Nonlinear Blood Pattern Reconstruction

by

Benjamin Thomas Cecchetto

B.Sc., The University of Toronto, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

October 2010

Abstract

We present a method of reconstructing the area of origin for blood droplets given the position and directions of impact stains. This method works for nonlinear trajectories, such as parabolic motion or motion with drag.

A model for fitting ellipses to the stains, obtaining impact velocities, blood drop mass and drag coefficient from blood splatter image densities, impact angle and pattern is also provided. We also show how to use this extra data to aid with our estimation.

We validate our method in several experiments involving blood splatters at varying velocities and angles.

Table of Contents

List of Tables

List of Figures

List of Symbols

Acknowledgements

I would like to acknowledge my supervisor Wolfgang Heidrich as well as Lukas Ahrenberg, Bradley Atcheson and Nasa Rouf for their thoughts and support over the span of this and previous projects.

Dedication

I would like to dedicate this thesis to my parents Thomas and Judith Cecchetto for their continued love and support over the years.

I would also like to dedicate this thesis to Kristina Mayor for introducing me to the television show *Dexter*. If it wasn't for that introduction, this project wouldn't exist.

Chapter 1

Introduction

1.1 Motivation

Forensic science, or forensics, is the practice of using science to gather information about a crime, or any other event that needs to be considered as evidence in a legal system. This puzzle often needs to be pieced together using clues from various sciences such as physics, biology, and chemistry to extrapolate to events that happened in the past. The information gathered must be as objective as possible as a safeguard against erroneous deductions.

One subset of forensics is bloodstain pattern analysis (BPA), i.e. gathering information from bloodstains on various surfaces to determine more about events that unfurled at the crime scene. The blood originates from a wound of sorts, usually involving a force or trauma to a victim, and ends up on a surface creating a pattern which can be photographed afterwards. There are many ways the blood may arrive at its destination. It may impact the surface after being in freefall after exiting a wound, it may be smeared from some other object. Sometimes the blood can even leave from one pool or stain to a new one. There are many different cases of patterns, each with a different meaning.

One common pattern is blood particles ejected from a wound. The only

1

forces acting upon a blood particle are assumed to be gravity and drag. The shape of the stains are projected spheres onto the recipient surface, which are ellipses on planar surfaces.

(a) A blood spatter event originating from the area of origin

(b) Given the impact angle and positions, we need to find the area of origin
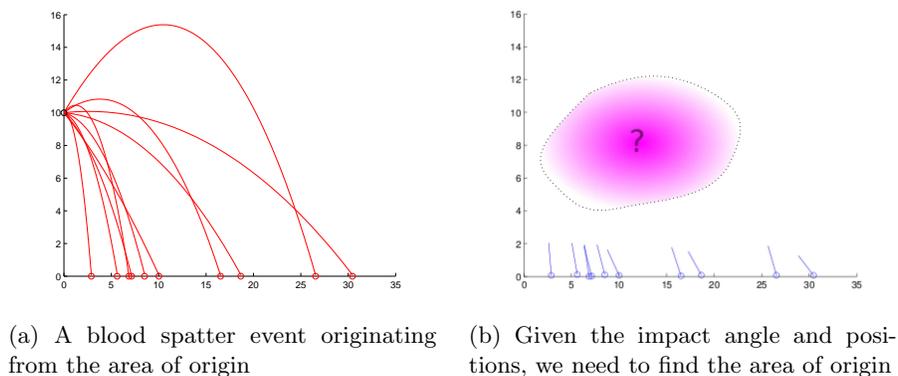
Figure 1.1: *Problem statement*

In Figure 1.1, on the left we see a typical blood spatter event. Blood particles travel through the air until reaching their destination at a recipient surface. We can only observe the impact positions and angles after the event has occured, thus leaving the original emitting position at the time of the event unknown. We aim to calculate this position.

Traditional techniques for this method involve identifying the angle of impact and extending that angle into 3D through various methods, assuming a linear flight path, using physical strings or software. The area of origin is assumed to lie at the intersection of the flight paths. For this method to work, all particles must have a high enough velocity for the trajectories to approximate lines. For high velocity events such as gun spatter, this works well, but for medium and low velocity events droplets do not generally travel

in straight lines and are influenced by the force of gravity and drag. Since the droplets are always accelerating downwards, the tangent of the flight path at the point of impact curve will be steeper than at the point of origin. In fact, all of the tangents at the points of impact will be steeper than at the area of origin and so it is known that the true area of origin will be lower than the intersection of these tangent lines. However, where exactly is not certain with the linear approach.

For such a calculation based method, it seems beneficial to automate it given input data, to more accurately reconstruct the area of origin. We will discuss how existing methods can be improved upon as they do not take into account gravity or drag and assume particles travel along straight lines. We will use computer vision techniques to infer the impact pattern and a probabalistic formulation to describe a constrained system of equations that may be solved for. We will also discuss different types of information that can be obtained from a blood droplet and how that can be used to bound the problem statement.

1.2 Scope

Correctly deducing what kind of event causes a given blood pattern, usually requires years of experience. This thesis does not seek to automate such an approach. However, we are trying to facilitate more automated procedures supporting experts in their analysis.

There are many patterns that seem intractable to solve, as blood could have come from any source in the crime scene. We examine one such case

of pattern, with the goal of accurately estimating the area of origin. Ths is also known as reconstruction, or directional analysis. This area is where the emitting wound was at the moment of the event, and leaves behind a very specific type of pattern. The droplets are assumed to fall under the force of drag and gravity. We also assume that the blood stain is on an untextured, planar surface.

1.3 Overview of the Thesis

In Chapter 2 we discuss various previous approaches to the problem of reconstruction. We also discuss their limitations, and cases where they do and do not perform well. Various related work beneficial to solving the task is also presented.

Chapter 3 describes the main algorithm to reconstruct an area of origin. We assume we know the impact positions and directions as input. We derive the trajectories from Newton's Laws and proceed to a probabalistic derivation, solving for a probability density function (PDF), or where a particle may in space be at a given time. From there we show how to construct this PDF for a whole trajectory, then use it for multiple trajectories to bound the search space with and without apriori knowledge of velocities. We also discuss an error bound on the estimate and in what situations it performs well or performs poorly.

In order to perform the reconstruction, we need to obtain stain positions and other parameters. We discuss various ways of doing so in Chapter 4. We begin by describing segmentation and ellipse fitting to obtain the impact

angle as done by traditional blood pattern analysts. We go on to describing how to estimate the velocity, drag and mass parameters of a given droplet to aid with bounding the reconstruction space. Some results may need to be pruned or modified as they are noisy. We discuss how to prune our results using RANSAC and flipping directions on ellipses (since they are symmetric entities) as they may be ambiguous.

Validation is performed for various claims in Chapter 5. A simple blood droplet experiment validates some claims stated in Chapter 4, and various real life scenarios of spatter events are also performed to show where our reconstruction algorithm from Chapter 3 performs well and where it fails.

We finish by discussing limitations of the algorithms presented here in Chapter 6. We also talk about where the works presented here may be added onto, or future work with this project.

Chapter 2

Literature Review

In this chapter, we will discuss all various related work either in traditional methods, existing automatic methods, or computer graphics and vision works related to ours. Existing methods for manual or traditional blood pattern analysis (BPA) are discussed in Section 2.1. Automatic BPA techniques will be discussed in Section 2.2. This includes the assumptions they make and what scenarios their algorithms perform well or terribly on.

2.1   Traditional Blood Pattern Analysis

A good resource which serves to introduce BPA is Blood Dynamics by A. Wonder [18]. This book describes various logical processes and how to deduce accurate facts from a crime scene, as well as the basic linear reconstruction techniques. A newer book on the matter [1] covers more modern techniques as well as the traditional ones. For example, it covers virtual stringing which we will discuss later. Good supplements to this work are [8] and [19], which focus more on case studies from various scenes. This shows a large number of various crime scenes and what type of blood patterns to expect in the real world.

Various BPA tools are presented in [1] using various flowchart diagrams

to deduce the order of events. Also presented are techniques for BPA reconstruction such as the stringing method, otherwise known as the tangent method. The tangent method utilizes the fact that the blood particles project onto the surface forming an elliptical stain upon impact, followed by secondary shapes. In order to perform this method, strings are run along the major axis of each particle's impact ellipse. These lines converge if the scene obeys the assumptions above. Let us say they converge at a point on the ground plane (the xy plane), P_{xy}. Obtaining the direction outside the plane can be determined by comparing the ellipse major and minor axis length (L and W respectively), to obtain the impact angle θ

$$\sin(\theta) = \frac{W}{L}. \tag{2.1}$$

Geometrically, this can be seen in Figure 2.1. In Figure 2.1a, a sphere is projected onto a plane with angle θ. This leaves an ellipse with minor axis width W and major axis length L as seen more easily in Figure 2.1b. The extended portion of the bloodstain is called a satellite stain which we will discuss in Chapter 4.

A tripod, or some other mounting structure is then placed at the point on the plane P_{xy}. Knowing the angle for each droplet, we connect a string from each droplet to the tripod so that the angle made by this string and the plane is the impact angle θ. The area of origin is an average of all these points connected to the tripod, which obtains the height of the area of origin z. Using the planar point P_{xy} with the height z as the third component we form P_{xyz}. We see this in Figure 2.2. The dotted black lines correspond to the

(a) Planar projection of a sphere   (b) Parameters of an ellipse
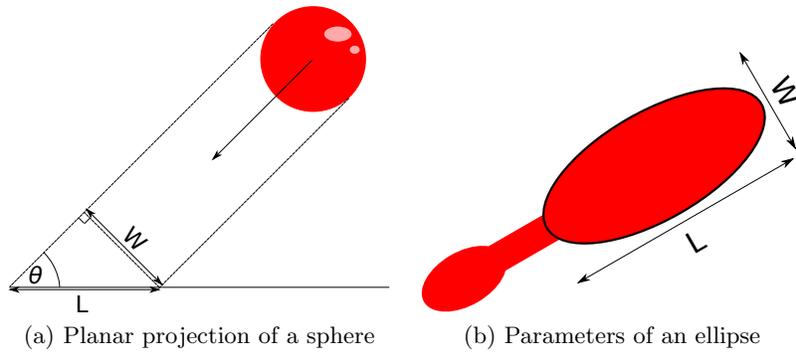
Figure 2.1: *Obtaining a particle's direction from a stain.*

strings on the plane and the white lines correspond to the strings attached to the tripod, which represent the linear flight paths for each droplet.
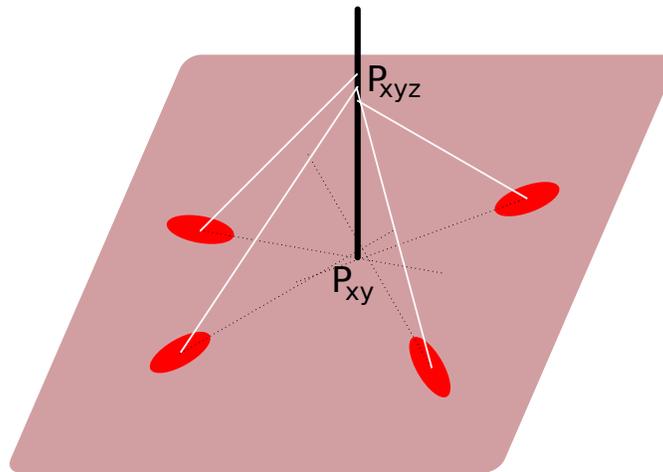
Figure 2.2: *The stringing method*

There have been other studies related to traditional BPA, such as [15] where the accuracy of the linear method was studied with varying hematocrit values in the blood. Hematocrit is the ratio of white to red blood cells, changing the dynamic properties and colour of the blood. It was shown that

there was very little change in the error estimates while varying these values.

2.2 Automatic Blood Pattern Analysis

There exists commercial software to accomplish virtual stringing. The most common program in use is BackTrack [4]. A user inputs blood stain ellipses, absolute angles of the ellipses relative to the wall, and positions in 3D space. The program facilitates this process by letting the user interactively move an ellipse to fit selected stains. The program then computes a linear estimate of where the area of origin is. This method is akin to the stringing method described above, but performed on a computer and potentially more accurately. However, the software is limited in that it only works for axis aligned walls, ceiling and floor. Also, the user still has to enter a large amount of data for input.

Evaluations of this program have been done, comparing it to the stringing method [5] and concluding that it is a reasonably accurate method for most results. For specific results where gravity is a key factor in the trajectories, a different evaluation [12] concludes that even though the software can calculate the horizontal plane coordinates accurately, the height coordinate has to be approached with caution as an estimate is only accurate as an upper bound.

Some attempts to describe a full system for BPA reconstructions have also been performed. Shen et al. [16] describe an outline of various computer vision techniques as applied to BPA. They attempt basic ellipse fitting to masks, pruning ellipses with significant deviations in direction. They also

explore rectifying images using a checkerboard calibration grid. They claim to obtain the area of origin, though no error results are provided for the fully reconstructed result.

There is also a more recent paper [2] describing an ellipse fitting technique specifically designed for blood droplet analysis. This algorithm identifies the blood based on statistics about blood colour, and segments it from a white background, approximating it with an ellipse. It then inscribes the ellipse in a rectangle to determine the major axis of the ellipse. They obtain approximately 10% error in direction, using tests on 30 ellipses. Another automatic approach fits homographies from coplanar ellipses from one image to another to infer information about the scene [20]. Although useful for mapping images from one to another, their reconstructed results as applied to BPA are far from accurate. Results from 2 experiments were shown with errors of 35 and 67 inches. Which is a significant error, as it may place the estimated area of origin a few feet away from the actual event. For context, in [5], they state 10-20cm (or 4-8 inches) is close enough to the true blood source location to allow an accurate interpreteation of the crime.

There have been portable technologies to scan a crime scene more accurately as well, such as the DeltaSphere 3000 [7]. The DeltaSphere obtains a dense model of the scene using a laser range scanner. It also captures colour at the sampled points. There has also been a case study investigating the usefulness of virtual crime scene environments [14]. It concludes that creation of an interactive virtual environment of a crime scene to be very useful and of significant importance to data analysis, witness statement evaluation, route visualisation, officer briefing, hypothesis evaluation, training

10

and security planning.

