OpenLensIO Lens Model Version 1.0.0

Key authors: Martin Parsley, Robert North, Hendrik Fehlis, James Uren

17th February 2025

Contents

Part I
The Model

1 Definitions and conventions

1.1 Definitions and measurement units

Pinhole camera: A simple camera model where all light passes through a single point.
Optical axis: The axis passing through the centre of projection and pinhole.

F The focal length of lens, in mm, that projects the undistorted image.

w Width of the active area of the image sensor, in mm.

R Rotation part of the extrinsic matrix.

t Translation part of the extrinsic matrix.

$[R|t]_s$ Camera body component of the extrinsic matrix relative to the centre of the active area of the camera sensor (displayed on the uncropped screen).

$[\ ]_W$ The subscript W denotes coordinates in the world frame. The units are metres and radians.

$[\ ]_p$ The subscript p denotes coordinates in the pinhole-model camera frame. The units are metres and radians.

$[\ ]_u, [\ ]_d$ Subscripts u and d on vectors refer to undistorted and distorted coordinates respectively.

$[\ ]_\Omega$ Subscript Ω on vectors denotes overscanned coordinates.

$\begin{bmatrix} \epsilon_x & \epsilon_y \end{bmatrix}_*^T$ In-image (screen) coordinates, where $*$ is a subscript referring to undistorted or distorted coordinates. Both the ϵ_x and ϵ_y are millimetres to correspond with w, and $\begin{bmatrix} 0 & 0 \end{bmatrix}_*^T$ is at the screen centre, as shown in Figure 1. ϵ_x is the coordinate in the horizontal direction +ve to the right, and ϵ_y in the vertical direction +ve downwards. The vector form is denoted $\epsilon_* = \begin{bmatrix} \epsilon_x & \epsilon_y \end{bmatrix}_*^T$.

ϵ'_u Undistorted screen coordinates in field of view characterisation.

ϵ_u Undistorted screen coordinates in projection matrix characterisation.

r_* The radius (norm) of in-image coordinates is $r_* = \sqrt{\epsilon_x^2 + \epsilon_y^2}$.

1.2 Coordinate frames

World frame

A right-handed world coordinate frame is used to define the pose of all other bodies. This is the world frame in which camera tracking system parameters are provided. The X_W, Y_W, Z_W vectors point in the world Right, Forward and Up directions respectively. Thus Z_W corresponds to height.

Camera frame

The camera frame is located at the effective pinhole location of the camera lens combination. This corresponds to entrance pupil and will vary based on the focus and zoom setting of the lens. The X_p, Y_p, Z_p vectors point Right, Down and Forward with respect to the lens body.

Image (screen) frame

In image space, the camera sensor's 2-dimensional array of photosite locations are defined in mm units relative to the image centre with ϵ_x pointing right and ϵ_y pointing down. Thus the image corresponds to the sensor area.

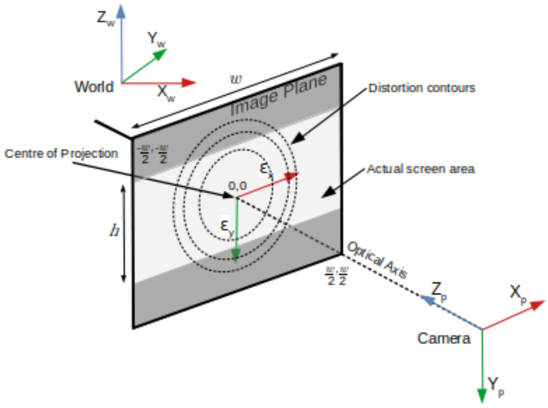

Figure 1: Coordinate frames including image coordinates. By defining the units in mm, means the actual screen area corresponds to the camera sensor area. The origin of the image coordinates is at the centre of the screen, with $-\frac{w}{2}, -\frac{w}{2}$ and $\frac{w}{2}, \frac{w}{2}$ at opposite diagonals for a square screen. As the axes are equal, the coordinates in the visible screen corners will extend from $\left(-\frac{w}{2}, -\frac{h}{2}\right)$ to $\left(\frac{w}{2}, \frac{h}{2}\right)$, while circular distortion appears circular.

