The FOV camera model used by Calibu and Project Tango

Technical Report
January 2015
David Gossow, Google Inc.
Email: dgossow@google.com

Abstract

At the time of writing, Project Tango uses two camera models: a more traditional one with polynomial distortion which is widely used in the computer vision community, and a less common one we call the FOV model, used to describe cameras with fisheye optics. This document is meant to provide the essential information needed to interpret the FOV camera parameters used by Google's Project Tango and the Calibu calibration software. It is written for people already familiar with the concepts of camera models used in computer vision.

The FOV model

The FOV model is based on the camera model proposed in [1]. It uses a slightly different parametrization though, and the relation between the two models will be described in the second section.

The following describes the image formation process modeled by the FOV model, going from a point (X, Y, Z) in camera coordinates to a *pixel location* (x_i, y_i). The first step is a pinhole camera projection with focal length 1, leading to *normalized image coordinates* (x_u, y_u). Following that, we distort the projected point with a radial distortion function and then map to pixel coordinates (x_i, y_i).

Variable naming

X, Y, Z	Point in camera coordinates.
x_i, y_i	Distorted pixel coordinates.
x_d, y_d	Distorted normalized image coordinates.
x_u, y_u	Undistorted normalized image coordinates.
C_x, C_y	Center of projection and radial distortion.
f_x, f_y	Horizontal/vertical focal length.
r_u	Distance to center of projection in normalized image coordinates.
r_d	Distance to center of projection in distorted image coordinates.
w	Distortion parameter.

Equations

$x_u = \frac{X}{Z}$

$y_u = \frac{Y}{Z}$
$r_u = \sqrt{x_u{}^2 + y_u{}^2}$
$r_d = \frac{1}{w}\ atan(2 r_u tan(\frac{w}{2}))$
$x_d = x_u \frac{r_d}{r_u}$
$y_d = y_u \frac{r_d}{r_u}$
$x_i = f_x x_d + C_x$
$y_i = f_y y_d + C_y$

Putting it all together, we get:
$r_u = \frac{1}{Z}\sqrt{X^2 + Y^2}$
$x_i\ = f_x x_u \frac{r_d}{r_u} + C_x\ = f_x \frac{X}{Z} \frac{1}{w} atan\left(2 r_u tan(\frac{w}{2})\right) / r_u\ + C_x$
$y_i\ = f_x y_u \frac{r_d}{r_u} + C_y\ = f_y \frac{Y}{Z} \frac{1}{w} atan\left(2 r_u tan(\frac{w}{2})\right) / r_u + C_y$

The inverse direction: From distorted pixel coordinates to normalized image coordinates

This inverts above equations, going from distorted pixel coordinates (x_i, y_i) to normalized image coordinates (x_u, y_u).
$x_d = (x_i - C_x)/f_x$
$y_d = (y_i - C_y)/f_y$
$r_d = \sqrt{x_d{}^2 + y_d{}^2}$
$r_u = tan(r_d w)\ /\ (2 tan(\frac{w}{2}))$
$x_u = x_d \frac{r_u}{r_d}$
$y_u = y_d \frac{r_u}{r_d}$
The associated ray is given by:
$X = x_u Z$
$Y\ = y_u Z$

Interpretation of the w parameter

The w parameter is used to describe the radial distortion introduced by the camera lens.

- For $w = 2atan\left(\frac{1}{2}\right) = 0.927295..\ \Leftrightarrow\ 2tan(\frac{w}{2}) = 1$, the model describes an equidistant fisheye lens where r_d is proportional to the angle θ between projected point and Z axis: $r_d = \frac{1}{w}\ atan(r_u) = \frac{1}{w} \cdot \theta$.
- For $w \rightarrow 0$, the model describes a pinhole camera where $x_i = f_x \frac{X}{Z} + C_x$ (y_i analogous). The intuitive explanation is that for small x, $tan(x) = atan(x) = x$.

Comparison with Devernay Model

The model described in [1] differs from the FOV model in that it represents focal length and pixel aspect ratio as f and S_x, instead of using two focal lengths f_x, f_y. It also puts the multiplication with f into the projection stage. Like the FOV model, it has one distortion parameter called ω. There is a direct mapping between the parameters used by Devernay and the FOV model, so even though the parameter values may be different, both camera models can equally describe the same lens.

The following gives an overview of the equations used for comparison. A more detailed description can be found in the paper.

$$x_u = f\,\frac{X}{Z}$$
$$y_u = f\,\frac{Y}{Z}$$
$$r_u = \sqrt{{x_u}^2 + {y_u}^2}$$
$$r_d = \frac{1}{\omega}\,atan(2r_u tan(\frac{\omega}{2}))$$
$$x_d = x_u \frac{r_d}{r_u}$$
$$y_d = y_u \frac{r_d}{r_u}$$
$$x_i = S_x x_d + C_x$$
$$y_i = y_d + C_y$$

$$r_u = \frac{f}{Z}\sqrt{X^2 + Y^2}$$
$$x_i = S_x x_u \frac{r_d}{r_u} + C_x = S_x f \frac{X}{Z}\frac{1}{\omega}\,atan\left(2r_u tan(\frac{\omega}{2})\right)/r_u + C_x$$

References

[1] *Frédéric Devernay, Olivier Faugeras. Straight lines have to be straight: automatic calibration and removal of distortion from scenes of structured enviroments. Machine Vision and Applications, Springer Verlag (Germany), 2001, 13 (1), pp.14-24. <10.1007/PL00013269>*