

Lens Distortion Encoding System

Version 1.0

Jakub Maksymilian Fober
jakub.m.fober@protonmail.com
talk@maxfober.space

Abstract

Lens Distortion Encoding System (LDES) allows for a distortion-accurate workflow, with a seamless interchange of high quality motion picture images regardless of the lens source. This system is similar in a concept to the Academy Color Encoding System (ACES)[Academy 2015], but for distortion. Presented solution is fully compatible with existing software/plugin tools for STMapping found in popular production software like Adobe After Effects or DaVinci Resolve. LDES utilizes common distortion space and produces single high-quality, animatable STMap used for direct transformation of one view to another, neglecting the need of lens-swapping for each shoot. The LDES profile of a lens consist of two elements; *View Map* texture, and *Footage Map* texture, each labeled with the FOV value. Direct distortion mapping is produced by sampling of the *Footage Map* through the *View Map*. The result; animatable mapping texture, is then used to sample the footage to a desired distortion. While the *Footage Map* is specific to a footage, *View Maps* can be freely combined/transitioned and animated, allowing for effects like smooth shift from anamorphic to spherical distortion, previously impossible to achieve in practice. Presented LDES Version 1.0 uses common 32-bit STMap format for encoding, supported by most compositing software, directly or via plug-ins. The difference between standard STMap workflow and LDES is that it encodes absolute pixel position in the spherical image model. The main benefit of this approach is the ability to achieve a similar look of a highly expensive lens using some less expensive equipment in terms of distortion. It also provides greater artistic control and never seen before manipulation of footage.

CCS Concepts: • **Computing methodologies** → **Computer vision; Image processing; Image manipulation; Perception;** Ray tracing; • **Applied computing** → *Media arts.*

Keywords: Camera calibration, Lens distortion correction, Image processing, Anamorphic lenses, Aximorphic distortion, Fisheye lenses, STMap workflow, Visual sphere mapping

1 Introduction

Typical STMap workflow usually involves a mapping between rectilinear projection and specific lens distortion. It is used for image-to-motion camera solving and compositing

of Computer Generated Images (CGI) with real lens footage. The typical process of STMap workflow is limited to a specific camera setup and does not allow for a third-party lens profile end-result. The tools are widely available in compositing and 3D imaging software, both for big-budget production as well as low-budget free software. STMapping is de-facto standard for exchangeable lens distortion management in the video-and-film production industry.[Houdini help 2024]

1.1 New System Allows for Effects Never Seen Before

Beside achieving all the goals of STMap workflow, the LDES solution gives in hands of artists an unlimited digital lens, not constrained by mechanical limitations of optical assembly.[Sasaki 2017b] It can smoothly transition between different lens types, parameters, and utilize synthetic projections.[Williams 2017] It is not constrained by the FOV limit of rectilinear projection, but can work with 0–360° footage. This is possible thanks to transition from rectilinear-pyramidal image base model to spherical model.[Alberti 1435] Which encapsulates all possible perspective projections.

The notion of sphere as a projective surface, a base model for the concept of image was already theorized in ancient times[McArdle 2013]. Even doe rectilinear projection is an old standard, at its conception, one of the most famous artists, LEONARDO DAVINCI was known to criticize it for unnatural presentation of proportions.[Argan and Robb 1946; Da Vinci 1632; Dixon 1987] Today we know that pictures are projections of visual sphere and exhibit same issues found in a field of cartography.[Fleck 1994; German et al. 2007; Peñaranda et al. 2015; Sharpless et al. 2010] The knowledge of the principles found in map projection is useful in creating artistically conscious decisions about desired lens distortion. It can lead an artist to direct viewers perception for the photographed space and its aspects, such that end result is pleasing to the eye and feels natural. Special projection parameters can be tailored to give accurate notion of spatial dimension, speed, distance or proportions to the viewer. This is achieved with LDES and synthetic projections workflow.

© 2024 Jakub Maksymilian Fober



This work is licensed under Creative Commons BY-NC-ND 3.0 license.
<https://creativecommons.org/licenses/by-nc-nd/3.0/legalcode>
For inquiries about custom licensing or collaboration opportunities, please contact the author.

