

Spatial Navigation Principles: Applications to Mobile Robotics

Anthony Suluh

Thomas Sugar

Michael McBeath

Mechanical and Aerospace Engineering, Psychology
Arizona State University
Tempe, AZ 85283

Abstract

Human navigational principles are researched and applied to mobile robotic applications. Two principles, constant optical rate and angular constancy, can be used by a fielder or mobile robot to define a trajectory for interception of a projectile without knowledge of the ball's or fielder's world coordinates. Two novel, robust, viewer-centered algorithms, a passive and an active version, are modeled and experimentally validated.

1 Introduction

Different spatial perception strategies for intercepting a projectile have been debated. We are focusing on perceptual strategies used for the initial phase of catching. Namely, how to determine a path so that the fielder converges to the destination of the projectile. The strategies assume that humans have a low level, unconscious, motor-visual control algorithm that is reliable and robust [1].

One strategy, the Optical Acceleration Cancellation strategy (OAC) [2], describes a baseball fielder catching a fly ball by selecting a running path to achieve optical acceleration cancellation of the ball in the image plane or *constant optical rate in the image plane*. For this strategy, fielders run along a path that horizontally maintains their alignment with the ball and maintains a constant change in the tangent of the optical angle ($\tan \alpha$). See Figure 1.

Another perceptual strategy uses a Linear Optical Trajectory (LOT) proposed by McBeath et. al. [3]. McBeath et. al. proposed that the fielder maintains a linear optical trajectory (LOT) of the ball relative to the home-plate and background scenery. The LOT model maintains the *optical projection angle, ψ , to be constant* so that there exists an imaginary linear line on the background scenery due to the projection image of the ball as viewed by the fielder. See Figure 2. By doing so, the image of the ball always rises above the fielder guaranteeing that the fielder travels to the correct destination. The models and the detail concepts of these two strategies are discussed later in this paper.

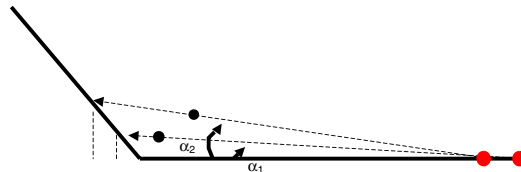


Figure 1: Fielders adjust their position so that the image of the ball rises at a constant rate. For parabolic ball trajectories, $\frac{d}{dt} \tan \alpha$ equals a constant.

In the present work, experiments were conducted to show the feasibility of the OAC model and to demonstrate the behavior of human-based visual-servoing in robotics. In the first part of the paper, the OAC and LOT models are discussed followed by simulations to show the validity of the mathematical models of the two strategies. Next, the OAC model and visual tracking of the ball are explained. In the last part of the paper, the experiments are described for three different ball trajectories to show the robustness of the OAC algorithm applied to the robot.

2 Literature Review

As mentioned above, the OAC model stipulates that the fielder selects a running path that is laterally aligned with the ball and adjusts his approach velocity so that the change in the vertical optical angle from the ground plane is constant with respect to time, ($\frac{d}{dt} \tan \alpha = c$) [2]. If the rate of change of $\tan \alpha$ increases, the fielder has run too far forward (or too slowly backwards), and the ball will land behind the fielder. Similarly, if the rate of change of $\tan \alpha$ decreases, the fielder has run too far backward (or forward too slowly) and the ball will fall in front. The rule to catch the ball is simple. The fielder needs to select a running path so that the $\frac{d}{dt} \tan \alpha$ is equal to a constant achieving an optical acceleration cancellation algorithm (OAC) in order to catch the ball. The algorithm nulls any errors by keeping the image of the ball always constantly rising in the image plane.