Chapter 3

Area of Origin Reconstruction

In this chapter we will derive a probabalistic formulation for trajectory reconstruction. We will begin by deriving what trajectory or path a particle may have from Newton's laws in Section 3.1. We do so first assuming no drag, then show a derivation with drag taking a similar approach. In Section 3.2, we discuss how knowing an analytical solution for the trajectories can help us know the path it took to get to that point. Section 3.3 we discuss how to create probability density functions (PDFs) for various trajectories and scenarios, showing where they are likely to be prior to impact. In Section 3.4 we describe how to obtain the 3D PDF from the 2D formulations and maximum likelihood estimate for the area of origin in 3D. Finally, in Section 3.5 we will discuss an error bound in order to ensure knowledge of the accuracy of the reconstruction results for a given scenario.

3.1 3D Projectile Motion

If we are to understand how to reconstruct a particle's trajectory based on the resultant impact site, we begin with a mathematical model of the

trajectory based on particular assumptions.

3.1.1   Newtonian Derivation for a Parabolic Trajectory

Projectile motion is one of the most basic type of physical motion where objects fall only under the effects of gravity. We assume here that there is no air resistance. Newton's laws state that the net force $\mathbf{F}_{NET}$ on an object is related to the mass m and acceleration $\mathbf{a}$ by the equation

$$\mathbf{F}_{NET} = m\mathbf{a}, \tag{3.1}$$

where $\mathbf{F}_{NET}$ is the sum of all forces on an object. If we draw a free body diagram for the object of interest, we can identify these forces visually.
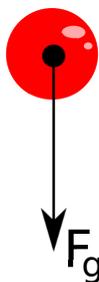
Figure 3.1: *A free body diagram of the forces acting upon a particle*

The only force is that of gravity, denoted $\mathbf{F}_g$. This force can be described by the equation

$$\mathbf{F}_g = m\mathbf{g}, \tag{3.2}$$

where $\mathbf{g} = (0, 0, -g)$ is the 3D acceleration vector of gravity with $g = 9.81 m/s^2$. For particles only being affected by gravity the spatial motion is

parabolic.

Integrating $\mathbf{a} = \mathbf{g}$ with respect to time t, since gravity is the only force we arrive at

$$\mathbf{v}(t) = \mathbf{g}t + \mathbf{v}_0, \tag{3.3}$$

where $\mathbf{v}_0$ is the initial velocity. Integrating again and introducing initial position $\mathbf{x}_0$, we obtain the analytical description of trajectory

$$\boxed{\mathbf{x}(t) = \frac{1}{2}\mathbf{g}t^2 + \mathbf{v}_0 t + \mathbf{x}_0.} \tag{3.4}$$

From this equation we can see that the trajectory is parabolic in motion. This is illustrated in Figure 3.2. This equation holds for all particles falling under gravity, they all fall with the same acceleration.
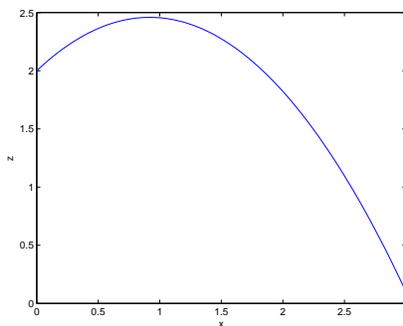
Figure 3.2: *A parabolic trajectory*

3.1.2 Newtonian Derivation for Trajectory with Drag Force

Thus far for simplicity we have overlooked a key force, the drag force or air resistance. It is denoted as $\mathbf{F}_d$. Incorporating it into the derivation yields

a much more complicated expression where the resulting trajectory starts parabolic in nature but turns linear as the particle approaches its terminal velocity, as we will show.

Let us begin by drawing the free body diagram of the particle, this time taking into account the force of air resistance.
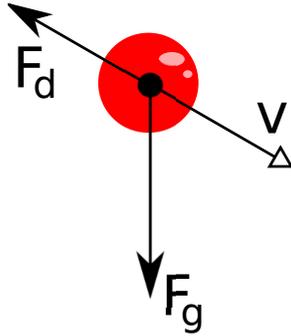
Figure 3.3: *A free body diagram of a particle falling under the influence of gravity, with drag force*

The drag force, and thus terminal velocity, vary for different particles in the scene depending on their size and shape. Stokes' Law applies for spherical objects in a fluid with low Reynold's numbers, and we will assume our objects are spherical in nature. This law states the drag force as

$$\|\mathbf{F}_d\| = 6\pi\mu r\|\mathbf{v}\|, \tag{3.5}$$

where μ is the dynamic viscosity of the fluid, which for air is $1.78 \times 10^{-5}\frac{kg}{m \cdot s}$, and $\mathbf{v}$ is the particle velocity. The law also states that this force is in the opposite direction of the velocity, so we can simplify the drag force as

$$\mathbf{F}_d = -6\pi\mu r\mathbf{v}. \tag{3.6}$$

If we combine all the constants into one term k, we can write

$$\mathbf{F}_d = -k\mathbf{v}, \tag{3.7}$$

where

$$k = 6\pi\mu r. \tag{3.8}$$

Now that we know how both forces $\mathbf{F}_d$ and $\mathbf{F}_g$ can be described as, from Newton's law in Equation 3.1,

$$\mathbf{F}_g + \mathbf{F}_d = m\mathbf{a} \tag{3.9}$$

$$\Rightarrow \qquad m\mathbf{g} - k\mathbf{v} = m\mathbf{a} \qquad . \tag{3.10}$$

We wish to integrate this equation, so we will separate the appropriate parts using a substitution $\mathbf{a} = \frac{d\mathbf{v}}{dt}$. Also, for simplicity, we will drop the vector notation and assume we are only working with one component of the vector at a time, e.g. g_x instead of $\mathbf{g}$. We can do this since x and y are independent as we are only dealing with linear or component-wise operators.

$$mg_x - kv_x = m\frac{dv_x}{dt}, \tag{3.11}$$

$$\Rightarrow \qquad dt = m\frac{dv_x}{mg_x - kv_x}, \tag{3.12}$$

which is now in a form to integrate and simplify to obtain the velocity,

$$\int dt = \int m \frac{dv_x}{mg_x - kv_x} \tag{3.13}$$

$$\Rightarrow \qquad t + C_1 = -\frac{m}{k} \ln|mg_x - kv_x| \tag{3.14}$$

$$\Rightarrow \qquad e^{t+C_1} = (mg_x - kv_x)^{-\frac{m}{k}} \tag{3.15}$$

$$\Rightarrow \qquad (e^{t+C_1})^{-\frac{k}{m}} = mg_x - kv_x \tag{3.16}$$

$$\Rightarrow \qquad kv_x = mg_x - e^{(t+C_1)(-\frac{k}{m})} \tag{3.17}$$

$$\Rightarrow \qquad v_x = \frac{1}{k}(mg_x - C_3 e^{-\frac{kt}{m}}). \tag{3.18}$$

If we do this for each component, we obtain the equation

$$\mathbf{v} = \frac{1}{k}(m\mathbf{g} - \mathbf{C}_3 e^{-\frac{kt}{m}}), \tag{3.19}$$

where the constant C_3 is different for each component. With initial conditions $\mathbf{v}(t) = \mathbf{v}_0$ for $t = 0$, we can write

$$\mathbf{v}(0) = \mathbf{v}_0 = \frac{1}{k}(m\mathbf{g} + \mathbf{C}_3) \tag{3.20}$$

$$\Rightarrow \qquad \mathbf{C}_3 = \mathbf{v}_0 k - m\mathbf{g}. \tag{3.21}$$

Substituting back in, we obtain the velocity equation,

$$\mathbf{v}(t) = \frac{m\mathbf{g}}{k} + (\mathbf{v}_0 - \frac{m\mathbf{g}}{k})e^{-\frac{kt}{m}}. \tag{3.22}$$

Analyzing this, we can obtain the terminal velocity $\mathbf{v}_\infty$ as the limit when

time approaches infinity,

$$\mathbf{v}_\infty = \lim_{t \to \infty} \mathbf{v}(t) \tag{3.23}$$

$$\Rightarrow \qquad = \lim_{t \to \infty} [\frac{m\mathbf{g}}{k} + (\mathbf{v}_0 - \frac{m\mathbf{g}}{k})e^{-\frac{kt}{m}}] \tag{3.24}$$

$$\Rightarrow \qquad = \frac{m\mathbf{g}}{k}. \tag{3.25}$$

Let us refer to this as the terminal velocity equation,

$$\boxed{\mathbf{v}_\infty = \frac{m\mathbf{g}}{k}.} \tag{3.26}$$

We integrate the velocity equation 3.22 once more to obtain the trajectory,

$$\mathbf{x}(t) = \int \frac{m\mathbf{g}}{k} + (\mathbf{v}_0 - \frac{m\mathbf{g}}{k})e^{-\frac{kt}{m}} dt \tag{3.27}$$

$$\Rightarrow \qquad \mathbf{x}(t) = \frac{m\mathbf{g}}{k}t - \frac{m}{k}(\mathbf{v}_0 - \frac{m\mathbf{g}}{k})e^{-\frac{kt}{m}} + \mathbf{C}_4. \tag{3.28}$$

To solve for $\mathbf{C}_4$, we impose the initial conditions again that $\mathbf{x}(t_0) = \mathbf{x}_0$ at time $t = 0$,

$$\mathbf{x}_0 = \frac{m\mathbf{g}}{k}t_0 - \frac{m}{k}(\mathbf{v}_0 - \frac{m\mathbf{g}}{k})e^{-\frac{kt_0}{m}} + \mathbf{C}_4, \tag{3.29}$$

$$\Rightarrow \qquad \mathbf{x}_0 = 0 - \frac{m}{k}(\mathbf{v}_0 - \frac{m\mathbf{g}}{k})(1) + \mathbf{C}_4, \tag{3.30}$$

$$\Rightarrow \qquad \mathbf{C}_4 = \mathbf{x}_0 + \frac{m}{k}(\mathbf{v}_0 - \frac{m\mathbf{g}}{k}) \qquad . \tag{3.31}$$

Substituting $\mathbf{C}_4$ back in, we obtain the equation

$$\mathbf{x}(t) = \frac{m\mathbf{g}}{k}t - \frac{m}{k}(\mathbf{v}_0 - \frac{m\mathbf{g}}{k})e^{-\frac{kt}{m}} + \frac{m}{k}(\mathbf{v}_0 - \frac{m\mathbf{g}}{k}) + \mathbf{x}_0. \qquad (3.32)$$

However, if we substitute the terminal velocity, this yields the drag trajectory equation.

$$\boxed{\mathbf{x}(t) = \mathbf{v}_\infty t + \frac{m}{k}(\mathbf{v}_0 - \mathbf{v}_\infty)(1 - e^{-\frac{kt}{m}}) + \mathbf{x}_0.} \qquad (3.33)$$
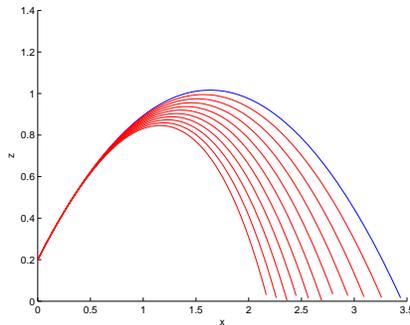
Figure 3.4: *The trajectory of a particle with drag. Different red paths correspond to different drag constants, k, with the same mass, m.*

We now have an analytic description for the particle flight paths. They will all start with a curved motion and settle on a linear trajectory after enough time has ellapsed.

3.2 Inverse Dynamics

Now that we have the formulation for simple trajectories (Equation 3.4) as well as those involving a drag force (Equation 3.33), we investigate how

to solve these equations in reverse given all the parameters to describe the motion. We will denote each variable specific to time of impact t_I the position $\mathbf{x}_I = \mathbf{x}(t_I)$ and velocity $\mathbf{v}_I = \mathbf{v}(t_I)$. We assume we know the drag coefficient, the mass of the object, and gravity.

Let us assume we know the parameters k, m, in the terminal velocity equation 3.26. A basic test to see if a particle yields enough information to reconstruct accurately would be to see if the impact velocity is below this speed. There are two cases why particles approaching terminal velocity are ill-conditioned. The first case is if a particle is travelling at terminal velocity, the net force is zero and so the particle moves at a uniform speed. This forms a line in space. If the particle then impacts a surface along this line, since the speed is uniform along this line, the impact will look the same no matter where along this line. Since there is no information in how long the line is, we cannot reconstruct the full trajectory.

The second case is due to the directional ambiguity. In the first case, we may at least obtain a direction in the ground plane as to where it was coming from. However, it is also known when one of the particles has reached terminal velocity, the xy component of its velocity is zero. This is true because $\mathbf{v}_\infty = \frac{m\mathbf{g}}{k}$ and $\mathbf{g}$ has zero for its first two components. Essentially, the terminal velocity can also be written as $(0, 0, -\frac{mg}{k})$. The reconstructed path for this would be pointing straight up and yields no xy planar information. As it approaches the terminal velocity it may be within our measurement error, and so the problem becomes ill-conditionned.

Supposing the velocity of the particle is below the terminal velocity, we know also some boundary conditions for the position of impact $\mathbf{x}_I$ and

velocity of impact $\mathbf{v}_I$ at the time of impact t_I which we can define to be $t_I = 0$. We can then solve for the trajectory at any point in time before this point using the trajectory equation 3.33.

3.3 2D Probability Density Function Formulations

In this section, we will derive different probability density functions (PDFs) for various scenarios. In particular, Section 3.3.1 derives a 2D PDF for a particle undergoing parabolic motion and in Section 3.3.2 we go on to describe how to obtain the 2D PDF for a particle undergoing ballistic motion with drag force applied. For both of these, we assume we know the impact angles, positions and speeds. Additionally, for the drag formulation we also assume we know the mass and drag coefficient of the particle.

Section 3.3.3 describes how we can use many of the 2D trajectory PDFs for different impact sites to a single 2D PDF for the area of origin, which assumes we know impact angles and positions along with the speeds still. Section 3.3.4 describes how to obtain the same 2D PDF for the area of origin without knowing the speed at each impact site, essentially reducing the problem to only needing to know the impact positions and directions of impact.

More intuitively, we can see the modularity of our algorithm with known and unknown inputs. All the PDFs require at least impact angles and directions at each impact site.