1.2 Document Mathematical Convention

This document adheres to the following naming conventions:

- ★ A left-handed coordinate system is used.
- ★ Vectors are presented in column format.
- ★ Matrices follow a row-major order and are denoted as “ $M_{\text{row col}}$ ”.
- ★ Matrix multiplication is represented as “[column] $_a$ · [row] $_b = M_{ab}$ ”.
- ★ A single bar enclosure “ $|u|$ ” applied to a scalar denotes its *absolute value*.
- ★ A single bar enclosure “ $|\vec{v}|$ ” applied to a vector denotes its *length* or *magnitude*.
- ★ Vectors, whether signed or unsigned, are computed component-wise to produce another vector.
- ★ The centered dot “ \cdot ” indicates the *dot product* of two vectors.
- ★ Square brackets with a comma “[f, c]” denote an *interval*.
- ★ Square brackets without a comma “[$x y$]” denote a *vector* or a *matrix*.
- ★ The exponent “ -1 ” signifies the *reciprocal* of the given value.
- ★ The *QED* symbol “ \square ” marks the final result or output.

These conventions streamline the translation of mathematical formulations into code.

2 Lens Distortion Correction Basics

In this section, we will discuss the types of lens distortions and provide a brief overview of their correction methods. For a more detailed explanation of the mathematical foundations, refer to section No. 4 on the facing page.

2.1 Lens Distortions

Geometric lens distortions can be categorized into two primary types: projection-based distortions and aberrations. Aberrations include barrel distortion, pincushion distortion, and a combination of the two, known as mustache (or radial) distortion. Other types include decentering distortion, which arises from misalignment of optical elements, and thin prism distortion, caused by the tilt of an optical element relative to the imaging sensor.[Wang et al. 2008]

In addition to geometric distortions, there are non-geometric aberrations such as vignetting, which darkens the image toward its periphery; chromatic aberration, which splits incoming light into a spectrum; bokeh, which affects the appearance of out-of-focus areas; and lens flares.

This document primarily focuses on geometric distortions and geometric aberrations, providing a limited overview of vignetting as caused by projection geometry. It is worth noting that vignetting can also result from the occlusion of light by the lens casing, an effect most pronounced when using wide aperture settings and is specific to a given lens setup.

Projection-based distortions result from the projection of the visual sphere onto a flat surface and can be categorized using cartographic terms. In motion picture applications, azimuthal projections are particularly useful as they exhibit only radial distortion, preserving the straightness of lines directed toward the vanishing point on the optical axis. Equirectangular projection can also be beneficial during the production stage, as it encapsulates the entire visual sphere.

Anamorphic or aximorphic distortions represent a special case where the distortion profile differs between the vertical and horizontal axes of the image. Anamorphic lenses are particularly popular in film, commercials, and music video production due to their unique aesthetics when the image is in motion.[Neil 2004; Sasaki 2017a] In contrast, they are rarely used in photography, which is predominantly characterized by wide-angle rectilinear lenses.[Kingslake 1989]

These distortions can arise from projection-based effects or may exhibit axis-dependent aberrations. Spherical fisheye lenses, which belong to the family of cartographic projections, are primarily used in sports and action shots where extensive all-axis camera motion is common.[Thoby 2012]

2.2 Correction Methods

The typical STMap workflow focuses on aberration correction and mapping to rectilinear projection. The most commonly used method in this case is the *Brown-Conrady model*, which addresses various radial aberrations and optical misalignments.[Wang et al. 2008] This model operates entirely in image space and is independent of the camera’s field of view (FOV).

However, the Brown-Conrady method has a limitation when correcting for severe barrel distortion, such as those found in fisheye lenses. In such cases, cartographic mapping provides a more effective solution.[Krause 2019]

Since the LDES is based on visual sphere mapping and is FOV-dependent, cartographic mapping is the preferred approach. When combined with the Brown-Conrady division model, this method enables comprehensive mapping of all lenses. For anamorphic and aximorphic distortions, a custom extended version of this model is employed, allowing individual control over the distortion for each axis of the image.[Fober 2021; Yuan and Sasian 2009]

2.3 Calibration Methods

The LDES provides two calibration methods for obtaining the *view map* and *footage map* of a real lens.

The first method involves fitting the projection model to the lens using checkerboard calibration.

The second method directly maps points in the frame to incidence angles using a specialized calibration rig. This rig includes a visible target at infinity (e.g., a collimated dot-sight) and a 2-axis motorized gimbal. The software tracks the position of the target in the frame and correlates it with the corresponding pan and tilt positions of the gimbal, thereby

determining the incidence direction for a given frame position. Additionally, the sparse data obtained from this method can be used to fit the projection model, offering a hybrid approach to calibration.

