NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INSECT-INSPIRED MINIATURE AIR VEHICLE OBSTACLE DETECTION

by

David A. Funni

September 2019

Thesis Advisor: Roberto Cristi
Co-Advisor: Monique P. Fargues

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2019	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
INSECT-INSPIRED MINIATURE AIR VEHICLE OBSTACLE DETECTION

5. FUNDING NUMBERS

6. AUTHOR(S) David A. Funni

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

In order to fly autonomously, micro air vehicles (MAV) must understand the surrounding three-dimensional environment. Due to size, weight, and power constraints, the collection and processing of environmental data must be done in the most efficient way possible. Biological and neurological research on insect vision has led to computationally inexpensive techniques for detecting the relative distance to objects. In this thesis we will develop and test an efficient implementation of these techniques to process video captured by a single camera sensor.

14. SUBJECT TERMS
unmanned, MAV, optical flow, computer vision, monocular vision, EMD

15. NUMBER OF PAGES
77

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

INSECT-INSPIRED MINIATURE AIR VEHICLE OBSTACLE DETECTION

David A. Funni
Captain, United States Marine Corps
BSME, Iowa State University, 2010

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2019

Approved by: Roberto Cristi
Advisor

Monique P. Fargues
Co-Advisor

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In order to fly autonomously, micro air vehicles (MAV) must understand the surrounding three-dimensional environment. Due to size, weight, and power constraints, the collection and processing of environmental data must be done in the most efficient way possible. Biological and neurological research on insect vision has led to computationally inexpensive techniques for detecting the relative distance to objects. In this thesis we will develop and test an efficient implementation of these techniques to process video captured by a single camera sensor.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

2D	Two-dimensional
3D	Three-dimensional
BPF	Band-pass filter
EDL	Edge device layer
EMD	Elementary motion detector
ESL	Edge server layer
FLA	Fast Lightweight Autonomy
FOC	Focus of contraction
FOE	Focus of expansion
FPGA	Field programmable gate array
HA/DR	Humanitarian Assistance and Disaster Relief
IMU	Inertial measurement unit
LPF	Low-pass filter
LPTC	Lobula Plate Tangential Cell
MAV	Miniature air vehicle
SLAM	Simultaneous localization and mapping
SWaP	Size, weight, and power constraints
UAV	Unmanned aerial vehicle

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would first like to thank my wife, Yukie, for her encouragement and support throughout my time at Naval Postgraduate School. I would also like to thank my advisors, Dr. Cristi and Dr. Fargues. Dr. Fargues, thank you for the “training” you provided during all of your classes and for your help clarifying my thesis work. Dr. Cristi, thank you for your enthusiasm and passion for teaching and taking your time to build my understanding in all areas of the signal processing field.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Demand for unmanned aerial systems (UAS) for both military and civilian applications is continually increasing. Within the Marine Corps in particular, the recently released "38th Commandant's Planning Guidance" [1] calls for a "significant increase in unmanned systems" including a "family of unmanned aerial systems." The miniature air vehicle (MAV) is one class of UAS that is currently being used for both military and civilian operations. These are generally defined as man portable, with dimensions from 1–50 cm, and therefore have significant size, weight, and power (SWaP) constraints.

Currently, efforts such as the Defense Advanced Research Projects Agency (DARPA) Fast Lightweight Autonomy (FLA) [2] and μBRAIN [3] programs seek to enable MAVs and other small unmanned systems to operate autonomously. Autonomous operation allows MAVs to fly in environments where links to operator control and global positioning satellites (GPS) are unavailable such as within buildings, beneath heavy forest canopies, or searching through damaged urban areas after a natural disaster.

A key component to autonomous flight is obstacle detection and avoidance. The processing of this type of environment awareness, called simultaneous localization and mapping (SLAM), is typically done by combining data from a multitude of sensors, which could include cameras, sonar, light detection and ranging (LIDAR), among others. With added sensors, the computational expense of processing sensor data to form a unified picture of the environment must be balanced with the SWaP constraints for a particular platform. To allow very small and inexpensive MAVs to operate autonomously in cluttered environments, researchers including DARPA [3] have been increasingly looking toward biology for inspiration. The animal kingdom is full of examples of small insects and birds that are able to fly in complicated environments with very limited computational resources. This thesis takes inspiration from the vision system of flies in an attempt to develop a method of obstacle detection for use in autonomous MAV flight.

A. OBJECTIVE

The objective of this thesis is to develop the means by which a small and lightweight autonomous MAV can sense nearby obstacles using a single camera. The MAV system should be capable of fully autonomous flight and therefore not reliant on data links to more powerful computational resources. Due to these constraints and the SWaP limitations of the MAV platform, the obstacle detection method needs to be computationally inexpensive to allow onboard real-time processing.

To accomplish this objective, we will look to biological vision systems due to their performance and computational efficiency. In particular, the ability of flies to use limited computational power to navigate and avoid hazards has been extensively studied over the past seven decades. This thesis will use the biological processes present in the fly vision system as a model from which to develop a simple monocular camera-based obstacle detection scheme. The developed system should be able to detect stationary objects while the MAV is in steady forward flight. Furthermore, the system should be computationally inexpensive and parallelizable for implementation on a small field programmable gate array (FPGA) or application specific integrated circuit (ASIC).

B. THESIS LAYOUT

We begin our research with a review of relevant concepts in Chapter II. First, methods for converting three-dimensional real-world object coordinates into two-dimensional image frame representations are reviewed along with the transformations involved when dealing with camera movement. Next, the concept of optical flow is introduced along with brief descriptions of commonly used algorithms. Finally, a review of research into the biological motion detection systems of flies is presented to form the theoretical underpinnings of our proposed obstacle detection method.

Chapter III discusses the theory behind the proposed method using concepts common to the signal processing field. We begin by analyzing the biologically-inspired elementary motion detector (EMD) in continuous time. We then develop a theory for the discrete time operation of a single EMD. Next, we look at the implementation of a one-

dimensional (1D) array of EMDs. Finally, we discuss a two-dimensional (2D) EMD array that can be used to process video frames produced by a typical camera.

In Chapter IV we present the findings from the implementation of the camera-based EMD. We start by testing various parameters using a scene specifically constructed to respond well to our system. After consolidating our initial findings, we test the EMD on videos captured in natural environments and analyze the effectiveness of the EMD system.

Finally, conclusions and recommendations for further are presented in Chapter V.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND AND LITERATURE REVIEW

First, we will introduce some of the theory and terminology to form a foundation with which to discuss our proposed biologically-inspired motion detector. We start by discussing the projection of three-dimensional real-world points to the two-dimensional image plane as well as the mathematics required to compute rigid body motion. Then, we will discuss a selection of conventional algorithms used to calculate optical flow, or the 2D vector field created in the image frame by motion. Lastly, we will discuss the current state of understanding of insect visual motion processing. The concepts presented in this chapter will provide a common footing for the discussion of our proposed system in later chapters of this thesis.

A. 3D TO 2D PERSPECTIVE-PROJECTIVE MODEL WITH RIGID BODY MOTION

It is common in image processing and computer vision applications to assume, as a first approximation, an ideal pinhole camera model so that lens effects can be ignored. In this model, all rays entering a camera travel in straight lines through an optical center *o* of the lens and intersect with the two-dimensional image plane. The distance between *o* and the image plane is the focal length, labeled *f*. A point *p* in three-dimensional space relative to the camera with origin *o* is given by coordinates $\boldsymbol{X}_C = \left[X_C, Y_C, Z_C \right]^T$ where the z-axis is parallel to the optical axis, or the axis normal to the image plane through *o*. Point *p* has a corresponding image point $\boldsymbol{x} = \left[x, y \right]^T$ on the image plane. A diagram of the relationships is shown in Figure 1.

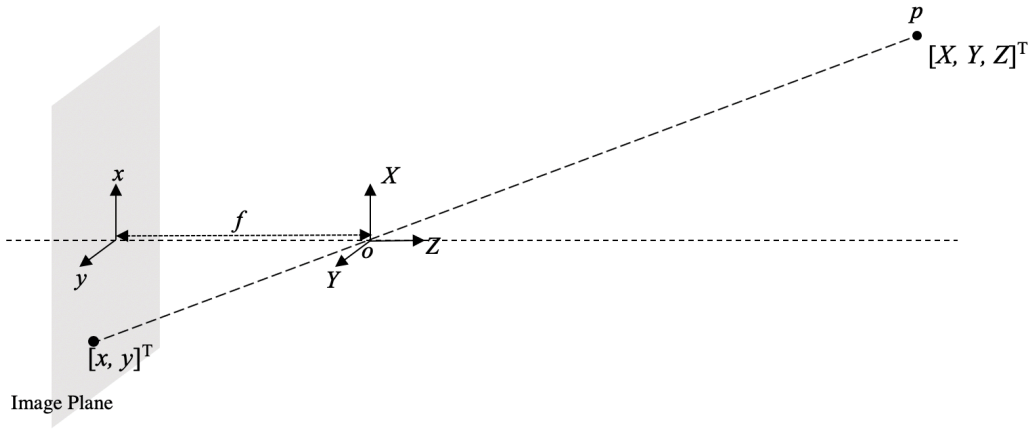

Figure 1. Pinhole camera model, rear projection.

From the above description the perspective projection relationships can be observed:

$$x = -f\frac{X}{Z} \tag{1}$$

$$y = -f\frac{Y}{Z}. \tag{2}$$

Although the coordinate Z along the focal axis is actually the distance with respect to the focal point, it can be approximately considered a distance from the image plane, since usually $Z \gg f$. The negative signs in equations (1) and (2) are due to the inversion of the image through the lens. In order to remove this effect, an equivalent model can be constructed with the image plane placed distance f in front of o. The resulting system is shown in Figure 2 and the equation in vector form is:

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix}. \tag{3}$$

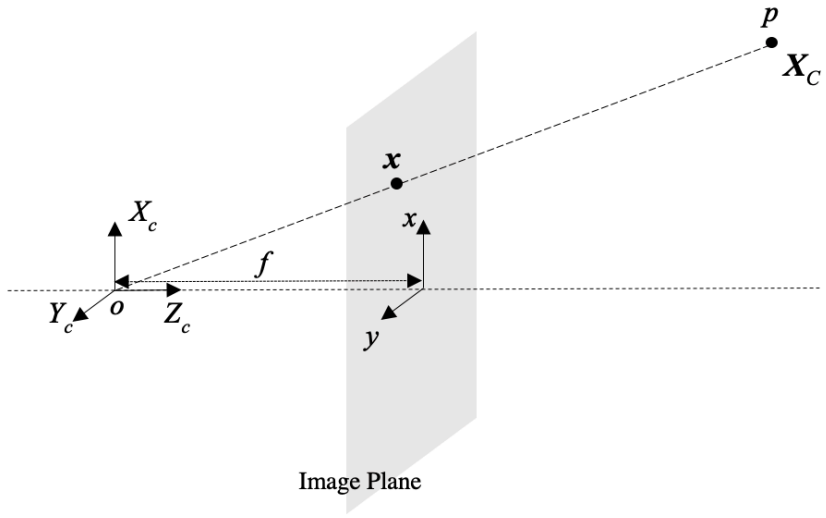

Figure 2. Pinhole camera model, front projection.

In homogeneous coordinates we represent the location of a point *p* with respect to the camera as $\hat{\boldsymbol{X}}_{\boldsymbol{C}} = \left[X_C, Y_C, Z_C, 1 \right]^T$ and $\hat{\boldsymbol{x}} = \left[\boldsymbol{x}, \boldsymbol{y}, 1 \right]^T$ so that equation (3) can be rewritten as:

$$Z_C \hat{\boldsymbol{x}} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \hat{\boldsymbol{X}}_{\boldsymbol{C}}. \tag{4}$$

Here $\boldsymbol{Z_C}$ is the depth of point *p* and is a positive real number, i.e. $Z_C \in \mathbb{R}_+$.

Rigid body transformations can be applied to the system, which represents either a fixed scene with a moving camera, or alternately a moving scene with respect to a fixed camera. Given a point *p* in world coordinates represented by $\boldsymbol{X}_{\boldsymbol{W}} = \left[X_W, Y_W, Z_W \right]^T$, a rigid body transformation can be applied to convert world coordinates into camera coordinates as shown in equation (5):

$$X_C = R_{WC} X_W + T_{WC} \quad \in \mathbb{R}^3. \tag{5}$$

The "WC" subscripts in equation (5) represent the world to camera coordinate transformations. Here, T_{WC} is the translation vector, or the distance in all three axes between *p* and ***x***, the point in space and its representation on the image plane. The rotation matrix, R_{WC}, is given by rotations of the camera with respect to the world frame by the Euler angles ϕ, θ, and ψ that correspond to roll, pitch and yaw. This matrix can be computed by multiplying rotation matrices about each axis, in a preassigned order determined by convention, which leads to:

$$R_{WC} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

Again, using the homogeneous representation equation (5) can be written as:

$$\hat{\boldsymbol{X}}_C = \begin{bmatrix} R_{WC} & T_{WC} \\ 0 & 1 \end{bmatrix} \hat{\boldsymbol{X}}_W \quad (7)$$

which allows us to combine equations (4) and (7) to produce the full model of rigid body motion:

$$Z_C \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{WC} & T_{WC} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}. \quad (8)$$

This three-dimensional movement of the camera with respect to the world frame as defined in equation (8) is called ego-motion.

B. OPTICAL FLOW

Optical flow is the technique of detecting a two-dimensional motion vector created by objects in the image plane due to relative motion between an observer and the environment. In camera-based systems, the flow is calculated between at least two successive frames of a video sequence taken at times t and $t + \Delta t$ where Δt is the inverse

of framerate. The output of the optic flow calculation is a field of velocity vectors showing the estimated motion of point between frames.

Note that the resulting optic flow field will have large velocity vectors toward the periphery that generally point away from the center of the image when a camera is moved by forward translation while pointing in the direction of movement. At the exact center a singularity is present called the Focus of Expansion (FOE). A similar singularity can be found if the camera is flipped to point in a retrograde direction, however the motion vectors will point toward it. This singularity is called the Focus of Contraction (FOC).

Depth, or the distance from the camera to the point in the world frame, can be inferred and is dependent on both the optical flow vector and its location in relation to the focus of expansion (or contraction) [4]. An analogy can be made using the example of a passenger in a car. If the passenger looks to the side of the car, nearby objects will appear to move very quickly, while objects further away appear to move slowly. Looking directly to the front, another car far in the distance may appear to be stationary due to the effect of the focus of expansion. The calculation of depth, or alternately the time to contact, will be discussed further in Chapter III.

Optic flow algorithms are often compared by the spatial density of the flow field produced. Various algorithms for calculating optical flow have been developed with differing spatial density, motion estimation accuracy, and computational complexity. The most common of these algorithms are the differential and block matching types. We will first give a brief description of these methods to provide some background before presenting the biologically-inspired system used in this thesis.

1. Differential

Differential methods calculate a dense flow field, which includes sub-pixel motion resolution. Consider a point p in three-dimensional world coordinates. The representation of p in the image frame is assumed to have the same intensity in consecutive frames such that:

$$I(x,y,t) = I(x+dx, y+dy, t+dt) \tag{9}$$

where I is the intensity of a pixel in the image frame [5]. If a Taylor series expansion is applied to this equation the standard optical flow equation is obtained:

$$\begin{bmatrix} \frac{\partial I(x,y,t)}{\partial x} & \frac{\partial I(x,y,t)}{\partial x} \end{bmatrix} \begin{bmatrix} v_x(x,y,t) \\ v_y(x,y,t) \end{bmatrix} = -\frac{\partial I(x,y,t)}{\partial t}. \tag{10}$$

The terms v_x and v_y present in the above equation represent the velocity component in both the x and y directions respectively. Considering an image made of discrete pixels, the spatial intensity gradients are generally calculated by convolving the image with a Sobel kernel with the appropriate directional response. The temporal derivative can be obtained by convolving a $[1,-1]$ kernel over the time dimension for each pixel. This temporal operation has the effect of taking the difference between two subsequent frames.

In attempting to solve for equation (10), we note that the issue arises that there is only one equation but two unknowns, v_x and v_y, for each point. Two methods are commonly used to solve this dilemma, the Horn-Schunck [6] and the Lucas-Kanade [7] methods. The Horn-Schunck method solves for pixel velocities by minimizing an error function using an iterative approach. The Lucas-Kanade method solves equation (10) by assuming regions of the image frame have constant velocity. Noise is then reduced by applying a threshold to remove the effect of small eigenvalues. For robotic vision applications, it is important to note that these methods are limited to calculating the optical flow due only to small pixel motions between subsequent frames. To calculate larger displacements, block-matching methods are commonly used.

