Inferring 3D object coordinates from multiple camera views

Michael G. Moore

May 29, 2019

1 Introduction

1.1 Notation

We will use capitalized vectors, e.g. $\vec{V}$, for ordinary vectors, and lower-case vectors, e.g. $\vec{v}$, to indicate unit-length vectors.

1.2 Cameras

A camera is defined by its location vector $\vec{C}$ in lab coordinates; its pixel dimensions, (N_x, N_y); its horizontal angle-of-view, α (https://en.wikipedia.org/wiki/Angle_of_view); and its orientation vectors $\vec{e_1}$, $\vec{e_2}$, and $\vec{e_3}$, which describe the camera's optical axis, horizontal axis, and vertical axis, respectively. Together $\vec{e_1}$, $\vec{e_2}$, and $\vec{e_3}$ must form a right-handed triplet.

1.3 Objects

An object is defined by its position vector $\vec{R}$ in lab coordinates. It will be imaged by a camera onto a point (X, Y) in the camera image. A single $\vec{R}$ will map to different (X, Y) for different cameras. The exact mapping from $\vec{R} \rightarrow (X, Y)$ is somewhat complex and will be described in the following sections. The primary goal of this algorithm is to infer an objects $\vec{R}$ from its (X, Y) image coordinates from multiple camera views, just like in binocular vision.

For simplicity, we assume the image coordinates, (X, Y), are shifted and equally-scaled, so that $X \in [-1, 1]$ and $Y \in [-N_y/N_x, N_y/N_x]$. If we indicate by (X_{raw}, Y_{raw}), the raw pixel data, with X_{raw} running left-to-right from 0 to N_x, and Y_{raw} running top-to-bottom from 0 to Ny, then the transformation from

(X_{raw}, Y_{raw}) to (X, Y) is given by

$$X = \frac{2}{N_x} X_{raw} - 1 \tag{1}$$

$$Y = \frac{2}{N_x} Y_{raw} - \frac{N_y}{N_x}, \tag{2}$$

with the inverse transformation given by

$$X_{raw} = \frac{N_x}{2}(X + 1) \tag{3}$$

$$Y_{raw} = \frac{N_x}{2}\left(Y + \frac{N_y}{N_x}\right). \tag{4}$$

2 Projecting an object onto the FOV

A camera can be defined by its location vector, $\vec{C}$, taken as the location in lab-coordinates of the center of the objective lens; and two additional unit vectors, $\vec{a}$, and $\vec{p}$, which define the camera orientation and angle-of-view. From $\vec{a}$ and $\vec{p}$ we can construct the camera-frame unit vectors $\vec{e}_1$, $\vec{e}_2$, and $\vec{e}_3$, as follows.

The unit vector $\vec{a}$ points along the optical axis of the camera. If we imagine a ray starting from the camera location, $\vec{C}$ and extending in the direction of $\vec{a}$, then the first object it encounters will be imaged at the center of the camera's field of view (FOV), which we will indicate as $(X, Y) = (0, 0)$. The second unit vector, $\vec{p}$, is a reference pixel vector. An object intersected by the line from $\vec{C}$ along $\vec{p}$ will be imaged at the far-right of the camera's FOV, which we will indicate as $(X, Y) = (1, 0)$.

Twice the angle between $\vec{a}$ and $\vec{p}$ is the horizontal angle-of-view of the camera, α, so that

$$\cos(\alpha/2) = \vec{p} \cdot \vec{a}. \tag{5}$$

From the two unit-vectors, a and p, we can define a right-handed orthonormal triplet

$$\vec{e}_1 = \vec{a}, \tag{6}$$

$$\vec{e}_2 = \frac{\vec{p} - \cos(\alpha/2)\vec{a}}{\sin(\alpha/2)}, \tag{7}$$

$$\vec{e}_3 = \frac{\vec{a} \times \vec{p}}{\sin(\alpha/2)}, \tag{8}$$

which can be taken as the unit vectors of the camera frame-of-reference.

Consider an object described by vector $\vec{R}$. How can we determine the coordinates (X, Y) where this object appears in a camera's FOV?

First, we note that it only depends on the unit-vector

$$\vec{u} = \frac{\vec{R} - \vec{C}}{|\vec{R} - \vec{C}|}, \tag{9}$$

as any object along a given ray from the center of the camera objective lens maps onto the same pixel. This is a general property of thin lens design, and is not based on the assumption of a "pinhole camera".

If we introduce polar coordinates via

$$X = \rho \cos\phi \tag{10}$$

$$Y = \rho \sin\phi, \tag{11}$$

then the radius is given by (see Fig. 1)

$$\rho = \frac{\tan(\theta)}{\tan(\alpha/2)}, \tag{12}$$

where $\theta \in [0, \pi/2]$ is the angle of the object-vector relative to the optical axis, satisfying

$$\cos\theta = \vec{u} \cdot \vec{e}_1. \tag{13}$$

To find the angle, ϕ, relative to the X-axis, we need to consider the projections of p and u orthogonal to $\vec{a}$,

$$\vec{P} = \vec{p} - \cos(\alpha/2)\vec{a} = \sin(\alpha/2)\vec{e}_2, \tag{14}$$

$$\vec{U} = \vec{u} - \cos(\theta)\vec{a}. \tag{15}$$

We then have

$$\cos\phi = \frac{\vec{U} \cdot \vec{P}}{\|\vec{U}\|\|\vec{P}\|} = \frac{\vec{u} \cdot \vec{e}_2}{\sin(\theta)}. \tag{16}$$