3 Lens Distortion Encoding System

The Lens Distortion Encoding System (LDES) incorporates a specialized type of STMaps, that seamlessly combine various lens profiles within a unified encoding scheme. This approach enables the creation of a lens distortion library, providing flexibility of choice in achieving the desired look and feel of the processed image, regardless of its source.

3.1 System Overview

The system relies on two primary texture map types, both associated with a single lens profile. This separation enables direct mapping from one lens to another through a synthesized texture map. During this process, multiple distortion profiles can be combined and animated, allowing for smooth transition between lenses—a technique that was previously nearly impossible to achieve.

Additionally, the LDES supports the application of exotic distortion profiles and the compositing of sources that traditionally were incompatible. While maintaining high fidelity of the final image, the system provides unprecedented creative flexibility and precision in lens distortion handling.

3.2 Core Principles

The system utilizes two types of STMapping files, each corresponding to a single lens profile. The second type can be derived from the first.

View maps match the dimensions of the footage, preserving the original resolution and aspect ratio. The blue channel of the texture can optionally encode vignetting in linear space. This encoding allows for correction of source footage using a division blending mode or the addition of vignetting to the final image using a multiplication blending mode. *View maps* do not encode masking in the alpha channel. The coordinates of *view maps* map to the equidistant projection space of a specified FOV value.

Footage maps are always square textures and do not encode vignetting in the blue channel. Instead, the alpha channel of the *footage map* visualizes the boundaries of the footage in equidistant space. The coordinates of *footage maps* map directly to the footage.

View maps from different sources can be combined using opacity blending, and then applied to a footage of a third source type.

3.3 Implementation Details

Texture maps are encoded in a 32-bit floating-point format, such as *TIFF* or *OpenEXR*. These file-types ensure high precision and compatibility across different software. Each texture

map filename should adhere to the specified naming convention:

FOV values are written as whole numbers, representing the horizontal axis in degrees.

$$\underbrace{\text{ViewMap}}_{\text{type}} \underbrace{\text{_Cinerama1950}}_{\text{description}} \underbrace{\text{_FOV146}}_{\text{FOV}} \text{_tif}$$

$$\underbrace{\text{FootageMap}}_{\text{type}} \underbrace{\text{_TtArtisan7.5mmFisheye}}_{\text{description}} \underbrace{\text{_FOV180}}_{\text{FOV}} \text{_tif}$$

An additional prefix “n” is added for normalized FOV values of view map filenames. For more details, see equation No. 5 on the next page.

$$\underbrace{\text{ViewMap}}_{\text{type}} \underbrace{\text{_Cooke14mmWideAngle}}_{\text{description}} \underbrace{\text{_nFOV92}}_{\text{alt. FOV}} \text{_tif}$$

Figure 1. Naming convention for the LDES files.

The inclusion of the FOV value in the filename is mandatory, for the mapping process to function. The FOV is always specified as a whole number in degrees. If the exact lens value is a decimal numeral, the texture coordinates must be scaled appropriately to match the noted FOV value. For further details, see equation No. 5 on the following page. As a good practice, the FOV noted value should be rounded up in the creation process.

4 Mathematical Foundations

Both *view maps* and *footage maps* encode STMap coordinates. The *footage map* represents footage *ST* coordinates in equidistant space θ . The boundary of the *footage map* corresponds to an FOV angle of $\Omega/2$.

$$\begin{bmatrix} \vec{f}_{\text{map}_s} \\ \vec{f}_{\text{map}_t} \end{bmatrix} \mapsto \begin{bmatrix} \vec{\text{tex}}_s \\ \vec{\text{tex}}_t \end{bmatrix} \quad (1)$$

The *view map* *ST* coordinates encode the *footage map* coordinates within the image space of the result. These coordinates are normalized to a specified FOV angle Ω .

$$\begin{bmatrix} \vec{v}_{\text{map}_s} \\ \vec{v}_{\text{map}_t} \end{bmatrix} \mapsto \begin{bmatrix} \vec{f}_{\text{map}_s} \\ \vec{f}_{\text{map}_t} \end{bmatrix} \quad (2a)$$

$$\Omega_{\text{v_map}} \left\| \begin{bmatrix} \vec{v}_{\text{map}_s} \\ \vec{v}_{\text{map}_t} \end{bmatrix} \right\| - \frac{1}{2} = \theta \quad (2b)$$

If there is a mismatch between the *footage map* and *view map* FOV angle Ω , a simple tile scaling is applied when sampling the *footage map* over the *view map* coordinates.