2. Block-Matching

Another class of algorithms commonly used in real time robotic time-to-contact calculations is the block-matching or region matching type [8]. This class works well if an object shifts by large amounts between frames, which could be due to framerate restrictions or the proximity of the object to the camera. Block-matching is based on a two-dimensional window function, W, and designed to match a region in one frame to the displaced equivalent image in a subsequent frame by minimizing a cost function. One common cost

function for block-matching algorithms is the sum-of-squared differences [9]. An example of this type of block-matching calculation is given by:

$$SSD(\boldsymbol{x},\boldsymbol{d}) = \sum_{i=-n}^{n}\sum_{j=-n}^{n} W(i,j)\left[I_1\left(x_1+i,y_1+j,t\right) - I_2\left(x_2+i,y_2+j,t+\Delta t\right)\right]^2 \qquad (11)$$

where *SSD* stands for sum-of-squared differences as a function of pixel location, $\boldsymbol{x}$, and the displacement, $\boldsymbol{d}(\boldsymbol{x}) = \left[d_x, d_y\right]^T$, of the matched region between the two frames I_1 and I_2. This displacement vector is taken as the motion vector of that region [9]. This type of algorithm is able to effectively calculate optical flow due to large displacement between frames, however it is computationally expensive as it must search the entire image for a matching region for each frame.

As differential and block-matching forms of optical flow calculation have limitations, specifically in their computational expense, we will next look at a biological solution.

C. BIOLOGICALLY-INSPIRED MOTION DETECTION

Many insects, including the much studied blowfly, use a form of optical flow processing to detect nearness to objects in the field of view [4], elevation and ground speed [10], and time to obstacle contact [8]. Over the past six decades researchers have studied the mechanisms of this "visual motion pathway" between the eye and brain and have discovered similarities between in many different insect, bird, and even mammalian vision motion processing [11]. In this section we will discuss the general anatomy of a fly visual motion pathway, which we modeled in our research.

A fly compound eye is composed of an array of thousands of facets called ommatidia (singular ommatidium). Each ommatidium contains a lens that focuses incoming light onto a cluster of a number of (usually eight) photoreceptor cells. The eight photoreceptors are arrayed with six (typically labeled R1-R6) spaced hexagonally around the perimeter and two (R7 and R8) toward the center of the ommatidium [11], [10]. This arrangement is shown in Figure 3. The R7 and R8 cells of each ommatidium have randomly

assigned spectral sensitivities and play a role in color vision [10], similar to cones in the human eye. The R1-R6 photoreceptors all have the same spectral sensitivity and are used for motion detection [10], [12].

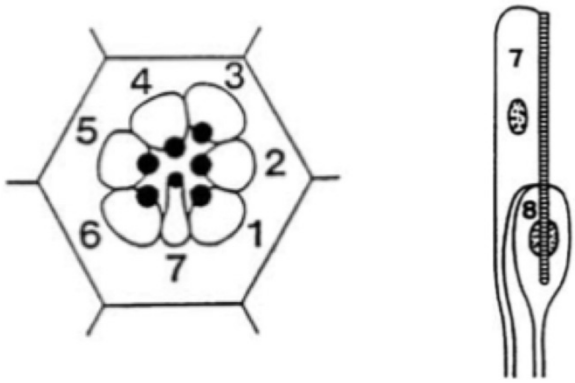

Figure 3. Photoreceptor arrangement in an ommatidium. Source: [12].

The full visual motion pathway, as shown in Figure 4, begins with the receipt of light intensity signal from the R1-R6 photoreceptors. As the photoreceptors are stimulated, the signal is passed through several vision processing layers called the lamina, medulla, lobula, and lobula Plate [11], [13] before a final control signal is sent to the wings, legs or head of the fly.

Signals from corresponding photoreceptors of neighboring ommatidium are sent to the Lamina where they are split into parallel paths as shown by cells L1-L4 in Figure 4. Critically, cells L1 and L2 respond to intensity changes with the same transient response. The characteristics of this transient response are thought to behave as either a high-pass [12], [14] or band-pass [4], [10] filtered form of the initial photoreceptor signals. This filtering, in the time domain, has the effect of extracting contrast information from a moving scene exciting the visual stimulus.

This concept is what distinguishes the traditional optical flow computation described in the previous section from the biologically-inspired optical flow described next. The traditional computation requires differentiation in both the two-dimension image

domain, for edge detection, and time domain. The biological approach is entirely in the time domain and well suitable to parallel computation.

The signals from the L1 and L2 cells are passed to medulla interneurons Mi1 and Tm1-9 before proceeding on to the T4 and T5 cells. The effect of the interneurons is not relevant to our study, however the fact that there are two distinct pathways is of importance. The first pathway, from L1 to the T4 cells has been shown to respond to rising edge or ON signals [11]. The other pathway, from L2 to the T5 cells responds to falling edge or OFF signals [11]. The T4 and T5 cells are wired in such a way as to respond to intensity changes relating vertical and horizontal contrast edges in the environment, thus providing a directional motion response. This directional information is combined in the lobula plate to produce the magnitude and direction of motion [11], [10]. The lobula plate tangential cells (LPTC) pool this motion data and output control signals to the legs, wings, and head of the insect [10], [11]. The processing of visual motion data as described is called an Elementary Motion Detector (EMD).

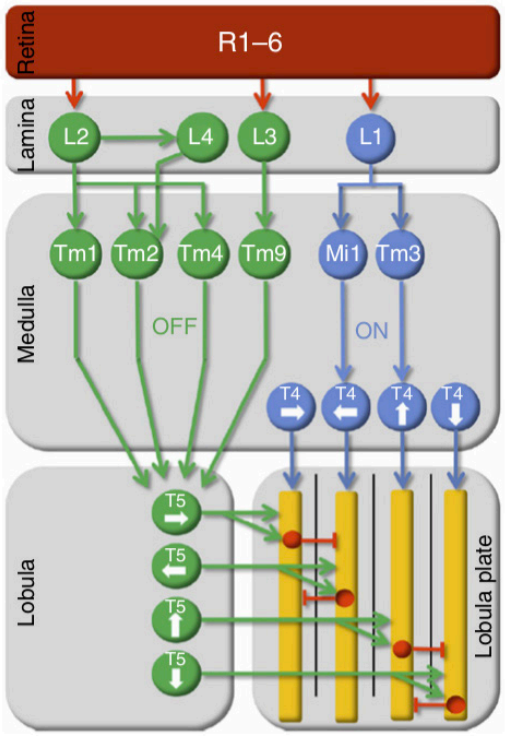

Figure 4. Fly visual motion pathway. Source: [11].

The exact workings of the visual motion pathway have been the focus of research for many decades and they continue to the present. Much of this research has treated the image processing pathways as a "black box" by observing insect responses to known inputs. In the 1950s and 1960s this was done by observing insect motion while various optical stimuli passed in front of the insect. This research resulted in the development of the Reichardt correlator EMD [15] (sometimes called Reichardt-Hassenstein detector) as shown in Figure 5.

In the Reichardt EMD model, the stimulus from two adjacent photoreceptors are each split into multiple signal paths. One of the paths for each photoreceptor is temporally filtered with a high-pass filter (F block in Figure 5) to produce a delayed signal. For each photoreceptor output, the delayed signal of one photoreceptor and the non-delayed signal of an adjacent photoreceptor are combined. This combination forms a directional response and is analogous with the effect of the T4 and T5 cells of the fly eye. Directional responses of arrays of these simple detectors can be combined, as in the LPTC of the fly, to form an

overall view of visual motion. In the next chapter we will develop a mathematical model of the EMD. In particular we will show that motion will be detected by combining the time responses of adjacent neurons, in a fashion that can be easily implemented by elementary operations.

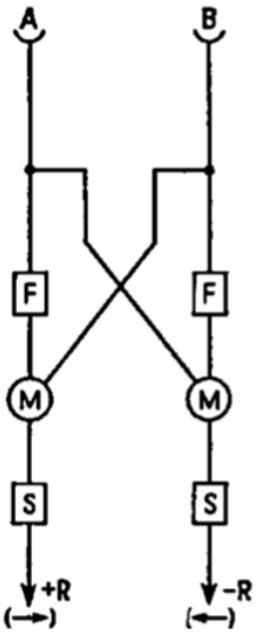

Figure 5. Original Reichardt EMD. Source: [15].

In this chapter we have presented the basic principles that form the foundation of the remainder of this thesis. The concepts of perspective-projective model and the image transformations due to camera ego-motion were presented as they are essential to many computer vision and robotics applications and will be referenced again in Chapters IV and V. We then introduced the concept of optical flow and discussed how relative scene depth can be computed from these motion vector fields. Lastly, we presented the summary of relevant facts on the biological and neuroscientific research into insect motion detection. In the next chapter we will use tools from the signal processing field to further study the EMD and develop our pixelEMD model.

THIS PAGE INTENTIONALLY LEFT BLANK

III. OPTICAL FLOW COMPUTATION BY EMD

In this chapter, we will use the concepts discussed in the previous chapter as well as the tools of the signal processing field to further develop a useful model of the EMD. We will begin by analyzing a single detector in continuous time. We will then convert the model to its discrete time equivalent to move toward a real-world implementation using digital cameras and computers. Following the analysis of a single detector, we will develop a one-dimensional array of EMDs. Lastly, we will develop a two-dimensional EMD array for implementation on video frames captured by a standard camera system.

A. MATHEMATICAL MODEL: CONTINUOUS TIME

As in the basic Reichardt detector, two adjacent photoreceptors are considered for the EMD detector. Call $x_1(t)$ and $x_2(t)$ respective excitations from light, as shown in Figure 6. The signals from these two photoreceptors are processed by first filtering each one with a band-pass filter (BPF) [4], [10]. The derivative action of this filter allows us to detect changes in illumination in the time domain caused by moving objects in front of each receptor. The time correlation between these stimuli from adjacent receptors is what allows the extraction of motion from local changes in illumination due to moving objects.

Next, the output of each BPF is split to provide correlation information, with one pathway passed through a low-pass filter (LPF) and the other not filtered further. The effect of the LPF is to extend the response of the filtered pathway in time [13]. The pathways are then combined in such a way that the LPF response from one photoreceptor is multiplied by the BPF response of the neighboring photoreceptor. The difference in these outputs is taken as the directional response of the EMD designated by $d(t)$ in the Figure 6. For comparison, we also considered the product of these two responses designated by *c(t)* as suggested by Parise [16].

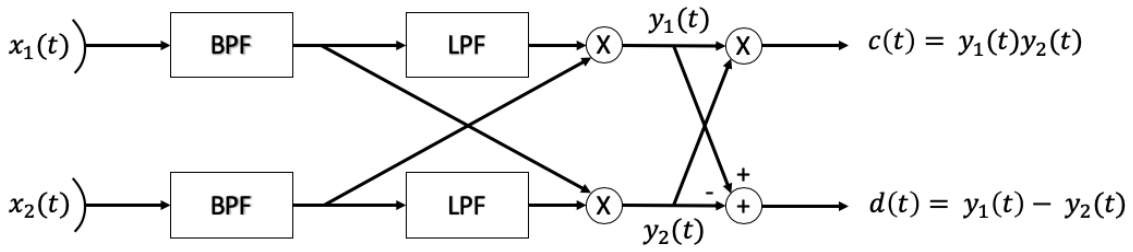

Figure 6. Continuous time EMD model.

In this EMD system, the BPF and LPF are represented by two transfer functions in the Laplace domain:

$$BPF(s) = \left(\frac{1}{1+s\tau_L}\right)\left(\frac{s\tau_H}{1+s\tau_H}\right) \tag{12}$$

$$LPF(s) = \frac{1}{1+s\tau_{LP}}. \tag{13}$$

The BPF is modeled from the activity of the cells of the lamina [4], [10]. As an example of time constants used in the above transfer functions, Schwegmann [4] suggest BPF time constants $\tau_L = 8$ ms and $\tau_H = 20$ ms and a LPF time constant $\tau_{LP} = 40$ ms based on biological tests.

Using equations (12) and (13), the EMD model can be rearranged as shown in Figure 7. This equivalent model is used to simplify later implementation. In the rearranged form, the transfer functions (12) and (13) are replaced by:

$$H(s) = \frac{1}{\left(1+s\tau_L\right)\left(1+s\tau_H\right)\left(1+s\tau_{LP}\right)} \tag{14}$$

$$G(s) = \frac{1}{\left(1+s\tau_L\right)\left(1+s\tau_H\right)}. \tag{15}$$

It should be noted from the previous four equations, the "*s*" term in the numerator of the BPF transfer function defined in equation (12), has been removed in the rearranged model represented by transfer functions (14) and (15). Recall the "*s*" term represents the

derivative function and acts as the high-pass portion of the BPF. In the rearranged system shown in Figure 7 this derivative operation is instead applied prior to the *H* and *G* filters as shown by the "*s*" block. This derivative block responds to discontinuities in intensity due to motion and is a form of edge detection in the time domain. This is an important point, as it results in one of the main limitations of the EMD, that is it only tracks spatial intensity gradients as they move over time. Therefore, to be detected objects must have texture or high contrast patterns to be detected. If no areas of high contrast are present on an object, the EMD will only detect its edges and will be blind to large, smooth, monochromatic surfaces.

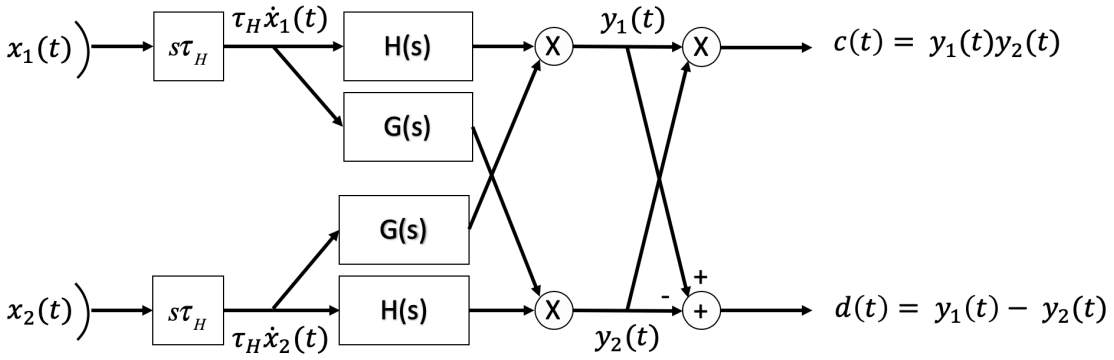

Figure 7. Rearranged continuous time EMD model.

If higher order filter effects are ignored, the edges detected by the derivative function act as unit impulses and result in the approximated filter responses (assuming $\tau_{LP} < \tau_L < \tau_H$):

$$h(t) = L^{-1}\{H(s)\} = \alpha e^{-\frac{t}{\tau_{LP}}}u(t) \tag{16}$$

$$g(t) = L^{-1}\{G(s)\} = \beta e^{-\frac{t}{\tau_L}}u(t). \tag{17}$$

The resulting decaying impulse response is of the form shown in Figure 8.

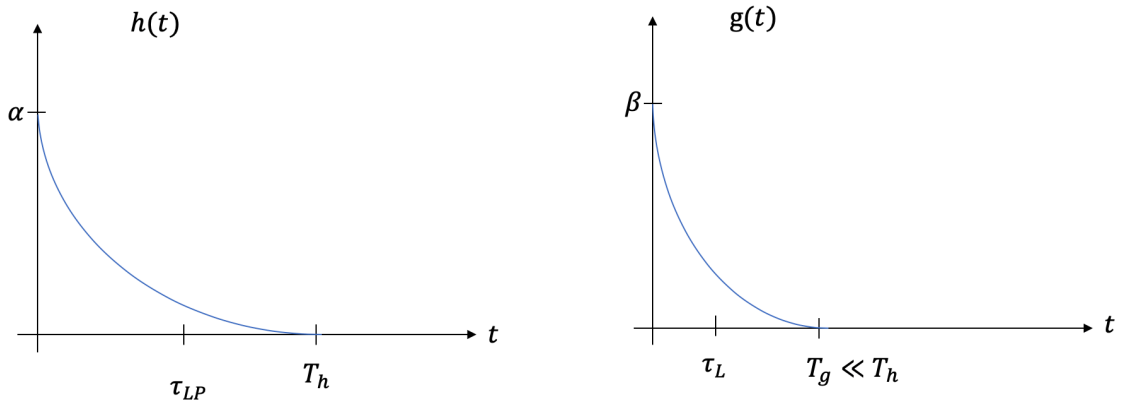

Figure 8. Time domain impulse response.

