Corrections for Snells Law

Mitchell Cobb

April 12, 2019

1 Introduction: Goals and Problem Statement

The goal of this document is to devise a mathematically rigorous way to account for refraction due to the air-glass-water interface in our calculation of Euler angles in the water channel. This document does not (yet) attempt to account for these effects in the calculation of position measurements.

2 Interpretation of data from cameras

This calculation will depend on two sources of information

- constant parameters input by the user, and
- dot locations within the image.

The second item, dot locations within the image, are encoded as the row and column within the image-matrix corresponding to the center of the dot. It's worth taking a minute to emphasize how this information is interpreted. A given set of dots is taken to represent a vector (not a unit vector) that is aligned with one of the primary body-fixed axes. Therefore, a photo of one of these dots sets is taken to represent the projection of the corresponding vector onto a plane. If we represent that plane as the span of two unit vectors, one aligned with the direction of increasing rows in the image, and the other aligned with increasing columns within the image, then the row and column separation of the dot centers can be interpreted as the projection (via dot product) of the vector in question onto those unit vectors that span the plane in question.

Thus, I am choosing to formulate this correction by first investigating how this projection plane (and corresponding unit vectors) can be propagates out from the camera lens, into the water channel, so that we can interpret the dot separation distances as scaled versions of the projection of the body-fixed vector onto these propagated unit vectors.

3 Problem Setup

This section addresses the generic problem of tracing a single ray or line of sight out from the camera, through the different media.

Assume that we're given the following:

- A camera coordinate system $\overline{C} = \{C, \vec{i}_{\overline{C}}, \vec{j}_{\overline{C}}, \vec{k}_{\overline{C}}\}$ located outside the water channel and pointed into it. So we know the position of the point C and expressions for the camera unit vectors (obtained from the camera rotation matrix, which is obtained from the camera Euler angles).

- The distance of the camera from the glass along the $\vec{k}_{\overline{C}}$ axis. Call this d.
- Two rotation angles ϕ and θ that describe the orientation of the camera with respect to the glass. Beginning with the camera $x-y$ plane parallel to the glass, we first rotate about $\vec{i}_{\overline{C}}$ by ϕ and then rotate about the new $\vec{j}_{\overline{C}}$ by θ to obtain the orientation of the camera relative to the glass. So in our specific case the slant camera would have $\phi = 0$ and $\theta =$ some increment of 45 degrees.

4 Ray tracing an arbitrary sight line

This section walks through the steps necessary to calculate the point where a sight line emerges inside the water channel, along with the direction in which it emerges. This generic ray tracing formulation will be used in later sections to calculate a plane which we will consider the body fixed vector to be projected onto.

4.1 Segment 1: From the camera to the glass

Note that the inputs from the user explicitly define one vector. Specifically, the vector from the origin of the camera coordinate system, along the center sight line of the camera, up to the surface of the glass. Define this point as CG for "center-glass". Then, in the camera coordinate system, we have

$$\{\vec{r}_{CG/C}\}_{\overline{C}} = \begin{bmatrix} 0 & 0 & -d \end{bmatrix}^T \tag{1}$$

Next, consider any other point/pixel in the image. This pixel corresponds to a sight line that we can obtain by rotating $-\vec{k}_{\overline{C}}$ through a sequence of two angles. First, rotate by γ_h about $\vec{i}_{\overline{C}}$ and then by γ_v about the new $\vec{j}_{\overline{C}}$. So the unit vector pointing in this "arbitrary" direction and striking the glass at the arbitrary point A is given by

$$\{\vec{u}_{A/C}\}_{\overline{C}} = R_y(\gamma_v) R_x(\gamma_h) \{\vec{r}_{CG/C}\}_{\overline{C}} \tag{2}$$

where $R_y(\gamma_v)$ and $R_x(\gamma_h)$ are 3×3 rotation matrices. So now the problem is to find the point where this sight line hits the glass, or the length of the vector $\vec{r}_{A/C}$. Note that the vectors $\vec{r}_{A/C}$, $\vec{r}_{CG/A}$, and $\vec{r}_{CG/C}$ form a triangle, so we know that