1.3 Parameters of the OpenLensIO model

The OpenLensIO model defines a set of parameters to represent a compound lens with radial and decentering distortion. The required parameters are the static parameter w and the dynamic parameters F, $\Delta\mathbb{C}_x$, $\Delta\mathbb{C}_y$, $\Delta\mathbb{P}_x$, $\Delta\mathbb{P}_y$, $\mathtt{k}_{1\ldots6}$, z_{epd}, Φ. The optional parameters are the dynamic parameters Ω, , Ω', p_1, p_2, $a_{1\ldots3}$ and the static parameters Ω_{max}, Ω'_{max}. Dynamic parameters may vary based on the state of the lens (including zoom, focus and iris positions), whereas static parameters are independent of this. The OpenLensIO parameters are defined in the following sections.

1.4 Summary model equations

World coordinates are $\begin{bmatrix} x & y & z & 1 \end{bmatrix}_W^T$.

Conversion into camera frame is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_p = [R|t]_s \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_W - \begin{bmatrix} 0 \\ 0 \\ z_{epd} \end{bmatrix}. \tag{1}$$

where $[R|t]_s$ is the extrinsic matrix component at the centre of the active area of the camera sensor, generally derived from the tracking system pose, and z_{epd} is the entrance pupil distance. We assume that transverse offsets of the entrance pupil (x_{epd} and y_{epd}) are negligible compared to the longitudinal movement (z_{epd}), and their effects are adequately modelled by the perspective offset terms $\Delta\mathbb{P}$, defined below.

In the pinhole camera model, the coordinates are projected on an image plane (normalised by depth) as follows;

$$\begin{bmatrix} x \\ y \end{bmatrix}_u = \frac{1}{z_p} \begin{bmatrix} x \\ y \end{bmatrix}_p, \tag{2}$$

To support differing rendering pipeline requirements, two alternatives are presented for characterising a camera. These models vary in the undistorted coordinates, but preserve the same relationship between

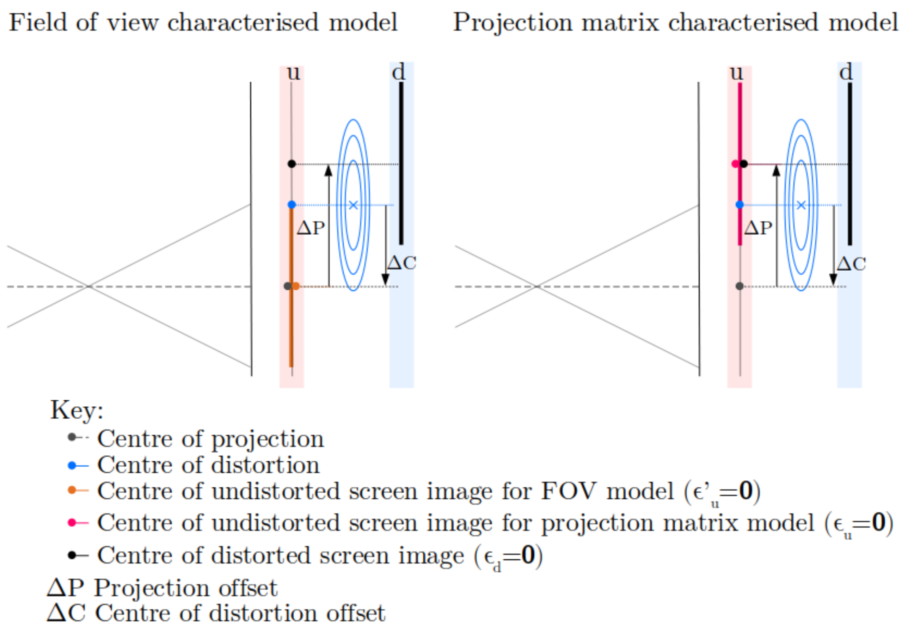

Figure 2: Diagram of a pinhole camera showing the locations of the five main centres on the imaging plane, and the offsets between them, $\Delta\mathbb{P}$ and $\Delta\mathbb{C}$. The two models differ in the location of the undistorted screen centre.

the scene and the distorted camera coordinates. These models are referred to as the *projection matrix characterisation* and the *field of view characterisation*.

1.5 Defining centre and perspective offsets

Informally, both characterisations require a geometric definition of two types of translation, a perspective offset $\Delta\mathbb{P}$ that is applied to the camera projection before the distortion is applied, and a centre offset $\Delta\mathbb{C}$ in the distorted image plane.

$\Delta\mathbb{C} = \begin{bmatrix} \Delta\mathbb{C}_x & \Delta\mathbb{C}_y \end{bmatrix}^T$ represents an offset of the distortion centre in units of mm, and $\Delta\mathbb{P} = \begin{bmatrix} \Delta\mathbb{P}_x & \Delta\mathbb{P}_y \end{bmatrix}^T$ represents a perspective offset from the centre of the screen, also in units of mm. These are shown graphically in Figure 2.

