The Standard Models for Lens Distortion in 3DE4

U. Sassenberg, Science-D-Visions

March 25, 2021

Contents

1 About this document

We describe the general concept of lens distortion as we understand it in the context of 3DE4, and the two most important lens distortion models, one for radial lenses and one for anamorphic lenses, including mathematical details and modifications based on input from users.

1.1 License

1.2 Current status

The document is under development. It is valid as of LDPK version 2.0.

Doc Vers.	Scope	Date	Notes
1.4	public	2018-12-14	First public release
1.3	internal	2018-09-19	Finetuning notation Bugfixes in reparametrization
1.2	internal	2018-07-09	Rescaled anamorphic model
1.1	internal	2018-06-22	Reparametrization
1.0	internal	2018-06-18	Coordinate systems; Extenders; The two standard models of 3DE4

2 Preliminaries

2.1 Basic concepts

In order to model lens distortion, we abstract from the real-world camera towards an idealized camera. The lens distortion models we are looking for do not contain any explicit time-dependency, and it does not care about chromatic effects. In order to point out what we understand by lens distortion we shall define a couple of terms in the following.

2.1.1 Physical filmback / imaging area

For a given point in time the output of a camera is an image, i.e. a mapping from a rectangular region in $\mathbb{R}^2$ into some target space (usually, but not necessarily, color space). We shall assume, that this image has a counterpart within the camera, i.e. a rectangular region on an image sensor or a strip of celluloid. We call this rectangle the *physical filmback* with width $w_{\text{fb,cm,phys}}$ and height $h_{\text{fb,cm,phys}}$. The index “cm” should remind us that these quantities are given in length units (be it centimeter or inch) as opposed to pixels or dimensionless quantities.

2.1.2 Pixel aspect ratio and virtual filmback

Whenever an image is rasterized from / displayed on a rectangle with real world length units, pixel aspect becomes an important quantity. For a digital imaging system the definition is the following: Assume the image of a rectangular object of size $w_{\text{cm}} \times h_{\text{cm}}$ without non-linear distortion is given in rasterized form with pixel size $w_{\text{px}} \times h_{\text{px}}$ so that the edges of the object are parallel to the edges of the rasterized image, then pixel aspect (ratio) is defined as

$$r_{\text{pa}} = \frac{w_{\text{cm}}}{h_{\text{cm}}} \frac{h_{\text{px}}}{w_{\text{px}}} \tag{1}$$

In the “VFX Database” [2], the workflow for anamorphic images is shown. We adopt the term *virtual filmback* for our purposes and write its size as

$$w_{\text{fb,cm}} \times h_{\text{fb,cm}} \tag{2}$$

The virtual filmback width for an anamorphic camera is the filmback as it would be without pixel aspect, so that we have the relation

$$w_{\text{fb,cm}} = w_{\text{fb,cm,phys}} r_{\text{pa}} \tag{3}$$

In 3DE4’s GUI, the camera / lens is described as a camera pyramid with size $w_{\text{fb,cm}} \times h_{\text{fb,cm}} \times f_{\text{cm}}$, where f_{cm} is the focal length. If an image of size $w_{\text{px}} \times h_{\text{px}}$ is loaded and associated to a lens, pixel aspect is calculated and displayed. It is important to keep in mind that

All lens distortion models in 3DE4 are based on the *virtual filmback*.

More precisely, the unit coordinates and the diagonally normalized coordinates we define later in this document are derived from virtual filmback size.

The quantities (as they appear in 3DE4's GUI) *Filmback Width*, *Filmback Height*, *Pixel Aspect*, image width and image height fulfill the definition given in equation (1). Since this is important, we should consider an example (see Figure 1). Assume, an anamorphic camera has a physical filmback of 21.936 mm × 18.672 mm. After rasterization we have images of size 1828 px × 1556 px. Then pixel aspect is 2.0. The virtual filmback width is 43.872 mm, and 3DE4 will display

Filmback Width	43.872 mm
Filmback Height	18.672 mm
Film Aspect	2.353
Pixel Aspect	2.000
Resolution	1828 × 1556

Whenever we write $w_{\mathrm{fb,cm}}$ in this document, we shall refer to the virtual filmback width, not the physical filmback width. Also, lens center offset in the context of 3DE4 (see next section) is defined with respect to the virtual filmback.

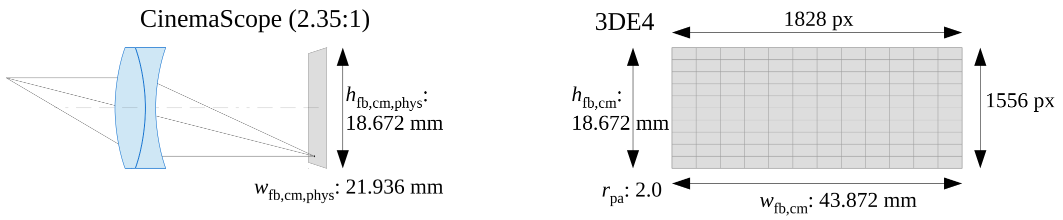

Figure 1: *Numeric example, physical vs. virtual filmback*

2.1.3 Lens Center

In a perfectly aligned optical system, all lenses are centered on the optical axis. We shall call the intersection of this axis with the filmback the *lens center*. This is again an idealized model, and experience shows that even slight displacements of the lenses from the optical axis have an impact on lens distortion. At least one of the models we present in this document does account for these displacements. The important thing is that there must be a point on the filmback which remains invariant under the lens distortion mapping (mathematically a *fixed point*), and we will consider this point as lens center. In practice, as you will see below, we will express our models by means of the difference vector between the filmback center and the lens center, which we call *lens center offset* with symbols $x_{\mathrm{lco,cm}}$ and $y_{\mathrm{lco,cm}}$.

2.1.4 Lens Distortion

In an ideal, linear camera, the relation between points in 3d-space and the corresponding point on the filmback is a homography, i.e. straight lines in 3d-space are mapped onto straight lines on the filmback. *Lens distortion* is a typically non-linear mapping from the filmback (plus some margin) onto the filmback (plus some other margin). This non-linearity results from the complex physics of real-world lens systems. Our objective is to model lens distortion with only little physical input, so that we can formulate it independently from the specific lens system.

3 Lens distortion

3.1 Coordinates

Before constructing our lens distortion models we need to define the coordinate systems involved. Starting from the virtual filmback with size $w_{\text{fb,cm}} \times h_{\text{fb,cm}}$ and lens center offset $(x_{\text{lco,cm}}, y_{\text{lco,cm}})$ we shall derive the following two systems:

1. *unit coordinates* (index: “unit”). In these coordinates the lower left position of the filmback has values (0,0), while the upper right position is (1,1). We use these coordinates for implementing the data structures in 3DE4 used for dispatching tracking data. On one hand, unit coordinates are more appropriate than length-unit coordinates, because they abstract from the camera; images do not necessarily come from a camera (not even a virtual one), so we prefer not to deal with length units. On the other hand, unit coordinates are better than e.g. pixel coordinates, since the distortion classes in 3DE4 have to work without recurse to image size (in pixel). For this reason we choose unit coordinates for any API when dealing with tracking data and lens distortion within 3DE4.

2. *diagonally normalized coordinates* (index “dn”) In these coordinates the origin (0,0) coincides with the lens center, and the image diagonal has a length of 2, i.e. the radius is 1. Additionally, we demand that the coordinates are supposed to be *isometric*: A line segment of length distance d_{dn} in diagonally normalized coordinates corresponds to a distance d_{cm} in virtual filmback coordinates regardless of the line segment’s orientation. We shall use these coordinates as base for our lens distortion models, because it is quite natural to have the origin (0,0) as fixed point of the distortion mapping.

The objective of this section is the following: Given a filmback and a lens center offset

$$w_{\text{fb,cm}}, h_{\text{fb,cm}}, x_{\text{lco,cm}}, y_{\text{lco,cm}} \tag{4}$$

we would like to express diagonally normalized coordinates by unit coordinates and vice versa, i.e. we are looking for a mapping

$$\phi : (x_{\text{unit}}, y_{\text{unit}}) \mapsto (x_{\text{dn}}, y_{\text{dn}}) \tag{5}$$

This enables us to formulate lens distortion models easily in dn-coordinates on one hand and an API for lens distortion classes in unit-coordinates on the other hand. Given a model function

$$g : (x_{\text{dn}}, y_{\text{dn}}) \mapsto (x'_{\text{dn}}, y'_{\text{dn}}) \tag{6}$$

which maps a distorted point into its undistorted counterpart (denoted by the prime), a lens distortion class will perform the following operation:

$$\boxed{(x'_{\text{unit}}, y'_{\text{unit}}) = \phi^{-1} \circ g \circ \phi(x_{\text{unit}}, y_{\text{unit}})} \tag{7}$$

First, let us define the *radius* of the filmback, i.e. the EUCLIDIAN distance from filmback center to either corner.

$$r_{\text{fb,cm}} = \sqrt{\left(\frac{w_{\text{fb,cm}}}{2}\right)^2 + \left(\frac{h_{\text{fb,cm}}}{2}\right)^2} \tag{8}$$

If we define coordinates in length units in a way that the lower left corner of the filmback is $(0, 0)$, we can easily write down the relationship between length coordinates and unit-coordinates:

$$\begin{aligned} x_{\text{cm}} &= x_{\text{unit}} w_{\text{fb,cm}} \\ y_{\text{cm}} &= y_{\text{unit}} h_{\text{fb,cm}} \end{aligned} \tag{9}$$

Concerning diagonally normalized coordinates, in item 2 above we demand that the origin coincides with lens center and that the diagonal distance is 2. This leads us to the following relation:

$$\begin{aligned} x_{\text{dn}} &= \frac{x_{\text{cm}} - x_{\text{lc,cm}}}{r_{\text{fb,cm}}} \\ y_{\text{dn}} &= \frac{y_{\text{cm}} - y_{\text{lc,cm}}}{r_{\text{fb,cm}}} \end{aligned} \tag{10}$$

where $x_{\text{lc,cm}}$ and $y_{\text{lc,cm}}$ are the lens center. Obviously, the lens center in length coordinates is mapped to (0,0). Let us check the length of the diagonal. For the upper right corner of the filmback we have the following position in dn-coordinates:

$$\begin{aligned} x_{\text{right,dn}} &= \frac{w_{\text{fb,cm}} - x_{\text{lc,cm}}}{r_{\text{fb,cm}}} \\ y_{\text{top,dn}} &= \frac{h_{\text{fb,cm}} - y_{\text{lc,cm}}}{r_{\text{fb,cm}}} \end{aligned} \tag{11}$$

and for the lower left corner:

$$
\begin{aligned}
x_{\mathrm{left,dn}} &= \frac{-x_{\mathrm{lc,cm}}}{r_{\mathrm{fb,cm}}} \\
y_{\mathrm{bottom,dn}} &= \frac{-y_{\mathrm{lc,cm}}}{r_{\mathrm{fb,cm}}}
\end{aligned}
\tag{12}
$$

The EUCLIDIAN distance between these positions is

$$
\sqrt{\left(\frac{w_{\mathrm{fb,cm}}}{r_{\mathrm{fb,cm}}}\right)^2 + \left(\frac{h_{\mathrm{fb,cm}}}{r_{\mathrm{fb,cm}}}\right)^2} = 2 \tag{13}
$$

using definition (8). Finally, we express the lens center by lens center offset and insert relations (9):

$$
\boxed{
\begin{aligned}
x_{\mathrm{dn}} &= \frac{x_{\mathrm{unit}} w_{\mathrm{fb,cm}} - x_{\mathrm{lco,cm}}}{r_{\mathrm{fb,cm}}} - \frac{w_{\mathrm{fb,cm}}}{2r_{\mathrm{fb,cm}}} \\
y_{\mathrm{dn}} &= \frac{y_{\mathrm{unit}} h_{\mathrm{fb,cm}} - y_{\mathrm{lco,cm}}}{r_{\mathrm{fb,cm}}} - \frac{h_{\mathrm{fb,cm}}}{2r_{\mathrm{fb,cm}}}
\end{aligned}
}
\tag{14}
$$

For each given filmback $w_{\mathrm{fb,cm}}$, $h_{\mathrm{fb,cm}}$ and lens center offset $x_{\mathrm{lco,cm}}$, $y_{\mathrm{lco,cm}}$ we shall call this (affine) mapping

$$
(x_{\mathrm{dn}}, y_{\mathrm{dn}}) = \phi(x_{\mathrm{unit}}, y_{\mathrm{unit}}) \tag{15}
$$

and it inverse mapping

$$
(x_{\mathrm{unit}}, y_{\mathrm{unit}}) = \phi^{-1}(x_{\mathrm{dn}}, y_{\mathrm{dn}}). \tag{16}
$$

3.2 Lens distortion models

In this section we define what we understand by a distortion model in the context of 3DE4. Essentially, a distortion model is a function which maps each point in the filmback to some other point. Yet, a distortion model is not an automorphism on the filmback, so instead of defining a model function exactly on the filmback it makes more sense to define it as a mapping from a set $P \subset \mathbb{R}^2$ containing the filmback to some other set $Q \subset \mathbb{R}^2$.