To find $\sin\phi$, rather than use $\sin\phi = \sqrt{1 - \cos^2\phi}$, which won't tell us the correct sign (quadrant), we can

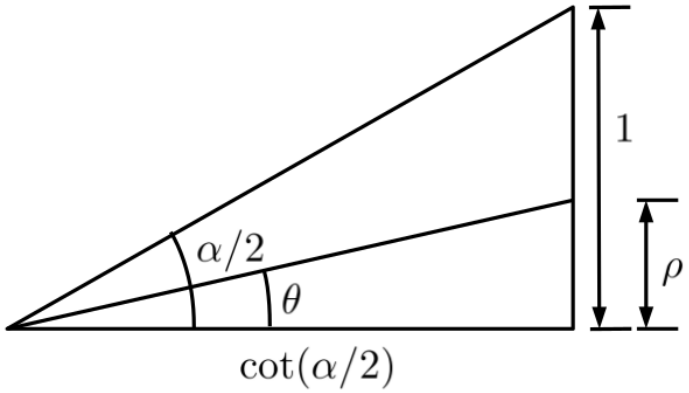

Figure 1: Determining the radius of the circle containing the image pixel. Here α is the angle-of-view of the camera, and θ is the angle of the object vector relative to the optical axis.

instead compute

$$\sin\phi = \frac{\vec{a}\cdot(\vec{P}\times\vec{U})}{\|\vec{U}\|\|\vec{P}\|} = \frac{\vec{u}\cdot\vec{e}_3}{\sin(\theta)}. \tag{17}$$

3 Inferring the object vector $\vec{u}$ from known X and Y

If we know everything about the camera, $(\vec{C},\vec{a},\vec{p})$, and we know X and Y for an object in the FOV of the camera, can we infer the object unit vector $\vec{u}$?

If X and Y are known, then we compute $\rho = \sqrt{X^2+Y^2}$, so that

$$\theta = \arctan\left(\sqrt{X^2+Y^2}\tan(\alpha/2)\right). \tag{18}$$

Next we compute ϕ from the four-quadrant arctan2 function,

$$\phi = \text{arctan2}(Y,X), \tag{19}$$

which allows us to construct $\vec{u}$ according to

$$\vec{u} = \vec{e}_1\cos\theta + \vec{e}_2\sin\theta\cos\phi + \vec{e}_3\sin\theta\sin\phi. \tag{20}$$

4 Learning camera orientation from training objects

In this section, we consider that $\vec{C}$ is known, but that the orientation vectors $\vec{e}_1$, $\vec{e}_2$, and $\vec{e}_3$ are unknown. Also known are the lab-frame coordinates of two reference objects, $\vec{R}_1$ and $\vec{R}_2$, and their image coordinates (X_1, Y_1) and (X_2, Y_2).

From (X_j, Y_j), where $j \in \{1, 2\}$, we can use the relations

$$\vec{u}_j = \frac{\vec{R}_j - \vec{C}}{\|\vec{R}_j - \vec{C}\|}, \tag{21}$$

$$\theta_j = \arctan\left(\sqrt{X_j^2 + Y_j^2}\tan(\alpha/2)\right), \tag{22}$$

$$\phi_j = \text{arctan2}(Y_j, X_j), \tag{23}$$

to write the following equations for the unknown camera orientation unit vectors

$$\vec{u}_1 = \vec{e}_1\cos\theta_1 + \vec{e}_2\sin\theta_1\cos\phi_1 + \vec{e}_3\sin\theta_1\sin\phi_1, \tag{24}$$

$$\vec{u}_2 = \vec{e}_1\cos\theta_2 + \vec{e}_2\sin\theta_2\cos\phi_2 + \vec{e}_3\sin\theta_2\sin\phi_2. \tag{25}$$

To find the optical axis, $\vec{e}_1$, we can solve the 3 equations

$$\vec{u}_1 \cdot \vec{e}_1 = \cos\theta_1, \tag{26}$$

$$\vec{u}_2 \cdot \vec{e}_1 = \cos\theta_2, \tag{27}$$

$$(\vec{u}_1 \times \vec{u}_2) \cdot \vec{e}_1 = \sin\theta_1\sin\theta_2\left(\cos\phi_1\sin\phi_2 - \sin\phi_1\cos\phi_2\right), \tag{28}$$

for the 3 components of $\vec{e}_1$ in lab coordinates. To find $\vec{e}_2$ we then solve

$$\vec{u}_1 \cdot \vec{e}_2 = \sin\theta_1\cos\phi_1, \tag{29}$$

$$\vec{u}_2 \cdot \vec{e}_2 = \sin\theta_2\cos\phi_2, \tag{30}$$

$$\vec{e}_1 \cdot \vec{e}_2 = 0. \tag{31}$$

An lastly, we can find $\vec{e}_3$ via

$$\vec{e}_3 = \vec{e}_1 \times \vec{e}_2. \tag{32}$$

5 Triangulation of object positions from multiple cameras

When the same object is viewed by two cameras, the object vector, $\vec{u}$, can be constructed for each camera. We then draw a line from each camera position, pointing along the object vector. If everything is properly aligned, the lines from all cameras will intersect at a single point. This will be the 3D location of the object. Algorithms to find the point of closest approach of two lines exist, and in the case of multiple cameras with data imperfections, the average of the triangulation points for all pairs of cameras can be used as a best estimate of the object location.