According to McBeath, the OAC model is elegant but has several weaknesses [3]. OAC solutions require a precise ability to discriminate acceleration, and typi-

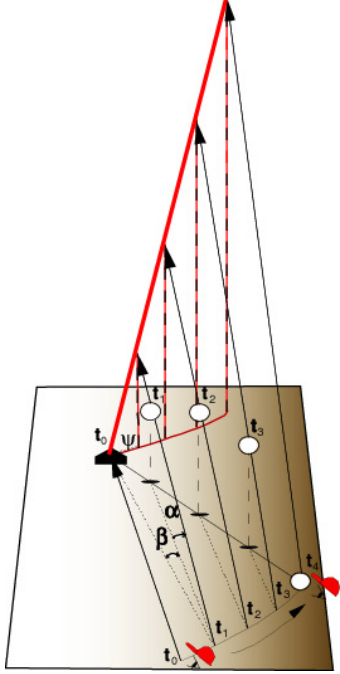


Figure 2: Fielders adjust their path to keep the image of the ball on the projection plane monotonically rising along the straight-line trajectory described by the angle, ψ .

cal outfielders lack sufficient acceleration resolution to account for their accuracy in catching a fly ball.

McBeath et. al. proposed the Linear Optical Trajectory, or LOT model. The projection of the ball is analyzed as a unified two-dimensional (2-D) image in which two angles, α (vertical optical angle) and β (lateral optical angle), are introduced to define a linear optical trajectory, projection angle, ψ . In this model, the fielder selects a running path that maintains a linear optical trajectory (LOT) with the ball relative to home plate and the background scenery, in order to catch the ball. See Figure 2. The magnitude of the angle ψ is related to the angles α and β by the relationship $\tan \psi = \frac{\tan \alpha}{\tan \beta}$.

Edward Aboufadel, a mathematician, has proposed two models, a Strong LOT and a Regular LOT model [4]. The Strong LOT model stipulates that a fielder's running path is only determined by the location of the ball (x, y, z) [4]. For this model, there is only one unique path of the fielder for every ball trajectory. The Regular LOT model, on the other hand, allows the image plane (or projection plane) to rotate, and there is no longer a unique path, but the fielder still keeps the $\tan \psi$ equal to a constant in the moving image plane, and it can be shown that the fielder still arrives at the destination.

In robotics, [5], Burgstadt and Ferrier have demonstrated a mobile robot system based on the OAC model. In their implementation, they collected accel-

eration data from the image, which is very noisy. In contrast, we have adapted the problem to use control variables in the visual plane simplifying the problem. Others have been interested in visual control in the image plane such as Zhang and Ostrowski. [6].

Other navigational strategies can be applied using map building methods. In this system, a world map of the robot is known, and the robot is directed to the target location determined by curve fitting past image data. At MIT, a ball catching robot has been built, but it is not mobile and relies on fitting a curve to the past image data to determine the future location of the projectile [7]. Other researchers have suggested modeling the ball trajectory and determining the destination of the ball in world coordinates. Physicists note that complex iterative models are required to capture the influence of everyday variables like temperature, humidity, friction and projectile spin, thus making this strategy very difficult [8].

3 Modeling and Simulation

3.1 Passive OAC Model

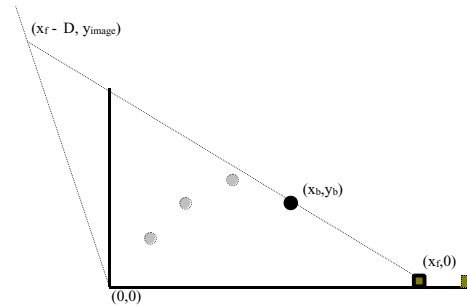


Figure 3: The actual image of the ball, y_{image} , varies depending on the robot's position. It is found by a directed line through the points $(x_f, 0)$ and (x_b, y_b) . With a stationary camera, (passive model), the image of the ball in the picture should increase at a constant rate if $\frac{d}{dt} \tan(\alpha) = c$. (Proof: Use similar triangles and a pin-hole camera model. See Figure 4.)