Let P and Q be (open, connected) subsets of $\mathbb{R}^2$, so that $(0,0) \in P$ and $(0,0) \in Q$. The model functions we are going to define map from P to Q and usually depend on parameters, called the *distortion parameters*. A set of distortion parameters c is a tuple $(c_0, c_1, \ldots, c_{n-1})$ of values, each from a parameter domain C_i, i.e. $c_i \in C_i$. The C_i can be e.g. an interval $[a,b] \subset \mathbb{R}$, a set of integers or BOOLEAN values $\{0,1\}$. This definition is pretty vague, and in fact there are hardly any conditions for this parameter space. We now consider a mapping

$$
g : P \times C \to Q \tag{17}
$$

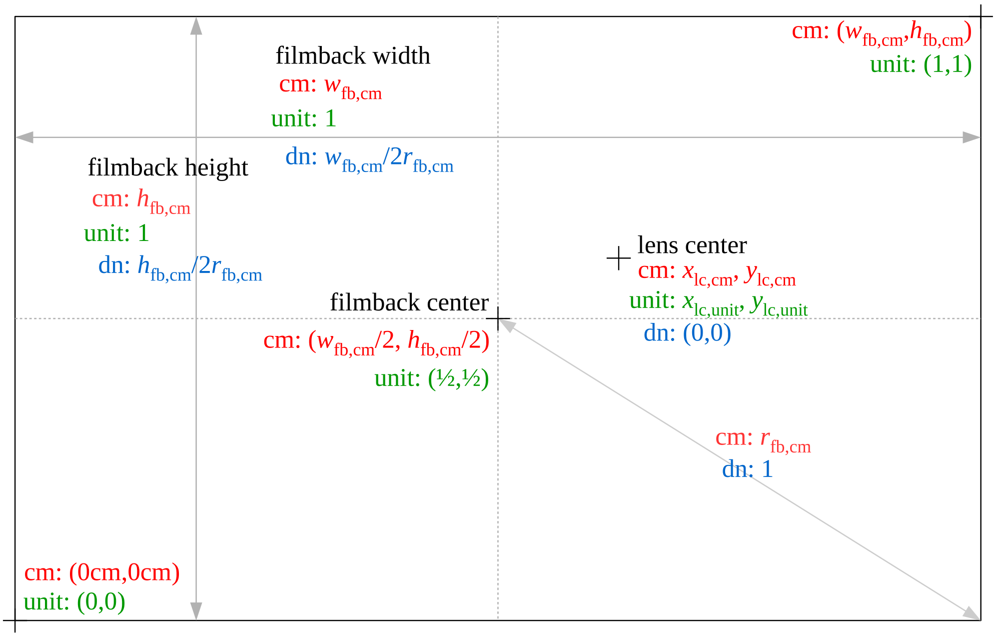

Figure 2: *Filmback and lens center in various coordinate systems*

For any given parameter set $c \in C$ we define the function g_c by

$$g_c : P \to Q : p \mapsto g(p, c) \tag{18}$$

We call g a *distortion model* if the following conditions are fulfilled:

1. **Fixed point** - $(0, 0)$ is a fixed point of g_c, i.e. $g_c(0, 0) = (0, 0)$. Since we express g_c in terms of dn-coordinates, (0,0) is the lens center, which remains invariant under g_c.

2. **Default parameters** - There is a parameter set $c_{\text{default}} \in C$, so that $g_{c_{\text{default}}}$ is the identity map on P, i.e.

$$g_{c_{\text{default}}} = \text{id}|_P \tag{19}$$

   We call c_{default} a *default parameter set* of g_c.

3. **Invertibility** - g_c is invertible on P, i.e. there is a mapping

$$g_c^{-1} : Q \to P \tag{20}$$

   so that

$$g_c^{-1} \circ g_c = \text{id}|_P \tag{21}$$

By definition, the function g_c maps **distorted** points into **undistorted** points. We refer to g_c as the *model function* of the distortion model. As a convention in this document we shall denote distorted points by (x, y) and undistorted points by (x', y') so that all distortion model functions map like

$$(x', y') = g_c(x, y) \tag{22}$$

The three conditions above are the minimum requirements for our distortion models. It should be mentioned that for a clean mathematical description we would have to add more conditions, like e.g. continuity or differentiability, but these conditions would not help us in getting a practical formulation of lens distortion models, so we will omit this for now. In sections 3.5 and 3.7 we will define the two lens distortion models most relevant in practice. As mentioned, all models are defined in dn-coordinates, even if we omit the index "dn".

3.3 Extenders

Some of the models used in 3DE4 are equipped with so-called extenders. At the current state of development an extender is a linear mapping

$$h_e : \mathbb{R}^2 \to \mathbb{R}^2 \tag{23}$$

parametrized by tuples $e = (e_0, \ldots, e_{n-1})$ from some parameter space E, similarly to the parameter space C of the model function. The conditions for model functions also apply to extenders:

1. **Fixed point** - Since h_e is linear, it is clear that the fixed point condition is fulfilled
$$h_e(0, 0) = (0, 0) \tag{24}$$
2. **Default parameters** - There is a parameter set $e_{\text{default}} \in E$, so that $h_{e_{\text{default}}}$ is the identity map on $\mathbb{R}^2$, i.e.
$$h_{e_{\text{default}}} = \text{id}|_{\mathbb{R}^2} \tag{25}$$
3. **Invertibility** - We demand that h_e is invertible, i.e. we can write[1] $h_e(x, y)$ as a product of an invertible 2×2-matrix H_e and a vector (x, y):
$$\begin{aligned} h_e(x, y) &= H_e \begin{bmatrix} x \\ y \end{bmatrix} \\ h_e^{-1}(x, y) &= H_e^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned} \tag{26}$$

[1]Our formulation is not exact here. We do not distinguish between tuples (x, y) and vectors $[x, y]^T$, but the meaning should be clear.

Given a distortion model g with parameter space C and an extender h with a parameter space E (disjoint to C) the composition $h_e \circ g_c$ is a model function as well. The fixed point condition is fullfilled:

$$h_e \circ g_c(0,0) = h_e(0,0) = (0,0) \tag{27}$$

The composition is invertible:

$$(h_e \circ g_c)^{-1} = g_c^{-1} \circ h_e^{-1} \tag{28}$$

and if we unite E and C into a parameter space $C' = E \times C$, we have a default parameter set $(e_{\text{default}}, c_{\text{default}}) \in C'$ for which the composition is the identity:

$$h_{e_{\text{default}}} \circ g_{c_{\text{default}}} = \mathrm{id}|_{\mathbb{R}^2} \circ \mathrm{id}|_P = \mathrm{id}|_P \tag{29}$$

Likewise, you can easily check that $g_c \circ h_e|_P$ is a model function for a parameter space $C \times E$.

In 3DE4's GUI the term "extender"does not appear. We simply incorporate the extender into the plain distortion model and present the united parameter set in the GUI. In this document however, it makes sense to separate the plain model and the extenders carefully.

3.4 Overview: Lens distortion models

In this document we define names for distortion models which describe the ingredients of the distortion model more precisely. Table 1 shows how these model names relate to the distortion models in 3DE4. Parameters ϕ_{bs}, b_{bs} and ϕ_{mnt} refer to "Cylindric Direction", "Cylindric Bending" and "Lens Rotation" which are explained later in this document.

Table 1: *Names of distortion models in this document vs.* 3DE4*'s GUI*

GUI	**Document**	**Spec. case**
3DE4 Radial - Standard, Degree 4	Poly-4-Radial	$u, v, \phi_{\text{bs}}, b_{\text{bs}} = 0$
3DE4 Radial - Standard, Degree 4	Poly-4-Radial-Decenter	$\phi_{\text{bs}}, b_{\text{bs}} = 0$
3DE4 Radial - Standard, Degree 4	Poly-4-Radial-Decenter-Elliptic	
3DE4 Anamorphic - Standard, Degree 4	Poly-4-Anamorphic	$s_{\text{x}}, s_{\text{y}} = 1, \phi_{\text{mnt}} = 0$
3DE4 Anamorphic - Standard, Degree 4	Poly-4-Anamorphic-Rpa-Sq	$\phi_{\text{mnt}} = 0$
3DE4 Anamorphic - Standard, Degree 4	Poly-4-Anamorphic-Rpa-Sq-Rot	
3DE4 Anamorphic - Rescaled, Degree 4	Poly-4-Anamorphic-Rpa-Re-Sq-Rot	

Whenever possible, we express our model functions by means of the Euclidian distance of a point $p = (x, y)$ from the lens center (0,0):

$$r = \sqrt{x^2 + y^2} \tag{30}$$

3.5 The standard model for radially symmetric lenses

In photography optical systems usually consist of a number of radially symmetric lenses lined up on the optical axis. An ideal system with perfectly aligned lenses can be modelled by a polynomial with even exponents of e.g. degree four (Poly-4-Radial),

$$
\begin{aligned}
x' &= x(1 + c_2 r^2 + c_4 r^4) \\
y' &= y(1 + c_2 r^2 + c_4 r^4).
\end{aligned} \tag{31}
$$

In practice the lenses can be slightly decentered or tilted with respect to this axis, which is usually not desired. The radially symmetric standard model used in 3DE4 is based on the distortion model developed for astronomy by Brown [3] and Conrady [4]. The original model (up to order four) is given by the mapping

$$
\begin{aligned}
x' &= x(1 + c_2 r^2 + c_4 r^4) + \left[t_1(r^2 + 2x^2) + 2t_2 xy\right]\ (1 + t_3 r^2) \\
y' &= y(1 + c_2 r^2 + c_4 r^4) + \left[t_2(r^2 + 2x^2) + 2t_1 xy\right]\ (1 + t_3 r^2).
\end{aligned} \tag{32}
$$

with five parameters c_2, c_4, t_1, t_2, t_3. This model takes into account that lenses might be decentered. As you see this model is not linear in its parameters, since it contains products of t_1, t_2 with t_3. For 3DE4 we have modified this model by introducing an additional parameter. Our model has six parameters c_2, c_4, u_2, v_2, u_4, v_4 which are related to the original parameters by

$$
\begin{aligned}
u_2 &= t_1 & \qquad u_4 &= t_3 t_1 \\
v_2 &= t_2 & \qquad v_4 &= t_3 t_2.
\end{aligned} \tag{33}
$$

From a physics point of view, the additional parameter is redundant, yet we found it helpful to have linear dependency for estimating coefficients e.g. from grid shots. The result is the following model (Poly-4-Radial-Decenter):

$$
\boxed{
\begin{aligned}
x' &= x(1 + c_2 r^2 + c_4 r^4) + (r^2 + 2x^2)(u_2 + u_4 r^2) + 2xy(v_2 + v_4 r^2) \\
y' &= y(1 + c_2 r^2 + c_4 r^4) + (r^2 + 2y^2)(v_2 + v_4 r^2) + 2xy(u_2 + u_4 r^2).
\end{aligned} \quad (34)
}
$$

which we will write as

$$
(x', y') = g_{\mathrm{rad,dec}}(x, y). \tag{35}
$$

This is a sub-model of *3DE4 Radial - Standard, Degree 4* used in 3DE4. The parameter names are summarized in Table 2.

3.5.1 Extender: Beam Splitter

In 3DE4 the radially symmetric model is also used for stereo rigs. If the two cameras in a stereo rig are placed side by side either camera can be modelled separately using the plain radial model defined in the previous section. In practice,

Table 2: *Parameters of model* Poly-4-Radial-Decenter

Doc	Code	GUI	Default
c_2	c2	*Distortion - Degree 2*	0
u_2	u2	*U - Degree 2*	0
v_2	v2	*V - Degree 2*	0
c_4	c4	*Quartic Distortion - Degree 4*	0
u_4	u4	*U - Degree 4*	0
v_4	v4	*V - Degree 4*	0

sometimes the interocular distance of the cameras is too small for arranging them this way. In this case the two cameras are arranged by using a beam splitter (see Figures 3 and 4).