$$\vec{r}_{CG/C} = \vec{r}_{A/C} + \vec{r}_{CG/A} \tag{3}$$

Note that the point where this sight line hits the glass will lie in the plane with the point that we found before in 1. Let this plane be defined by a unit normal vector $\vec{n}_g$, which is located at the point CG. Therefore we know that the dot product of this normal vector and the vector from CG to A should be zero,

$$0 = \vec{r}_{CG/A} \cdot \vec{n}_g \tag{4}$$

$$0 = (\vec{r}_{CG/C} - \vec{r}_{A/C}) \cdot \vec{n}_g \tag{5}$$

$$0 = (\{\vec{r}_{CG/C}\}_{\overline{C}} - ||\vec{r}_{A/C}|| \{\vec{u}_{A/C}\}_{\overline{C}})^T \{\vec{n}_g\}_{\overline{C}} \tag{6}$$

$$\tag{7}$$

Solving this for the magnitude of the vector, $||\vec{r}_{A/C}||$, we get

$$||\vec{r}_{A/C}|| = \frac{\{\vec{r}_{CG/C}\}_{\overline{C}}^T \{\vec{n}_g\}_{\overline{C}}}{\{\vec{u}_{A/C}\}_{\overline{C}}^T \{\vec{n}_g\}_{\overline{C}}} \tag{8}$$

So now we have an expression for the vector that points from the origin of the coordinate system C along any arbitrary sight line in the frame, and ends at the point A on the glass, formed by combining equations 1, 2, and 8

$$\vec{r}_{A/C} = \frac{\{\vec{r}_{CG/C}\}_{\overline{C}}^{T}\{\vec{n}_g\}_{\overline{C}}}{\{\vec{u}_{A/C}\}_{\overline{C}}^{T}\{\vec{n}_g\}_{\overline{C}}} R_y(\gamma_v) R_x(\gamma_h) \begin{bmatrix} 0 \\ 0 \\ -d \end{bmatrix} \tag{9}$$

Note that the unit vector normal to the glass can be calculated by taking the $\vec{k}_{\overline{C}}$ unit vector and "undoing" the rotations by ϕ and θ that are input by the user. Specifically,

$$\{\vec{n}_g\}_{\overline{C}} = \big(R_y(\theta) R_x(\phi)\big)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{10}$$

To summarize, we now have expressions for the vector that points from C to an arbitrary point A and the vector that is normal to the glass $\vec{n}_g$.

4.2 Refraction with Snells law in 3D

Snells law for the planar case states that

$$\frac{\sin(\Gamma_1)}{\sin(\Gamma_2)} = \frac{n_1}{n_2}. \tag{11}$$

In the general 3D case, we have to be careful about how we define these angles Γ_1 and Γ_2. If $\vec{u}_{A/C}$ is a unit vector pointed at the glass, and $\vec{n}_g$ is a normal vector that points out from the glass, then the angle between them is given by

$$\Gamma_1 = \arccos\big(-\vec{u}_{A/C} \cdot \vec{n}_g\big) \tag{12}$$

where the negative sign results from the fact that $\vec{u}_{A/C}$ is pointed in the "opposite" direction from $\vec{n}_g$. We can then find Γ_2 by solving the equation above

$$\Gamma_2 = \arcsin\left(\frac{n_2}{n_1}\sin\big(\Gamma_1\big)\right) \tag{13}$$

