Intercepting a Falling Object: Digital Video Robot

Keshav Mundhra, Anthony Suluh, Thomas Sugar, Michael McBeath

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

Abstract

Human based algorithms to catch fly balls have been research and studied. In this paper, the validity of the human models are tested by catching balls that are dropped vertically downward. The regular OAC (Optical Acceleration Cancellation) and the inverted OAC models are simulated first, and then tested experimentally with a mobile robot. The simulation and the experimental results show that both methods are able to intercept the dropped ball, but the initial and final motions are different. A new digital image processing program, DVRobot, was written to process digital images very quickly.

1 Introduction

The digital video robot (DV Robot) is a high-speed robot programmed to catch fly balls based on human navigational principles. This high-speed system involves distribution of tasks between computers. Vision based navigational strategies require tremendous image processing to extract desired parameters from the video. A fast computer does this task and then the information is transmitted to the computer on the robot, which uses this data in the implementation of the navigational algorithm.

Digital video systems use IEEE 1394 that provides a high-speed, Plug and Play-capable bus. The goal of the protocol is to provide an easy-to-use, low-cost, high-speed communication line. The protocol is also very scalable, and provides for both asynchronous and isochronous applications. High speed video data is needed for interception tasks because the task lasts only for 1 to 5 seconds.

A falling object is an interesting interception case for a mobile robot or a user because the object is seen falling during the entire task. In psychology case studies, only the interception of thrown objects which rise and fall have been studied. We are interested in modeling, simulating, and experimentally testing vision based algorithms for navigating towards a falling object.

Figure 1: An extra computer is added to the Nomad Scout for image processing. A new program, DVRobot is written which can capture and analyze images from 1394 Digital Cameras or DV Camcorders.

2 Literature Review

The algorithm for catching dropped balls is based on the Optical Acceleration Cancellation (OAC) model [1]. In this model, the fielder selects a running path to achieve optical acceleration cancellation of the ball in the image plane or constant optical rate in the camera's image plane. For this strategy, fielders maintain their alignment with the ball and also maintain a constant change in the tangent of the optical angle, $\tan(\alpha)$.

There are two modes to simulate the motion of the camera. In the passive model, the camera is stationary and the image of the ball in the picture should rise/fall at a constant rate if $\frac{d}{dt}(\tan(\alpha))$ is to be kept constant. In active mode, the tilt of the camera is constantly adjusted using the formula $\alpha_{des} = \arctan(Ct)$ [2].

In our experiments, we have used the active OAC model [2]. Brancazio had suggested that the acceleration of the angle α is a better control variable than acceleration of $\tan(\alpha)$, because it works better for ball flights undergoing drag [3]. He introduces a two-component strategy to catch the fly ball - one for motion initialization and the other for controlling the running speed. We use the OAC model because it has also shown that the acceleration of the angle works only when the projection of the ball is not too steep [1].

In robotics, Burgstadt and Ferrier have demonstrated a mobile robot based on the OAC strategy. In their implementation, they collected acceleration data from the image, which is very noisy [4]. In our previous work, we developed perceptual algorithms in the visual plane by modeling the camera as a passive or an active device. With the passive OAC model, the camera is always stationary (no tilting) and the robot moves to keep the image of the object constantly rising in the image plane. With the active OAC model, however, the tilt of the camera is constantly adjusted according to the rate at which the center of the ball moves in the image plane, and the robot moves to maintain the ball at the center of the image. [2, 5, 6]. Experimentally, the mobile robot intercepts the object more consistently using the active model.

3 Mathematical Modeling and Simulation

3.1 Model for a Falling Object

We developed two mathematical models for determining how a robot will catch a ball that is dropped straight down. As compared to previous work in psychology, the fielder intercepts an object that rises first and then falls. In this situation, the fielder only watches the object fall and then must determine a motion to intercept the object.

In the regular, passive model, the fielder will move forward to keep the ball rising at a constant rate in the image plane. We used a drawing package to simulate a ball falling and a fielder moving towards the interception point. See Figure 10 at the end of the paper. As the fielder moves forward, the ball rises a fixed distance, 100 mm, in the image plane.

