BULLET DETECTION AND TRAJECTORY ESTIMATION
USING A SINGLE CAMERA

by

Kevin H. Murray
A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
In Partial Fulfillment of
The Requirements for the Degree
of
Master of Science
Computer Science

Committee:

_____	Dr. Zoran Duric, Thesis Director
_____	Dr. Yotam Gingold, Committee Member
_____	Dr. Chris Weiland, Committee Member
_____	Dr. Sanjeev Setia, Chairman, Department of Computer Science
_____	Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering
Date: _____	Fall Semester 2017 George Mason University Fairfax, VA

Bullet Detection and Trajectory Estimation Using a Single Camera

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

By

Kevin H. Murray
Master of Science
Virginia Tech, 2011

Director: Dr. Zoran Duric, Professor
Department of Computer Science

Fall Semester 2017
George Mason University
Fairfax, VA

Acknowledgments

I would like to thank my adviser, Dr. Zoran Duric, who was always available to offer advice and feedback, despite the issues and delays frequently encountered with this project.

I would also like to thank my other committee members, Dr. Chris Weiland and Dr. Yotam Gingold, for supporting this project. In particular, I'd like to thank Dr. Weiland for his continued professional support and encouragement.

Lastly, I'd like to thank my wife, Jessica. With two young kids at home, this project was as much work for her as it was for me.

Table of Contents

List of Tables

List of Figures

Abstract

BULLET DETECTION AND TRAJECTORY ESTIMATION USING A SINGLE CAMERA

Kevin H. Murray

George Mason University, 2017

Thesis Director: Dr. Zoran Duric

Closed-loop fire control systems greatly increase a weapon system's ability to successfully engage a target by measuring the trajectories of its outgoing projectiles. This has been demonstrated with both line-of-sight and non-line-of-sight systems, but research in autonomous Remote Weapon Stations (RWS's) currently lacks this capability. One of the major challenges in bringing this capability to RWS's is that radar is not seen as an appropriate sensor due to its size, cost, and active nature.

The goal of this research was to determine if projectile trajectories could be accurately measured with a camera. Algorithms were developed to detect small-caliber tracer rounds in images and then combine those detections with a ballistic model to estimate trajectories. These algorithms were successfully tested with live gunfire data, which showed that tracer rounds were readily detectable and that estimated trajectories accurately predicted impact points. Simulations also show that the algorithms can distinguish individual tracer trajectories from machine gun fire and eliminate several types of outliers. These results open the possibility of providing closed-loop fire-control to RWS's by using a camera as the main sensor.

Chapter 1: Introduction

1.1 Motivation

Closed-loop fire-control systems are able to correct a weapon's aim by observing outgoing projectile trajectories in comparison to the target location. This enables the system to 'walk-in' its gunfire to the target. One example of this concept is the Phalanx CWIS, which is a line-of-sight weapon system that uses radar to measure both the target position and the angle of outgoing bullets [2]. Another example is the Visual Automated Scoring System, which uses a UAV to make optical measurements of projectile impact locations for non-line-of-sight weapons [3].

This capability is currently lacking in autonomous remote weapon stations (RWSs). An RWS is a remotely operated, small caliber weapon that is often mounted on vehicles, such as Humvees and small boats. RWSs typically include cameras and other sensors that allow the weapon to be operated from a safe location. Recently, there has been a push to add autonomy to RWSs. In [4], cameras are used to detect and track potential targets, and the weapon can automatically aim and fire at the target when ordered to do so.

Manually controlled RWSs allow the operator to make fine adjustments to the weapon aim based on visual observations of tracer rounds. This feedback is missing from current approaches of autonomous RWSs. The motivation of this research is to provide closed-loop fire-control for autonomous RWSs, using camera observations of outgoing bullets.

The approach taken with this research is in some ways a combination of the two previously mentioned closed-loop fire-control systems. RWSs are line-of-sight weapons, so like the Phalanx, we will be making observations of bullets in flight. Unlike the Phalanx, radar is not seen as a suitable instrument for an RWS for several reasons. The first is that precision radar systems appear to be impractically large for a mobile RWS. The second reason

is that radar is an active sensor, which can potentially reveal the existence and position of the RWS. The third is cost: radar is expensive while cameras are not. In fact, it may be possible to use existing cameras on the RWS.

1.2 Scope

The scope of this research is to develop and test methods for estimating bullet trajectories with a camera. It is assumed that if these estimates can be done accurately, closed-loop fire-control based on this concept is possible.

I identified three main problems that needed to be solved in order to estimate bullet trajectories:

1. Bullet detection - in order to measure the trajectory with a camera, we must detect the bullet's pixel coordinates in several camera frames.

2. Camera calibration - we need to accurately relate camera pixels to three dimensional space. This allows the bullet's pixel coordinates to be transformed into three dimensional direction vectors.

3. Trajectory estimation - we need to transform direction measurements into a three-dimensional trajectory.

1.3 Outline

This research is easily broken down into several loosely-connected subproblems, which are presented here as individual chapters. Most of these chapters will present their own methodology, experimental results, and conclusions.

We will begin with background information on exterior ballistics, which is important for both bullet detection and trajectory estimation. Chapter 2 provides background on extrinsic ballistics in general, and some relevant ballistic models in particular.

A literature review is presented in chapter 3. This will be broken up in sections for each of the individual subproblems.

In chapter 4, I will detail the camera calibration routine used in this research. This is a significant topic because typical methods of calibration are not accurate for the narrow angle of view lenses needed here.

Chapters 5 and 6 detail the investigation in bullet detection and trajectory estimation, respectively, as completely separate topics. Here we are assuming that these will be completely separate processes, where the output of bullet detection will feed into trajectory estimation. Then in chapter 7, we will see how these two processes can be combined using random sample consensus (RANSAC).

Finally, overall conclusions and future work are discussed in chapter 8.

Chapter 2: Background

2.1 Introduction

The purpose of this chapter is to provide background information on the science of exterior ballistics that is relevant to this project. We are particularly concerned with small-caliber, spin-stabilized bullets fired at a maximum range of 1,000 meters. Exterior ballistics is used in two major ways in this project:

1. Bullet detection: combined with a camera model, an exterior ballistic model is used to predict the pixel locations of a bullet within a sequence of camera frames.

2. Trajectory estimation: bullet trajectories will be estimated by combining camera observations of a bullet with an exterior ballistic model.

We will begin with an overview of exterior ballistics, including the relevant kinematics and dynamics that affect bullet trajectories. The end of this chapter presents two exterior ballistic models that are relevant to the type of trajectories in this project.

2.2 Exterior Ballistics

2.2.1 Overview

Exterior ballistics is the study of projectiles in free-flight. In the case of firearms, the free-flight phase is the period after the bullet exits the barrel and before it impacts the target. The trajectory of the bullet is determined by its initial conditions, environmental effects, and the properties of the bullet. In this section, we will detail the most significant factors that shape bullet trajectories; effects that are only significant to large-caliber weapons will

be ignored. This information was predominantly obtained from [5] and [6], but may be considered common knowledge in the field of ballistics.

2.2.2 Kinematics

The kinematic state describes the bullet's position, orientation, velocity, and axial spin rate at time t. These components are shown in figure 2.1 and detailed below. The initial conditions are represented by the state at t_0.

- Position ($\mathbf{x}$): the location of the bullet at any time, relative to the chosen coordinate system. The initial position is typically considered to be the end of the gun barrel.
- Velocity ($\mathbf{u}$): the instantaneous velocity. The initial velocity magnitude is also known as the muzzle velocity.
- Orientation: the pitch (β) and yaw (α) of the bullet relative to the velocity direction. Since we are dealing with axially-symmetric projectiles, the roll angle does not affect the trajectory. Ideally, the pitch and yaw angles are zero, but 1-3 degrees is typical. This causes an increase in drag and a lateral velocity.
- Axial spin rate (p): Rifle bullets are typically spun at several thousand Hz in order to produce a gyroscopic effect that maintains alignment with the velocity direction. This spin is assumed to be entirely about the bullet's axis of revolution, so we are only concerned with the scalar magnitude of spin. We will not be concerned with the rate of change in the pitch and yaw angles, but an effect that partially results from the yaw rate will be approximated in section 2.3.

2.2.3 Dynamics

The kinematic state of a bullet is continuously altered by various forces and torques, particularly from the environment. For instance, the velocity and axial spin of the bullet are damped by atmospheric friction. The significant effects are shown in figure 2.2 and detailed below.

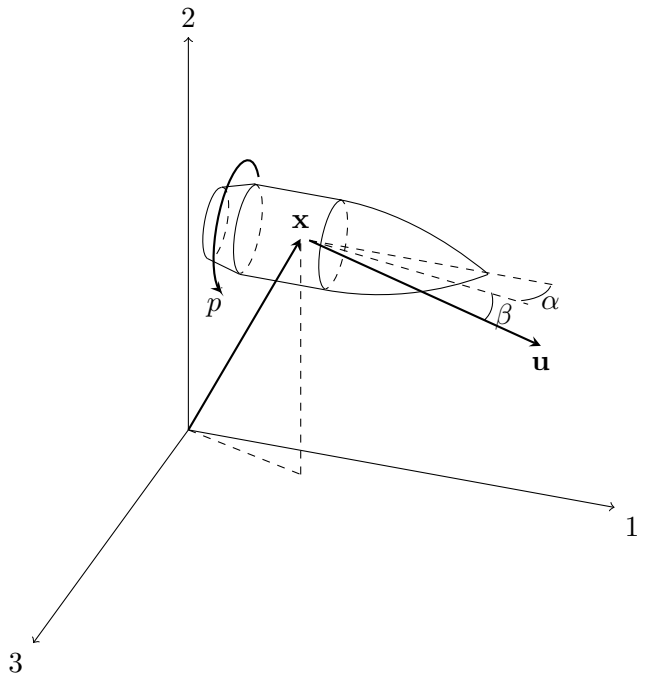

Figure 2.1: The components of a bullet's kinematic state.

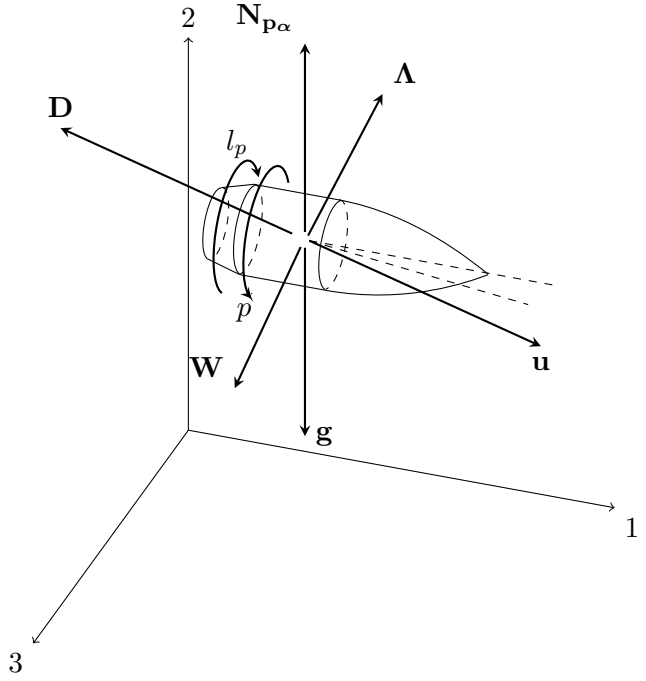

Figure 2.2: The environmental effects that alter bullet trajectories.

- Gravity ($\mathbf{g}$): for small-caliber weapons, this is very well-approximated by a vector of constant magnitude and direction.

- Drag ($\mathbf{D}$ and l_p): resistance force or torque due to friction between the bullet and air. The magnitude is roughly proportional to the velocity squared and the density of air (ρ). The drag force ($\mathbf{D}$) direction is always opposite of the velocity direction and the drag torque $\mathbf{l_p}$ is always opposite the rotational velocity. Since we assume the only significant rotational velocity is about the axial direction, we are only concerned with the magnitude of the drag torque, l_p.

- Wind ($\mathbf{W}$): the speed and direction of the air mass relative to the ground. Crosswind (wind perpendicular to shooting direction) causes a deflection of the bullet in the same direction as the wind. Rangewind (wind parallel to shooting direction) alters the force of drag, since drag is dependent on the bullet velocity relative to the air.

- Gyroscopic drift: a lateral velocity caused by the gyroscopic effect. Spin-stabilization results in high angular momentum, with its vector representation parallel to the bullet's axial direction. Additionally, the bullet has a small spin component along its pitch axis throughout its trajectory, due to gravity. This results in a torque vector pointing perpendicular to the velocity direction. A torque applied to an object with angular momentum in a different direction results in gyroscopic precession, where the object adds angular momentum that is perpendicular to both the angular momentum and the applied torque. In this case, a bullet tends to rotate to the right or left when viewed from a shooter looking down range. This results in a lift force that accelerates the bullet to the right or left.

- Coriolis force ($\mathbf{\Lambda}$): a fictitious force that arises due to describing bullet motion in a non-inertial reference frame. Since we typically use a reference frame where points on the surface of the Earth remain fixed, we do not account for the Earth's rotation, and thus we do not have a true inertial reference frame. The Earth's rotation is accounted for in this non-inertial reference frame by including this fictitious force.

- Magnus effect ($\mathbf{N_{p_\alpha}}$): a force perpendicular to the bullet's rotational velocity and lateral air movement (caused either by a crosswind or the bullet having a nonzero yaw). When this occurs, one side of the bullet has a tangential velocity in the same direction as the airflow, and another side has a tangential velocity in the opposite direction of the airflow. The airflow near the side of opposite tangential velocity tends to detach from the surface of the bullet and become turbulent. The airflow near the other side tends to attach to the surface of the bullet and gets thrown in a direction further along the rotation. This causes a lift force in the direction the air is thrown.

2.2.4 Projectile Properties

The properties of the projectiles influence the magnitudes of the forces and torques described above, and the resulting linear and angular accelerations.

- Mass (m): resistance to force. In exterior ballistics, the main effect of mass is to resist drag.
- Diameter (d): the caliber of the projectile.
- Axial moment of inertia (I_x): resistance to torque along the bullet's axis of revolution (axial direction). Since we are not concerned with the rate of change of the pitch and yaw angles, we are not concerned with the moments of inertial about those axes.
- Bullet reference area (S): the projected frontal area used in drag calculations.
- Drag coefficient (C_D): dimensionless number used to quantify drag based on bullet shape and airflow conditions. The value is typically measured experimentally and it varies significantly based on Mach.
- Lift coefficient (C_{L_α}): dimensionless number used to quantify lift based on bullet shape and airflow conditions. The value is typically measured experimentally and it varies significantly based on Mach.

- Magnus force coefficient ($C_{N_{p\alpha}}$): dimensionless number used to quantify the Magnus force based on bullet shape and airflow conditions. The value is typically measured experimentally and it varies significantly based on Mach.

