

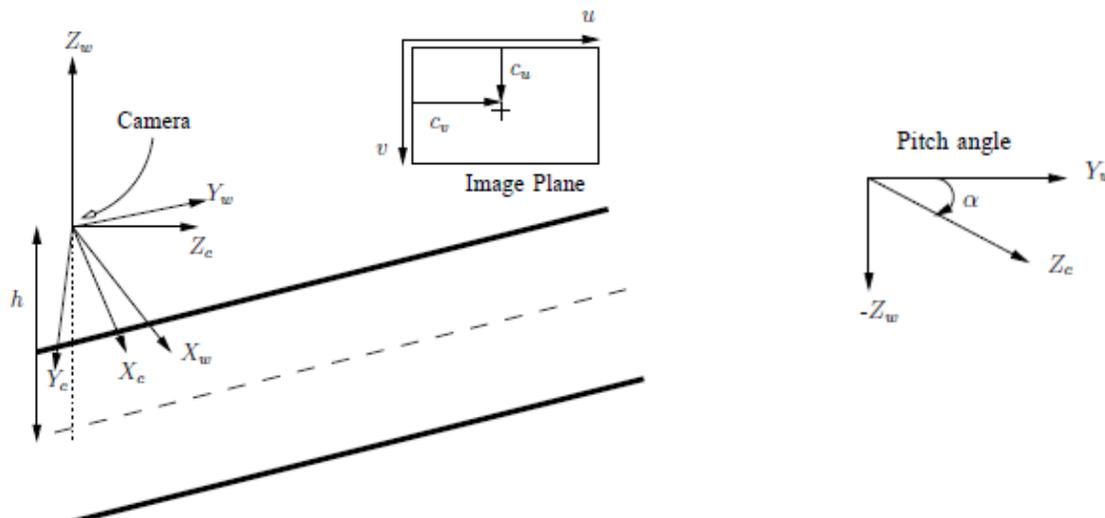
Homework #1: Inverse Perspective Mapping

In this homework we will implement a set of functions to transform an image into its bird's eye view, known as an Inverse Perspective Mapping. This is the same image projected on the ground plane, where the camera is at height h above the ground.



Fig. 3. IPM sample. Left: input image with region of interest in red. Right: the IPM view.

Part 1: Transformations



We will assume the camera frame is centered at the world frame so there is no translation. We will only have one rotation angle, α , which is called the pitch angle, and rotates the Z -axis of the camera downwards. The camera frame and world frame are as defined in the figure, where the origin is at a



height h above the ground, and so any point on the ground has a z coordinate that equals $-h$.

The camera intrinsic parameters are the focal lengths f_x , f_y , c_x , c_y such that to transform a point on the image plane from the camera frame to the image frame (pixels) the transformation is:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix}$$

1. Write the 4x4 transformation matrix that transforms points from the image frame (pixels)

$${}^i p = [u \ v \ 1 \ 1]^T \text{ to the ground plane } {}^w p = [x_g \ y_g \ -h \ 1]^T .$$

1.1. What is the transformation for a pixel in the image frame ${}^i p = [u \ v \ 1 \ 1]^T$ to the same point on the image plane in the camera frame ${}^c p = [x_c \ y_c \ 1 \ 1]^T$? (Hint: look at the transformation above).

1.2. What is the transformation for a point in the camera frame on the image plane ${}^c p = [x_c \ y_c \ 1 \ 1]^T$ to the same point in the world frame ${}^w p = [x_w \ y_w \ z_w \ 1]^T$? (Hint: rotation by the pitch angle around what axis?)

1.3. What is the projection for a point on the image plane (the plane at a distance of 1 from the origin) ${}^w p = [x_w \ y_w \ z_w \ 1]^T$ to a point on the ground plane ${}^w p = [x_g \ y_g \ -h \ 1]^T$? Write down the transformation matrix. (Hint: similar to perspective projection).

1.4 Combine these three transformations to find the 4x4 homogeneous transformation matrix.

2. Write the inverse of the above 4x4 transformation matrix i.e. the matrix that transforms points on the ground plane in world coordinates ${}^w p = [x_g \ y_g \ -h \ 1]^T$ to points on the image plane in the image frame (pixels) ${}^i p = [u \ v \ 1 \ 1]^T$.

2.1. What is the transformation that takes a point on the ground plane ${}^w p = [x_g \ y_g \ -h \ 1]^T$ to a point on the image plane ${}^w p = [x_w \ y_w \ z_w \ 1]^T$ i.e. the intersection of the image plane with the ray from the origin passing through the point on the ground plane. (Hint: The image plane is at a distance of 1 from the origin. What is the normal to the plane? What's the equation of the plane? What's the parametric equation of the ray from the origin through the point on the ground plane? What's the intersection of the ray and the plane?)

2.2. What is the transformation that takes points on the image plane from the world frame ${}^w p = [x_w \ y_w \ z_w \ 1]^T$ to the camera frame ${}^c p = [x_c \ y_c \ 1 \ 1]^T$? (Hint: inverse of 1.2).

2.3. What is the transformation that takes points on the image plane from the camera frame ${}^c p = [x_c \ y_c \ 1 \ 1]^T$ to the image frame ${}^i p = [u \ v \ 1 \ 1]^T$?



2.4 Combine the three transformations above to find the inverse 4x4 transformation matrix.

Part 2: Rendering the image

Write a program (C++ with OpenCV, Matlab, Octave, ... etc.) that takes in an image and the camera intrinsic parameters, and produces the top view of that image. You will need to implement the above transformations to transform points from the image frame to the ground plane and vice versa.

The general steps will be something like this:

1. Define a region in the input image that will be transformed. Typically this should be below the vanishing line in the image (the horizon) which will intersect the ground plane at infinity.
2. Find the world coordinates on the ground plane of the corners of this region.
3. Sample points equally in that region on the ground plane, according to the required resolution of the output image (i.e. number of pixels in width and height). Every such point will get a color from the image.
4. Transform these points on the ground plane to points in the image plane. To find the color, just round the pixel coordinates to the nearest pixel in the image. Points that lie outside the image get black color.

Assume the following for the camera parameters when working with the sample images provided:

$$\begin{aligned}f_x &= 309.43 \\f_y &= 344.21 \\c_x &= 317.9 \\c_y &= 256.5 \\h &= 2179.8 \\pitch &= 14 \text{ degrees}\end{aligned}$$

Look at this paper for the transformation matrices, though you still need to derive them :)

<https://www.mohamedaly.info/files/aly08realtime.pdf?attredirects=0>

Note: There is a mistake in the first transformation matrix in the paper s_iT in the element on the second row and second column. It is written as $s_1 c_1 / f_v$ and should be $s_1 c_2 / f_v$.