In the inverted, passive model, the fielder will move backward and forward to keep the ball falling at a constant rate in the image plane. Again a drawing package is used to simulate the ball falling and a fielder moving towards the destination. See Figure 11 at the end of the paper. As the fielder moves, the ball falls at a constant distance of 60 mm in the image plane.

The first model to be simulated by a computer is the regular, active OAC strategy similar to the model discussed in our previous paper [2]. In this model, the robot actively tilts the camera upward with a constant rate and maintains the image of the ball to be at the center of the camera by moving forward or backward. The robot does not know the exact velocity of the ball, but it can estimate the rate at which the center of the ball increases or decreases in the image plane.

In the OAC model, the elevation angle, α, from the fielder to the object is the fundamental variable. In the model, the derivative of the $\tan(\alpha)$ is constant. Therefore, $\tan(\alpha)$ equals a constant multiplied by time as well as second constant representing that at the start of the task the ball has a certain height. The ball is assumed to fall at $x_b = 0$.

$$\frac{d}{dt}\tan(\alpha) = C_1 \tag{1}$$

$$\tan\alpha = C_1 t + C_2 \tag{2}$$

$$x_f = \frac{y_b}{C_1 t + C_2} \tag{3}$$

where

$$C_1 = \frac{\dot{y}_b(0)}{D} \tag{4}$$

$$C_2 = \frac{y_b(0)}{D} \tag{5}$$

$$D = \; = x_f(0) - x_b(0) = x_f(0) \tag{6}$$

The second OAC model is similar to the first model except that the camera is tilted down from its initial tilt angle at a rate determined by $-C_1$. The robot moves backward or forward to center the image of the ball.

For both models, the tilt angle of the camera is constantly adjusted using the formula

$$\alpha_{des} = \arctan(C_1 t + C_2) \tag{7}$$

By tilting the camera, the desired position of the image of the ball is always at the center. 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. The camera is modeled as a pinhole and the CCD image rotates about the pinhole axis. Before the ball is dropped, the robot initially tilts the camera to center the ball. The regular active OAC model uses a positive value for the variable C_1. The rate of the image of the ball in the image plane increases. In the inverted OAC model, we use a negative value for C_1.

We use a proportional controller to determine the velocity of the robot. The error is measured in the image plane while the camera is being constantly tilted.

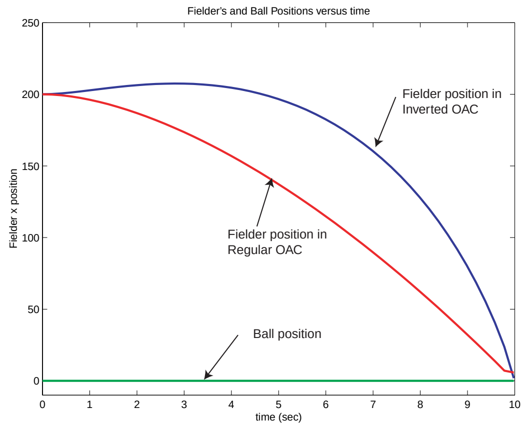

Figure 2: The fielder and ball position versus time. The fielder position for the regular model is always below the position for the inverted model because in the first model the fielder's initial velocity is much larger.

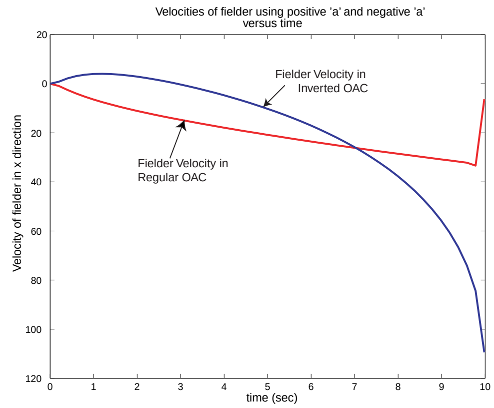

Figure 3: The fielder velocities during the interception task versus time. In the simulation for the regular model, the fielder moves quickly backward to catch the ball. In the simulation for the inverted OAC model, the fielder moves forward (positive velocity away from the ball) and then backwards to catch the ball. In the inverted model, the fielder's velocity is very large (negative) at the end of the task.