In order to further understand the system, consider a discontinuity that moves across the image frame and is observed by photoreceptors x_1 and x_2. Since the two receptors are not co-located, there will be a time delay of *T* between the two stimuli. In particular, the resulting photoreceptor responses $\dot{x}_1$ and $\dot{x}_2$ due to a moving edge can be represented by impulses corresponding to contrast around the edge, as:

$$\dot{x}_1 = \delta(t) \tag{18}$$

$$\dot{x}_2 = \delta(t - T). \tag{19}$$

Passing the resulting impulses through the previously described filters yield outputs described by the following expressions:

$$y_1(t) = h(t) g(t - T) \tag{20}$$

$$y_2(t) = h(t - T) g(t). \tag{21}$$

From equations (20) and (21), the time delay *T* that gives the maximum response can be determined on the basis of the following definition:

$$S(T) = \max_t h(t) g(t - T). \tag{22}$$

Equation (22) represents the maximum response of each photoreceptor at the first multiply stage of the EMD (see Figure 7). Using simple filter impulse responses as shown

in Figure 9, the resulting $S(T)$ response is shown in Figure 10. This finding indicates the response amplitude is high when the time delay T is within a certain range given by the width of the response curve $S(T)$ in Figure 10.

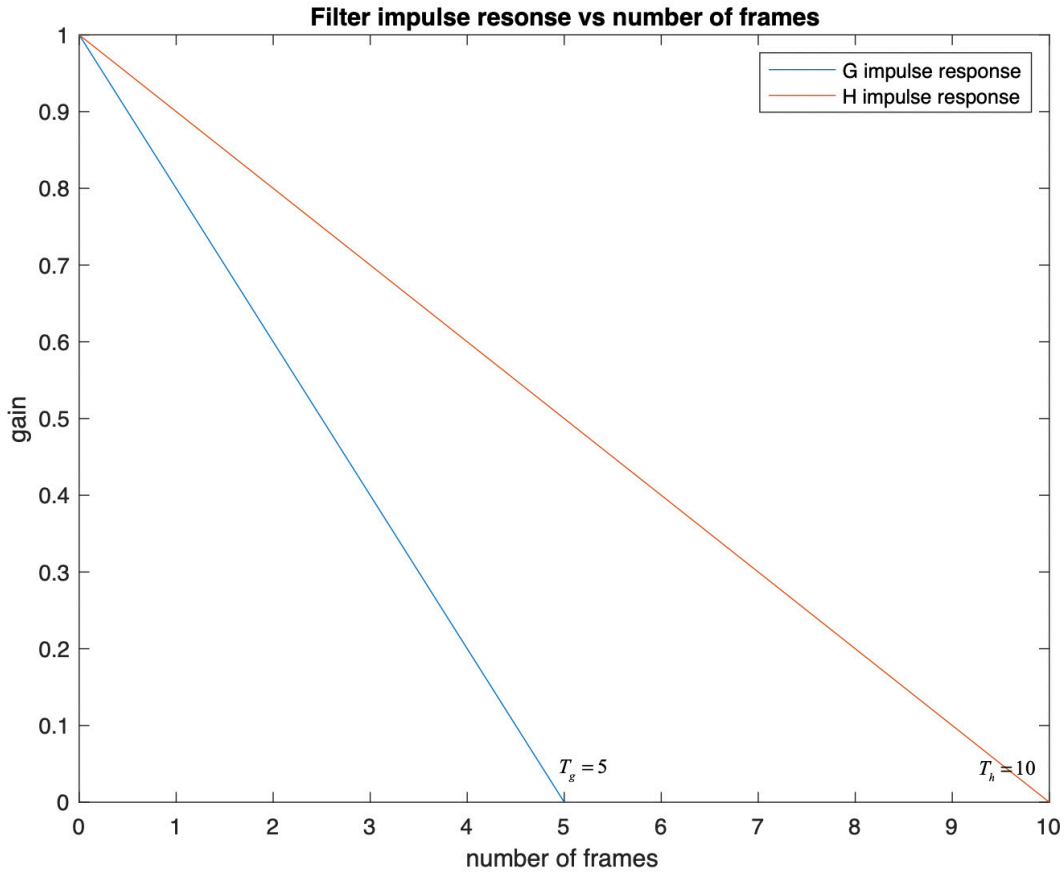

Figure 9. Example filter impulse responses

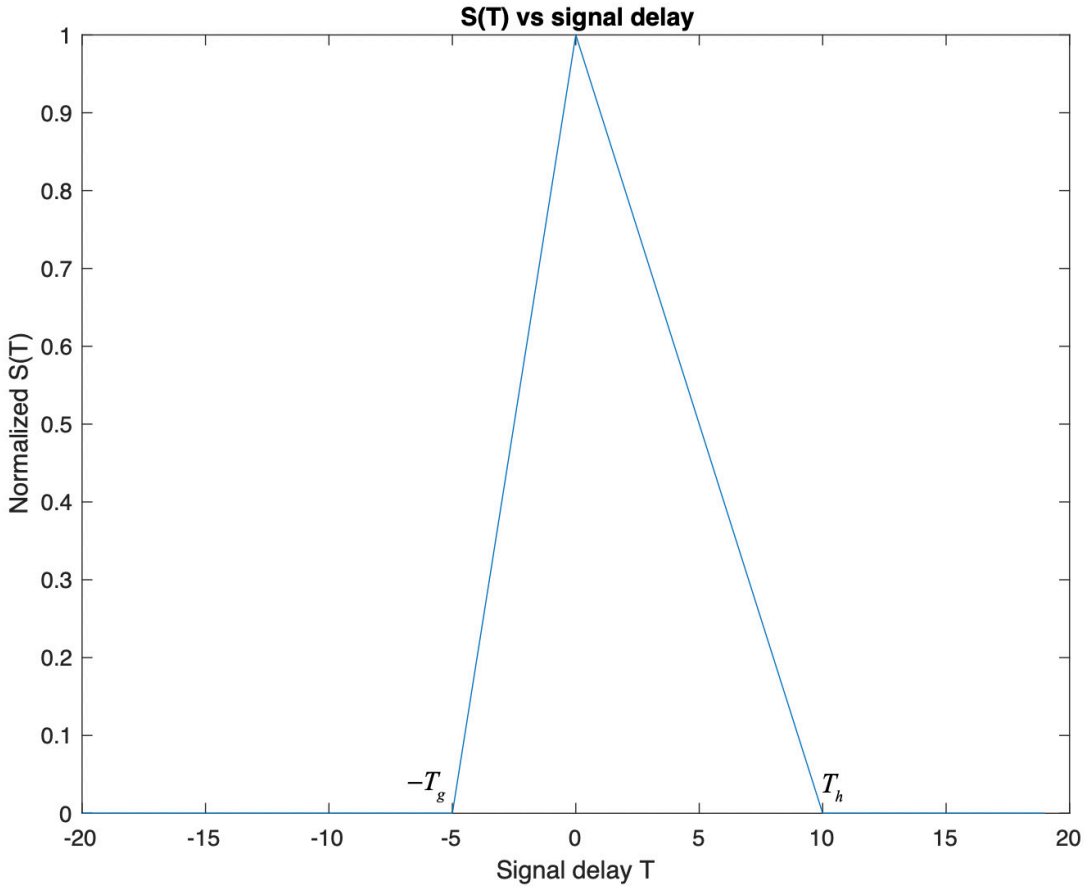

Figure 10. Example plot of equation (22), the maximum response of $y(t)$ as a function of time delay T.

From equation (22), a new function can be defined as in equation (23), which yields the maximum of the product of the signals from the two opposing pathways. A plot of this type of response is shown in Figure 11.

$$c(T) = \max_t h(t)g(t-T)g(t)h(t-T) \tag{23}$$

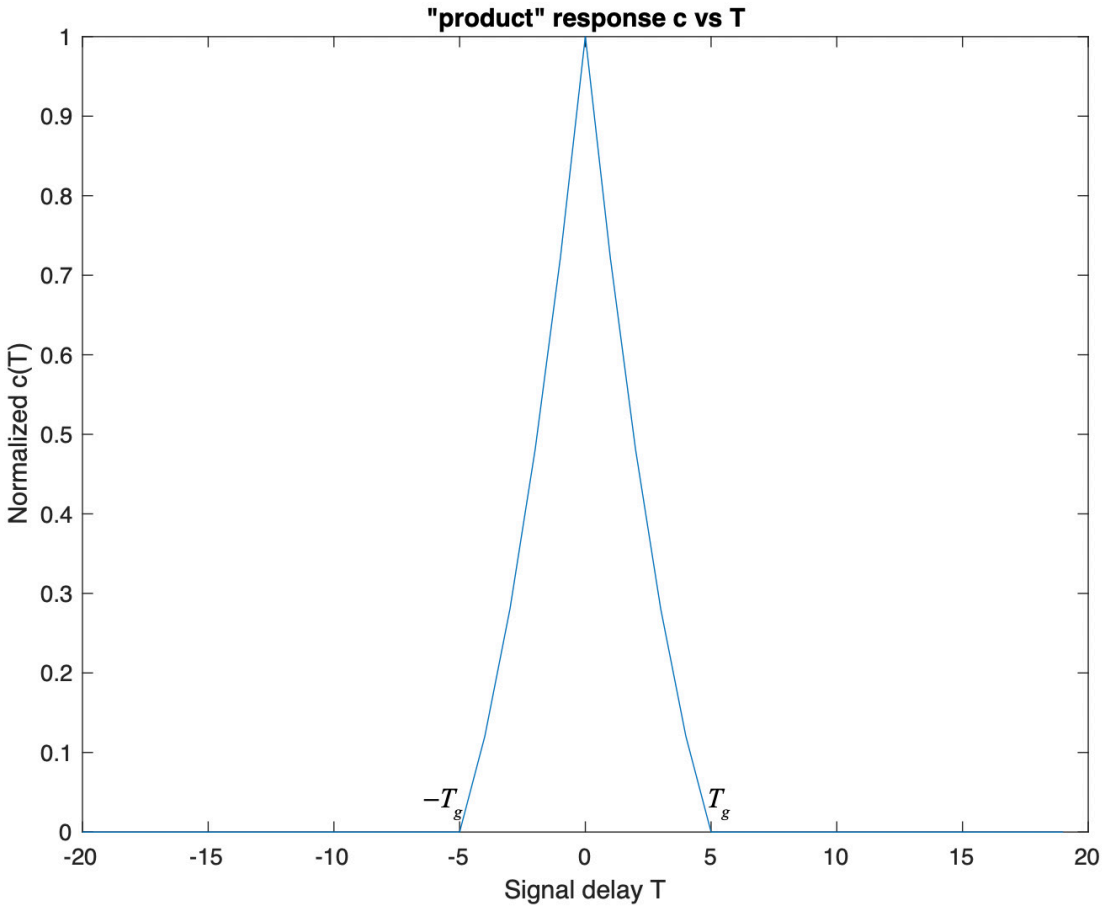

Figure 11. Example plot of equation (23), the maximum response of $c(T)$ as a function of time delay T.

Note that in the plots shown in Figure 10 and Figure 11, the exact shape is not important, only the interval of duration is important as this interval determines the range of time delays that will result in an EMD response. In the case of the product response, $c(T)$, the interval of duration is $|T| < T_g$.

Similarly, the maximum value the EMD difference response, $d(T)$, is given by:

$$d(T) = \max_t \left(g(t)h(t+T) \right) - \max_t \left(h(t)g(t+T) \right) \tag{24}$$

A sample plot of this response as a function of delay T is shown in Figure 12. Again, note the important factor is the interval in which a response is obtained. In this case, the interval is $|T| < T_h$ with the response being positive or negative based on the interval T.

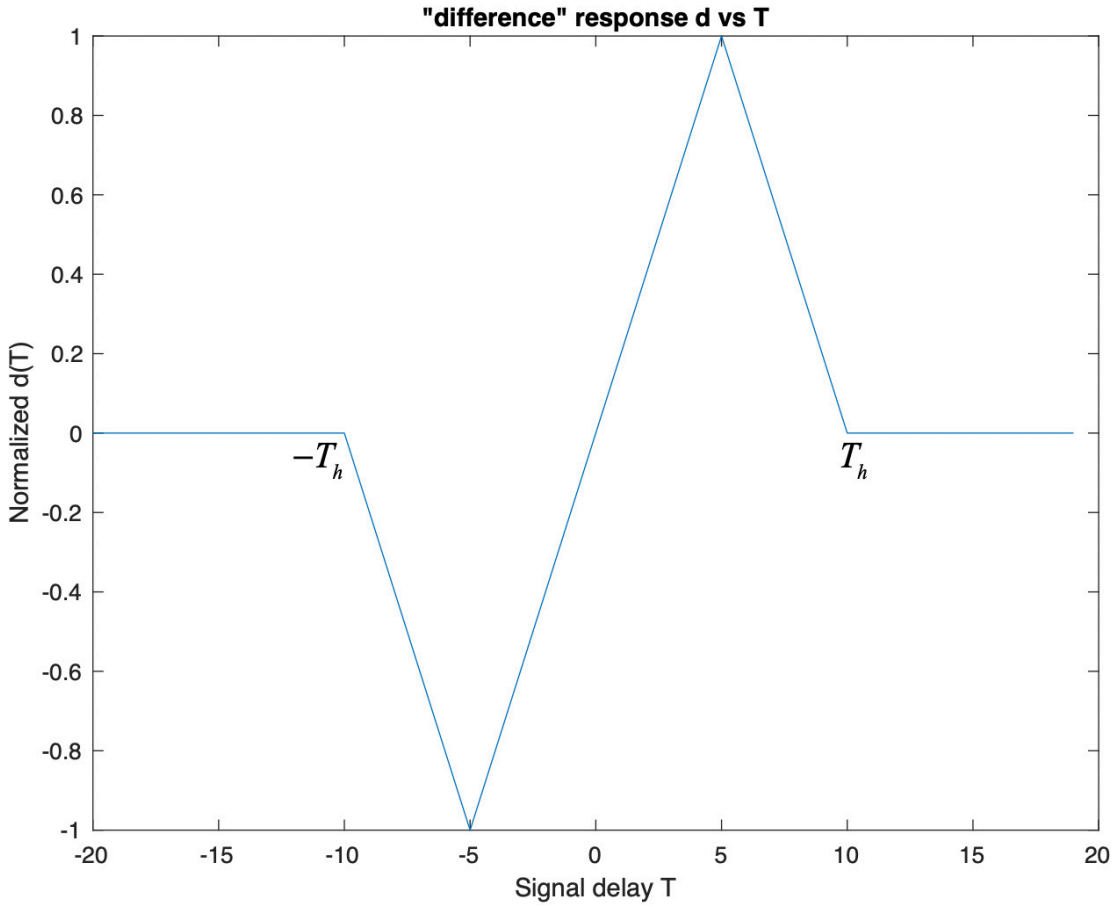

Figure 12. Example plot of the maximum response of $d(T)$ as a function of time delay T.

B. MATHEMATICAL MODEL: DISCRETE TIME

In this section we relate the result of the EMD model with the standard Optical Flow approach.

Taking the front projection pinhole camera model discussed in Chapter II.A, a point *p* in three-dimensional world frame is given by coordinates $\boldsymbol{X} = [X, Y, Z]^T$ where *Z* is the optical axis perpendicular the image plane through the optical center *o*. Point *p* is therefore projected onto the image frame at point $\boldsymbol{x} = [x, y]^T$ where *x* and *y* are coordinates of the point projection to the 2D image frame. The relationship between the point *p* in the world frame and its representation $\boldsymbol{x}$ in the image frame is given by equation (3). Considering only one of the image plane dimensions (here we will consider only the *x* dimension, but the case for *y* is similar), the relationship is given by:

$$x = f\frac{X}{Z} \tag{25}$$

where *f* is the focal length of the camera system. Taking the time derivative of equation (25) leads to:

$$\dot{x} = \frac{\dot{X}}{Z} - fX\frac{\dot{Z}}{Z^2} \tag{26}$$

with $v = -\dot{Z}$ as the velocity of the camera in the direction toward the obstacle. Assuming a stationary object, the first term of the right hand side of (26) is zero. Further since $X = xZ / f,$ from the pin-hole camera model, and setting the focal length to one yields:

$$\frac{\dot{x}}{x} = \frac{v}{Z} = \frac{1}{T_Z} \tag{27}$$

or:

$$\dot{x}T_Z = x. \tag{28}$$

The time T_Z in equation (27) is very important, since, by its own definition, it represents the time needed to come in contact with the obstacle. In discrete time, where we process frames at the frame rate F_0 and period $T_0 = F_0^{-1},$ the time to contact becomes $T_Z = NT_0$ where *N* is the number of frames to contact. Finally, the optical flow for the discrete system can be calculated:

$$\dot{x}T_0 = \frac{x}{N}. \tag{29}$$

From the discussion of the continuous EMD model in the previous section, two adjacent photoreceptors observing the same edge signal at times T_1 and T_2 respectively will yield a "high" EMD product response in the "product" output if $|T_2 - T_1| < T_g \ll T_h$. Similarly, there is a "high" EMD response in the "difference" response when $|T_2 - T_1| < T_h$. If the two photoreceptors are separated by a distance Δx with an average position in the image plane given by $\bar{x}$, then the relation in equation (27) can be approximated as:

$$\frac{\Delta x}{\bar{x}\Delta T} = \frac{1}{T_Z}. \tag{30}$$

Therefore, the time to contact, T_Z, can be calculated using:

$$T_Z = \frac{\bar{x}}{\Delta x}|T_2 - T_1|. \tag{31}$$

The above result leads to the EMD "product" response when the time to contact is calculated with T_g as in the following:

$$T_Z < \frac{\bar{x}}{\Delta x}T_g. \tag{32}$$

Similarly, the "difference" output would fire when the time to contact is given by:

$$T_Z < \frac{\bar{x}}{\Delta x}T_h. \tag{33}$$

Recall that the two channels ("product" and "difference") have two different time windows of response where $T_g \ll T_h$. As a result, the EMD difference output will trigger for objects that are relatively farther away (greater time to contact) when compared to objects causing the EMD product output to trigger. Therefore, using an analogy of traffic lights, the "difference" output can be thought of as a yellow light and the "product" output as a red light.

