USING UNDERGRADUATE MATHEMATICS AND CONSUMER LEVEL CAMERAS TO TRACK THE FLIGHT OF A "TOY" HELICOPTER IN THREE DIMENSIONS

John Bacon
United States Military Academy
601 Thayer Road
West Point, NY 10966
john.bacon@usma.edu

Johann Thiel
United States Military Academy
601 Thayer Road
West Point, NY 10966
johann.thiel@usma.edu

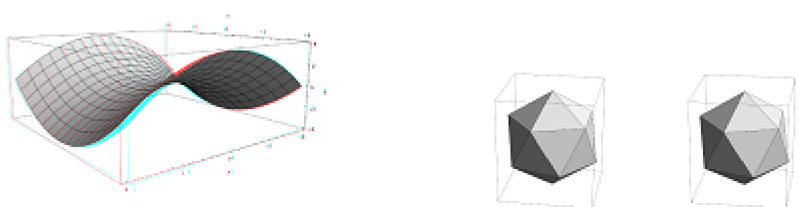
Victor Trujillo II
United States Military Academy
601 Thayer Road
West Point, NY 10966
victor.trujillo@usma.edu

Frank Wattenberg
United States Military Academy
601 Thayer Road
West Point, NY 10966
frank.wattenberg@usma.edu

1 Introduction

The tracking of objects traveling in $\mathbb{R}^3$ has several applications. Government agencies such as the Missile Defence Agency's Aegis Ballistic Missile Defense System have developed anti-ballistic missile weapons to defend against projectiles[1]. NASA's Near-Earth Object Program tracks the location of some objects in space that may pose a threat to earth[2]. Even television stations, like FOX and NBC, have used technology to track the position of a puck during hockey games[3,4]. Our aim here is to describe a method that allows one to track and recover an object's position in $\mathbb{R}^3$ using 2D photographic images.

There are several different ways by which 3-dimensional information can be conveyed and recovered via flat, 2D images. For example, anaglyphs combine two images of a single object taken from slightly different perspectives using different colors (see Figure 1a). By using properly oriented "3D glasses", the images are filtered and processed independently by each eye. The combination of the images in the brain gives rise to a 3-dimensional effect.



(a) Hyperboloid anaglyph (red-cyan) (b) Icosahedron stereoscopic image

Figure 1: Examples of flat "3D images"

A different, but similar, method of producing 3D images involves simply placing the two images of an object side by side. The observer is then forced to align his vision (usually in the form of being "cross-eyed," merging the two images into a third 3D image) properly to achieve the desired effect (see Figure 1b).

Both of the examples mentioned above have a common feature. To achieve a 3-dimensional effect, they required the use of two images taken of the same object from slightly different perspectives. If one were interested in tracking the path of an object traveling in $\mathbb{R}^3$, this might suggest that one must photograph the object in flight from two different positions *simultaneously*. The tracking method outlined in this paper

[1] http://www.mda.mil/system/aegis_bmd.html

[2] http://neo.jpl.nasa.gov/

[3] http://dl.acm.org/citation.cfm?id=618415

[4] http://usatoday30.usatoday.com/sports/columnist/mccarthy/2008-05-11-McCarthy_N.htm

(a) Michie Stadium (b) Parrot drone

Figure 2: Setup items

shows that this is not always necessary. By using the sun and a football field, we can recover 3-dimensional data from a *single* image.

2 Setup

We begin with a list of the necessary items:

- a standard, commercially available digital camera with a tripod,
- a football field,
- Mathematica (Excel or some other CAS can also be used), and
- a flying object to track.

We used Michie Stadium (see Figure 2a) in West Point, NY as our setting. Ideally, the camera should be placed on some bleachers overlooking the field. The reason for this is that the lines on the field will serve as reference points to be used in our computations.

In our particular setup, for our flying object we utilized a Parrot quad rotor drone[5] (see Figure 2b). An exact duplicate of the Parrot drone is not necessary for this activity. The only major requirement of the object to be tracked is that it should be large enough to be photographed from a distance.

[5] `http://www.brookstone.com/parrot-ar-drone-2-quadricopter`

3 Formulas

In this section, we develop the formulas needed to convert between the 2-dimensional coordinates of an object in a photographic image and its 3-dimensional position. Note that in the process, we make several assumptions about idealized cameras and rays of light. The subsequent section will give an example application of this process from beginning to end.

Suppose that an idealized camera is is placed at a point $\vec{q}$ in $\mathbb{R}^3$ facing an object (see Figure 3). By an idealized camera, we mean the following:

- the camera occupies a single point in space,
- there is no length to the camera's lens, and
- light rays striking the camera sensor are not distorted or bent in any way, i.e., no pinhole or barrel distortion.