2.3 The Modified Point-Mass (MPM) Model

This model was first derived in [7]. It was developed as a simplification of computationally expensive six degree of freedom models, which numerically solve for the position and orientation of the projectile throughout its trajectory. Rather than estimating the varying pitching and yawing motion during flight, the MPM model estimates the yaw due to gyroscopic drift, described in 2.2.3. This estimation is called the *yaw of repose*, and is labeled $\boldsymbol{\alpha}_{\mathbf{R}}$.

The MPM model is shown in equations 2.1 and 2.2.

$$\frac{d\mathbf{u}}{dt} = -\frac{\rho S C_D}{2m} v\mathbf{v} + \frac{\rho S C_{L_\alpha}}{2m} v^2 \boldsymbol{\alpha}_{\mathbf{R}} + \frac{\rho S d C_{N_\alpha}}{2m} p(\mathbf{v} \times \boldsymbol{\alpha}_{\mathbf{R}}) + \mathbf{g} + \boldsymbol{\Lambda} \tag{2.1}$$

$$\boldsymbol{\alpha}_{\mathbf{R}} = \frac{2 I_x p(\mathbf{g} \times \mathbf{v})}{\rho S d v^4 C_{M_\alpha}} \tag{2.2}$$

Equation 2.1 solves for the linear acceleration of the projectile by summing accelerations due to drag, lift, Magnus force, gravity and Coriolis force. Here, $\mathbf{u}$ is the velocity while $\mathbf{v}$ is the velocity relative to the wind. Equation 2.2 calculates the instantaneous yaw of repose and is coupled with equation 2.1. The position of the projectile is calculated by twice numerically integrating equation 2.1 with known initial conditions.

2.4 The Flat-Fire Point-Mass (FFPM) Model

This model reduces the point mass model into closed-form equations of motion by making the following approximations:

1. The projectile trajectory is flat: i.e. one component of its velocity is always much higher than the other two components.

2. The relationship between Mach number and drag coefficient can be modeled with a single parameter.

3. The effect of range wind is insignificant.

4. The Magnus effect, Coriolis force, and gyroscopic drift are all insignificant.

The FFPM equations of motion are derived in [5] and shown in equation 2.3. The FFPM model requires that the coordinate system be set up such that there is no z-component of velocity and the negative y-component lines up exactly with the direction of gravity.

$$x = x_0 + \frac{2V_{0_x}t}{\sqrt{V_{0_x}}k_3t + 2} \tag{2.3a}$$

$$y = y_0 + x\frac{V_{0_y}}{V_{0_x}} - \frac{1}{2}gt^2\left[\frac{1}{3}\left(1 + \sqrt{V_x/V_{0_x}}\right)\right] \tag{2.3b}$$

$$z = z_0 + W_z\left(t - \frac{x}{V_{0_x}}\right) \tag{2.3c}$$

$$k_3 = \frac{\rho S K_3\sqrt{a}}{2m} \tag{2.3d}$$

$$V_x = \left(\sqrt{V_{0_x}} - \frac{1}{2}k_3x\right)^2 + 2W_x\left[1 - \frac{\left(\sqrt{V_{0_x}} - \frac{1}{2}k_3x\right)^2}{V_{0_x}}\right] \tag{2.3e}$$

Chapter 3: Literature Review

As mentioned in chapter 1, this thesis is easily broken down into several subproblems. This literature review is organized based on each of these subproblems.

3.1 Camera Calibration

Zhang's method [8] is perhaps the de facto standard method of camera calibration. It is performed by imaging a planar pattern at multiple unknown poses, providing correspondences between world points on the pattern and observed pixel locations. It is simple to carry out and is generally very accurate. However, as noted by [9], this method loses precision as the focal length is increased. Estimating the focal length requires observations of the perspective foreshortening of the calibration target. At narrow angles of view, the focal length becomes extremely sensitive to this foreshortening.

Another method is to use stars in the night sky as the calibration target, which I will label *stellar calibration*. This works because the angular position of stars relative to Earth is known to high precision. One particular approach to this method is found in [10]. In this approach, the centroids of stars are detected using a series of steps: percentage threshold, upscaling, image gradient, and maximization of an energy function (the normalized sum of the border gradients). The detected stars are then matched with known stars in an almanac, using a RANSAC algorithm to find the camera orientation and focal length that best matches detected stars with known stars. Finally, an optimization algorithm is used to calculate the optical center and distortion coefficients, and to refine the orientation and focal length estimates. According to the authors, the advantage of this technique over planar target calibration is that only a single image is required.

Identifying the stars in the image is perhaps the most difficult part of stellar calibration.

In [11], a method of performing this identification was developed for the purpose of automatically providing astrometric data to arbitrary images of the night sky. They first define a scale and rotation invariant feature of a quadruple of stars. Then they compute an inverted index of a star almanac using this feature. When an image is input into the method, features are computed and then searched for in the index. The result is an extremely robust method that can identify stars in images of arbitrary focal length and orientation.

3.2 Bullet Detection

The literature on bullet detection is focused on making standard ammunition (non-tracers) detectable in cameras. Thus, there are few details on actual image processing techniques.

Bullets tend to emit strongly in the mid-wave infrared (MWIR) band due to heating from the propulsion in the gun barrel and from friction as they move at high speeds through the atmosphere. In [12], a computational fluid dynamics analysis estimated bullet surface temperatures of 500 K for bullets traveling at $800\,\mathrm{m\,s^{-1}}$. For a blackbody, this temperature corresponds to a peak in electromagnetic radiation of about 6 µm, well within the MWIR range of 3 µm to 8 µm.

In [13], it was shown that bullets could be detected with a (MWIR) camera. As their application was for defensive systems, they showed detections with bullets traveling perpendicular to the camera's principal axis, toward the camera, or somewhere in between.

In [14], surface temperatures of a bullet were experimentally measured with a MWIR camera. It was shown that the temperature of a bullet varies greatly across its surface, with the highest temperatures observed at the bullet tip and rifling grooves. High temperatures were also measured at the tail of the bullet, but it was suspected that this was actually MWIR from the muzzle blast being reflected off the bullet's tail.

While the literature shows that bullets are generally detectable in MWIR, an exception may be when viewing the rear of the bullet. This is a major concern for this project, since the RWS cameras are located close to the gun, and thus only observe the rear of the bullet.

3.3 Trajectory Estimation

The existing literature on trajectory estimation is thin, and only two relevant approaches have been found. The previously discussed system in [13] uses estimated pixel-space trajectories to locate the pixel coordinate of the shooter. Each bullet is tracked and fit with a line in pixel space. As the firing angle of the weapon changes, these lines intersect with each other at the pixel location of the shooter. For the purposes of this project, there are two main limitations of this approach. The first is that it is limited to pixel space and cannot estimate distance. The second is that while the linear fit of the trajectories is suitable for short distances (and thus for locating the shooter), it doesn't work well for estimating full trajectories.

A more relevant approach is discussed in [15]. The purpose of that research was to combine sensor data of a projectile in flight with a ballistic model. Specifically, they used optimization algorithms to estimate the projectile's ballistic parameters based on the sensor data. In this way, they could reconstruct trajectories even when the sensor data only covered a portion of the trajectory. The ballistic model they used was a closed-form approximation of 6 degree-of-freedom models for fin-stabilized projectiles. Radar was the primary sensor, which provided three-dimensional positions at known points in time.

3.4 Random Sample Consensus (RANSAC)

RANSAC is an iterative algorithm that isolates outliers from a dataset. It does this by searching for model parameter values that result in the greatest number of inliers. At each iteration, a set of parameter values are calculated based on a random subset of the data. The entire dataset is then evaluated against these parameters, and any data that agree with the parameter values (within some threshold) are considered inliers, and everything else is considered outliers. The algorithm requires the following to be specified:

- The model that the data will be fitted to.

- The minimum number of data points required to fit the model.
- A threshold value that distinguishes outliers from inliers.
- The minimum number of inliers required to accept a set of model parameters.
- The maximum number of iterations.

In order to obtain valid parameters, one iteration must randomly select a set of data points that is entirely inliers. If the ratio of inliers to data points is known, the number of iterations can be chosen such that there is a particular probability of obtaining valid parameters. There is no way to guarantee the algorithm will compute a valid set of model parameters.

The RANSAC algorithm was introduced by [16] in 1981. Since, then several modifications and improvements have been proposed to increase performance and extend the capability.

In [17], it was recognized that the computation time of RANSAC is proportional to the size of the dataset, since each set of parameter values is evaluated against the entire dataset. It was shown that the computation time could be significantly decreased by evaluating each set of parameters against another random subset of the data, without a significant cost in accuracy.

The original RANSAC algorithm made the assumption that there was a single instance of the specified model in the data. Several researchers have developed extensions of RANSAC that can identify multiple instances of a model. The most basic approach, *sequential RANSAC*, works by running RANSAC multiple times. With each run, one of the model instances is randomly identified, and its associated inliers are removed from the dataset. This approach was successfully used in [18] and [19] for detecting planar regions in stereo cameras. An advantage of this approach is that the number of model instances does not need to be known; the process can be repeated until an insufficient number of inliers remain. A disadvantage of this approach is that if an early extracted model instance contains inaccurate inliers, this compounds into unstable extractions of subsequent model

instances. This deficiency was resolved in [20] with their multiRANSAC algorithm, which extracts model instances in parallel. The cost of this approach is a significantly increased computation time, however.

Chapter 4: Camera Calibration

4.1 Introduction

Since we are relating camera observations of a bullet to a three-dimensional trajectory, we need an accurate method of calibrating camera pixels with three-dimensional space. This chapter shows that using stars in the night sky is potentially more convenient and accurate than standard methods of calibration.

Section 4.2 provides background information on the intrinsic camera model used in this chapter, along with camera calibration in general. In section 4.3, we detail how the two calibration methods discussed in the literature review will be evaluated. Section 4.4 provides experimental results and discussions.

4.2 Background

An intrinsic camera model is used to relate three-dimensional space with camera pixels. The camera model I use in this chapter is from [21], and copied in equations 4.1-4.5 below:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} fc(1) & alpha\_c * fc(1) & cc(1) \\ 0 & fc(2) & cc(2) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d(1) \\ x_d(2) \\ 1 \end{bmatrix} \tag{4.1}$$

$$x_d = \begin{bmatrix} x_d(1) \\ x_d(2) \end{bmatrix} = (1 + kc(1)r^2 + kc(2)r^4 + kc(5)r^6)x_n + dx \tag{4.2}$$

$$x_n = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.3}$$

$$dx = \begin{bmatrix} 2kc(3)xy + kc(4)(r^2 + 2x^2) \\ kc(3)(r^2 + 2y^2) + 2kc(4)xy \end{bmatrix} \tag{4.4}$$

$$r^2 = x^2 + y^2 \tag{4.5}$$

This model allows a point in world coordinates ($[X_c\ Y_c\ Z_c]$) to be projected into pixel coordinates ($[x_p\ y_p]$), based on the intrinsic parameters of the camera, which are listed below:

- fc: a 2x1 vector containing the x and y focal length, in pixels
- cc: a 2x1 vector containing the principal point, in pixel coordinates
- $alpha\_c$: the skew coefficient between the x and y axes
- kc: a 5x1 vector containing radial and tangential distortion coefficients

The purpose of intrinsic camera calibration is to estimate the values of these parameters. When the parameters of the camera are known, the camera model can be used to calculate the pixel location of a known three-dimensional point. And by inverting the camera model, a three-dimensional ray can be calculated for any pixel coordinate.

Calibration is performed by imaging something with known geometry, which provides correspondences between pixel coordinates and world coordinates. Optimization algorithms are used to estimate the intrinsic parameters in the camera model that best transform the known world coordinates into the observed pixel coordinates. Specifically, the algorithms minimize the reprojection error, which is the average distance between the observed pixel

coordinates of the geometry, and the expected pixel coordinates based on the estimated intrinsic parameters and the geometry.

4.3 Methodology

The purpose of this research was to evaluate an appropriate method of camera calibration for the narrow angle-of-view cameras used in this project. Specifically, I wanted to evaluate the accuracy of the two calibration procedures mentioned in section 3.1.

According to [21], the principal point is often difficult to estimate precisely, and it may be better to set it to the center pixel. That appears to be the case with this camera, as the calibration procedure from [21] would not converge on a value for principal point, and the stellar calibration method converged on nonsensical values. Also according to [21], the skew coefficient can safely be assumed to be zero. For these reasons, both methods were only used to estimated focal length and distortion.

The literature indicated that Zhang's method may not be able to precisely estimate focal length for a narrow angle-of-view camera. I expect that Zhang's method is ill-conditioned for narrow angle-of-view cameras because the observed perspective foreshortening changes very little with large changes in focal length. Conversely, I expect that stellar calibration does not have this problem, since the angles between stars are precisely known.

Both calibration procedures were evaluated with a monochrome, visible light Point Grey camera (model FL3-U3-13Y3M-C). A 100 mm fixed focal length lens provided 3.52° by 2.82° angle of views (horizontal by vertical). The known focal length provides one known intrinsic model parameter, but I do not have a method of obtaining truth data on the distortion parameters. However, if Zhang's method is ill-conditioned for narrow angle-of-view cameras, then the intrinsic model it computes should have low reprojection error on the planar pattern, but high reprojection error on the stellar data. Conversely, the intrinsic model computed with stellar calibration should have low reprojection error on both datasets. The methodologies for both calibration procedures are described in the following subsections.

4.3.1 Zhang's Method

The particular implementation of Zhang's method I tested came from [21], which includes MATLAB ® software to estimate the camera's intrinsic parameters.

The calibration target is shown in Figure 4.1. I used a target with a large pattern so that I could make large angles between the target and the camera sensor plane and still see the pattern. This way, the foreshortening effect would be maximized and I'd provide the greatest opportunity to estimate the focal length. Even at a severe angle, however, it's easily apparent that the foreshortening effect is minimal, as shown in Figure 4.2.

I captured a total of 32 images of the target at widely varying poses, extracted the pattern centroids, and then ran the pixel and target coordinate correspondences through the algorithm provided in [21], which produces the estimated intrinsic parameters.