C. 1D IMPLEMENTATION

If a one-dimensional case is considered, the single EMD system consists of an array of photoreceptors as shown in Figure 13. In this figure, the *G* filter is duplicated with a left and right variant on each photoreceptor for clarity. This system is equivalent with that shown in Figure 7 repeated along one axis. One addition to the single EMD model from Figure 7 is that a double threshold hysteresis is applied between the derivative block and the lowpass filters as suggested by Franceschini [10]. The function purpose of the double threshold hysteresis will be discussed in further detail in the following paragraphs.

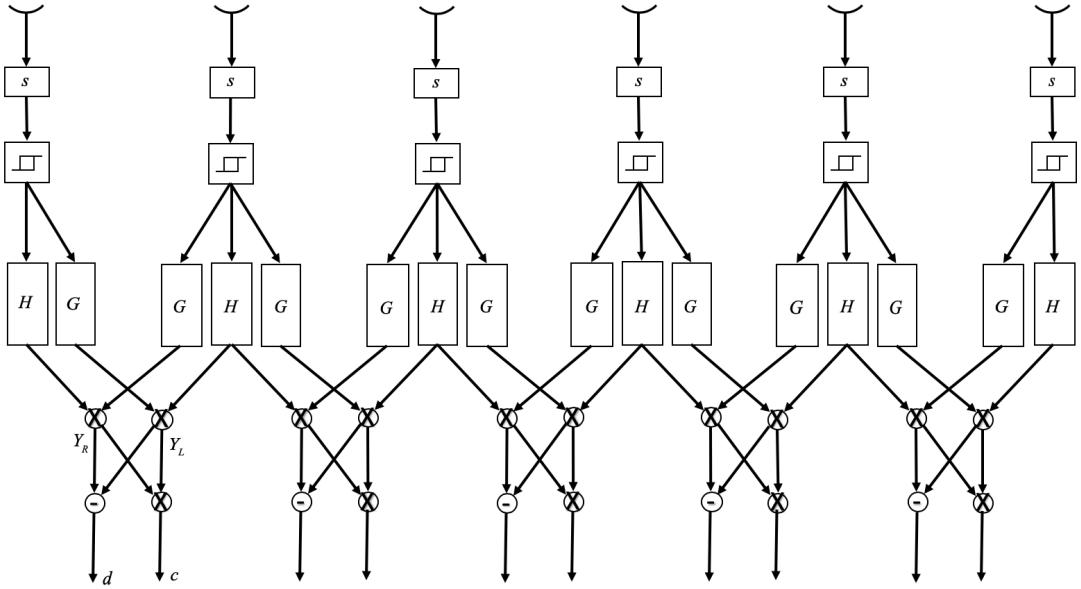

Figure 13. One-dimensional EMD array.

In this one-dimensional case, the photoreceptor inputs are pixels from one row of an image. Since we are processing in the time domain by digital filters, a number of frames, *n*, must be stored in memory to be processed recursively. To compute the derivative function shown by the "s" block in Figure 13, the difference between two frames is taken. This operation is accomplished by convolving image intensity values with a $[1,-1]$ kernel along the time axis. As discussed in previous sections, this filtering acts as a temporal edge detector that responds to the motion of contrasting edges from one frame to the next.

An important issue in this approach is sensitivity to noise. Any small amount of noise, or local change in illumination, is going to trigger the neurons and produce false results. In order to cope with this problem, Franceschini [7] has proposed the addition of a double threshold hysteresis, which not only it is triggered by values over a given threshold but, more important, once it is triggered, it stops smaller (noisy) values. A diagram of double threshold hysteresis is shown in Figure 14.

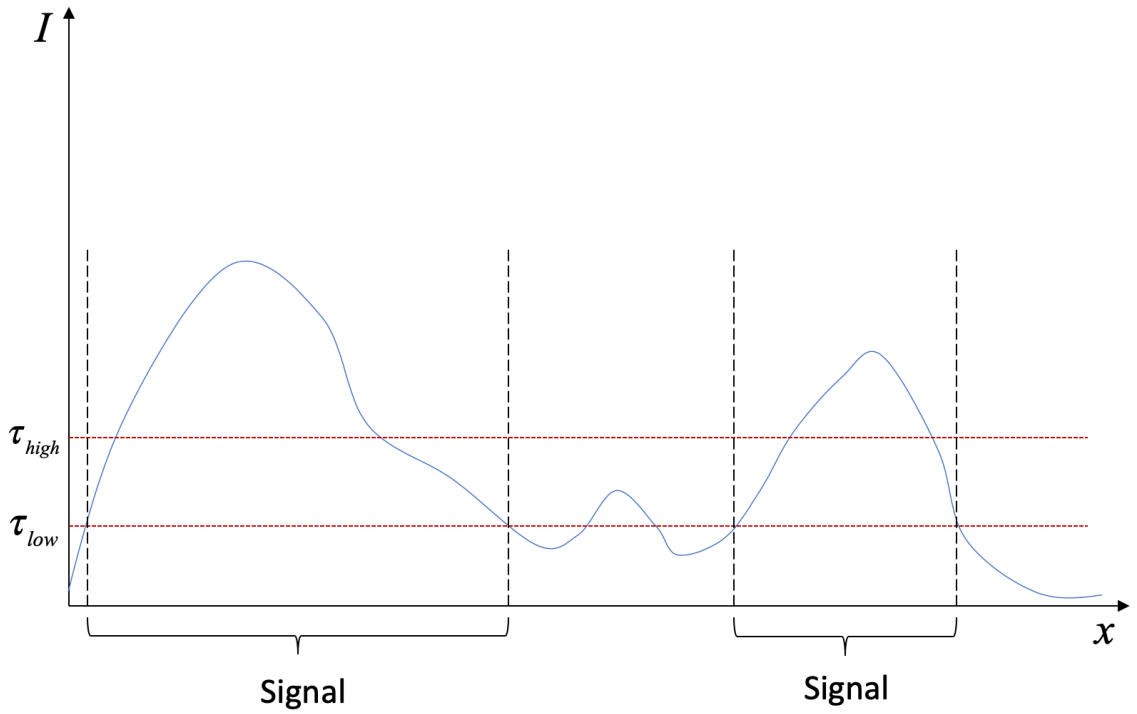

Figure 14. Double threshold hysteresis.

Double threshold hysteresis is applied to the output of the derivative function to reduce false edges [10]. The double thresholding works by applying a low and high threshold, τ_{low} and τ_{high}, to the edges detected by the derivative function. Pixel values below the low threshold value or above the high threshold value are set to 0 or 1 logic values respectively. Pixel values between the two thresholds are thought of as a weak responders that may or may not be part of the signal, in this case a temporal edge. Hysteresis is then applied to these pixels by comparing them to their immediate neighbors.

The hysteresis function assigns a 1 value if a neighbor is above the high threshold and a 0 if not. As a result, weakly responding pixels that are not neighbors with an edge are assumed to be noise, while weak pixels that are neighbors with an edge are assumed to be part of that edge. For the two-dimensional array of pixels over *n* frames, neighboring pixels are defined as the eight pixels surrounding the pixel in question.

After thresholding, the one-dimensional data is convolved in the time domain with filters *H* and *G*. Both of these filters are very simple low pass filters and they are characterized more by their time duration than their frequency response. Linear responses are used for these filters with *G* given a delay and longer response while *H* has a relatively short response with no delay. The filters are constrained to have the same area under the curve to preserve overall motion energy between the two branches of the EMD. Also, they can assume integer values, which further simplifies the implementation. An example of the filter responses is shown in Figure 15.

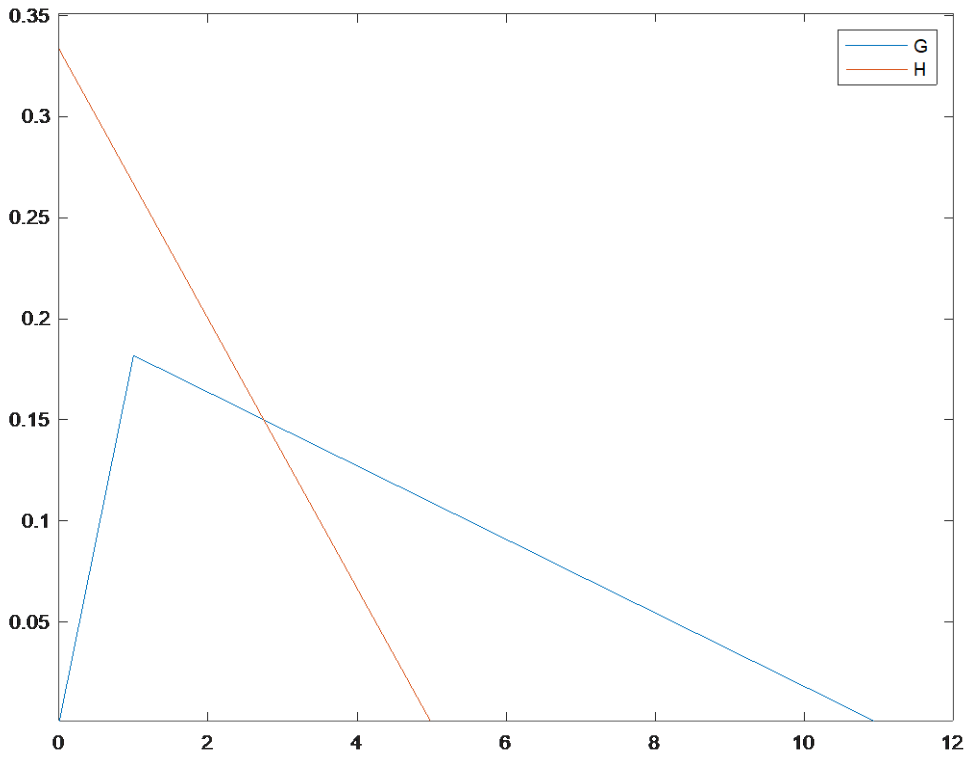

Figure 15. Example filter impulse responses.

The filter outputs are combined as shown in Figure 13 to produce Y_L and Y_R for each photoreceptor (pixel position). In order to simplify the computation of these products, the output of the *G* filter (an $n \times x$ array) is shifted by one column to the right (positive *x*) and multiplied element-wise with the *H* output to produce the Y_R product. Similarly, a left shifted *G* variant is used to produce the Y_L product. These *Y* outputs form the directionally selective responses similar to the T4 and T5 cells of the biological model. A stimulus moving from left to right triggers a Y_L response and a right to left stimulus triggers Y_R.

Finally, as shown in Figure 13, the product (*c*) and difference (*d*) outputs are produced from the results of the *Y* products. For the difference response, Y_R is subtracted from Y_L, producing a positive *d* response if there is motion in from left to right and a negative response for motion from right to left. The *c* response is formed from an element-wise multiplication of the two *Y* arrays.

D. 2D IMPLEMENTATION

The one-dimensional case can easily be extended to two-dimensions. To process two-dimensional image date, additional vertically aligned EMDs are required that have the same arrangement as the one-dimensional EMD from Figure 13.

The derivative operation is again performed over the time domain for each pixel in the image frame. Following the derivative operation, a double threshold is applied to the temporal edges as in the one-dimensional case. Hysteresis is then applied in three-dimensions using the 26 neighboring pixels of a weak response.

Filter responses as shown in Figure 15 are also used in this implementation. However, since there is a three-dimensional output, the *G* filter output is shifted in the up and down (positive and negative *y*) directions in addition to the left and right directions as described above.

As there are now four filter outputs, there are four *Y* products (with *Y* representing the products of the *H* and *G* filters as shown in Figure 13). In the two-dimensional EMD,

we will label the two products from the horizontally aligned EMDs Y_L and Y_R, and the vertically aligned EMDs products Y_U and Y_D for the up and down directions.

The horizontal and vertically aligned EMDs each produce a *d* and *c* output. As discussed above, the pooling of left, right, up, and down responses by the LPTC of the fly visual system is still a topic of research. We use the motion vector length equation to combine the horizontal (d_h) and vertical (d_v) EMD difference outputs to obtain the overall difference signal [4]:

$$d = \sqrt{d_h^{\,2} + d_v^{\,2}}. \tag{34}$$

As there is no equivalent to the product response in the literature, there is no biological precedent for the combination of the horizontal and vertical EMD product responses. We tested sum, product, and vector length methods as defined by the following equations:

$$c = c_h + c_v \tag{35}$$

$$c = c_h c_v \tag{36}$$

$$c = \sqrt{c_h^{\,2} + c_v^{\,2}}. \tag{37}$$

The key difference between the EMD method proposed here and previous work on EMD based depth perception is that this model uses pixels from an image taken by a standard camera as photoreceptor inputs with minimal processing. We will therefore call the two-dimensional EMD described in this section the pixelEMD. In the next chapter we will test the pixelEMD model with videos recorded by cameras in various scenarios ranging from ideal to natural scenery.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. TESTING AND RESULTS

In order to test the camera based pixelEMD, videos were recorded in a variety of environments. Section A describes results obtained using an image sequence designed to be a nearly ideal candidate for the pixelEMD in order to test the effects of various model parameters. Section B through D present results obtained by applying the pixelEMD to more natural scenery images. Natural scenes selected in the study represent realistic locations for MAV operation, specifically forested, urban, and indoor locations. Test videos and resulting pixelEMD output videos are available at https://github.com/dfunni/pixelEMD.

A. IDEAL SCENE

The first video was produced from a sequence of 42 images at a resolution of 480 by 720 pixels. The image sequence was taken by a camera while translating forward on a track at a rate of 1 cm per frame. The scene consists of small boxes covered in high contrast patterns placed at distances of 20, 30, 40, and 50 cm from the front of the track. The video produced represents a nearly ideal scenario for the pixelEMD due to the high contrast patterns on the foreground objects and the elimination of unwanted ego-motion by the track system. Figure 16 shows a selection of frames from the video.

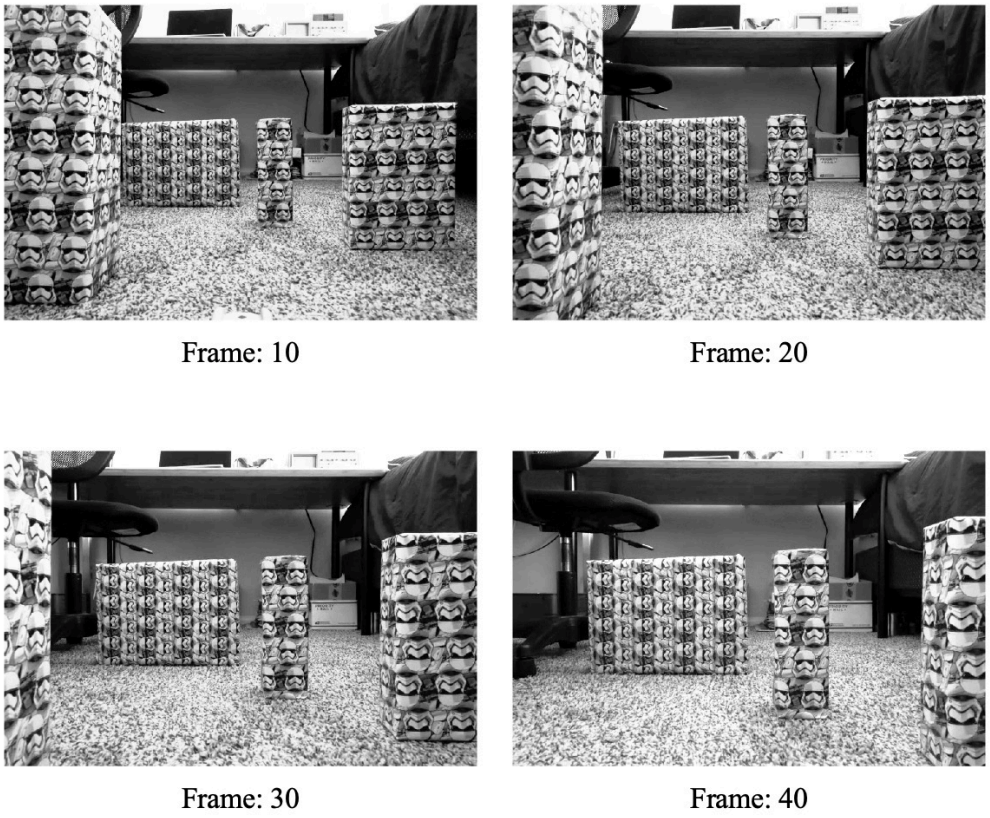

Figure 16. Sample frames from seq12.mov.