The formal definitions are as follows:

- $\Delta\mathbb{P}$ is the translation from undistorted coordinate frame, ϵ_u, sharing centre with distorted frame, ϵ_d, to the coordinate frame centred an the centre of projection, ϵ'_u.
- $\Delta\mathbb{C}$ is the translation from coordinate frame centred on the centre of projection, ϵ'_u, to the coordinate frame centred at the centre of distortion.
- ϵ_u and ϵ_d are defined such that they share the same centre, independent of $\Delta\mathbb{P}$ and $\Delta\mathbb{C}$.

Behaviour of $\Delta\mathbb{P}$ and $\Delta\mathbb{C}$:

- When viewed in the coordinate system ϵ_d, $\Delta\mathbb{P}$ moves the centre of projection, and centre of distortion equally. If $\Delta\mathbb{C} = 0$, then centre of projection and centre of distortion will be coincident.
- When viewed in the coordinate system ϵ_d, $\Delta\mathbb{C}$ moves the centre of distortion without changing the centre of projection.

In essence the distortion centre offset $\Delta\mathbb{C}$ allows for offsetting the centre of distortion from the centre of the screen, while the perspective offset $\Delta\mathbb{P}$ allows for offsetting the centre of perspective, and centre of distortion jointly.

The perspective offset is applied on the distortion side of the model to enable use in environments where perspective offset cannot be applied to the projection matrix, e.g. a virtual camera specified only in terms of pose and field of view.

2 Projection matrix characterisation camera model

The projection matrix characterisation of the model can be defined by the following equations. The undistortion function U, defined in section 4.1, maps from distorted screen coordinates to undistorted screen coordinates, this direction being convenient for implementing in a GPU shader:

$$\epsilon_u = F \begin{bmatrix} x \\ y \end{bmatrix}_u + \Delta\mathbb{P} \tag{3}$$

$$\epsilon_u = U\left(\epsilon_d - \Delta\mathbb{C} - \Delta\mathbb{P}\right) + \Delta\mathbb{C} + \Delta\mathbb{P}. \tag{4}$$

This can be re-stated to use the distort model U^{-1}

$$\epsilon_d = U^{-1}\left(\epsilon_u - \Delta\mathbb{C} - \Delta\mathbb{P}\right) + \Delta\mathbb{C} + \Delta\mathbb{P}. \tag{5}$$

With these equations we can define an angle of view

$$\frac{r_u}{F} = tan(\frac{\alpha}{2}). \tag{6}$$

But this is centred at the point ΔP, so cannot be used to construct a projection matrix directly. Please refer to section 3 for the field of view characterisation model.

2.1 Applying overscan

A virtual camera may produce an image with unrendered areas when distortion is applied, for instance in the corners for a lens with barrel distortion. To remedy this, we apply overscan; where the virtual camera requires a larger image plane and thus a wider field of view than the pinhole camera. After rendering, in order for the projection of the final rendered image to match that of the real camera, it is necessary to zoom into the image by the inverse of the overscan factor.

The projection matrix with overscan factor Ω is

$$\epsilon_\Omega = \frac{1}{\Omega}\left(F \begin{bmatrix} x \\ y \end{bmatrix}_u + \Delta\mathbb{P}\right). \tag{7}$$

A similar undistort transform, with overscan is

$$\epsilon_\Omega = \frac{1}{\Omega}\left(U\left(\epsilon_d - \Delta\mathbb{C} - \Delta\mathbb{P}\right)\right). \tag{8}$$

Overscan is typically either provided as a constant multiplier, or computed by approximation.

To help users that require a constant multiplier, OpenLensIO optionally provides Ω_{max} as a static parameter, the largest Ω across all zoom and focus settings. To preserve quality, it may be necessary to render the undistorted image with a higher screen percentage than the distorted image.

3 Field of view characterisation camera model

The field of view characterisation of the model can be defined by the following equations

$$\epsilon'_u = \begin{bmatrix} \epsilon_x \\ \epsilon_y \end{bmatrix}'_u = F \begin{bmatrix} x \\ y \end{bmatrix}_u. \tag{9}$$

The general form for the conversion of screen coordinates is given by

$$\epsilon'_u = U\left(\epsilon'_d - \Delta\mathbb{C}\right) + \Delta\mathbb{C}, \tag{10}$$

where $\epsilon_d = \epsilon'_d + \Delta\mathbb{P}$, with the inverse given by

$$\epsilon'_d = U^{-1}\left(\epsilon'_u - \Delta\mathbb{C}\right) + \Delta\mathbb{C}, \tag{11}$$

which can be solved using numerical iterative methods depending on the application.

