Conversion of OpenCV to OpenTrackIO (OpenLensIO) lens calibration parameters

If OpenCV lens calibration parameters are available for a camera/lens combination, how are these parameters converted to corresponding OpenTrackIO (OpenLensIO) lens calibration parameters?

OpenCV definitions of parameters can be found here:

`https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html`

whereas OpenTrackIO/OpenLensIO definitions of parameters can be found here:

`https://www.opentrackio.org/`

(search for "OpenLensIO mathematical lens model") and will be referred to from now on as just OpenTrackIO parameters (omitting OpenLensIO).

A few additional parameters for OpenTrackIO will be helpful and are defined first:

The radius r_u (distance from origin) of undistorted screen coordinates ϵ_u:

$$r_u = \sqrt{{\epsilon'_{x,u}}^2 + {\epsilon'_{y,u}}^2}$$

The pixel coordinates ϵ_s, here expressed as shader coordinates ε_{shader} but relative to the image center (not the upper left corner) and using a texture that has the same resolution as the camera image with width w_{shader} and height h_{shader} (so the camera image is w_{shader} pixels wide and h_{shader} pixels high, and h_{shader} is different from w_{shader} as opposed to the assumption of a square texture in the OpenLensIO documentation):

$$\epsilon_{x,s} = \varepsilon_{x,shader} - \frac{w_{shader}}{2} = w_{shader} \cdot \frac{\epsilon_{x,d}}{w} = \frac{w_{shader}}{w}(\epsilon'_{x,d} + \Delta\mathbb{P}_x)$$

$$\epsilon_{y,s} = \varepsilon_{y,shader} - \frac{h_{shader}}{2} = h_{shader} \cdot \frac{\epsilon_{y,d}}{h} = \frac{h_{shader}}{h}(\epsilon'_{y,d} + \Delta\mathbb{P}_y)$$

Parameters $l_1, ..., l_6$ are the radial, q_1, q_2 the tangential distortion parameters of the distortion function, which is the inverse of undistortion function U (formerly named D) defined in the OpenLensIO documentation with $k_1, ..., k_6$ as its radial, p_1, p_2 as its tangential distortion parameters (not to be confused with the OpenCV parameters of the same name but referring to the inverse operation).

Using these additional definitions, the following table lists corresponding parameters for both lens models:

OpenCV	Parameter	OpenTrackIO
x', y'	undistorted screen coordinates	$\epsilon'_{x,u}, \epsilon'_{y,u}$
r	radius of undistorted coordinates	r_u
x'', y''	distorted screen coordinates	$\epsilon'_{x,d}, \epsilon'_{y,d}$
u, v	pixel coordinates	$\epsilon_{x,s}, \epsilon_{y,s}$
f_x, f_y	focal length	F
c_x, c_y	principal point / projection offset	$\Delta\mathbb{P}_x, \Delta\mathbb{P}_x$
$k_1, ..., k_6$	radial distortion	$l_1, ..., l_6$
p_1, p_2	tangential distortion	q_1, q_2

The following equations follow straight from the parameter definitions:

$$\begin{aligned} \epsilon'_{x,u} &= F \cdot x' & \epsilon'_{x,d} &= F \cdot x'' & \epsilon_{x,s} &= u \\ \epsilon'_{y,u} &= F \cdot y' & \epsilon'_{y,u} &= F \cdot y'' & \epsilon_{y,s} &= v \\ r_u &= F \cdot r \end{aligned}$$

Focal length and projection offset parameters can be converted as follows:

$$f_x \cdot x'' + c_x = u = \epsilon_{x,s} = \frac{w_{shader}}{w}(\epsilon'_{x,d} + \Delta\mathbb{P}_x) = \frac{w_{shader}}{w} \cdot F \cdot x'' + \frac{w_{shader}}{w} \cdot \Delta\mathbb{P}_x$$

