A Vision-Based System for a UGV to Handle a Road Intersection

Javed Ahmed[1,2], Mubarak Shah[1], Andrew Miller[1], Don Harper[1], and M.N. Jafri[2]

[1]Computer Vision Lab, University of Central Florida, Orlando, FL-32816, USA
[2]Electrical (Telecom.) Eng. Dept., National University of Sciences and Technology, Rawalpindi-46000, Pakistan
jahmed@cs.ucf.edu, shah@cs.ucf.edu, amiller@cs.ucf.edu, harper@cs.ucf.edu, mnjafri@mcs.edu.pk

Abstract

We propose a real-time computer vision system that enables a UGV to safely cross urban road-intersections. Specifically, when the UGV approaches the stop sign at a 4-way intersection, it must be aware of the vehicles at the other three roads and adhere to traffic rules by waiting for its turn to proceed. The proposed solution consists of three main components: a vehicle detector, a tracker, and a finite-state-machine to model the traffic. We use an OT-MACH filter to detect the leading vehicle in each of three camera-views of the corresponding roads. Then, we track the vehicles using an edge-enhanced dynamic correlation tracker, which estimates the current and next positions, velocities, and accelerations of the vehicles. Finally, the finite-state-machine describes the traffic in each road with one of four possible states (i.e. *No Vehicle Waiting*, *Arriving*, *Waiting*, and *Passing*), and signals an autopilot system when it is safe to pass the intersection. We provide the results from an actual intersection with real traffic to demonstrate that the UGV is able to automatically navigate the intersection using the proposed system.

Introduction

Unmanned vehicles are steadily growing in demand since they save humans from having to perform hazardous or tedious tasks and the equipment is often cheaper than personnel. It further reduces cost to have unmanned vehicles remotely controlled by fewer remote operators by implementing autonomous robotics to partially assume the burden of control. Unmanned ground vehicles (UGVs) in particular have been successfully used for reconnaissance, inspection and fault detection, and active tasks like removal of dangerous explosives. These deployments have usually required the UGV to operate in relative isolation. However, future uses of UGVs will require them to be more aware of their surroundings. Deployment in an urban environment, for example, will require a UGV to behave within challenging constraints in order to avoid endangering or interfering with humans.

To foster the growth of research in practical autonomous UGVs, the United States defense research agency DARPA is organizing the Urban Challenge 2007 event, in which the participating organizations develop roadworthy vehicles that can navigate a mock urban obstacle course under complete autonomy. The original DARPA Grand Challenge event in 2005 required the UGVs to autonomously cross the Mojave desert. The new event is more challenging since the UGV must deal with a number of more difficult obstacles and constraints, such as parking in a confined space, following traffic laws, and avoiding interference with other vehicles on the same course.

Computer vision approaches have been successfully used in several previous UGV applications. For example, vision has been used to make the vehicle stay in its lane while driving on a road (Rasmussen 2002; Chun, Faukoner, & Munkeby 1995), to detect unexpected obstacles in the road (Matthies *et al.* 1995; Zhang, Weiss, & Hanson 1994; Matthies 1992; Matthies & Brown 1996; Talukder *et al.* 2002), to recognize road signs (Matthies & Brown 1996), and to avoid collisions with pedestrians (Shashua, Gdalyahu, & Hayun 2004). Some research that enables multiple agents (UGVs and UAVs) to coordinate with each other to accomplish a task has also been reported (Vidal *et al.* 2001).

In this paper, we discuss a new problem that must be solved for a UGV to successfully travel the streets of a real city. The UGV must be able to pass a street intersection regulated by a stop sign where vehicles from up to four directions are expected to voluntarily take turns crossing the intersection according to the order of their arrivals. In the DARPA Grand Challenge 2005, we entered a UGV that was equipped with sweeping laser range finders to detect and avoid obstacles. However, these sensors can only detect objects along a single line in space, so they are ill-suited to the recovery of higher level information about the scene. Additionally, if the UGV travels on uneven roads, the laser range finders often point off-target. Cameras are preferable in this situation because they have a conical field of view. Thus, we propose a computer vision solution, which consists of three main components: a vehicle detector, a tracker, and a finite-state-machine model of the traffic that determines when it is safe to cross the intersection. We implemented this algorithm as a complete real-time system that is installed onboard the UGV.