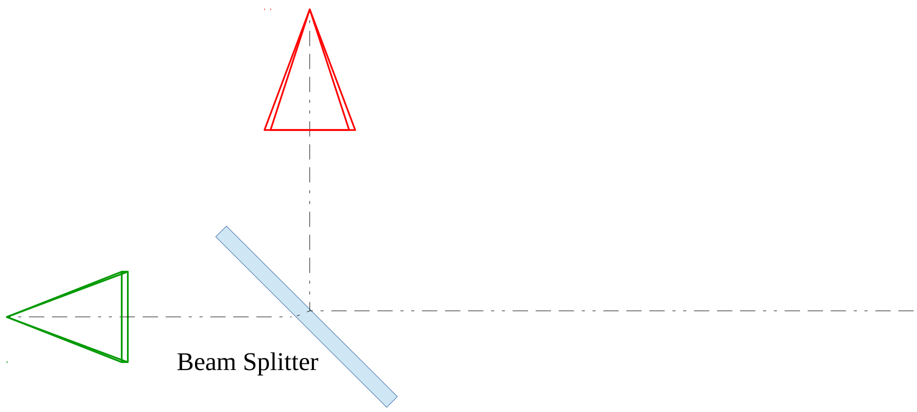

Figure 3: *Stereo rig with beam splitter, side view*

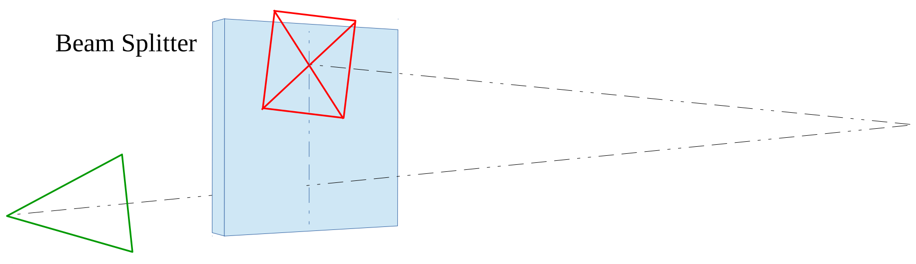

Figure 4: *Stereo rig with beam splitter, top view*

Historically, there were grid shots for calibrating stereo cameras which could not be handled by the plain radial model. This problem lead us to implementing an extender in order to compensate for possible beam splitter artefacts. The question is: If the beam splitter suffers from some kind of deformation for instance due to exposure to heat on set, what mathematical function will describe this deformation best?

Imagine a flat disc made, exposed to heat on one side, but not on the other side. The material will expand on the exposed side more than on the non-exposed side, which leads to mechanical stress. The disc will bend to compensate for this stress but in a way as to keep overall deformation energy minimal. This lead us to the following extender function. We introduce two parameters named ϕ_{bs} and b_{bs}. Using the short hand notations

$$\begin{aligned} q &= \sqrt{1 + b_{\mathrm{bs}}} \\ c &= \cos\phi_{\mathrm{bs}} \\ s &= \sin\phi_{\mathrm{bs}} \end{aligned} \tag{36}$$

the extender matrix reads

$$H_{\phi_{\mathrm{bs}},b_{\mathrm{bs}}} = \begin{bmatrix} c^2 q + \frac{s^2}{q} & (q - \frac{1}{q})cs \\ (q - \frac{1}{q})cs & \frac{c^2}{q} + s^2 q \end{bmatrix} \tag{37}$$

We will write this matrix as function

$$h_{\phi_{\mathrm{bs}},b_{\mathrm{bs}}}(x, y). \tag{38}$$

It is easy to check that the eigenvectors of this matrix are $[c, s]$ and $[-s, c]$ with eigenvalues q and $1/q$, respectively. Geometrically this means that a circle is deformed under $H_{\phi_{\mathrm{bs}},b_{\mathrm{bs}}}$ into an ellipsis of same area with axes $[c, s]$ and $[-s, c]$. The extender matrix remains invariant under the mapping

$$\begin{aligned} q &\mapsto \frac{1}{q} \\ \phi_{\mathrm{bs}} &\mapsto \phi_{\mathrm{bs}} + \frac{\pi}{2} \end{aligned} \tag{39}$$

which is approximately equivalent (for $|b_{\mathrm{bs}}| \ll 1$) to

$$\begin{aligned} b_{\mathrm{bs}} &\mapsto -b_{\mathrm{bs}} \\ \phi_{\mathrm{bs}} &\mapsto \phi_{\mathrm{bs}} + 90^\circ \end{aligned} \tag{40}$$

which you can easily verify in 3DE4. Please note that this matrix is not purely based on physics input. We have "symmetrized" it by introducing a factor

$$\sqrt{1 + b_{\mathrm{bs}}} \tag{41}$$

for practical reasons. Without this factor we would have a symmetry which involves focal length. We omit the details here, but in the end we decided to decouple the symmetry of $\phi_{\mathrm{bs}}, b_{\mathrm{bs}}$ from focal length.

Finally, we should clarify how the extender is applied. In a stereo rig with beam splitter light ray directions are first distorted by the beam splitter and second by the lens system. Our model function has to remove distortion effects in the opposite direction. If we write this down, we finally get our distortion model Poly-4-Radial-Decenter-Elliptic

$$(x', y') = h_{\phi_{\mathrm{bs}}, b_{\mathrm{bs}}} \circ g_{\mathrm{rad,dec}}(x, y) \tag{42}$$

where $g_{\mathrm{rad,dec}}$ is the model function of model Poly-4-Radial-Decenter as defined in (34). This is the eight-parameter model *3DE4 Radial - Standard, Degree 4* used in 3DE4. The parameter names added by the extender are given in Table 3.

Table 3: *Additional parameters of model* Poly-4-Radial-Decenter-Elliptic

Doc	Code	GUI	Default
ϕ_{bs}	`phi_bs`	*Phi - Cylindric Direction*	0°
b_{bs}	`b_bs`	*B - Cylindric Bending*	0

Although this model is no longer linear in its coefficients it is still a polynomial model. It does not make much sense to expand the composite expression into powers of x and y, so we shall leave it in its compact form given above.

3.6 The polynomial approach for anamorphic lenses

In order to model the distortion of an anamorphic lens system, we can use a polynomial approach as we did for the radially symmetric lens. Let g_c be the model function for a perfect anamorphic lens system. We examine the correctional part of this model function:

$$d(x, y) = g_c(x, y) - (x, y) \tag{43}$$

Then for the components of $d(x, y)$ we have the following symmetries:

$$\begin{aligned} d_{\mathrm{x}}(x, y) &= -d_{\mathrm{x}}(-x, y) \\ d_{\mathrm{y}}(x, y) &= d_{\mathrm{y}}(-x, y) \\ d_{\mathrm{x}}(x, y) &= d_{\mathrm{x}}(x, -y) \\ d_{\mathrm{y}}(x, y) &= -d_{\mathrm{y}}(x, -y) \end{aligned} \tag{44}$$

So, the x-component of the correctional part is an odd function in x but even in y, while the y-component is an even function in x but odd in y. Our polynomial model must reflect this symmetry, but it must also be formulated as broadly as possible. These requirements lead to a sum of all bi-variate monomials with matching parity. Up to order six of the correctional term, this polynomial reads

$$
\begin{aligned}
x' = x(\quad & 1 && + a_{02}y^2 && + a_{04}y^4 && + a_{06}y^6 \\
& + a_{20}x^2 && + a_{22}x^2y^2 && + a_{24}x^2y^4 && \\
& + a_{40}x^4 && + a_{42}x^4y^2 && && \\
& + a_{60}x^6) && && &&
\end{aligned}
\tag{45}
$$

and

$$
\begin{aligned}
y' = y(\quad & 1 && + b_{02}y^2 && + b_{04}y^4 && + b_{06}y^6 \\
& + b_{20}x^2 && + b_{22}x^2y^2 && + b_{24}x^2y^4 && \\
& + b_{40}x^4 && + b_{42}x^4y^2 && && \\
& + b_{60}x^6) && && &&
\end{aligned}
\tag{46}
$$

where a_{ij} is the coefficient for monomial xx^iy^j for the x-component and b_{ij} for monomial yx^iy^j for the y-component. More generally the polynomial up to order n reads:

$$
\begin{aligned}
x' &= x \sum_{\substack{i=0 \\ i \text{ even}}}^{n} \sum_{\substack{j=0 \\ j \text{ even}}}^{n-i} a_{ij}x^iy^j \\
y' &= y \sum_{\substack{i=0 \\ i \text{ even}}}^{n} \sum_{\substack{j=0 \\ j \text{ even}}}^{n-i} b_{ij}x^iy^j
\end{aligned}
\tag{47}
$$

with $a_{00} = b_{00} = 1$. In practice we re-formulated this in polar coordinates (based on dn-coordinates):

$$
\begin{aligned}
r &= \sqrt{x^2 + y^2} \\
\phi &= \operatorname{atan}(y, x)
\end{aligned}
\tag{48}
$$

where atan stands for the arctan inverse trigonometic extended to all four quadrants so that $\cos\phi = x$ and $\sin\phi = y$. Using addition theorems for trigonometric functions the polynomials (47) can be re-formulated as

$$
\begin{aligned}
x' &= x \sum_{\substack{i=0 \\ i \text{ even}}}^{n} \sum_{\substack{j=i \\ j \text{ even}}}^{n} c_{ij}^{(\mathrm{x})} r^j \cos(i\phi) \\
y' &= y \sum_{\substack{i=0 \\ i \text{ even}}}^{n} \sum_{\substack{j=i \\ j \text{ even}}}^{n} c_{ij}^{(\mathrm{y})} r^j \cos(i\phi)
\end{aligned}
\tag{49}
$$

with $c^{(\mathrm{x})}_{00} = c^{(\mathrm{y})}_{00} = 1$. The explicit relationship between a, b and $c^{(\mathrm{x})}$, $c^{(\mathrm{y})}$ is not important here, we just need to trust (or prove) that both (47) and (49) are equivalent in what they are doing. For degree up to six we have established this relationship explicitly. Note, that these relations are symmetric with respect to exchanging $a \leftrightarrow b$ and $c^{(\mathrm{x})} \leftrightarrow c^{(\mathrm{y})}$.

$$
\begin{aligned}
a_{20} &= c^{(\mathrm{x})}_{02} + c^{(\mathrm{x})}_{22} & b_{20} &= c^{(\mathrm{y})}_{02} + c^{(\mathrm{y})}_{22} \\
a_{02} &= c^{(\mathrm{x})}_{02} - c^{(\mathrm{x})}_{22} & b_{02} &= c^{(\mathrm{y})}_{02} - c^{(\mathrm{y})}_{22} \\
a_{40} &= c^{(\mathrm{x})}_{04} + c^{(\mathrm{x})}_{24} + c^{(\mathrm{x})}_{44} & b_{40} &= c^{(\mathrm{y})}_{04} + c^{(\mathrm{y})}_{24} + c^{(\mathrm{y})}_{44} \\
a_{22} &= 2c^{(\mathrm{x})}_{04} - 6c^{(\mathrm{x})}_{44} & b_{22} &= 2c^{(\mathrm{y})}_{04} - 6c^{(\mathrm{y})}_{44} \\
a_{04} &= c^{(\mathrm{x})}_{04} - c^{(\mathrm{x})}_{24} + c^{(\mathrm{x})}_{44} & b_{04} &= c^{(\mathrm{y})}_{04} - c^{(\mathrm{y})}_{24} + c^{(\mathrm{y})}_{44} \\
a_{60} &= c^{(\mathrm{x})}_{06} + c^{(\mathrm{x})}_{26} + c^{(\mathrm{x})}_{46} + c^{(\mathrm{x})}_{66} & b_{60} &= c^{(\mathrm{y})}_{06} + c^{(\mathrm{y})}_{26} + c^{(\mathrm{y})}_{46} + c^{(\mathrm{y})}_{66} \\
a_{42} &= 3c^{(\mathrm{x})}_{06} + c^{(\mathrm{x})}_{26} - 5c^{(\mathrm{x})}_{46} - 15c^{(\mathrm{x})}_{66} & b_{42} &= 3c^{(\mathrm{y})}_{06} + c^{(\mathrm{y})}_{26} - 5c^{(\mathrm{y})}_{46} - 15c^{(\mathrm{y})}_{66} \\
a_{24} &= 3c^{(\mathrm{x})}_{06} - c^{(\mathrm{x})}_{26} - 5c^{(\mathrm{x})}_{46} + 15c^{(\mathrm{x})}_{66} & b_{24} &= 3c^{(\mathrm{y})}_{06} - c^{(\mathrm{y})}_{26} - 5c^{(\mathrm{y})}_{46} + 15c^{(\mathrm{y})}_{66} \\
a_{06} &= c^{(\mathrm{x})}_{06} - c^{(\mathrm{x})}_{26} + c^{(\mathrm{x})}_{46} - c^{(\mathrm{x})}_{66} & b_{06} &= c^{(\mathrm{y})}_{06} - c^{(\mathrm{y})}_{26} + c^{(\mathrm{y})}_{46} - c^{(\mathrm{y})}_{66}
\end{aligned} \tag{50}
$$

We will need these relations later when we formulate the JACOBI-matrix of the model function.