The field of view and projection matrix characterisations are related by

$$\epsilon_u = \epsilon'_u + \Delta\mathbb{P}, \tag{12}$$

$$\epsilon_d = \epsilon'_d + \Delta\mathbb{P}. \tag{13}$$

3.1 Applying overscan

When characterising the virtual camera by field of view, overscan is applied to the field of view of the virtual camera to ensure the final rendered image matches that of the real camera. As with the projection characterisation it is necessary to zoom into the image by the inverse of the overscan factor.

The overscanned field of view of the virtual camera is given by

$$\theta_{\Omega'} = 2\ tan^{-1}\left(\frac{w\Omega'}{2F}\right), \tag{14}$$

where $w\Omega'$ can be thought of as the overscanned sensor width.

Then after rendering, the screen coordinates are scaled by the inverse of the overscan factor

$$\epsilon'_{\Omega'} = \frac{1}{\Omega'}\left(U\left(\epsilon_d - \Delta\mathbb{C} - \Delta\mathbb{P}\right) + \Delta\mathbb{C}\right). \tag{15}$$

OpenLensIO provides both $\Omega'$and $\Omega'_{max}$for the field of view characterisation model in same way as for projection matrix characterised model in section 2.1.

It can be proven that these equations for the field of view and projection matrix characterisations can generate equivalent renders when overscan is applied, but overscan computed on one form is not guaranteed to fill the screen of the other form.

4 Common aspects of the model

Much of the OpenLensIO model is common to both characterisations.

4.1 Distortion model

The undistortion function $U(\epsilon)$ uses the Brown-Conrady distortion model to map from distorted screen coordinates to undistorted screen coordinates,

$$U(\boldsymbol{\epsilon}) = \text{diag}\begin{bmatrix} R + 2p_1\epsilon_y + p_2\left(\frac{1}{\epsilon_x}r^2 + 2\epsilon_x\right) \\ R + 2p_2\epsilon_x + p_1\left(\frac{1}{\epsilon_y}r^2 + 2\epsilon_y\right)\end{bmatrix}\boldsymbol{\epsilon} \tag{16}$$

where R is the radial distortion term

$$R = \frac{1 + \mathtt{k}_1 r^2 + \mathtt{k}_3 r^4 + \mathtt{k}_5 r^6}{1 + \mathtt{k}_2 r^2 + \mathtt{k}_4 r^4 + \mathtt{k}_6 r^6} \tag{17}$$

with radial distortion coefficients $\mathtt{k}_{1,3,5}$ in units of mm^{-n-1}, $\mathtt{k}_{2,4,6}$ in units of mm^{-n}, and p_n are the decentering coefficients in units of mm^{-2}, and where the radius $r = \sqrt{\epsilon_x^2 + \epsilon_y^2}$. Note that coefficient subscripts are alternating numerator and denominator in order that the distortion model is extensible to higher powers.

4.2 Other screen units

Equation (8) gives the screen coordinates in mm relative to the centre of the screen, however for implementation in shaders the screen coordinates need to be mapped to shader coordinates. For example, for a square texture of width w_{shader} with origin at the top left, the ϵ coordinates (in mm) can be converted into shader coordinates, ε_{shader}, by

$$\varepsilon_{shader} = w_{shader}\begin{bmatrix}\frac{\epsilon_x}{w} \\ \frac{\epsilon_y}{h}\end{bmatrix} + \frac{w_{shader}}{2}, \tag{18}$$

w and h are the sensor width and height in mm.

And for the field of view characterisation camera model ε'_{shader} has an identical relationship to ε'.

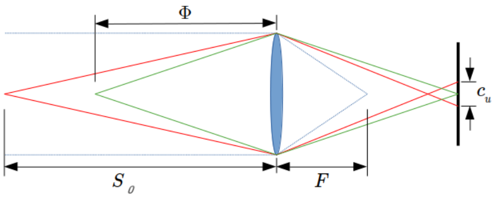

Figure 3: Diagram showing the circle of confusion, c_u for a simple paraxial optical system.

4.3 Aperture

The accuracy of the following accepted expression for circle of confusion is currently under investigation:

$$c_u = \frac{|S_o - \Phi|}{S_o} \frac{F^2}{N\,(\Phi - F)} \tag{19}$$

where c_u is the circle of confusion in mm, in undistorted screen space, N is the f-number of the lens (f-stop), F is the focal length, Φ is the focal distance and S_o is the object distance (which comes from the scene), as shown in Figure 3.