The rest of the paper is organized as follows. The next section defines the scope of the problem in detail. Our solution and implementation are summarized in the section, "Overview of the Proposed Solution". Each of the three
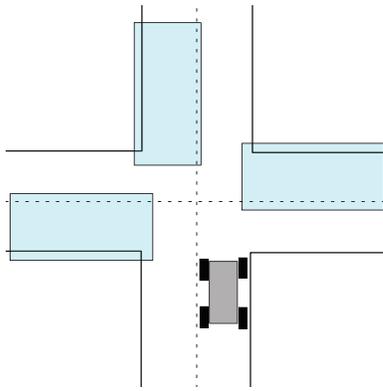
Figure 1: The four way intersection scenario. All vehicles must come to a stop before entering the intersection. The UGV must be aware of incoming vehicles in the right-hand lane of each of the three roads (*left*, *front*, and *right*), in the regions indicated by the shaded boxes.

Figure 2: Our experimental UGV is a Subaru Outback with an autopilot system and three cameras mounted to the roof.

components is described in further detail in separate sections. We describe the experiments we performed to test our system in the "Results" section. Finally, we summarize our research and suggest future work in the "Conclusion" section.

Problem Description

For this paper, we consider the scenario of a UGV approaching a four way intersection regulated by a stop sign. Traffic laws require that each vehicle must come to a stop before entering the intersection and allow any other vehicles that arrive earlier to pass first. The UGV must effectively wait for its turn and look for the leading vehicles at the other roads that want to pass the intersection. It is not sufficient to simply detect the presence of other vehicles at the intersection, since the UGV should have the right-of-way if other vehicles approach the intersection after it has already stopped. Thus, it will be necessary to determine the behavior of each other vehicle, i.e. whether it is just arriving at the intersection, waiting for its turn to go, or already passing through. We assume that all intersections will have at most four incoming roads oriented at perpendicular angles and that all vehicles will travel in the right-hand lanes only. Thus, our system must be aware of cars in three regions (see Figure 1) once the UGV has come to a stop.

We have to make some further relaxations regarding traffic flow to simplify the problem. If another vehicle approaches from the road straight ahead, beats the UGV to the intersection, and begins to make a right hand turn (a turn to the UGV's left), the UGV could safely make a right hand turn without having to wait. For simplicity, we will decide not to cross the intersection until the intersection is completely clear of other vehicles. We also assume that we will only encounter small vehicles with four or more wheels, but of any color or shape (e.g., a compact convertible or a pickup truck, but not a motorcycle or a semi-truck).

The DARPA Urban Challenge 2007 provides us with a

map of GPS paths along the lanes of the road, so it is unnecessary to perform path planning. The map also includes the locations of each regulated intersection and the possible directions of cross-traffic, so we do not have to detect the number of roads or the presence of a stop sign. Our experimental UGV is a Subaru Outback station wagon that has been used in previous autonomous vehicle projects. It is equipped with a GPS receiver, several racks of off-the-shelf computers, mechanical controls for the steering wheel, brakes, and accelerator, and an autopilot system that autonomously follows GPS waypoints. Thus, it is also unnecessary to consider low-level controls; it is instead sufficient to inform the autopilot when it is the UGVs turn to proceed through intersection. Since the UGV will come to a complete stop before entering the intersection, we can also assume that we have motionless cameras, further simplifying the scope of the computer vision system.

Overview of the Proposed Solution

We propose to mount three video cameras to the roof of the UGV, as shown in Figure 2. The cameras are pointed towards the three other roads leading to the intersection, i.e. to the right, to the left, and straight ahead. Each camera provides RGB color frames with a resolution of 320×240 at a frame rate of 30 fps. Each camera is connected to a separate off-the-shelf computer installed in the UGV. Each computer will run our proposed software, which is written in C++. The three computers communicate with the autopilot through a UDP Ethernet connection.