$$\text{tile_scale} = \frac{\Omega_{\text{view_map}}}{\Omega_{\text{footage_map}}} \quad (3)$$

Both maps are not intended to directly sample the footage, as this would result in quality loss. Instead, the *footage map* is sampled through the *view map* to produce the final distortion mapping SMap, which is then used to sample the footage.

$$\begin{bmatrix} \vec{v}_{\text{map}_s} \\ \vec{v}_{\text{map}_t} \end{bmatrix} \xrightarrow{\text{samples}} \begin{bmatrix} \vec{f}_{\text{map}_s} \\ \vec{f}_{\text{map}_t} \end{bmatrix} \mapsto \begin{bmatrix} \vec{\text{tex}}_s \\ \vec{\text{tex}}_t \end{bmatrix} \quad (4)$$

Animated View Maps. View maps can be combined to create animated *view maps* that transition between different distortion parameters. This animation is achieved by interpolating between the two maps, for example, by adjusting the opacity of one map over the other.

To ensure a seamless transition, it is recommended to first match the FOV value Ω of each processed *view map*, avoiding necessity for tile-scaling during animation. Normalization of the encoded FOV angle is performed by scaling the *view map* coordinates to a common FOV value, which then is used to control the tile scaling.

$$\begin{bmatrix} \vec{v}_{\text{map}'_s} \\ \vec{v}_{\text{map}'_t} \end{bmatrix} = \frac{\Omega_{\text{v_map}}}{\Omega_{\text{common}}} \left(\begin{bmatrix} \vec{v}_{\text{map}_s} \\ \vec{v}_{\text{map}_t} \end{bmatrix} - \frac{1}{2} \right) + \frac{1}{2} \quad (5)$$

Ray-Tracing with Lens Distortion. View maps can be utilized to directly drive the ray-tracing engine in 3D imaging software, such as BLENDER, by transforming the encoded coordinates into incidence \hat{I} vectors.[Blender Manual 2024]

$$\theta = \Omega_{\text{v_map}} \left\| \begin{bmatrix} \vec{v}_{\text{map}_s} \\ \vec{v}_{\text{map}_t} \end{bmatrix} - \frac{1}{2} \right\| \quad (6a)$$

$$\begin{bmatrix} \hat{I}_x \\ \hat{I}_y \\ \hat{I}_z \end{bmatrix} = \begin{bmatrix} \sin(\theta) \left\| \begin{bmatrix} \vec{v}_{\text{map}_s} \\ \vec{v}_{\text{map}_t} \end{bmatrix} - \frac{1}{2} \right\| \\ \cos(\theta) \end{bmatrix} \quad (6b)$$

4.1 Distortion Models

The LDES employs a cartographic fisheye projection model as the foundation for synthetic view maps.[Krause 2019] These maps can be further refined through extensions to align with the distortion profile of real-life lens.

Synthetic Projection Base Model. This model allows for a seamless transition from rectilinear projection to a fisheye-orthographic view. Each value of k below 1 corresponds to a different type of fisheye lens. The table No. 1 outlines the major values and their corresponding lens type.

$$f = \begin{cases} \frac{\tan\left(\frac{\Omega}{2}k\right)}{k}, & \text{if } k > 0 \\ \frac{\Omega}{2}, & \text{if } k = 0 \\ \frac{\sin\left(\frac{\Omega}{2}k\right)}{k}, & \text{if } k < 0 \end{cases} \quad (7a)$$

$$r = |\vec{v}| = \sqrt{\vec{v}_x^2 + \vec{v}_y^2} = \begin{cases} \frac{\tan(\theta k)f}{k}, & \text{if } k > 0 \\ \theta f, & \text{if } k = 0 \\ \frac{\sin(\theta k)f}{k}, & \text{if } k < 0 \end{cases} \quad (7b)$$

Value of k	Fisheye Type
$k = 1$	Rectilinear (<i>Gnomonic</i>)
$k = 1/2$	Stereographic
$k = 0$	Equidistant
$k = -1/2$	Equisolid
$k = -1$	Orthographic (<i>azimuthal</i>)

Source: PTGui 11 fisheye factor [Krause 2019].

Table 1. Primary k values and corresponding fisheye projection type.

$$\theta = \begin{cases} \frac{\arctan\left(\frac{r}{f}k\right)}{k}, & \text{if } k > 0 \\ \frac{r}{f}, & \text{if } k = 0 \\ \frac{\arcsin\left(\frac{r}{f}k\right)}{k}, & \text{if } k < 0 \end{cases} \quad (7c)$$

where:

- ★ f is the focal length derived from horizontal FOV angle Ω ,
- ★ r is the image space radius originating from the center, which is derived from:
- ★ \vec{v} the view coordinate \mathbb{R}^2 vector in image space, with correct aspect ratio, originating from the center.

