Chapter 3

The Tennis Ball Problem

Andrej Bona,[3] Chris Bose,[9] Kell Cheng,[3] Wolgan Engler,[1] Minglun Gong,[7] Cyril Guyot,[8]
Cristian Ivanescu,[8] John King[5] Dan Kenway,[1] Nathan Krislock,[6] Claude Laflamme,[3]
Abid M. Malill,[4] Suresh Pillai,[2] Anamaria Savu,[8] Fridolin Ting,[8] Satoshi Tomoda[3]

3.1 Introduction

Stereoscopic vision is a well-established phenomenon: biological evolution showed its utility in ancient times. In this workshop, we have examined some subtleties and limitations in applying this old concept to an entirely new application: with modern technology, we attempt to track the position of an early segment of a flying object, and then extrapolate its later trajectory.

The concept is easily described as two cameras peering into a region of interest, through which a tennis ball is tossed along some trajectory. With knowledge of two simultaneous measurements from the cameras, one should be able to triangulate a measured ball position. With a set of such measurements, iterated over some timing set, one should be able, in principle, to estimate the trajectory of that ball. With aid from the mechanics of the situation, moreover, one should be able to extrapolate to later times.

In practice, one is confronted by several difficulties. Some imaging difficulties stem from the optics, and others derive from computer image discretization. Some ball tracking problems are due to background geometry, and others arise in the mechanics of ball motion. Practicalities greatly expand the scope and complexity of the original concept. The workshop study group divided the base problem into the following clearly definable elements:

[1] VisionSmart
[2] University of British Columbia
[3] University of Calgary
[4] University of Guelph
[5] University of Notingham
[6] University of Regina
[7] University of Saskatchewan
[8] University of Toronto
[9] University of Victoria

1. Camera parameters, including camera type, frame rate, optics and placement.

2. Camera calibration methods and detection of bad calibration conditions.

3. Ball image detection and centroid estimation: lighting and background conditions and further complications from shadows and motion blur.

4. Mechanics of basic ball motion including gravity, air resistance and ball spin.

5. Algorithms using extra data where only one camera reports.

6. Direct algorithms on raw data from individual camera data.

7. Error assessment and data weighting.

The results from the workshop study follow.

3.2 Optimal Locations of the Cameras

Our main objective is to determine the best approximation of the centroids of the balls in trajectory by choosing the best possible placement of two cameras. In order to determine the optimal locations of the cameras, one needs to first consider what the least requirements are. Thanks to VisionSmart's experiments, we are give the minimum number of pixels required to compute for the centroid within reasonable errors. VisionSmart estimates this number to be eight. We must also know what is the maximum allowable distance d from the camera to the ball to satisfy this pixel requirement. See Figure 3.2 for the calculation of d.

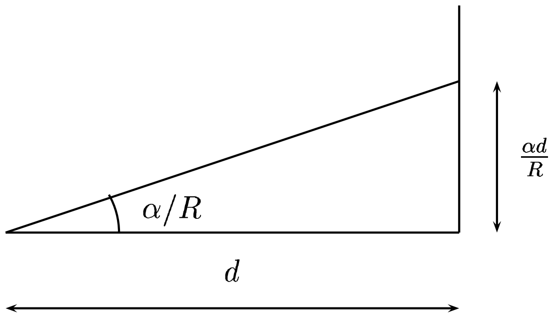

Figure 3.1: The maximum distance d

Let r be the radius of a tennis ball in meters, p the minimum number of pixels required on an image for the diameter of the tennis ball, α the lens visibility angle measured in radians , and R the resolution. In our case, these values are

$$r \approx 0.035, \text{ meters } p = 8, \text{ pixels } \alpha \approx 0.42, \text{ radians and } R \approx 500 \text{ pixels.}$$

At distance d, one pixel corresponds to approximately $\frac{\alpha d}{R}$. To guarantee p pixels on the image, we need $2r \geq p\frac{\alpha d}{R}$. So, $d \leq \frac{2rR}{\alpha p}$, that is, $d \leq \frac{2\times 0.035\times 500}{0.42\times 8} \approx 10.41m$. Hence, the maximum allowable distance from a camera to the tennis ball is around 10 meters.

Now that we have found the maximum effective distance from the camera to the ball, we can discuss the optimal locations for the cameras. One may first start by placing the two cameras opposite each other. A problem arises in this situation. The depth of the image is lost, which translates to difficulty in obtaining the ball trajectory. Instead of placing the cameras opposite each other, say, put them next to each other, pointing at the same direction. Again, the depth problem arises.

To solve the depth problem, one will need to place the cameras at some optimally chosen angle. In other words, the normal lines of the lenses form a certain angle other than multiples of 180 degrees. Note that we must intersect these lines; otherwise, we cannot recover the depth loss. So, let us try placing the cameras perpendicularly. That is, the directions of the two cameras form a right angle. It turns out that this is the optimal angle for getting the maximum number of pictures that provide the best approximation of the centroids.