When the autopilot determines that the UGV has reached the intersection and has come to a stop, it will send a message to the three computers signaling them to begin looking for vehicles in their fields of view. Our proposed software consists of three main components: a vehicle detector, a tracker, and a finite-state-machine model of the intersection, as shown in Figure 3. First, the vehicle detector tries to detect a vehicle in each video frame by using an OT-MACH (Optimal Trade-off Maximum Average Correlation Height) filter (Refregier 1991; Mahalanobis *et al.* 1994;
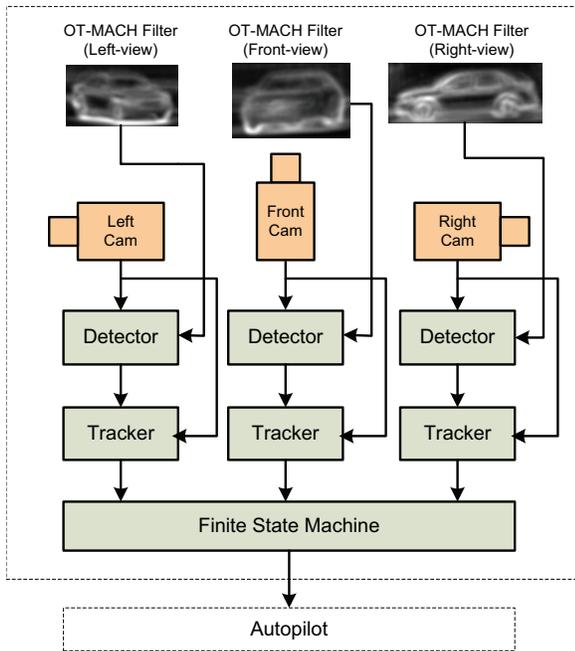
Figure 3: Block diagram of our proposed system. The *Vehicle Detector* and *Tracker* are run on the three cameras simultaneously. Also shown are the actual OT-MACH filters used for each camera.

Zhou & Chao 1999) constructed from training images of vehicles captured from each camera. Once a vehicle is detected in a single frame, the detector hands off control to a tracker that follows the vehicle in the subsequent frames, adapts to the changing appearance of the vehicle, handles occlusions, and estimates the current and next position, velocity, and acceleration of the vehicle. The detection and tracking is performed simultaneously for all three cameras. Finally, we use a finite-state-machine model to determine the behavior of each leading vehicle on the other three roads, and decide when it is safe for the autopilot to drive the UGV through the intersection. The three components of the proposed solution are discussed in detail in the next three sections.

Vehicle Detector

The first step of our proposed method is to detect a vehicle in a video frame by matching an appearance template that has been prepared for each of the three views. The template is basically an OT-MACH (Optimal Trade-off Maximum Average Correlation Height) filter, which combines the training images into a single composite template by optimizing four performance metrics: the *Average Correlation Height* (ACH), the *Average Correlation Energy* (ACE), the *Average Similarity Measure* (ASM), and the *Output Noise Variance* (ONV), as explained in (Refregier 1991; Mahalanobis *et al.* 1994; Zhou & Chao 1999). Since each incoming vehicle from a given road is oriented in approximately the same pose, and since the UGV always sees each road in the intersection from roughly the same point of view,

this method produces a template that expresses the general shape of a vehicle in each view. We achieve color invariance by using edge-enhanced images instead of the original color frames as described in (Ahmed *et al.* 2007). The edge-enhanced OT-MACH filter generated for each of the three views is shown in Figure 3.

Once we have prepared the OT-MACH filter, we can apply it to incoming video frames by performing cross-correlation in the frequency domain. In order to make the correlation result robust to varying lighting in the frame, we normalize the correlation output very efficiently using the concept of integral image (Porikli 1995; Lewis 1995).

Cross-correlation between an edge-enhanced search window with the template produces a surface in which the highest peak corresponds to the most likely location of a vehicle in the frame. When a vehicle is visible in the scene, there will be a sharp and distinct peak at that location in the frame. If there is no peak in the surface with a height greater than a threshold, then we determine that there is no vehicle in the scene.

The most significant change in the appearance of vehicles between different intersections comes from varying number and width of lanes. We propose a specific solution to this uncertainty in distance in which we resize the training images to an average size but perform detection with several rescaled versions of the original template. The scales that we use are 80%, 100%, and 120% of the size of the original OT-MACH filter. After computing the correlation at each scale, we choose the scale that produces the maximum correlation peak.

Tracker

While the vehicle detector locates a vehicle in a single frame, the tracker is intended to follow the vehicle in the subsequent frames to determine its current and next position, velocity, and acceleration. The tracker is initialized using the image rectangle identified by the vehicle detector. While the detector uses only the prior knowledge of the appearance of a vehicle, the tracking stage ignores the prior knowledge and instead exploits the temporal and spatial consistency of appearance from frame to frame. We use a modified version of the "real-time edge-enhanced dynamic correlation tracker" (Ahmed *et al.* 2007), which is based on cross-correlating an adaptive template within a dynamic search window around the vehicle's predicted position. Since we intend to develop a system that can operate effectively in a real environment, we modify this tracking algorithm with several enhancements that specifically address the sources of noise and uncertainty that occur in a less constrained setting. We describe each of these problems and our solutions in the following three subsections.

