Catching fireflies

Xaq Pitkow

January 14, 2015

Abstract

Notes on things useful for the firefly project.

1 Virtual Reality projection geometry

1.1 Goal

Our goal is to define a mapping from a rendered image to a projected image, such that the projection appears the same from the animal's vantage point as it would in the virtual world (Figure 1, magenta spheres).

For simplicity, instead of using spherical angles, we use vectors on unit spheres. This means that we can write our goal mathematically as

$$\frac{\boldsymbol{z}-\boldsymbol{a}}{|\boldsymbol{z}-\boldsymbol{a}|}=\frac{\boldsymbol{w}-\boldsymbol{o}}{|\boldsymbol{w}-\boldsymbol{o}|} \tag{1}$$

1.2 Distortion mapping

A point $\boldsymbol{u}$ in the virtual world maps to a point on the unit sphere, given by the unit vector $\boldsymbol{v}$. OpenGL renders this point onto an image point $\boldsymbol{s}$ by some function $f : \boldsymbol{v} \to \boldsymbol{s}$. We will apply a distortion map $g : \boldsymbol{s} \to \boldsymbol{t}$ to map this to a projected image point with coordinate $\boldsymbol{t}$. This is then mapped by the mirror system to a point on the perceptual unit sphere $\boldsymbol{\zeta}$ through $h : \boldsymbol{t} \to \boldsymbol{\zeta}$. Combining all of these, we have

$$\boldsymbol{\zeta} = h(g(f(\boldsymbol{v}))) \tag{2}$$

and we want $\boldsymbol{\zeta} = \boldsymbol{v}$.

To solve for g, we have $\boldsymbol{v} = h(g(f(\boldsymbol{v})))$, so $g(f(\boldsymbol{v})) = h^{-1}(\boldsymbol{v})$. Since g applies to an image point $\boldsymbol{s}$, we must solve for

$$g(\boldsymbol{s}) = h^{-1}(f^{-1}(\boldsymbol{s})) \tag{3}$$

This will provide the identity map we desire from virtual to real angles.

1.3 Camera and Projector

The virtual camera renders images at unit vector $\boldsymbol{v}$ in the virtual world onto a flat image, at position $\boldsymbol{w} = \boldsymbol{b} + \boldsymbol{s}$, where $\boldsymbol{s}$ are screen coordinates and live on the plane orthogonal to the direction the camera is pointing: $\boldsymbol{s} \cdot \boldsymbol{b} = 0$.

For convenience, I imagine the projection of the screen onto the sphere is circular, so we only have to handle one scale or field of view (FOV) for each image (although the planar projection need