$$\dot{x}_f = K_p D \left(\left(\frac{y_b}{x_f} \right) \mp C_1 t - C_2 \right) \cos\left(\arctan\left(\pm C_1 t + C_2\right)\right) \tag{8}$$

In all simulations, the proportional gain, $Kp = 1.5$, $C_1 = \pm 40$, $x_b = 0$, and $z_b = 1000 - 10t^2$. To show the robustness of the control algorithm, the simulations for both models are performed with three different initial positions, $x_f = 50$, 200, and 300. The simulation results show that the fielder intercepts the ball in all the above initial conditions. See Figures 2 and 5. In the regular, active OAC strategy, the fielder starts with a high initial velocity so that the image of the dropping ball increases in the image plane, and the fielder starts slowing down as he nears the ball. On the other hand, in the inverted, active OAC model, the fielder starts with a low velocity in the beginning and speeds up at the end to intercept the ball. See Figure 3. In Figure 4, the actual ball image for inverted OAC model drops at the end of the simulation and causes the fielder to run very fast to intercept the ball.

4 Experiment

4.1 Set Up

In our experiment, we used a Nomad Super Scout robot from Nomadic Technologies Inc. with an additional pan-tilt mechanism installed on it. The computer on the robot runs Linux as the operating system. We used a SONY digital camcorder with a built in Firewire^{TM} IEEE 1394 port for video transmission. The laptop computer processing the images has a Firewire card installed on it. The image processing computer and the computer in the robot communicate via an Ethernet cable. We have developed our own data acquisition software - DVRobotVideo that runs on Windows 98, NT, and 2000 platforms. The software tracks the target by matching the RGB (red, blue, green) intensities of the pixels of the image with preset target colors. The RGB values can be set at run time. A standard calculation is used to find the center of mass of the target. The software allows for four resolutions of the image - 720x480, 360x240, 180x120 and 80x60. Our computer can process images at the rate of 15 frames per second at 320x240 resolution. Faster machines can easily process 30 frames per second at the 720x480 resolution.

In this paper, we discuss the OAC model applied to the special case of a ball falling vertically down instead of it following a usual parabolic path. It is assumed that the robot is initially aligned with the ball and so this task consists of a one-dimensional motion of the robot. The ball is first centered in the image by tilting the camera. The fall rate is then determined in the next few frames. Two cases of this OAC strategy are implemented. In the first case, the regular OAC, the drop rate, C_1 is taken to be a positive value. Hence the camera tilts upwards at a constant rate and the robot velocity is computed so as to keep the center of mass of the ball at the center of the image (i.e. the desired center). In the second case, the inverted OAC, a negative value of C_1 is implemented in the

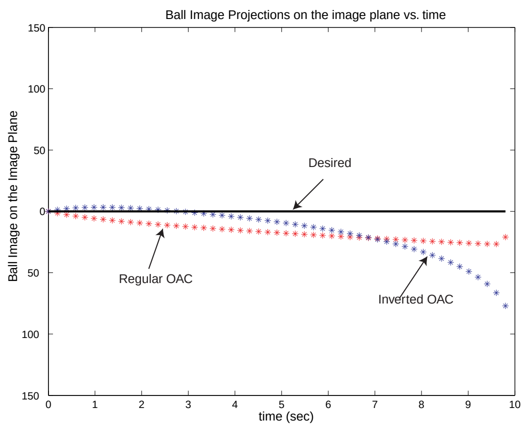

Figure 4: The image of the ball is projected on the background scenery. In the regular OAC model, the image of the ball stays close to the desired image.

controller such that the camera dips at a constant rate. Again the robot velocity is adjusted to cancel the error between the actual position of the center of mass of the target and the desired position.

