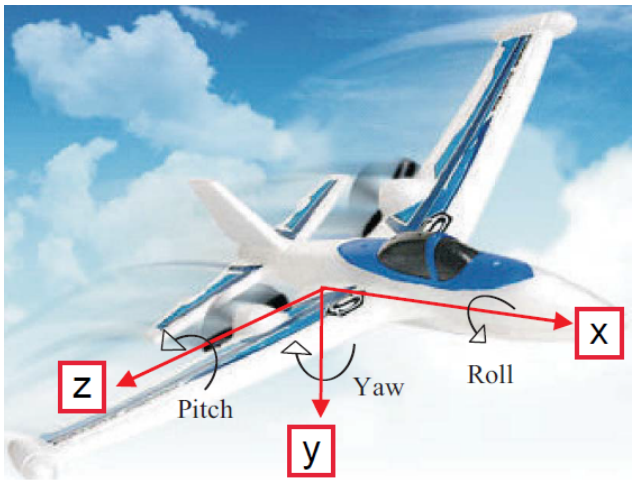

Figure 4: The rotations of a frame attached to a moving object.

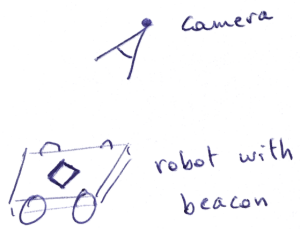

Figure 5: Tracking a moving robot with a camera by using a beacon.

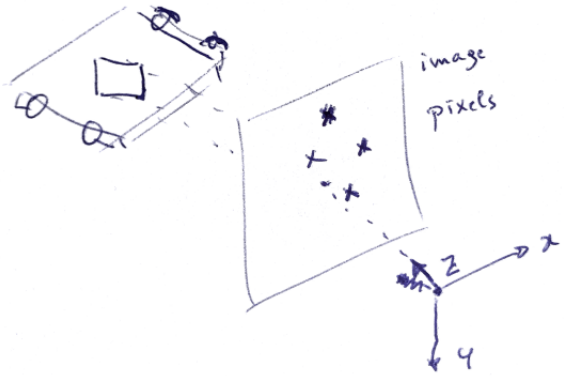

Figure 6: The beacon projected on the image.

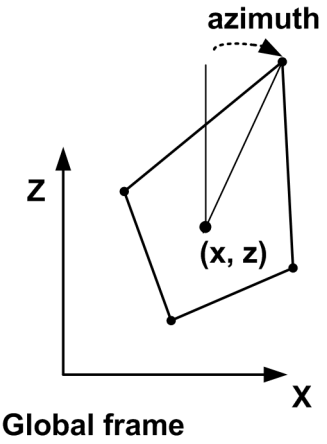

Figure 7: The beacon on the robot and its parameters.

2.1 Simultaneous calibration and estimation

Besides estimation of the position of the robot, the camera should also be *calibrated*: determine the camera's intrinsic and extrinsic parameters. The intrinsic parameters are its focal distance f and pixel size ρ. These can be combined into parameter which we call $frho$. If we consider the coordinate system at the center of the camera (the camera frame), the perspective projection (from camera to pixel frame) is given by the following equations:

$$x_{pix}, y_{pix} = persp(x, y, z, frho). \tag{6}$$

The extrinsic parameters of a camera are its pose with respect to the global axis system: position in the global axis system (3 parameters: x, y and z) and orientation (3 parameters: roll, pitch, yaw). We will assume the roll to be zero.

The robot has 3 free parameters: position on the ground (x and z coordinates of the center of the beacon, we assume an horizontal ground: a constant y) and its orientation (yaw, which we will call the $azimuth$). If we fix the global axis system at the position of the camera and in the direction of the camera (yaw=0), we end up with 6 free parameters : $frho$, y (height of the camera), $pitch$ (of the camera), x, z and $azimuth$ (robot). However, the camera height and $frho$ are related, they both determine the scale. Since the camera height can easily be measured, we will consider it to be known.

The beacon has a known size and position on the robot. The global coordinates of the beacon's 4 corners are functions of the 3 free robot parameters: $p^i = f(x, y, azimuth)$ (Fig. 2.1). The size of the beacon is known. Note that a non-zero azimuth results in a non-linear transformation.

The Inverse Problem

Each of the 4 corners of the beacon gives rise to 2 pixel coordinates based on the perspective projection. Now, the problem is to estimate the unknown calibration and state parameters with the knowledge of the corner pixels. It's an inverse problem since

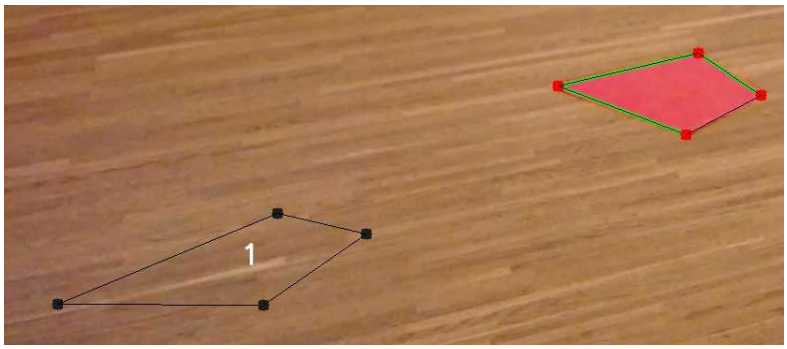

Figure 8: The rotations of a frame attached to a moving object.

we want to know the state which generated the observed pixels.

First the corner should be expressed in the camera axis system (using the camera's $pitch$ and y) by the $global2camera$ transformation function. Next, the perspective transformation is applied which results in:

$$p^i_{pix} = persp(global2camera(p^i,),) \tag{7}$$
$$= persp(global2camera(f(x, y, azimuth),),) \tag{8}$$
$$= h() \tag{9}$$

with h the observation function. We end up with 5 unknowns and 8 equations. I think that 2 equations are redundant (the fourth point is determined by the 3 others), but it is not a problem to have redundant information. The error equations are thus of the following form: $e_i = h() - p^i_{pix}$.

The calibration and state parameters are then established by minimizing the sum of squared errors of the error equations. We iteratively apply Newton on an initial guess. The meaning of guess $_0$ can be expressed as follows 'would the camera and robot state be $_0$, then the beacon in the image would be $h(_0)$'. This can be seen in Fig. 2.1. The iterative approximation is shown in Fig. 2.1.

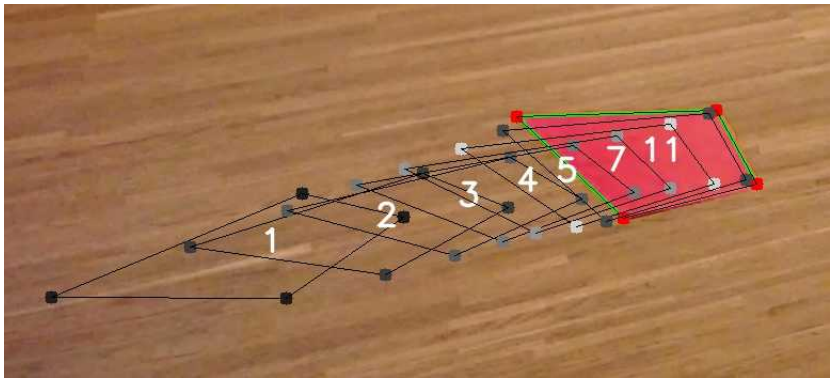

Figure 9: The rotations of a frame attached to a moving object.