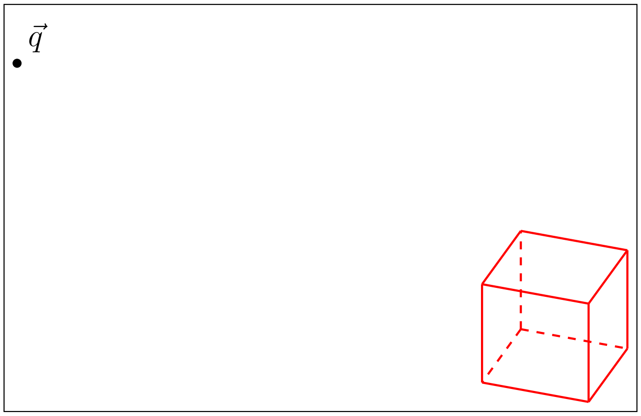


Figure 3: Initial camera position

After photographing the object, suppose that a semitransparent copy of the photograph is held at the right distance and tilt so that the image and the background are indistinguishable from the camera's perspective (see Figure 4). The vector $\vec{p}$ corresponds to the direction and distance from $\vec{q}$ to the center of the photograph.

Before proceeding, we need to choose an orientation for the plane corresponding to the photograph. This is done by selecting two perpendicular unit vectors, $\vec{u}$ and $\vec{v}$, in the plane of the photograph so that $\vec{u} \times \vec{v}$ points in the direction of $-\vec{p}$ (see Figure 4). These vectors describe the tilt about $\vec{p}$ when the photograph was taken and they serve as the basis of a 2-dimensional coordinate system for the photographic image.

The vectors $\vec{q}$, $\vec{p}$, $\vec{u}$ and $\vec{v}$ are known as the **camera vectors**. It may sound counterintuitive at first, but only seven distinct parameters are needed to completely describe all four camera vectors. Both $\vec{q}$ and $\vec{p}$ require three parameters each. Once these

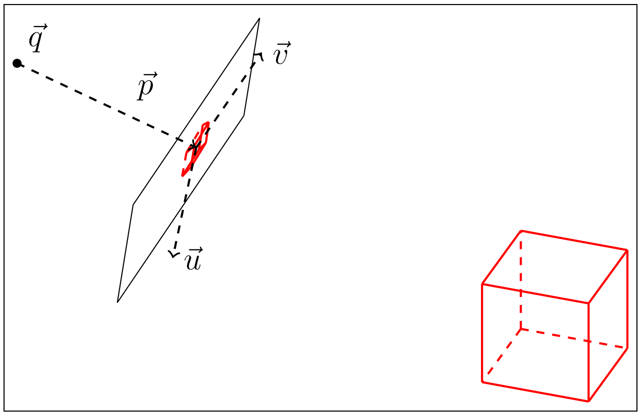


Figure 4: Semitransparent image with camera vectors

two vectors are known, since $\vec{u}$ is a unit vector restricted to the plane of the photographic image, it is completely determined by a single additional parameter. It follows immediately from the above discussion that $\vec{v}$ is completely determined by $\vec{u}$.

Once the camera vectors are known, we are ready to convert between 3-dimensional positions in $\mathbb{R}^3$ and a 2-dimensional coordinate system in the image with basis $\vec{u}$ and $\vec{v}$.

3.1 3D to 2D

Let $\vec{a}$ be the 3-dimensional position of an object of interest (see Figure 5). By assumption, there is a light ray that travels in a straight line between $\vec{a}$ and $\vec{q}$ that is parallel to

$$\vec{m} = \vec{a} - \vec{q}. \tag{1}$$

This light ray passes through the plane of the photographic image.

Let $\vec{h}$ be the vector projection of $\vec{m}$ onto $\vec{p}$ (see Figure 5). That is,

$$\vec{h} = \left(\frac{\vec{m} \cdot \vec{p}}{||\vec{p}||^2}\right)\vec{p}. \tag{2}$$

It follows that the vector $\vec{m} - \vec{h}$ is perpendicular to $\vec{h}$ and forms a right triangle with $\vec{m}$ playing the role of the hypothenuse. Using similar triangles, we see that

$$\vec{w} = \frac{||\vec{p}||}{||\vec{h}||}(\vec{m} - \vec{h}) \tag{3}$$

is parallel to $\vec{m} - \vec{h}$ and it is contained in the plane of the photographic image (see Figure 5). In particular, it points from the center of the photographic image to the intersection point of the plane and the line segment traced out by $\vec{m}$.