The outgoing vector can then be formed by rotating the normal vector that points into the glass ($-\vec{n}_g$) about the appropriate axis by Γ_2. This axis of rotation is the axis perpendicular to both the outwards pointing unit vector and the incoming ray (really the negative of the incoming ray, which is a vector pointing back along $\vec{u}_{A/C}$, note that $\vec{u}_{C/A} = -\vec{u}_{A/C}$)

$$\vec{u}_{axis} = \vec{n}_g \times \vec{u}_{C/A} \tag{14}$$

The outgoing unit vector $\vec{u}_{out}$ is then formed by using Rodrigues' rotation formula, which rotates one vector about another, by a given angle. In this case, we want to rotate $-\vec{n}_g$ about $\vec{u}_{axis}$ by Γ_2, so we have

$$\vec{u}_{out} = \vec{u}_{C/A}\cos(\Gamma_2) + \big(\vec{u}_{axis} \times \vec{u}_{C/A}\big)\sin(\Gamma_2) + \vec{u}_{axis}\big(\vec{u}_{axis} \cdot \vec{u}_{C/A}\big)\big(1 - \cos(\Gamma_2)\big) \tag{15}$$

Note, I think there's a typo in that equation that propagates through the document, so correct that before implementing. Also, note that if the sight line that we're considering points straight into the glass, then this might break down, but in that case $\vec{u}_{out} = \vec{u}_{in}$ so we might just need some if-statement to check for that.

4.3 Segment 2: Inside the glass

So to summarize, given an incoming "ray" or sight line represented as a 3D vector, we can calculate the outgoing unit vector using these equations. So now, given the incoming ray $\vec{u}_{C/A}$ we predict that it points at the point B on the inside surface of the glass and lies along the line defined by the the point A and the unit vector $\vec{u}_{B/A}$ which is given by

$$\vec{u}_{B/A} = \vec{u}_{C/A}\cos(\Gamma_2) + \big(\vec{u}_{axis} \times \vec{u}_{C/A}\big)\sin(\Gamma_2) + \vec{u}_{axis}\big(\vec{u}_{axis} \cdot \vec{u}_{C/A}\big)\big(1 - \cos(\Gamma_2)\big) \tag{16}$$

To find the exact position of the point on the side glass, we need to find the length of this vector that points in the direction $\vec{u}_{B/A}$. If we know the thickness of the glass, T_g (which we do), then defining this vector as

$$\vec{v}_{B/A} = ||\vec{v}_{B/A}||\vec{u}_{B/A}, \tag{17}$$

we know that the component in the $-\vec{n}_g$ direction must have a magnitude of T_g. Therefore,

$$T_g = ||\vec{v}_{B/A}||\vec{u}_{B/A} \cdot -\vec{n}_g \implies ||\vec{v}_{B/A}|| = -\frac{T_g}{\vec{u}_{B/A} \cdot \vec{n}_g}. \tag{18}$$

Therefore

$$\vec{v}_{B/A} = -\frac{T_g}{\vec{u}_{B/A} \cdot \vec{n}_g}\vec{u}_{B/A}. \tag{19}$$

So now we have an expression for the point on the inside of the glass that our arbitrary sight line passes through.

4.4 Segment 3: In the water

From here, we can repeat the process of using Rodrigues' rotation formula to get the unit vector that describes the path of our sightline in the water, $\vec{u}_W$,

$$\vec{u}_W = \vec{u}_{B/A}\cos(\Gamma_2) + \big(\vec{u}_{axis} \times \vec{u}_{B/A}\big)\sin(\Gamma_2) + \vec{u}_{axis}\big(\vec{u}_{axis} \cdot \vec{u}_{B/A}\big)\big(1 - \cos(\Gamma_2)\big) \tag{20}$$

where in this case,

$$\vec{u}_{axis} = \vec{n}_g \times \vec{u}_{A/B} \tag{21}$$

5 Our specific case

So, how do we use this to calculate what we want, which is the length, in centimeters, of the components of the unit vectors?

First, I should be clear about how we are interpreting the image. I think that the correct interpretation of the image is that it represents a scaled version of the projection of the vector in question on to the plane that is perpendicular to a sight line that comes straight out from the center of the camera. Equivalently, we can imagine the sight line that results from ray-tracing the $\vec{k}_{\overline{C}}$ axis out, through the air, then the glass and into the water.

From here on I'll refer to this ray inside the water channel as the "centerline". So just to be clear, in this context, the centerline is defined by a point and a direction vector that points away from that point. This centerline is the ray that results in inside the water channel from the process above if we choose $\gamma_h = \gamma_v = 0$.