There are two major decisions concerning which PDF to use. The first

is knowledge of the impact speed and the second is knowledge of the drag parameters k and m. If we do not know the impact speed, we show in Section 3.3.4 how we can iterate over various impact speeds to obtain a good area of origin estimate given some assumptions. If we do know the impact speeds for every impact site, we can simply use the area of origin estimate as described in Section 3.3.3. These two methods can use either the parabolic or drag equations of projectile motion.

If we know the drag parameters, we can use the drag trajectory equation to reconstruct the main path and PDF as in Section 3.3.2 and if we do not we can revert to using the parabolic motion equation as described in Section 3.3.1.

3.3.1   A Single Particle with Parabolic Trajectory

In the previous section, we have shown how to analytically solve for the trajectory starting at the impact position $\mathbf{x}_I$. Now we will solve for an approximate solution encoding errors in the impact variables, which will be the ones that shall be measured. This will show us, at a time t where the particle may have come from.

For this first derivation, we shall restrict ourselves to reconstructing the simple parabolic trajectory described by Equation 3.4. We do this to show how to reconstruct accurately the area of origin with only knowing the impact positions and velocities, and not the mass and drag coefficient of the particle. Let us consider the 2D Cartesian Plane with z representing vertical height, and x representing horizontal distance.

Since parabolas are symmetric, we can obtain the same flight path by

firing a projectile from the same impact position with the same velocity as the impact velocity. For simplicity, let the impact position be $\mathbf{x}_0 = (0,0)$ with impact speed v_0 and angle θ. These parameters will be the position and velocity of impact $\mathbf{x}_I$ and $\mathbf{v}_I$ for each measured stain. We will denote this main reconstruction path as,

$$\mathbf{c}_m(t) = (x_m(t), z_m(t)). \tag{3.34}$$

That is to say:

$$x_m(t) = v_0 \cos(\theta) t \tag{3.35}$$

$$\text{and} \qquad z_m(t) = v_0 \sin(\theta) t - \frac{1}{2} g t^2. \tag{3.36}$$

Let now us consider projectile fired at a slightly different angle $\theta + \delta_\theta$ for some small positive or negative δ_θ as well as a slightly different initial speed δ_v. This then yields a new path as a function of small perturbations in the angle and initial speed:

$$\mathbf{c}_\delta(t, \delta_\theta, \delta_v) = (x_\delta(t, \delta_\theta, \delta_v), z_\delta(t, \delta_\theta, \delta_v)) \tag{3.37}$$

Where,

$$x_\delta(t, \delta_\theta, \delta_v) = (v_0 + \delta_v) \cos(\theta + \delta_\theta) t \tag{3.38}$$

$$\text{and,} \qquad z_\delta(t, \delta_\theta, \delta_v) = (v_0 + \delta_v) \sin(\theta + \delta_\theta) t - \frac{1}{2} g t^2 \qquad . \tag{3.39}$$

We can also consider this path $\mathbf{c}_\delta(t, \delta_\theta, \delta_v)$ relative to the known recon-

structed trajectory $\mathbf{c}_m(t)$ with the equation

$$\mathbf{c}_\delta(t, \delta_\theta, \delta_v) = \mathbf{p}(t, \delta_\theta, \delta_v) + \mathbf{c}_m(t), \tag{3.40}$$

where $\mathbf{p}(t, \delta_\theta, \delta_v)$ is a perturbation vector from the main path varying over time. We can solve for $\mathbf{p}(t, \delta_\theta, \delta_v)$ by rearranging Equation 3.40,

$$\mathbf{p} = \mathbf{c}_\delta - \mathbf{c}_m. \tag{3.41}$$

We can solve for this perturbation vector $\mathbf{p}(t, \delta_\theta, \delta_v) = (x_p(t, \delta_\theta, \delta_v), z_p(t, \delta_\theta, \delta_v))$ as follows,

$$
\begin{align}
x_p &= x_\delta - x_m \tag{3.42}\\
\Rightarrow &= (v_0 + \delta_v) \cos(\theta + \delta_\theta)t - v_0 \cos(\theta)t \tag{3.43}\\
\Rightarrow &= t((v_0 + \delta_v) \cos(\theta + \delta_\theta) - v_0 \cos(\theta)) \tag{3.44}\\
\text{and,} \quad z_p &= z_\delta - z_m \tag{3.45}\\
\Rightarrow &= (v_0 + \delta_v) \sin(\theta + \delta_\theta)t - \frac{1}{2}gt^2 - v_0 \sin(\theta)t + \frac{1}{2}gt^2 \tag{3.46}\\
\Rightarrow &= t((v_0 + \delta_v) \sin(\theta + \delta_\theta) - v_0 \sin(\theta)). \tag{3.47}
\end{align}
$$

We then obtain

$$
\boxed{
\begin{aligned}
\mathbf{p}(t, \delta_\theta, \delta_v) &= (x_p(t, \delta_\theta, \delta_v), z_p(t, \delta_\theta, \delta_v)),\\
\text{with} \quad x_p(t, \delta_\theta, \delta_v) &= t((v_0 + \delta_v) \cos(\theta + \delta_\theta) - v_0 \cos(\theta)),\\
\text{and,} \quad z_p(t, \delta_\theta, \delta_v) &= t((v_0 + \delta_v) \sin(\theta + \delta_\theta) - v_0 \sin(\theta)).
\end{aligned}
} \tag{3.48}
$$

The equation for the relative vector from the main path to a perturbed path is given by the above equation. This can be seen in Figure 3.5.
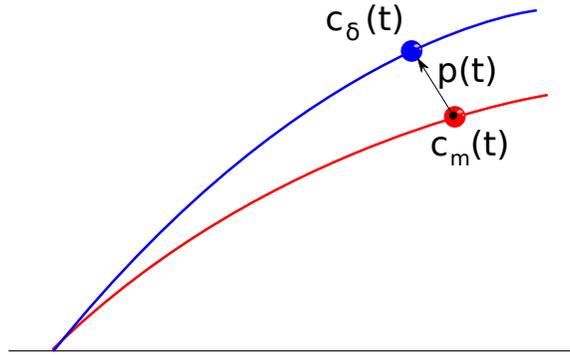
Figure 3.5: *Formulation of main path* $\mathbf{c}_m$*, perturbed path* $\mathbf{c}_\delta$ *and perturbation vector* $\mathbf{p}$*.*

Now, let us examine the magnitude of this perturbation vector. There is a common t term in both x and y components, and this scales linearly over time. We can see this effect in Figure 3.6. We see a progression of neighbouring particles, i.e. particles fired with some perturbation and/or angle.

If we wish to reconstruct a particle's past trajectory by a single path, then the error in reconstruction scales linearly in the amount of time backward. However, if we know the distribution of the error in our angle and speed measurements then we can create a 2D probability density function representing this data.

We assume normally distributed noise in both the angle and speed measurements. This forms a mean zero 2D Gaussian with standard deviations σ_θ and σ_v in the angle and speed respectively. At a given time t, a probability for the position can be created by warping this 2D Gaussian to 2D
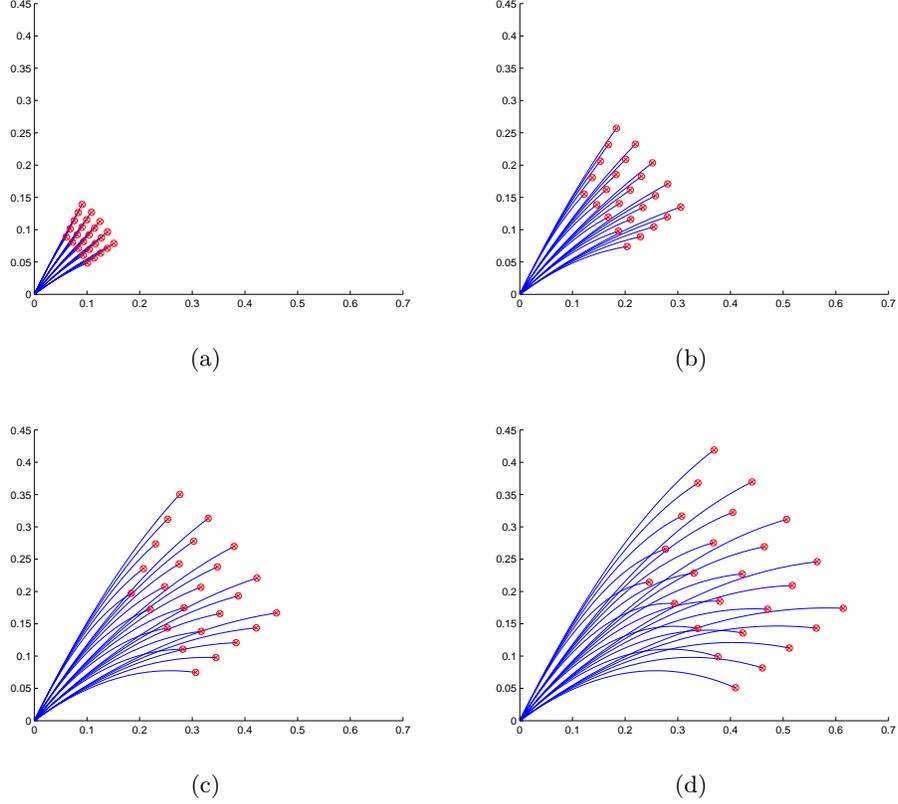
Figure 3.6: *Trajectories of nearby perturbations over time in the xz plane.*

world space, centered at $\mathbf{c}_m$. The warp is a circular warp described in Equation 3.48 and illustrated in Figure 3.7. For simplicity and computation expense, we approximate this by a 2D Gaussian.

We now need to compute the covariance matrix for this distribution. We approximate this by a uniform Gaussian distribution. We just need a standard deviation for this multiple of the identity matrix. Let σ_θ and σ_v be the standard deviations for the error in the angle and impact speed respectively. The covariance matrix for this distribution will be defined

26
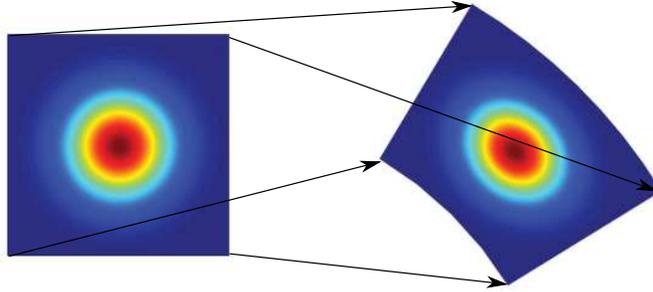
Figure 3.7: *Circular warp of the Gaussian distribution*

with the help of the norm of the perturbation vector equation 3.48 with these input arguments

$$COV(t, \sigma_\theta, \sigma_v) = \|max_{p_\theta \in \{-\sigma_\theta, \sigma_\theta\}, p_v \in \{-\sigma_v, \sigma_v\}} \mathbf{p}(t, p_\theta, p_v)\| \mathbb{I}, \qquad (3.49)$$

where $\mathbb{I}$ is the identity matrix.

An example of this Gaussian can be seen in Figure 3.8. Notice the scale changing over time. This also means the further we are from the source, the less probable it is that the particle will be at that point along the reconstructed path.

(a)   (b)   (c)
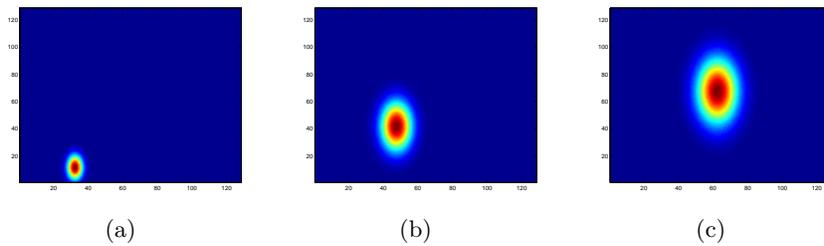
Figure 3.8: *Probabilities for a given instant in time*

This is fine for one point in time, however we want to know what the

probability is over the space for all points in time. We thus integrate over time. Let our 2D normal distribution be $\mathcal{N}$ with mean $\mathbf{c}_m(t)$ and covariance matrix $COV(t, \sigma_\theta, \sigma_v)$.

It should be noted that the $\mathbf{x}_I$ and $\mathbf{v}_I$ are parameters to both the covariance matrix as well as the main reconstruction path. They provide the initial conditions for the reconstruction path. For clarity of notation we omit them from the equations.

$$P = \int_{t \in [t_I - t_E, t_I]} \mathcal{N}(\mathbf{c}_m(t), COV(t, \sigma_\theta, \sigma_v)) \, dt, \qquad (3.50)$$

where t_E is how far back in time we are interested in. This forms a PDF which denotes the probability of a particle being at a given point in space in some known time interval. This is illustrated in Figure 3.9.

(a)   (b)   (c)
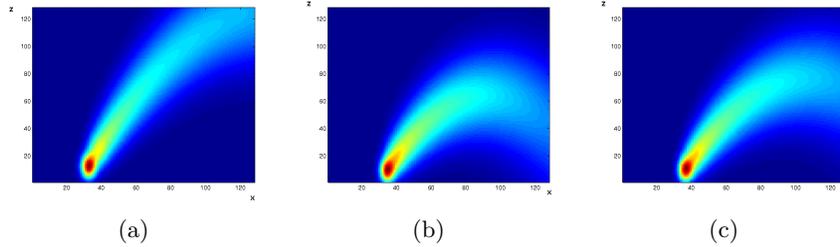
Figure 3.9: *Probability densities for different reconstructed paths*

This section has been derived for objects without a drag force. We can take a similar approach to derive the probability density function for ones with drag force, or just change the main path $\mathbf{c}_m$ to the one in Equation 3.33, while keeping the covariance matrix for the probabilities the same as for the simple projectile motion.

28

3.3.2 A Single Particle with Drag Force Trajectory

In Section 3.3.1, we derived a probability density function for particles undergoing parabolic motion. We want to do the same thing for particles undergoing motion with drag forces applied. We will now assume we can obtain the parameters for each particle describing the drag coefficient k, and mass of the object m.