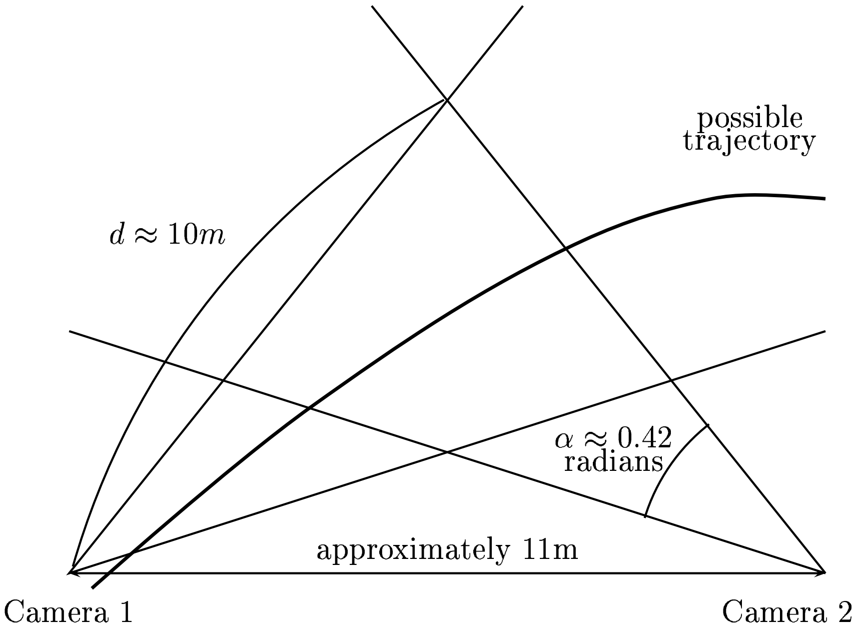

Figure 3.2: Camera positions viewed from above

Before proving the optimal angle is the right angle, we will sketch how we shall place the cameras relative to the tennis ball trajectory. With respect to the algorithm computing the trajectory described in Section 3.6, one of the cameras must capture the entire trajectory. The images from earlier parts of the trajectory provide more accurate information about the positions of the tennis ball, while the images from the later trajectory supply the effect due to spin of the ball and the gravitation. Thus, we place the cameras relative to the trajectory as in Figure 3.2.

To see the right angle is the optimal angle, consider Figure 3.2. In our case $\epsilon \approx 8.4 \times 10^{-4}$ radians ≈ 0.02 degrees and $m \approx 5.5$meters. If $\theta < \frac{\pi}{4}$, then the maximum possible error is the distance of the two vertices which form a horizontal line segment. For $\theta > \frac{\pi}{4}$, the maximum possible error is the distance of the two vertices that form a vertical line.

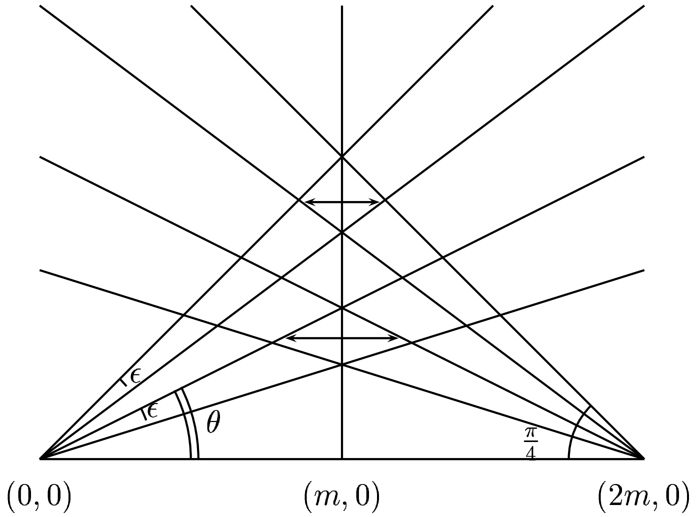

Figure 3.3: Camera relation

Without loss of generality, assume that $\theta < \frac{\pi}{4}$. We will find the x-coordinate of the left side point of the horizontal line segment. This point is the intersection of the line from the origin with slope $\tan(\theta)$ and the line with slope $-\tan(\theta-\epsilon)$ passing through the point $(2m,0)$, that is, the intersection of the lines

$$\begin{aligned} y &= \tan(\theta)\cdot x, \text{ and} \\ y &= -\tan(\theta-\epsilon)\cdot x + 2m\tan(\theta-\epsilon). \end{aligned}$$

Thus, the x-coordinate of the required point is $\frac{2m\tan(\theta-\epsilon)}{\tan(\theta)+\tan(\theta-\epsilon)}$. It follows that the distance required is, by symmetry,

$$2\left(m - \frac{2m\tan(\theta-\epsilon)}{\tan(\theta)+\tan(\theta-\epsilon)}\right) = 2m\frac{\tan(\theta)-\tan(\theta-\epsilon)}{\tan(\theta)+\tan(\theta-\epsilon)}.$$

Thus, we need to show that $\frac{\tan(\theta)-\tan(\theta-\epsilon)}{\tan(\theta)+\tan(\theta-\epsilon)}$ is decreasing on $(0,\frac{\pi}{4})$.