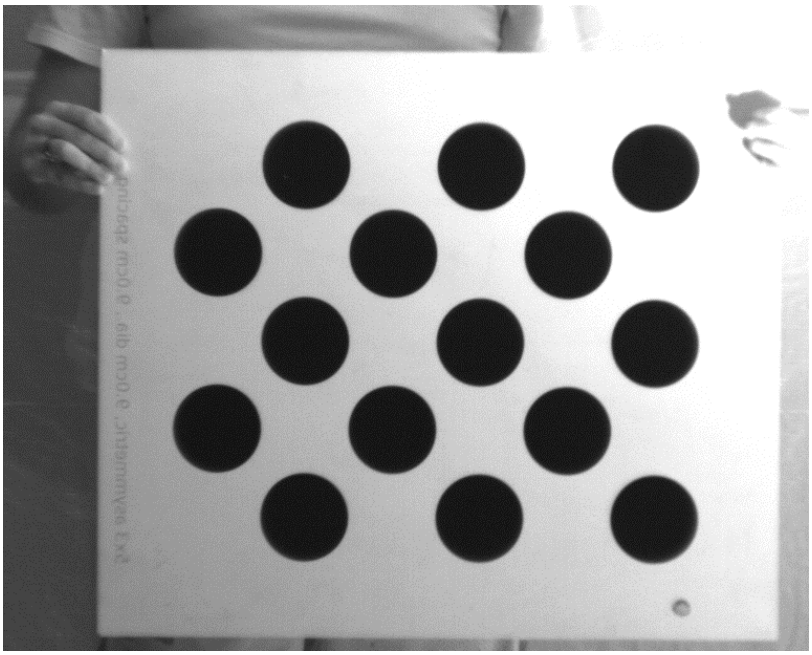

Figure 4.1: The calibration target.

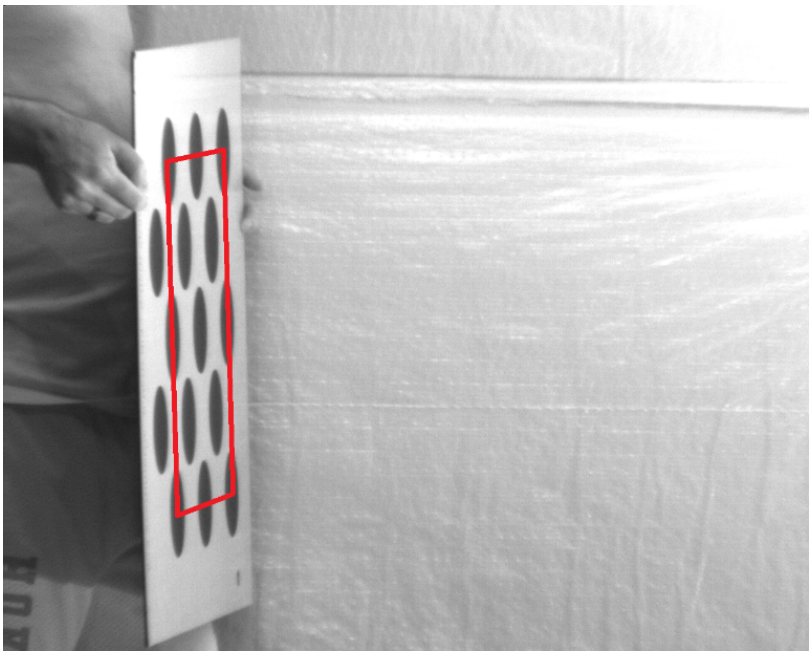

Figure 4.2: The calibration target at a severe angle relative to the image sensor.

4.3.2 Stellar Calibration

For this procedure, I captured an image of the night sky of the constellation Orion's Belt. I used a two second exposure to maximize the number of stars seen without getting significant motion blur from the Earth's rotation. The image is shown in Figure 4.3.

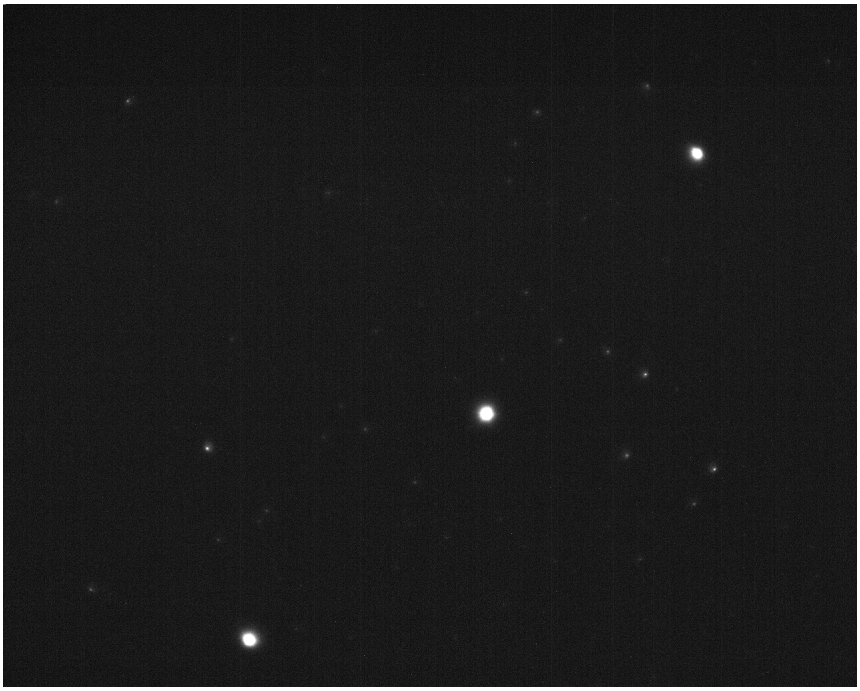

Figure 4.3: Captured image of Orion's belt.

I then input this image into the star identification software provided by [11]. The detected and recognized stars are shown in Figure 4.4, and the successfully recognized constellation is shown in Figure 4.5. A total of 23 stars were recognized in this image.

One of the outputs of this software is a set of correspondences between observed pixel coordinates of the stars and the spherical coordinates of the stars. I used these correspondences with the routine from [10] to estimate the intrinsic parameters of the camera.

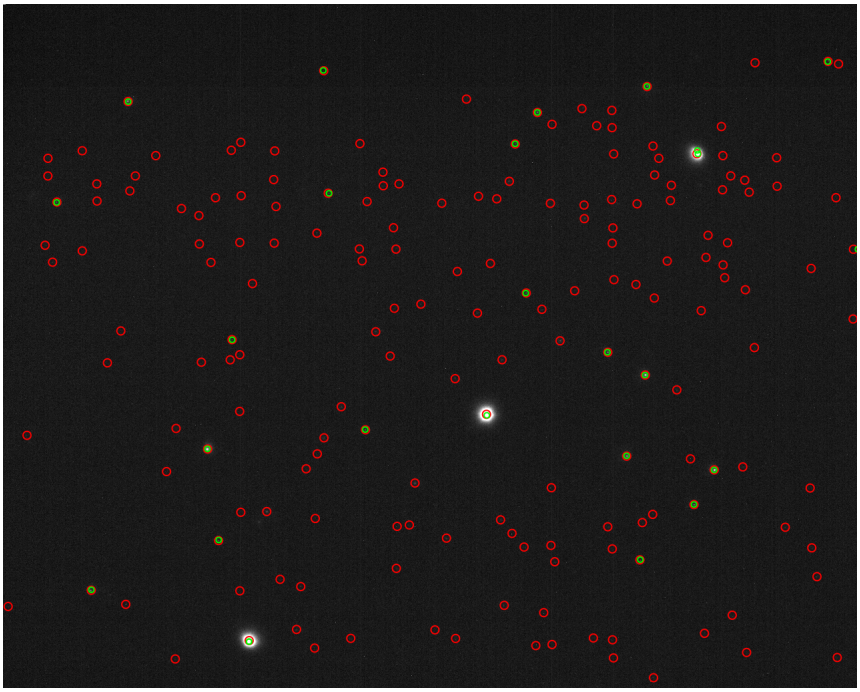

Figure 4.4: Detected and identified stars shown in the image of Orion's Belt. Detected stars are circled in red and identified stars are circled in green.

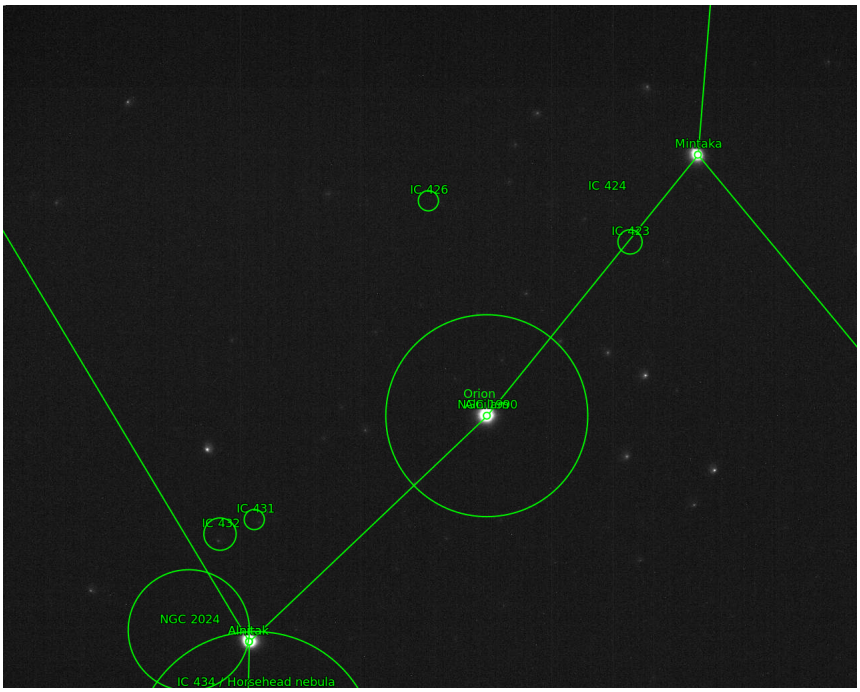

Figure 4.5: Constellation of Orion is successfully recognized.

4.3.3 Reprojection Error

One of the algorithms provided by [21] calculates reprojection error by estimating the pose of each planar target position, projecting the features onto the camera model, and comparing the projected pixel locations to the actual pixel locations. Since both calibration procedures used the same model parameters, I was able to calculate the reprojection error of both models on the planar target data.

A similar approach was used to estimate the reprojection error using the stellar data. I used the camera models and stellar correspondences to estimate the camera orientations, then projected the known direction to the stars onto the camera models, and compared the projected pixel locations to the actual pixel locations.

4.4 Experimental Results

The results of the calibration are shown in Table 4.1. As expected, Zhang's method does not obtain a precise estimate of the focal length, and it has a large reprojection error using the stellar data. The stellar calibration method obtains a very precise estimate of the focal length and has small reprojection error for both data sets.

Table 4.1: Calibration Results

Method	Focal length (mm)	Distortion	E_{planar} (px)	E_{stars} (px)
Zhang's	108.1	[8.01, -8152, -0.03, -0.02, 0]	0.73	35.96
Stellar	99.96	[-1.11, 0.39, -0.01, 0.00, 0.48]	0.77	0.37

Chapter 5: Bullet Detection

5.1 Introduction

As we are intending to measure bullet trajectories with a camera, extracting the pixel coordinates of bullets is a fundamental requirement. This chapter discusses three image processing filters that are selected or developed, and tested with actual gunfire data. Testing is completely done using tracer rounds, but the possibility of detecting standard ammunition will also be discussed. Once the filters are applied, pixel-time observations can be extracted using standard thresholding and connected-component labeling.

Since RWSs are the primary application of this research, I am only considering configurations that are relevant to an RWS. Namely, that the gun and camera are close to each other and well-aligned.

In section 5.2, I present three image filters that have been selected or developed for isolating bullets in images. Section 5.3 details how these filters will be evaluated and section 5.4 provides experimental results and discussion.

5.2 Theory

This section details the development and selection of three image processing filters used for detecting tracer rounds. I expect that these filters would also be applicable to detecting standard ammunition in MWIR images, but this has not been tested.

Perhaps the most challenging aspect of bullet detection is that the appearance of a bullet is so simple, there is little to distinguish it from the rest of the image. Only a single appearance-based filter was tested: a difference of Gaussians filter, described in section 5.2.1. The other two filters are based on the motion of bullets: a simple difference filter

(section 5.2.2) and a prediction filter (section 5.2.3).

5.2.1 Difference of Gaussians Filter

A Gaussian filter is a spatial low-pass filter, applied by convolving an image with a two-dimensional Gaussian function (shown in eq. 5.1a). The cutoff frequency is described in eq. 5.1b. This is alternatively represented as the cutoff wavelength, shown in eq. 5.1c. The cutoff wavelength is in pixels, and represents the feature sizes that are suppressed. The content at this wavelength has a response according to eq. 5.1d. Thus we can specify σ to suppress features at or below a particular size, and to a magnitude specified by eq. 5.1d (roughly, since this is not an ideal filter). The effect of varying σ is readily seen in figure 5.1.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5.1a}$$

$$f_c = \frac{\sqrt{2\ln c}}{2\pi\sigma} \tag{5.1b}$$

$$w_c = \frac{2\pi\sigma}{\sqrt{2\ln c}} \tag{5.1c}$$

$$c = \frac{1}{response} \tag{5.1d}$$

Figure 5.1: The effect of varying σ in the Gaussian filter. Image obtained from [1].

We can also use the Gaussian function as a high-pass filter by simply subtracting the low-pass filtered image from the original image. We can then create a bandpass filter by using the Gaussian function as both a high-pass filter and a low-pass filter. This is the difference of Gaussians (DoG) filter, and it is shown in eq. 5.2.

$$DoG(I) = I * \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - I * \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}} \tag{5.2}$$

The first term in eq. 5.2 is the high-frequency filter and the second term (including the negative sign) is the low-frequency filter. Thus it is required that $\sigma_2 > \sigma_1$.

For bullet detection, σ_1 and σ_2 need to be specified to produce cutoff wavelengths surrounding the size distribution of bullets as they appear in images. This distribution depends on the caliber of bullet being fired, the image sensor size, lens focal length, and distance of the bullet from the camera. The first three factors are easily accounted for with

known hardware. The last factor could be accounted for using a ballistic model, but with machine gun fire, there may be multiple bullets in the field of view at the same time. For that reason, I used a single DoG filter that surrounded the size distribution of a bullet throughout its trajectory.

5.2.2 Difference Filter

The difference filter simply subtracts the previous image from the current image. This is used as a simple motion detector. With an RWS having its camera near and aligned with the gun, the apparent motion of the bullet is fairly low, with much of its motion due to gravity. However, the motion between frames is generally greater than the size of the bullet. The major drawback of this filter is that it assumes the camera is not moving. Any camera motion would have to be accurately known and compensated for. For this research, I assume a static camera.

5.2.3 Prediction Filter

Since we are detecting our own gunfire, we should be able to predict where we will see it. The prediction filter computes a predicted trajectory using the Flat Fire ballistic model (described in 2.4) and projects it into image space. This allows us to predict a pixel coordinate location of the bullet at any point in time. This assumes that we know the ballistic parameters required to generate a trajectory (trigger time, muzzle velocity, drag, etc.) and the extrinsic and intrinsic parameters of the camera. I believe this is a reasonable assumption, since these parameters can all be measured or controlled.

However, there will be some uncertainty in these ballistic and camera parameters, otherwise we would never miss the target. The goal of the prediction filter is to propagate uncertainty in these parameters into uncertainty in the pixel coordinates of a bullet. In this way, it estimates a probability density function (PDF) of the bullet's pixel location at any point in time. The PDF can then be convolved with the original image, weighting each pixel in the input image by the relative probability that the bullet appears at that pixel.