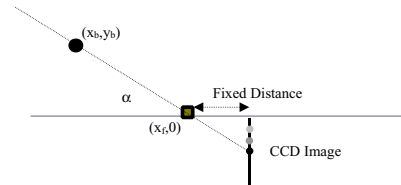


Figure 4: The ball's position on the image plane is directly related to y_{image} .

In contrast to previous work in robotics, our calculations are done in the image plane. See Figure 3. In

Model 1, the camera is *fixed* (does not tilt), and the actual and desired position of the ball in the image will be compared. In the first algorithm, the ball must rise at a constant rate in the *stationary*, image plane

$$\frac{d}{dt}(\tan \alpha) = \frac{d}{dt} \left(\frac{y_b}{(x_f - x_b)} \right) = C \quad (1)$$

where $(x_f, 0)$ and (x_b, y_b) are the world coordinates of the fielder and ball positions. It can easily be seen that with perfect knowledge,

$$\tan \alpha = Ct \quad (2)$$

$$x_f = x_b + \frac{y_b}{Ct} \quad (3)$$

where

$$C = \frac{\dot{y}_b(0)}{D} \quad D = x_f(0) - x_b(0) = x_f(0). \quad (4)$$

In the *stationary* image plane, the ball height will equal $(\tan \alpha)D$ and the change in the height of the ball in the image plane will equal $CD = \dot{y}_b(0)$. The fielder cannot know the initial upward velocity of the ball, but can estimate it after a fraction of a second. The variable, a , will be the estimate of the initial upward velocity of the ball.

Our control algorithm for the fielder or robot contains two terms. The first term is a feedforward velocity and the second term is the error between the actual height of the ball and the desired height of the ball in the image plane.

$$\dot{x}_f = K_p \left(\left(\frac{y_b}{(x_f - x_b)} \right) D - at \right) + B \quad (5)$$

$$B = \frac{\dot{y}_b(t_f)D}{at_f} + \dot{x}_b(t_f) \quad (6)$$

From simulation, the fielder's position will converge to the destination of the ball, $x_b(t_f)$, where t_f equals the landing time. Different ball trajectories are used including the addition of drag on the ball as well as different landing locations in front of or in back of the fielder. With perfect knowledge of a and B , the fielder's acceleration equals zero; otherwise, the fielder must quickly chase the ball at the end of the trajectory. This model allows the fielder to catch the ball on the run as in human studies.

The effects of the estimation of a are shown in three simulations. In all three simulations, $K_p = 2$, $\Delta t = 1/15$ (frame rate), $D = 150$, $y_b = -5t^2 + 100t$, $x_b = 5t$, and $B = 0$. The feedforward velocity, B , is zero because it is very difficult to determine experimentally. The upward velocity, a , can be estimated experimentally with the first 4 frames, and in the simulations, $a = 50, 100$, and 150 . As shown in the simulation if a is estimated too low, $a = 50$, then the

fielder does not catch the ball and the fielder runs forward and backward resulting in a poor running path choice. See Figure 5. In the second two cases, $a = 100$ and $a = 150$, the fielder catches the ball. See Figures 6-7.

In all simulation figures, time is measured in seconds on the horizontal axis. In the left figures, the ball trajectory is a straight line, $x_b = 5t$, and the fielder's path along the x-axis is shown, x_f . In the right figures, the desired image of the ball is a straight line (dashed). The actual image of the ball is a solid line.

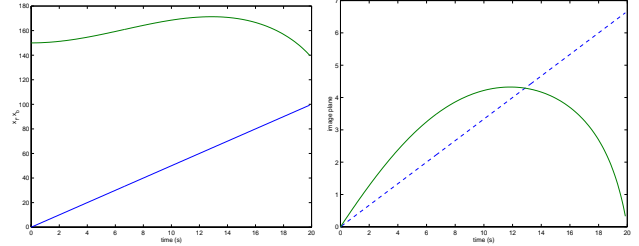


Figure 5: In simulation 1, the estimation for the upward velocity is too small, $a = 50$. In the left figure, the fielder does not reach the destination of the ball, $x_b = 100$ at $t = 20s$. The fielder's trajectory actually moves forward and backward, and the ball's trajectory is a straight line. In the right figure, the desired position of the ball in the image is a straight line (dashed), but the actual image of the ball (solid) varies quite a bit and the fielder must accelerate quickly near the end of the trajectory to try and compensate.