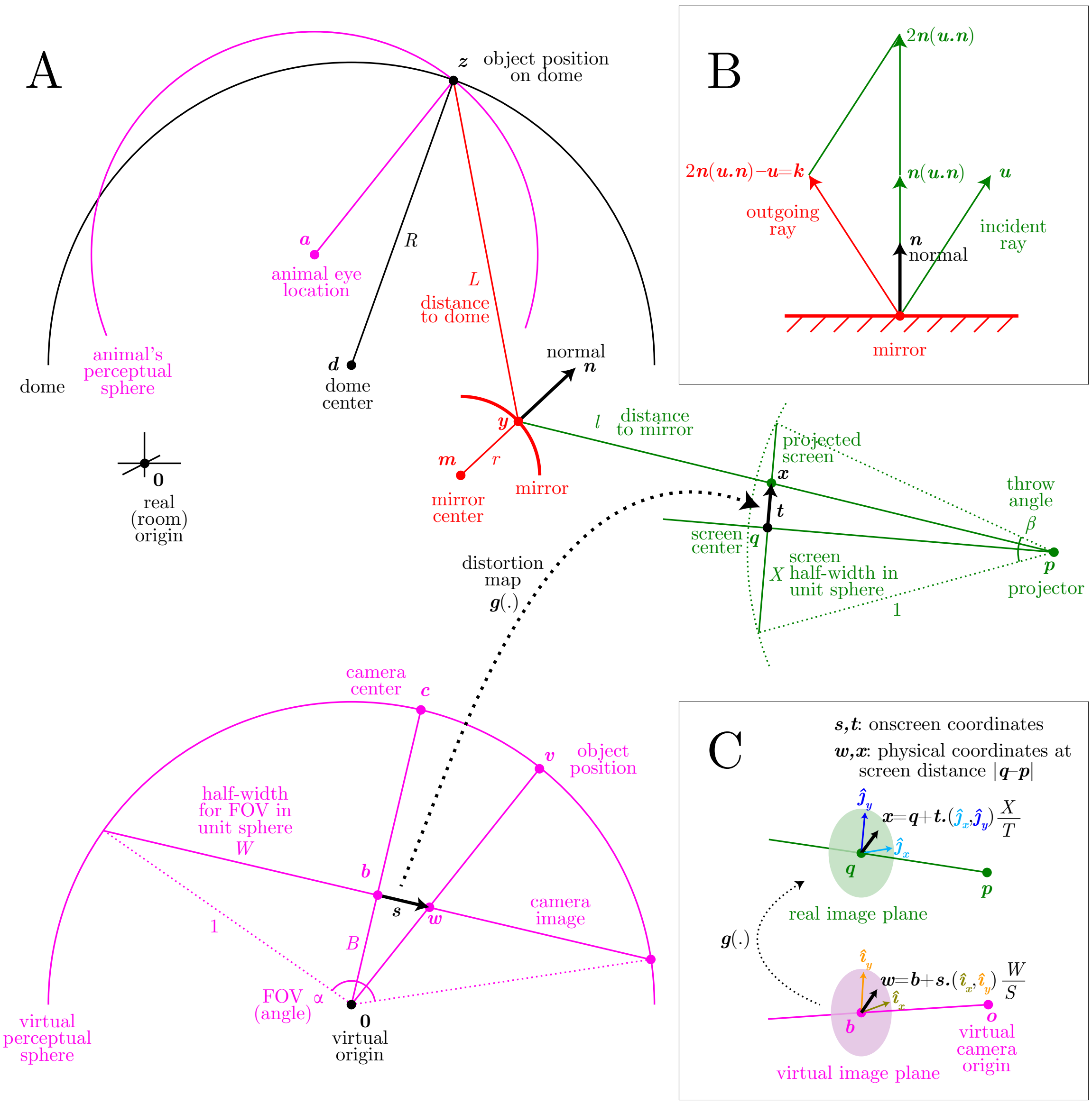

Figure 1: (A) Geometry of spherical projection. (B) Geometry of mirror reflection. (C) Screen coordinates can be expressed as 2D vectors: $\boldsymbol{s} = (s_x, s_y)$ for virtual camera render, $\boldsymbol{t} = (t_x, t_y) = \boldsymbol{g}(\boldsymbol{s})$ for projected image; or as 3D vectors in virtual space: $\boldsymbol{w} = \boldsymbol{b}+s_x\hat{\boldsymbol{\imath}}_x+s_y\hat{\boldsymbol{\imath}}_y$ for virtual world, $\boldsymbol{x} = \boldsymbol{q}+t_x\hat{\boldsymbol{\jmath}}_x+t_y\hat{\boldsymbol{\jmath}}_y$ for real world. Here the basis vectors $(\hat{\boldsymbol{\imath}}_x, \hat{\boldsymbol{\imath}}_y)$ and $(\hat{\boldsymbol{\jmath}}_x, \hat{\boldsymbol{\jmath}}_y)$ are orthonormal pairs. (Note that every letter in the english alphabet is used at least once!)

not be a circle unless the projector is pointed at the mirror center!). The sizes of the images are S for the camera or T for the projection image (and may be equal, or both equal to 1). Cropping to rectangular screens is a straightforward extension. These screens subtend an angle in the virtual or real world given by α and β. These are related to the screen half-widths W and X according to

$$W = \sin\frac{\alpha}{2} \tag{4}$$

$$X = \sin\frac{\beta}{2} \tag{5}$$

These quantities set the scale of coordinates. The $\boldsymbol{s}$ and $\boldsymbol{t}$ orthonormal bases are $I = (\hat{\boldsymbol{\imath}}_x, \hat{\boldsymbol{\imath}}_y)$ and $J = (\hat{\boldsymbol{\jmath}}_x, \hat{\boldsymbol{\jmath}}_y)$, respectively. For convenience, we have chosen these vectors to be orthogonal to the center ray leaving the camera or projector. Using these as basis vectors, the ray through the projection of screen point $\boldsymbol{s}$ or $\boldsymbol{t}$ intersects the unit sphere centered on camera or projector at the points

$$\boldsymbol{w} = \boldsymbol{b} + I\boldsymbol{s}\frac{W}{S} \tag{6}$$

$$\boldsymbol{x} = \boldsymbol{q} + J\boldsymbol{t}\frac{X}{T} \tag{7}$$

Therefore, a point on the virtual camera image plane $\boldsymbol{s}$ appears at the unit vector