3.6.1 Extender: Anamorphic compression / Pixel aspect

When images from an anamorphic camera are imported into 3DE4, their pixel aspect reflects the anamorphic squeeze introduced by the anamorphic lens. In order to compensate for distortion, pixel aspect must be estimated precisely by 3DE4 parameter adjustment procedures. In this sense, it is a distortion parameter similar to the parameters $c^{(\mathrm{x})}$ and $c^{(\mathrm{y})}$ in the polynomial model. One method of incorporating the anamorphic compression into the model function would be to allow values different from 1 for parameter $c^{(\mathrm{x})}_{00}$, while leaving $c^{(\mathrm{y})}_{00}$ at 1. Yet there are quite a few arguments against this method.

- In compositing, a common workflow is to undistort the footage, compose it with rendered content and then re-distort it. As far as we understand it, anamorphic distortion -but not the squeeze- is often a means of artistic expression in movie-making. If the linear compression (of around 2.0) was part of the model function, the rendered content would be squeezed when the composed images are re-distorted, which is not what you want.
- Let us assume, pixel aspect as well as distortion data are encoded in the image files coming from the camera. Even if software for rendering, compositing or any kind of editing does not have the means for dealing with lens distortion, it is still able to display images more or less correctly, as long as it has access to pixel aspect encoded in the file.

Given this argumentation we could completely separate pixel aspect from our distortion model, however, experience shows that we cannot do this. One reason is that the model has to remain usable, even if the image sequence is subject to lens breathing as a result of focus pull or zoom. For this reason we define the following extenders, which allow squeeze in x- and y-direction:

$$
\begin{aligned}
S_q^{(\mathrm{x})} &= \begin{bmatrix} q & 0 \\ 0 & 1 \end{bmatrix} \\
S_q^{(\mathrm{y})} &= \begin{bmatrix} 1 & 0 \\ 0 & q \end{bmatrix}
\end{aligned} \tag{51}
$$

In order to write the extenders as functions we define

$$
\begin{aligned}
\mathrm{sqx}_q &: (x, y) \mapsto (qx, y) \\
\mathrm{sqy}_q &: (x, y) \mapsto (x, qy)
\end{aligned} \tag{52}
$$

In practice we use these extenders for modelling compression effects of the anamorphic lens. We split this compression in two parts:

- A large **time-independent** squeeze in x-direction, which we describe as pixel aspect. Values are often around 2.0, but other squeeze ratios are known as well.
- A small, often **time-dependent** deformation which acts independently in x- and in y-direction. More precisely, there is a dependency, but it is highly non-trivial. "small" means, we describe this deformation by squeezing parameters which are close to 1. By these parameters we model effects like lens breathing.

Putting this back together again, we expect that the anamorphic model function contains an expression:

$$
\mathrm{sqx}_{s_\mathrm{x}} \circ \mathrm{sqy}_{s_\mathrm{y}} \circ \mathrm{sqx}_{r_\mathrm{pa}} \tag{53}
$$

or alternatively, since all these operations commute:

$$
\mathrm{sqy}_{s_\mathrm{y}} \circ \mathrm{sqx}_{s_\mathrm{x} r_\mathrm{pa}}, \tag{54}
$$

with two new squeeze parameters s_x and s_y, both time-dependent and near 1.

3.7 The standard model for anamorphic lenses

The standard model for anamorphic lenses in 3DE4 is based on (49) with correction terms up to order four. We shall denote the polynomial by g_anam in this section (also Poly-4-Anamorphic). The pure distortional part (with g_anam split into components $g_{\mathrm{anam,x}}$ and $g_{\mathrm{anam,y}}$) reads

$$\begin{aligned}
g_{\mathrm{anam,x}}(x,y) = x(1 \quad & + c_{02}^{(\mathrm{x})} r^2 && + c_{04}^{(\mathrm{x})} r^4 \\
& + c_{22}^{(\mathrm{x})} r^2 \cos 2\phi && + c_{24}^{(\mathrm{x})} r^4 \cos 2\phi \\
& && + c_{44}^{(\mathrm{x})} r^4 \cos 4\phi) \\
g_{\mathrm{anam,y}}(x,y) = y(1 \quad & + c_{02}^{(\mathrm{y})} r^2 && + c_{04}^{(\mathrm{y})} r^4 \\
& + c_{22}^{(\mathrm{y})} r^2 \cos 2\phi && + c_{24}^{(\mathrm{y})} r^4 \cos 2\phi \\
& && + c_{44}^{(\mathrm{y})} r^4 \cos 4\phi)
\end{aligned} \tag{55}$$

Additionally we have a squeeze-x extender and a squeeze-y extender with values s_{x} and s_{y} for dealing with lens breathing. As mentioned, this construction allows us to consider pixel aspect constant over time and translocate time-dependent lens breathing effects into $\mathrm{sqx}_{s_{\mathrm{x}}}$ and $\mathrm{sqy}_{s_{\mathrm{y}}}$ which in practice will be close to identity.

Finally we get our model function for Poly-4-Anamorphic-Rpa-Sq:

$$(x', y') = \mathrm{sqy}_{s_{\mathrm{y}}} \circ \mathrm{sqx}_{s_{\mathrm{x}} r_{\mathrm{pa}}} \circ g_{\mathrm{anam}} \circ \mathrm{sqx}_{r_{\mathrm{pa}}}^{-1}(x, y) \tag{56}$$

The model has twelve parameters which relate to 3DE4's GUI representation as shown in Table 4.

Please note, that for the default parameter set $(0, \ldots, 0, s_{\mathrm{x}} = 1, s_{\mathrm{y}} = 1)$ the model function is the identity regardless of the value for r_{pa}.

We have been nudged by users to provide the parameter s_{y} for the following reason: In sequences with constant focal length and lens breathing due to focus pull, users prefer to consider focal length as static while lens breathing is modelled by the time-dependent distortion model parameters s_{x} and s_{y}.

In order to motivate this form we should describe in words what is happening here. As mentioned, our model function maps dn-coordinates onto dn-coordinates. In principle, dn-coordinates are nothing else but a uniformly scaled version of virtual filmback coordinates. The rightmost term

$$\mathrm{sqx}_{r_{\mathrm{pa}}}^{-1} \tag{57}$$

transforms (x, y) into (diagonally normalized) physical filmback coordinates, i.e. we can imagine $\mathrm{sqx}_{r_{\mathrm{pa}}}^{-1}(x, y)$ as a point on the camera sensor. The left hand side

$$\mathrm{sqy}_{s_{\mathrm{y}}} \circ \mathrm{sqx}_{s_{\mathrm{x}} r_{\mathrm{pa}}} \circ g_{\mathrm{anam}} \tag{58}$$

describes the effect of the anamorphic lens, including the large squeeze (e.g. by 2.0). We shall see in section 3.7.1 that it is important to clearly separate the effect of the anamorphic lens from the squeezing transform on the filmback.

Yet, we have to motivate the left hand side (58) of our model function (56). It is not a-priori clear, that $\mathrm{sqx}_{s_{\mathrm{x}}}$ and $\mathrm{sqy}_{s_{\mathrm{y}}}$ have to be applied after g_{anam}. Without having investigated this in detail we think that applying them before (i.e. on the right hand side of) g_{anam} would lead to a different, but equivalent lens distortion model. The reason for us to place the squeezes on the left hand side (i.e. on the "undistorted side") is the following: There is an ambiguity between parameters s_{x}, s_{y} and focal length f. The transformation

$$\begin{aligned} s_{\mathrm{y}} &\mapsto q s_{\mathrm{y}} \\ s_{\mathrm{x}} &\mapsto q s_{\mathrm{x}} \\ f &\mapsto q f \end{aligned} \tag{59}$$

for positive q will leave the camera model invariant. This ambiguity would also be present if the squeezes were applied on the right hand side, but in this case mapping all three parameters due to this ambiguity would only work along with some complicated transformation of the coefficients in g_{anam}, which is impossible to handle for the user.

Once we have decided to apply $\mathrm{sqx}_{s_{\mathrm{x}}}$ on the left hand side, we have to do the same with pixel aspect ratio since we would like to consider s_{x}, s_{y} and r_{pa} as the dynamic and the static part of the overall compression of the anamorphic lens, in order to model lens breathing. The bottom line is:

- We clearly have to identify in our model which part corresponds to the anamorphic lens and which part is just rescaling on the filmback side.
- The squeezes $\mathrm{sqx}_{s_{\mathrm{x}}}$ and $\mathrm{sqy}_{s_{\mathrm{y}}}$ must be applied on the left hand side of g_{anam} in order to avoid weird ambiguities.
- Pixel aspect must be applied along with $\mathrm{sqx}_{s_{\mathrm{x}}}$ so that we can split the entire dynamic compression into a dynamic part near 1.0 and a static part encoded in r_{pa}.

The model does not compensate for lens decentering, and up to now this has not been requested by users. An extension for decentering as for the radial model would of course be possible, but it is not clear what kind of parameters would need to be added.

3.7.1 Extender: Rotation

Historically, when we implemented the model Poly-4-Anamorphic-Rpa-Sq and even another model Poly-6-Anamorphic/*3DE4 Anamorphic, Degree 6*, it turned out that there still was a considerable amount of gridshots we could not handle with any of these models. We then got reports from users that they were using camera bodies, the anamorphic lens was attached to and aligned "by hand", without e.g. bayonet lock. Clearly, our models Poly-4-Anamorphic-Rpa-Sq and Poly-6-Anamorphic would fail since the symmetry assumptions (44) the models are based on are not fulfilled.

Table 4: *The parameters of model* Poly-4-Anamorphic-Rpa-Sq

Doc	Code	GUI	Default
$c_{02}^{(x)}$	`cx02`	*Cx02 - Degree 2*	0
$c_{02}^{(y)}$	`cy02`	*Cy02 - Degree 2*	0
$c_{22}^{(x)}$	`cx22`	*Cx22 - Degree 2*	0
$c_{22}^{(y)}$	`cy22`	*Cy22 - Degree 2*	0
$c_{04}^{(x)}$	`cx04`	*Cx04 - Degree 4*	0
$c_{04}^{(y)}$	`cy04`	*Cy04 - Degree 4*	0
$c_{24}^{(x)}$	`cx24`	*Cx24 - Degree 4*	0
$c_{24}^{(y)}$	`cy24`	*Cy24 - Degree 4*	0
$c_{44}^{(x)}$	`cx44`	*Cx44 - Degree 4*	0
$c_{44}^{(y)}$	`cy44`	*Cy44 - Degree 4*	0
s_x	`sx`	*Squeeze-X*	1
s_y	`sy`	*Squeeze-Y*	1

In order to address this problem we have introduced an extender which rotates the entire anamorphic lens around the optical axis. The matrix for this extender is a simple rotation by an angle ϕ_mnt (where the index stands for "mount" or "mounted"):

$$R_{\phi_\mathrm{mnt}} = \begin{bmatrix} \cos\phi_\mathrm{mnt} & -\sin\phi_\mathrm{mnt} \\ \sin\phi_\mathrm{mnt} & \cos\phi_\mathrm{mnt} \end{bmatrix} \tag{60}$$

We shall write this as a function $\mathrm{rot}_{\phi_\mathrm{mnt}}$. Since in the previous section we split our model function (56) into a lens term and the filmback squeeze term, we can now easily model the effect of a rotated lens and get our final model Poly-4-Anamorphic-Rpa-Sq-Rot which corresponds to model *3DE4 Anamorphic - Standard, Degree 4* in 3DE4:

$$(x', y') = \mathrm{rot}_{\phi_\mathrm{mnt}} \circ \mathrm{sqy}_{s_\mathrm{y}} \circ \mathrm{sqx}_{s_\mathrm{x} r_\mathrm{pa}} \circ g_\mathrm{anam} \circ \mathrm{rot}^{-1}_{\phi_\mathrm{mnt}} \circ \mathrm{sqx}^{-1}_{r_\mathrm{pa}}(x, y) \tag{61}$$

Table 5 shows the additional parameter due to the rotation extender. In practice, in case camera bodies and lenses with manual mount are used, we have found values ranging from $-2°$ to $+2°$ in support projects from users.

Table 5: *Additional parameter of model* Poly-4-Anamorphic-Rpa-Sq-Rot

Doc	Code	GUI	Default
ϕ_mnt	`phi_mnt`	*Lens Rotation*	$0°$

3.7.2 Extender: Rescaling

In this section we will extend the model Poly-4-Anamorphic-Rpa-Sq-Rot by an extender which again is based on user input. The drawback of working with anamorphic footage is clearly that not all software allows to specify pixel aspect for an uncompressed representation of the images.