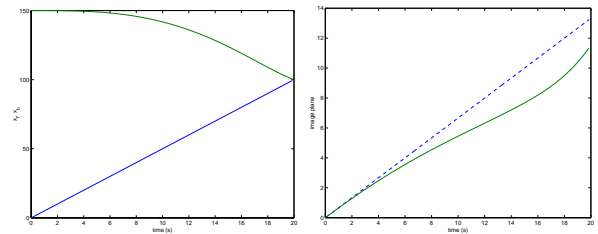


Figure 6: In simulation 2, the estimation for the upward velocity is correct, $a = 100$. In the left figure, the fielder reaches the destination of the ball, $x_b = 100$ at $t = 20s$. In the right figure, the actual image of the ball (solid) varies some and the fielder must accelerate quickly near the end of the trajectory to try and compensate.

3.2 Active OAC Model

In model 2, the tilt angle of the camera is constantly adjusted using the formula:

$$\alpha_{des} = \arctan(Ct) \quad (7)$$

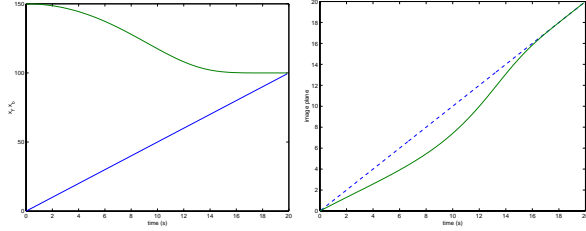


Figure 7: In simulation 3, the estimation for the upward velocity is too large, $a = 150$. The actual image of the ball (solid) becomes straight.

The desired image position of the ball is always at the center of the image. If the ball is not at the center of the image, the robot corrects the error in the image plane by moving forward or backward. It is assumed that the pin-hole is fixed and the CCD image rotates about the pin-hole. In this case, a similar formula is derived.

$$\dot{x}_f = K_p \left(\left(\frac{y_b}{x_f - x_b} \right) D - at \right) \cos \left(\arctan \left(\frac{at}{D} \right) \right) \quad (8)$$

In these simulations, the maximum velocity is significantly decreased and K_p can be greatly increased. The effects of the estimation of a are shown in two simulations. In the simulations, $K_p = 40$, $a = 100$, or 150, and the other parameters are not changed. As shown in the simulations, the fielder catches the ball and the errors in the image plane are decreased significantly. See Figures 8-9.

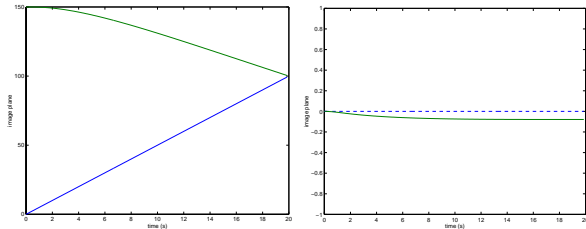


Figure 8: In simulation 4, the estimation for the upward velocity is correct, $a = 100$. In the left figure, the fielder reaches the destination of the ball, $x_b = 100$ at $t = 20s$. In the right figure, the desired position of the ball in the image is zero (dashed), and the actual image of the ball (solid) is very close to the desired value; therefore, the acceleration is low.