We will take a similar approach as in Section 3.3.1 to obtain the PDF for drag trajectories by estimating the Gaussian distribution at a time t. Since we described how to reconstruct the motion given an endpoint in Section 3.2, we know this Gaussian distribution should have a mean $\mathbf{c}_m(t, \mathbf{x}_I, \mathbf{v}_I, m, k)$, which is the reconstruction path given by the drag trajectory equation 3.33 and looking in the negative axis of time. Now we seek to obtain the covariance matrix $COV(t, \sigma_\theta, \sigma_v)$. We will use the same definition as in the covariance equation 3.49, which uses the perturbation vector.

We will now derive the perturbation vector for trajectories undergoing drag. To define the perturbation vector we must first state the perturbed path for a trajectory with drag. Unlike the case of parabolic motion, it is not symmetric about the gravity vector. However, like with the case of inverse dynamics let us fix a boundary value at the time of impact and work backwards in time before this.

From the drag equation 3.33, a trajectory with drag force has the form

$$\mathbf{x}(t) = \mathbf{v}_\infty t + \frac{m}{k}(\mathbf{v}_0 - \mathbf{v}_\infty)(1 - e^{-\frac{kt}{m}}) + \mathbf{x}_0. \tag{3.51}$$

However, if we are interested in reconstruction path, we can let time be

negative, and so we obtain the main path

$$\mathbf{c}_m(t) = \frac{m}{k}(\mathbf{v}_0 - \mathbf{v}_\infty)(1 - e^{\frac{kt}{m}}) - \mathbf{v}_\infty t + \mathbf{x}_0. \tag{3.52}$$

Now we want to know what a perturbed reconstruction path would look like, so we let $\mathbf{v}_{0\delta} = (v_0 + \delta_v)(\cos(\theta + \delta_\theta), \sin(theta + \delta_\theta))$. Substituting that into Equation 3.52, we get the perturbed path equation for drag trajectories,

$$\mathbf{c}_\delta(t, \delta_\theta, \delta_v) = \frac{m}{k}(\mathbf{v}_{0\delta} - \mathbf{v}_\infty)(1 - e^{\frac{kt}{m}}) - \mathbf{v}_\infty t + \mathbf{x}_0. \tag{3.53}$$

Now to obtain the perturbation vector, which defines the covariance matrix, we subtract the perturbation path from the main path,

$$\mathbf{p}(t, \delta_\theta, \delta_v) = \mathbf{c}_\delta - \mathbf{c}_m \tag{3.54}$$

$$\Rightarrow \qquad = \quad \begin{aligned} &\frac{m}{k}(\mathbf{v}_{0\delta} - \mathbf{v}_\infty)(1 - e^{\frac{kt}{m}}) - \mathbf{v}_\infty t + \mathbf{x}_0 - \\ &\frac{m}{k}(\mathbf{v}_0 - \mathbf{v}_\infty)(1 - e^{\frac{kt}{m}}) + \mathbf{v}_\infty t - \mathbf{x}_0 \end{aligned} \tag{3.55}$$

$$\Rightarrow \qquad = \frac{m}{k}(1 - e^{\frac{kt}{m}})(\mathbf{v}_{0\delta} - \mathbf{v}_0). \tag{3.56}$$

Expanding, this gives us the perturbation vector equations for trajecto-

ries with drag,

$$
\mathbf{p}(t, \delta_\theta, \delta_v) = (x_p(t, \delta_\theta, \delta_v), z_p(t, \delta_\theta, \delta_v)),
$$

$$
\text{with} \quad x_p(t, \delta_\theta, \delta_v) = \frac{m}{k}(1 - e^{\frac{kt}{m}})((v_0 + \delta_v)\cos(\theta + \delta_\theta) - v_0\cos(\theta)),
$$

$$
\text{and,} \quad z_p(t, \delta_\theta, \delta_v) = \frac{m}{k}(1 - e^{\frac{kt}{m}})((v_0 + \delta_v)\sin(\theta + \delta_\theta) - v_0\sin(\theta)).
$$

$$(3.57)$$

Looking at the magnitude of this, we see the exact same shape factor as before, coming from $\mathbf{v}_{0\delta} - \mathbf{v}_0$. This initial shape is scaled by $\frac{m}{k}$ as well, which was not present in the initial shape of the parabolic trajectory as drag was not present. We also have a new scaling factor with time, in both components $(1 - e^{\frac{kt}{m}})$ as opposed to linearly in the parabolic case. That means, to compute the PDF for the drag trajectory we simply need to scale the Gaussian approximations initially by $\frac{m}{k}$ and over time with $(1 - e^{\frac{kt}{m}})$. The circular warp is the same as in Section 3.3.1.

3.3.3 Area of Origin PDF Given Velocities

In the previous section we have created a probabalistic model of where a particle may be for all times in a given space. The highest probability is near the measured point. However, we wish to find the area of origin, or where it is likely to be for all points. We assume that all particles are independent observations from the same source, and therefore multiply all the values together for all particles.

Suppose we have a 2D probability density function for each impact particle denoted P_i for the ith point. The input to these PDFs are known

estimates for the position $\mathbf{x}_I^i$ and velocity $\mathbf{v}_I^i$ at the time of impact. To obtain the area of origin PDF, denoted P_{AOO}, we multiply all the P_i to find a common likely area all the particles will have been through,

$$P_{AOO} = \prod_i P_i. \qquad (3.58)$$

(a)

(b)

(c)

(d)
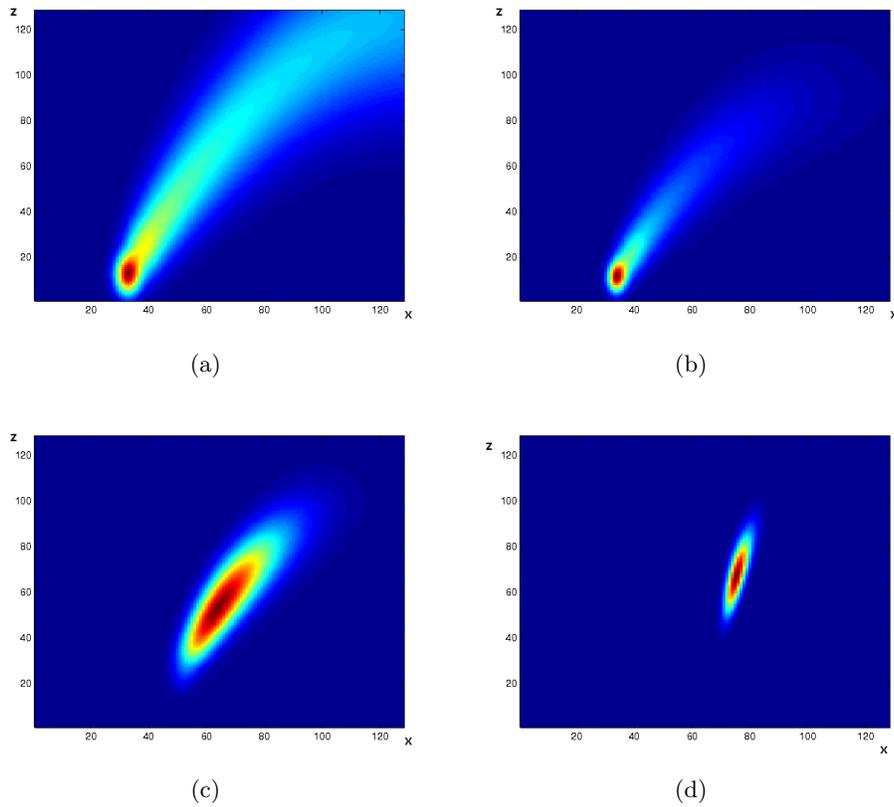
Figure 3.10: *Multiplied probabilities for 1, 2, 4, 21 different particles respectively.*

In Figure 3.10, we see the two probability density functions from the previous section multiplied together. This tells us where the two regions

agree. The lighter colours indicate more probable areas for the particles to have both come from. If we do this for more and more particle probabilities, we converge on a more likely solution for the area of origin.

3.3.4   Area of Origin PDF Without Velocites

The previous section works works assuming one knows the impact velocities with error in measurement for each particle. Although velocities may be estimated from the impact sites, some impacts only allow us to measure direction. We now show how to estimate the area of origin for impacts without a velocity measurement.

|(a)|(b)|(c)|
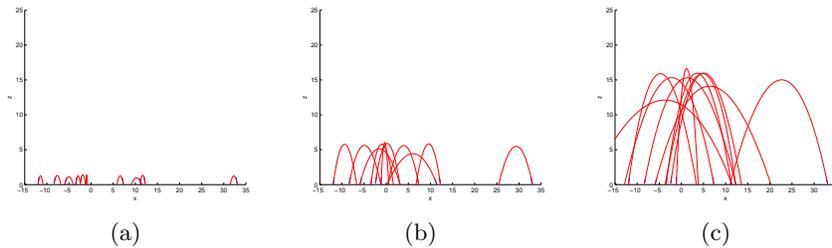
Figure 3.11: *We sample different impact speeds. One speed defines the shape of all the parabolas, as we constrain them to have the same initial speed. Shown in Figure 3.11a is a very low speed. Note that the parabolas do not converge on a unique solution. We increase the speeds in Figure 3.11b and note that more of the parabolas converge. In Figure 3.11c most of the parabolas converge at a single point which would be our area of origin estimate for this particular velocity.*

We introduce a constraint where all the impact speeds should be equal for a given scenario. We may do this as we expect the velocites to be similar as the same initial kinetic energy was added and at this time the particles had the same potential energy. An example of how the paths may converge

while uniformly increasing the speed is illustrated in Figure 3.11.

Suppose the approximate speed for all the particles is in an interval $S = [s_{start}, s_{end}]$. Then we look at the maximum likelihood over all these speeds by again taking the max over all the frames

$$P_v = \int\limits_{s \in S} P_{AOO}(s)ds. \tag{3.59}$$

The area of origin should lie where this new density function P_v is maximal, otherwise known as the maximum likelihood estimate (MLE). An example of this PDF can be seen in Figure 3.12.

3.4 Calculating the 3D Area of Origin

So far all of the PDFs have been for the 2D case. If we want to use this for the 3D case, we can simply project onto the xz-plane and yz-plane. If we want to know what this probability distribution is in 3D we can multiply the values of the two perpendicular planes at different voxel values, then solve for the MLE in this 3D space. If we have the xz PDF defined as

$$\pi_{xz} = P_V(\mathbf{x}_I^{xz}, \mathbf{v}_I^{xz}), \tag{3.60}$$

where $\mathbf{x}_I^{xz}$ and $\mathbf{v}_I^{xz}$ are the x and z components of the impact positions and angles respectively. Similarly the yz PDF defined as

$$\pi_{yz} = P_V(\mathbf{x}_I^{yz}, \mathbf{v}_I^{yz}), \tag{3.61}$$

Figure 3.12: *Summed Probabilities for: 1, 4, 9, and 12 different velocities respectively.*

for the y and z components. Then we can write the 3D PDF as

$$P_{3D}(x, y, z) = \pi_{xz}(x, z)\pi_{yz}(y, z). \tag{3.62}$$

The assumed solution to the area of origin is most likely region that all particles have passed through at some point in time. This solution is defined

as a maximum likelihood estimate solving for $\mathbf{x}_A$, and is written as

$$\mathbf{x}_A = (x_A, y_A, z_A) = \arg \max_{(x,y,z)} P_{3D}(x, y, z). \qquad (3.63)$$

3.5 Error Bound on Linear Estimate

In this section we will discuss conditions needed for which the algorithm to work well. Consider the 2D xy plane, where gravity has no effect on the particles in question. Suppose we have two different scenarios, each where we obtain three impact sites with the same measuremnt error but different positions. In scenario 1, we shall have the three impact sites arranged close together and in scenario 2 they are spread out equidistantly over a circle. This can be seen in Figure 3.13.

The blue lines are the estimated trajectories, the red lines are the actual trajectories. The black dots are the impact site locations and the red dot is the area of origin. We can see that if we were to minimize the distance between the three lines for each scenario, it would be located somewhere in the blue region. This estimation of the origin is obviously more well constrained in the second scenario.

If we have more estimates from the same area in scenario 1 then our estimate for the area of origin will not improve. If we have more estimates in scenario 2, in a different place on the circle then our estimate for the AOE will improve. If we use our probabalistic framework, we this is the case in Figure 3.14. In the first and second images, we have the PDFs for two particles that are close in position. In the third image we multiply them to

(a) Impact sites close together   (b) Impact sites spread out
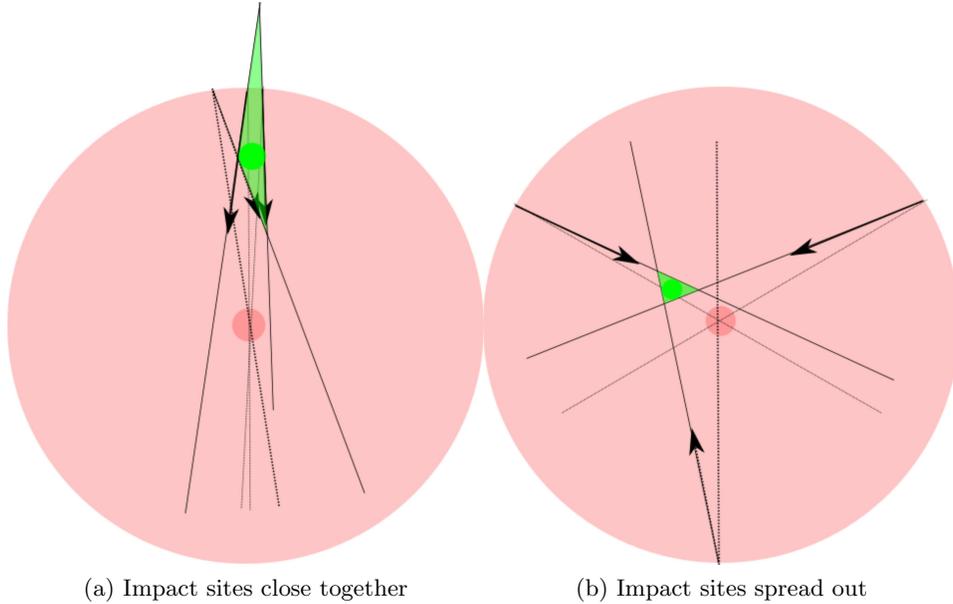
Figure 3.13: *The effect of position on reconstruction error. The green circle is where the reconstructed area of origin is. The actual area of origin is in the centre of the large circle. With added noise and sparse positions, the bounded error is closer to the actual area of origin.*

obtain a PDF and estimate of where the area of origin is. The actual area of origin is around the middle of the image, but the highest likelihood estimate is near the two particles, which is clearly false. More estimates from this area will not constrain the system any better. In the linear case, this is simply an illconditioned system, as the lines are collinear within measurement error.