By differentiating it, we get $\frac{-\tan(\theta)\sec^2(\theta-\epsilon)+\tan(\theta-\epsilon)\sec^2(\theta)}{(\tan(\theta)+\tan(\theta-\epsilon))^2}$ and we need to show that this is negative for all $\theta \in (0,\frac{\pi}{4})$. Since the denominator is non-negative, by simplifying the numerator, we get

$$\begin{aligned} & -\tan(\theta)\sec^2(\theta-\epsilon)+\tan(\theta-\epsilon)\sec^2(\theta) \\ = & -\frac{\sin(\theta)}{\cos(\theta)}\frac{1}{\cos^2(\theta-\epsilon)} + \frac{\sin(\theta-\epsilon)}{\cos(\theta-\epsilon)}\frac{1}{\cos^2(\theta)} \\ = & \frac{-\sin(\theta)\cos(\theta)+\sin(\theta-\epsilon)\cos(\theta-\epsilon)}{\cos^2(\theta)\cos^2(\theta-\epsilon)}. \end{aligned}$$

Since $-\sin(\theta)\cos(\theta) = -\frac{1}{2}\sin(2\theta)$ has negative values on $(0, \frac{\pi}{4})$, the original expression $2m\frac{\tan(\theta)-\tan(\theta-\epsilon)}{\tan(\theta)+\tan(\theta-\epsilon)}$ is decreasing on $(0, \frac{\pi}{4})$ as required.

Remark:

Although we have proved that the most accurate information is retrieved when the normal lines of the lenses intersect at a right angle, it maybe the case, in practice, that the closer the better. That is, the calculated maximum error may not occur in real life. To find this, we need to run more experiments. If this is indeed the case, then we would have to rearrange the orientation of the two cameras.

3.3 Calibration

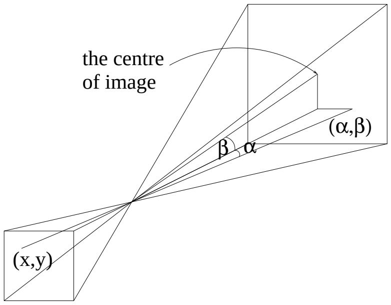

Figure 3.4: Optical Centre

To account for the lens distortion, we have to calibrate the data. We have to determine (α, β) as a function of (x, y), where (α, β) is an angular coordinate of a ray from the optical centre to a point on the grid, and (x, y) is a pixel coordinate of this point. This can be done by fitting a polynomial of sufficient degree, say four, through the points $((\alpha, \beta) \leftrightarrow (x, y))$. The second-degree polynomial is enough according to the VisionSmart's experiments. We chose four just to be on the safe side.

The next step is to fix the axis with respect to which we determined (α, β). This is done by mounting a pair of laser pointers on the camera in such a way that they are parallel to the axis and collinear with the reference axis. We can set the lasers in such a way that $x_1 = x_2$ (see Figure 3.3).

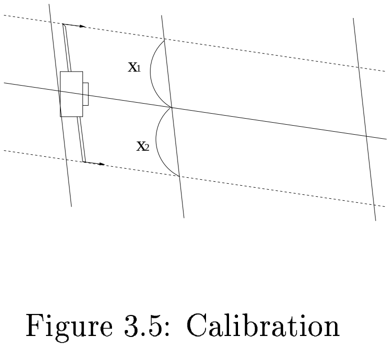

Figure 3.5: Calibration

This way, we can use the cameras in different places without recalibrating at the new location. Knowing these calibration polynomials, we can get the 3-dimensional coordinate of an object, in particular, the centroid of the tennis ball, by the standard surveyor's method.

3.4 Determining the Centroids (Lighting, Background, Shadows)

3.4.1 Determining the centroid of the ball

In the situation of a uniformly lit ball, we can compute the "centre of intensity" of the subtracted picture.

$$(x, y) = \frac{\sum w_{ij}(i, j)}{\sum w_{ij}},$$

where (x, y) is the coordinates of the centre and (i, j) is the coordinates of a pixel with intensity w_{ij}.

The major problem of finding the centroid of a ball lies in the shadowed portion of the ball. Improper lighting of the environment causes the ball to be bright on one side and dark on the other. (See Figure 3.4.1)

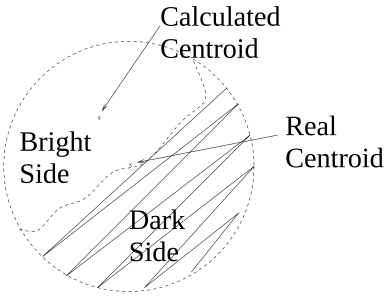

Figure 3.6:Shadow Image

If one were to calculate the centre of intensity of the ball in Figure 3.4.1, then one would find that it would be off the position of the real centroid. Of course, the centroid of the ball means the actual centre of mass of the ball.