The PDF is approximated as a two-dimensional Gaussian distribution with zero rotation. The center of the Gaussian is the predicted pixel location of the bullet. The horizontal and vertical standard deviations are estimated using uncertainty propagation. The procedure for generating the prediction filter is as follows:

1. Estimate bullet location in Flat Fire (FF) coordinates according to eq. 2.3, with estimated ballistic parameters:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = FF(t, t_0, x_0, y_0, z_0, V_0, ...) \tag{5.3}$$

2. Transform bullet location into camera coordinates with estimated intrinsic and extrinsic parameters, providing the center pixel of the Gaussian (using pinhole camera approximation with zero skew):

$$\begin{bmatrix} c \\ r \\ 1 \end{bmatrix} = \begin{bmatrix} f/s & 0 & o_x \\ 0 & f/s & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5.4}$$

3. Propagate uncertainty in all ballistic and camera parameters into uncertainty in pixel column and row. This is done via Taylor expansion and assuming all parameters are independent. The derivation of this uncertainty propagation is shown in Appendix A. The result of this are estimates of the Gaussian standard deviations, in pixel columns and rows.

5.3 Methodology

The filters were tested with a rifle firing 5.56 mm tracer rounds and a monochrome, visible light camera (Point Grey model FL3-U3-13Y3M-C). The camera was placed at a roughly 1.5 m offset from the gun and aligned with the firing direction to simulate the geometry of an RWS. Targets were placed at ranges of 250 m to 300 m from the gun. Tracer trajectories were recorded at frame rates between 70 Hz and 90 Hz. The resulting images were then processed with the previously discussed filters.

For the difference of Gaussians filter, σ_1 and σ_2 were assigned based on the smallest and largest apparent sizes of the tracer rounds throughout their trajectory. The assumption here is that these values would be obtained by experiment and remain relatively consistent in practice. For the value c, a response of 0.15 appeared to work best. The values obtained for σ_1 and σ_2 were 0.62 px and 4.65 px, respectively. The Gaussians were discretized on windows of 5x5 px and 29x29 px, respectively.

For the difference filter, each frame was simply subtracted by its previous frame.

For the prediction filter, ballistic and camera parameters were either measured or obtained from other sources (e.g. the manufacturer or range tables). Uncertainties were more difficult to obtain. In some cases, such as for muzzle velocity, this data does exist. In many cases, I could only use values that appeared reasonable. Future testing would hopefully refine these estimates.

5.4 Experimental Results

5.4.1 Difference of Gaussians

The difference of Gaussians filter shows good results throughout tracer trajectories. On average, the filter reduced the brightness of input images by 94%, and only moderately dampened the appearance of the tracer. An example of this filter is shown in figures 5.2 and 5.3, which are the input and output, respectively. The input and output image histograms are shown in figures 5.4 and 5.5, respectively. The effect of the filter on tracers

is shown in figure 5.6. While there is clearly some dampening of the tracer brightness, the effect is moderate despite widely varying apparent sizes of the tracer.

Figure 5.2: The input image to the difference of Gaussians filter. The tracer is slightly left and up from the image center.

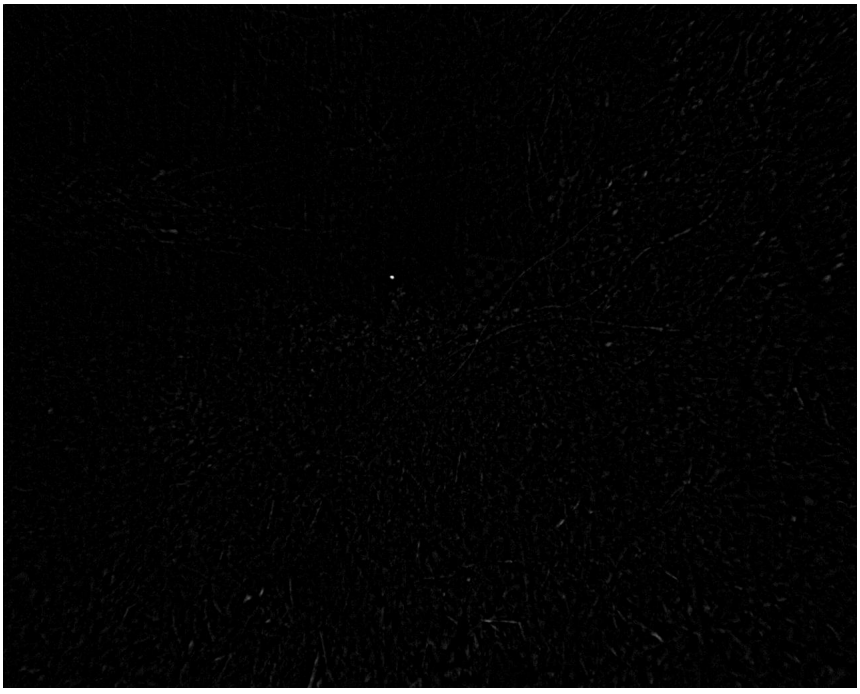

Figure 5.3: The output image from the difference of Gaussians filter. The tracer is slightly left and up from the image center.

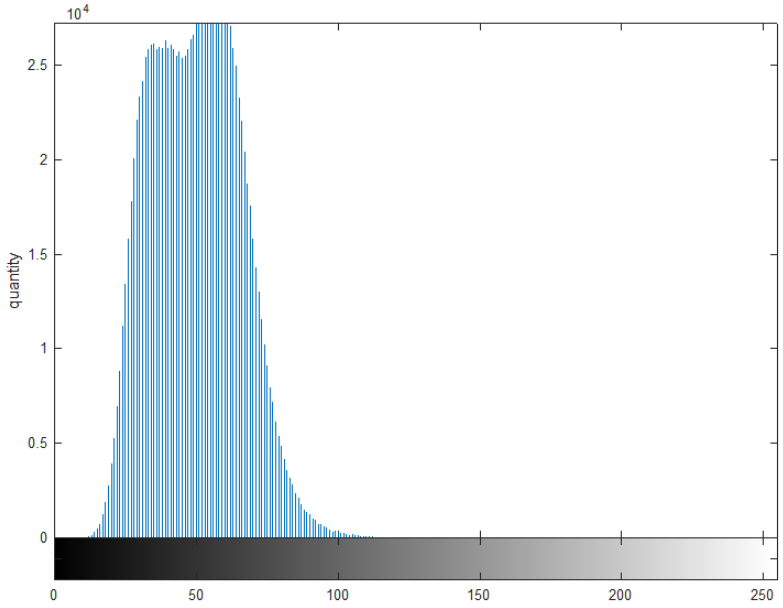

Figure 5.4: The brightness histogram of the input image to the difference of Gaussians filter.

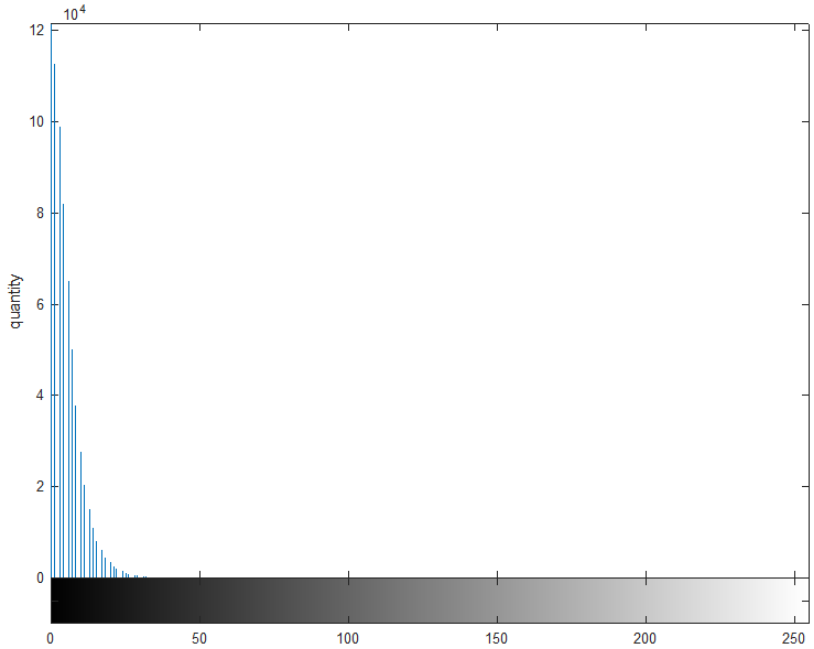

Figure 5.5: The brightness histogram of the output image from the difference of Gaussians filter.

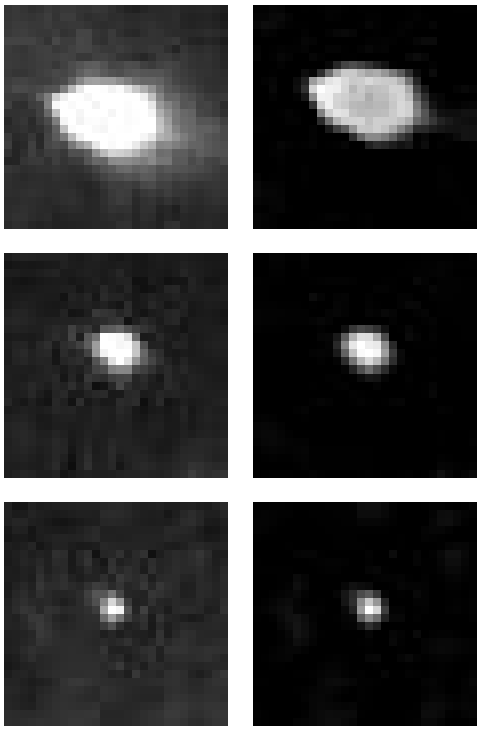

Figure 5.6: The effect of the difference of Gaussian filter on three tracer rounds at varying distances. The left column are inputs and the right column are corresponding outputs.

5.4.2 Difference Filter

The difference filter shows excellent results throughout tracer trajectories. On average, the filter reduced the brightness of input images by 95%, and had no noticeable effect on the appearance of the tracer. However, this level of performance would not be expected in practice, where the camera could be moving and experience vibration from the gun firing.

5.4.3 Prediction Filter

The prediction filter successfully tracks the motion of tracer rounds, and effectively crops out regions of the image that are far from the expected location of the tracers. In this way it acts as an outlier filter.

An example of a PDF calculated with this filter is shown in figure 5.7. This PDF was calculated for a tracer at 100 ms into its flight, and just 38 ms after entering the camera's field of view. The effect of using this PDF as a filter is shown in figures 5.8 and 5.9, which are the input and output, respectively.

For this PDF, there is more uncertainty in the pixel column than the pixel row. This is largely due to firing time uncertainty, which has a greater effect on the pixel column because the tracer moves primarily in the column direction shortly after it enters the camera's field of view. This effect decreases over time, at which point orientation uncertainty is the most significant uncertainty.

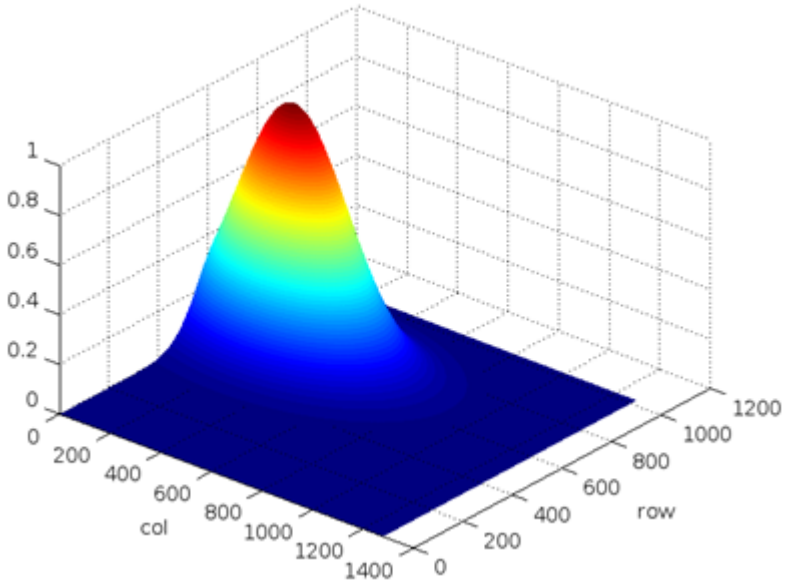

Figure 5.7: The probability density function (PDF) calculated for a tracer at 100 ms after being fired.

Figure 5.8: The input image to the prediction filter. The tracer is left of the image center.

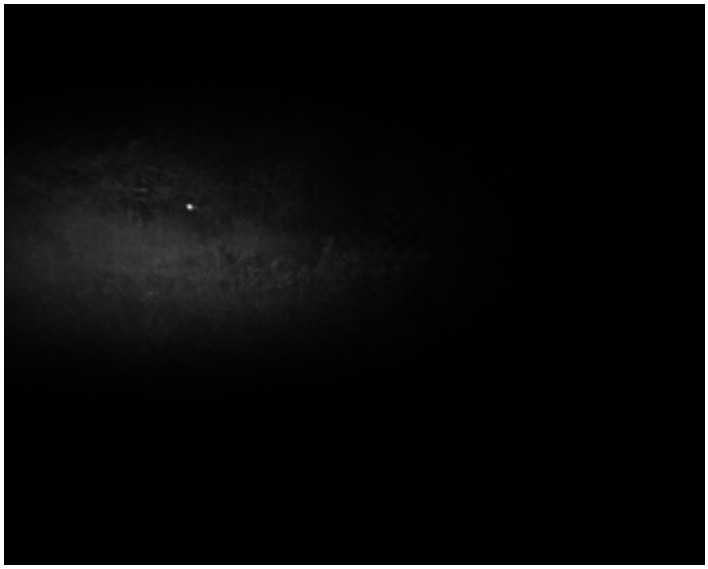

Figure 5.9: The output image from the prediction filter. The tracer is left of the image center.

Chapter 6: Trajectory Estimation

6.1 Introduction

This chapter documents my research in transforming detected bullets into estimated trajectories. Here it is assumed that we have already detected a bullet in several frames, giving us a list of pixel-time (time, column, row) coordinates of the bullet.

In section 6.2, I present a method for transforming pixel-time observations of a bullet into a three-dimensional trajectory. Section 6.3 details how this method will be evaluated, and section 6.4 shows experimental results with live gunfire.

6.2 Theory

The approach used here is similar to [15], where ballistic model parameters are estimated based on observations of the projectile in flight. I also use a parameter estimation technique, but with a ballistic model more appropriate for small caliber projectiles, and with a camera as the sensor rather than radar.

The first part of this section provides justification for using the Flat Fire ballistic model in the parameter estimation procedure. The second part details the actual parameter estimation procedure.