Let us examine what would better condition our system, and a bound for the error estimate. If we have sites uniformly spread out over a circle of radius d from the area of origin, and have an error angle δ_θ, then the error will be

$$\epsilon = d\sin(\delta_\theta), \tag{3.64}$$

(a) PDF for particle A   (b) PDF for particle B   (c) PDF obtained by multiplying those of A and B
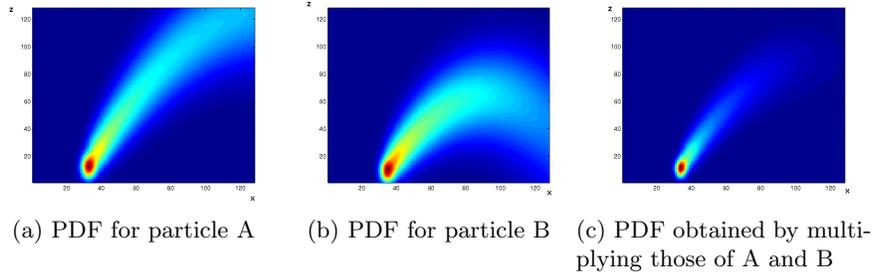
Figure 3.14: *An ill-conditionned scenario*

where ϵ is the distance from the area of origin to the ray created with angle δ_θ.

We can consider this error for a circular region around the area of origin. If we have points that are closer to the area of origin, they will more accurately inform us where the center is. This is true as it scales with distance from the center, d. However, one should exercise caution because although the impact angle may be estimated to within a few degrees, the direction of the impact site near the AOE will be more vertical and unstable.

Chapter 4

Bloodstain Analysis

In Chapter 3, we derived a probabalistic estimation for the inverse trajectory. We went from impact angles and positions to a PDF denoting the likely areas the particle to have come from. In this section, we describe a methodology to go from images to the desired angles and positions to be used in our reconstruction approach in the previous chapter. We begin with a review of previous techniques and general observations about the stains in Section 4.1. In section 4.3, we describe an approach to fit ellipses to stains in the image. This is important to do accurately as this is going to ensure a good reconstruction estimate. Section 4.4 describes a method of obtaining blood droplet parameters from the images. This step is necessary if we want to use the drag equation 3.33 for inverse dynamics computation.

Even though we have obtained ellipse estimates, we need to compute the direction of impact from this. We thus describe a way to ensure all the reconstructed directions are consistent in Section 4.5. If we wish to bound the problem more significantly, it is very important to get an estimate for the average velocity of all the droplets or velocities on a per droplet basis. We show how to do this in Sections 4.6 and 4.7. Lastly, given a series of impact angles with associated positions, we show how to prune the outlier

droplets using RANSAC in Section 4.8.

4.1 Blood Droplet Shape

Now that we know generally what the trajectory of the droplet should look like, it will also help us to know what the shape of the droplet is so we know what to expect from a droplet in the air given the impact stain.

In [18] we learn that blood is a liquid with high surface tension force, and thus in midair it will force itself into a sphere very quickly. Initially, there will be oscillations but it will settle into a sphere after a very small amount of time.

Since the droplets are all spheres, by time they reach a moment of impact they will have an initial impact shape of approximately an ellipse, since the projection of a sphere onto a $2D$ surface is an ellipse. Since the blood is free to move after that initial impact, it may move into what is called satellite splatters. These spatters may compliment the ellipse spatter, or may not exist at all if the blood is wholly contained in the ellipse. This is illustrated in Figure 4.1.

Figure 4.1: *An example cow blood stain, with satellite spatters*

We will now consider the effect of different impact speeds on the shape of the stain. Suppose we have two identical droplets, but one impacting at a higher velocity. Their stains will be different, the stain of the droplet with the higher velocity will be larger, and less dense in terms of colour than the one with lower velocity. This is because the increase in energy will push the fluid particles outwards with more force to overcome the surface tension force moreso than its lower velocity counterpart. This can be seen in Figure 4.2.

Figure 4.2: *Cow blood droplets as speed of impact is varied*

Also, let us compare the result of the same droplet, with exact same velocities but falling on the surface at different angles. The droplet with the higher angle will have a stain that is more elliptical, but with a larger surface area. This is true since it will always have the same radius and thus minor axis, but different major axis length. This effect can be seen in Figure 4.3.
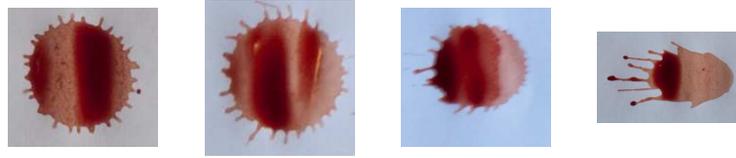
Figure 4.3: *Cow blood droplets as angle of impact is varied*

We use this information in the next few sections, but will note that there is always an ellipse present in the stains even while varying all the impact parameters.

4.2 Obtaining A Mask of the Stains

We obtain a mask of the stains to perform ellipse fitting on, as well as to label different connected components to separate stains. We have tried different masks, including some dependent on the absorption coefficients of blood, however the best resultant mask used was by thresholding the blue channel. The blood absorbs most of the green and blue colours, leaving it with a red appearance. The blue channel had better contrast than the green channel which made it easier to threshold so we used it instead of the green. Once we performed this threshold, we are left with a mask. To work with stains separately, we consider the ones that are disjoint in the mask. That is to say we obtain a labeling for each connected component in the mask. The mask and connected component label are both shown in Figure 4.4,

42

(a) Masked Stains

(b) Different labeled stains
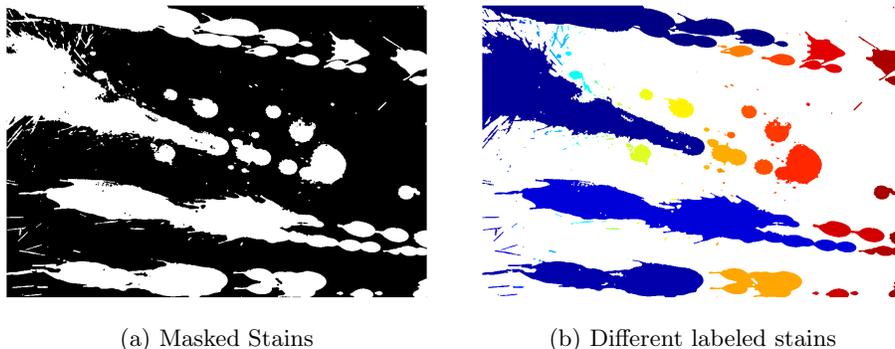
Figure 4.4: *The result of masking bloodstain images*

4.3   Ellipse Fitting

In the computer vision community there are various methods of fitting ellipses ranging in complexity from a direct least squares fitting [10], to a multi-population genetic algorithm that evolves ellipse parameters over time [21]. In BPA reconstruction, fitting ellipses is key to obtain the impact directions for each stain.

For bloodstains, especially low and medium velocity stains, we provide a new approach to robustly detect ellipses of the bloodshop variety. Our method is robust and can handle many satellite stains. We find the maximum convex region of the droplet and perform a standard ellipse detection algorithm edges on the boundary of that convex region. We ignore the parts of the ellipse outside this region, which is notably the satellite stains.

First we assume we are given a binary mask M (with elements either *true* or *false*) of the stain in question. We perform a distance transform on this mask to find the region with largest inscribed circle (which is assumed

to be inside the ellipse region). We now consider all pixels that are visible to this circle, the ones that are not visible get set to $false$, creating a visibility mask. This is performed by casting rays from points on the circle and the ones that leave the mask are not visible. This is referred to as a visibility test.

This utilizes a definition for convex regions, that all points in the convex region must be visible to all other points in the same region. We sample many points from a region we assume to be in the convex area of the mask to prune out regions we know not to be part of the convex region. The pixels in this visibility mask that correspond to edges in the original mask should be part of the main ellipse as they were visible to a large portion of the convex region.

(a) Unprocessed Cow Blood Stain

(b) Stain Masked
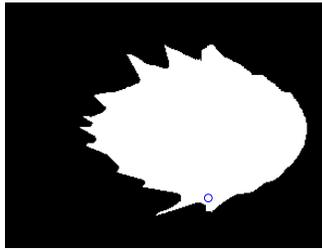
(c) Stain Visibility Test - 1 point
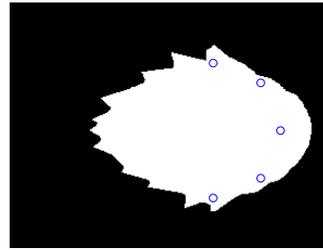
(d) Stain Visibility Test - 5 points

Figure 4.5: *Different stages of bloodstain ellipse fitting*

In practice, we discretize the circle and choose a few points from it. We perform this visibility test for this set of points as we can see for 1 and 5 points in Figure 4.6. Once we have performed the visibility tests for all the points on the circle, the resultant mask will have a large convex region with sharp exterior regions. The sharp exterior regions come from the shadows of the visibility test. Now if we only use the pixels that correspond to edges in the original mask, they should lie on the ellipse.
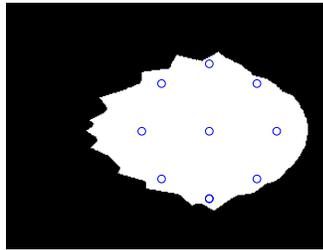
(a) Circular Visibility Test
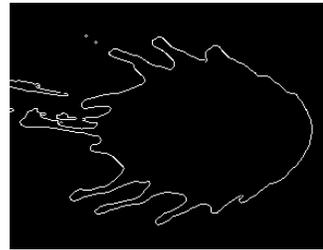
(b) Edges in the Original Mask
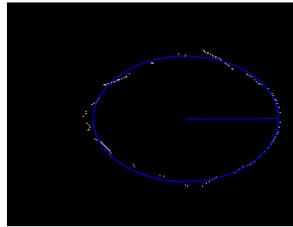
(c) Initial Edges Remaining after Visibility Tests
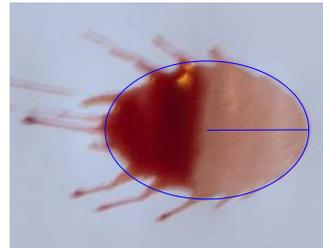
(d) Final Ellipse Fit

Figure 4.6: *Final stages of bloodstain ellipse fitting*

In Figure 4.6, we see the final visibility test for all the discretized points on the circle. Many of the regions are jagged regions caused by the visibility tests, corresponding to shadows. We ignore all the pixels in the mask that do not correspond to edges of the original mask. We then fit an ellipse to this set of points.

4.4 Blood Droplet Parameter Estimation

A useful metric which may help determine the velocity of a droplet as well as the drag coefficient is the droplet's density. We can estimate the radius of the droplet given the fitted ellipse, and since the drag coefficient is only a function of surface area, and the droplets are spherical, we can calculate the drag.

Suppose we fit an ellipse with minor axis radius r. We can estimate the volume of the droplet from the equation for the volume of a sphere

$$V = \frac{4}{3}\pi r^3. \tag{4.1}$$

We also know from Equation 3.8, $k = 6\pi\mu r$ which are all known now. We can also estimate the mass m

$$m = \rho_B V, \tag{4.2}$$

where ρ_B is the density of the given blood. Now we have enough blood droplet parameters to estimate the trajectory. However, if we would like to obtain the impact speed of a droplet as in Section 4.7, it is beneficial to have a density estimation for the droplet in image space, denoted ρ_i. This is formally defined as the thickness of a blood droplet at a given pixel. We can obtain this information from the Beer-Lambert law. This law states that for an incoming ray with intensity I_0 that passes through a medium with absorption coefficient α, and thickness l and outgoing ray I_1, then

(a) Beer-Lambert Law Through a Medium   (b) Beer-Lambert Law applied to a blood-stain
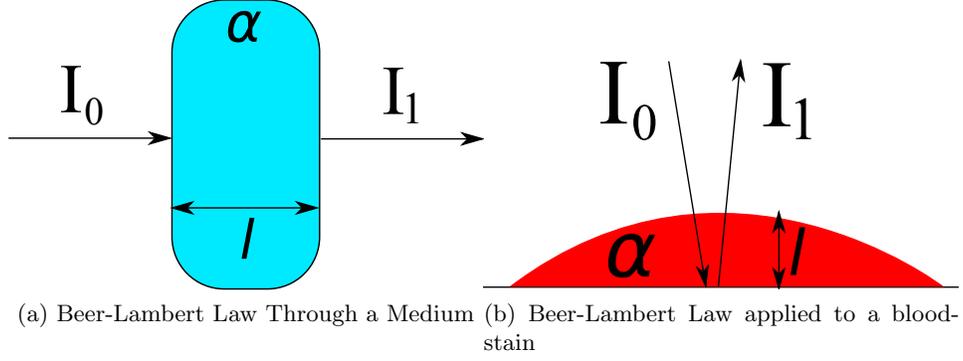
Figure 4.7: *The Beer-Lambert Law*

$$\frac{I_1}{I_0} = 10^{-\alpha l}. \tag{4.3}$$

However for our case, light passes through the blood twice, and we get,

$$\frac{I_1}{I_0} = 10^{-2\alpha l}. \tag{4.4}$$

If we know the input intensity I_0 from the rest of the background and output intensity I_1 from the measured value on a blood droplet then, we can define the image density $\rho_i \equiv l$,

$$\frac{I_1}{I_0} = e^{-2\alpha\rho_i} \tag{4.5}$$

$$\Rightarrow \qquad \ln(\frac{I_1}{I_0}) = -2\alpha\rho_i. \tag{4.6}$$

Rearranging, we obtain the image density equation,

$$\boxed{\rho_i = -\frac{1}{2\alpha}\ln(\frac{I_1}{I_0}).} \tag{4.7}$$

48

4.5 Major Axis Direction Flipping

At this point we have shown how to estimate the ellipses from the blood-stains. However, the impact angle is still ambiguous to a computer. It could have come from either direction along the major axis. We propose a method that obtains the correct direction based on the tangent method.
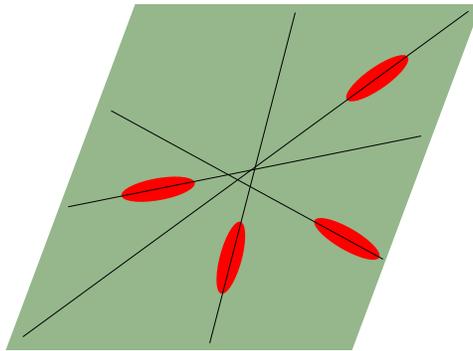
Figure 4.8: *Where the tangent method succeeds. Top view of a spatter scene. For each stain on the floor consider both major axis directions*

The tangent method described in Section 2.1 only works accurately for particles or stains on the floor, as seen in Figure 4.8. If we look back to our free body diagram, Figure 3.3, there is no force moving it outside of a plane formed by the drag force, and gravity. Thus, all trajectories projected to the xy-plane, i.e. the plane perpendicular to gravity, remain lines. This is why the tangent method works in this case. However, for splatters on walls or any surface that isn't coplanar with the ground plane, the tangent fails.