$$\boldsymbol{v} = \frac{\boldsymbol{w}}{|\boldsymbol{w}|} \tag{8}$$

For convenience, if we define the unit normalization function $U(\boldsymbol{x}) = \boldsymbol{x}/|\boldsymbol{x}|$, then

$$\boldsymbol{v} = U(\boldsymbol{b} + \frac{W}{S}I\boldsymbol{s}) = f^{-1}(s) \tag{9}$$

which is one part of the function we need for the distortion map g. The other part is determined by the transformation $h : \boldsymbol{t} \to \zeta$.

For the mapping from unit vector to screen coordinate, $f : \boldsymbol{v} \to \boldsymbol{s}$, we can use similar triangles. Decompose $\boldsymbol{v}$ into the component in the direction of the projection $\boldsymbol{b}$, which is $v_\parallel$, and the components in the plane of the screen, $\boldsymbol{v}_\perp$. We drop the parallel coordinate by multiplying with the 2×3 orthonormal matrix $I^\top = (\hat{\boldsymbol{\imath}}_x, \hat{\boldsymbol{\imath}}_y)^\top$. The 2D vector that remains is proportional to $\boldsymbol{s}$, with a proportionality determined by $\frac{v_\parallel}{B}$ which is the ratio of distance to $\boldsymbol{v}$ over distance B to the image plane. Defining the image plane as touching the unit sphere at the subtended angle given by $\alpha = \text{FOV}$, trig gives us have $B = \cos\alpha/2$. The half-width in this image plane is W, not the target image size S, so we then rescale by S/W. With this decomposition of $\boldsymbol{v}$ and the two rescalings we get

$$\boldsymbol{s} = \frac{S}{W}\frac{B}{v_\parallel}I^\top\boldsymbol{v} = S\frac{B}{W}\left(\frac{I^\top\boldsymbol{v}}{v_\parallel}\right) \tag{10}$$

where $\boldsymbol{c} = \frac{\boldsymbol{b}}{|\boldsymbol{b}|}$. Finally, we find

$$\boldsymbol{s} = f(\boldsymbol{v}) = S\cot\frac{\alpha}{2}\left(\frac{\boldsymbol{v}}{\boldsymbol{v}\cdot\boldsymbol{c}} - \boldsymbol{c}\right) \tag{11}$$

1.4 Projector to Dome

Image coordinates $\boldsymbol{s}$ are transformed by the distortion map function $\boldsymbol{t} = \boldsymbol{g}(\boldsymbol{s})$. This new distorted screen coordinate is projected to a virtual flat screen to position $\boldsymbol{t} = \boldsymbol{x} - \boldsymbol{q}$. The ray from the projector source $\boldsymbol{p}$ through $\boldsymbol{x}$ hits the mirror at position $\boldsymbol{y}$. We can define a tangent plane to the spherical mirror at this point, with normal vector $\boldsymbol{y}-\boldsymbol{m}$. This tangent plane allows us to define the reflection unit vector, which extends to hit the dome at $\boldsymbol{z}$. This position on the dome appears along the vector $\boldsymbol{z}-\boldsymbol{a}$ from the animal observer at $\boldsymbol{a}$.

The target point $\boldsymbol{y}$ on the mirror that is hit by the ray from $\boldsymbol{p}$ to the screen position $\boldsymbol{x}$ is defined by the equations

$$\boldsymbol{y}=\boldsymbol{p}+l\,U(\boldsymbol{x}-\boldsymbol{p}) \tag{12}$$

and

$$|\boldsymbol{y}-\boldsymbol{m}|=\mu \tag{13}$$

where l is the distance from the projector source to the mirror contact point, and μ is the mirror radius. We must solve for (the smallest value of) l, which gives the quadratic equation,

$$\mu^2=|\boldsymbol{p}-\boldsymbol{m}+l\,U(\boldsymbol{x}-\boldsymbol{p})|^2 \tag{14}$$

$$=|\boldsymbol{p}-\boldsymbol{m}|^2+l^2+2l(\boldsymbol{p}-\boldsymbol{m})\cdot U(\boldsymbol{x}-\boldsymbol{p}) \tag{15}$$

which has the solution

$$l=-(\boldsymbol{p}-\boldsymbol{m})\cdot U(\boldsymbol{x}-\boldsymbol{p})\pm\sqrt{[(\boldsymbol{p}-\boldsymbol{m})\cdot U(\boldsymbol{x}-\boldsymbol{p})]^2-(|\boldsymbol{p}-\boldsymbol{m}|^2-\mu^2)} \tag{16}$$

Now that we have the distance to the mirror, and thus the target spot for the incident ray, $\boldsymbol{y}$, we can compute a normal vector for the mirror. This is the vector $\boldsymbol{y}-\boldsymbol{m}$ connecting the point of reflection $\boldsymbol{y}$ to the mirror center $\boldsymbol{m}$.

