

Note for coordinate system conversion

Yuta Itoh

6th Nov, 2015

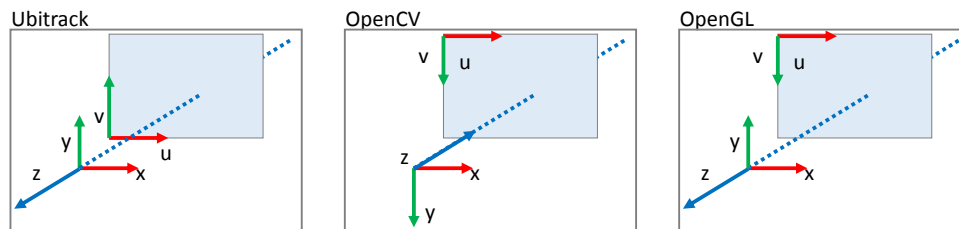


Figure 1: Coordinate systems of Ubitrack, OpenCV and OpenGL ¹.

1 Camera parameter conversions

Perspective projection with radial and tangential distortion²:

$$\begin{aligned}x' &= x/z \\x'' &= x'S + 2p_1x'y' + p_2(r^2 + 2x'^2) \\u &= f_x x'' + c_x\end{aligned}$$

$$\begin{aligned}y' &= y/z \\y'' &= y'S + p_1(r^2 + 2y'^2) + 2p_2x'y' \\v &= f_y y'' + c_y\end{aligned}$$

where $S = \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6}$. S is constant regardless of the signs of x and y .

¹Under a typical OpenGL programming convention before converting to Normalized Device Coordinates, which is *left-handed*

²http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

1.1 From OpenCV (image top-left) to Ubitrack

The v -axis of the image plane and y -axis and z -axis of the camera coordinates have to be flipped. Namely, $-y$ has to be projected to $h - 1 - v$, and x to u , this follows

$$\begin{cases} \hat{p}_1 &= p_1 \\ \hat{p}_2 &= -p_2 \\ \hat{f}_x &= -f_x \\ \hat{f}_y &= -f_y \\ \hat{c}_x &= c_x \\ \hat{c}_y &= h - 1 - c_y \end{cases} .$$

Note that f_y *should* stay the same. The same conversion applies from Ubitrack to OpenCV.

1.2 From OpenCV (image bottom-left) to Ubitrack

The y -axis and z -axis of the camera coordinates have to be flipped. Namely, $-y$ has to be projected to v , and x to u

$$\begin{aligned} u &= \hat{f}_x \left(x'S + 2\hat{p}_1 x'y' + \hat{p}_2 (r^2 + 2x'^2) \right) + \hat{c}_x \\ u &= \hat{f}_x \left(-x'S - 2\hat{p}_1 x'y' + \hat{p}_2 (r^2 + 2x'^2) \right) + \hat{c}_x \\ f_x (x'S + 2p_1 x'y' + p_2 (r^2 + 2)) + c_x &= -\hat{f}_x (x'S + 2\hat{p}_1 x'y' - \hat{p}_2 (r^2 + 2)) + \hat{c}_x \\ \\ v &= \hat{f}_y \left(y'S + \hat{p}_1 (r^2 + 2y'^2) + 2\hat{p}_2 x'y' \right) + \hat{c}_y \\ v &= \hat{f}_y (y'S + \hat{p}_1 (r^2 + 2y'^2) - 2\hat{p}_2 x'y') + \hat{c}_y \\ f_y (y'S + p_1 (r^2 + 2y'^2) + 2p_2 x'y') + c_y &= \hat{f}_y (y'S + \hat{p}_1 (r^2 + 2y'^2) - 2\hat{p}_2 x'y') + \hat{c}_y \\ f_y (y'S + p_1 (r^2 + 2y'^2) + 2p_2 x'y') + c_y &= \hat{f}_y (y'S + \hat{p}_1 (r^2 + 2y'^2) - 2\hat{p}_2 x'y') + \hat{c}_y \end{aligned}$$

, this follows

$$\begin{cases} \hat{p}_1 &= p_1 \\ \hat{p}_2 &= -p_2 \\ \hat{f}_x &= -f_x \\ \hat{f}_y &= f_y \\ \hat{c}_x &= c_x \\ \hat{c}_y &= c_y \end{cases} .$$

Note that f_y *should* stay the same. The same conversion applies from Ubitrack to OpenCV.

1.3 From Ubitrack to OpenGL

The v -axis of the image plane has to be flipped

$$\begin{aligned}h - 1 - v &= \hat{f}_y y'' + \hat{c}_y \\h - 1 - f_y y'' - c_y &= \hat{f}_y y'' + \hat{c}_y \\-f_y y'' + h - 1 - c_y &= \hat{f}_y y'' + \hat{c}_y.\end{aligned}$$

This follows

$$\begin{cases} \hat{f}_y &= -f_y \\ \hat{c}_y &= h - 1 - c_y \end{cases}.$$