CSE-483: Mobile Robotics **Monsoon 2019**

Camera Modelling

Prepared by: Projit (20161014) and Abhinav (20171059)

1 Introduction

Our primary objective is to map a point from the 3D world onto the image. Why? Because we need to understand the geometry of the 3D world, and hence, we need to know the mapping from the 3D world to the image plane.

Images are formed on the whole by taking a 3D world and projecting it onto a 2D plane. While cameras seem to be complicated devices(and they truly are), almost all of their functionality can be described through geometry and projections. In fact, this 3D to 2D image formation process can be described through composing a few matrices(in the simplest of cases).

2 Pinhole Camera

One of the simplest ways to see how a camera works is through a pinhole camera.

Some terminology:

1. Aperture: size of the opening through which the light rays enter the camera.
2. Focus: Specific distance from the lens at which images appear "in focus". Is a property of the lens

The way a pinhole camera works can be understood as an approximation of the thin lens equation. We know in thin lens

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$$

Pinhole Approximation: The assumption is that hole(aperture) in the pinhole camera can be approximated to be a thin lens. Additionally, the focal length of the lens is less than the objects being imaged.

Substituting the very large distance, we get the image forms at the focal length f, which is determined by the aperture. In practical scenarios, one must move the image plane around in order to bring the image into focus, as determining the focal length of the aperture is very difficult.

3 Reference Frames

To be able to track objects between the world and the image that we take, we must come up with some way to relate the positions of things.

1. World Coordinates: Assume some arbitrary (but well defined) coordinate system in the world. There would be 3 axes to specify the position of any point in it uniquely.

$$P_w = (X_w, Y_w, Z_w)$$

2. Camera Coordinates: When the camera is placed in the world, it has its own set of axes with which it references all of the objects in the world.

$$P_c = (X_c, Y_c, Z_c)$$

   An arbitrary rotation + translation would need to be applied to move the World Coordinate frame to the Camera Coordinate frame.

3. Image Coordinates: Objects can be identified on the 2D image plane using pixel coordinates as well. This is done by applying the projection onto the Camera coordinates.

$$P = (u, v)$$

4 Projections

So when we have map a point from the 3D world to the image plane, we have to move around the reference frames in the following manner:

1. We start at in the World/Object coordinate system
2. We move from the world frame to the Camera coordinate system. Here, everything is relative to the camera and not the origin of the world. It's how the camera sees the world.
3. We then move to from the camera's coordinate system to the Image plane coordinate system. This is where the image is actually formed!

$$\begin{pmatrix} \mathrm{x}_c \\ \mathrm{y}_c \\ 1 \end{pmatrix} = H_k^c * H_o^k * \begin{pmatrix} \mathrm{X}_o \\ \mathrm{Y}_o \\ \mathrm{Z}_o \\ 1 \end{pmatrix}$$

Looking at the above equation from right to left, we start with the world coordinates of the object in 3D, denoted by 'o'. We multiply it by a matrix that maps it from object coordinates 'o' to camera coordinates 'k'. This is then mapped again from 'k' to the image plane 'c' to get the 2D point coordinates.

the transformation from the world to the camera's coordinate frame does not result in any loss of information. However, when we move further to the image plane frame of reference, we essentially move from a 3d world to a 2D world, which inevitably leads to a loss if information. When moving from 3D to 2D, what is lost?

1. lengths: Phenomena like depth.
2. angles: Phenomena like parallel lines seem to intersect

Preserved:

1. Straight Lines (Projective Geometry won't cause them to curve)

Some Terminology:

1. Vanishing Point: Point at which parallel lines intersect
2. Vanishing Line: Line at which parallel planes intersect

These are both perspective projection phenomena. Cameras perform a Perspective Projection on the world, thus the images they form are affected by these.

5 Homogeneous Coordinates + Motivation

A rotation matrix about any axis is a 3x3 matrix. For example, a rotation about the Z axis could be represented by:

$$A = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

While rotation matrices can be multiplied in chains, translation requires addition of two matrices. However, translation can also be represented by a product, with the nifty trick of introducing an additional dimension, which we shall use as a scaling factor.