For this reason it has become fashion to unsqueeze the footage by some well-determined factor, usually pixel aspect ratio. We shall call images handled this way “rescaled” images. Yet, in presence of non-zero lens rotation as modelled in the previous section, pixel aspect ratio is no longer a well-defined number[2].

In our standard model Poly-4-Anamorphic-Rpa-Sq-Rot we used pixel aspect (along with s_x and s_y) in order to describe the compression effect of the anamorphic lens, but this method fails for rescaled images: the anamorphic lens does not squeeze along the x-axis because of lens rotation, yet the user-rescaling is done exactly along the x-axis.

In practice, the user sets pixel aspect ratio to 1, due to the rescaling, which means that our model (61) will fail. Therefore we introduce an extender $\mathrm{sqx}_{s_\mathrm{rscl}}$ and add it to our lens rotation model. In the following expression we have merged all squeeze-x terms into a single function. The result is the model Poly-4-Anamorphic-Rpa-Re-Sq-Rot which in 3DE4's GUI is called *3DE4 Anamorphic - Rescaled, Degree 4*:

$$(x', y') = \mathrm{rot}_{\phi_\mathrm{mnt}} \circ \mathrm{sqy}_{s_\mathrm{y}} \circ \mathrm{sqx}_{s_\mathrm{x} r_\mathrm{pa} s_\mathrm{rscl}} \circ g_\mathrm{anam} \circ \mathrm{rot}^{-1}_{\phi_\mathrm{mnt}} \circ \mathrm{sqx}^{-1}_{r_\mathrm{pa} s_\mathrm{rscl}}(x, y) \tag{62}$$

Compared to the rotated lens model we have one additional parameter s_rscl as in Table 6.

Table 6: *Additional parameter of model* Poly-4-Anamorphic-Rpa-Re-Sq-Rot

Doc	Code	GUI	Default
s_rscl	`s_rscl`	*Rescale*	1

4 Jacobian

In this section we present the Jacobi-matrices of our lens distortion models. We use greek indices for addressing components in two dimensions. These indices can assume symbolic values 'x' and 'y' for x- and y-component. Summing over double indices is implied (sum convention for covariant expressions). Given a

[2] We have not investigated this in detail, but we assume that pixel aspect can be described by a symmetric matrix in case of lens rotation.

bi-variate mapping $g(p)$, $p = (x, y)$, from $\mathbb{R}^2$ to $\mathbb{R}^2$, the JACOBI-matrix is defined as

$$J_{\mu\nu}(p) = \frac{\partial}{\partial p_\nu} g_\mu(p). \tag{63}$$

Since the lens distortion models are equipped with extenders, we should work out, how the JACOBIAN is altered by these extenders. Let us consider a model function like

$$g = a \circ \hat{g} \circ b \tag{64}$$

where a and b are linear functions. As mentioned before, the extenders can be represented as matrices A and B. We consider a model function, composed from a function $\hat{g}$ with JACOBIAN $\hat{J}$.

$$g_\mu(p) = A_{\mu\sigma} \hat{g}_\sigma(Bp) \tag{65}$$

We are interested in

$$J_{\mu\nu}(p) = \frac{\partial}{\partial p_\nu} A_{\mu\sigma} \hat{g}_\sigma(Bp). \tag{66}$$

We extract the constant matrix and apply the chain rule:

$$\begin{aligned} J_{\mu\nu}(p) &= A_{\mu\sigma} \frac{\partial}{\partial p_\nu} \hat{g}_\sigma(Bp) \\ &= A_{\mu\sigma} \left. \frac{\partial}{\partial q_\tau} \right|_{q=Bp} \hat{g}_\sigma(q) B_{\tau\nu} \\ &= A_{\mu\sigma} \hat{J}_{\sigma\tau}(Bp) B_{\tau\nu} \end{aligned} \tag{67}$$

Or simply

$$J(p) = A \hat{J}(Bp) B \tag{68}$$

Hence, for our composite model functions it is sufficient to consider the non-linear part. In the following sections we shall denote the JACOBIAN of the non-linear component by $\hat{J}(x, y)$ and the resulting JACOBIAN including extenders by $J(x, y)$.

4.1 The radial model

The JACOBIAN for the radial model is easily obtained by deriving (34) with respect to x and y:

$$\begin{aligned}
\hat{J}_{\mathrm{xx}}(x,y) &= 1 + c_2(y^2+3x^2) + c_4(y^2+5x^2)r^2 \\
&\quad + 6u_2x + u_4(8xy^2+12x^3) + 2v_2y + v_4(2y^3+6x^2y) \\
\hat{J}_{\mathrm{xy}}(x,y) &= 2c_2xy + 4c_4xyr^2 \\
&\quad + 2u_2y + u_4(8x^2y+4y^3) + 2v_2x + v_4(2x^3+6xy^2) \\
\hat{J}_{\mathrm{yx}}(x,y) &= 2c_2xy + 4c_4xyr^2 \\
&\quad + 2u_2y + u_4(6x^2y+2y^3) + 2v_2x + v_4(4x^3+8xy^2) \\
\hat{J}_{\mathrm{yy}}(x,y) &= 1 + c_2(x^2+3y^2) + c_4(x^2+5y^2)r^2 \\
&\quad + 6v_2y + v_4(8x^2y+12y^3) + 2u_2x + u_4(2x^3+6xy^2).
\end{aligned} \tag{69}$$

With beam splitter extender (37) we have:

$$J(x,y) = H_{\phi,b}\hat{J}(x,y) \tag{70}$$

4.2 The anamorphic model

The non-linear part of the anamorphic standard model is given by the model function Poly-4-Anamorphic in (55). Starting from the CARTESIAN representation (47) we derive with respect to x and y. The result is:

$$\begin{aligned}
\hat{J}_{\mathrm{xx}}(x,y) &= 1 + 3a_{20}x^2 + a_{02}y^2 + 5a_{40}x^4 + 3a_{22}x^2y^2 + a_{04}y^4 \\
\hat{J}_{\mathrm{xy}}(x,y) &= 2a_{02}xy + 4a_{04}xy^3 + 2a_{22}x^3y \\
\hat{J}_{\mathrm{yx}}(x,y) &= 2b_{20}xy + 4b_{40}x^3y + 2b_{22}xy^3 \\
\hat{J}_{\mathrm{yy}}(x,y) &= 1 + 3b_{02}y^2 + b_{20}x^2 + 5b_{04}y^4 + 3b_{22}x^2y^2 + b_{40}x^4
\end{aligned} \tag{71}$$

We have already established the relationship between a_{ij}, b_{ij} and $c^{(\mathrm{x})}$, $c^{(\mathrm{y})}$ in (50) so this is the final result for the non-linear part. The JACOBIAN for the anamorphic model Poly-4-Anamorphic-Rpa-Sq without rotation reads:

$$J(x,y) = S^{(\mathrm{y})}_{s_{\mathrm{y}}} S^{(\mathrm{x})}_{s_{\mathrm{x}}r_{\mathrm{pa}}} \hat{J}(x,y) S^{(\mathrm{x})\,-1}_{r_{\mathrm{pa}}} \tag{72}$$

For the anamorphic model Poly-4-Anamorphic-Rpa-Sq-Rot as used in 3DE4 we get:

$$J(x,y) = R_{\phi_{\mathrm{mnt}}} S^{(\mathrm{y})}_{s_{\mathrm{y}}} S^{(\mathrm{x})}_{s_{\mathrm{x}}r_{\mathrm{pa}}} \hat{J}(x,y) R^{-1}_{\phi_{\mathrm{mnt}}} S^{(\mathrm{x})\,-1}_{r_{\mathrm{pa}}} \tag{73}$$

With additional rescaling due to Poly-4-Anamorphic-Rpa-Re-Sq-Rot we get:

$$J(x,y) = R_{\phi_{\mathrm{mnt}}} S^{(\mathrm{y})}_{s_{\mathrm{y}}} S^{(\mathrm{x})}_{s_{\mathrm{x}}r_{\mathrm{pa}}s_{\mathrm{rscl}}} \hat{J}(x,y) R^{-1}_{\phi_{\mathrm{mnt}}} {S^{(\mathrm{x})}_{r_{\mathrm{pa}}s_{\mathrm{rscl}}}}^{-1} \tag{74}$$

5 Reparametrization

5.1 Subimage

By *reparametrization* we understand the following: Consider a filmback, a lens center offset and a model function g_c defined on this filmback. Now, for some reason, somewhere in the production pipeline a rectangular sub-area of this filmback is used instead of the original one. For instance, the original footage was recorded in open-gate size, but then for compositing and final rendering it is reduced to broadcast size.

The question is: how can we avoid re-estimating the distortion parameters for the altered filmback and lens center offset? How can we calculate the new distortion coefficients from the original ones? We shall investigate this in this section.

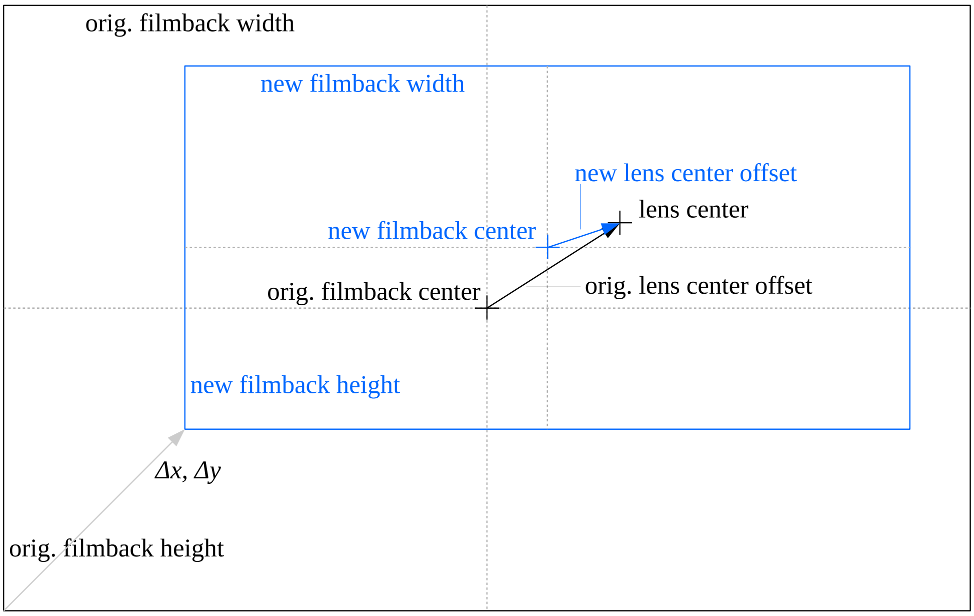

Figure 5: *New and original filmback*

First of all, let us recall our general expression for applying a model function g_c to unit coordinates (7)

$$(x'_{\text{unit}}, y'_{\text{unit}}) = \phi^{-1} \circ g_c \circ \phi(x_{\text{unit}}, y_{\text{unit}}) \tag{75}$$

where g_c is the model function with parameters c and ϕ maps unit- to dn-coordinates as in (15). We will now map all numbers and functions to new numbers and functions corresponding to the new filmback. The new filmback and lens center offset is (see blue graphics in Figure 5)

$$\tilde{w}_{\text{fb,cm}}, \tilde{h}_{\text{fb,cm}}, \tilde{x}_{\text{lco,cm}}, \tilde{y}_{\text{lco,cm}} \tag{76}$$

The radius for this filmback is written as $\tilde{r}_{\text{fb,cm}}$. We will have new unit coordinates and new diagonally normalized coordinates:

$$\tilde{x}_{\text{unit}}, \tilde{y}_{\text{unit}}, \tilde{x}_{\text{dn}}, \tilde{y}_{\text{dn}} \tag{77}$$

For the new filmback the undistortion mapping reads:

$$(\tilde{x}'_{\text{unit}}, \tilde{y}'_{\text{unit}}) = \tilde{\phi}^{-1} \circ g_{\tilde{c}} \circ \tilde{\phi}(\tilde{x}_{\text{unit}}, \tilde{y}_{\text{unit}}) \tag{78}$$

which means, we would like to use the same distortion model g but we will get new parameters $\tilde{c}$. Please note, that we now map unit to dn-coordinates by a function $\tilde{\phi}$ which relies on the new filmback data.