So then the horizontal and vertical separation between two dots that represent a body-fixed vector is considered to be the projection of that vector onto a plane perpendicular to this centerline. The next logical question is, *which* plane?

I think that we should choose the plane that intersects the last known position (from the last time step) of the dot set centroid (the halfway point between the two dots). This is analogous to using the distance from the camera to the dots from the last time step, which is what was done in the previous formulation. I think this should break an algebraic loop, just like in previous versions.

Another question is, how do we map pixels to centimeters? I think we can do that by considering the quadrilateral mapped out by the edges of the field of view. Details on that are provided in the subsequent sub-sections.

5.1 Finding the correct plane of projection

This section addresses how to find the plane onto which we will consider our vector to be projected. Start by assuming that we've implemented the process in section 4 and found our centerline, which will be defined by

- the point at the end of the vector $\vec{r}_C$ which points from the origin of the ground coordinate system to the point on the inside of the glass where the centerline emerges from the glass and enters the water, and
- the unit vector $\vec{u}_c$ which is a unit vector pointing in the direction the direction of this centerline. Note that this vector defines the normal to our projection plane.

So in order to find the correct projection plane, we need to choose a scalar variable α which multiplies our unit vector, $\vec{u}_C$, such that the previous known position of the dots in question, $\vec{r}^{dots}_{last}$ lies in the projection plane. Mathematically, the condition that we want is that:

$$0 = (\vec{r}^{dots}_{last} - \vec{r}_C - \alpha\vec{u}_c) \cdot \vec{u}_c. \tag{22}$$

Here, the first term in the parentheses is the vector that points from and arbitrary point along our centerline, defined by α, to last known position of the dots. Solving this equation for α gives

$$\alpha = (\vec{r}^{dots}_{last} - \vec{r}_C) \cdot \vec{u}_C. \tag{23}$$

Therefore, our projection plane has the normal vector $\vec{u}_C$ and passes through the point

$$\vec{r}^{C}_{P} \triangleq \vec{r}_C + \alpha\vec{u}_C = \vec{r}_C + \big((\vec{r}^{dots}_{last} - \vec{r}_C) \cdot \vec{u}_C\big)\vec{u}_C \tag{24}$$

5.2 Calculating component lengths

Start by considering the "edges" of the field of view. We can model the way in which this field of vision changes as it propagates through the different media by ray-tracing the top-left, top-right, bottom-left, and bottom-right corners of the field of view. The result is a set of 4 points and a set of 4 unit vectors that represent the outer edges of the field of view inside the water channel. As these vectors propagate through the water, they will hit the plane that we calculated in the previous section. So the result will be a set of 4 co-planar points that define a quadrilateral. We can use this quadrilateral for two purposes:

- first, we can use the lengths of the sides of that quadrilateral to determine the pixels-to-cm conversion factors, and

- second, we can use it to determine the unit vectors onto which the vector in question is considered to be projected onto.

First, we will consider the problem of how to find the set of 4 points that define the edges of this quadrilateral then the next two subsections will spell out processes for the bullet points above.

5.2.1 Finding The Edges

Begin by assuming that we have found

- the set of four unit vectors $\vec{u}_{TL}$, $\vec{u}_{TR}$, $\vec{u}_{BL}$, and $\vec{u}_{TR}$ that correspond to the sight lines inside the water channel that align with the top-left, top-right, bottom-left, and bottom-right corners of the field of view as well as

- the set of four points that lie at the end of the vectors $\vec{r}_{TL}$, $\vec{r}_{TR}$, $\vec{r}_{BL}$, and $\vec{r}_{TR}$ that point from the origin of the ground fixed system to the point where the the top-left, top-right, bottom-left, and bottom-right sight lines emerge from the glass and enter the water.