3.4.2 Suggested solution

1. Placing lights directly behind cameras

   Of course, we cannot put the lights directly behind the cameras, otherwise all the light will be blocked off by the cameras. Therefore, we suggest putting the lights just above or below the cameras.

   The reason why we want to do this is because of the following situation.

Consider Figure 1.

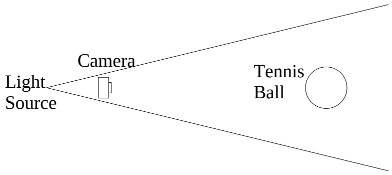

Figure 3.7: Light Source

Since the light is directly behind the camera, the reflected light will bounce directly back to the camera. This will certainly remove the shadow from the ball, and hence we will see a completely circular ball, not the shadowed version in Figure 3.4.1.

We can even improve this setup by using only green light for one light source and red light for the other light source. This can be easily implemented by putting a filter in front of the light sources. In addition, we can put a filter in front of the cameras so that: **each camera only sees the light reflected from its own illumination source.** All we can detect is green light and red light, and hence: **each camera image will be a full disc with centre approximately at the centre of intensity** allowing us to find the position of the centroid more easily.

2. Using black ball and white background
   Lighting of the environment plays a major role in the production of the shadows on the tennis ball. As we saw above, this causes major problems in finding the centroid. We can reduce the difficulty of this problem of lighting by using a black ball and a white background. This is because the shadow on the ball is black. That is, the difference between the actual colour of the tennis ball and the shadow is small. Therefore, what we expect to see is a circular black ball in the image.

3. Tracing the shadow of the ball
   As well as tracking the ball itself, we can track the shadow of the ball on the background. This way, we can get two estimates on the centroid of the ball. For instance, if the shadow travels on the ground, we can get another estimated xy-coordinate of the centroid of the ball, thereby improving the estimate for the centroid of the ball. Of course, we need to arrange the lighting so that there will be a shadow.

3.5 Dynamics

3.5.1 Formulation

Under appropriate assumptions, the trajectories can be taken to be governed by the sixth order system (in which $\hat{\boldsymbol{k}} = (0,0,1)$ and g is the acceleration due to gravity):

$$\begin{aligned} m\frac{d\boldsymbol{v}}{dt} &= -C_D\boldsymbol{v} + C_L\boldsymbol{\omega}\times\boldsymbol{v} - mg\hat{\boldsymbol{k}}, \\ I\frac{d\boldsymbol{\omega}}{dt} &= -C_A\boldsymbol{\omega}, \end{aligned}$$

where m and I are the mass and the moment of inertia of the sphere and $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are its velocity and angular velocity. Asymptotic expressions (Rubinow and Keller, 1961) are available for the drag coefficient C_D, the lift coefficient C_L (corresponding to the Robins-Magnus effect) and the angular drag coefficient C_A in the limit of small Reynolds number; however, for the high Reynolds numbers of interest here, more phenomenological expressions are adopted, and we take

$$C_D = c_D|\boldsymbol{v}|, \qquad C_L = c_L, \qquad C_A = 0 \ \ \text{(negligible angular drag)}$$

with c_D and c_L constant. These expressions follow from experimental fits (allowing c_D and c_L to depend on $a|\boldsymbol{\omega}|\big/|\boldsymbol{v}|$, where a is the radius of the sphere) as given by Stephanek (1988) for a tennis ball; we anticipate that $a|\boldsymbol{\omega}|\big/|\boldsymbol{v}|$ will be negligible for most cases of interest here. Taking initial conditions

$$\text{at } t=0 \qquad \boldsymbol{x} = \boldsymbol{X}, \quad \boldsymbol{v} = \boldsymbol{V}, \quad \boldsymbol{\omega} = \boldsymbol{\Omega},$$

we thus arrive at

$$\begin{aligned} m\frac{d\boldsymbol{v}}{dt} &= -c_D|\boldsymbol{v}|\boldsymbol{v} + c_L\boldsymbol{\Omega}\times\boldsymbol{v} - mg\hat{\boldsymbol{k}}, \qquad \frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}, \\ \text{at } t=0 \quad & \boldsymbol{x} = \boldsymbol{X}, \qquad \boldsymbol{v} = \boldsymbol{V} \end{aligned} \tag{3.1}$$

which is the system we will study.

3.5.2 Small-time expansion

From (3.1) we have as $t \to 0$ that

$$\begin{aligned} \boldsymbol{v} &= \boldsymbol{V} + \frac{1}{m}\Big(-c_D|\boldsymbol{V}|\boldsymbol{V} + c_L|\boldsymbol{V}|\boldsymbol{\Omega}\times\boldsymbol{V} - mg\hat{\boldsymbol{k}}\Big)t + O(t^2) \\ \boldsymbol{x} &= \boldsymbol{X} + \boldsymbol{V}t + \frac{1}{2m}\Big(-c_D|\boldsymbol{V}|\boldsymbol{V} + c_L\boldsymbol{\Omega}\times\boldsymbol{V} - mg\hat{\boldsymbol{k}}\Big)t^2 + O(t^3) \end{aligned} \tag{3.2}$$