6.2.1 Model Selection

In chapter 2, two ballistic models were presented: the Modified Point Mass Model (MMPM) and the Flat Fire Model (FFM). The MPMM was developed for complex trajectories, such as artillery. For small-caliber projectiles, the trajectories it computes are considered nearly exact when input parameters are precisely known [5].

The disadvantage of the MPMM is that it is an open-form model, and thus requires numerics to solve. This is especially problematic for the parameter estimation technique developed in this chapter, where many trajectories are iteratively computed.

The FFM is a closed-form approximation for small-caliber trajectories. As such, it is much faster to compute than the MPMM, but less accurate. The purpose of this section is to evaluate the FFM's accuracy for trajectories relevant to an RWS. I made this evaluation by providing the same initial conditions, environment conditions (standard atmospheric conditions), and projectile properties to both models, and compared the difference in target impact position and time. To be relevant to an RWS, I used 7.62x51 mm NATO projectile properties fired at 850 m/s.

I compared the models firing at elevations between -30° and $+30^\circ$, and at targets between 100 and 900 yards. The distance between impact points is shown in figure 6.1 and the difference in impact times is shown in figure 6.2.

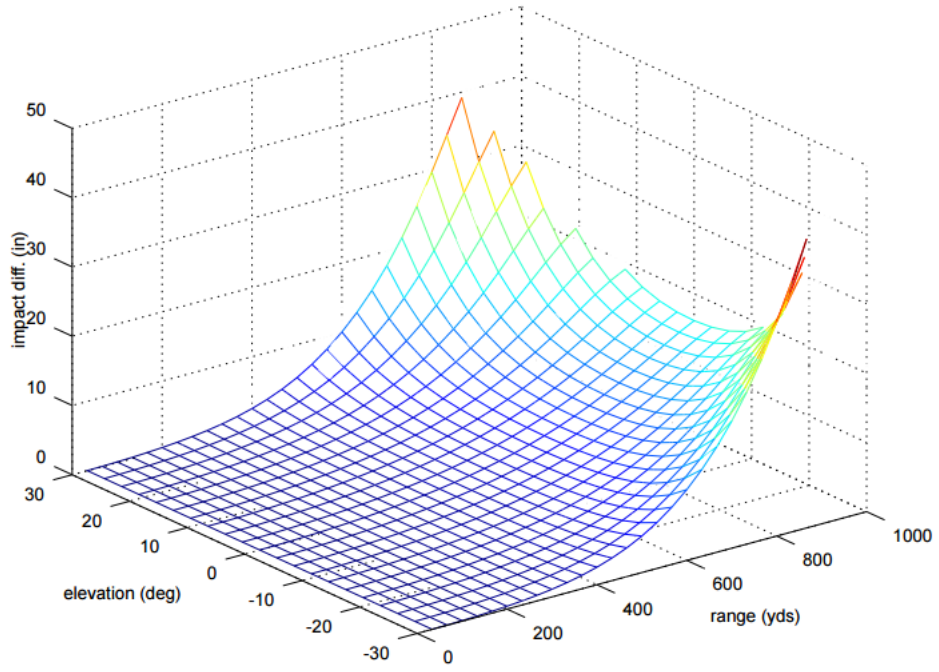

Figure 6.1: Difference in impact position between Flat Fire model and Modified Point-Mass model for various weapon elevations and target distances.

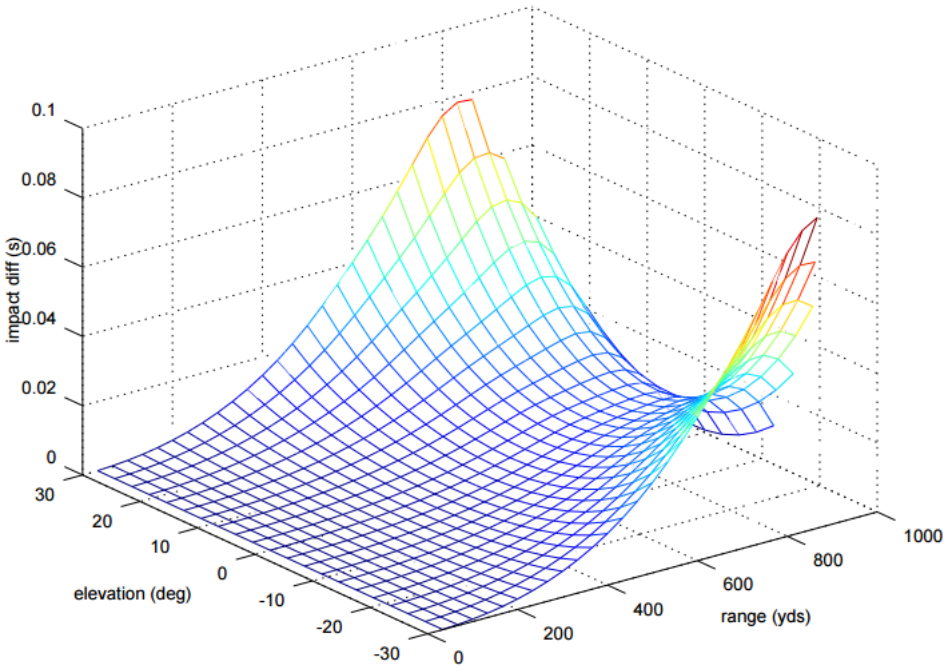

Figure 6.2: Difference in impact time between Flat Fire model and Modified Point-Mass model for various weapon elevations and target distances.

The FFM clearly deviates from the MPMM at long ranges and severe elevation angles. Combining long ranges with severe elevations is especially problematic, but this would be an uncommon situation. Hitting a target at 1,000 yards requires less than 1° of elevation. Severe elevations would only occur when the RWS and the target are at significantly different altitudes. Based on these plots, I would expect the FFM to estimate impact position errors within 20 in and impact time errors within 0.02 s for the majority of RWS engagements. The impact position error could easily be well within the spread of the RWS weapon, and the impact time error would only be significant for targets moving at very high speeds perpendicular to the bullet trajectory. For these reasons, I use the FFM for the trajectory estimation procedure developed in this chapter.

6.2.2 Parameter Estimation

The parameter estimation technique searches for the set of FFM parameters that generate a trajectory in agreement with camera observations. This concept is illustrated in figures 6.3, 6.4, and 6.5.

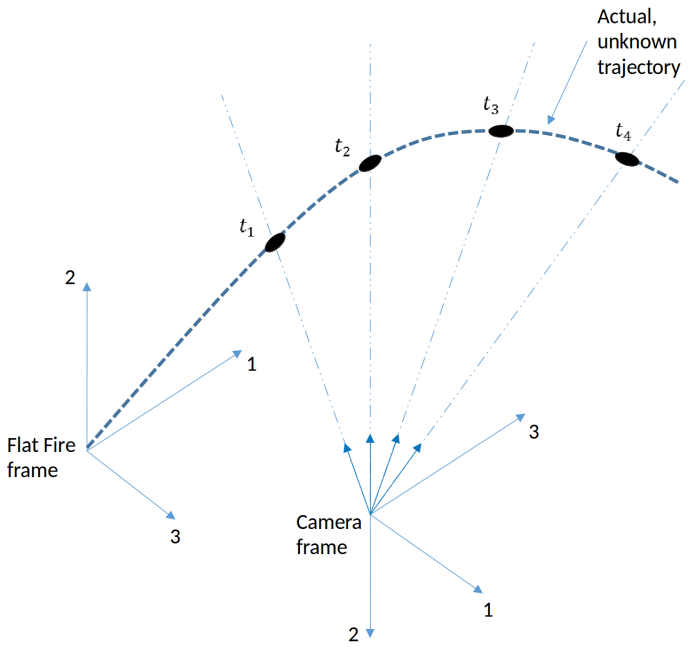

Figure 6.3: The camera makes pixel-time observations of the bullet in flight.

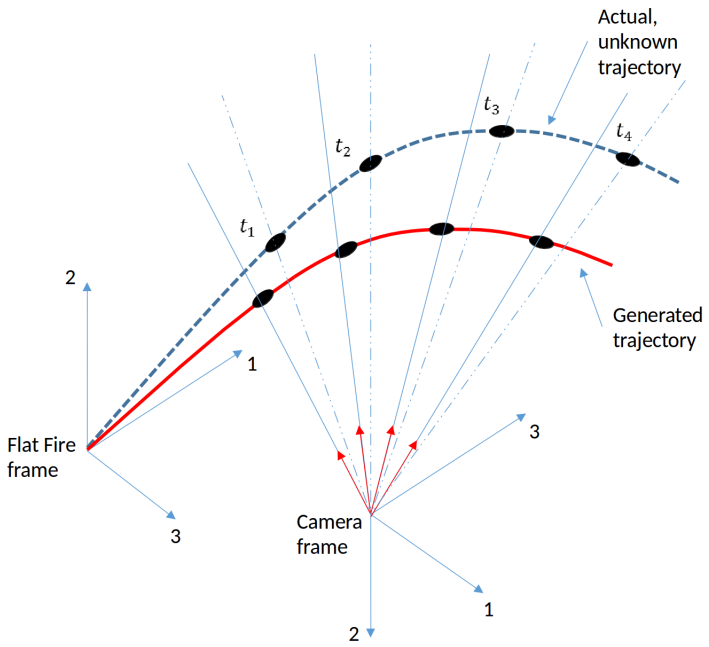

Figure 6.4: A trajectory is generated with a set of FFM parameters and projected into camera space using the known intrinsics of the camera, generating predicted pixel-time measurements that disagree with those observed.

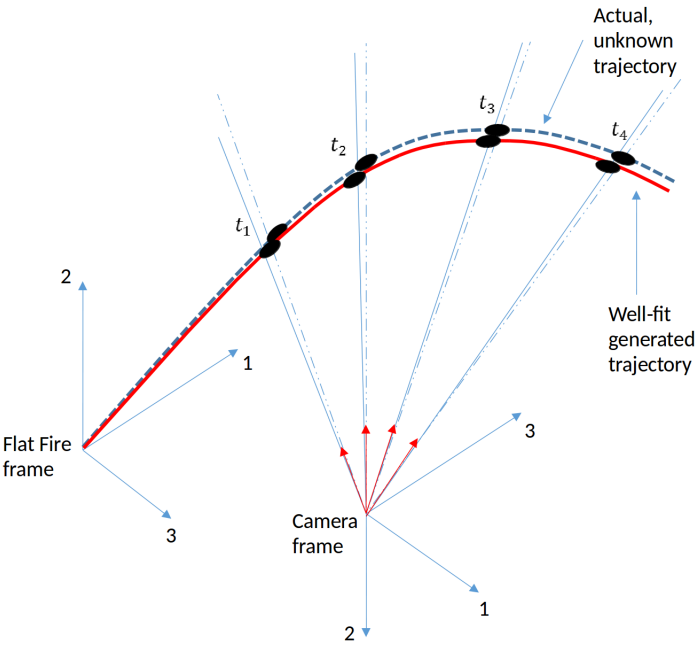

Figure 6.5: A trajectory is generated with a set of FFM parameters and projected into camera space using the known intrinsics of the camera, generating predicted pixel-time measurements that agree with those observed.

These figures show that we can evaluate how well a set of FFM parameters fits the camera observations of the bullet. The problem then becomes a matter of efficiently searching for the set of FFM parameters that best agrees with the observations. The optimization algorithm used to perform this search is covered in section 6.3.

Since we are using this procedure for self-correcting fire-control, many of the FFM parameters are known with fairly low uncertainty. For example, the bullet mass and diameter should be precisely known and don't need to be optimized. Alternatively, we may have good, but imprecise, estimates on the initial velocity and gun azimuth/elevation. These may need to be optimized, but the good initial estimates decrease the run time of the optimization algorithm.

One parameter with significant uncertainty is the firing time of the bullet. While an electronic trigger could provide a precise firing time, there is ambiguity between the firing of tracer rounds and non-tracer rounds. Assuming a typical machine gun firing 850 rounds per minute, with 1 in 5 rounds being a tracer, there will be a firing time uncertainty of 175 ms. If this were to remain unaccounted for, it would result in significant error in the estimated trajectory. For example, this would result in an impact error of 65 cm for a target at 300 m.

6.3 Methodology

The trajectory estimation procedure was tested with actual gunfire data. The objective of the test was to evaluate how precisely this procedure could estimate impact positions of tracer rounds based on observed pixel-time measurements. The geometry and instrumentation of the test is illustrated in figure 6.6.

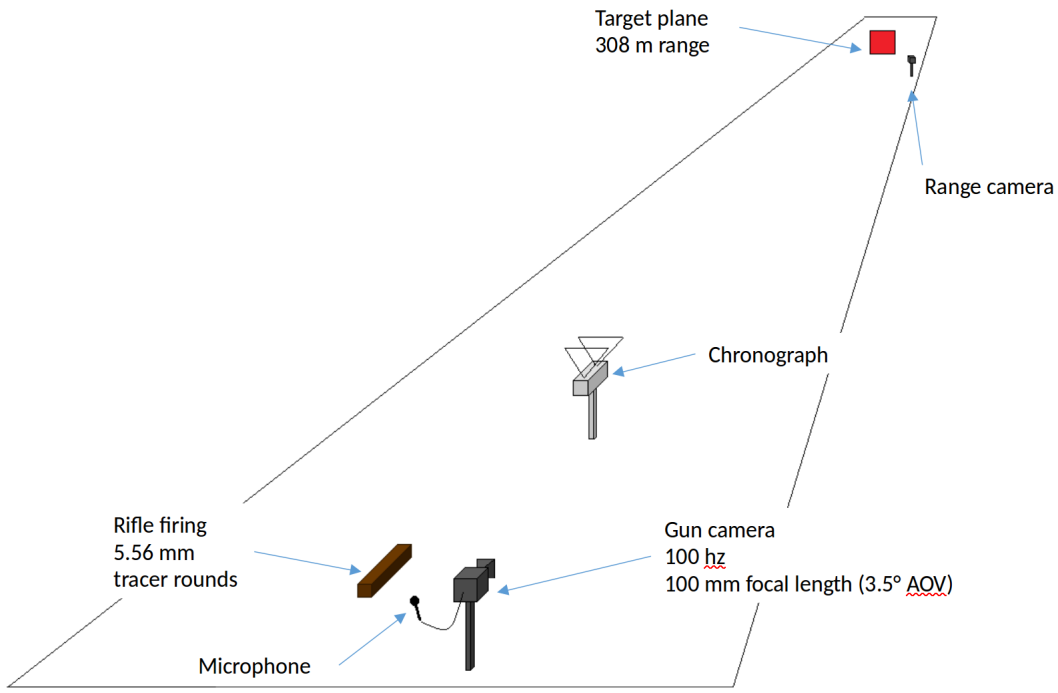

Figure 6.6: The trajectory estimation test set up.

To be representative of an RWS, the gun and camera were placed close together and well-aligned. The gun fired single-shot tracer rounds at a target 308 m downrange. A monochrome, visible light camera (the "gun camera" - Point Grey model FL3-U3-13Y3M-C) recorded the tracer rounds in flight at 100 Hz. For some shots, a microphone was used to trigger the camera from the muzzle blast, providing ground truth on the firing time. A chronograph was used to obtain the average muzzle velocity of the tracers.