To find the reflection off the mirror, we consider the tangent plane at $\boldsymbol{y}$. In this case, consider an incident vector $\boldsymbol{u}$ and a unit normal vector $\boldsymbol{n}$ (Figure 1B). Then the outgoing vector is $\boldsymbol{k}=2\boldsymbol{n}(\boldsymbol{u}\cdot\boldsymbol{n})-\boldsymbol{u}$. Substituting the values for the normal and incident vector, we obtain

$$\boldsymbol{k}=2U(\boldsymbol{y}-\boldsymbol{m})\,[U(\boldsymbol{y}-\boldsymbol{m})\cdot(\boldsymbol{p}-\boldsymbol{y})]-(\boldsymbol{p}-\boldsymbol{y}) \tag{17}$$

We can find the target location $\boldsymbol{z}$ on the dome using the same geometric argument as for the mirror:

$$\boldsymbol{z}=\boldsymbol{y}+L\,U(\boldsymbol{k}) \tag{18}$$

where $L=|\boldsymbol{z}-\boldsymbol{y}|$ is the distance from mirror to dome, given by

$$L=-(\boldsymbol{y}-\boldsymbol{d})\cdot U(\boldsymbol{k})\pm\sqrt{[(\boldsymbol{y}-\boldsymbol{d})\cdot U(\boldsymbol{k})]^2-(|\boldsymbol{y}-\boldsymbol{d}|^2-R^2)} \tag{19}$$

This gives us the formulae that determine the point $\boldsymbol{z}$ where the projector hits the dome.

The perceived unit vector for this projection is then $\boldsymbol{\zeta}=U(\boldsymbol{z}-\boldsymbol{a})$. The outcome of all of this ray tracing is a mapping $h:\boldsymbol{t}\to\boldsymbol{\zeta}$.

1.5 Projector to Mirror

A cone of rays from the projector will hit the mirror. The half-angle is $\mu/2=\sin^{-1}r/|\boldsymbol{m}-\boldsymbol{p}|$. The cone axis is $\boldsymbol{m}-\boldsymbol{p}$. A set of points $\boldsymbol{\tau}$ are equidistant from $\boldsymbol{m}$ and $\boldsymbol{p}$, defined by

$$\boldsymbol{\tau}=\boldsymbol{\rho}+P\hat{\boldsymbol{\imath}} \tag{20}$$

where $\hat{\boldsymbol{\imath}}$ is any unit vector orthogonal to the cone axis $\boldsymbol{m}-\boldsymbol{p}$, and $P=\sqrt{|\boldsymbol{m}-\boldsymbol{p}|^2-r^2}\sin\mu/2$. The point $\boldsymbol{\rho}$ is located at

$$\boldsymbol{\rho}=\boldsymbol{p}+\frac{\boldsymbol{m}-\boldsymbol{p}}{|\boldsymbol{m}-\boldsymbol{p}|}(|\boldsymbol{m}-\boldsymbol{p}|-r\sin\mu/2) \tag{21}$$

$$=\boldsymbol{m}-\frac{\boldsymbol{m}-\boldsymbol{p}}{|\boldsymbol{m}-\boldsymbol{p}|}r\sin\mu/2 \tag{22}$$

We should only map points from the projector that both hit the mirror, and fall within the field of view of an imaginary perceptual camera that matches the virtual camera characteristics. We can find the points that hit the mirror by mapping the mirror's shadow onto the projector screen. First we find the unit vectors $\hat{\boldsymbol{x}}$ that correspond to this cone, according to

$$\hat{\boldsymbol{x}} = U[\boldsymbol{\rho} - \boldsymbol{p} + \gamma P(\cos\theta, \sin\theta) \cdot (\hat{\boldsymbol{k}_x}, \hat{\boldsymbol{k}_y})] \tag{23}$$

where $\gamma \in [0, 1)$ and $\theta \in [0, 2\pi)$ are polar coordinates parameterizing the mirror area, and $(\hat{\boldsymbol{k}_x}, \hat{\boldsymbol{k}_y})$ is a coordinate system orthogonal to the vector $\boldsymbol{m} - \boldsymbol{p}$ connecting the projector to the mirror center (which is not necessarily where the projector is pointing). This coordinate system is distinct from the other projector system centered on the vector $\boldsymbol{q} - \boldsymbol{p}$!