Background Clutter

Every intersection will have a different background appearance because of buildings, trees, or other unanticipated objects. Since the appearance-based tracker primarily relies on edge information, background objects that have strong edges can dominate the measure of cross-correlation. The undesirable effect of this problem is very easy to identify; the
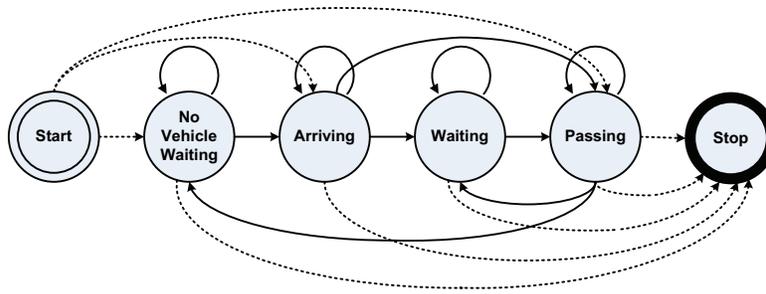
Figure 4: Finite-state-machine intersection model.

tracking window will 'get stuck' on the background when the vehicle passes by a location with strong edges.

We propose the following solutions to this problem. First, we apply a Gaussian weighting window to the template before computing the cross-correlation so that the pixels near the center of the template get more weight than the pixels far away from the center. This method is effective because tracked objects are usually at the center of the template, while the background clutter usually occurs around the object. Second, we update the template pixels using an IIR low pass filter to smoothen the transitions and to nullify the effects of the nearby clutter pixels which are short-lived as compared to the object pixels which are long-lived. More details are given in (Ahmed *et al.* 2007).

Intermittent Occlusions

As a vehicle moves through the scene, it can be briefly occluded by objects like a stop sign, pedestrians, or other vehicles passing the intersection. As with background clutter, this can cause the tracker to follow the occluding object instead of the vehicle. We compensate for this problem by applying a Kalman filter to the trajectory of the vehicle, using knowledge about the consistency of a vehicle's motion to account for uncertainty in the correlation process. We use a "constant acceleration with random walk" model for the dynamic system in the Kalman filter, since this gives better accuracy than the "constant velocity with random walk" model. Also, we dynamically change the size of the correlation search window and the learning rate for updating the vehicle template, since a vehicle is more likely to change in appearance when it is moving quickly.

Changing Scale

When the vehicle moves perpendicularly to the image plane, it will appear to grow larger or smaller. The adaptive template updating can already correct for this, but slowly and ineffectively. We propose addressing this uncertainty explicitly by using the following two strategies.

First, we correlate the search window with three rescaled versions of the template, at 90%, 100% and 110% of the original size. We compare the highest peaks for each of these three correlations, and choose the scale that produces the maximum value. This achieves a broad but coarse compensation for a rapid change in scale.

Second, we split the template in nine non-overlapping rectangular sections and get their mean values. Then, we compare the mean value of an outer section with those of the middle and the opposite sections. Finally, we perform a weighted voting mechanism in which each section gives a weighted vote whether the corresponding side of the rectangular template should be moved outward or inward. This way, the template expands or shrinks according to the size and shape of the object. The mechanism tries to get the vehicle at the center of the template, especially when the vehicle is smaller than the template, or when it is a little drifted from the center.

While tracking a vehicle, we determine the ratio of the size of the updated template to the size of the previous template. Thus, we obtain a relative estimate of the vehicle's distance from the camera, which is used to determine the behavior of each vehicle by the finite-state-machine discussed in the next section.

Finite-State-Machine Intersection Model

The tracker provides us with kinematics information about the detected vehicle in each view. The goal of the intersection model is to use this information, along with prior knowledge about the geometry of the scenario, to determine which cars have the right of way and to signal the autopilot when it is the UGV's turn to go.

We describe the state of the traffic in each view with a finite-state-machine consisting of four states: *No Vehicle Waiting*, *Arriving* at the intersection, *Waiting*, and *Passing* the intersection. The relationships between these states are shown graphically in Figure 4. The state transitions are controlled by rules that operate on the dynamics information from the tracker.