Our method is to replace all expressions in (75) by tilde expressions except for g_c. Then by comparing to (78) we will see how to obtain $g_{\tilde{c}}$ from g_c. From Figure 5 we can extract the following relations:

$$\begin{aligned}
\Delta x &= x_{\text{lco,cm}} - \tilde{x}_{\text{lco,cm}} + \frac{1}{2}(w_{\text{fb,cm}} - \tilde{w}_{\text{fb,cm}}) \\
\Delta y &= y_{\text{lco,cm}} - \tilde{y}_{\text{lco,cm}} + \frac{1}{2}(h_{\text{fb,cm}} - \tilde{h}_{\text{fb,cm}})
\end{aligned} \tag{79}$$

Using $\Delta x, \Delta y$ we can establish the relationship between $x_{\text{unit}}, y_{\text{unit}}$ and $\tilde{x}_{\text{unit}}, \tilde{y}_{\text{unit}}$.

$$\begin{aligned}
x_{\text{unit}} &= \frac{1}{w_{\text{fb,cm}}}(\Delta x + \tilde{x}_{\text{unit}} \tilde{w}_{\text{fb,cm}}) \\
y_{\text{unit}} &= \frac{1}{h_{\text{fb,cm}}}(\Delta y + \tilde{y}_{\text{unit}} \tilde{h}_{\text{fb,cm}})
\end{aligned} \tag{80}$$

If we express $\phi(x_{\text{unit}}, y_{\text{unit}})$ by means of these relations (do this as exercise), we get:

$$\begin{aligned}
\phi_{\text{x}}(x_{\text{unit}}, y_{\text{unit}}) &= \frac{\tilde{x}_{\text{unit}} \tilde{w}_{\text{fb,cm}} - \tilde{x}_{\text{lco,cm}}}{r_{\text{fb,cm}}} - \frac{\tilde{w}_{\text{fb,cm}}}{2 r_{\text{fb,cm}}} \\
\phi_{\text{y}}(x_{\text{unit}}, y_{\text{unit}}) &= \frac{\tilde{y}_{\text{unit}} \tilde{h}_{\text{fb,cm}} - \tilde{y}_{\text{lco,cm}}}{r_{\text{fb,cm}}} - \frac{\tilde{h}_{\text{fb,cm}}}{2 r_{\text{fb,cm}}}
\end{aligned} \tag{81}$$

which means (this is important):

$$\boxed{\phi(x_{\text{unit}}, y_{\text{unit}}) = \rho \tilde{\phi}(\tilde{x}_{\text{unit}}, \tilde{y}_{\text{unit}}),} \tag{82}$$

where we have defined:

$$\rho = \frac{\tilde{r}_{\text{fb,cm}}}{r_{\text{fb,cm}}}. \tag{83}$$

Now let us transform equation (75). This step is easier if we apply ϕ to both sides:

$$\phi(x'_{\text{unit}}, y'_{\text{unit}}) = g_c \circ \phi(x_{\text{unit}}, y_{\text{unit}}) \tag{84}$$

We apply (82) and get

$$\rho\tilde{\phi}(\tilde{x}'_{\text{unit}}, \tilde{y}'_{\text{unit}}) = g_c(\rho\tilde{\phi}(\tilde{x}_{\text{unit}}, \tilde{y}_{\text{unit}})) \tag{85}$$

We move the factor ρ to the other side and apply $\tilde{\phi}^{-1}$ to both sides:

$$(\tilde{x}'_{\text{unit}}, \tilde{y}'_{\text{unit}}) = \tilde{\phi}^{-1}(\rho^{-1}g_c(\rho\tilde{\phi}(\tilde{x}_{\text{unit}}, \tilde{y}_{\text{unit}}))) \tag{86}$$

By comparing to (78) we now easily see that

$$\boxed{g_{\tilde{c}}(\tilde{x}_{\text{dn}}, \tilde{y}_{\text{dn}}) = \rho^{-1}g_c(\rho\tilde{x}_{\text{dn}}, \rho\tilde{y}_{\text{dn}})} \tag{87}$$

We can now establish the relationship between $\tilde{c}$ and c separately for each lens distortion model.

5.1.1 Extenders

Assume, a model function g_c has the form $a \circ \hat{g}_c \circ b$, where a and b are extenders. We have derived equation (87) for $\hat{g}_c$, now we show that it works for $a \circ \hat{g}_c \circ b$ as well:

$$\begin{aligned} g_{\tilde{c}}(\tilde{x}_{\text{dn}}, \tilde{y}_{\text{dn}}) &= a(\hat{g}_{\tilde{c}}(b(\tilde{x}_{\text{dn}}, \tilde{y}_{\text{dn}}))) \\ &= a(\rho^{-1}\hat{g}_c(\rho b(\tilde{x}_{\text{dn}}, \tilde{y}_{\text{dn}}))) \end{aligned} \tag{88}$$

Since a and b are linear functions, we can exchange the factors ρ^{-1} and ρ with a and b. Then we get:

$$\begin{aligned} g_{\tilde{c}}(\tilde{x}_{\text{dn}}, \tilde{y}_{\text{dn}}) &= \rho^{-1}a(\hat{g}_c(b(\rho\tilde{x}_{\text{dn}}, \rho\tilde{y}_{\text{dn}}))) \\ &= \rho^{-1}a \circ \hat{g}_c \circ b(\rho\tilde{x}_{\text{dn}}, \rho\tilde{y}_{\text{dn}}) \\ &= \rho^{-1}g_c(\rho\tilde{x}_{\text{dn}}, \rho\tilde{y}_{\text{dn}}). \end{aligned} \tag{89}$$

This means, extenders and their parameters remain unchanged under reparametrization. Only the non-linear function $\hat{g}_c$ is affected. This will simplify reparametrization considerably.

5.1.2 The radial model

As an exercise for warming up we reparametrize the simple model Poly-4-Radial

$$\begin{aligned} x' &= x(1 + c_2r^2 + c_4r^4) \\ y' &= y(1 + c_2r^2 + c_4r^4). \end{aligned} \tag{90}$$

Again, for the sake of simplicity, we omit the index "dn". All x and y are understood in dn-coordinates in this section. Equation (87) for this model is

$$\begin{aligned} \tilde{x}(1 + \tilde{c}_2\tilde{r}^2 + \tilde{c}_4\tilde{r}^4) &= \rho^{-1}\rho\tilde{x}(1 + c_2\rho^2\tilde{r}^2 + c_4\rho^4\tilde{r}^4) \\ \tilde{y}(1 + \tilde{c}_2\tilde{r}^2 + \tilde{c}_4\tilde{r}^4) &= \rho^{-1}\rho\tilde{y}(1 + c_2\rho^2\tilde{r}^2 + c_4\rho^4\tilde{r}^4) \end{aligned} \tag{91}$$

and hence by comparison of coefficients

$$\begin{aligned}\tilde{c}_2 &= c_2\rho^2\\ \tilde{c}_4 &= c_4\rho^4.\end{aligned}\tag{92}$$

For the decentered model Poly-4-Radial-Decenter we can examine the two decentering terms separately:

$$\begin{aligned}(\tilde{r}^2+2\tilde{x}^2)(\tilde{u}_2+\tilde{u}_4\tilde{r}^2) &= \rho^{-1}(\rho^2\tilde{r}^2+2\rho^2\tilde{x}^2)(u_2+u_4\rho^2\tilde{r}^2)\\ (\tilde{r}^2+2\tilde{y}^2)(\tilde{v}_2+\tilde{v}_4\tilde{r}^2) &= \rho^{-1}(\rho^2\tilde{r}^2+2\rho^2\tilde{y}^2)(v_2+v_4\rho^2\tilde{r}^2)\end{aligned}\tag{93}$$

and

$$\begin{aligned}2\tilde{x}\tilde{y}(\tilde{v}_2+\tilde{v}_4\tilde{r}^2) &= \rho^{-1}2\rho^2\tilde{x}\tilde{y}(v_2+v_4\rho^2\tilde{r}^2)\\ 2\tilde{x}\tilde{y}(\tilde{u}_2+\tilde{u}_4\tilde{r}^2) &= \rho^{-1}2\rho^2\tilde{x}\tilde{y}(u_2+u_4\rho^2\tilde{r}^2)\end{aligned}\tag{94}$$

which leads to the result:

$$\begin{aligned}\tilde{u}_2 &= u_2\rho & \tilde{v}_2 &= v_2\rho\\ \tilde{u}_4 &= u_4\rho^3 & \tilde{v}_4 &= v_4\rho^3\end{aligned}\tag{95}$$

In the previous section we have shown that extenders remain invariant under reparametrization. Hence, for our beam splitter model Poly-4-Radial-Decenter-Elliptic, the beam splitter parameters are unchanged:

$$\tilde{\phi}_{\text{bs}} = \phi_{\text{bs}} \qquad \tilde{b}_{\text{bs}} = b_{\text{bs}}\tag{96}$$

5.1.3 The anamorphic model

For the anamorphic models, reparametrization is similar as for the radial models. We apply our method to model Poly-4-Anamorphic. All other models are constructed by extenders to this model. The easiest way is to apply (87) to the general form for anamorphic lenses in polar coordinates (49):

$$\begin{aligned}\tilde{x}\sum_{\substack{i=0\\ i\,\text{even}}}^{n}\sum_{\substack{j=i\\ j\,\text{even}}}^{n}\tilde{c}_{ij}^{(\text{x})}\tilde{r}^j\cos(i\phi) &= \rho^{-1}\rho\tilde{x}\sum_{\substack{i=0\\ i\,\text{even}}}^{n}\sum_{\substack{j=i\\ j\,\text{even}}}^{n}c_{ij}^{(\text{x})}\rho^j\tilde{r}^j\cos(i\phi)\\ \tilde{y}\sum_{\substack{i=0\\ i\,\text{even}}}^{n}\sum_{\substack{j=i\\ j\,\text{even}}}^{n}\tilde{c}_{ij}^{(\text{y})}\tilde{r}^j\cos(i\phi) &= \rho^{-1}\rho\tilde{y}\sum_{\substack{i=0\\ i\,\text{even}}}^{n}\sum_{\substack{j=i\\ j\,\text{even}}}^{n}c_{ij}^{(\text{y})}\rho^j\tilde{r}^j\cos(i\phi)\end{aligned}\tag{97}$$

The angular factors are not affected, since $\phi = \operatorname{atan}(y,x) = \operatorname{atan}(\rho y,\rho x)$. The result is:

$$\begin{aligned}\tilde{c}_{ij}^{(x)} &= c_{ij}^{(\text{x})}\rho^j\\ \tilde{c}_{ij}^{(y)} &= c_{ij}^{(\text{y})}\rho^j\end{aligned}\tag{98}$$

or, for our case Poly-4-Anamorphic-Rpa-Sq:

$$\begin{aligned}
\tilde{c}^{(\mathrm{x})}_{02} &= c^{(\mathrm{x})}_{02}\rho^2 & \tilde{c}^{(\mathrm{x})}_{22} &= c^{(\mathrm{x})}_{22}\rho^2 & & \\
\tilde{c}^{(\mathrm{x})}_{04} &= c^{(\mathrm{x})}_{04}\rho^4 & \tilde{c}^{(\mathrm{x})}_{24} &= c^{(\mathrm{x})}_{24}\rho^4 & \tilde{c}^{(\mathrm{x})}_{44} &= c^{(\mathrm{x})}_{44}\rho^4 \\
\tilde{c}^{(\mathrm{y})}_{02} &= c^{(\mathrm{y})}_{02}\rho^2 & \tilde{c}^{(\mathrm{y})}_{22} &= c^{(\mathrm{y})}_{22}\rho^2 & & \\
\tilde{c}^{(\mathrm{y})}_{04} &= c^{(\mathrm{y})}_{04}\rho^4 & \tilde{c}^{(\mathrm{y})}_{24} &= c^{(\mathrm{y})}_{24}\rho^4 & \tilde{c}^{(\mathrm{y})}_{44} &= c^{(\mathrm{y})}_{44}\rho^4 \\
\tilde{s}_{\mathrm{x}} &= s_{\mathrm{x}} & \tilde{s}_{\mathrm{y}} &= s_{\mathrm{y}} & & \\
\tilde{\phi}_{\mathrm{mnt}} &= \phi_{\mathrm{mnt}} & & & &
\end{aligned} \tag{99}$$

The parameters s_{x}, s_{y} and ϕ_{mnt} are not affected, since they only occur in extenders.