With this set of points, we can then project onto the $\boldsymbol{t}$ coordinates to obtain a set of points that could possibly map to the perceptual sphere, according to

$$\boldsymbol{t} = f(\hat{\boldsymbol{x}}) = T \cot\frac{\beta}{2} J \frac{\hat{\boldsymbol{x}}}{\hat{\boldsymbol{x}} \cdot \boldsymbol{\chi}} \tag{24}$$

where $\boldsymbol{\chi} = U[\boldsymbol{q} - \boldsymbol{p}]$ is the unit vector pointing along the projector axis and $J = (\hat{\boldsymbol{\jmath}}_x, \hat{\boldsymbol{\jmath}}_y)$ is a projection matrix that returns the coordinates orthogonal to $\boldsymbol{\chi}$.

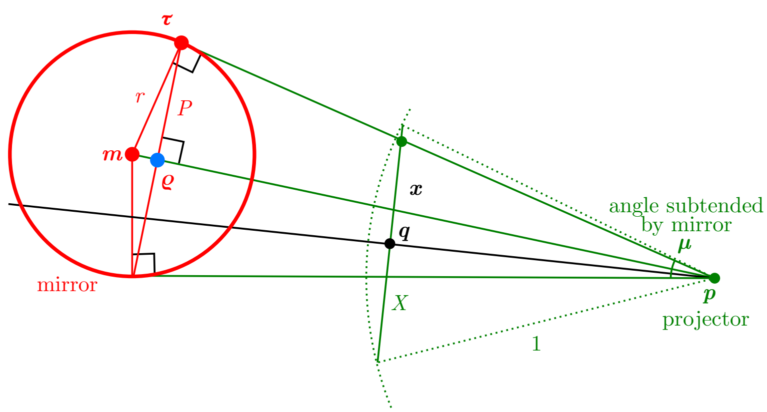

Figure 2: Not all of the rays from the projector may hit the mirror.

1.6 Distortion mapping

We have thus derived the mapping from projected $\boldsymbol{t}$ coordinates to the perceived screen coordinates $\boldsymbol{s}$, as shown in Figures 3,4. This successfully maps the virtual camera onto the perceptual camera (Figure 5).

1.7 Notes

Dont forget to change rendering intensity as well as mapping. Intensity should vary as a function of dome angle and distance!

1.8 References

- Lars Buesing and colleagues describe “Inferring neural population dynamics from multiple partial recordings of the same neural circuit” (NIPS).

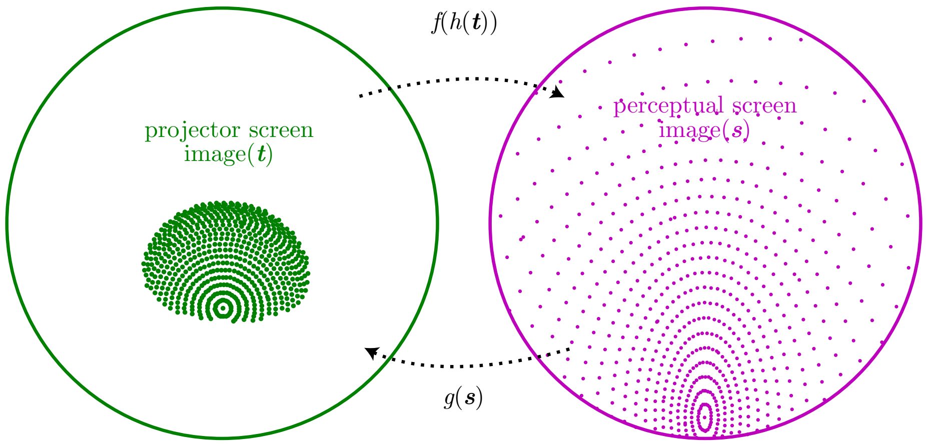

Figure 3: The points $\boldsymbol{t}$ map to the perceptual screen via $\boldsymbol{s} = f(h(\boldsymbol{t}))$. We want to invert this map to obtain $\boldsymbol{t} = g(\boldsymbol{s})$. We can do this by computing an interpolation function $\hat{g}$ based on the depicted sampled data $(\boldsymbol{s}_i, \boldsymbol{t}_i)$. Linear interpolation seems to be adequate. Due to the distortion, samples in $\boldsymbol{t}$ may be best chosen non-uniformly. Here I have chosen the set of radii according to $\{r\} = \tanh 2\mathcal{U}$ where $\mathcal{U}$ are uniformly sampled. I have also sampled more angles at greater radii.