The target was a flat sheet of cardboard with fluorescent panels glued on top to provide visual indications of bullet holes. An additional camera (the "range camera") was placed near the target to observe each impact, so the location of each impact could be corresponded with the tracers observed by the gun camera. A measured pattern on the target allowed for a homography to be computed that related pixels in the range camera to coordinates on the target. This allowed for each impact position to be accurately measured relative to the target.

The target was also tracked in the gun camera, as I would expect in an actual RWS engagement. The target pixel coordinate, combined with the known intrinsics of the camera and the target range, allowed for the target's three-dimensional position in camera coordinates to be known.

The dlib [22] implementation of the BOBYQA optimization algorithm [23] was used to estimate the FFM parameters. BOBYQA is a nonlinear, constrained optimization algorithm that does not require derivatives. The FFM parameters estimated for this test were as follows: weapon firing time, weapon azimuth and elevation, weapon vertical and horizontal offset relative to camera.

The post-processing procedure for each shot was as follows:

1. Extract pixel-time observations of the tracer in flight.
2. Estimate the FFM parameters that best agree with the observations.
3. Generate the FFM trajectory from the best-fit parameters, intersect it with the target location, and estimate an impact position relative to the target.

The estimated impact position could then be compared with the actual impact position derived from the range camera.

6.4 Experimental Results

This experiment was conducted with a total of 55 tracer round shots. An average of 27 pixel-time observations were made per shot. The resulting error between estimated impact positions and actual impact positions was 4.53 cm on average, with a 2.59 cm standard deviation.

A total of 8 shots included a recorded firing time via the microphone. This allowed the estimated firing time to be compared to ground truth. The resulting error was 6.0 ms on average, with a 4.5 ms standard deviation.

While an average of 27 observations of the tracer were available to estimate each trajectory, the estimation was also relatively accurate with significantly fewer observations. This was evaluated by taking particular subsets of the tracer observations (e.g. the first 5 observations) and reestimating the trajectory. The accuracies of various subsets is shown in table 6.1. This indicates that the trajectory estimation procedure has some robustness against lower frame rates and missing segments of the trajectory.

Table 6.1: Trajectory estimation mean (μ) and standard deviation (σ) with subsets of observations. All units in cm.

	Random 5	Random 10	First 5	First 10	Mid 5	Mid 10	Last 5	Last 10
μ	8.56	4.73	25.16	12.02	7.86	6.44	12.86	11.32
σ	3.77	2.75	20.30	12.15	2.98	10.98	5.25	3.01

Chapter 7: Outlier Removal and Multiple Model Extraction with RANSAC

7.1 Introduction

Chapter 6 demonstrated how we can estimate trajectories for a set of pixel-time observations of a bullet. The assumption made there was that the set of observations contained no outliers. This chapter shows that we can make the trajectory estimation procedure robust by extending it with the RANdom SAmple Consensus (RANSAC) algorithm.

This is important because three types of outliers are expected:

1. Non-bullet features: due to how simple bullets are in appearance, there will be similar features that get through the filters.

2. Multiple bullets: if the RWS is firing a typical machine gun, there will be multiple bullets in the field of view at the same time.

3. Ricocheted bullets: a ricocheted bullet can change direction dramatically, which would drastically alter the estimated trajectory.

These outliers are resolved by extending the previously developed trajectory estimation procedure with sequential RANSAC. Sequential RANSAC was chosen for its ease of use and relatively fast compute times.

Section 7.2 details my extension of the trajectory estimation procedure with sequential RANSAC. Section 7.3 describes how it will be tested against the three types of outliers detailed above, and section 7.4 details the results.

7.2 Theory

The trajectory estimation procedure is extended with the basic RANSAC algorithm as shown in Algorithm 1. This allows a single trajectory to be extracted among a set of pixel-time observations containing outliers. Note that if a set of FF params cannot be found that produces enough inliers (*mininliers*), then the algorithm will return null.

Algorithm 1 Trajectory RANSAC

```
1:  procedure TRANSAC
2:      numparams ← number of FF params to optimize
3:      n ← round(numparams ÷ 2)              ▷ each point provides two equations
4:      bestparams ← {}
5:      bestinliers ← {}
6:      iteration ← 0
7:      while iteration < max_iterations do
8:          points ← n randomly selected observations
9:          CS ← FF params fitted to points                    ▷ the candidate set
10:         inliers ← all observations that agree with CS      ▷ within threshold
11:         if |inliers| > mininliers and |inliers| > |bestinliers| then
12:             bestparams ← CS
13:             bestinliers ← inliers
14:         increment iterations
```

The Trajectory RANSAC algorithm is then extended with sequential RANSAC in Algorithm 2. This allows multiple trajectories to be extracted from a set of pixel-time observations. This algorithm repeats until no trajectories with at least *mininliers* can be extracted.

Algorithm 2 Sequential Trajectory RANSAC

```
    procedure SEQTRANSAC
2:      all_bestparams ← {{}}                  ▷ list of bestparams
        all_bestinliers ← {{}}                 ▷ list of bestinliers
4:      bestparams ← {}
        bestinliers ← {}
6:      iteration ← 0
        repeat
8:          run  TRANSAC
            if |bestinliers| > 0 then
10:             add bestinliers to all_bestinliers
                add bestparams to all_bestparams
12:             remove bestinliers from observation set
        until |bestinliers| == 0
```

7.3 Methodology

Trajectory estimation with RANSAC is first tested against simulated data. This data contains projected FF trajectories with the first two types of outliers listed in section 7.1.

The FF trajectories are generated by randomly selecting parameter values within fixed ranges. They are then projected into pixel-time coordinates based on the camera and geometry used in section 6.3. Gaussian noise is added to the pixel coordinates ($\sigma = 3px$) and time coordinate ($\sigma = 0.0005s$).

To simulate outliers caused by non-bullet features, a uniform random distribution of pixel-time coordinates is added, with the same size as the set of pixel-time coordinates formed from the actual trajectory. The domain of this distribution is the resolution of the camera and the timespan of the inlier set.

To simulate outliers caused by multiple bullets, two additional trajectories are projected to pixel-time coordinates, in subsequent time steps of 0.376 s. This is in order to simulate a machine gun burst firing 800 rounds per minute, with 1 in 5 bullets being a tracer round (and only detecting the tracers). I consider two main types of trajectory groupings. In the first type, all three trajectories are the same, just separated in time. This is to simulate a very tight burst group. In the second type, each trajectory is generated from randomly selected parameters.

The effectiveness of using RANSAC will be evaluated by comparing the known inliers for each bullet against those that were extracted. The false negative and positive rates are defined in equations 7.1 and 7.2, respectively, where A is the set of actual inliers and E is the set of extracted inliers.

$$FN = \frac{|A| - |A \cap E|}{|A|} \tag{7.1}$$

$$FP = \frac{|E| - |A \cap E|}{|E|} \tag{7.2}$$

The first simulation tests the Trajectory RANSAC algorithm for extracting the trajectory when the set of pixel-time observations contains only one trajectory and outliers. The second simulation tests the Sequential Trajectory RANSAC algorithm for extracting three trajectories when the set of pixel-time observations contains three trajectories and outliers.

Trajectory estimation with RANSAC is then tested against actual gunfire data containing ricocheted tracer rounds. This data contains two subsequent tracer rounds (separated by about 0.35 seconds), where both tracers dramatically change direction after striking the ground, but retain their brightness. One image from this data is shown in figure 7.1, which shows both tracers being clearly visible in the same frame, one of which has already struck the ground.

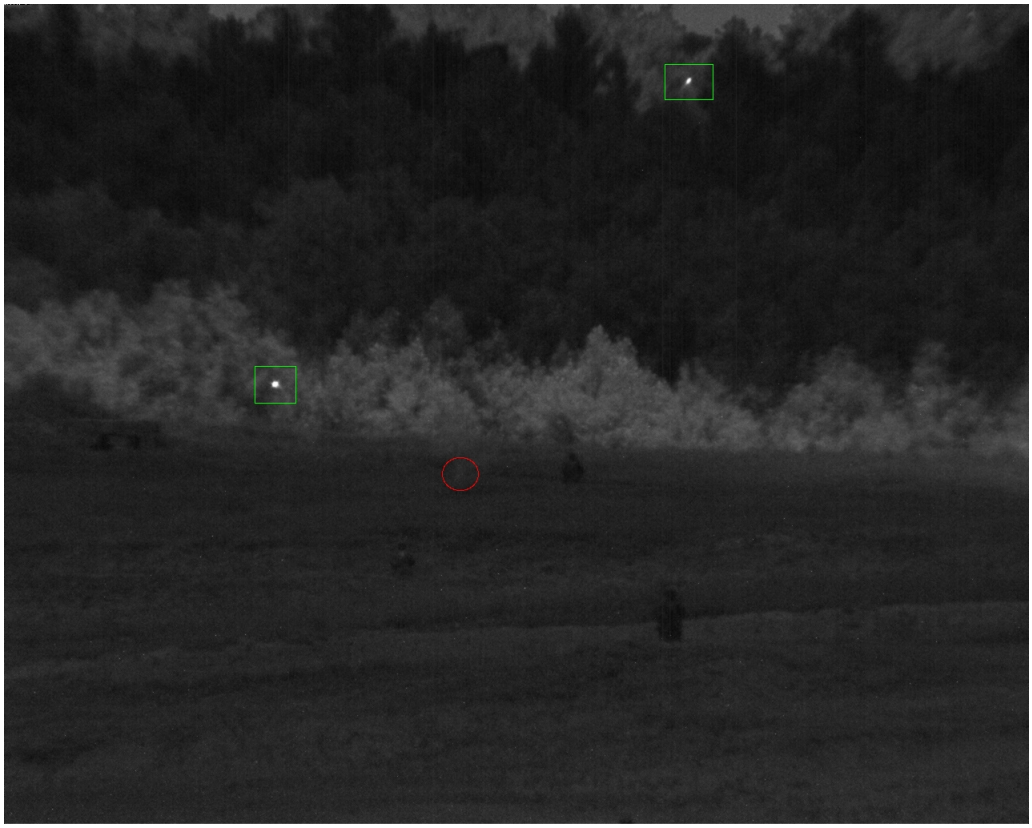

Figure 7.1: Two tracers visible in the same frame (highlighted with green rectangles). The tracer on the right side of the image has already struck the ground (at the location marked with the red circle), and quickly moved upward in image space. The second tracer later impacts the same area and ricochets in a similar direction.

7.4 Experimental Results

The first simulation experiment was evaluated 100 times with randomly generated trajectories. The Trajectory RANSAC algorithm was successful in extracting the vast majority of inliers in all cases. One example is shown in figures 7.2 and 7.3, which show the input set of observations (with outliers) and the extracted set of inliers, respectively. On average, the false positive rate was 0.001 and the false negative rate was 0.048.

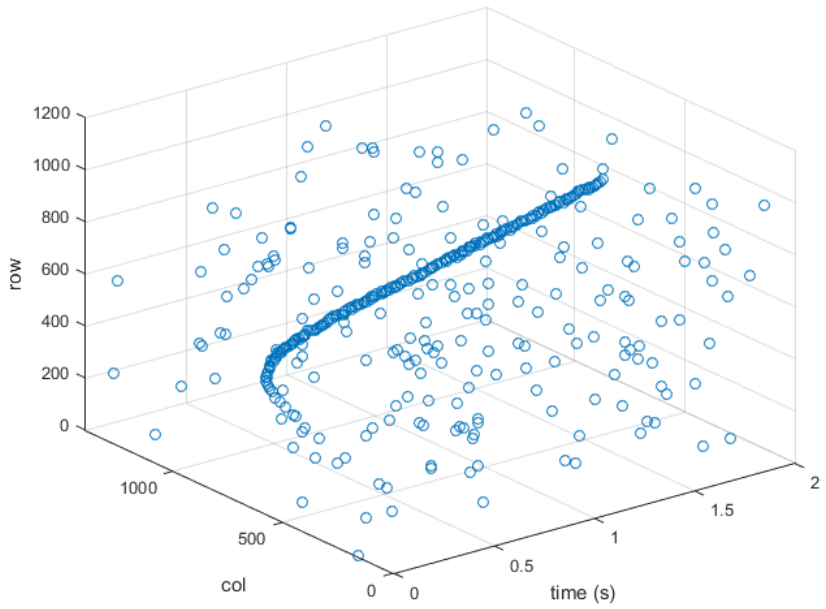

Figure 7.2: Input pixel-time observations containing one trajectory and outliers.

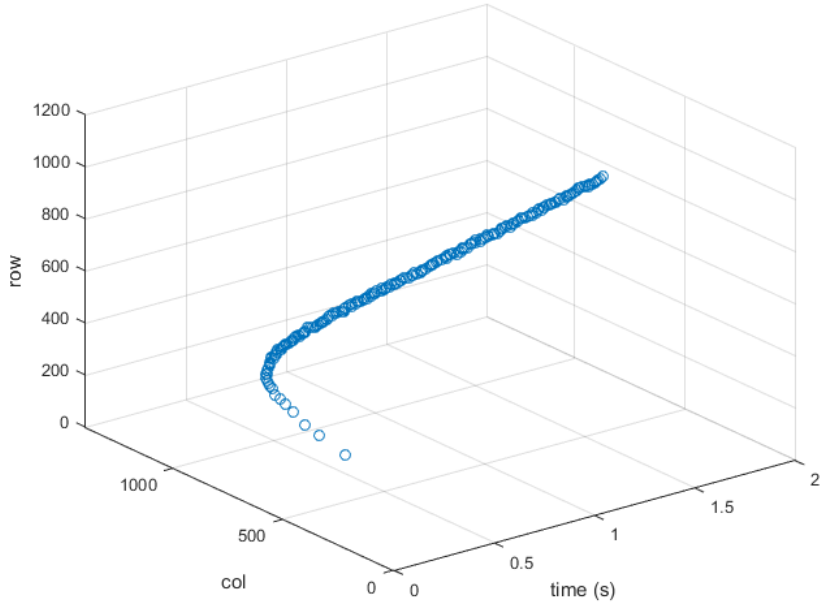

Figure 7.3: The extracted inliers using the Trajectory RANSAC algorithm.

The second simulation experiment was evaluated 100 times with both types of trajectory groupings. The results are shown in table 7.1. The Sequential Trajectory RANSAC algorithm was successful in extracting the vast majority of inliers in all cases. The extraction is especially accurate when all three trajectories are the same. Figures 7.4 and 7.5 show an example of three trajectories with the same ballistic parameters successfully extracted.

Table 7.1: The false negative and false positive rates for the pixel-time extractions of the three trajectories.