In 2D, the homogeneous coordinate (x, y, s) represents $(x/s, y/s)$ Similarly for 3D, the homogeneous coordinate (x, y, z, s) represents $(x/s, y/s, z/s)$

Example, translate (x, y) by (a, b)

$$\begin{pmatrix} \text{x+a} \\ \text{y+a} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \text{a} \\ 0 & 1 & \text{b} \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \text{x} \\ \text{y} \\ 1 \end{pmatrix}$$

(To make this work for 3D coordinates, only one more dimension needs to be added)

Now, given a 3D point we can now express translations and rotations in the form of matrix multiplications. So now we have a 3D point is represented as:

$$P_h = (X, Y, Z, S) => P(X/S, Y/S, Z/S)$$

and a 2D point as:

$$P_h = (X, Y, S) => P(X/S, Y/S)$$

6 Camera Intrinsics and Extrinsics

To map a point from the 3D world onto the image, it is essential to know where the camera and the object are exactly located. Camera Extrinsics are a set of camera parameters which describe where the camera is located in the 3D world i.e. it is the configuration of the camera with respect to the 3D world coordinate system.

The location of the object is known to us, in the world coordinate system. If we also know the location of the camera in the world, we can easily map it to the camera coordinates as we have already discussed above. But will transforming from the world to the camera coordinate system suffice? No, as evident from what we saw in section 3. We also have to move from the camera's frame to the image plane coordinate system. And this is exactly what camera intrinsics are - parameters that are internal or special to the camera like the camera constant. The intrinsic parameters have an influence on how the 3D world point ends up on the image.

So in a nutshell, extrinsics describe the pose of the camera in the world. Intrinsics describe the mapping of the scene in front of the camera to the pixels in the final image.

7 Camera Parameters

We now claim to start with some world coordinate P_w. Let us say we can, through arbitrary rotation and translation, bring it into the camera frame. We then have P_c Now we assume that the camera coordinate

frame was constructed such that the image plane passes through the origin as so ¡Insert stuff from page 47-48¿

Taking advantage of the homogenous coordinate system, we can express these transformations on P_c by:

$$\begin{pmatrix} \lambda u \\ \lambda v \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathrm{k}_u f & 0 & \mathrm{u}_0 \\ 0 & \mathrm{k}_v f & \mathrm{v}_0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \mathrm{X}_c \\ \mathrm{Y}_c \\ \mathrm{Z}_c \end{pmatrix}$$

On the LHS, (u, v) represents the pixel coordinates The 3x3 matrix is usually represented by K It is the Calibration Matrix or the Matrix of Intrinsic Parameters. *Note* it is not necessary that f along the x and y be equal as is assumed here.

Coming back to the earlier issue, we hand-waved getting from World Coordinates to the Camera Coordinates. Right now we have:

$$\begin{pmatrix} \lambda u \\ \lambda v \\ \lambda \end{pmatrix} = K * \begin{pmatrix} \mathrm{X}_c \\ \mathrm{Y}_c \\ \mathrm{Z}_c \end{pmatrix}$$

As we have seen, through homogenous coordinates, translation can be expressed as a product. A rotation + translation can be expressed conveniently in a single 3x4 matrix. Multiply it out for yourself to see how it does the rotation followed by translation on each coordinate. So a generic matrix of this form captures the change in reference from World Coordinate(in homogeneous form) to Camera Coordinates

$$\begin{pmatrix} \mathrm{X}_c \\ \mathrm{Y}_c \\ \mathrm{Z}_c \end{pmatrix} = \begin{pmatrix} \mathrm{r}_{11} & \mathrm{r}_{12} & \mathrm{r}_{13} & \mathrm{t}_1 \\ \mathrm{r}_{21} & \mathrm{r}_{22} & \mathrm{r}_{23} & \mathrm{t}_2 \\ \mathrm{r}_{31} & \mathrm{r}_{32} & \mathrm{r}_{33} & \mathrm{t}_3 \end{pmatrix} * \begin{pmatrix} \mathrm{X}_W \\ \mathrm{Y}_W \\ \mathrm{Z}_W \\ 1 \end{pmatrix}$$

This can be concisely represented as:

$$\lambda \begin{pmatrix} \mathrm{u} \\ \mathrm{v} \\ 1 \end{pmatrix} = K * [R|T] * \begin{pmatrix} \mathrm{X}_W \\ \mathrm{Y}_W \\ \mathrm{Z}_W \\ 1 \end{pmatrix}$$

References

[1] Prof. Cyrill Stachniss's lectures on Photogrammetry-I, University of Bonn

[2] Prof. Davide Scaramuzza's lectures on Image Formation, University of Zurich