Anamorphic Extension. The anamorphic extension to the radius r enables vertical scaling of the distortion based on the anamorphic squeeze factor s .

$$r = \sqrt{\vec{v}_x^2 + \frac{\vec{v}_y^2}{s}} \quad (8a)$$

$$\vec{v}' = \begin{bmatrix} \vec{v} \\ r \end{bmatrix} \quad (8b)$$

Aximorphic Extension. The aximorphic extension to the θ angle provides independent control over the projection type for the vertical and horizontal axes. It serves as an alternative to anamorphic projection. The angles θ_x and θ_y are calculated separately, each with its own k value, while sharing a common focal length f .

$$\begin{bmatrix} \vec{w}_x \\ \vec{w}_y \end{bmatrix} = \begin{bmatrix} \cos^2 \varphi \\ \sin^2 \varphi \end{bmatrix} = \begin{bmatrix} \frac{1}{2} + \frac{\cos(2\varphi)}{2} \\ \frac{1}{2} - \frac{\cos(2\varphi)}{2} \end{bmatrix} = \begin{bmatrix} \frac{\vec{v}_x^2}{\vec{v}_x^2 + \vec{v}_y^2} \\ \frac{\vec{v}_y^2}{\vec{v}_x^2 + \vec{v}_y^2} \end{bmatrix} \quad (9a)$$

$$\theta' = \vec{w} \cdot \vec{\theta}_{xy} = \vec{w}_x \vec{\theta}_x + \vec{w}_y \vec{\theta}_y \quad (9b)$$

where:

- ★ \vec{w} consists of aximorphic interpolation weights,
- ★ φ is the azimuthal angle of the incidence vector in the visual sphere space,
- ★ θ is the polar angle of the incidence in the visual sphere space.

Asymmetrical-Aximorphic Extension. The aximorphic projection can be further extended to provide independent control over the top and bottom portions of the image with respect to the projection type.

$$k_y = \begin{cases} k_{top}, & \text{if } \vec{v}_y > 0 \\ k_{bottom}, & \text{if } \vec{v}_y \leq 0 \end{cases} \quad (10)$$

Brown-Conrady Extension. The Brown-Conrady extension offers geometric aberration control over the distortion profile. It can be seamlessly integrated with both anamorphic and aximorphic extensions.

$$\begin{bmatrix} \vec{f}_x \\ \vec{f}_y \end{bmatrix} = \underbrace{\begin{bmatrix} \vec{v}_x - c_1 \\ \vec{v}_y - c_2 \end{bmatrix}}_{\text{cardinal offset (A)}} \quad (11a)$$

$$r^2 = \vec{f}_x^2 + \vec{f}_y^2 \quad (11b)$$

$$\begin{bmatrix} \vec{v}'_x \\ \vec{v}'_y \end{bmatrix} = \underbrace{\begin{bmatrix} \vec{f}_x \\ \vec{f}_y \end{bmatrix} \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \dots \right)^{-1}}_{\text{radial}} + \underbrace{\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}}_{\text{cardinal offset (B)}} \quad \square$$

$$+ \underbrace{\begin{bmatrix} \vec{f}_x \\ \vec{f}_y \end{bmatrix} \left(\begin{bmatrix} \vec{f}_x \\ \vec{f}_y \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \right)}_{\text{decentering}} + r^2 \underbrace{\begin{bmatrix} q_1 \\ q_2 \end{bmatrix}}_{\text{thin prism}} \quad \square \quad (11c)$$

where:

- ★ c_1 and c_2 are the cardinal offset parameters for horizontal and vertical axis,
- ★ k_1, k_2, k_n are the radial distortion coefficients (different from the k projection parameter),
- ★ p_1 and p_2 are the decentering parameters,
- ★ q_1 and q_2 are the thin prism distortion parameters.