The quadratic form (3.2) will be a good approximation provided

$$t \ll \frac{m}{c_D|\boldsymbol{V}|}, \qquad \frac{m}{c_L|\boldsymbol{\Omega}\times\boldsymbol{V}|};$$

if the drag and lift terms are negligible, it will be adequate for all time but otherwise the non-linear system (3.1) should be solved for the later stages of the trajectory, rather than extrapolating the quadratic form (3.2). A best fit of the full data (using that from all the cameras at once) to the quadratic form (3.2) (using the known values of g and m) would enable the nine independent quantities from

$$\boldsymbol{X},\ \boldsymbol{V},\ c_D, \quad c_L\boldsymbol{\Omega}\times\boldsymbol{V} \tag{3.3}$$

to be estimated; alternatively (and preferably) c_D could be measured beforehand and the remaining eight quantities fitted ($\boldsymbol{X}$ may also be known *a priori*). Note that $\boldsymbol{\Omega}$ appears in (3.2) only as in the combination $c_L\boldsymbol{\Omega}\times\boldsymbol{V}$, so no information can be obtained in this way about the component of $\boldsymbol{\Omega}$ parallel to $\boldsymbol{V}$; moreover,measuring c_L beforehand would be of little benefit since only the combination $c_L\boldsymbol{\Omega}$ is relevant, with $\boldsymbol{\Omega}$ unknown *a priori*. In fact, it may in practice be best to neglect $c_L\boldsymbol{\Omega}$ in performing the fitting, at least in the first instance (if $\boldsymbol{v}$ deviates noticeably from $\boldsymbol{V}$, the lack of information on $c_L\boldsymbol{\Omega}\cdot\boldsymbol{V}$ may become a significant drawback). Fitting the data to the parameter set (3.3) (instead of using the data to estimate $\boldsymbol{x}(t)$ directly) would enable information from a single camera to be used at a given location (by projecting (3.2) onto the appropriate surface), so all the available data could be treated on the same footing and synchronization of the cameras would not be necessary (indeed, in principle a single camera could suffice); moreover, since (3.2)–(3.3) aim to incorporate in a systematic fashion the dynamics which determine the trajectories, it is hoped that errors resulting from the various measurements would to some degree be self-correcting (since consistency with the governing physics is being demanded), rather than accumulating.

3.5.3 Later times

The available evidence is that the drag terms in (3.1) are non-negligible (the lift (spin) terms may be small), so a parabolic fit will be inappropriate over the later stages of the trajectory. Thus (3.1) should be solved numerically until impact occurs, with the values of c_D, $\boldsymbol{X}$, $\boldsymbol{V}$ and $c_L\boldsymbol{\Omega}\times\boldsymbol{V}$ estimated as above. Evaluation of $c_L\boldsymbol{\Omega}\cdot\boldsymbol{V}$ would require more terms in the expansion (3.2) to be taken (these can be readily calculated) and used in the fitting; however, it is likely that the contribution of this component will be negligible, in which case it can be set to zero.

3.5.4 Recommendations

In summary, we propose:

(1) Fitting all the available camera data to the parameter set (3.3), or a subset thereof (whereby c_D and/or $\boldsymbol{X}$ are determined beforehand; it should be relatively straightforward to estimate c_D from separate experiments), the other fitting parameters being the distance of the sphere to the camera for each data point for each camera.

(2) Extrapolating to give the later stages of the trajectories by using the above estimates as parameter values and initial conditions in a rapid and straightforward numerical solution of the nonlinear system (3.1).

3.6 An Iterative Method for Computing Full Trajectories

In the previous section it was observed that the sequence of ball images from a single camera already contains more information than that which is used in the simple-minded triangulation approach for determining the trajectory of the ball. In this section we discuss an iterative algorithm which aims to reconstruct the 'missing coordinate' from the sequence of images obtained from one dominant camera which sees a long segment of the ball trajectory, while incorporating the triangulation data available in overlap with the second camera. Essentially we combine the original approach of VisionSmart, reworked to incorporate the full dynamics, with the projection method described in Section 3.5.

Briefly, at the k-th iteration of the routine, estimates are given on the 'missing coordinate' from each camera. Nonlinear regression against the parametric family of true dynamic trajectories is performed to produce a new table of pseudo-data for the trajectory. These computed pseudo-data are then compared with the camera images providing updated estimates on missing coordinates. These new estimates provide input for the next iteration. There are a number of reasonable options for stopping rules for the algorithm which we will discuss.