1. EMD Pooling Tests

To better understand the types of output possible from the pixelEMD, we first compared the three proposed pooling methods described in equations (35), (36), and (37). The results of the three methods are shown in Figure 17. The left most column, labeled "Intensity," shows the grayscale image of each frame tested for comparison with the objects detected by the EMD output. The remaining columns show results obtained for the three pooling methods investigated. All figures are displayed in the "jet" colormap, where "hotter" colors (orange and red) represent large values (i.e., relatively close objects), and low values are represented with "cooler" colors (greens and blues). Results show that the EMD produces very fine-grained results with excessive noise when the input video resolution is high. Thus, these fine-grained results were spatially averaged with a 5x5 pixel median blur filter on the final output to more easily visually observe trends. It is important

to note, in all three *c* calculation methods the effect of the FOE can be observed. As a result of the camera orientation facing directly toward its own motion vector, the FOE is located in the center of the frame. The effect of the FOE does skew the flow magnitudes of objects near the center of the frame, effectively creating a "blind spot" in the direction of motion.

The second column of Figure 17 shows the "product" response using the pooling method as defined in equation (35). This method does well at displaying the foreground objects, however response is strong with an abundance of "hot" pixels on all of the detected boxes making it difficult to determine the relative depth.

The third column of Figure 17 displays the method described by equation (36). Results show this scheme responds to the two closest objects well. However, this method has a weaker response with objects further from the camera being barely detected.

The final method, calculating the magnitude of the horizontal and vertical response vectors using the vector length method defined in equation (37) is shown in the right most column of Figure 17. This method detects all foreground objects while producing a wide range of values corresponding to the distance from the camera. From inspecting the outputs, the closest object on the left side of the image (the box placed 20 cm in front of the track end) produces the most points in the "red" value range. The rightmost box (30 cm from the track end) produces a slightly "cooler" range of values. The middle boxes (50 cm from track end on the middle left, 40 cm on the middle right) produce even lower responses, while the background is deep blue representing the area of the image furthest from the camera. Finally, a "blind spot" is observed that partially obscures the response of the tall skinny box in the middle of the image (40 cm from the front of track), due to the effect of the FOE as described above.

From these tests, it is concluded that the most useful method for pooling the vertically and horizontally aligned pixelEMD "product" responses is the method described by equation (37). As a result, this method will be used in all future tests presented in this work.

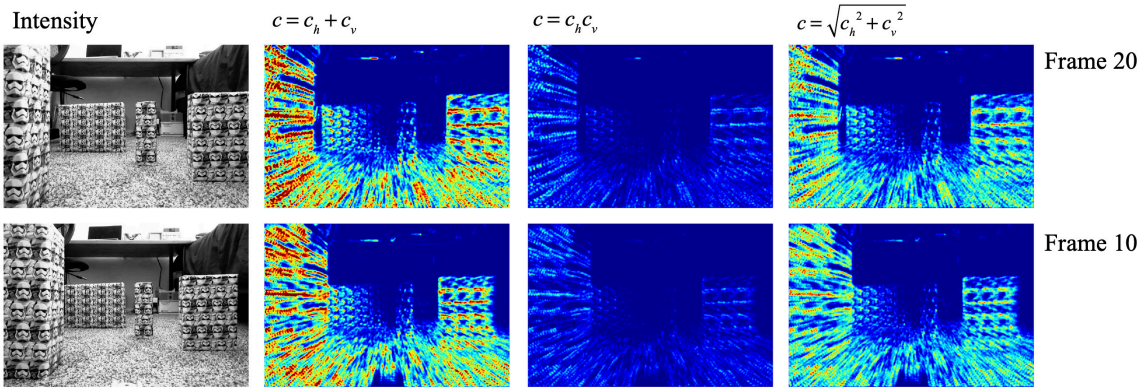

Figure 17. "Product" output pooling comparison using seq12.mov.

2. Filter Shape Tests

To obtain a better understanding of the effects caused by changing the shape of filter impulse response of the *H* and *G* lowpass filters, tests of various filter length combinations and filter delays were conducted. Many filter length combinations were considered during the investigation and results presented are used to show general trends. Plots obtained for a few of the filter impulse responses considered in this study are displayed in Figure 18. The filters are generated so that the sum of the filter coefficients of the two filters are equal. The left column shows responses of various filter lengths with a fixed delay applied to *G*. The right column displays the responses of various delays for *G* with fixed filter lengths of 5 and 10 frames for *H* and *G* respectively.

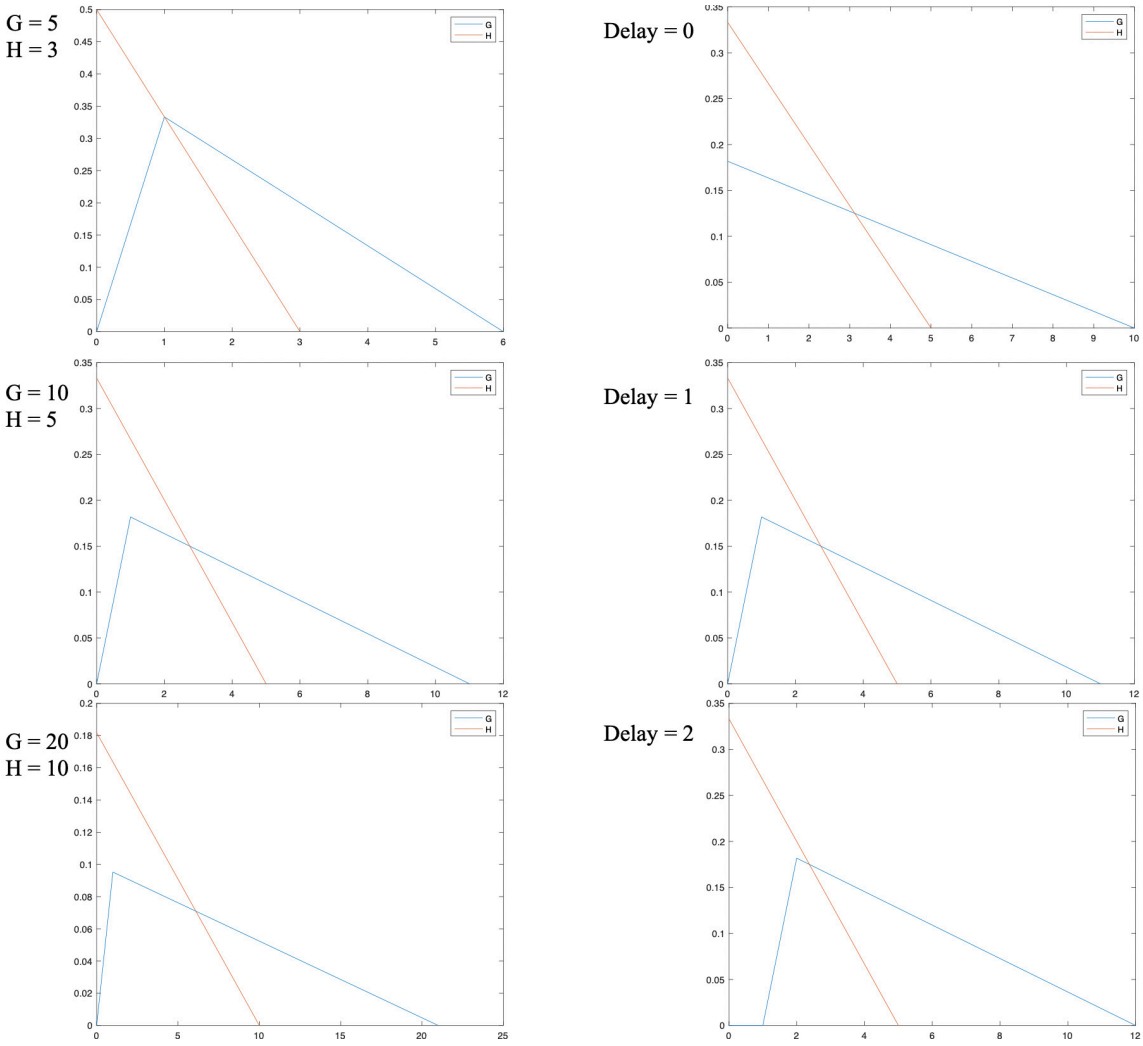

Figure 18. Impulse responses of tested filters.

The pixelEMD responses displayed in Figure 19 are given for each of the filters considered on the left column of Figure 18. The intensity (grayscale) image of the frame tested, labeled *I*, is displayed for comparison with the EMD output. Both the "product" and "difference" outputs are shown in rows labeled *c* and *d* respectively. Results show that increased temporal blurring occurs as the filter length increases while short filters lead to a sharper response. This effect is particularly noticeable in the "product" response.

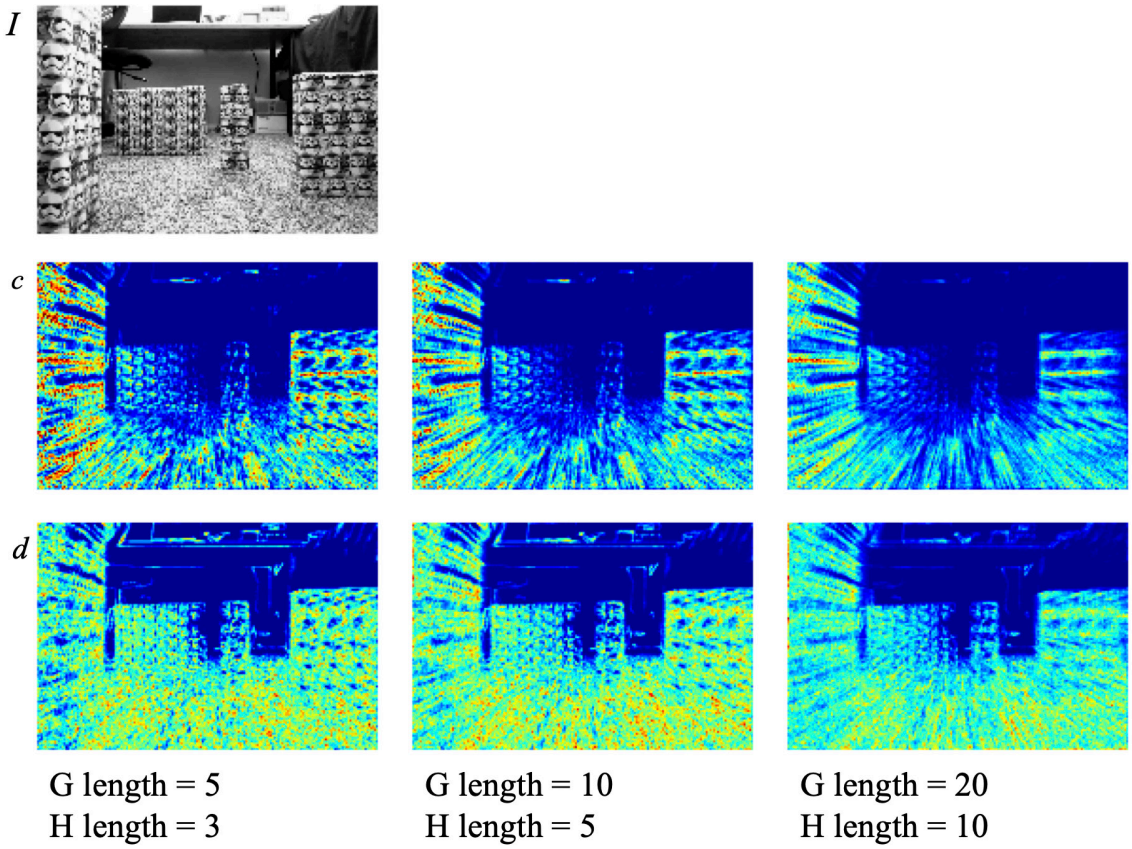

Figure 19. EMD filter length testing on frame 20 of seq12.mov.

The effect of delaying the response of *G* is shown in Figure 20. This testing shows very small differences are observed by changing delay sizes. Again, a slight increase in temporal blurring in the "product" output is observed as the delay increases. This behavior can be attributed to the overall increased length of the *G* filter due to the addition of the delay. More notably, the "difference" output shows progressively lower values as the delay parameter is increased. Since the value of the delay represents the temporal distance (in number of frames) between the output of the *G* and *H* filters, the decreasing output magnitudes noted for larger delays likely results from reduced correlation between frames separated by more time.

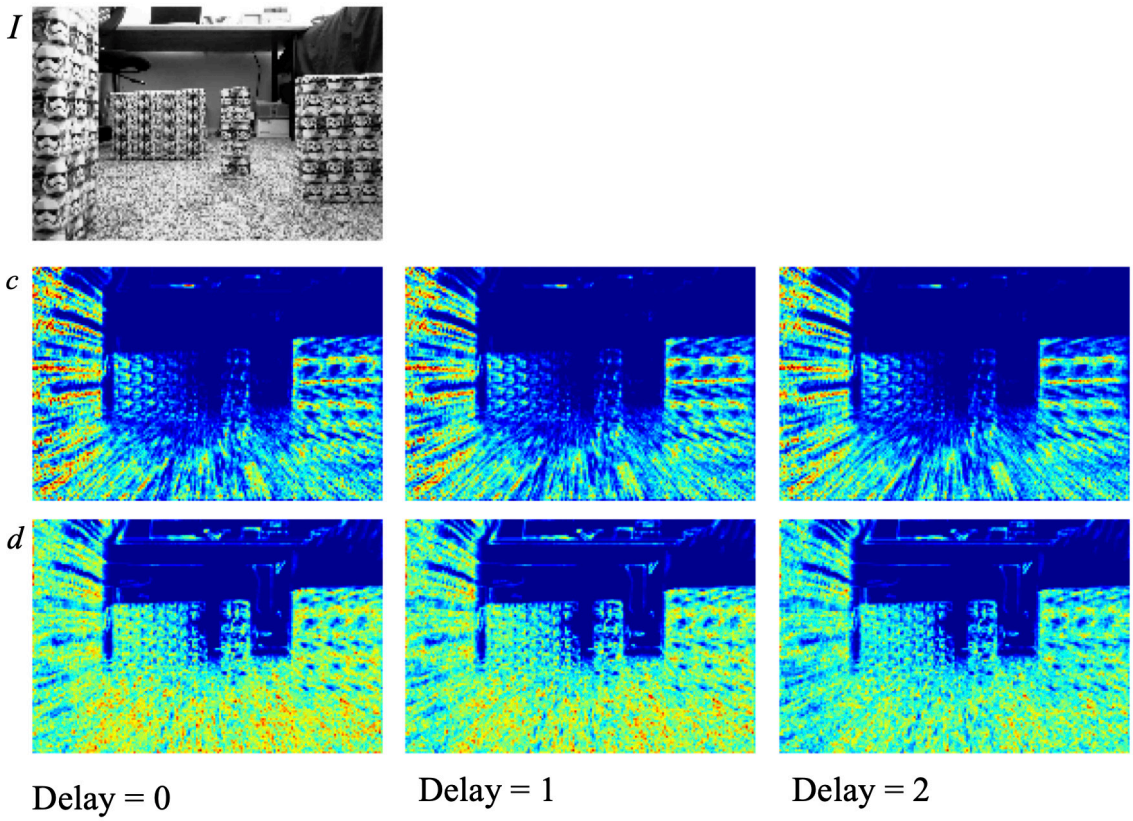

Figure 20. Filter *G* delay testing on frame 20 of seq12.mov.

Although there is little appreciable difference in EMD output with varying filter shapes, we still can reach some conclusions from these tests. First, very long filter lengths should be avoided to reduce the associated temporal blurring. Secondly, delaying the *G* filter by large values should likewise be avoided. Additionally, when considering implementing the EMD in a system with computational constrains, both the memory and computational requirements for buffering and processing the added frames due to longer filters become undesirable. In the remainder of this thesis we will use filters of length 5 and 10 for *H* and *G* filters respectively, and a *G* response delayed by 1.