Aximorphic Brown-Conrady Extension. The Brown-Conrady extension can be further enhanced to provide independent control over radial distortion for the vertical and horizontal axes. This is particularly useful for matching anamorphic lens distortion.

$$\begin{bmatrix} \vec{f}_x \\ \vec{f}_y \end{bmatrix} \underbrace{\left(\begin{bmatrix} \vec{w}_x \\ \vec{w}_y \end{bmatrix} \cdot \begin{bmatrix} 1 + \vec{k}_{x1} r^2 + \vec{k}_{x2} r^4 + \vec{k}_{x3} r^6 \dots \\ 1 + \vec{k}_{y1} r^2 + \vec{k}_{y2} r^4 + \vec{k}_{y3} r^6 \dots \end{bmatrix} \right)^{-1}}_{\text{aximorphic radial}} \quad (12)$$

where:

- ★ \vec{k}_{xn} and \vec{k}_{yn} are same as defined before, but driving horizontal and vertical axis separately.

4.2 Interpolation and Sampling

Bilinear interpolation can be applied throughout the entire workflow with satisfactory results. However, *view map* sampling of the *footage map* may benefit from bi-cubic Catmull-Rom interpolation, although it is not strictly necessary.

Due to the nature of STMaps, which consist of smooth gradients, high resolution is generally unnecessary. Conversely, the final mapping STMap can benefit from resolution scaling beyond the size of the original footage. This is particularly relevant when using third-party plugins for STMapping in compositing software, as some of these plugins lack support for MIP-mapping or anisotropic filtering. In such cases, additional resolution scaling of the STMap can act as a multi-sampling filter, mitigating potential quality loss in the final distorted image.

5 Workflow Example

In this section, we will explore two types of workflows: the application of LDES maps to a footage and the process of selecting a synthetic distortion profile for a given scene.

5.1 Compositing Example

In this thought exercise, we will examine the compositing process for combining Computer Generated Images (CGI) with real camera footage, viewed through an animated synthetic projection lens.

Overview of Assets. We begin with high-resolution footage shot using a wide fisheye lens. Calibration assets include a *view map* with measured vignetting and a derived *footage map*. For the final output, we select two synthetic *view maps* from a library. Additionally, a rectilinear *view map* is generated for solving camera motion from the footage. Below is a summary of the assets:

1. High-resolution live footage captured with a wide-angle fisheye lens.
2. Calibration *view map* corresponding to the footage, including vignetting.
3. Calibration *footage map* matching the footage.
4. Synthetic *view map* resembling an anamorphic lens.
5. Synthetic *view map* resembling a spherical lens with the same FOV.
6. Rectilinear *view map* for solving camera motion.
7. 3D scene for rendering.

The compositing software used is *Adobe After Effects* with the *B.Map* plug-in. All assets are imported into the program, ensuring that color management is disabled for the maps. The composition is set to 32-bit floating-point precision to maintain the proper image quality.

Tracking Rectified Footage. Next, we rectify the footage and track the camera motion for use in our 3D software. To begin, we create a composition using the rectilinear *view map* and import the footage along with its corresponding

footage map. The *B.Map* effect is applied to the rectilinear *view map* layer. We flip the vertical tiling and adjust the tiling scale to account for the FOV mismatch between the *view map* and the *footage map* (see equation No. 3 on page 3).

The sampled texture is then set to the *footage map* layer, followed by applying a second *B.Map* effect, which samples the footage layer. The result is a layer with rectified footage ready for camera tracking.

Before proceeding to render the 3D scene, we can prepare the animation of the lens distortion, which will later be used for ray-tracing the scene. This approach eliminates the need to warp the CGI sequence to match it with the final output.

Animating View Maps. To animate the two synthetic *view maps*, we create a new composition matching the *view map* size and import both maps. The distortion is animated by adjusting the opacity of one map over the other, and the two layers are pre-composed. This pre-composition can then be exported as an image sequence to drive the ray-tracing engine (see equation No. 6 on page 4).

On the pre-composed layer, the *B.Map* effect can be applied to sample the *footage map*. Following the previous steps, the warp-sampled footage can then be finalized to achieve the desired look.

Ray-Tracing with Distortion. In our 3D software, here *Blender*, we import the animated *view map* sequence and set-up nodes to convert the map coordinates into ray vectors (see equation No. 6 on page 4). These ray vectors are used to drive the *Ray Portal BSDF* node. [Blender Manual 2024]

The final 3D render can then be directly ray-traced with the embedded distortion, resulting in an animated lens that matches the warped footage from the compositing software.

Note. To enhance the quality of footage warping, $4\times$ MSAA filtering can be applied. This process involves pre-composing the animated *view map* with the *footage map* and increasing the resolution of the pre-composition by a factor of 4. The final *B.Map* effect, used for sampling the footage, is then applied to this high-resolution pre-composition.