then if we (kind of awkwardly) define a set of indices $\mathbf{P} = \{TL, TR, BL, BR\}$, where $i \in \mathbf{P}$, we can write a generic expression that describes the solution to the vector-intersects-plane problem that we are considering

$$0 = (\vec{r}_i + \alpha_i \vec{u}_i - \vec{r}_P^C) \cdot \vec{u}_C \tag{25}$$

solving this for α_i like before gives

$$\alpha_i = \frac{(\vec{r}_P^C - \vec{r}_i) \cdot \vec{u}_C}{\vec{u}_i \cdot \vec{u}_C} \tag{26}$$

So then the generic expression for a point lying in our plane is

$$\vec{r}_i^P \triangleq \vec{r}_i + \alpha_i \vec{u}_i = \vec{r}_i + \frac{(\vec{r}_P^C - \vec{r}_i) \cdot \vec{u}_C}{\vec{u}_i \cdot \vec{u}_C} \vec{u}_i \tag{27}$$

So by calculating this point for every element of the set $\mathbf{P}$ we can get a set of 4 co-planar points, $\mathbf{R}_P = \{\vec{r}_{TL}^P, \vec{r}_{BL}^P, \vec{r}_{TR}^P, \vec{r}_{BR}^P\}$, which define a quadrilateral.

5.2.2 Pixel-To-Inch Conversion

Begin by calculating the length of each side of the quadrilateral,

$$l_{top} = ||\vec{r}_{TL}^P - \vec{r}_{TR}^P||, \tag{28}$$

$$l_{rgt} = ||\vec{r}_{TR}^P - \vec{r}_{BR}^P||, \tag{29}$$

$$l_{bot} = ||\vec{r}_{BR}^P - \vec{r}_{BL}^P||, \tag{30}$$

$$l_{lft} = ||\vec{r}_{BL}^P - \vec{r}_{TL}^P||. \tag{31}$$

Now, suppose that the center of the dot set appears at row i and column j in the image matrix, and the total image height in pixels is H while the total image width is W, furthermore, assume that the separation distance between the two dots in pixels is $\Delta_p x$ and $\Delta_p y$ respectively. We can then calculate the dot separation distance in units of length, $\Delta_l x$ and $\Delta_l y$ as a percentage of the linearly weighted average of the side lengths,

$$\Delta_l x = \left(\frac{j}{W} l_{lft} + \left(1 - \frac{j}{W}\right) l_{rgt} \right) \frac{\Delta_p x}{W}, \tag{32}$$

$$\Delta_l y = \left(\frac{i}{H} l_{top} + \left(1 - \frac{i}{H}\right) l_{bot} \right) \frac{\Delta_p y}{H}. \tag{33}$$

5.2.3 Finding The Appropriate Unit Vectors

NOTE, I think this is correct, but I also think I've mixed up some x's and y's in the notation which could confuse things. I should fix that.

After finding the lengths in the previous section, we need to consider the question: for which unit vectors are these considered to be the projections? That is the subject of this section.

I think that the best answer can be understood by considering the following explanation. For a given position of dots, there is a sight line that intersects the projection plane and passes through the dot set centroid (average position). In the image then, there is a specific pixel row and column corresponding to the current dot position. If we take that pixel position and increase the row by 1, then we will get a new sight line. This new sight line will intersect the projection plane slightly lower (from the cameras perspective). The vector that points from the original intersection point to this new intersection point is then the unit vector corresponding to $\Delta_l p$. Similarly, if we take the original pixel location of the dot set centroid and increase the column by 1 pixel, then we can obtain the unit vector corresponding to $\Delta_l x$.

Although this is probably a pretty rigorous way to go about finding the correct unit vectors, I'm going to propose a slightly different solution that I think will be more computationally lightweight (and possibly equivalent). Specifically, I propose that we use the set of corner points from before along with a similar weighting scheme to eqns 32, 33 to determine the end points of a vector that aligns with the correct unit vectors.