3.3 LOT Model

A very good mathematical description of the LOT model in world coordinates was developed by Aboufadel [4]. LOT algorithms for mobile robotics are developed by Sugar [9]. We have used this model to simulate different ball trajectories as well as limiting

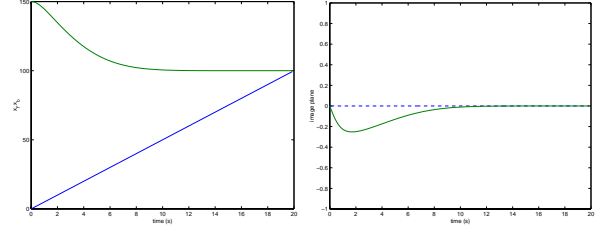


Figure 9: In simulation 5, the estimation for the upward velocity is too large, $a = 150$. In the right figure, as the actual image of the ball (solid) converges to zero, the velocity of the robot nears zero as well.

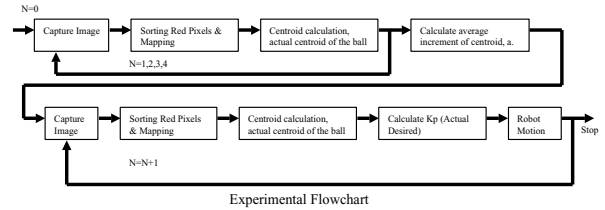


Figure 10: The upward velocity of the image of the ball is first estimated and then the control algorithm for Model 1 or Model 2 is executed.

the fielder's speed. In this model, the spatial-temporal problem is reduced to an optical geometry problem.

4 Experiments

4.1 Setup

Currently, we are using a Nomad Scout from Nomadic Technologies Inc. with an additional pan/tilt mechanism installed from Direct Perception. The biscuit PC uses the Linux operating system and controls the drive wheels, pan/tilt mechanism, and PCI frame grabber. We use a standard calculation to find the center of mass of the red ball in our experiments. The calculations along with capturing the image takes about 0.0625 seconds. A flowchart for the experimental calculations is shown in Figure 10.

4.2 Validation

In the first experiment, the OAC strategy is shown to be valid for a fielder whose location is at the ideal destination of the ball. Because the fielder is at the ideal location, its position is stationary. The image of the ball moves upward in the image plane of the fielder when the ball is launched until the ball is intercepted. This experiment is conducted by placing a stationary robot at an ideal location and setting the camera tilt angle equal to zero (the camera is horizontal). The average increment of the centroid of the ball in the image plane for the first, second, and third trials are 7, 9, and 8 pixels with the standard deviation of 4, 6,

and 3 pixels, respectively. The high standard deviation occurs because of three major reasons. The first reason is that there are errors in calculating the centroid of the ball. Secondly, when the ball gets close to the camera, the image captured is not a solid image because the shadow of the ball covers most of the camera lens. Lastly, the motion of the ball is not a perfect parabolic motion. The results obtained agree with the OAC model, with the ball moving upward monotonically in the image plane as viewed from a stationary robot at an ideal location (the image captured never moves downward). See Figure 11.

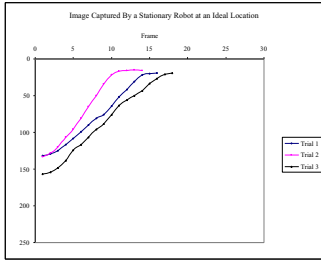


Figure 11: The actual image of the ball for three trials when the robot remains stationary.

4.3 Experiment 1, Passive OAC Model

In this experiment, the OAC Model 1 is used with a stationary tilt angle. The ball is guided by a human hand (frame rate is too slow) with three different ball motions.

In the first case, the ball moves in a projectile motion so that it will land in front of the robot. In this case, the actual pixel centroid of the ball moves linearly upward in the image plane during three quarters of the experiment, and the robot is able to maintain a constant optical rate of the image of the ball. At the end of the experiment, the difference between the actual centroid and the desired centroid in the image plane increases, and hence, the robot accelerates quickly. See Figure 12.

In all figures, the camera frame is along the horizontal axis. In the left figures, the actual robot velocity and the desired robot velocity is measured in inches/s. In the right figures, the centroid of the ball is along the vertical axis and is measured in pixels.