5.2 Choosing Synthetic View Parameters

Understanding how distortion impacts the viewer’s experience is crucial. Film directors typically select lenses based on their familiarity with specific lens characteristics and by conducting test shots using various options. [Giardina 2016; Neil 2004; Sasaki 2017a]

By carefully choosing the synthetic projection, we can parameterize the viewing experience and make deliberate decisions about the visual feel and aesthetic of the image. Each type of lens projection imparts a distinct sensation to the viewer. Using the aximorphic model, it is possible to combine multiple projection characteristics into a single, non-animated distortion.

The table No. 2 illustrates the typical sensory associations linked to the k value from the aximorphic model.

k Value & Projection Type	Perception of Space
1 Rectilinear	straight lines
$1/2$ Stereographic	shape, angles, space
0 Equidistant	speed, aim
$-1/2$ Equisolid	distance, size
-1 Orthographic (<i>fisheye</i>)	brightness (<i>no vignetting</i>)

Source: Empirical self study.

Table 2. Projection type and corresponding perception of space, combined with k value.

Exercise No. 1. Consider a dynamic shot with a camera mounted on the bumper of a car. The goal is to convey the speed of the car, the approaching distance to the target ahead, and the spatial awareness of the surroundings.

To achieve this, we can divide the frame into three regions: the bottom of the frame, showing the asphalt moving beneath the car; the top half of the frame, where the target appears to approach the car; and the sides of the frame, depicting the surrounding environment. Different k values can be assigned to each region based on the information in Table 2, utilizing the aximorphic camera model.

- For the asphalt bottom region, we assign $k_{\downarrow}0$ to enhance the sensation of speed.
- For the top half of the frame, where the target is approaching, we assign $k_{\uparrow} - 1/2$ to provide a sense of distance.
- For the sides of the frame, we assign $k_x^{1/2}$ to enhance the perception of space.

This approach allows for a nuanced manipulation of the viewer’s sensory experience, tailored to the unique characteristics of each frame region.

Exercise No. 2. Consider a generic scene featuring a family seated around a table. The goal is to capture the entire scene using a super wide-angle shot, while avoiding any distortion of the actors’ faces. The camera is stationary on the table, and the family members are positioned around it. For this task, we select a projection that emulates the aesthetics of anamorphic photography.

- For the horizontal axis of the frame, we assign a k_x value of $1/2$ to preserve the proportions of the actors’ faces.
- To minimize the distracting effects of the wide-angle shot and maintain focus on the performance, we can straighten the vertical lines of the shot by choosing a k_y value of 0.88, which closely resembles a rectilinear projection.

This configuration ensures the scene retains a natural and engaging appearance, emphasizing the performance while leveraging the expansive field of view provided by the wide-angle lens.

Note. Both exercises can be accomplished using a wide-angle fisheye lens, with the desired look tailored during the post-production process by incorporating the LDES workflow.

6 Future Work for Version 1.0

Future work on this subject will include an evaluation of the calibration methods mentioned in Subsection 2.3 on page 2. Further areas of exploration include:

- **Real-Time Applications:** Explore the implementation of LDES in real-time systems, such as Virtual Production and in-camera special effects, where lens distortion correction and projection switching need to occur dynamically.
- **Integration with Rendering Pipelines:** Evaluate the application of LDES in rendering pipelines for visual effects, gaming, and 3D applications, including other ray-tracing workflows.
- **Subjective User Studies:** Conduct studies to evaluate how different synthetic distortions and transitions affect viewer perception, helping to refine the design of view maps for storytelling or aesthetic purposes.
- **Cross-Platform Compatibility:** Ensure compatibility of LDES across various software ecosystems, such as compositing, rendering, and editing tools.
- **Library of Lens View Maps:** Develop an extensive, standardized library of lens *view maps* for a variety of lens types, focal lengths, and distortion characteristics. This library could serve as a shared resource for professionals in filmmaking and visual effects, streamlining workflows and increasing creative flexibility.

These directions aim to broaden the utility and applicability of LDES while addressing potential limitations, and advancing the field of lens distortion modeling, directing, and correction.