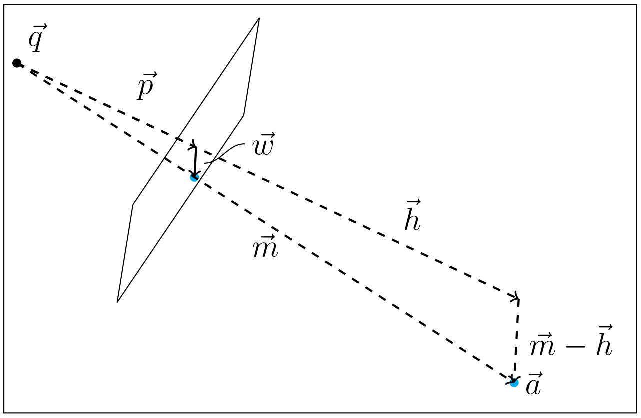


Figure 5: Projection of $\vec{a}$ onto the photographic image

To complete the conversion from 3D to 2D, we need to decompose $\vec{w}$ as a linear combination of the vectors $\vec{u}$ and $\vec{v}$. Since $\vec{u}$ and $\vec{v}$ are unit vectors, we must simply compute the dot products

$$s = \vec{w} \cdot \vec{u} \text{ and } t = \vec{w} \cdot \vec{v}. \tag{4}$$

We can now use the point (s, t) as our 2-dimensional coordinate for the projection of $\vec{a}$ onto the photographic image.

3.2 2D to 3D

In this situation, we assume that we have the 2-dimensional coordinates (s, t) from the photographic image as in (4). Our goal is to reconstruct the position vector $\vec{a}$ from the previous section.

Using the 2-dimensional coordinates (s, t), we see that the location of the point on the photographic image in $\mathbb{R}^3$ is given by $\vec{q}+\vec{p}+s\vec{u}+t\vec{v}$. This means that $\vec{a}$ must lie on the line

$$L(r) = \vec{q} + r(\vec{p} + s\vec{u} + t\vec{v}) \tag{5}$$

where $r \in \mathbb{R}$. Unfortunately, this is not enough information to determine $\vec{a}$, as there are an infinite number of choices for the value r.

There are a few ways to overcome the difficulty mentioned above. If two images were made of the same object from different perspectives, then one could find $\vec{a}$ by determining the intersection of two nonparallel lines created by applying (5) to two different sets of camera vectors and 2-dimensional photographic image coordinates. Alternatively, if the height of the object at $\vec{a}$ is known, then the value of r could be computed by setting the z-component of (5) equal to the known height of the object (assuming that $L(r)$ is not parallel to the xy-plane).

The intent here is to avoid having to use two cameras and to assume nothing about

Figure 6: Initial image of the football field

the position of the object. In the next section, we show how we can overcome the difficulty of only having a single photographic image by using the sun.

4 Implementation

Figure 6 shows an example of a typical image containing the Parrot drone in flight. (Note that the red circle in the image has been inserted to highlight the location of the Parrot drone on the lower right portion of the image.) We would like to find the 3-dimensional location of the drone given nothing more than this image. To do so, we must first find the camera vectors associated with this image so that we can compute (5) using (1), (2), (3), and (4).

In practice, it can be difficult to know the camera vectors $\vec{q}$, $\vec{p}$, $\vec{u}$, and $\vec{v}$ before a photograph is taken. Careful measurements can be made after fixing the camera's location in $\mathbb{R}^3$, but this is not necessary. One way to determine the camera vectors is to photograph a background with known reference points whose positions in $\mathbb{R}^3$ can be easily determined. In our case, we used a football field to accomplish this.

First we choose a 3-dimensional coordinate system to describe the position of any photographed object in $\mathbb{R}^3$. We took the origin to be at the center of the football field at a height that is in the horizontal plane intersecting the sidelines of the field. It is not obvious from Figure 6, but the football field is "crowned," meaning that the center of the field is raised to allow for better water drainage. Therefore, our origin actually lies below the center of the field of play. Figure 7a shows an overlay of our choice of coordinate system on the football field in $\mathbb{R}^3$. The positive x-direction is

(a) 3D coordinate system (b) 2D coordinate system

Figure 7: Initial image with different coordinate systems

taken to go from left to right, the positive y-direction is taken to go away from the camera, and the positive z-direction is taken to go upwards towards the sky.

Next we chose a 2-dimensional coordinate system that is specific to the image from Figure 6. Figure 7b shows an overlay of our choice of coordinate system on the photographic image. In this case the origin is simply located at the center of the image. The positive x-direction is taken to go from left to right and the positive y-direction is taken to go from the bottom to the top of the image.