3. Double Threshold Tests

Lastly, we investigated various values for the double threshold. Note that, it is beneficial to view the histogram of values prior to thresholding before choosing specific threshold values. The histogram of the derivative function output is shown in Figure 21 as

an example. The histogram displays pixel values in the range of [0,1] on the x-axis and the number of pixels with each value on the y-axis.

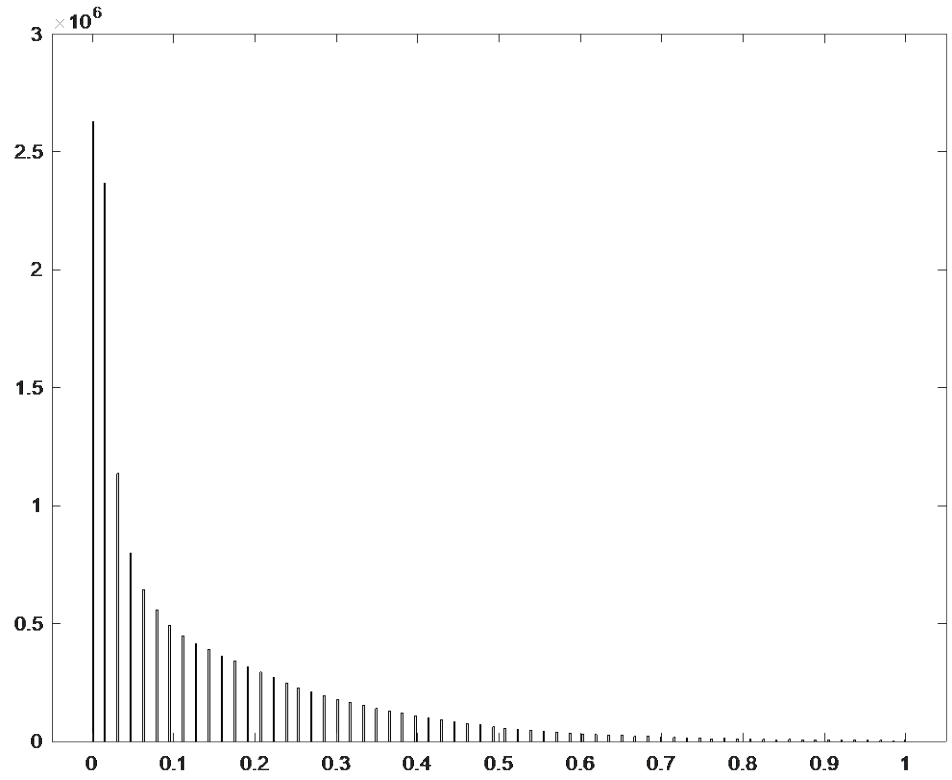

Figure 21. Histogram of temporally differentiated values of frame 20 seq12.mov.

For the double threshold operation, pixels with values in the range between the lower and upper threshold are considered "weak" responders and are processed further to determine if they are noise or part of a temporal edge. From the histogram in Figure 21, it is clear that a majority of the pixels have low values in the range of 0 to 0.2 and relatively few have values above 0.5. Since the histogram shows that the number of pixels with higher intensity falls off dramatically, upper threshold placement has a small effect. We will therefore focus our testing on various low threshold value placements. Figure 22 shows the result of threshold tests with the lower threshold ranging from 0.03 to 0.12 and a fixed

upper threshold of 0.5. From the figure, it can be seen that the “product” response shows dramatic differences where the “difference” response changes subtly.

Considering the “product” responses illustrated in Figure 22 the pixelEMD response is highly dependent on the lower threshold value. On the low end of tested values (lower threshold of 0.03 and 0.05), the response shows a significant amount of high valued outputs for the foreground objects and even responds to some background objects. As the threshold is increased further, the EMD response has more variation for the foreground objects while background objects are largely not detected.

Observing the “difference” response, lower threshold values again tend to more clearly show objects in the background of the scene while higher thresholds mask these objects. An important phenomenon displayed in these tests is that at lower thresholds, the effect of the FOE is reduced. In fact, in the case of the 0.03 the FOE effect is reversed, with the highest EMD responses at the center of the scene where foreground objects are actually further from the camera than those on the periphery.

From the observations made based on Figure 22, it can be concluded that decreasing the lower threshold has the effect of increasing the detection distance of the EMD at the expense of depth resolution for nearby objects. With very low values there are some erroneous responses on the “difference” EMD output. Alternately, increasing the lower threshold has the effect of shortening the detection distance while increasing the depth resolution available for nearby objects. We therefore will use low threshold values for scenes where obstacles are anticipated to be in the distance, and higher threshold values if resolution of nearby objects is desired.

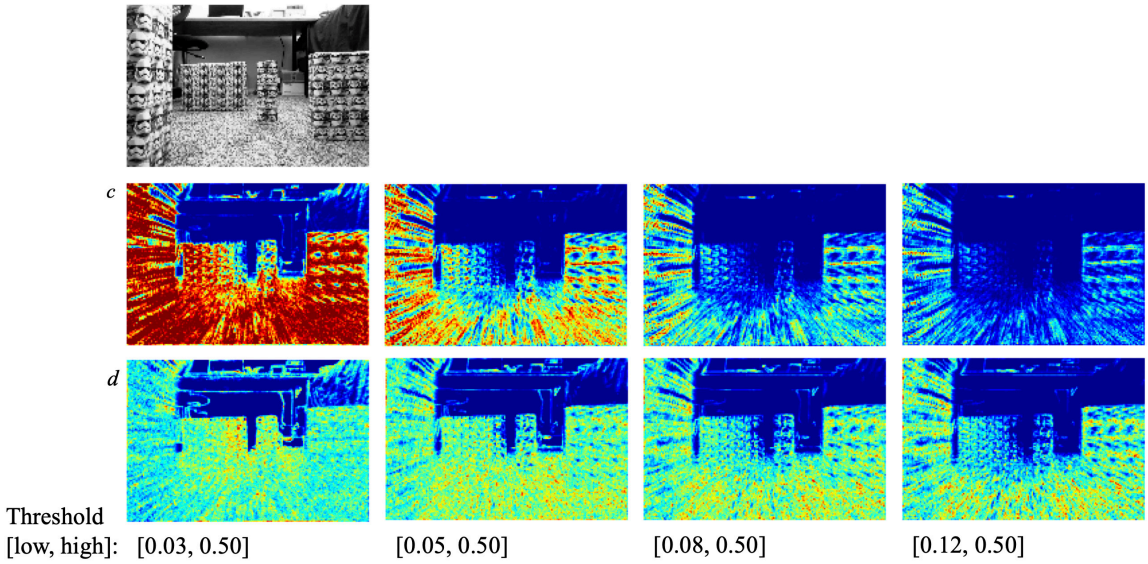

Figure 22. Double threshold testing on frame 20 seq12.mov.

In summary, from the tests conducted on an ideal scene, some significant conclusions were reached that will aid in the discussion of EMD implementation on more natural scenes. 1) The vector length calculation, equation (37), was determined to be the best performing pooling method for the "product" output. 2) Tests of multiple impulse responses for the two lowpass filters resulted in choosing a conservative length that produces satisfactory results while bearing in mind the associated computational and memory costs for implementation in a SWaP constrained system. 3) Varying the lower threshold affected both detection distance and depth resolution, where low values detect further objects with low depth resolution while higher values have greater depth resolution for nearby objects. These conclusions will be applied to the remainder of the scenes presented in this thesis.

B. FOREST SCENE

In order to determine EMD functionality in a realistic scenario a video was recorded by hand carrying a camera while walking forward in a moderately wooded outdoor environment. With a framerate of 30 frames per second, 99 frames at 270 by 480-pixel resolution were converted to grayscale (intensity) for input into the EMD. Sample frames from the video are shown in Figure 23. This video represents what could be expected from

an autonomous MAV operating in a forested area. Since the camera was handheld, motion was not restricted to only forward translation. This unwanted motion results in added noise in this test. In a real system this ego-motion can be expected both from control signals and interaction with wind. Additionally, shadows and patches of sunlight in the image frame create visual contrast in the absence of a physical object.

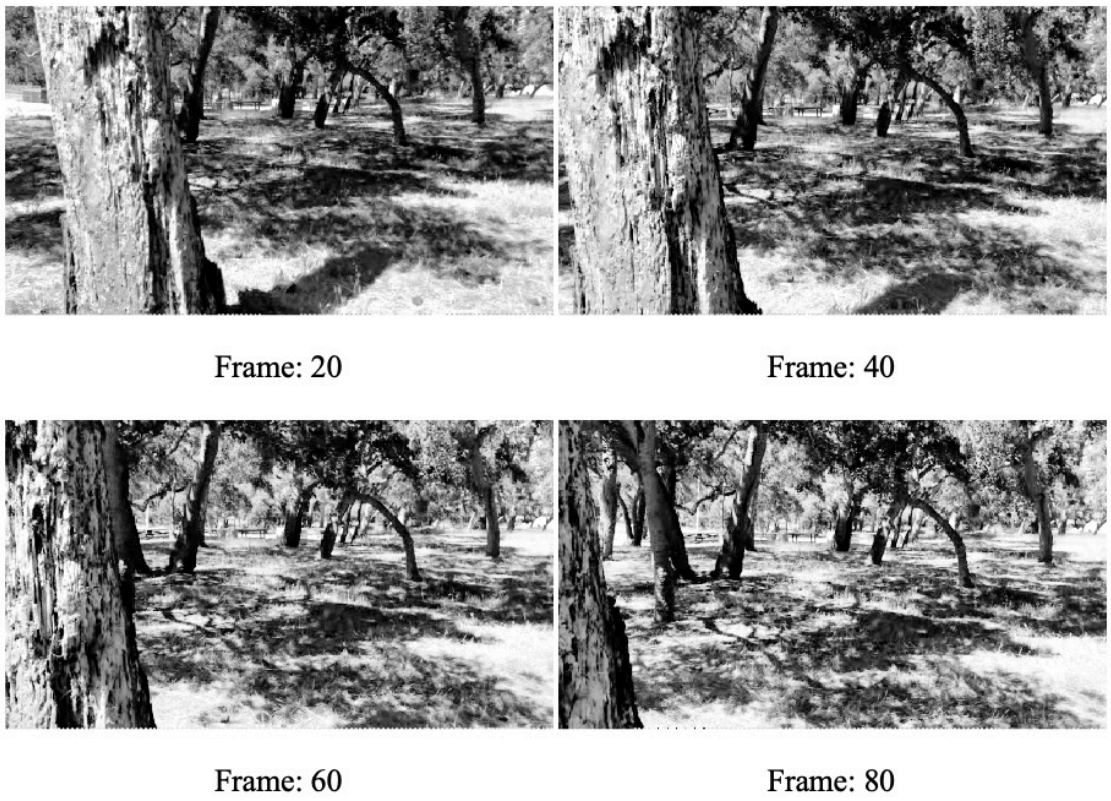

Figure 23. Sample frames from park.mov.

Studying the scene shown in Figure 23, one tree is seen to pass by the left side of the camera at a relatively close distance while multiple trees in the background are also observed. Due to the wide variance in depths present in the scene, we started with the lower threshold of the double threshold set to 0.05 to attempt to observe both foreground and background objects.

The output of the EMD is shown in Figure 24. From the "product" responses in the figure (column labeled *c*), the nearby tree is clearly detected in all frames and presents a larger response as it closes distance with the camera. Objects in the distance produce noisy depth results as seen in the relatively high responses present in frames 40 and 60. This behavior is the result of unwanted camera ego-motion around those frames and highlights the sensitivity to rotational motion inherent in the EMD system (also present in other optical flow methods). The EMD "difference" output (column *d*) has less useful information with mid-level responses propagating across the entire frame.

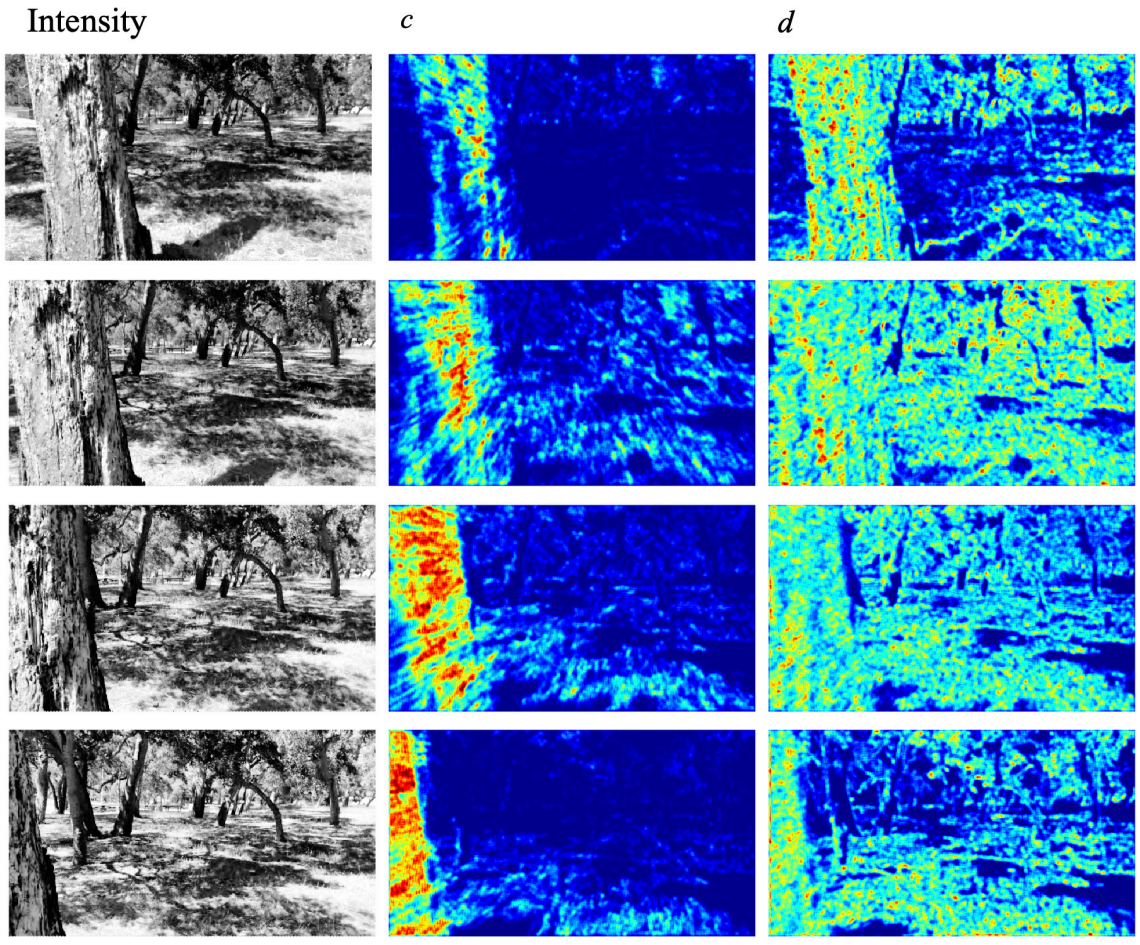

Figure 24. Forest scene EMD output with low threshold set to 0.05 (rows display frames 20, 40, 60, 80 from top to bottom).

In Figure 24 the overall "gain" in the output is quite high. Next, an attempt was made to reduce noise from unwanted camera ego-motion by increasing value of the lower threshold parameter from 0.05 to 0.10. The resulting output is shown in Figure 25. This figure shows an improved response on the "difference" output, which is now able to detect the nearby tree while suppressing background noise, especially in frames 20 and 80 where unwanted camera movement is minimal. The resulting "product" responses were, however, much more subdued as can be expected from increasing the lower threshold. Although the "product" is not able to detect objects in the background, the tree is still detected clearly.