References

Academy. 2015. What are the Advantages of using ACES for Color Correction? Oscars.org. <https://oscars.org/aces/what-are-advantages-using-aces-color-correction>

Leon B. Alberti. 1435. *On Painting* (1970 ed.). New Haven: Yale University Press. <http://www.noteaccess.com/Texts/Alberti/index.htm>

Giulio C. Argan and Nesca A. Robb. 1946. The Architecture of Brunelleschi and the Origins of Perspective Theory in the Fifteenth Century. *Journal of the Warburg and Courtauld Institutes* 9 (1946), 96. <https://doi.org/10.2307/750311>

Blender Manual. 2024. *Ray Portal BSDF* (4.3 ed.). Blender Foundation. https://docs.blender.org/manual/en/latest/render/shader_nodes/shader/ray_portal.html

Leonardo Da Vinci. 1632. *A treatise on painting* (2014 ed.). Project Gutenberg, Chapter Linear Perspective, 49–59. <http://gutenberg.org/ebooks/46915>

Robert A. Dixon. 1987. *Mathographics*. Basil Blackell Limited, Chapter Perspective Drawings, 82–83.

Margaret M. Fleck. 1994. *Perspective Projection: the Wrong Imaging Model*. techreport 95-01. Computer Science, University of Iowa. <https://mfleck.cs.illinois.edu/my-papers/stereographic-TR.pdf>

Jakub Maksymilian Fober. 2021. Aximorphic Perspective Projection Model for Immersive Imagery. (Feb. 2021). <https://doi.org/10.48550/ARXIV.2102.12682> arXiv:2102.12682 [cs.GR]

Daniel M. German, Pablo D'Angelo, Michael Gross, and Bruno Postle. 2007. New Methods to Project Panoramas for Practical and Aesthetic Purposes. In *Computational Aesthetics in Graphics, Visualization, and Imaging*. The Eurographics Association. <https://doi.org/10.2312/COMPAESTH/COMPAESTH07/015-022>

Carolyn Giardina. 2016. How 'The Hateful Eight' Cinematographer Revived Lenses From the 1960s. *The Hollywood Reporter* (Jan. 2016). <https://hollywoodreporter.com/news/how-hateful-eight-cinematographer-revived-852586> Online.

Houdini help. 2024. *Lens effects* (20.5.423 ed.). Houdini. https://www.sidefx.com/docs/houdini/composite/lens_effects.html

Rudolf Kingslake. 1989. *A History of the Photographic Lens*. Academic Press, Chapter IV, 59–62.

Erik Krause. 2019. *Fisheye Projection*. PanoTools wiki. https://wiki.panotools.org/index.php?title=Fisheye_Projection&oldid=16077 Online.

James M. McArdle. 2013. From the corner of your eye?... Personal blog. <https://drjamesmcardle.com/2013/04/07/from-the-corner-of-your-eye/> Online.

Iain A. Neil. 2004. Lenses in Cinematography. *Optics and Photonics News* 15, 1 (2004), 26. <https://doi.org/10.1364/opn.15.1.000026>

Luis Peñaranda, Luiz Velho, and Leonardo Sacht. 2015. Real-time correction of panoramic images using hyperbolic Möbius transformations. *Journal of Real-Time Image Processing* 15, 4 (may 2015), 725–738. <https://doi.org/10.1007/s11554-015-0502-x>

Dan Sasaki. 2017a. The Aesthetics of Anamorphic in Film and Digital. Panavision Inc. <https://vimeo.com/167052303> Video.

Dan Sasaki. 2017b. The Five Pillars of Anamorphic - Disproportionate Breathing. Panavision Inc. <https://vimeo.com/167045643> Video.

Thomas K. Sharpless, Bruno Postle, and Daniel M. German. 2010. Panini: A New Projection for Rendering Wide Angle Perspective Images. *Computational Aesthetics in Graphics Visualization, and Imaging* (2010). <https://doi.org/10.2312/COMPAESTH/COMPAESTH10/009-016>

Michael Thoby. 2012. About the various projections of the photographic objective lenses. Personal website. https://web.archive.org/web/20230821044935/http://michel.thoby.free.fr/Fisheye_history_short/Projections/Various_lens_projection.html Online.

Jianhua Wang, Fanhuai Shi, Jing Zhang, and Yuncai Liu. 2008. A new calibration model of camera lens distortion. *Pattern Recognition* 41, 2 (2008), 607–615. <https://doi.org/10.1016/j.patcog.2007.06.012>

Shaun E. Williams. 2017. Flex FOV. GitHub, Inc. <https://github.com/shaunlebron/flex-fov> Modification, Minecraft.

Sheng Yuan and Jose Sasian. 2009. Aberrations of anamorphic optical systems I: the first-order foundation and method for deriving the anamorphic primary aberration coefficients. *Applied Optics* 48, 13 (2009), 2574. <https://doi.org/10.1364/ao.48.002574>

Received November 2024