5.2 Flipped image

Reparametrization is not only relevant for picking a subimage, as we did in the previous sections. In practice, at some point in the pipeline images might be flipped horizontally or vertically, and we would like to see how this affects lens distortion. Given an image which maps from $(x_{\mathrm{unit}}, y_{\mathrm{unit}})$ to its target space, we define the *horizontal* and the *vertical reflector* by

$$\begin{aligned}
\mathrm{rflx} &: (x_{\mathrm{unit}}, y_{\mathrm{unit}}) \mapsto (1 - x_{\mathrm{unit}}, y_{\mathrm{unit}}) \\
\mathrm{rfly} &: (x_{\mathrm{unit}}, y_{\mathrm{unit}}) \mapsto (x_{\mathrm{unit}}, 1 - y_{\mathrm{unit}})
\end{aligned} \tag{100}$$

Reflectors and their inverse are the same:

$$\begin{aligned}
\mathrm{rflx}^{-1} &= \mathrm{rflx} \\
\mathrm{rfly}^{-1} &= \mathrm{rfly}.
\end{aligned} \tag{101}$$

We will also need negation functions like

$$\begin{aligned}
\mathrm{negx} &: (x, y) \mapsto (-x, y) \\
\mathrm{negy} &: (x, y) \mapsto (x, -y)
\end{aligned} \tag{102}$$

Starting from a model function in unit coordinates

$$(x'_{\mathrm{unit}}, y'_{\mathrm{unit}}) = \phi^{-1} \circ g_c \circ \phi(x_{\mathrm{unit}}, y_{\mathrm{unit}}) \tag{103}$$

the question is: How do we have to modify g_c and ϕ if we flip the image? We will get new parameters $\tilde{c}$ and a new coordinate mapping $\tilde{\phi}$. As an example we consider horizontal flipping. The filmback size remains invariant:

$$\tilde{w}_{\mathrm{fb,cm}} = w_{\mathrm{fb,cm}} \qquad \tilde{h}_{\mathrm{fb,cm}} = h_{\mathrm{fb,cm}} \tag{104}$$

Lens center offset is transformed like

$$\tilde{x}_{\mathrm{lco,cm}} = -x_{\mathrm{lco,cm}} \qquad \tilde{y}_{\mathrm{lco,cm}} = y_{\mathrm{lco,cm}} \tag{105}$$

For vertical flipping it would be

$$\tilde{x}_{\mathrm{lco,cm}} = x_{\mathrm{lco,cm}} \qquad \tilde{y}_{\mathrm{lco,cm}} = -y_{\mathrm{lco,cm}}, \tag{106}$$

but let us go ahead with the horizontal case. Flipping the image now leads to the equation

$$(x'_{\mathrm{unit}}, y'_{\mathrm{unit}}) = \mathrm{rflx}^{-1} \circ \tilde{\phi}^{-1} \circ g_{\tilde{c}} \circ \tilde{\phi} \circ \mathrm{rflx}(x_{\mathrm{unit}}, y_{\mathrm{unit}}) \tag{107}$$

First we establish the relationship between ϕ and $\tilde{\phi}$.

$$\begin{aligned}
[\tilde{\phi} \circ \mathrm{rflx}(x_{\mathrm{unit}}, y_{\mathrm{unit}})]_{\mathrm{x}} &= \frac{(1 - x_{\mathrm{unit}})\tilde{w}_{\mathrm{fb,cm}} - \tilde{x}_{\mathrm{lco,cm}}}{\tilde{r}_{\mathrm{fb,cm}}} - \frac{\tilde{w}_{\mathrm{fb,cm}}}{2\tilde{r}_{\mathrm{fb,cm}}} \\
[\tilde{\phi} \circ \mathrm{rflx}(x_{\mathrm{unit}}, y_{\mathrm{unit}})]_{\mathrm{y}} &= \frac{y_{\mathrm{unit}}\tilde{h}_{\mathrm{fb,cm}} - \tilde{y}_{\mathrm{lco,cm}}}{\tilde{r}_{\mathrm{fb,cm}}} - \frac{\tilde{h}_{\mathrm{fb,cm}}}{2\tilde{r}_{\mathrm{fb,cm}}}
\end{aligned} \tag{108}$$

We replace the tilde expressions by the original expressions and simplify:

$$\begin{aligned}
[\ldots]_{\mathrm{x}} &= \frac{-x_{\mathrm{unit}}w_{\mathrm{fb,cm}} + x_{\mathrm{lco,cm}}}{r_{\mathrm{fb,cm}}} + \frac{w_{\mathrm{fb,cm}}}{2r_{\mathrm{fb,cm}}} = -[\phi(x_{\mathrm{unit}}, y_{\mathrm{unit}})]_{\mathrm{x}} \\
[\ldots]_{\mathrm{y}} &= \frac{y_{\mathrm{unit}}h_{\mathrm{fb,cm}} - y_{\mathrm{lco,cm}}}{r_{\mathrm{fb,cm}}} - \frac{h_{\mathrm{fb,cm}}}{2r_{\mathrm{fb,cm}}} = [\phi(x_{\mathrm{unit}}, y_{\mathrm{unit}})]_{\mathrm{y}}
\end{aligned} \tag{109}$$

We insert these expressions in (107) and get

$$(x'_{\mathrm{unit}}, y'_{\mathrm{unit}}) = \phi^{-1} \circ \mathrm{negx} \circ g_{\tilde{c}} \circ \mathrm{negx} \circ \phi(x_{\mathrm{unit}}, y_{\mathrm{unit}}) \tag{110}$$

Similar to reparametrizing for subimages, we compare this expression to (103), which leads us to

$$g_c = \mathrm{negx} \circ g_{\tilde{c}} \circ \mathrm{negx} \tag{111}$$

for horizontal flipping and of course

$$g_c = \mathrm{negy} \circ g_{\tilde{c}} \circ \mathrm{negy} \tag{112}$$

for vertical flipping. We can now apply these two equations for each model function explicitly and get the new parameters $\tilde{c}$.

5.2.1 The radial model

Let us begin with model Poly-4-Radial in dn-coordinates. Again, we omit the index "dn".

$$\begin{aligned} x' &= x(1 + c_2 r^2 + c_4 r^4) \\ y' &= y(1 + c_2 r^2 + c_4 r^4). \end{aligned} \tag{113}$$

Since $(x, y) \mapsto r^2 = x^2 + y^2$ is invariant under negx, equation (112) for this model is

$$\begin{aligned} x(1 + c_2 r^2 + c_4 r^4) &= (-1)[-x(1 + \tilde{c}_2 r^2 + \tilde{c}_4 r^4)] \\ y(1 + c_2 r^2 + c_4 r^4) &= y(1 + \tilde{c}_2 r^2 + \tilde{c}_4 r^4) \end{aligned} \tag{114}$$

which simply means, the coefficients remain invariant under flipping, which is intuitively clear for radially symmetric lenses.

$$\tilde{c}_2 = c_2 \qquad\qquad \tilde{c}_4 = c_4. \tag{115}$$

For the decentered model Poly-4-Radial-Decenter we can examine the two decentering terms separately:

$$\begin{aligned} (r^2 + 2x^2)(u_2 + u_4 r^2) &= -(r^2 + 2x^2)(\tilde{u}_2 + \tilde{u}_4 r^2) \\ (r^2 + 2y^2)(v_2 + v_4 r^2) &= (r^2 + 2y^2)(\tilde{v}_2 + \tilde{v}_4 r^2) \end{aligned} \tag{116}$$

and

$$\begin{aligned} 2xy(v_2 + v_4 r^2) &= -2(-x)y(\tilde{v}_2 + \tilde{v}_4 r^2) \\ 2xy(u_2 + u_4 r^2) &= 2(-x)y(\tilde{u}_2 + \tilde{u}_4 r^2) \end{aligned} \tag{117}$$

In the case of horizontal flipping, these four equations are fulfilled for

$$\begin{aligned} \tilde{u}_2 &= -u_2 & \tilde{v}_2 &= v_2 \\ \tilde{u}_4 &= -u_4 & \tilde{v}_4 &= v_4 \end{aligned} \tag{118}$$

For vertical flipping, the procedure is the same. The result is

$$\begin{aligned} \tilde{u}_2 &= u_2 & \tilde{v}_2 &= -v_2 \\ \tilde{u}_4 &= u_4 & \tilde{v}_4 &= -v_4 \end{aligned} \tag{119}$$

Finally, we should have a look at the beam splitter extender. Equation (112) for model Poly-4-Radial-Decenter-Elliptic reads:

$$h_{\phi_{\mathrm{bs}}, b_{\mathrm{bs}}} \circ g_{\mathrm{rad,dec,c}} = \mathrm{negx} \circ h_{\tilde{\phi}_{\mathrm{bs}}, \tilde{b}_{\mathrm{bs}}} \circ g_{\mathrm{rad,dec,\tilde{c}}} \circ \mathrm{negx} \tag{120}$$

The plan is to find modified parameters $\tilde{\phi}_{\text{bs}}$ and $\tilde{b}_{\text{bs}}$ so that we can exchange h and negx:

$$h_{\phi_{\text{bs}},b_{\text{bs}}} \circ \text{negx} = \text{negx} \circ h_{\tilde{\phi}_{\text{bs}},\tilde{b}_{\text{bs}}} \tag{121}$$

which will make (120) equivalent to

$$g_{\text{rad,dec,c}} = \text{negx} \circ g_{\text{rad,dec},\tilde{c}} \circ \text{negx} \tag{122}$$

which we have already solved by computing parameters $\tilde{c}$ from c. In order to solve (121), let us have a look at the matrix representation as in (37). For our mapped quantities $\tilde{\phi}_{\text{bs}}$ and $\tilde{b}_{\text{bs}}$ we use the short hand notations

$$\begin{aligned} \tilde{q} &= \sqrt{1+\tilde{b}_{\text{bs}}} \\ \tilde{c} &= \cos\tilde{\phi}_{\text{bs}} \\ \tilde{s} &= \sin\tilde{\phi}_{\text{bs}} \end{aligned} \tag{123}$$

Then the right hand side of (121) is

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{c}^2\tilde{q} + \frac{\tilde{s}^2}{\tilde{q}} & (\tilde{q}-\frac{1}{\tilde{q}})\tilde{c}\tilde{s} \\ (\tilde{q}-\frac{1}{\tilde{q}})\tilde{c}\tilde{s} & \frac{\tilde{c}^2}{\tilde{q}} + \tilde{s}^2\tilde{q} \end{bmatrix} = \begin{bmatrix} -\tilde{c}^2\tilde{q} - \frac{\tilde{s}^2}{\tilde{q}} & -(\tilde{q}-\frac{1}{\tilde{q}})\tilde{c}\tilde{s} \\ (\tilde{q}-\frac{1}{\tilde{q}})\tilde{c}\tilde{s} & \frac{\tilde{c}^2}{\tilde{q}} + \tilde{s}^2\tilde{q} \end{bmatrix} \tag{124}$$

For the left hand side of (121) we have

$$\begin{bmatrix} c^2 q + \frac{s^2}{q} & (q-\frac{1}{q})cs \\ (q-\frac{1}{q})cs & \frac{c^2}{q} + s^2 q \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -c^2 q - \frac{s^2}{q} & (q-\frac{1}{q})cs \\ -(q-\frac{1}{q})cs & \frac{c^2}{q} + s^2 q \end{bmatrix} \tag{125}$$

These two matrices are equal if

$$\begin{aligned} \tilde{c} &= c \\ \tilde{s} &= -s \\ \tilde{q} &= q \end{aligned} \tag{126}$$

which means

$$\boxed{\tilde{\phi}_{\text{bs}} = -\phi_{\text{bs}} \qquad \tilde{b}_{\text{bs}} = b_{\text{bs}}} \tag{127}$$

For the vertical flip the procedure is similar and the result is the same.