We will consider both directions formed by each of the ellipse major axes. 2 potential directions from each stain, we include both of them, this is illustrated in Figure 4.9.

Figure 4.9: *Two separate directions made with major axes of an ellipse*

In a practical scene, with many stains, it looks like something from Figure 4.10. On the left side we have our input stains with varying axes and angles. We also consider the directions formed by using the opposite major axis. We then project these direction estimates onto the xy plane, where we know the trajectories to be linear. We now have two sets of lines in the xy plane. Since the lines are reflections about the surface, we can solve for two different points of intersection.

(a) The input angles. Some may be flipped the wrong way

(b) All the angles

(c) The correct angles
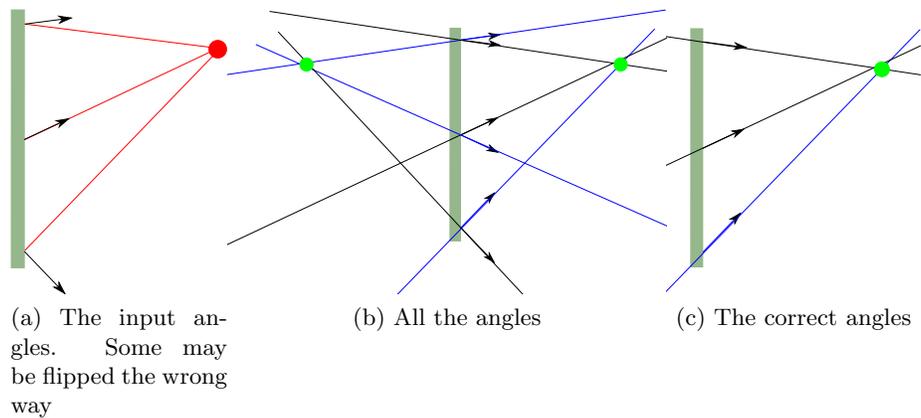
Figure 4.10: *The wall axes flipping algorithm. Top view of a spatter scene. For each stain on the wall with normal facing the right, consider both major axis directions*

If we cluster this data, we will obtain two points of intersection in the xy-plane. One of the points is in front of the wall, and another virtual one is

behind. Obviously we will choose that in front of the wall. Some scenarios will not have a plane perpendicular to the ground, in which case the point that is closer to the plane will be the correct one.
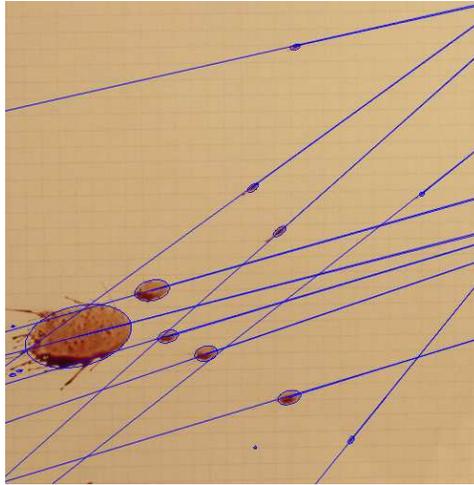
Figure 4.11: *An example where the tangent method does not apply for a wall spatter. Note where the lines intersect all in this region of interest, however the actual area of origin is off to the right. Down in the image corresponds to down in world space.*

If we were to do the tangent method, without projecting to the xy-plane, also known as the ground plane, then the height component z may diverge in the direction of the area of origin. This is more extreme in the case of particles with parabolic trajectory. In Figure 4.11 is a counterexample where we cannot use the tangent method to accurately predict the area of origin from the height component. We are looking at the xz plane, with gravity pointing downwards in the image. There are droplets with varying plane angles in the image. Intersecting them like this would form meaningless lines. However, if we obtain the impact directions for both axes of the

ellipse, project them to the plane and cluster as described above, we can obtain the right orientation for the ellipse. Doing this also obtains a linearx estimate for the xy position of the area of origin.

4.6 Average Velocity Estimation

If we are not interested in individual droplet velocities, we can estimate the average velocity for all droplets in a given area based on the average size of all droplets in that area. This is true because at higher velocities, the surface tension breaks the droplets into smaller ones. Lower velocities allows for bigger droplets.

4.7 Droplet Velocity Estimation

The velocity can be estimated from the stains as well, though not all too apccurately. However, these velocities still give an adequate bound to search the space of possible velocities as described in Section 3.3.4.

If we vary the speed of impact, the splatter size and thus density changes. If we vary the angle of impact, the splatter size and density changes. Suppose we have the density of a droplet in image space ρ_i (not to be confused with ρ_B) , speed of impact v_I and angle of impact θ (measured to the normal of the surface), we can constrain the problem. We know that there exists a maximum density of a droplet of a given size ρ_{max}, which corresponds to a perfect hemisphere on the surface. This occurs when the angle and speed of impact are both 0. We also know that as the speed increases, the density decreases as we're spreading the droplet over a larger area. As the speed

(a) A plot for impact velocity vs density   (b) A plot for the impact angle vs density

Figure 4.12: *Velocity and impact angle versus density*

approaches infinity, the density approaches zero. Similarly for the angle, as it approaches $\frac{\pi}{2}$, the stain will smear over an infinite distance and thus the density becomes zero. We can thus write:

$$\rho = f(\theta, v_I), \tag{4.8}$$

$$\text{and,} \qquad f(0, v_I) = \rho_{max} e^{-A v_I}, \tag{4.9}$$

$$\text{and,} \qquad f(\theta, 0) = \rho_{max} \cos(\theta). \tag{4.10}$$

We can thus form f as follows:

$$f(\theta, v_I) = \rho_{max} \cos(\alpha_{impact}) e^{-A v_I}. \tag{4.11}$$

We then solve for an inversion of this, $v_I = f^{-1}(\theta, \rho)$, which can be

shown to be

$$\rho = \rho_{max} \cos(\theta) e^{-Av_I} \tag{4.12}$$

$$\Rightarrow \qquad \ln(\rho) = \ln(\rho_{max} \cos(\theta)) - Av_I \tag{4.13}$$

$$\Rightarrow \qquad Av_I = \ln(\rho_{max} \cos(\theta)) - \ln(\rho). \tag{4.14}$$

Divding both sides by A, we denote this as the velocity estimation equation,

$$v_I = \frac{1}{A} \ln(\frac{\rho_{max} \cos(\theta)}{\rho}), \tag{4.15}$$

where the value A can be found through experiment. Supposing the experimental data has a large standard deviation in the velocity component, we can obtain approximations by fitting the vertical means of the data with this function, and can also encode the vertical standard deviation, which can be used with the area of origin estimation.

4.8   Pruning

There are many stains in the image, especially with added noise. We get rid of stains that are either too small and would yield no information, or too large and are composed of many different stains. There are enough stains in the image to find good area of origin estimates. We also disallow stains whose computed main stain is smaller than the satellite stains, which is usually because of multiple overlapping stains.

Once we obtain a set of ellipses, a lot of them may be outliers. We perform a random sample concensus, or RANSAC [9], to keep only the

inliers. The ones that agree on an xy-plane coordinate, are the ones we seek to keep. For RANSAC we perform over 9000 iterations, looking for at least 20 good stains (which is the amount forensic investigators usually choose for stain estimates), with varying minimum error dependent on the scale of the scene.

Chapter 5

Experiments

In this chapter, we will describe various experiments and validation we have performed. For all experiments, cow blood was used in substitution of human blood for various scenarios. The first blood experiment is dropping blood from various heights impacting at various angles. In Section 5.1, we determine that our velocity estimation equation 4.15 is feasible. In Section 5.2, we recreate various real world scenarios of blood spatter events and perform our algorithm on the captured data.

5.1   Blood Droplet Experiment

Procedure

In this low velocity experiment, blood was dropped out of an eyedropper at various heights onto a paper surface, inclined at various angles. The heights corresponded to different impact velocities evenly spaced out at one every meter per second. This is illustrated in Figure 5.1. Immediately after the particle hit the surface, the paper is taken off the inclined surface and put on a horizontal surface to along with a calibration pattern and photographed. This calibration pattern is used to warp the image into an orthographic

view, to ensure comparable measurement between stains. Manual cropping was used and the stains were segmented as in Chapter 4. We also computed the image density from Equation 4.7 for each pixel.
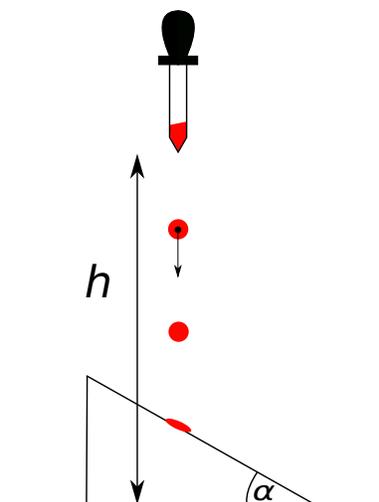
Figure 5.1: *The setup for the blood droplet experiment: varied parameters h for height, which dictates the impact speed and θ the impact angle.*

Results

We did this test for 4 varying angles and 5 varying velocities with 3 tests per angle-velocity combination. Shown are one of the images from this set in Figure 5.2.
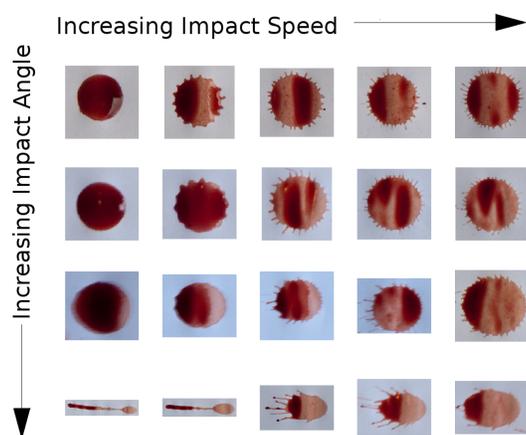
Figure 5.2: *Various blood droplet test stains. Horizontal is varying the speed of impact and vertical is varying the angle of impact*

Computed image densities are shown in Figure 5.3. We can see visually the brightest or most dense images are near the top left, associated with no angle and very low speed. As we go right or down from there the image density decreases on average.
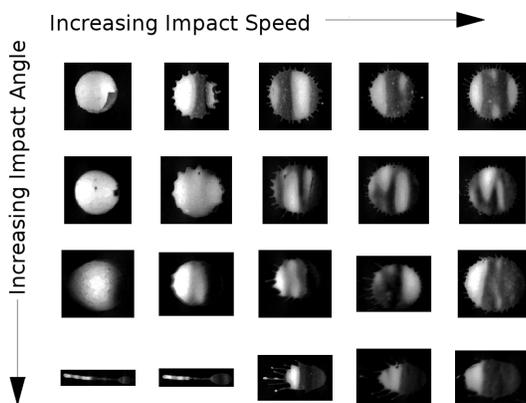
Figure 5.3: *Computed density images for the blood droplet experiment.*

We plot the average image density for the region of the bloodstains in each image in Figure 5.4. If we look at the plot for angles vs. density in Figure 5.4a, we can see that as the velocity gets higher, the density approaches zero as stipulated in the velocity estimation equation, 4.15. This seems to be true for all the different angles we used. If we consider the angles vs. the density in Figure 5.4b, we see a general decrease as we would with the cosine function.