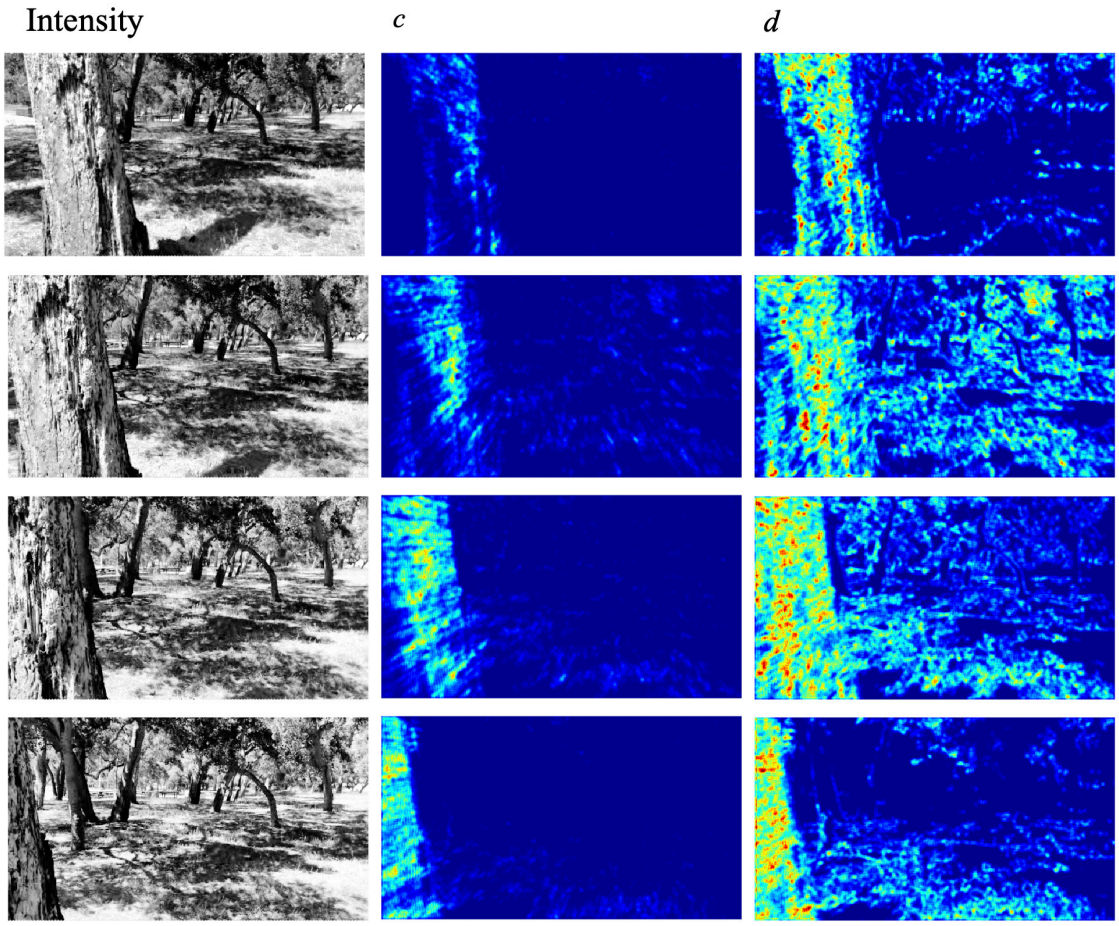

Figure 25. Forest scene EMD output with low threshold set to 0.10 (rows display frames 20, 40, 60, 80 from top to bottom).

Overall, the EMD functions well for the natural outdoor scene. Results show objects near the camera are easily detected by the "product" output. Given the proper lower threshold setting and minimal unwanted camera motion, the "difference" also is able to detect both foreground and background objects. The "traffic light" effect discussed in Chapter III is shown by the EMD responses in Figure 25. The "difference" response, or "yellow light," conservatively detects objects at a great distance, while the "product" response acts as a "red light" detecting only the nearby tree.

C. INDOOR SCENE

The second realistic scenario considered for MAV operation is the indoor environment. To test the EMD with an indoor scene, a video was again recorded by hand carrying a camera while walking forward through a doorway into a brightly lit student lounge. Again, the camera recorded at a framerate of 30 frames per second. The full video consists of 209 frames with 270 by 480-pixel resolution. Sample frames from the video are shown in Figure 26. Again, noise is present in this test due to unwanted ego-motion of the handheld camera. In this scene, man-made objects such as tables and painted walls produce large areas of low contrast in the recorded image, which limits the effectiveness of the EMD due to reliance on temporal edge detection.

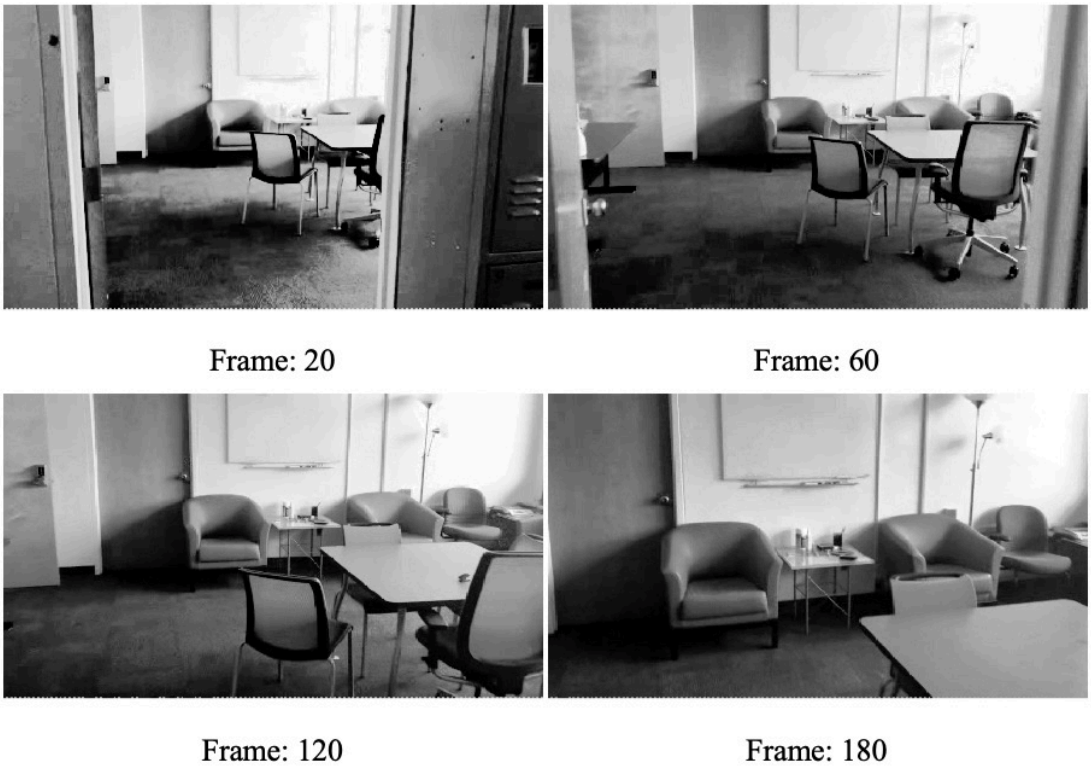

Figure 26. Sample frames from lounge.mov.

The EMD output is shown in Figure 27 for the indoor scene. As in the outdoor example, we chose a lower threshold of 0.05 to start. From observing the frames in the figure, two trends emerge. The first trend is that objects that get very near the camera, and therefore have a large optical flow velocity, do tend to trigger a strong response from the EMD as can be seen by the doorframe in frame 60 (the second row). The more significant observation is the limitation of the EMD to detect motion of objects with little spatial contrast (i.e., intensity contrast within a single frame). This characteristic can be observed by the smooth textured walls, tabletop, chair backs, and floor all failing to trigger an EMD response while edges of the same objects trigger strong responses.

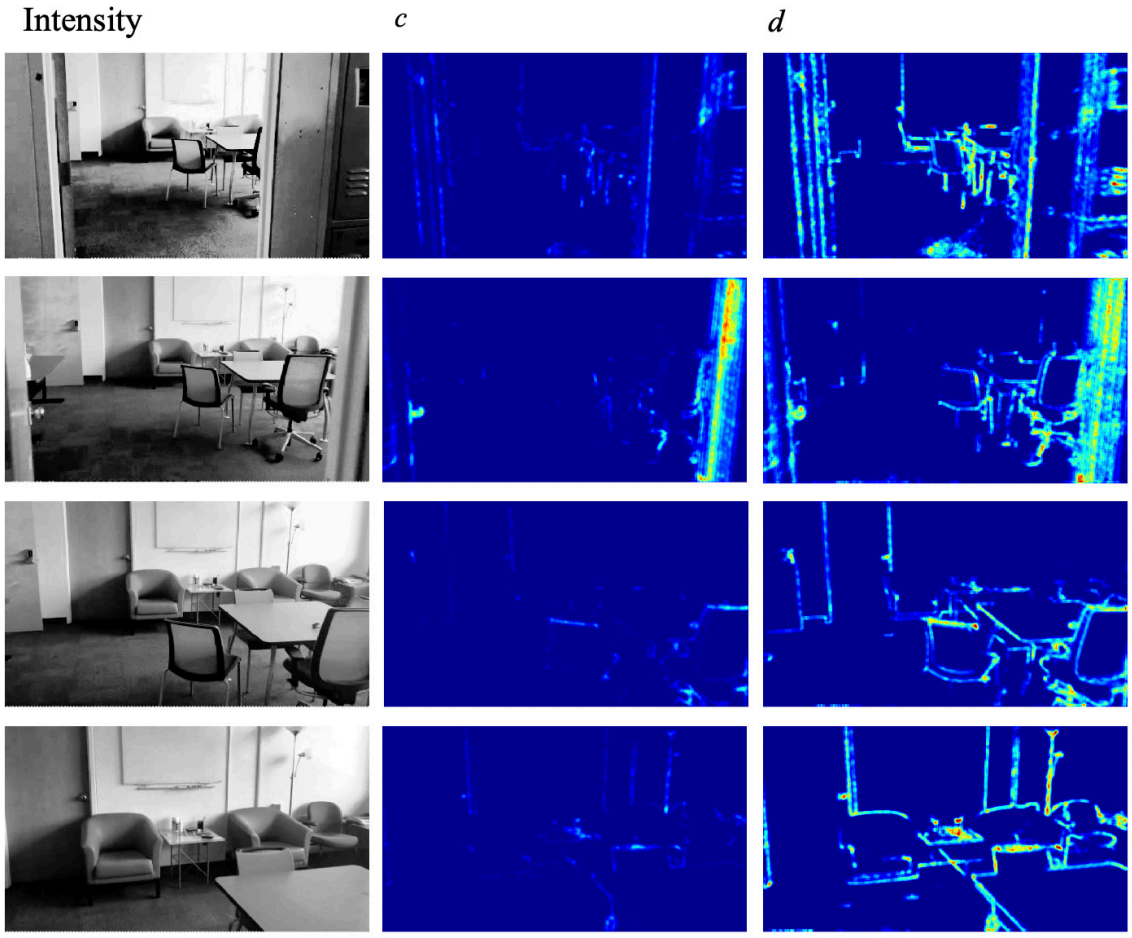

Figure 27. Indoor scene EMD output with low threshold set to 0.05 (rows display frames 20, 60, 120, 180 from top to bottom).

Thus, the lower threshold parameter was reduced to 0.03 in an attempt to increase the sensitivity of the EMD and possibly detect lower contrast features and results illustrated in Figure 28. Resulting output videos frames show edges are displayed more prominently.

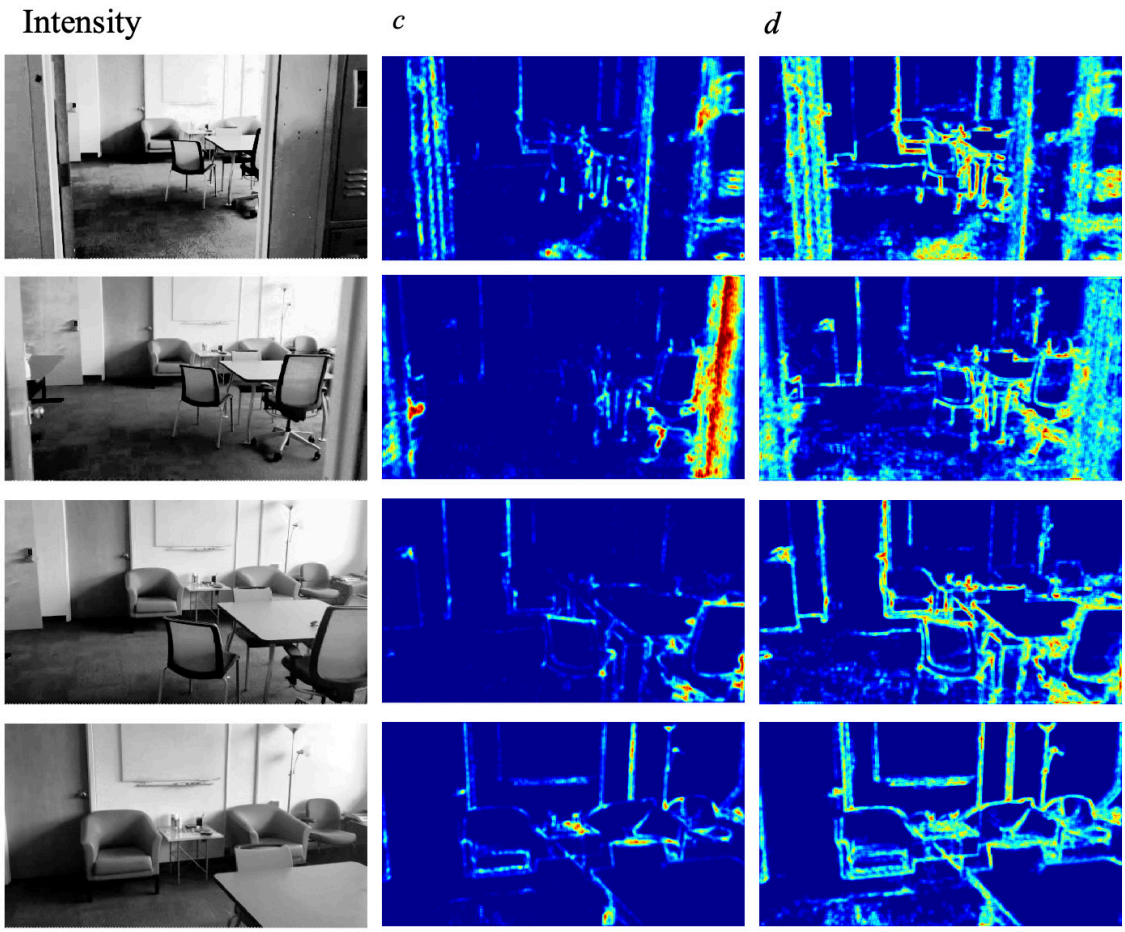

Figure 28. Indoor scene EMD output with low threshold set to 0.03 (rows display frames 20, 60, 120, 180 from top to bottom).

Overall, results show the EMD has a limited capability in modern indoor environments. This behavior is due to the reliance on spatial contrast on objects for detection. Modern environments tend to have objects consisting of large, flat surfaces like walls and tables with only sparsely distributed points that have enough texture to be detected by the temporal edge detector. Additionally, in a confined indoor space the camera is forced into close proximity with objects in the scene, further limiting the spatial distribution of point of interest within the frame. Thus overall, the EMD is not an ideal candidate for MAV object detection during confined, indoor flight.

D. OUTDOOR URBAN SCENE

The final scenario tested was an outdoor, urban environment of a downtown street. The video was taken by placing a camera on the dashboard of a vehicle traveling at 15 miles per hour. The video has a framerate of 30 frames per second, a resolution of 270 by 480 pixels, and is 99 frames in length. Sample frames are shown in Figure 29. Of note, the bottom of each frame shows the dashboard and hood of the test vehicle, which appears stationary throughout the video and therefore creates no EMD response. This urban environment was chosen as it could resemble many urban scenarios in which a MAV could operate. This scene displays aspects of both the forest and indoor scenes, such as the high contrast trees of the forest, and large, low contrast surfaces such as the cars, buildings, road, and sky.