Since this equation is true for any value of x'' it follows that

$$F = \frac{w}{w_{shader}} \cdot f_x \qquad \wedge \qquad \Delta\mathbb{P}_x = \frac{w}{w_{shader}} \cdot c_x$$

Likewise in y direction:

$$f_y \cdot y'' + c_y = v = \epsilon_{y,s} = \frac{h_{shader}}{h}(\epsilon'_{y,d} + \Delta\mathbb{P}_y) = \frac{h_{shader}}{h} \cdot F \cdot y'' + \frac{h_{shader}}{h} \cdot \Delta\mathbb{P}_y$$

$$\Rightarrow \qquad F = \frac{h}{h_{shader}} \cdot f_y \qquad \wedge \qquad \Delta\mathbb{P}_y = \frac{h}{h_{shader}} \cdot c_y$$

Comparing individual components in the lens distortion formulas leads to the following conversions for distortion parameters, starting with radial distortion parameters in the numerator of the formula:

$$\begin{aligned} k_1 \cdot r^2 = l_1 \cdot r_u^2 = l_1 \cdot F^2 \cdot r^2 \qquad &\Rightarrow \qquad l_1 = \frac{k_1}{F^2} \\ k_2 \cdot r^4 = l_3 \cdot r_u^4 = l_3 \cdot F^4 \cdot r^4 \qquad &\Rightarrow \qquad l_3 = \frac{k_2}{F^4} \\ k_3 \cdot r^6 = l_5 \cdot r_u^6 = l_5 \cdot F^6 \cdot r^6 \qquad &\Rightarrow \qquad l_5 = \frac{k_3}{F^6} \end{aligned}$$

Likewise for the radial distortion parameters in the denominator of the formula:

$$l_2 = \frac{k_4}{F^2} \qquad l_4 = \frac{k_5}{F^4} \qquad l_6 = \frac{k_6}{F^6}$$

Finally, the same type of comparison leads to this conversion of tangential distortion parameters:

$$p_1 \cdot x' \cdot y' = q_1 \cdot \epsilon'_{x,u} \cdot \epsilon'_{y,u} = q_1 \cdot F \cdot x' \cdot F \cdot y' \qquad \Rightarrow \qquad q_1 = \frac{p_1}{F^2}$$

$$\begin{aligned} p_2(r^2 + 2x'^2) &= q_2(r^2 + 2{\epsilon'_{x,u}}^2) \\ &= q_2((F \cdot r)^2 + 2(F \cdot x')^2) \\ &= q_2 \cdot F^2(r^2 + 2x'^2) \qquad \Rightarrow \qquad q_2 = \frac{p_2}{F^2} \end{aligned}$$

This is a summary of all conversions:

focal length:	$F = \frac{w}{w_{shader}} \cdot f_x = \frac{h}{h_{shader}} \cdot f_y$		
projection offset:	$\Delta\mathbb{P}_x = \frac{w}{w_{shader}} \cdot c_x$		$\Delta\mathbb{P}_y = \frac{h}{h_{shader}} \cdot c_y$
radial distortion:	$l_1 = \frac{k_1}{F^2}$	$l_3 = \frac{k_2}{F^4}$	$l_5 = \frac{k_3}{F^6}$
	$l_2 = \frac{k_4}{F^2}$	$l_4 = \frac{k_5}{F^4}$	$l_6 = \frac{k_6}{F^6}$
tangential distortion:	$q_1 = \frac{p_1}{F^2}$	$q_2 = \frac{p_2}{F^2}$	

Important note: The OpenCV parameters implicitly depend on the pixel resolution of the camera image (w_{shader}, h_{shader}) and the size of the image sensor (w, h). By applying the conversions above, the resulting OpenTrackIO parameters do *not* depend on camera image resolution and image sensor size anymore. It is therefore possible (within practical limits) to replace the camera or switch it to a different sensor resolution (different sampling mode) without the need to recalibrate all lens parameters!