Note this requires the f-stop (aperture) of the lens. T-stop (transmission) cannot be used directly in this equation. Conversion from t-stop to f-stop is normally via measurement or a lookup table that can be provided by the lens manufacturer.

4.4 Vignetting

OpenLensIO models optical vignetting, $\upsilon_n(r)$. This occurs when stacking multiple lenses which reduces the intensity of the light towards the outside of the image. This is modelled by the polynomial

$$\upsilon_n(r) = 1 - (\alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6), \tag{20}$$

where the radius $r = \sqrt{\epsilon_x^2 + \epsilon_y^2}$.

A Informative sections

A.1 Specifying ideal overscan

The ideal overscan factor can be approximated by the render engine based on a selection of the screen coordinates, and may be recorded for future re-rendering. The minimum required overscan Ω and resulting field of view θ_Ω are defined as follows.

If we define the set of coordinates inside the distorted screen rectangle to be

$$\{\boldsymbol{\epsilon_d^i}\} = \{\epsilon_d : -\frac{w}{2} < \epsilon_d.x < \frac{w}{2} \wedge -\frac{h}{2} < \epsilon_d.y < \frac{h}{2}\}. \quad (21)$$

For the projection matrix characterisation, we define $\{\boldsymbol{\epsilon_u^i}\}$ to be the subset of corresponding undistorted coordinates

$$\{\boldsymbol{\epsilon_u^i}\} = \{\boldsymbol{\epsilon_u^i} : \boldsymbol{\epsilon_u^i} = \left(U\left(\boldsymbol{\epsilon_d^i} - \Delta\mathbb{C} - \Delta\mathbb{P}\right) + \Delta\mathbb{C} + \Delta\mathbb{P}\right)\}. \quad (22)$$

Then the smallest image width, w_Ω that ensures that the overscanned region contains all of $\{\boldsymbol{\epsilon_u^i}\}$ given by

$$w_\Omega = \quad 2\ max\left(\left\{max_i\left(\left\{\left|\epsilon_x^i\right|\right\}\right), \tfrac{w}{h}max_i\left(\left\{\left|\epsilon_y^i\right|\right\}\right)\right\}\right). \quad (23)$$

From this we can derive the overscan factor, Ω,

$$\Omega = \frac{w_\Omega}{w}. \quad (24)$$

This can then be used as defined in section 3.1. The same methodology can be applied for the field of view characterisation, by modifying the definition of $\boldsymbol{\epsilon_u^i}$ in Equation (22).

B Future additions

B.1 Entrance pupil off-axis offset, and optical axis rotation.

Currently the entrance pupil off-axis offset and rotation have been considered negligible. The off-axis offset could be considered in order to better represent wide angle lenses when imaging something 1 metre or closer, and for rotation, may be necessary to express misalignment of the lens elements, or the mount of the lens to the camera body.

B.2 Anamorphic lens distortion

To undistort an anamorphic lens, we could use a distortion model (just showing radial terms for readability, tangential terms can be added), however this needs further investigation and the practical evaluation of several anamorphic models.

$$\begin{bmatrix} x \\ y \end{bmatrix}_u = \begin{bmatrix} \frac{x_d}{1+\left(\frac{k_1}{A_{sq}}\right)r^2+\left(\frac{k_2}{A_{sq}}\right)r^4+\left(\frac{A_x}{A_{sq}}\right)y_d^2} \\ \frac{y_d}{1+k_1r^2+k_2r^4+A_yx_d^2} \end{bmatrix} \tag{25}$$

where k_1, k_2 are the radial distortion terms, A_x, A_y are the asymmetric distortion terms and A_{sq} is anamorphic squeeze. The rest of the pipeline remains unchanged.

B.3 Chromatic aberration

Axial (longitudinal)

Most noticeable at long focal lengths. A lens with specific refractive index cannot effectively “focus” all wavelengths of light at the same time. If blue is in focus, red and green will be out of focus and will appear in a different image location. This could be modelled by having three sets of focal distances.

Transverse (lateral)

Magnification and distortion of the lens is also dependent on wavelength. This could be modelled with three sets of distortion parameters, one for each primary colour.

B.4 Vignetting

OpenLensIO models optical vignetting, however natural and mechanical vignetting could also be modelled.

Natural

Light that hits the sensor at an angle has lower intensity based on Cosine fourth law of optical falloff. Proposition coming in a future revision.

Mechanical

This occurs when physical lens components are in the way of the lens. To be considered (even if negligible) in a future revision.