First, let's find the unit vector corresponding to $\Delta_l x$. The idea is to find the point $\frac{\Delta_p y}{H}$ down the left edge of the quadrilateral, then connect it with a vector $\frac{\Delta_p y}{H}$ down the right edge of the quadrilateral and normalize the resulting vector. Because I'm running out of variables, I'm going to define the function $N(\cdot) : \mathbb{R}^3 \to \mathbb{R}^3$ to be the normalization function that takes in a vector and returns the normalized vector. So then the unit vector that corresponds to $\Delta_l x$, $\vec{i}_x$ would be

$$\vec{i} = N\left((\vec{r}^P_{TR} + \frac{\Delta_p y}{H} l_{rgt}(\vec{r}_{BR} - \vec{r}_{TR}) - (\vec{r}^P_{TL} + \frac{\Delta_p y}{H} l_{rgt}(\vec{r}_{BL} - \vec{r}_{TL}) \right) \tag{34}$$

there is an analogous expression for the unit vector corresponding to $\Delta_l y$.

6 Conclusions

In summary, this document has detailed a process by which we can calculate two things

1. vectors represented in the ground fixed frame, and
2. projections of the vector being tracked onto those unit vectors

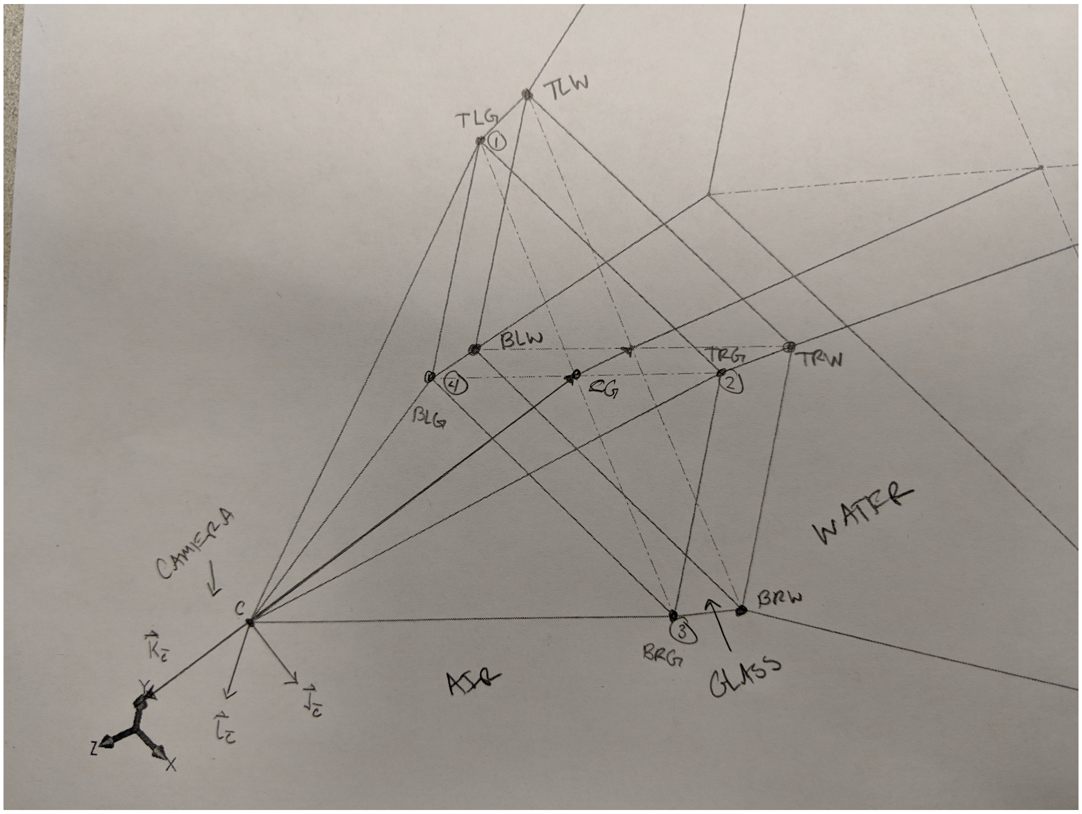

Figure 1: Diagram showing how various sight lines propagate out through the various interfaces.