It should be noted that when regular OAC is used, the desired location of the center of mass of the target should rise in the image plane. But because the pan-tilt mechanism is actively controlled and tilted at the desired rate given by C_1, the desired position of the center of the target is always at the center of the image. The same argument is true for the inverted active OAC model.

4.2 Dropped Ball, regular OAC

The response of the robot in this case is shown in Figure 6 and Figure 7. As the camera tilts upward and the ball is falling, the image of the ball drops to the lower half of the image plane, thus, causing the robot to move forward. In the first few seconds of the motion, the deviation from the desired center is large; therefore, the robot velocity is high as well. As the ball continues its motion downwards, the error between the actual position of the ball and the desired position (the center of the image) decreases. Because of the smaller errors close to interception, the robot's velocity is low.

4.3 Dropped Ball, inverted OAC

In this case, the camera first centers the ball by tilting up or down depending on the position of the ball in the image plane. The data from the next few frames are used to determine the drop rate of the ball.

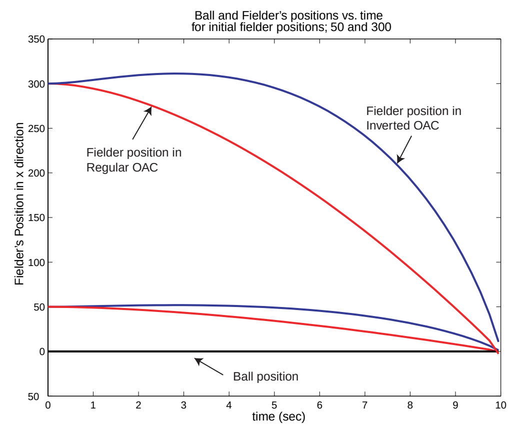

Figure 5: The fielder's and ball position versus time for different initial fielder positions, $x_f = 50$ and 300. At a given time instant, the fielder position for the regular OAC model is always below the position for the inverted model because in the first model the fielder's initial velocity is much larger.

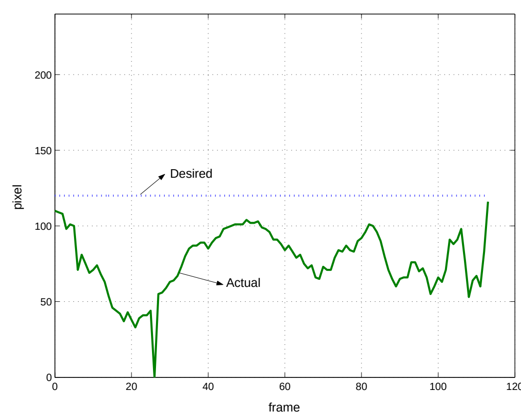

Figure 6: y coordinate of the ball, Regular OAC.

The camera is then tilted down at a constant rate proportional to the drop rate. Since the camera is tilting down and the ball is also falling, the error in the actual position of the ball during the initial motion is small, causing the robot to move slowly. See Figure 8. As the ball gains velocity, the error also increases in magnitude and so does the robot velocity. The increase in velocity can be seen in Figure 9.

5 Conclusion

In both experiments and simulations, keeping the ball rising at a constant rate or falling at a constant rate in the image plane (positive and negative C_1) determine strategies for catching a vertically falling object. The experiments corroborate the simulations in