The state-machine transitions from *No Vehicle Waiting* to *Arriving* when a new vehicle is detected. After arriving at the intersection, the vehicle can be in two possible states: either *Waiting* or *Passing*. The model transitions from *Arriving* or *Passing* to *Waiting* when the vehicle stops moving, i.e. when the velocity and acceleration of the vehicle in the X and Y directions as well as the change in scale ratio drop below some threshold. As a guard against spurious state transitions, the vehicle must be still for about 15 frames (half a second) before it is considered *Waiting*. While in the *Arriving* state, the vehicle transitions directly into the *Passing*

	Road			Total	%
	Left	Front	Right		
Visible	62	55	53	170	
Detected	61	50	50	161	94.7
Misdetected	1	5	3	9	5.3
False Alarm	0	0	0	0	0

Table 1: Detection results in uncontrolled traffic.

	Road			Total	%
	Left	Front	Right		
Success	4	2	3	9	90
Failure	1	0	0	1	10

Table 2: Results of experiments under autonomous control. In each experiment, the driver of the other vehicle arrived from one of the three roads.

state if it crosses a spatial threshold, i.e. if the X coordinate crosses a threshold in the left and right views, or if either X or Y coordinates cross a threshold in the front view. A transition from *Passing* to *Waiting* is permitted because the vehicle may unexpectedly have to wait due to some disturbance, such as a pedestrian, while it is passing the intersection. If the vehicle starts moving again, then it transitions back into the *Passing* state. Once in the *Passing* state, the finite-state-machine transitions again to *No Vehicle* once the vehicle reaches the edge of the frame.

Since the vision system is turned on by the autopilot once the UGV has stopped at the intersection, other vehicles may already be in the scene. If a vehicle is detected in the first frame, we allow the state machine to begin with the *Arriving* or *Passing* state depending on the position of the vehicle in the frame. Similarly, the autopilot may turn off the vision system while the state machine is in any state.

To decide when to cross the intersection, our system combines the traffic information from all the three views. If the other vehicle is already *Waiting* when the UGV arrives at the intersection, then it has the right-of-way, and the UGV is commanded to wait for it to leave the scene before proceeding. If at any time a vehicle is *Passing* the intersection, the UGV is commanded not to cross. Since a vehicle will reach the edge of the frame before it has crossed the intersection, the UGV waits for two seconds after the finite-state-machine stops indicating that the vehicle is passing. Any vehicle that is *Arriving* or *Waiting* after the UGV arrives does not have the right-of-way, so we assume they will let the UGV pass.

Results

We tested the low-level components of our system by parking our UGV at several real four-way intersections at which one of the roads was a dead end so we did not interfere with the traffic. It is easy to objectively evaluate the vehicle detector since the total number of visible vehicles is a discrete and unambiguous value. Table 1 shows the number of successfully detected vehicles, as well as the false alarms and misdetections for each camera view. The trade-off between false alarms and misdetections is controlled by the parameters and thresholds. We favored misdetections over false alarms, since if the tracker 'got stuck' on a patch of stationary background clutter, the UGV would wait indefinitely. Since these results were gathered from several intersections with uncontrolled traffic, they demonstrate the robustness and generalization of the detection system.

For testing the entire system as a whole, we staged each experiment as a coordinated event where the UGV and another vehicle (operated by a human driver) would approach the intersection in an agreed-upon order. The results for these experiments are shown in Table 2. If the safety driver (in the UGV) had to override the autopilot to avoid a collision, or if the UGV did not begin to move, then the experiment was marked as a failure. In most cases (90%), the UGV effectively waited for the other vehicle to pass when the other vehicle arrived first, and then automatically proceeded through the intersection.

An annotated sequence of frames from one experiment is provided in Figure 5, which illustrates how our system successfully handles the intersection. Since there were no vehicles on the front and right roads in this instance, we only show frames from the left camera (processed by our software) and from an extra camcorder that we mounted inside the UGV to show the automatic motion of the steering wheel and the path of the UGV. In each frame from the left camera, the bounding box indicates the detected vehicle. The overlay in the bottom left corner shows either the OT-MACH filter (in detection mode), or the updated template (in tracking mode). The first line of text indicates the maximum correlation value and the coordinates of the centroid of the template. The second line of text indicates the scale of the OT-MACH filter that detected the vehicle. The third line of text indicates the decision of the finite-state-machine.