(a) Velocities Vs Density   (b) Angles Vs Density
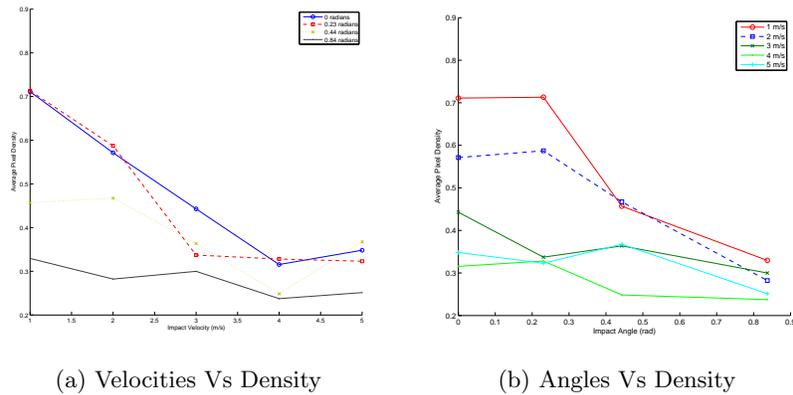
Figure 5.4: *Plots comparing angles, velocities versus densities*

If we look at at 3D plot of all the data, as in Figure 5.5, we can see the overall trend better. We also compare it to the theoretical plot for the same boundaries with value $\rho_{max} = 0.7598$ and $A = 0.1835$ with a residual norm of 0.0850.
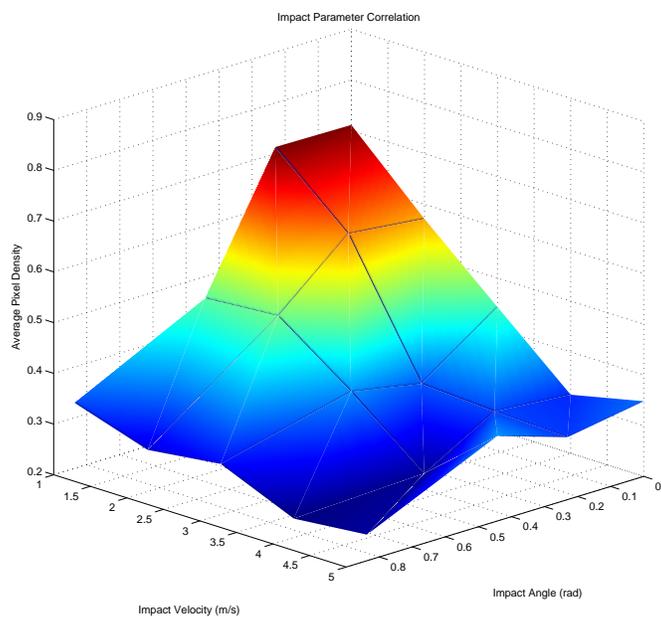
(a) Experimental Image density as a function of impact angle and speed
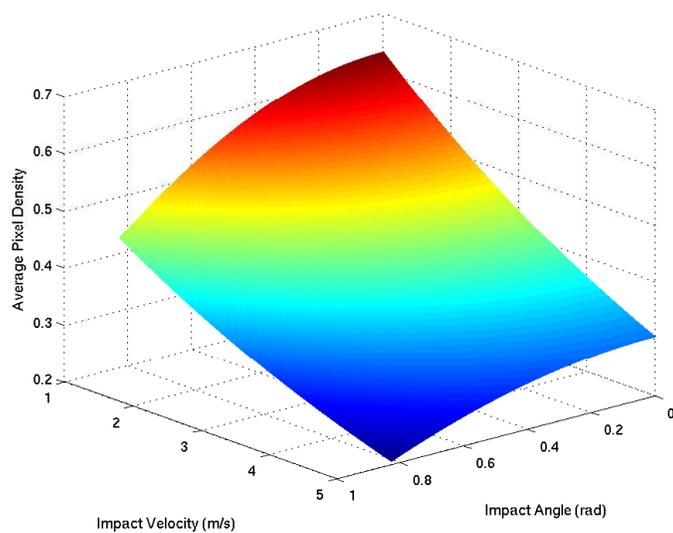
(b) Theoretical Image density as a function of impact angle and speed

Figure 5.5: *3D plots of density as a function of angles and velocities.*
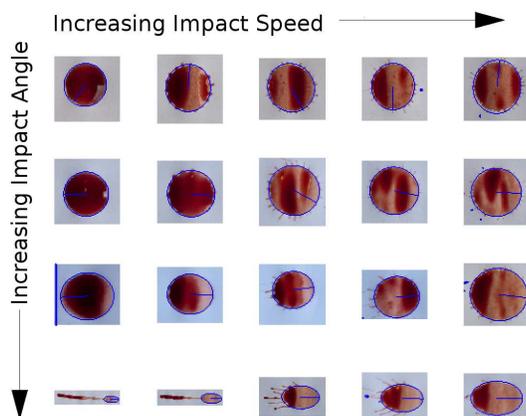
Figure 5.6: *Various computed ellipses for the blood droplet test stains.*

Discussion

The estimates resulting from this experiment are plausible. We have shown that our velocity estimation equation fits the experimental data decently as seen in Figure 5.5. We also show that our ellipse detection method works well regardless of satellite stains. However, using a threshold for the mask is the bottleneck of the process in terms of accuracy and could undergo further investigation. We can see this in Figure 5.6, the top right ellipse has a poor fit due to the mask given. All of these ellipses were calculated with the same threshold.

5.2   Validation Experiments

In this section we will describe three different experimental setups in which we validate various aspects of our method on real world scenarios. We project blood at medium to high velocities from a slingshot and paintball

gun in Sections 5.2.1 and 5.2.2 respectively. Finally, we do a low-medium velocity impact simulation by hitting a pool of blood with a hammer in Section 5.2.3.

5.2.1 Slingshot Experiment

In the slingshot experiment, we aim to see how well we can reconstruct the area of origin. We also also want to use knowledge of the ground truth area of origin to calculate the impact speeds based on the parabolic trajectory model by fixing two points of the parabola and a tangent angle at one of the points. We want to do so to fit Equation 4.15. The slingshot allows us to test a range of velocities from medium to high in BPA terms.

Procedure

For the slingshot experiment, we mounted a slingshot to a sturdy box. The payload of the slingshot was a few droplets of cow blood in a bottle cap attached to the sling. We fired this slingshot at three different speeds onto a $8\frac{1}{2}$ by 11 inch sheet of paper mounted on a vertical surface. We then moved the box so that it was the same distance a few feet from the wall as before, but at a different angle to the paper. We did this for 3 angles and 3 velocity combinations. The layout is seen in Figure 5.7.

(a) Top down view of the slingshot setup   (b) A photo of the slingshot setup

Figure 5.7: *Top down view of the setup for the Slingshot Experiment: θ is different angles fired at the surface and v is different velocities fired from the slingshot.*

Results

(a)   (b)

Figure 5.8: *Ellipse fits for slingshot experiment*

(a) Example yz reconstruction from a slingshot test

(b) Example xz reconstruction from a slingshot test

(c) Example xy reconstruction from a slingshot test

Figure 5.9: *Results from a slingshot experiment with headon direction*
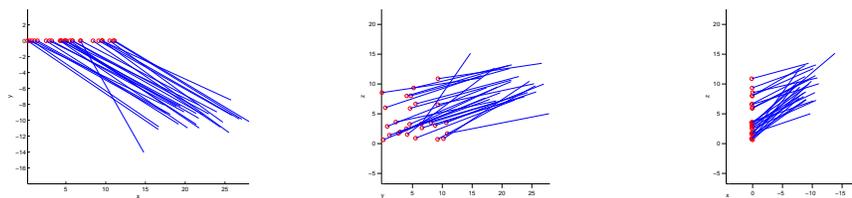
(a) Example yz reconstruction from a slingshot test

(b) Example xz reconstruction from a slingshot test

(c) Example xy reconstruction from a slingshot test

Figure 5.10: *Results from a slingshot experiment with oblique direction*

Discussion

Based on these results, we have determined that this experiment was ill-conditionned. The actual area of origin is a 4-5 feet away, however the lines converge a few inches away from the surfaces. Our target area was too small for the distance from the source. We were unable to perform everything we sought to do in this experiment however, we have determined that our ellipse fitting works exceptionally well on real world stains. Also, we have identified a failure case for previous methods as described in Figure 4.11 in Section 4.5. It should also be noted that the slingshot experiment was not

very reproducible as the slingshot was hard to pull back to the exact same position and orientation.

5.2.2   Paintball Gun Experiment

In this experiment, we attempt to accomplish the same goals as in Section 5.2.1. We increase the target area, as well as bringing the source closer to the target. We also use a more controlled device, a paintball gun, which allows us to control the velocity directly.

If we are to reconstruct the area of origin, we need the camera and scene to be calibrated properly. That is to say, we need to know accurately on the computer where the stains are in the real 3D world. Other work, such as research on scale invariant features, could prove useful to calibrate the scene without using any markers [13] and using these features matches to obtain epipolar geometry about the scene is described in [11]. However, the method described in [20] would probably serve to be more useful as it is tailored for ellipses, which we know there to be many of in our scene. We have decided to use calibration markers as we're dealing with planar surfaces.

Procedure

This is very similar to the slingshot experiment, but instead we used a paintball gun. specially manufactured paintballs were filled with cow blood and used for the payload. The gun let us regulate the air pressure, controlling the muzzle velocity. We force the paintball to hit a potato masher to break the shell off and allow the blood to continue to travel. We did the test closer to the target than in Section 5.2.1, as it could not reproduce the

correct xy area of origin estimate. We also used a large 11 by 17 sheet of paper to increase the impact area and lessen the error. In this approach we had the ground plane calibrated with the use of a calibration marker, leveled properly as well as a different marker for the impact plane. We see an illustration of the setup in Figure 5.11.
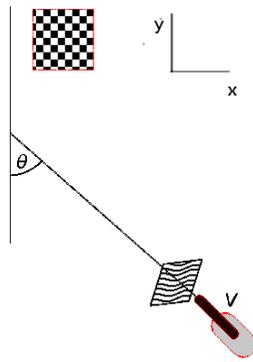
(a) Top down view of the paintball setup. The paintball is shot at a grate, where the paintball separates and blood continues to impact the surface

(b) A photo of the paintball setup

Figure 5.11: *Setup for the Paintball Experiment: v being different velocities and θ being different angles fired at the surface.*
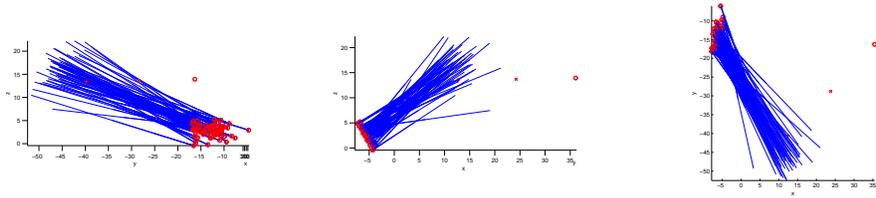
Results

(a) Example yz reconstruction from a paintball test

(b) Example xz reconstruction from a paintball test

(c) Example xy reconstruction from a paintball test

Figure 5.12: *A paintball experiment result with an oblique impact angle*
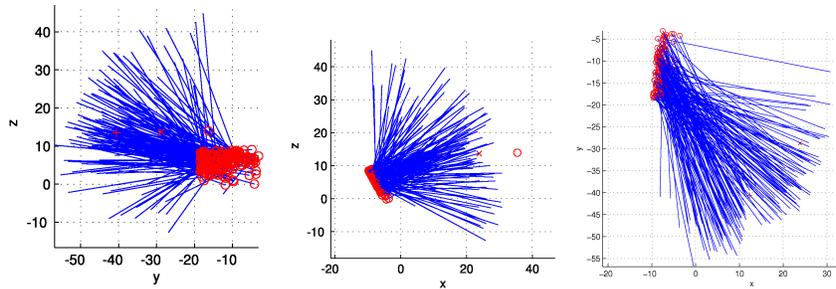
(a) yz reconstruction

(b) xz reconstruction

(c) xy reconstruction

Figure 5.13: *A paintball experiment result with a head on impact angle*

(a) Before Flips

(b) After Flips
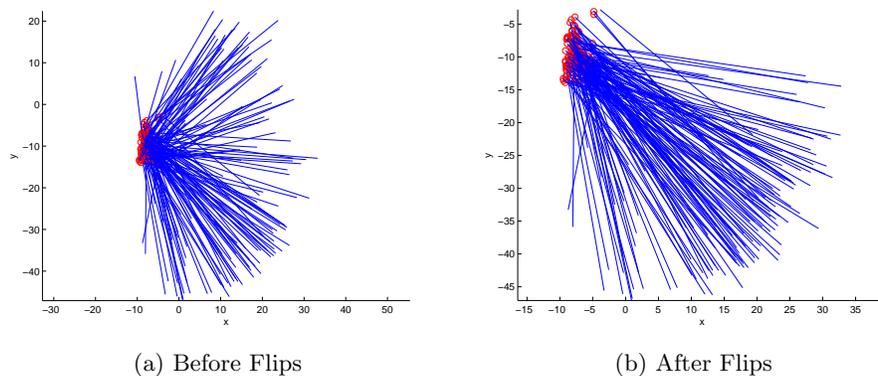
Figure 5.14: *The result of flipping ellipse major axis directions*

Discussion

Despite our efforts to better condition the experiment, it was still ill-conditionned and so neither our method nor the tangent method perform well. The lines converge up to 10 inches away, whereas the actual areas of origin is 3-5 feet away. Due to complications in the specially manufactured paintballs, where putting any glue on them to seal them caused misfires, this experiment was not very reproducible. We therefore moved on to a simpler experiment that gives us better conditionned directions and positions.

5.2.3 Hammer Experiment

We now will describe the hammer experiment. We know that by hitting a pool of cow blood with a hammer, there will be stain positions all over a target area around the pool. This ensures that the stains will come from every direction in the xy plane, thus ensuring a good estimate for the xy area of origin as described in Section 3.5. This will allow us to calculate the

error for our method and the linear method and compare results.

Procedure

In this experiment we hit a pool of blood with a hammer. We varied the height of an object for 3 different values, and put 10-20 droplets of cow blood on the object. We proceeded to hit the blood with a hammer at a force high enough to cover the test area. The test area was composed of 6 sheets of $8\frac{1}{2}$ by 11 inch sheets of paper arranged in a 2 by 3 sheet array.
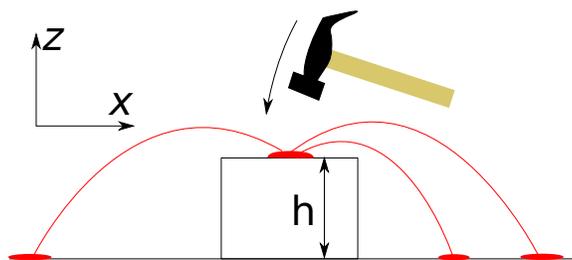
Figure 5.15: *Setup for the Hammer Experiment: h is the only parameter we vary, the height of the object. We hit the hammer with enough energy to cover the whole test area.*

Results

In Figures 5.17, 5.18, and 5.19 are the results for the test described above for heights of 1.25, 5 and 7.25 inches respectively. Each figure shows the ground truth flight paths (the red paths), the estimated linear estimate (the black paths) and our estimate (the blue paths) each of the experiments, we see that our result predicts more accurately the area of origin.

We also show the probability density functions for each region in the xz and yz cross sections. Although we may not 100% accurately reconstruct

the area of origin, it lies in the non-zero probability region of the PDF.