To begin, we assume that the following data are known

- Camera 1 tracks the entire trajectory (100 images approximately) (roughly) from behind. To establish the orientation, we assume Camera 1 is fixed in the yz-plane. The ball is launched in such a way that its trajectory lies in the positive orthant of $\mathbf{R}^3$.
- Camera 2 is fixed in the xz-plane and records at least 8 images from the first 20 images captured by Camera 1. Thus we have at least 8 images in the overlap for the two cameras
- Time $t = 0$ is defined to coincide with the first image in Camera 1.
- 'Error cones' on each data point obtained from all the images recorded by each camera overlap are assumed to be known from preprocessing of the data. We denote these as $\mathcal{C}_k(j)$, for the error cone associated with the k-th image in the j-th camera, $k = 0, 1, \ldots, 100$, $j = 1, 2$. The error cones are convex and compact (since we know the dynamics is restricted to lie within some big closed ball centered at the origin) and their vertex angles depend on the distance between adjacent pixels in the camera, the distance from the focal plane in the camera to the centre of the lens, and on the acuity of our centre of mass calculation from the pixelated images which are our raw data. Where the k-th image is missing from Camera 2, we may as well define $\mathcal{C}_k(2) = \mathbf{R}^3$.

Moreover, we adopt the assumption from the previous section that

- The spin vector $\mathbf{\Omega} =< \omega_1, \omega_2, \omega_3 >$ is assumed constant throughout the trajectory.

With these assumptions, we consider the full dynamic equations (3.1). Since the ball trajectories are very flat, we adopt the linearizing assumption that the speed $|\mathbf{v}|$ is nearly constant over the time of flight, so the drag term is well approximated by

$$c_D|\boldsymbol{V}||\boldsymbol{v}|. \tag{3.4}$$

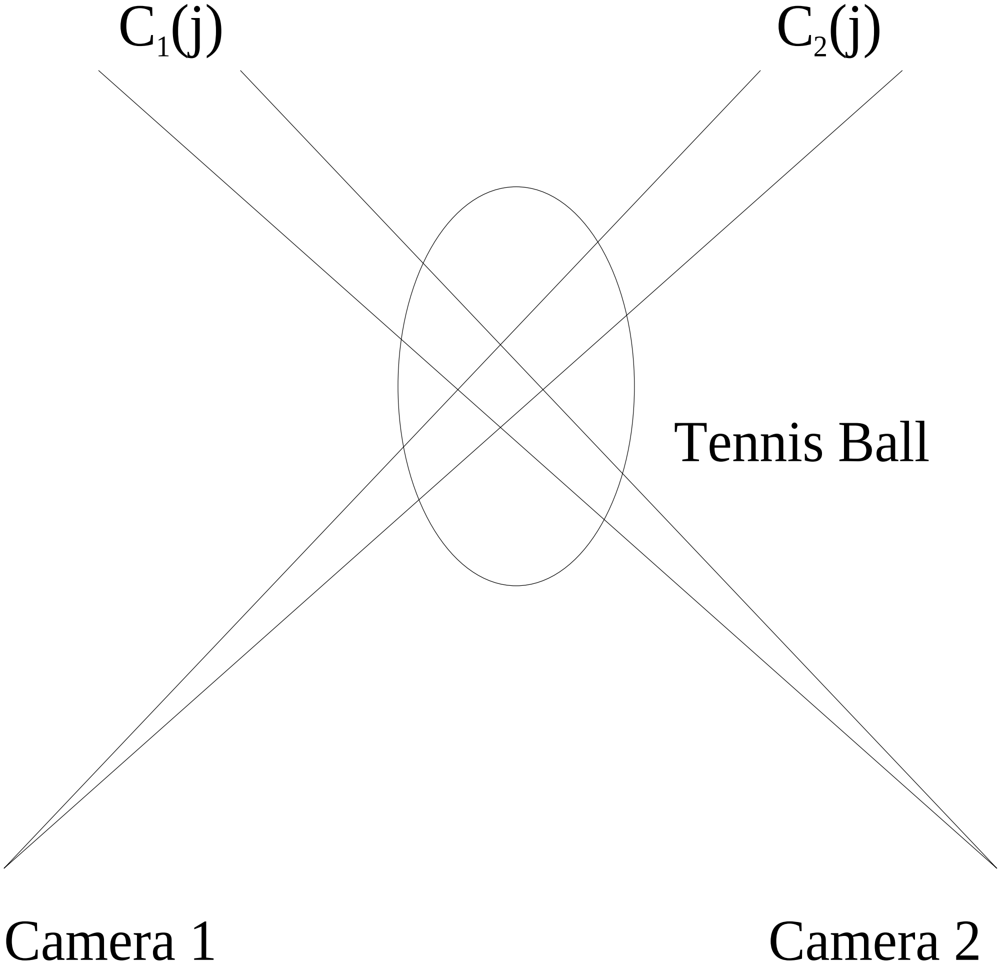

Figure 3.8: Error Cones

Recall that $\mathbf{V}$ is the ball's initial velocity.
In component form we are left with the equations

$$\begin{aligned}
\ddot{x}_1 &= -(\frac{c_D|\boldsymbol{V}|}{m})\dot{x}_1 - \frac{c_L}{m}\omega_3\dot{x}_2 + \frac{c_L}{m}\omega_2\dot{x}_3 \\
\ddot{x}_2 &= -(\frac{c_D|\boldsymbol{V}|}{m})\dot{x}_2 + \frac{c_L}{m}\omega_3\dot{x}_1 - \frac{c_L}{m}\omega_1\dot{x}_3 \\
\ddot{x}_3 &= -g - (\frac{c_D|\boldsymbol{V}|}{m})\dot{x}_3 + \frac{c_L}{m}\omega_1\dot{x}_2 - \frac{c_L}{m}\omega_2\dot{x}_1
\end{aligned}$$