5.2.2 The anamorphic model

In order to map the anamorphic models we should recall how we defined our polar coordinates:

$$r = \sqrt{x^2+y^2} \qquad \phi = \text{atan}(y,x) \tag{128}$$

r remains invariant when (x, y) are mapped by negx. For ϕ we have

$$\operatorname{atan}(y, -x) = \pi - \operatorname{atan}(y, x) \tag{129}$$

so ϕ is modified but $\cos(i\phi)$ remains invariant, since i is even. For the vertical flip we have

$$\operatorname{atan}(-y, x) = -\operatorname{atan}(y, x) \tag{130}$$

which also modifies ϕ but leaves $\cos(i\phi)$ invariant. Hence, for the plain anamorphic model Poly-4-Anamorphic, equation (112) reads

$$\begin{aligned} x \sum_{\substack{i=0 \\ i\,\text{even}}}^{n} \sum_{\substack{j=i \\ j\,\text{even}}}^{n} c_{ij}^{(\mathrm{x})} r^j \cos(i\phi) &= -(-x) \sum_{\substack{i=0 \\ i\,\text{even}}}^{n} \sum_{\substack{j=i \\ j\,\text{even}}}^{n} \tilde{c}_{ij}^{(x)} r^j \cos(i\phi) \\ y \sum_{\substack{i=0 \\ i\,\text{even}}}^{n} \sum_{\substack{j=i \\ j\,\text{even}}}^{n} c_{ij}^{(\mathrm{y})} r^j \cos(i\phi) &= y \sum_{\substack{i=0 \\ i\,\text{even}}}^{n} \sum_{\substack{j=i \\ j\,\text{even}}}^{n} \tilde{c}_{ij}^{(y)} r^j \cos(i\phi) \end{aligned} \tag{131}$$

which simply means, the coefficients are invariant under the horizontal as well as unter vertical flipping:

$$\boxed{\tilde{c}_{ij}^{(\mathrm{x})} = c_{ij}^{(\mathrm{x})} \qquad \tilde{c}_{ij}^{(\mathrm{y})} = c_{ij}^{(\mathrm{y})}} \tag{132}$$

For the model Poly-4-Anamorphic-Rpa-Sq we have

$$\mathrm{sqy}_{s_\mathrm{y}} \circ \mathrm{sqx}_{s_\mathrm{x} r_\mathrm{pa}} \circ g_\mathrm{anam} \circ \mathrm{sqx}_{r_\mathrm{pa}}^{-1} = \mathrm{negx} \circ \mathrm{sqy}_{\tilde{s}_\mathrm{y}} \circ \mathrm{sqx}_{\tilde{s}_\mathrm{x} r_\mathrm{pa}} \circ g_\mathrm{anam} \circ \mathrm{sqx}_{r_\mathrm{pa}}^{-1} \circ \mathrm{negx} \tag{133}$$

It is easy to see that negx commutes with all squeeze extenders, since all matrices involved are diagonal. Therefore we have

$$= \mathrm{sqy}_{\tilde{s}_\mathrm{y}} \circ \mathrm{sqx}_{\tilde{s}_\mathrm{x} r_\mathrm{pa}} \circ \mathrm{negx} \circ g_\mathrm{anam} \circ \mathrm{negx} \circ \mathrm{sqx}_{r_\mathrm{pa}}^{-1} \tag{134}$$

and since we have already shown that g_anam is invariant under negx:

$$= \mathrm{sqy}_{\tilde{s}_\mathrm{y}} \circ \mathrm{sqx}_{\tilde{s}_\mathrm{x} r_\mathrm{pa}} \circ g_\mathrm{anam} \circ \mathrm{sqx}_{r_\mathrm{pa}}^{-1} \tag{135}$$

and hence

$$\boxed{\tilde{s}_\mathrm{x} = s_\mathrm{x} \qquad \tilde{s}_\mathrm{y} = s_\mathrm{y}} \tag{136}$$

for both the horizontal and the vertical flip. Finally we have a look at the rotation extender which is required for model Poly-4-Anamorphic-Rpa-Sq-Rot. From the matrix representations of negx and $\mathrm{rot}_{\phi_\mathrm{mnt}}$ we have

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi_\mathrm{mnt} & -\sin\phi_\mathrm{mnt} \\ \sin\phi_\mathrm{mnt} & \cos\phi_\mathrm{mnt} \end{bmatrix} = \begin{bmatrix} \cos\tilde{\phi}_\mathrm{mnt} & \sin\tilde{\phi}_\mathrm{mnt} \\ -\sin\tilde{\phi}_\mathrm{mnt} & \cos\tilde{\phi}_\mathrm{mnt} \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \tag{137}$$

Since $\sin\phi_{\mathrm{mnt}}$ is odd and $\cos\phi_{\mathrm{mnt}}$ is even, the rotation angle is mapped according to

$$\tilde{\phi}_{\mathrm{mnt}} = -\phi_{\mathrm{mnt}} \tag{138}$$

for both the horizontal and the vertical flip.

6 Appendix

6.1 Polar coordinates and Cartesian coordinates

Let us have a closer look at the two representations (47) and (49) of the anamorphic polynomial. In order to calculate the Jacobi-matrix we have to expand the polar coordinate representation in powers of x and y. First of all, we can express the cosine functions with arguments $i\phi$, i even, by Chebyshev-polynomials of first kind:

$$\cos i\phi = T_i(\cos\phi) = \sum_{\substack{k=0\\ k\,\text{even}}}^{i} t_{ik}\cos^k\phi. \tag{139}$$

If we insert this in (49) we get products of powers of r and cosine, which we can re-formulate in terms of x and y:

$$r^j\cos^k\phi = x^k(r^2)^{\frac{1}{2}(j-k)} = x^k\sum_{\substack{l=0\\ l\,\text{even}}}^{j-k}\binom{\frac{1}{2}(j-k)}{\frac{1}{2}l}x^l y^{j-k-l} \tag{140}$$

which leads to the Cartesian representation expressed by coefficients of the polar representation (here for x, similar for y):

$$x' = x\sum_{\substack{i=0\\ i\,\text{even}}}^{n}\sum_{\substack{j=i\\ j\,\text{even}}}^{n}\sum_{\substack{k=0\\ k\,\text{even}}}^{i}\sum_{\substack{l=0\\ l\,\text{even}}}^{j-k} c_{ij}^{(\mathrm{x})}t_{ik}\binom{\frac{1}{2}(j-k)}{\frac{1}{2}l}x^{k+l}y^{j-k-l} \tag{141}$$

We now have to replace indices in a way that we can extract the bi-variate monomials in the form $x^u y^v$ with new indices u and v. In order to do this we will have to change the order of summation several times. This is way less complicated if we simplify summation ranges. Let us define $t_{ik} = 0$ for $k > i$. Then we can replace

$$\sum_{\substack{k=0\\ k\,\text{even}}}^{i} \to \sum_{\substack{k=0\\ k\,\text{even}}}^{n} \tag{142}$$

Also, let $c_{ij}^{(\mathrm{x})} = 0$ for $i > j$ or $j > n$, which allows us to replace

$$\sum_{\substack{j=i\\ j\,\text{even}}}^{n} \to \sum_{\substack{j=0\\ j\,\text{even}}}^{n} \tag{143}$$

And third, let us define that binomial coefficients n over k are 0 if $k > n$ or $k < 0$, so that this replacement is possible:

$$\sum_{\substack{l=0 \\ l\,\text{even}}}^{j-k} \to \sum_{\substack{l=0 \\ l\,\text{even}}}^{n} \tag{144}$$

Then we have

$$x' = x \sum_{\substack{i=0 \\ i\,\text{even}}}^{n} \sum_{\substack{j=0 \\ j\,\text{even}}}^{n} \sum_{\substack{k=0 \\ k\,\text{even}}}^{n} \sum_{\substack{l=0 \\ l\,\text{even}}}^{n} c^{(\mathrm{x})}_{ij} t_{ik} \binom{\frac{1}{2}(j-k)}{\frac{1}{2}l} x^{k+l} y^{j-k-l} \tag{145}$$

First we replace l by $j - k - v$ and summate over v.

$$x' = x \sum_{\substack{i=0 \\ i\,\text{even}}}^{n} \sum_{\substack{j=0 \\ j\,\text{even}}}^{n} \sum_{\substack{k=0 \\ k\,\text{even}}}^{n} \sum_{\substack{v=0 \\ v\,\text{even}}}^{n} c^{(\mathrm{x})}_{ij} t_{ik} \binom{\frac{1}{2}(j-k)}{\frac{1}{2}(j-k-v)} x^{j-v} y^{v} \tag{146}$$

Now we move the summation over j to the right (which we can do since the summation ranges are not linked to each other), relabel j by u and shift it by v:

$$x' = x \sum_{\substack{i=0 \\ i\,\text{even}}}^{n} \sum_{\substack{k=0 \\ k\,\text{even}}}^{n} \sum_{\substack{v=0 \\ v\,\text{even}}}^{n} \sum_{\substack{u=0 \\ u\,\text{even}}}^{n} c^{(\mathrm{x})}_{i,u+v} t_{ik} \binom{\frac{1}{2}(u+v-k)}{\frac{1}{2}(u-k)} x^{u} y^{v} \tag{147}$$

We move the summations over u and v to front and extract $c^{(\mathrm{x})}$, x and y:

$$x' = x \sum_{\substack{v=0 \\ v\,\text{even}}}^{n} \sum_{\substack{u=0 \\ u\,\text{even}}}^{n} x^{u} y^{v} \sum_{\substack{i=0 \\ i\,\text{even}}}^{n} c^{(\mathrm{x})}_{i,u+v} \sum_{\substack{k=0 \\ k\,\text{even}}}^{n} t_{ik} \binom{\frac{1}{2}(u+v-k)}{\frac{1}{2}(u-k)} \tag{148}$$

Summations three and four will now generate the coefficients a_{uv} we are looking for. We can restrict our sums that we do not have to rely on our zero-extensions for t, $c^{(\mathrm{x})}$ and the binomial coefficients:

$$x' = x \sum_{\substack{v=0 \\ v\,\text{even}}}^{n} \sum_{\substack{u=0 \\ u\,\text{even}}}^{n-v} x^{u} y^{v} \sum_{\substack{i=0 \\ i\,\text{even}}}^{u+v} c^{(\mathrm{x})}_{i,u+v} \sum_{\substack{k=0 \\ k\,\text{even}}}^{\min(i,u)} t_{ik} \binom{\frac{1}{2}(u+v-k)}{\frac{1}{2}(u-k)} \tag{149}$$

The inner sum can be evaluated e.g. by some simple python program for each triple of indices u, v and i. We have verified that the coefficients to the monomials coincide with (50).

7 List of symbols

$\tilde{}$	Tilde, for marking quantities after reparametrization
$\mathrm{atan}(y, x)$	Arcus tangens for entire x-y-plane
b_{bs}	Bending value of beam splitter
c_2, c_4	Coefficients for the radial models
$c_{ij}^{(\mathrm{x})}$, $c_{ij}^{(\mathrm{y})}$	Coefficients of anamorphic models in polar coordinates
g_{anam}	Polynomial for anamorphic models
$g_{\mathrm{rad,dec}}$	Polynomial for radial models with decentering
$h_{\phi_{\mathrm{bs}},b_{\mathrm{bs}}}(x, y)$	Beam splitter extender
$H_{\phi_{\mathrm{bs}},b_{\mathrm{bs}}}$	Matrix representation of the beam splitter extender
$\hat{J}(x, y)$	Jacobian matrix for plain polynomial models
$J(x, y)$	Jacobian matrix
$\mathrm{negx}(x, y)$, $\mathrm{negy}(x, y)$	Negation as functions
$\phi(x, y)$	Mapping from unit- to dn-coordinates
ϕ_{bs}	Rotation angle of beam splitter
ϕ_{mnt}	Lens rotation for anamorphic models
$\mathrm{rflx}(x, y)$, $\mathrm{rfly}(x, y)$	Flip operations in unit-coordinates
$r_{\mathrm{fb,cm}}$	"Radius" of filmback, distance filmback center to corner
$\mathrm{rot}_{\phi_{\mathrm{mnt}}}$	Rotation extender for anamorphic models
r_{pa}	Pixel aspect ratio
ρ	Ratio of filmback radius' in reparametrization
s_{rscl}	Rescale value for the rescaled anamorphic model
s_{x}, s_{y}	Squeeze values in anamorphic models
$\mathrm{sqx}_q(x, y)$, $\mathrm{sqy}_q(x, y)$	Squeeze extenders for anamorphic models
$S_q^{(\mathrm{x})}$, $S_q^{(\mathrm{y})}$	Matrix representation of squeeze extenders
T_i	i-th Chebyshev polynomial of first kind
t_{ik}	Coefficient for monomial x^k of $T_i(x)$
u_2, u_4	Coefficients for horizontal decentering in radial models
v_2, v_4	Coefficients for vertical decentering in radial models
$w_{\mathrm{fb,cm}}$, $h_{\mathrm{fb,cm}}$	Virtual filmback size in length units
$w_{\mathrm{fb,cm,phys}}$, $h_{\mathrm{fb,cm,phys}}$	Physical filmback size in length units
x, y	Distorted point position
x_{cm}, y_{cm}	Point position in length units
x_{unit}, y_{unit}	Point position in unit coordinates
x_{dn}, y_{dn}	Point position in diagonally normalized coordinates
x', y'	Undistorted point position
$x_{\mathrm{lc,cm}}$, $y_{\mathrm{lc,cm}}$	Lens center in length units
$x_{\mathrm{lco,cm}}$, $y_{\mathrm{lco,cm}}$	Lens center offset in length units

References

[1] Creative Commons Attribution-ShareAlike 3.0 🅭🅯🄎
https://creativecommons.org/licenses/by-sa/3.0/legalcode

[2] VFX Camera Database, http://vfxcamdb.com/anamorphic-workflow/

[3] D. C. Brown, Decentering Distortion of Lenses, *Photom. Eng.*, p.444-462, 1966

[4] A. E. Conrady, Decentered Lens Systems, *Monthly Notices of the Royal Astronomical Society*, Vol. 79, p.384-390, 1919