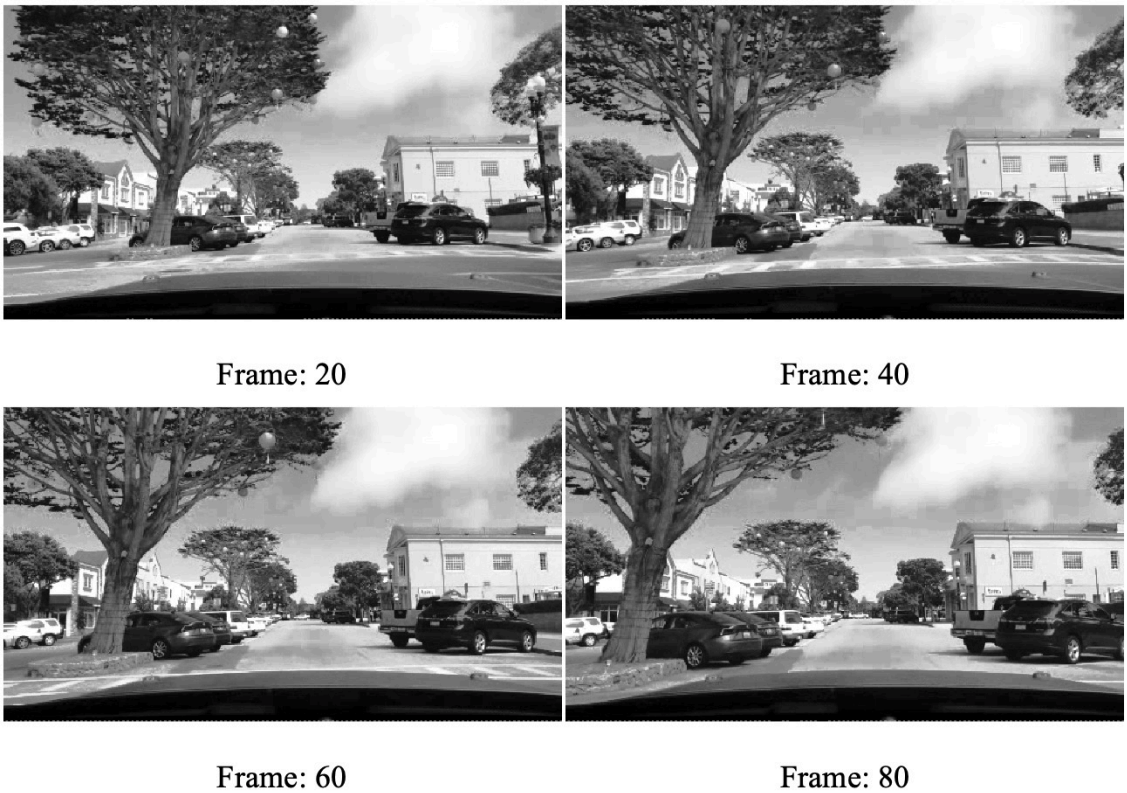

Figure 29. Sample frames from street.mov.

The output of the EMD is shown in Figure 30. Again, we used a lower threshold value equal to 0.05 at the double threshold hysteresis stage. As can be expected, the trees are easily detected by both the "product" and "difference" outputs. However, more important for use in an urban environment is the response to vehicles and buildings. In this case, the large surfaces of low contrast like building walls and vehicle side panels are much further from the camera than in the indoor scene. This added distance ensures points of contrast such as windows, wheels, and license plates are more densely packed within the frame, which allows them to be detected by the EMD. Therefore, these results indicate the EMD can be an effective means of detecting obstacles given a scene with sufficient contrast or a large enough area.

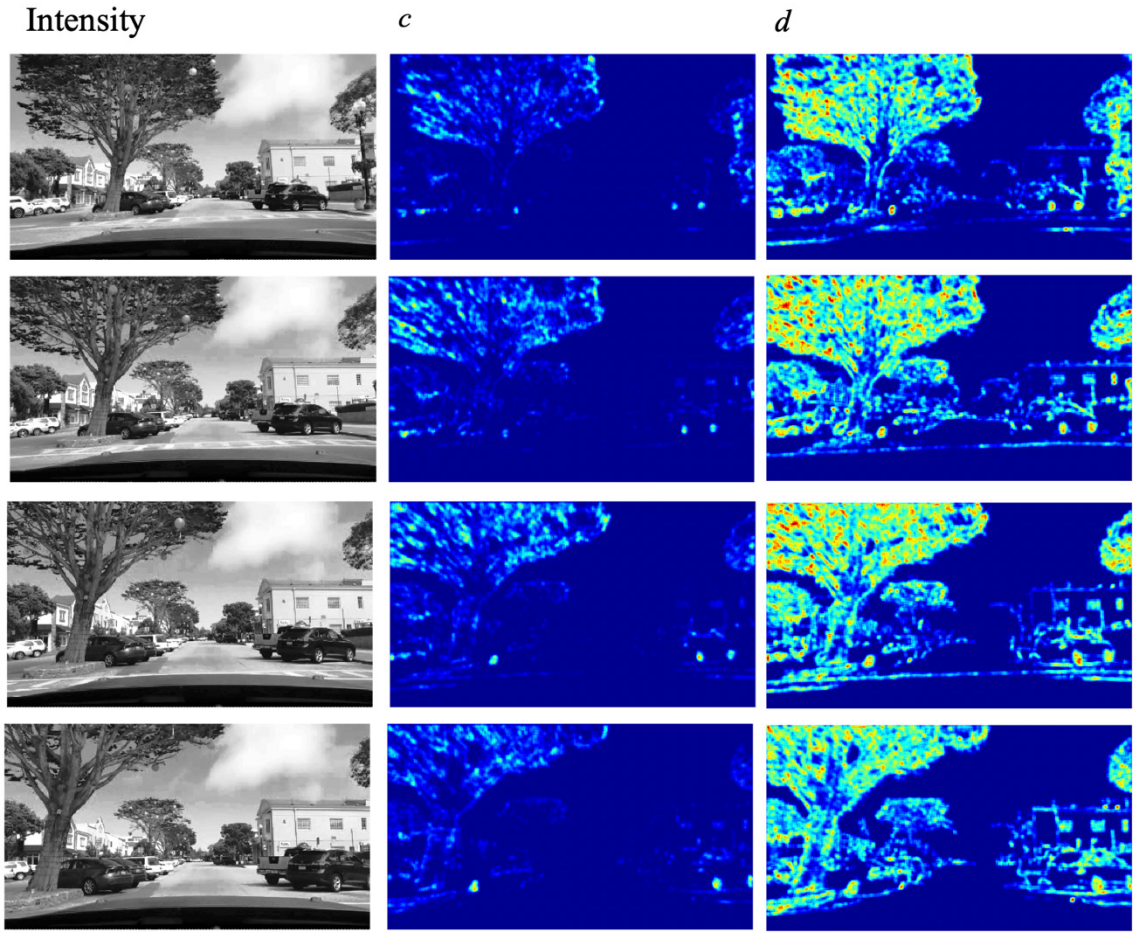

Figure 30. Outdoor urban scene EMD output with low threshold set to 0.05 (rows display frames 20, 40, 60, 80 from top to bottom).

In this chapter we have displayed and discussed the results of the camera based EMD on both ideal and realistic scenes. In the next chapter we will consolidate our recommendations and propose the next steps forward for this research.

V. CONCLUSIONS AND FUTURE WORK

In this section we will consolidate and assess the results presented in Chapter IV and discuss recommendations for appropriate use-cases of the proposed camera-based EMD. We will then discuss the future work required to further develop and evaluate the effectiveness of this biologically-inspired obstacle detection method.

A. CONCLUSIONS

The pixelEMD proposed in this thesis does meet our objectives by providing a computationally inexpensive means of monocular obstacle detection. While it did show some limitations, it proves effective in certain scenarios. In particular, the pixelEMD performs well in natural outdoor settings where obstacles have high contrast like grasses, tree bark, branches and leaves. Additionally, this method is an ideal candidate for efficient implementation on an FPGA or ASIC due to the easily parallelizable array operations used.

Two significant limitations were found for the pixelEMD model presented in this thesis. The first limitation is that any camera motion other than smooth forward translation will cause large erroneous spikes in the motion detected. In real MAV systems, unwanted motion of the drone body often occurs due to variations in wind or turbulence created by propeller wash. The second limitation is the need for spatially dense contrasting features on approaching obstacles. This limitation was observed during indoor tests where large, flat, monochromatic obstacles such as table tops and walls were not detected by the pixelEMD. Therefore, it is recommended that this obstacle detection method not be used in cluttered indoor settings where large, flat, monochromatic obstacles need to be detected at close proximity.

B. FUTURE WORK

Results showed the pixelEMD to be sensitive to unwanted ego-motion resulting in increased visual noise. It is recommended to stabilize the video source prior to processing further to reduce or eliminate much of the unwanted ego-motion due to environmental

effects like wind. These effects could also be reduced through calculating the ego-motion by visual odometry or an inertial measurement unit (IMU).

Another direction for future work is a hardware implementation of the EMD model via FPGAs. Alternately, highly parallel execution could be tested on an embedded vision accelerator like the Myriad 2 vision processing unit from Intel or combination embedded graphics processing unit platform such as NVIDIA Jetson line. A working implementation using any of these methods could then be tested on a MAV flying in a variety of environments and the obstacle detection system integrated with a control system for autonomous obstacle avoidance.

Lastly, to improve the overall usefulness of the model, additional research could investigate detecting distance to points in the environment from the EMD model. In such a scenario, an IMU would be needed to determine air speed, which could then be combined with framerate and time-to-contact information. Assuming accurate results, three-dimensional environment mapping could be considered.

APPENDIX. CODE

```
% pixelEMD.m
%
% This script tests the pixelEMD with the parameters listed under the
% "Constants" heading. The following user-defined functions are used:
%
%   hysteresis3d.m
%   ommatidia_grid3d.m (optional)
%
% Created by David Funni

clear;
close all;

%% Constants
tl = 0.08;      % hysteresis low threshold
th = 0.5;       % hysteresis high threshold
hh = 5;         % H filter length
gg = 10;        % G filter length
delay = 2;      % delay amount, 1 is no delay
sel = 20;       % frame to be displayed
blur_ksize = 5; % kernel size for EMD output spatial pooling
imsize = 200;

%% Read video file and get parameters
video_file = 'park.mov';

video_object = VideoReader(video_file);
video_array = read(video_object);
[h, w, c, n] = size(video_array); % dimension of original video

% initial processing: convert video array to histogram equalized grayscale
x = zeros(h, w, n);
for i = 1:n
    gray = double(rgb2gray(video_array(:,:,:,i))) / 255.0;
    x(:,:,i) = histeq(gray);
end

% % apply ommatidia grid to video array (optional)
% om_ksize = 5;
% sigma = 2;
% shift = 2 * om_ksize;     % vertical and horizontal distance between photoreceptors
% x = x .* ommatidia_grid3d([h,w], n, om_ksize, sigma);

%% Apply EMD model

% Apply derivative function in the time domain
dx = filter([1,-1], 1, x, [], 3);

% Implement 3D hysteresis
```

```
dx_hat = hysteresis3d(abs(dx), tl, th, 26);

% create the filter responses
G = zeros(1,gg+delay);

% G and H lowpass filters with equal area
H = (hh:-1:0) / sum(1:hh);              % FIR Filter response for H
G(delay:end) = (gg:-1:0) / sum(1:gg);   % FIR Filter for side H

% Implement filter
GX = filter(G, 1, dx_hat, [], 3);
HX = filter(H, 1, dx_hat, [], 3);

% Shift the output of the G filter left, right, up, down
% This allows an array multiply to be conducted in the next
% stage to multiply the filter outputs of neighboring
% photoreceptors (pixels).
GLX = circshift(GX, -shift, 1);
GRX = circshift(GX, shift, 1);
GUX = circshift(GX, -shift, 2);
GDX = circshift(GX, shift, 2);

% Compute the four multiply stages for each pixel
YL = HX .* GLX;
YR = HX .* GRX;
YU = HX .* GUX;
YD = HX .* GDX;

% Compute 1D horizontally aligned EMD output
Ch = YL .* YR;
Dh = YL - YR;

% Compute 1D vertically aligned EMD output
Cv = YU .* YD;
Dv = YU - YD;

% 2D pooling of the 1D outputs
D = sqrt(Dh.^2 + Dv.^2);
C = sqrt(Ch.^2 + Ch.^2);

%% Display results

blurfilt = ones(blur_ksize) / blur_ksize^2;

% Show original video
for i = 1:n
   imshow(x(:,:,i), 'DisplayRange', [], 'InitialMag', imsize)
end

% Show D output video
for i = 1:n
   frame = imfilter(D(:,:,i), blurfilt);
```

```
    imshow(abs(frame), 'colormap', jet, 'InitialMag', imsize)
end

% Show C output video
for i = 1:n
    frame = imfilter(C(:,:,i), blurfilt);
    imshow(abs(frame), 'colormap', jet, 'InitialMag', imsize)
end

% Display selected frame
for i = 1:n
    if i == sel
        % display original image
        figure
        imshow(x(:,:,i), 'InitialMag', imsize, 'DisplayRange', [])

        % display histogram of derivative function
        figure
        histogram(abs(dx))

        % display C output of selected frame
        figure
        C(:,:,i) = imfilter(C(:,:,i), blurfilt);
        imshow(C(:,:,i), 'colormap', jet, 'InitialMag', imsize);

        % display D output of selected frame
        figure
        D(:,:,i) = imfilter(D(:,:,i), blurfilt);
        imshow(D(:,:,i), 'colormap', jet, 'InitialMag', imsize);
    end
end
```

```
function hys=hysteresis3d(array,t1,t2,conn)
% Hysteresis3d is a simple function that performs
% hysteresis for 2D and 3D images. Hysteresis3d was inspired by Peter
% Kovesi's 2D hysteresis function
% (http://www.csse.uwa.edu.au/~pk/research/matlabfns/). This 3D function
% takes advantage of the 3D connectivity's of imfill.m instead of the 2D
% connectivity's of bwselect.
%
% Usage:        hys=hysteresis3d(img,t1,t2,conn)
%
% Arguments:    img - image for hysteresis (assumed to be non-negative)
%               t1 - lower threshold value (fraction b/w 0-1, e.g.: 0.1)
%               t2 - upper threshold value (fraction b/w 0-1, e.g.: 0.9)
%                  (t1/t2 can be entered in any order, larger one will be
%                  set as the upper threshold)
%               conn - number of connectivity's (4 or 8 for 2D)
%                                       (6, 18, or 26 for 3D)
% Returns:
%               hys - the hysteresis image (logical mask image)
%
% Adapted from code by Luke Xie:
% https://www.mathworks.com/matlabcentral/fileexchange/44648-hysteresis-thresholding-for-3d-
images-or-2d

% swap values if t1 > t2
if t1 > t2
   [t2, t1] = deal(t1, t2);
end

% scale thresholds by intensity range
minv = min(array(:));
maxv = max(array(:));

t1v = t1 * (maxv - minv) + minv;
t2v = t2 * (maxv - minv) + minv;

% hysteresis
abovet1 = array > t1v;                          % indices of values above lower threshold
seed_indices = sub2ind(size(abovet1),find(array > t2v));  % indices of values above upper
threshold
hys = imfill(~abovet1,seed_indices,conn);               % obtain all connected regions in abovet1
that include points with values above t2
hys = double(hys & abovet1);
```

LIST OF REFERENCES

[1] "38th Commandant's planning guidance." [Online]. Available: https://www.marines.mil/News/--Publications/MCPEL/Electronic-Library-Display/Article/1907265/38th-commandants-planning-guidance. [Accessed: 10-Aug-2019].

[2] "Fast Lightweight Autonomy (FLA)." [Online]. Available: https://www.darpa.mil/program/fast-lightweight-autonomy. [Accessed: 10-Aug-2019].

[3] "μBRAIN." [Online]. Available: https://www.darpa.mil/program/microbrain. [Accessed: 10-Aug-2019].

[4] A. Schwegmann, J. P. Lindemann, and M. Egelhaaf, "Depth information in natural environments derived from optic flow by insect motion detection system: a model analysis," *Frontiers in computational neuroscience*, vol. 8, p. 83, 2014.

[5] F. Kendoul, I. Fantoni, and K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Robotics and Autonomous Systems*, vol. 57, no. 6–7, pp. 591–602, 2009.

[6] B. Horn and B. G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185–203, Aug. 1981.

[7] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proceedings of the 1981 DARPA Image Understanding Workshop*, pp. 121–130, 1981.

[8] T. A. Camus, "Calculating time-to-collision with real-time optical flow," in *Visual Communications and Image Processing '94*, 1994, vol. 2308, pp. 661–670.

[9] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Int J Comput Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[10] N. Franceschini, F. Ruffier, J. Serres, and S. Viollet, "Optic Flow Based Visual Guidance: From Flying Insects to Miniature Aerial Vehicles," *Aerial Vehicles*, Jan. 2009.

[11] A. Borst and M. Helmstaedter, "Common circuit design in fly and mammalian motion vision," *Nature Neuroscience*, vol. 18, no. 8, pp. 1067–1076, Aug. 2015.

[12] N. Franceschini, "Small Brains, Smart Machines: From Fly Vision to Robot Vision and Back Again," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 751–781, 2014.

[13] K. Yonehara and B. Roska, “Motion Detection: Neuronal Circuit Meets Theory,” *Cell*, vol. 154, no. 6, pp. 1188–1189, 2013.

[14] J. P. Lindemann, R. Kern, J. H. van Hateren, H. Ritter, and M. Egelhaaf, “On the computations analyzing natural optic flow: Quantitative model analysis of the blowfly motion vision pathway,” *The Journal of Neuroscience*, vol. 25, no. 27, pp. 6435–6448, 2005.

[15] W. Reichardt, “Nervous Integration in the Facet Eye,” *Biophysical Journal*, vol. 2, no. 2, pp. 121–143, 1962.

[16] C. V. Parise and M. O. Ernst, “Correlation detection as a general mechanism for multisensory integration,” *Nature Communications*, vol. 7, p. 11543, Jun. 2016.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California