which we write in compact form as

$$\ddot{\boldsymbol{x}} = m^{-1}A\dot{\boldsymbol{x}} - \boldsymbol{g}$$

where

$$A = \begin{bmatrix} -c_D|\boldsymbol{V}| & -c_L\omega_3 & c_L\omega_2 \\ c_L\omega_3 & -c_D|\boldsymbol{V}| & -c_L\omega_1 \\ -c_L\omega_2 & c_L\omega_1 & -c_D|\boldsymbol{V}| \end{bmatrix},$$

$$\boldsymbol{g} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

where c_D and c_L are the drag and lift coefficients (constants) respectively for the ball, m is its mass, and g is the constant acceleration due to gravity.

Since $A = -c_D|\boldsymbol{V}|I - c_L\Lambda_\Omega$ where it is expected that $c_D|\boldsymbol{V}| \gg c_L||\Lambda_\Omega||$, A is invertible and one easily solves the linear equation in closed form:

$$\boldsymbol{x}(t) = \boldsymbol{X} + mA^{-1}\boldsymbol{g}t - (e^{m^{-1}At} - I)(m^2A^{-2}\boldsymbol{g} - mA^{-1}\boldsymbol{V}) \tag{3.5}$$

We assume that the ball's mass m and the value of g is known (but see later discussion), however there remain 10 independent free parameters in this solution, represented by the initial position $\boldsymbol{X}$, velocity $\boldsymbol{V}$ the spin vector $c_L\boldsymbol{\Omega}$ and the drag coefficient c_D. VisionSmart's experience is that it is very difficult to estimate these parameters accurately enough from the few images in the camera overlap. For example, they typically reported a value of the gravitational constant g to be low by a factor of 2 when computed from their limited data! This is not too surprising since a quick calculation shows that the expected effect of gravitational acceleration (for example) is to introduce a curvature in the trajectory on the order of one pixel over the duration of the camera overlap. That is why we believe it is essential to incorporate the additional camera images in some way to improve accuracy, especially in the accelerative coefficients.

3.6.1 The first estimate on positions

The routine we recommend begins with a preliminary no-spin estimate on the missing data from the second camera. At this point we also use a textbook value for the drag coefficient for a sphere

$$c_D = 0.22D^2 \approx 0.00108$$

where $D = 0.07$ is the diameter of the ball. We adopt the no-spin condition $\boldsymbol{\Omega} =< 0, 0, 0 >$ and $g = 9.8m/s^2$. We are given the spatial data from the 8 images in the camera overlap, which we display as a table:

t=k	$x_1(k)$	$x_2(k)$	$x_3(k)$
5	$x_1(5)$	$x_2(5)$	$x_3(5)$
6	$x_1(6)$	$x_2(6)$	$x_3(6)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
12	$x_1(12)$	$x_2(12)$	$x_3(12)$

Using this data, along with equation (3.5) in a nonlinear least squares approximation we obtain our first estimate on the positions for all times $t = 0, 1, \ldots, 100$. Denote this new set of spatial data by

$$\hat{x}_1(k) = x_1|_{t=k}, \quad \hat{x}_2(k) = x_2|_{t=k}, \quad \hat{x}_3(k) = x_3|_{t=k}.$$

Now, it is possible that some of these computed data points are not consistent with the camera observations, so before we proceed with the next iteration we need to adjust these computed data as follows. The camera observation of the k-th image indicates the position of the centroid of the ball at $t = k$ only within the error cone $\mathcal{C}_k$ emanating from the camera, centered on a ray through the centre of the camera lens and the position of the computed centroid on the camera focal plane. The true centroid lies somewhere inside $\mathcal{C}_k$. It may happen that our computed spatial data, for some $t = k$ fails to lie inside the error cone determined by the k-th image recorded by some camera. In that case we perturb our computed spatial data to lie in the error cone:

$$(x_1(k), x_2(k), x_3(k)) = (x_1, x_2, x_3)_{\min}$$

where $(x_1, x_2, x_3)_{\min}$ is the unique solution to

$$\text{minimize } |(x_1, x_2, x_3) - (\hat{x}_1(k), \hat{x}_2(k), \hat{x}_3(k))|$$

subject to $(x_1, x_2, x_3) \in \mathcal{C}_k$. For values of $k = 5, 6, \ldots, 12$ we use the constraint $(x_1, x_2, x_3) \in \mathcal{C}_k(1) \cap \mathcal{C}_k(2)$ (convex, cpct) where, again, $\mathcal{C}_k(j)$ is the k-th image error cone for the j-th camera. In this way we determine our first estimates on the 100 ball positions:

t=k	$x_1^{(1)}(k)$	$x_2^{(1)}(k)$	$x_3^{(1)}(k)$
0	$x_1^{(1)}(0)$	$x_2^{(1)}(0)$	$x_3^{(1)}(0)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	$x_1^{(1)}(100)$	$x_2^{(1)}(100)$	$x_3^{(1)}(100)$

as well as the first estimate on the drag parameter $c_D^{(1)}$.