Group type	Trajectory 1		Trajectory 2		Trajectory 3	
	False Neg	False Pos	False Neg	False Pos	False Neg	False Pos
Same trajectory	0.010	0.001	0.013	0.001	0.009	0.001
Random trajectory	0.030	0.002	0.030	0.007	0.036	0.015

For the gunfire data with ricochets, pixel-time observations were first made using the algorithms developed in chapter 5. As expected, the tracers were detected after they ricocheted, however there were no random outliers. The set of pixel-time observations are shown in figure 7.6. The Sequential Trajectory RANSAC algorithm successfully detected the ricochet outliers and isolated the two trajectories. Figure 7.7 shows the extracted inliers.

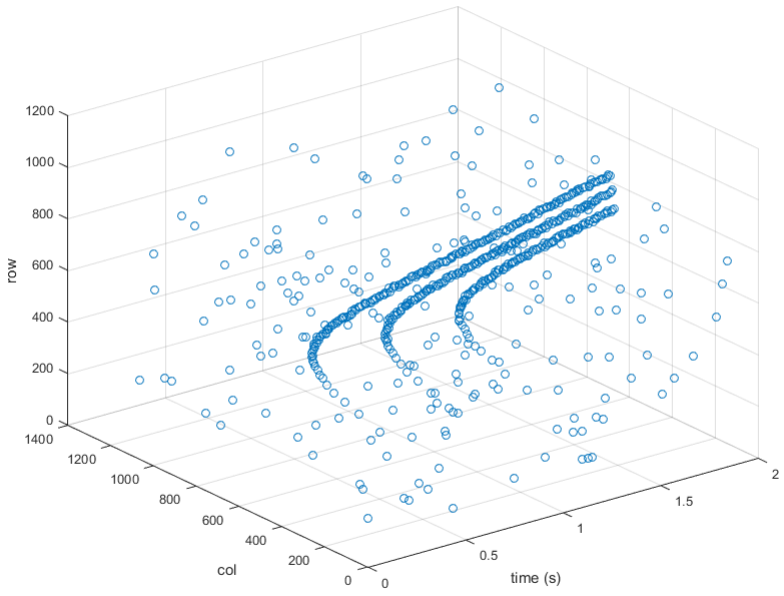

Figure 7.4: Input pixel-time observations containing three trajectories and outliers.

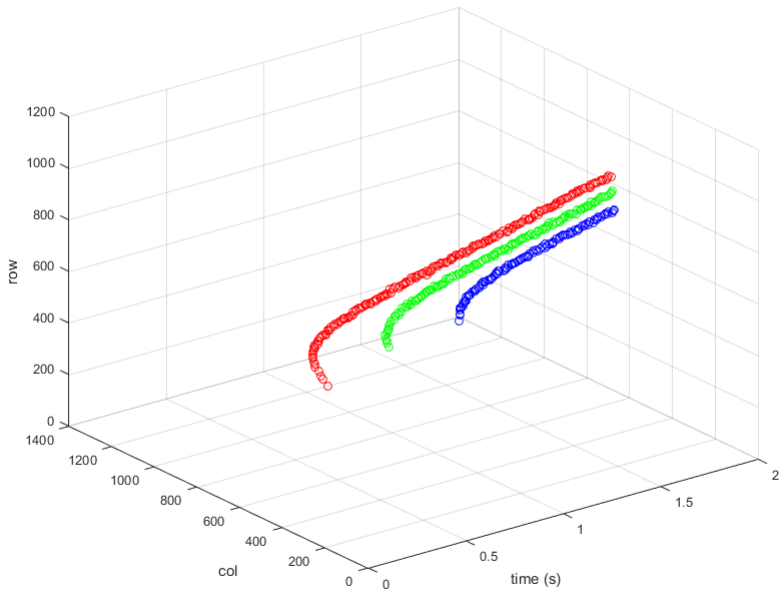

Figure 7.5: The extracted inliers using the Sequential Trajectory RANSAC algorithm, with each trajectory successfully isolated.

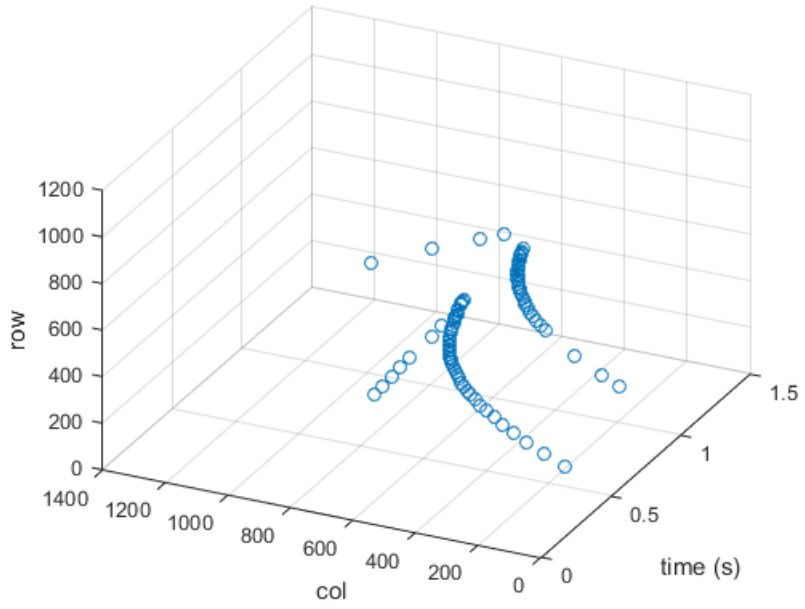

Figure 7.6: Pixel-time observations containing two trajectories that ricocheted after striking the ground.

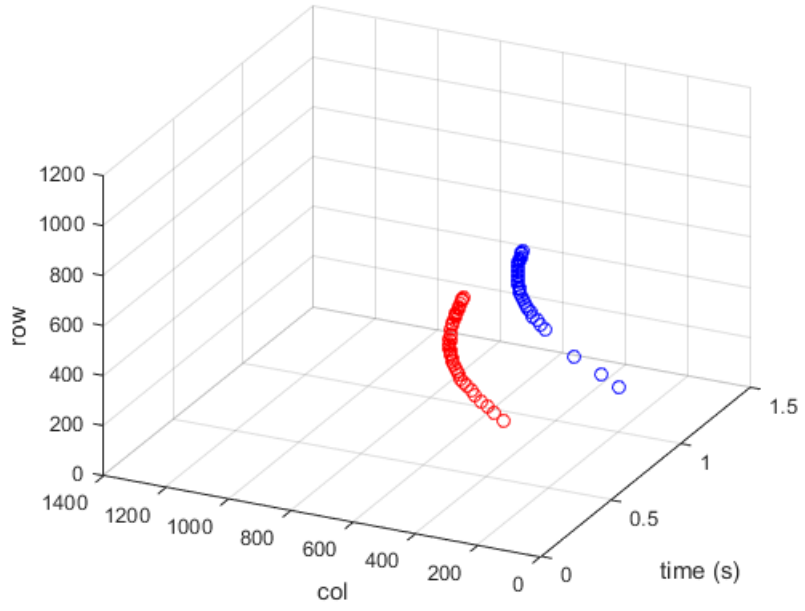

Figure 7.7: The extracted inliers using the Sequential Trajectory RANSAC algorithm, with ricochet outliers detected and discarded, and the two trajectories isolated.

Chapter 8: Conclusions

This thesis has demonstrated a proof of concept method of estimating trajectories based on camera observations of bullets in flight. Tracer rounds were successfully detected with the development of three image filters, and bullet impact points were precisely estimated using a parameter estimation technique based on the Flat Fire ballistic model. This technique was also made robust to outliers by extending it with the sequential RANSAC algorithm.

This work opens up the possibility of a closed-loop fire-control system using camera observations of bullets. However, there are still some important hurdles to overcome to make this a reality.

Firstly, these concepts were developed for a static camera that was vibrationally isolated from the gun. This must be extended to a camera mounted on an RWS, where it may be panning and vibrating. This motion would affect both bullet detection and trajectory estimation procedures.

Secondly, the algorithms developed in this research were tested by post-processing buffered data, where computation speed was not a factor. The bullet detection and trajectory estimation procedures easily take several seconds in their current form. In order to be useful in a self-correcting fire-control system, they must be made considerably faster.

While these issues are significant, I do not have reason to believe they are insurmountable. If they can be resolved, this approach to closed-loop fire-control would fill a significant void with small-caliber weapons.

Appendix A: Flat Fire Uncertainty Propagation

This appendix shows the derivation of the uncertainty propagation discussed in section 5.2. We first propagate uncertainty through the flat fire reference frame, then through camera coordinates, and finally into pixel coordinates.

Through Ballistic Model

$$\sigma_x = \left[\left(\frac{\partial x}{\partial t} \right)^2 {\sigma_t}^2 + \left(\frac{\partial x}{\partial V_{0_x}} \right)^2 {\sigma_{V_{0_x}}}^2 + \left(\frac{\partial x}{\partial x_0} \right)^2 {\sigma_{x_0}}^2 + \left(\frac{\partial x}{\partial k_3} \right)^2 {\sigma_{k_3}}^2 + \left(\frac{\partial x}{\partial d_x} \right)^2 {\sigma_{d_x}}^2 \right]^{\frac{1}{2}} \tag{A.1a}$$

$$\frac{\partial x}{\partial t} = \frac{4V_{0_x}}{k_3 t\sqrt{V_{0_x}} + 2} + d_x t \tag{A.1b}$$

$$\frac{\partial x}{\partial V_{0_x}} = \frac{t\left(k_3 t\sqrt{V_{0_x}} + 4\right)}{(k_3 t\sqrt{V_{0_x}} + 2)^2} \tag{A.1c}$$

$$\frac{\partial x}{\partial x_0} = 1 \tag{A.1d}$$

$$\frac{\partial x}{\partial k_3} = -\frac{2t^2 {V_{0_x}}^{3/2}}{\left(k_3 t\sqrt{V_{0_x}} + 2\right)^2} \tag{A.1e}$$

$$\frac{\partial x}{\partial d_x} = 0.5t^2 \tag{A.1f}$$

$$\sigma_{k_3} = \left[\left(\frac{\partial k_3}{\partial \rho} \right)^2 {\sigma_\rho}^2 + \left(\frac{\partial k_3}{\partial S} \right)^2 {\sigma_s}^2 + \left(\frac{\partial k_3}{\partial K_3} \right)^2 {\sigma_{K_3}}^2 + \left(\frac{\partial k_3}{\partial a} \right)^2 {\sigma_a}^2 + \left(\frac{\partial k_3}{\partial m} \right)^2 {\sigma_m}^2 \right]^{\frac{1}{2}} \tag{A.2a}$$

$$\frac{\partial k_3}{\partial \rho} = \frac{SK_3\sqrt{a}}{2m} \tag{A.2b}$$

$$\frac{\partial k_3}{\partial S} = \frac{\rho K_3\sqrt{a}}{2m} \tag{A.2c}$$

$$\frac{\partial k_3}{\partial K_3} = \frac{\rho S\sqrt{a}}{2m} \tag{A.2d}$$

$$\frac{\partial k_3}{\partial a} = \frac{\rho SK_3}{4m\sqrt{a}} \tag{A.2e}$$

$$\frac{\partial k_3}{\partial m} = -\frac{\rho SK_3\sqrt{a}}{2m^2} \tag{A.2f}$$

$$\sigma_y = \left[\left(\frac{\partial y}{\partial t} \right)^2 {\sigma_t}^2 + \left(\frac{\partial y}{\partial V_{0_x}} \right)^2 {\sigma_{V_{0_x}}}^2 + \left(\frac{\partial y}{\partial y_0} \right)^2 {\sigma_{y_0}}^2 + \left(\frac{\partial y}{\partial d_y} \right)^2 {\sigma_{d_y}}^2 + \left(\frac{\partial y}{\partial V_{0_y}} \right)^2 {\sigma_{V_{0_y}}}^2 + \right.$$

$$\left. \left(\frac{\partial y}{\partial g} \right)^2 {\sigma_g}^2 + \left(\frac{\partial y}{\partial V_x} \right)^2 {\sigma_{V_x}}^2 + \left(\frac{\partial y}{\partial x} \right)^2 {\sigma_x}^2 \right]^{\frac{1}{2}} \tag{A.3a}$$

$$\frac{\partial y}{\partial t} = -gt \left(\frac{2}{3} \sqrt{\frac{V_x}{V_{0_x}}} - \frac{1}{3} \right) + d_y t \tag{A.3b}$$

$$\frac{\partial y}{\partial V_{0_x}} = \frac{\frac{1}{6} g t^2 V_{0_x} \sqrt{\frac{V_x}{V_{0_x}}} - V_{0_y} x}{{V_{0_x}}^2} \tag{A.3c}$$

$$\frac{\partial y}{\partial y_0} = 1 \tag{A.3d}$$

$$\frac{\partial y}{\partial d_y} = 0.5 t^2 \tag{A.3e}$$

$$\frac{\partial y}{\partial V_{0_y}} = \frac{x}{V_{0_x}} \tag{A.3f}$$

$$\frac{\partial y}{\partial g} = -\frac{1}{6} t^2 \left(2 \sqrt{\frac{V_x}{V_{0_x}}} + 1 \right) \tag{A.3g}$$

$$\frac{\partial y}{\partial V_x} = -\frac{g t^2 \sqrt{\frac{V_x}{V_{0_x}}}}{6 V_x} \tag{A.3h}$$

$$\frac{\partial y}{\partial x} = \frac{V_{0_y}}{V_{0_x}} \tag{A.3i}$$

$$\sigma_{V_x} = \left[\left(\frac{\partial V_x}{\partial V_{0_x}} \right)^2 {\sigma_{V_{0_x}}}^2 + \left(\frac{\partial V_x}{\partial k_3} \right)^2 {\sigma_{k_3}}^2 + \left(\frac{\partial V_x}{\partial x} \right)^2 {\sigma_x}^2 + \left(\frac{\partial V_x}{\partial W_x} \right)^2 {\sigma_{W_x}}^2 \right]^{\frac{1}{2}} \tag{A.4a}$$

$$\frac{\partial V_x}{\partial V_{0_x}} = \frac{0.5 {k_3}^2 W_x x^2}{{V_{0_x}}^2} + \frac{k_3 x \left(-0.5 V_{0_x} - W_x\right)}{{V_{0_x}}^{3/2}} + 1 \tag{A.4b}$$

$$\frac{\partial V_x}{\partial k_3} = x \left(\frac{2W_x}{V_{0_x}} - 1 \right) \left(\sqrt{V_{0_x}} - 0.5 k_3 x \right) \tag{A.4c}$$

$$\frac{\partial V_x}{\partial x} = k \left(\frac{2W_x}{V_{0_x}} - 1 \right) \left(\sqrt{V_{0_x}} - 0.5 k_3 x \right) \tag{A.4d}$$

$$\frac{\partial V_x}{\partial W_x} = \frac{k_3 x \left(2\sqrt{V_{0_x}} - 0.5 k_3 x \right)}{V_{0_x}} \tag{A.4e}$$