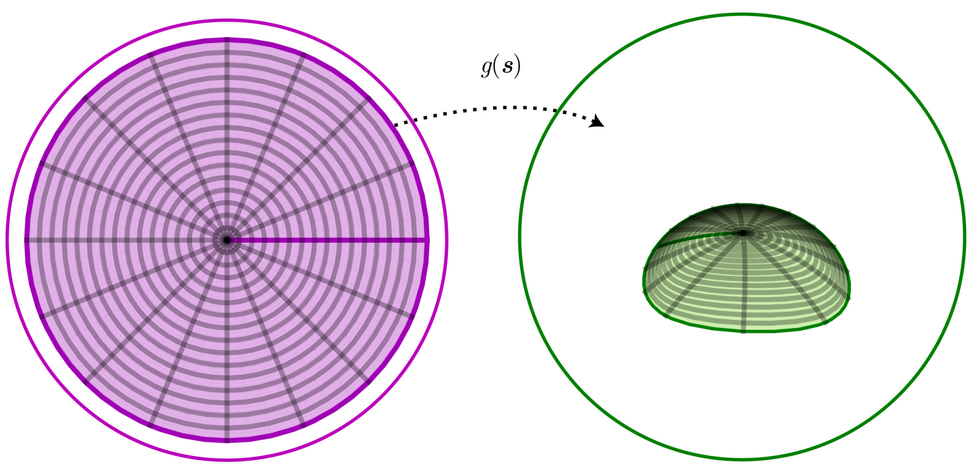

Figure 4: The result of the linear interpolation is a piecewise linear function – actually, two of them: $g_x(\boldsymbol{s})$ and $g_y(\boldsymbol{s})$.

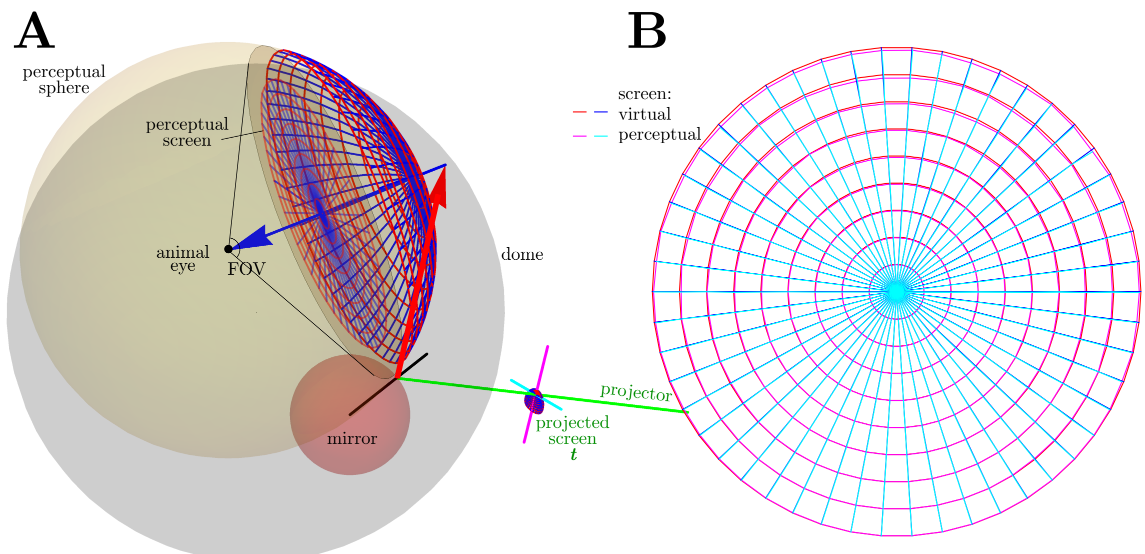

Figure 5: Using this interpolated map $\hat{g}$ we successfully map a radial coordinate system in the virtual screen coordinates $\boldsymbol{s}$ via the 3D geometry (**A**), to a radial coordinate system in the perceptual screen. It is hard to tell the difference between the two coordinates in (**B**) because they align so well.