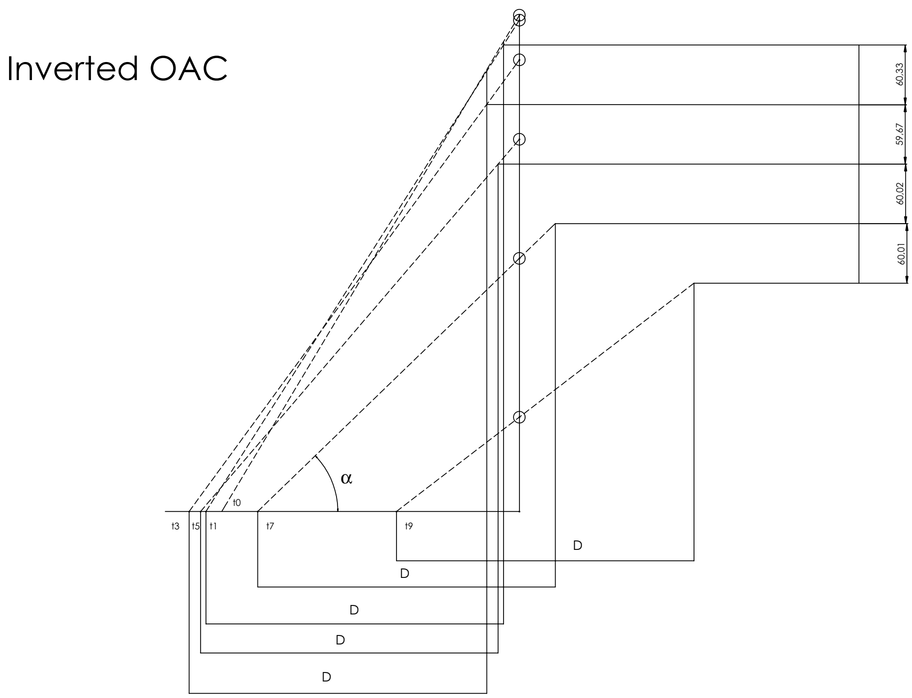

Figure 11: The fielder moves from position $(t_0, t_1, t_3, t_5, t_7)$ to position t_9. In the inverted case, the fielder actually moves backward at the beginning and then forwards to intercept the object. The ball falls a constant 60 mm in the image plane.

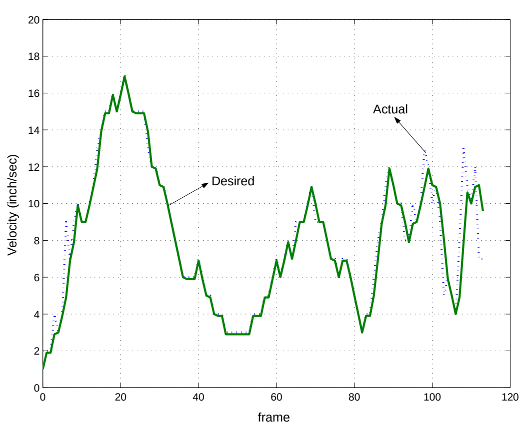

Figure 7: Velocities, Regular OAC.

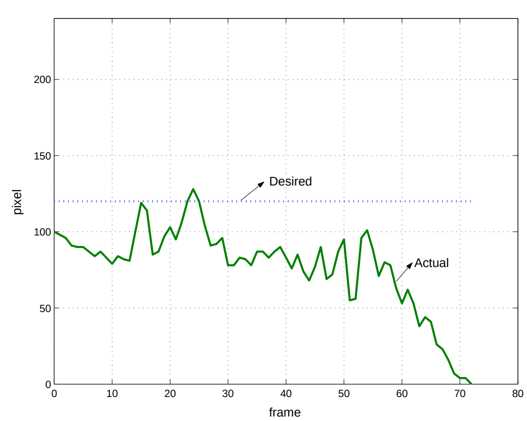

Figure 8: y coordinate of the ball, Inverted OAC.

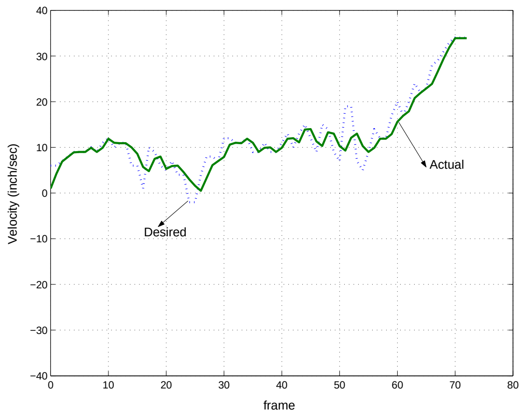

Figure 9: Velocities, Inverted OAC.

showing that the initial motion of the robot is fast in the case of positive C_1 in contrast to negative C_1 where the robot must drastically speed up at the end of the task.