In the second experiment, the ball moves in a parabolic motion for half of its motion and then the ball is dropped straight down to the ground in front of the robot simulating an increase in the drag force. The robot responds in similar manner and the robot accelerates forward faster when the ball drops suddenly, see Figure 13.

In the third experiment, the ball moves in a parabolic motion so that the ball lands behind the robot. In this case, the image of the ball rises fast,

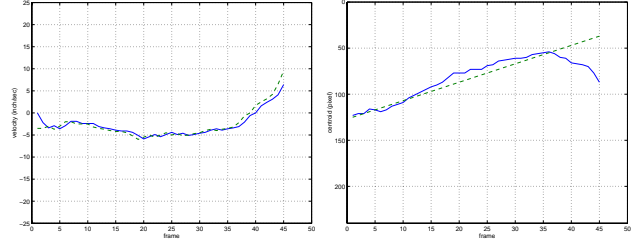


Figure 12: In the left figure, the desired (dashed) and actual velocity of the robot is given for a ball which lands in front of the robot. In the right figure, the desired and actual center of mass of the object is shown in pixels.

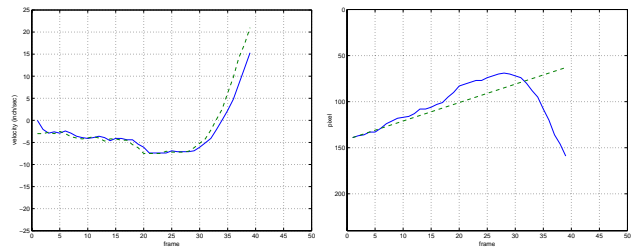


Figure 13: In the left figure, the desired (dashed) and actual velocity of the robot is given for a ball which lands in front of the robot. In the right figure, the desired and actual center of mass of the object is shown in pixels. The robot accelerates quickly as the ball suddenly drops at the end of the trajectory.

and thus, the robot moves backward. Eventually, the actual pixel of the centroid of the ball in the image plane starts slowing down (the ball starts to curve downward) causing the robot to accelerate forward, see Figure 14.

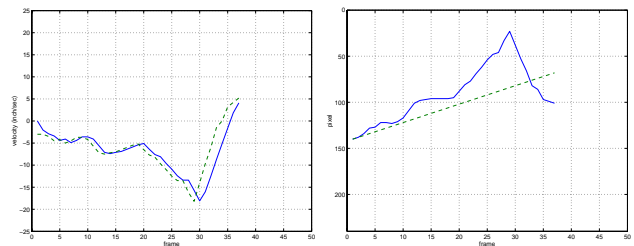


Figure 14: In the left figure, the desired (dashed) and actual velocity of the robot is given for a ball which lands behind the robot. In the right figure, the desired and actual center of mass of the object is shown in pixels. The robot accelerates backward as the actual ball image is greater than the desired image. At the end of the experiment, the robot accelerates forward to try and catch the ball.

4.4 Experiment 2, Active OAC Model

The same three experiments are repeated using the Active OAC Model. In this case, the camera is constantly tilted. As predicted in simulations, the difference between the actual centroid and the desired centroid is close. The robot moves with much slower velocity and better accuracy with the second model. In comparison, the actual and the desired centroid of the ball in the passive experiments are much further apart causing the robot to move with higher velocity. See Figures 15, 16, and 17.

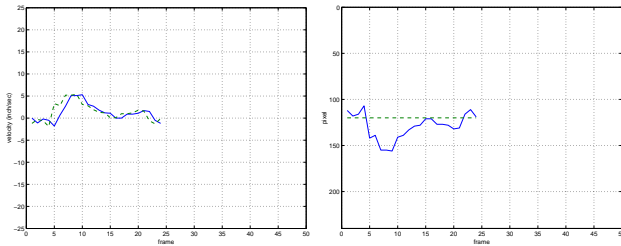


Figure 15: In the left figure, the desired (dashed) and actual velocity of the robot is given for a ball which lands in front of the robot. In the right figure, the desired and actual center of mass of the object is shown in pixels. Using the second control algorithm, the robot moves slowly forward to catch the ball.