$$\sigma_z = \left[\left(\frac{\partial z}{\partial t} \right)^2 {\sigma_t}^2 + \left(\frac{\partial z}{\partial V_{0_x}} \right)^2 {\sigma_{V_{0_x}}}^2 + \left(\frac{\partial z}{\partial z_0} \right)^2 {\sigma_{z_0}}^2 + \left(\frac{\partial z}{\partial d_z} \right)^2 {\sigma_{d_z}}^2 + \left(\frac{\partial z}{\partial x} \right)^2 {\sigma_x}^2 + \right.$$

$$\left. \left(\frac{\partial z}{\partial W_z} \right)^2 {\sigma_{W_z}}^2 \right]^{\frac{1}{2}} \tag{A.5a}$$

$$\frac{\partial z}{\partial t} = d_z t + W_z \tag{A.5b}$$

$$\frac{\partial z}{\partial V_{0_x}} = \frac{W_z x}{{V_{0_x}}^2} \tag{A.5c}$$

$$\frac{\partial z}{\partial z_0} = 1 \tag{A.5d}$$

$$\frac{\partial z}{\partial d_z} = 0.5 t^2 \tag{A.5e}$$

$$\frac{\partial z}{\partial x} = -\frac{W_z}{V_{0_x}} \tag{A.5f}$$

$$\frac{\partial z}{\partial W_z} = t - \frac{x}{V_{0_x}} \tag{A.5g}$$

Through Camera Coordinates

$$\sigma_{{}^{G}p_1} = \Bigg[\left(\frac{\partial\, {}^{C}p_1}{\partial\, {}^{C}_{G}R_{11}} \right)^2 \sigma^2_{{}^{C}_{G}R_{11}} + \left(\frac{\partial\, {}^{C}p_1}{\partial\, {}^{C}_{G}R_{12}} \right)^2 \sigma^2_{{}^{C}_{G}R_{12}} + \left(\frac{\partial\, {}^{C}p_1}{\partial\, {}^{C}_{G}R_{13}} \right)^2 \sigma^2_{{}^{C}_{G}R_{13}} +$$

$$\left(\frac{\partial\, {}^{C}p_1}{\partial\, {}^{G}p_1} \right)^2 \sigma^2_{{}^{G}p_1} + \left(\frac{\partial\, {}^{C}p_1}{\partial\, {}^{G}p_2} \right)^2 \sigma^2_{{}^{G}p_2} + \left(\frac{\partial\, {}^{C}p_1}{\partial\, {}^{G}p_3} \right)^2 \sigma^2_{{}^{G}p_3} + \left(\frac{\partial\, {}^{C}p_1}{\partial\, {}^{C}_{G}T_1} \right)^2 \sigma^2_{{}^{C}_{G}T_1} \Bigg]^{\frac{1}{2}}$$

$$= \Big[\left({}^{G}p_1 \right)^2 \sigma^2_{{}^{C}_{G}R_{11}} + \left({}^{G}p_2 \right)^2 \sigma^2_{{}^{C}_{G}R_{12}} + \left({}^{G}p_3 \right)^2 \sigma^2_{{}^{C}_{G}R_{13}} +$$

$$\left({}^{C}_{G}R_{11} \right)^2 \sigma^2_{{}^{G}p_1} + \left({}^{C}_{G}R_{12} \right)^2 \sigma^2_{{}^{G}p_2} + \left({}^{C}_{G}R_{13} \right)^2 \sigma^2_{{}^{G}p_3} + (1)^2 \sigma^2_{{}^{C}_{G}T_1} \Bigg]^{\frac{1}{2}} \tag{A.6a}$$

$$\sigma_{{}^{G}p_2} = \Bigg[\left(\frac{\partial\, {}^{C}p_2}{\partial\, {}^{C}_{G}R_{21}} \right)^2 \sigma^2_{{}^{C}_{G}R_{21}} + \left(\frac{\partial\, {}^{C}p_2}{\partial\, {}^{C}_{G}R_{22}} \right)^2 \sigma^2_{{}^{C}_{G}R_{22}} + \left(\frac{\partial\, {}^{C}p_2}{\partial\, {}^{C}_{G}R_{23}} \right)^2 \sigma^2_{{}^{C}_{G}R_{23}} +$$

$$\left(\frac{\partial\, {}^{C}p_2}{\partial\, {}^{G}p_1} \right)^2 \sigma^2_{{}^{G}p_1} + \left(\frac{\partial\, {}^{C}p_2}{\partial\, {}^{G}p_2} \right)^2 \sigma^2_{{}^{G}p_2} + \left(\frac{\partial\, {}^{C}p_2}{\partial\, {}^{G}p_3} \right)^2 \sigma^2_{{}^{G}p_3} + \left(\frac{\partial\, {}^{C}p_2}{\partial\, {}^{C}_{G}T_2} \right)^2 \sigma^2_{{}^{C}_{G}T_2} \Bigg]^{\frac{1}{2}}$$

$$= \Big[\left({}^{G}p_1 \right)^2 \sigma^2_{{}^{C}_{G}R_{21}} + \left({}^{G}p_2 \right)^2 \sigma^2_{{}^{C}_{G}R_{22}} + \left({}^{G}p_3 \right)^2 \sigma^2_{{}^{C}_{G}R_{23}} +$$

$$\left({}^{C}_{G}R_{21} \right)^2 \sigma^2_{{}^{G}p_1} + \left({}^{C}_{G}R_{22} \right)^2 \sigma^2_{{}^{G}p_2} + \left({}^{C}_{G}R_{23} \right)^2 \sigma^2_{{}^{G}p_3} + (1)^2 \sigma^2_{{}^{C}_{G}T_2} \Bigg]^{\frac{1}{2}} \tag{A.6b}$$

$$\sigma_{{}^{G}p_3} = \Bigg[\left(\frac{\partial\, {}^{C}p_3}{\partial\, {}^{C}_{G}R_{31}} \right)^2 \sigma^2_{{}^{C}_{G}R_{31}} + \left(\frac{\partial\, {}^{C}p_3}{\partial\, {}^{C}_{G}R_{32}} \right)^2 \sigma^2_{{}^{C}_{G}R_{32}} + \left(\frac{\partial\, {}^{C}p_3}{\partial\, {}^{C}_{G}R_{33}} \right)^2 \sigma^2_{{}^{C}_{G}R_{33}} +$$

$$\left(\frac{\partial\, {}^{C}p_3}{\partial\, {}^{G}p_1} \right)^2 \sigma^2_{{}^{G}p_1} + \left(\frac{\partial\, {}^{C}p_3}{\partial\, {}^{G}p_2} \right)^2 \sigma^2_{{}^{G}p_2} + \left(\frac{\partial\, {}^{C}p_3}{\partial\, {}^{G}p_3} \right)^2 \sigma^2_{{}^{G}p_3} + \left(\frac{\partial\, {}^{C}p_3}{\partial\, {}^{C}_{G}T_3} \right)^2 \sigma^2_{{}^{C}_{G}T_3} \Bigg]^{\frac{1}{2}}$$

$$\sigma_{{}^{G}p_1} = \Big[\left({}^{G}p_1 \right)^2 \sigma^2_{{}^{C}_{G}R_{31}} + \left({}^{G}p_2 \right)^2 \sigma^2_{{}^{C}_{G}R_{32}} + \left({}^{G}p_3 \right)^2 \sigma^2_{{}^{C}_{G}R_{33}} +$$

$$\left({}^{C}_{G}R_{31} \right)^2 \sigma^2_{{}^{G}p_1} + \left({}^{C}_{G}R_{32} \right)^2 \sigma^2_{{}^{G}p_2} + \left({}^{C}_{G}R_{33} \right)^2 \sigma^2_{{}^{G}p_3} + (1)^2 \sigma^2_{{}^{C}_{G}T_1} \Bigg]^{\frac{1}{2}} \tag{A.6c}$$

Through Pixel Coordinates

$$\sigma_c = \left[\left(\frac{\partial c}{\partial f} \right)^2 \sigma_f^2 + \left(\frac{\partial c}{\partial s} \right)^2 \sigma_s^2 + \left(\frac{\partial c}{\partial\ ^C p_3} \right)^2 \sigma_{^C p_3}^2 + \left(\frac{\partial c}{\partial\ ^C p_1} \right)^2 \sigma_{^C p_1}^2 + \left(\frac{\partial c}{\partial o_x} \right)^2 \sigma_{o_x}^2 \right]^{\frac{1}{2}} \tag{A.7a}$$

$$\frac{\partial c}{\partial f} = \frac{^C p_1}{s\ ^C p_3} \tag{A.7b}$$

$$\frac{\partial c}{\partial s} = \frac{f\ ^C p_1}{s^2\ ^C p_3} \tag{A.7c}$$

$$\frac{\partial c}{\partial\ ^C p_3} = -\frac{f\ ^C p_1}{s\ ^C p_3} \tag{A.7d}$$

$$\frac{\partial c}{\partial\ ^C p_1} = \frac{f}{s\ ^C p_3} \tag{A.7e}$$

$$\frac{\partial c}{\partial o_x} = 1 \tag{A.7f}$$

$$\sigma_r = \left[\left(\frac{\partial r}{\partial f} \right)^2 \sigma_f^2 + \left(\frac{\partial r}{\partial s} \right)^2 \sigma_s^2 + \left(\frac{\partial r}{\partial\ ^C p_3} \right)^2 \sigma_{^C p_3}^2 + \left(\frac{\partial r}{\partial\ ^C p_2} \right)^2 \sigma_{^C p_2}^2 + \left(\frac{\partial r}{\partial o_x} \right)^2 \sigma_{o_x}^2 \right]^{\frac{1}{2}} \tag{A.8a}$$

$$\frac{\partial r}{\partial f} = \frac{^C p_2}{s\ ^C p_3} \tag{A.8b}$$

$$\frac{\partial r}{\partial s} = \frac{f\ ^C p_2}{s^2\ ^C p_3} \tag{A.8c}$$

$$\frac{\partial r}{\partial\ ^C p_3} = -\frac{f\ ^C p_2}{s\ ^C p_3} \tag{A.8d}$$

$$\frac{\partial r}{\partial\ ^C p_2} = \frac{f}{s\ ^C p_3} \tag{A.8e}$$

$$\frac{\partial r}{\partial o_y} = 1 \tag{A.8f}$$

Bibliography

[1] W. Commons. (2015) Cappadocia gaussian blur. Accessed: 2017-08-02. [Online]. Available: https://commons.wikimedia.org/wiki/File%Cappadocia_Gaussian_Blur.svg

[2] D. Serakos, "Phalanx cwis control system stability, aim bias compensation, and noise sensitivity," Naval Surface Warfare Center, Dahlgren Division, Tech. Rep. 92/243, 1992.

[3] C. Weiland and K. Murray, "An analysis of the visual automated scoring system performance during live gunfire events," in *Proceedings of 2016 Armament Systems Forum*. DTIC, April 2016.

[4] D. Simons. Supervised autonomous fires technology (saf-t). Accessed: 2017-06-17. [Online]. Available: https://www.onr.navy.mil/en/Media-Center/Fact-Sheets/Supervised-Autonomous-Fires

[5] R. L. McCoy, *Modern Exterior Ballistics*. Schiffer Publishing, Ltd., 2012.

[6] D. E. Carlucci and S. S. Jacobson, *Ballistics: Theory and Design of Guns and Ammunition*. CRC Press, 2013.

[7] R. F. Lieske and M. L. Reiter, "Equations of motion for a modified point mass trajectory," U.S. Army, Tech. Rep. 1314, 1966.

[8] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov 2000.

[9] X. Huang, J. Gao, and R. Yang, *Calibrating Pan-Tilt Cameras with Telephoto Lenses*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 127–137. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-76386-4_11

[10] A. Klaus, J. Bauer, K. Karner, P. Elbischger, R. Perko, and H. Bischof, "Camera calibration from a single night sky image," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, June 2004, pp. I–151–I–157 Vol.1.

[11] D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis, "Astrometry.net: Blind astrometric calibration of arbitrary astronomical images," *Astronomical Journal*, vol. 137, pp. 1782–2800, 2010, arXiv:0910.2233.

[12] H. Li and Z. Lei, "Calculation research on infrared radiation characteristic on flying projectile in sky-screen," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1959–1964, May 2013.

[13] S. Snarski, K. Scheibner, S. Shaw, R. Roberts, A. LaRow, E. Breitfeller, J. Lupo, D. Nielson, B. Judge, and J. Forren, "Autonomous UAV-based mapping of large-scale urban firefights," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 6209, May 2006, p. 620905.

[14] A. Richards. Infrared camera measures bullet heating. Accessed: 2017-07-20. [Online]. Available: http://www.advancedimagingpro.com/print/Advanced-Imaging-Magazine/Infrared-Camera-Measures-Bullet-Heating/1$180

[15] V. Condaminet, F. Delvare, D. Cho, H. Ene Demailly, C. Grignon, and S. Heddadj, "Identification of aerodynamic coefficients of a projectile and reconstruction of its trajectory from partial flight data," vol. 21, pp. 177–186, 01 2014.

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[17] O. Chum and J. Matas, "Randomized ransac with td, d test," in *Proc. British Machine Vision Conference*, vol. 2, 2002, pp. 448–457.

[18] E. Vincent and R. Laganiére, "Detecting planar homographies in an image pair," in *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on.* IEEE, 2001, pp. 182–187.

[19] Y. Kanazawa and H. Kawakami, "Detection of planar regions with uncalibrated stereo using distributions of feature points." in *BMVC*, 2004, pp. 1–10.

[20] M. Zuliani, C. S. Kenney, and B. Manjunath, "The multiransac algorithm and its application to detect planar homographies," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 3. IEEE, 2005, pp. III–153.

[21] J.-Y. Bouguet. Camera calibration toolbox for matlab: Description of the calibration parameters. Accessed: 2017-06-25. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html

[22] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[23] M. Powell, "The bobyqa algorithm for bound constrained optimization without derivatives," Department of Applied Mathematics and Theoretical Physics, Cambridge University, Tech. Rep. DAMTP 2009/NA06.

Biography

Kevin Murray was born in 1984 in Fairfax, Virginia. He graduated from James W. Robinson high school in 2002.

He attended Virginia Tech from 2003 to 2008, graduating with a B.S. in Mechanical Engineering. While at Virginia Tech, he worked on an autonomous helicopter project at the Unmanned Systems Lab for his senior design project.

From 2008 to 2011, Kevin worked as a mechanical engineer at E.K. Fox & Associates. His responsibilities included designing commercial and government HVAC systems and conducting full-building energy models.

In 2011, Kevin returned to Virgina Tech as a masters student. He worked in the Mechatronics Lab for Dr. Al Wicks. His thesis was focused on a camera-based positioning system for autonomous agriculture vehicles. He graduated with a M.S. in Mechanical Engineering in 2012.

In 2013, Kevin was admitted to George Mason University as a masters student in Computer Science. His focus is on robotics and computer vision.