We believe that collaborative research between perceptual Psychology and Robotics will aid in the development and validation of human perceptual algorithms that will allow us to explore and systematically quantify the limits of control variable parameters in the image plane for real world conditions.

Acknowledgments

Support from ASU is gratefully acknowledged.

References

[1] P. McLeod and Z. Dienes, "Running to catch the ball," *Nature*, vol. 362, no. 6415, 1993.

[2] A. Suluh, T. Sugar, and M. McBeath, "Spatial navigational principles: Applications to mobile robotics," in *IEEE International Conference on Robotics and Automation*, 2001.

[3] P. J. Brancazio, "Looking into chapman's homer: The physics of judging a fly ball," *American Journal of Physics*, pp. 849–855, 1985.

[4] 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.

[5] T. Sugar and M. McBeath, "Spatial navigational algorithms: Applications to mobile robotics," in *Vision Interface, VI 2001*, 2001.

[6] M. McBeath, T. Sugar, and D. Shaffer, "Comparison of active versus passive ball catching control algorithms using robotic simulations," in *Vision Sciences Conference*, 2001.

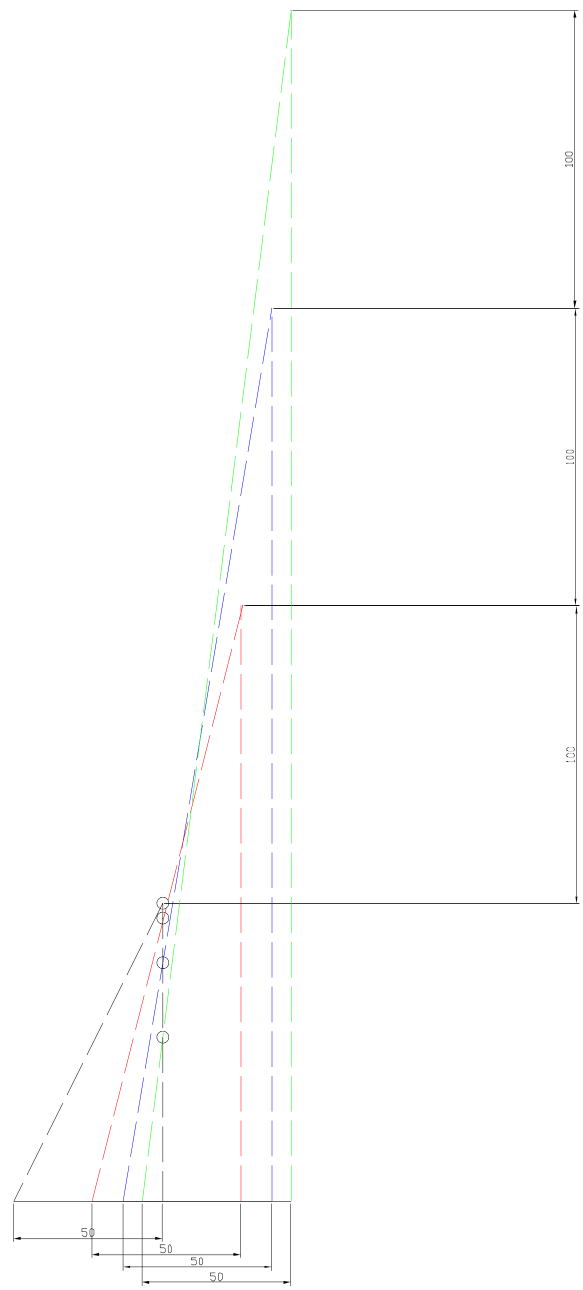

Figure 10: The fielder moves from the left to the right to intercept the object. The distance from the fielder to the image plane is a constant 50 mm simulating that the focal distance is constant in the camera. As the fielder moves forward, the ball rises in the image plane by a constant 100 mm. Note, initially, there is a finite elevation angle between the fielder and the ball.