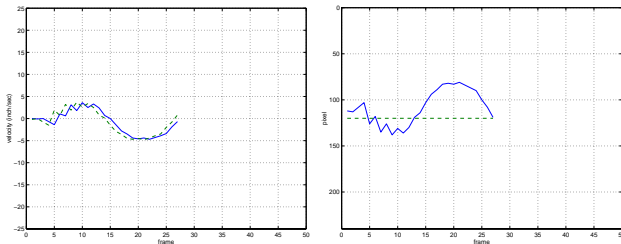


Figure 16: In the left figure, the desired (dashed) and actual velocity of the robot is given for a ball which suddenly drops. In the right figure, the desired and actual center of mass of the object is shown in pixels. The robot accelerates forward as the actual ball image dips below the desired image.

5 Discussion

We are investigating principles such as constant image rate, and optical constancy and applying these principles to navigational pursuit. With these principles, control algorithms for visual-servoing using image data will allow mobile robots to intercept varying, projectile trajectories. We demonstrated that the ball image constantly rises in the image plane of a stationary robot, a nonintuitive result. Both the Passive

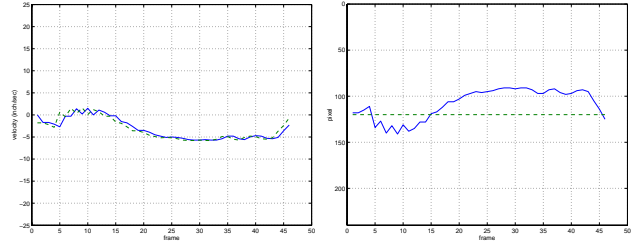


Figure 17: In the left figure, the desired (dashed) and actual velocity of the robot is given for a ball which lands behind the robot. In the right figure, the desired and actual center of mass of the object is shown in pixels. In the beginning of the experiment, the robot accelerates backward as the actual ball image is greater than the desired image.

and Active OAC Models are new algorithms for implementing the concept of *constant optical rate in the image plane*, and the active algorithm performed better in both simulations and experiments implying that the pursuer is actively participating in the interception of a ball. Our experimental work is still in progress, and future work will be motivated towards validating the LOT algorithm as well. We believe the multidisciplinary research between perceptual psychology and robotics will help us develop navigational algorithms that will be robust, powerful, and simple.

Acknowledgments

Support from ASU is gratefully acknowledged.

References

- [1] A. Milner and M. Goodale, *The Visual Brain in Action*. Oxford Psychology Series 27, Oxford University Press, 1995.
- [2] P. McLeod and Z. Dienes, "Running to catch the ball," *Nature*, vol. 362, no. 6415, 1993.
- [3] M. K. McBeath, D. M. Shaffer, and M. K. Kaiser, "How baseball outfielders determine where to run to catch fly balls," *Science*, vol. 268, no. 5210, pp. 569–573, 1995.
- [4] E. Aboufadel, "A mathematician catches a baseball," *American Mathematics Monitor*, vol. 103, pp. 870–878, 1996.
- [5] J. A. Borgstadt and N. J. Ferrier, "Interception of a projectile using a human vision-based strategy," in *IEEE International Conference on Robotics and Automation*, 2000.
- [6] H. Zhang and J. P. Ostrowski, "Visual servoing with dynamics: Control of an unmanned blimp," in *1999 International Conference on Robotics and Automation*, 1999.
- [7] W. Hong, "Robotic catching and manipulation using active vision," Master's thesis, MIT, 1995.
- [8] R. G. Watts and R. T. Bahill, *Keeping Your Eye on the Ball*. NY: Freeman and Co., 1990.
- [9] T. G. Sugar and M. McBeath, "Spatial navigation algorithms: Applications to mobile robotics," in *Vision Interface, 2001, VI2001*, 2001.