We are now ready to begin the process of **calibrating** the image, that is, determining the camera vectors for that particular image. To do this we select some points on the image whose coordinates can be easily found in both the 2D and 3D coordinate systems selected. For example, since a regulation college football field is 160 feet across, the 50 yard sideline on the far side of the field in Figure 6 is located at the point $(0, 80, 0)$ (units are in feet). In the photographic image, the same point is located at the point $(-51, 347)$ (units are in pixels). We can combine these two coordinates into the single 5-tuple $(-51, 347, 0, 80, 0)$.

Repeating this process multiple times gives us a list of n 5-tuples

$$(s_1, t_1, x_1, y_1, z_1), \ldots, (s_n, t_n, x_n, y_n, z_n).$$

The objective is then to find the seven parameters determining the camera vectors that best approximate (s_i, t_i) given (x_i, y_i, z_i) for $i = 1, \ldots, n$. More precisely, we want to find the seven parameters determining the camera vectors that minimize the sum of squared differences

$$\mathcal{E} = \sum_{i=1}^{n} (\hat{s}_i - s_i)^2 + \sum_{i=1}^{n} (\hat{t}_i - t_i)^2$$

where $(\hat{s}_i, \hat{t}_i)$ is the *predicted* value of (s_i, t_i) using (4) when given (x_i, y_i, t_i) for $i = 1, \ldots, n$. By importing such 5-tuples into Mathematica, this can be accomplished using the **FindMinimum** function.

Having estimated the camera vectors, we can now use the 2D coordinate of the Parrot drone in Figure 6, along with (5), to create a 3-dimensional line $L_1(r)$ that

Figure 8: Shadows cast by a goal post and the Parrot drone

passes through the camera's location $\vec{q}$ and the Parrot drone's position in $\mathbb{R}^3$. As was mentioned in Section 3.2, knowing the formula for $L_1(r)$ is not enough to determine the drone's 3-dimensional position.

Figure 8 shows an image of a goal post taken at about the same time as Figure 6. The shadows in the image all roughly form similar triangles with the object that produced them because all of the sun rays striking the the field are essentially parallel. On large scales this is not true, but since we are working over a fairly localized area, we assume that all sun rays in the image are parallel.

We can now use the known dimensions of the goal post (these are not all standard) and the location of the of the goal post's shadow to find a 3-dimensional vector $\vec{f}$ that is parallel to the sun rays in the image. The reader might wonder how the location of the mentioned shadow can be found in $\mathbb{R}^3$. This goes back to the discussion in Section 3.2 which states that if the height of the desired object is known, 0 feet in the case of a shadow on the field, then the 2D to 3D conversion can be determined using only a single line. If we let q_3, p_3, u_3, and v_3 stand for the z-components of the camera vectors $\vec{q}$, $\vec{p}$, $\vec{u}$, and $\vec{v}$, respectively, then it follows that when (5) is evaluated at

$$r = \frac{q_3}{p_3 + su_3 + tv_3},$$

we obtain the position of a shadow on the field in $\mathbb{R}^3$. From this we can compute the vector $\vec{f}$ as the difference between the goal post's and its corresponding shadow's position. The above procedure can also be used to find the location of the Parrot drone's shadow in $\mathbb{R}^3$. Together with $\vec{f}$, we can construct a second line, $L_2(r)$, passing through the Parrot drone's location in $\mathbb{R}^3$.

Due to errors associated with the estimation of the camera vectors and the 2D positions of objects in Figure 6, $L_1(r)$ and $L_2(r)$ may not intersect. The Parrot drone's location is therefore estimated using a two-step procedure. First we minimize the function

$$F(r_1, r_2) = ||L_1(r_1) - L_2(r_2)||, \tag{6}$$

which computes the distance between the two lines. This can be done in Mathematica. We then estimate the Parrot drone's position as given by the point

$$\frac{L_1(r_1') - L_2(r_2')}{2} \tag{7}$$

where (r_1', r_2') is the absolute minimum of (6). Geometrically, (7) represents the midpoint of the shortest line segment connecting $L_1(r)$ and $L_2(r)$.

5 Conclusion

The goal of this paper was to demonstrate a relatively inexpensive method for determining the location of an object moving in $\mathbb{R}^3$ using a single photographic image. The mathematics involved did not exceed material covered in typical multivariable calculus and linear algebra course. This suggests that this activity could be applied in a undergraduate level class. For more advanced students, those interested in image recognition could take this activity one step further, automating the detection of the object of interest. Many options exist, giving educators a wide range of choices, from basic applications of geometry to more open-ended projects.