Conclusion

UGVs will have to develop the capability to safely cross intersections with stop signs if they are going to be deployed in urban environments. In this paper, we provide some analysis of this specific problem and propose a complete computer vision solution that combines a vehicle detection-and-tracking algorithm with a finite-state-machine model of the traffic near the intersection to automatically navigate the UGV through the intersection. The initial experiments of our system suggest that a computer vision approach can be the basis of an effective solution, but a truly robust system will need to better deal with subtle uncertainties in such a scenario. For example, we currently assume that the vehicle will take no more than two seconds to pass the intersection, but it is possible that the vehicle will unexpectedly stop while it is outside the view of the camera. Our next work will be to address this problem by combining the output from the computer vision system with other systems, like a laser range finder, which can detect obstacles in the immediate path of the UGV.

(a) The UGV is arriving at the intersection, but another car is already waiting on the left road.

(b) The UGV stops and turns on the computer vision system. The system detects that the car is at the intersection and commands the UGV to wait.

(c) The car begins to pass the intersection.

(d) The car has exited the view of the left camera, although it is still visible in the video from the camcorder.

(e) Two seconds later, the UGV begins to pass the intersection, and the computer vision system is turned off.

Figure 5: Annotated sequence of video frames from an experiment under autonomous control of the UGV. The left column shows some frames from the left camera (processed by our system), and the right column shows the corresponding frames from an additional camcorder that was mounted inside the UGV.

References

Ahmed, J.; Jafri, M. N.; Shah, M.; and Akbar, M. 2007. Real-time Edge-enhanced Dynamic Correlation and Predictive Open-loop Car-following Control for Robust Tracking. Accepted in Machine Vision and Applications Journal, Manuscript submission ID MVA-May-06-0110.

Chun, W.; Faukoner, J.; and Munkeby, S. 1995. UGV Demo II Reuse Technology for AHS. In *Proc. IEEE Intelligent Vehicles Symposium*.

Lewis, J. P. 1995. Fast Normalized Cross-Correlation. Industrial Light and Magic.

Mahalanobis, A.; Kumar, B. V. K. V.; Song, S.; Sims, S.; and Epperson, J. 1994. Unconstrained Correlation Filters. *Applied Optics* 33(17).

Matthies, L., and Brown, E. 1996. Machine Vision for Obstacle Detection and Ordnance Recognition. In *Proc. AUVSI*.

Matthies, L.; Kelly, A.; Litwin, T.; and Tharp, G. 1995. Obstacle Detection for Unmanned Ground Vehicles: A Progress Report. *Robotics Research 7 Springer-Verlag*.

Matthies, L. 1992. Stereo Vision for Planetary Rovers: Stochastic Modeling to Near Real-Time Implementation. *IJCV* 8(1).

Porikli, F. 1995. Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces. In *Proc. CVPR*.

Rasmussen, C. 2002. Combining Laser Range, Color, and Texture Cues for Autonomous Road Following. In *Proc. ICRA*.

Refregier, P. 1991. Optimal Trade-off Filters for Noise Robustness, Sharpness of the Correlation Peak, and Horner Efficiency. *Optics Letters* 16(11).

Shashua, A.; Gdalyahu, Y.; and Hayun, G. 2004. Pedestrian Detection for Driving Assistance Systems: Single-frame Classification and System Level Performance. In *Proc. IEEE Intelligent Vehicle Symposium*.

Talukder, A.; Manduchi, R.; Matthies, L.; and Rankin, A. 2002. Fast and Reliable Obstacle Detection and Segmentation for cross country navigation. In *IEEE Intelligent Vehicle Symposium*.

Vidal, R.; Shakernia, O.; Kim, J.; Shim, H.; and Sastry, S. 2001. Multiple Agent Probabilistic Pursuit-Evasion Games with Unmanned Ground and Aerial Vehicles. In *IEEE Trans. on Robotics and Automation*.

Zhang, Z.; Weiss, R.; and Hanson, A. R. 1994. Qualitative Obstacle Detection. In *Proc. IEEE CVPR '94*.

Zhou, H., and Chao, T.-H. 1999. MACH Filter Synthesizing for Detected Targets in Cluttered Environments for Gray-scale Optical Correlator. In *Proc. SPIE*, volume 3715.