3.6.2 The iteration loop ($n = 2, 3, \ldots$)

The input is the table of positions computed from the previous step:

t=k	$x_1^{(n)}(k)$	$x_2^{(n)}(k)$	$x_3^{(n)}(k)$
0	$x_1^{(n)}(0)$	$x_2^{(n)}(0)$	$x_3^{(n)}(0)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	$x_1^{(n)}(100)$	$x_2^{(n)}(100)$	$x_3^{(n)}(100)$

We should note at this point that there is non-uniform reliability of this data for geometric reasons. Since the cross sectional area of the error cone for each camera grows linearly in distance from the camera, the accuracy of the computed centroid decays linearly in distance from the camera. For the images $k = 13, 14, \ldots, 100$ the distance of the ball from camera 1 can be estimated by $x_1^{(n)}(k)$ so we should weight the k-th data point in the above table in proportion to $(x_1^{(n)}(k))^{-1}$. For simplicity we weight the data in the camera overlap uniformly as $C_2 \quad k = 5, 6, \ldots 12$ and for $k = 0, 1, \ldots, 4$ uniformly as C_1. We should assume $C_1 < C_2$, and, again for simplicity we suggest $C_1 = \frac{C_2}{2}$. Finally, we need a probability distribution for the weights, which determines C_2:

$$(5/2 + 8)C_2 + \sum_{k=13}^{100} C_2(x_1^{(n)}(13))(x_1^{(n)}(k))^{-1} = 1$$

giving weights for the n-th iteration dataset

$$w^{(n)}(k) = \begin{cases} \frac{C_2}{2} & \text{if } 0 \leq k \leq 4 \\ C_2 & \text{if } 5 \leq k \leq 12 \\ C_2(x_1^{(n)}(13))(x_1^{(n)}(k))^{-1} & \text{if } k \geq 13 \end{cases}$$

Using the n-th data set, with the weights above, nonlinear least squares approximation on equation (3.5) where $\boldsymbol{V}$, $\boldsymbol{X}$, $c_L\boldsymbol{\Omega}$, c_D and g are the parameters to be fitted, yields the $(n+1)-$st estimate on the ball positions:

$$\hat{x}_1^{n+1}(k), \hat{x}_2^{n+1}(k), \hat{x}_3^{n+1}(k).$$

Again we use the hats to indicate that some of these computed data points may not be consistent with the camera images, so before we leave this iteration we perturb the data to lie in each camera's error cones (in exactly the same way which was done for the first iteration). This final step yields the output for the loop:

t=k	$x_1^{(n+1)}(k)$	$x_2^{(n+1)}(k)$	$x_3^{(n+1)}(k)$
0	$x_1^{(n+1)}(0)$	$x_2^{(n+1)}(0)$	$x_3^{(n+1)}(0)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	$x_1^{(n+1)}(100)$	$x_2^{(n+1)}(100)$	$x_3^{(n+1)}(100)$

along with estimates on the dynamic parameters $\boldsymbol{V}^{(n+1)}$, $\boldsymbol{X}^{(n+1)}$, $c_L\boldsymbol{\Omega}^{(n+1)}$, $c_D^{(n+1)}$ and $g^{(n+1)}$. Note that we are keeping the gravitational constant as an unknown parameter in the iterative procedure. A quick count gives 11 free parameters and 303 data points as input.

3.6.3 Stopping criterion

We suggest monitoring the value of $g^{(n)}$ and stopping when $g^{(n)} \approx 9.8$. It may happen that the spatial data stabilize before this is reached, in which case we should also stop, but then it is probably reasonable to question the initial data from the camera images. Of course any real implementation of this routine should have a maximal number of iterations fixed. The size of this cutoff could be determined by running the algorithm on various sample data sets.

3.6.4 Fine tuning

We have made some choices in the description above which should be experimented with in any implementation.

- The first iteration uses uniform weights for the data points coming from the camera overlap, when, in fact, geometric considerations give a non-uniform reliability to each point. The reliability could be estimated in order to apply a weighted least squares approximation, even at this first step.

- Similarly, there is quite a bit of flexibility in the weights profile $w^{(n)}(k)$ which might be used to advantage.

- It is possible to make estimates on the spin and drag effects from the camera 1 images only. These could be used in the first iteration in order to get more accurate first-run estimates on the positions of the ball for large k.

Bibliography

[1] Rubinow, S. I.; Keller, Joseph B. *The transverse force on a spinning sphere moving in a viscous fluid. Journal of Fluid Mechanics* **11** 1961 447 – 459.

[2] Stepanek, Antonin. *The aerodynamics of tennis balls – The topspin lob. The American Journal of Physics* **56**, 138 (1988).