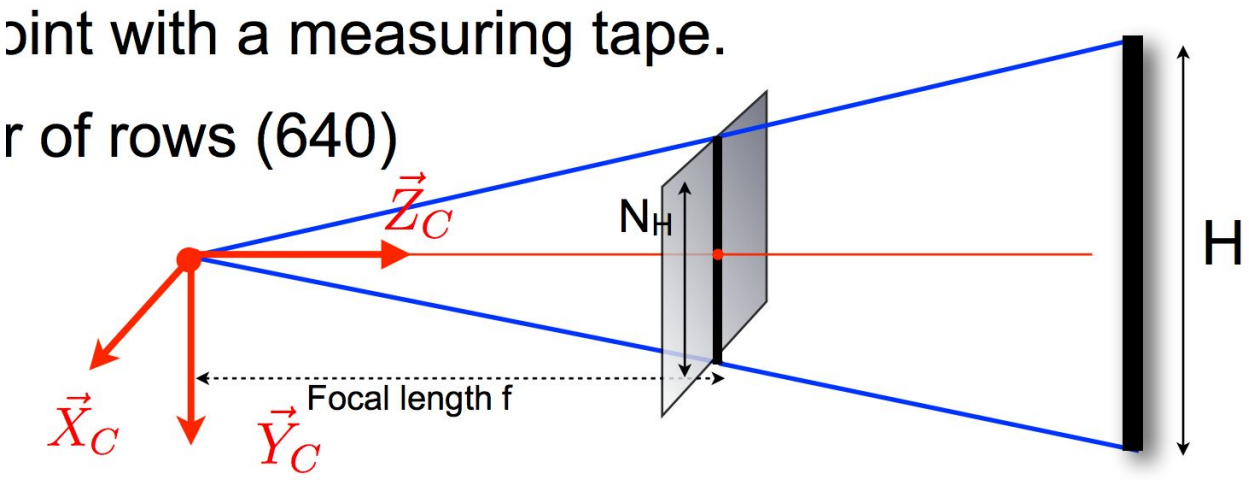
CMPE 161 Assignment 3 Report
Nirav Agrawal nkagrawa
Jerry Porter japorter

1) We computed the focal length by taking a photo of a notebook on the wall making the notebook the entire vertical length of the screen and measuring the distance from the phone and the notebook on the wall and the vertical pixel length of the phone's screen.



D = 10.5 inches (distance from the wall)
H = 11 inches (height of object against wall)
Nh = 640 pixels (vertical pixel length of the screen)

F = (D/H) * Nh
F = 611 Pixels

2) Take the equation of a vanishing line:

$$N_x^C(p_x^S - c_x) + N_y^C(p_y^S - c_y) + N_z^C f = 0$$

We calculate N^c by aligning the axis and finding the gravity vector. Px^s and Py^s are our two unknowns so we set one to a value and solve for the other since we know the size of the screen.

$P_x^s = 0 \quad (0, P_y^s)$

$N_x^c P_x^s + N_y^c P_y^s - N_x^c c_x - N_y^c c_y + N_z^c \cdot F = 0$

$$\frac{-N_x^c c_x - N_y^c c_y + N_z^c \cdot F}{-N_y^c} = P_y^s$$

$P_y^s = 0 \qquad (P_x^s, 0)$

$N_x^c P_x^s + N_y^c P_y^s - N_x^c c_x - N_y^c c_y + N_z^c \cdot F = 0$

$$N_y^c(0) - \frac{N_x^c c_x - N_y^c c_y + N_z^c \cdot F}{-N_x^c} = P_x^s$$

$c_x = 320$
$c_y = 240$
$F = 611$

but, if $P_x^s > 480$ or $P_y^s > 640$
then we set $P_x^s = 480$ & $P_y^s = 640$

$P_x^s = 480 \qquad (480, P_y^s)$

$$\frac{N_x^c(480) - N_x^c c_x - N_y^c c_y + N_z^c \cdot F}{-N_y^c} = P_y^s$$

$P_y^s = 640 \qquad (P_x^s, 640)$

$$\frac{N_y^c(640) - N_x^c c_x - N_y^c c_y + N_z^c \cdot F}{-N_x^c} = P_x^s$$

3) Our system works by taking in the accelerometer data and device motion data first. From there we align the values to our camera's axis and draw out two points according to the gravity vector. These two points can be anywhere on the axis because the next step is finding two more points on that line created by the first two points. These second set of points are set up to be within the screen on our phone so we can easily draw a line between the two points creating the horizon line we visually see.

Jerry did that majority of the algorithms we implemented, from re aligning the axes to calculating the second set of points for our horizon line and Nirav did the majority of implementing that logic into the program. We did switch off and help each other with both roles, but the majority of the programming was done by Nirav and the majority of the logic behind the programming was done by Jerry.