(a) Input Image

(b) Orthogonal Warp of Input Image
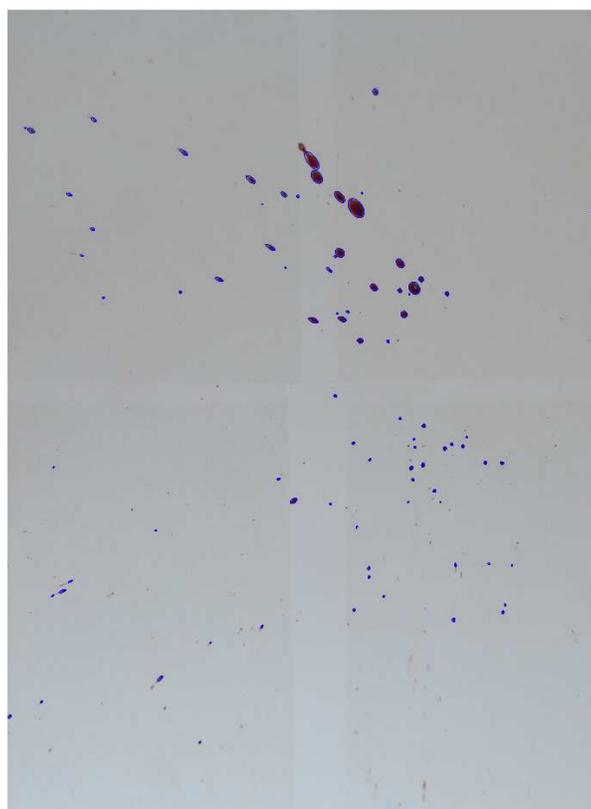
(c) Fitted Ellipses

Figure 5.16: *Different stages in our system*

(a)

(b)

(c)

(d)
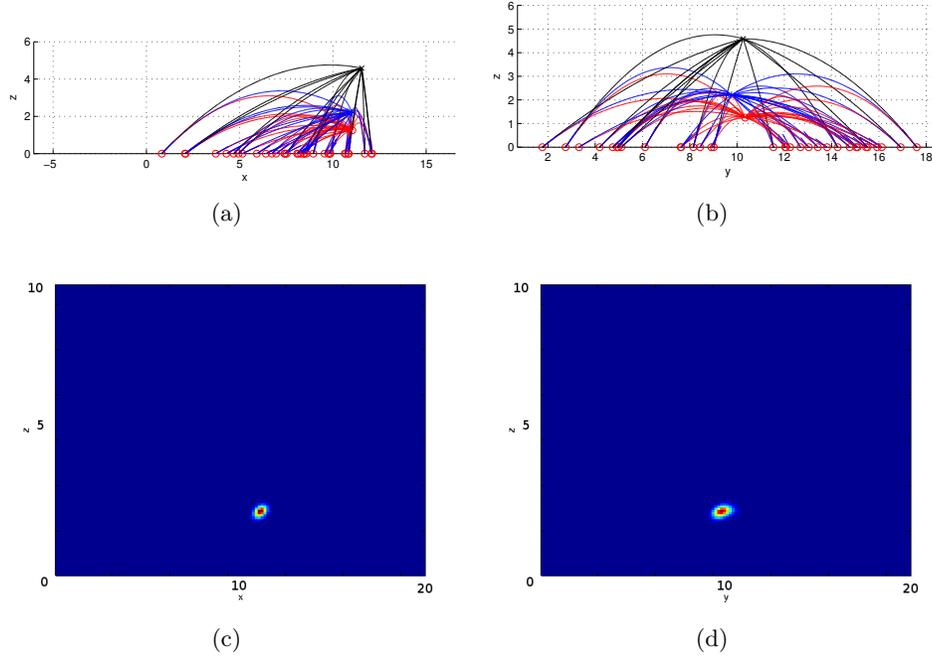
Figure 5.17: *Results for the 1.25 inch height experiment.*

Error Type	Linear Method	Probabalistic Method
xyz	3.3768	1.1165
xy	0.5116	0.5789
z	3.3378	0.9547

Table 5.1: *Error (in inches) for the 1.25 inch experiment.*
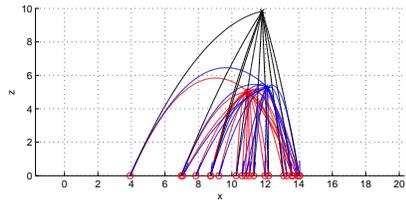
(a)
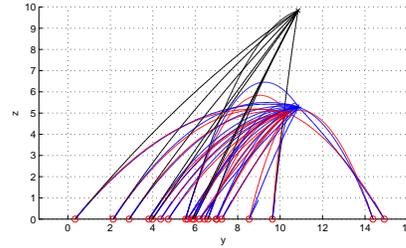
(b)
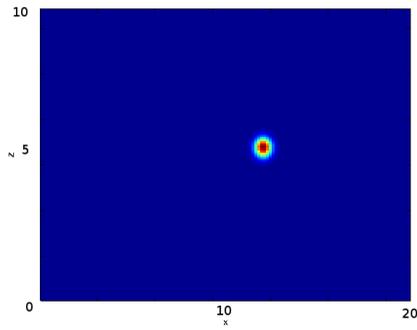
(c)
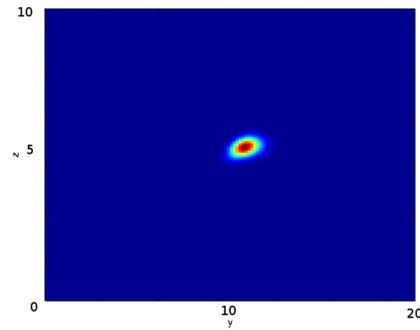
(d)

Figure 5.18: *Results for the 5 inch height experiment.*

Error Type	Linear Method	Probabalistic Method
xyz	4.9281	1.2866
xy	0.9619	1.2567
z	4.8333	0.2756

Table 5.2: *Error (in inches) for the 5 inch experiment.*

(a)

(b)

(c)

(d)
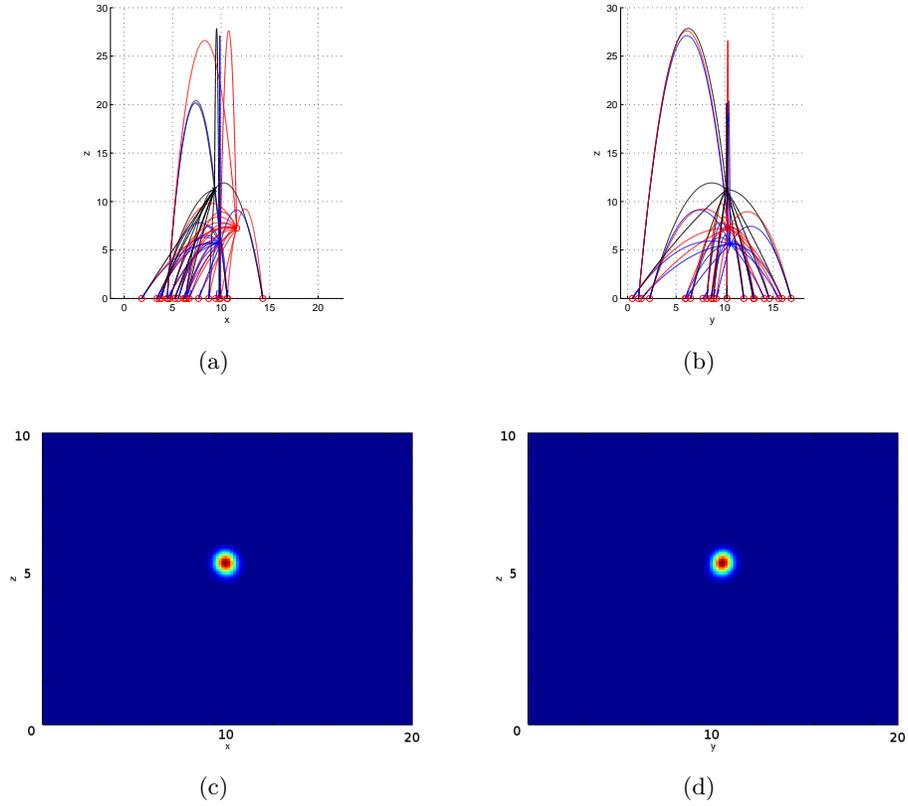
Figure 5.19: *Results for the 7.25 inch height experiment.*

Error Type	Linear Method	Probabalistic Method
xyz	4.5954	2.3687
xy	2.2929	1.6903
z	3.9826	1.6594

Table 5.3: *Error (in inches) for the 7.25 inch experiment.*
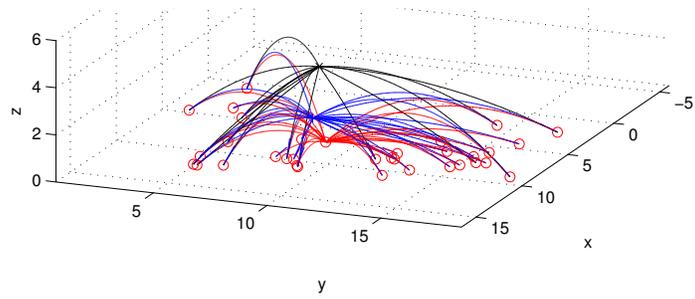
(a)
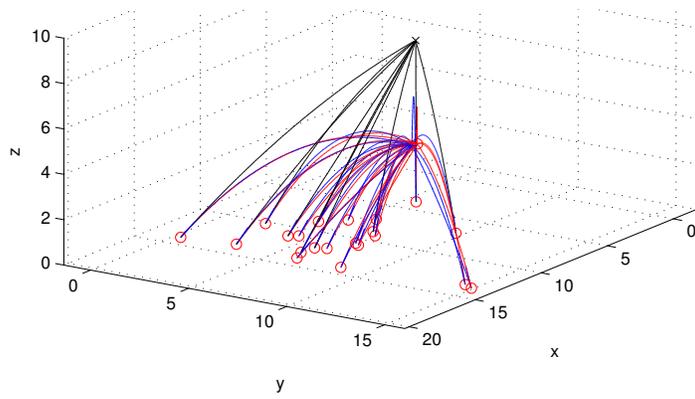
(b)
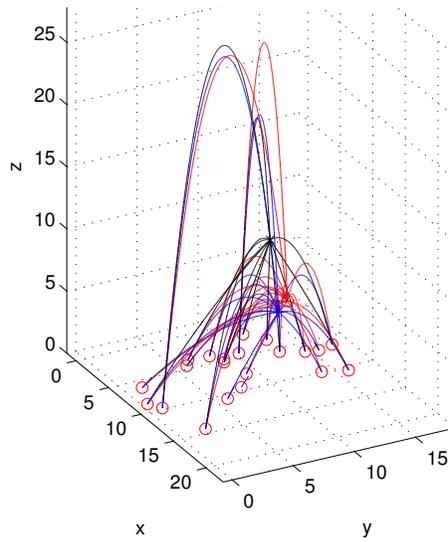
(c)

Figure 5.20: *3D views for the 3 heights.*

Discussion

We can see that we estimate the area of origin accurately. The same reconstruction parameters were used for all three methods, however different mask thresholds were chosen.

We say accurately, as the z error of our method is comparable to the xy error of the linear estimate. The xy components of the linear estimate have undergone validation and are suitably accurate as shown in [1], [5], [12].

For visualization, we plotted the estimated trajectories for all the methods using parabolic motion. This is why the linear case are not composed of lines, we use the reconstructed area of origin estimate along with the direction and position of each stain to form a parabola.

To determine the ground truth area of origin position, we marked a point of the location on the target plane. We then put the object on which to hit a blood pool with a hammer centered on this point. We also measured the height of the object. The planar location gives us the xy coordinates and the measured height gives us the z coordinate of the area of origin.

Chapter 6

Conclusions and Future Work

6.1 Discussion and Limitations

We have described a method of reconstructing the area of origin in a non-linear manner, whereas no previous works have done so. It works with the same input data as in existing methods, the stain angles and positions in 3D space.

We have also described a method to robustly calculate the impact angle given the image of a blood droplet, regardless of satellite stains. Furthermore, we have described a model to estimate blood droplet impact velocities and error in this estimation. This can be exploited into our probabalistic model as the deviation in error is an input parameter.

This approach is limited in that we must assume we can accurately obtain the velocities from the stains, or that the velocities have the same magnitude. Both have failure cases. The velocity estimation is dependent on a given stain having all the blood from the associated particle contained in it. Essentially we have a volume constraint in the stain. This is not true,

since some droplets may impact at an oblique grazing angle to the surface and rebound off further down in separate stains. It may also occur that the velocity of a particle may be so high that it splashes into other nearby stains. Both of these types of scenarios can be seen in Figure 6.1a.

Also, we decided to use a simplified Gaussian for the particle probability equation 3.50, where the covariance matrix is a multiple of the identity matrix. In reality, to accurately estimate the probability, a circular warp of the Gaussian would be necessary.

(a) Volume loss from either rebounding after hitting an oblique angle or having a high enough velocity to introduce secondary splatters

(b) An extreme example where not all the velocities are similar
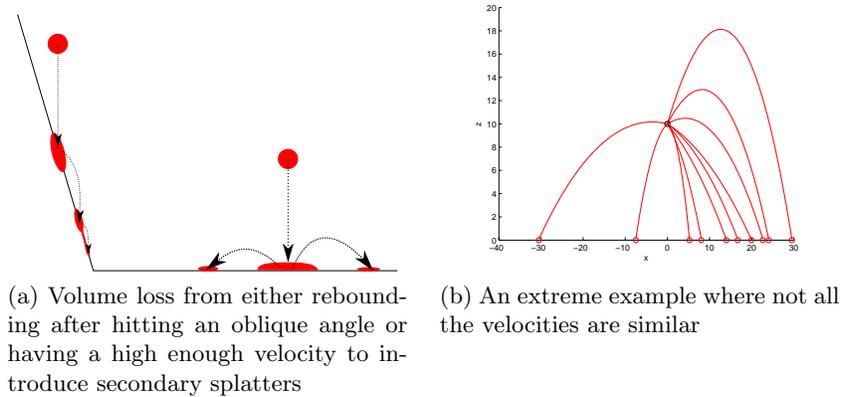
Figure 6.1: *Different failure cases for our method*

The other failure case, that of assuming all velocities are equal or similar, is also a falsifiable assumption with extreme cases. Suppose many of the trajectories are downward pointing from the initial area of origin with no vertical velocity component, with one that is upward pointing as in Figure 6.1b. The upward pointing trajectory will have a much higher velocity than the downward pointing ones. This is an extreme case and would seem unnatural to have this kind of behaviour in a practical setting.

Another potential limitation is as with many computer vision approaches, there are a few parameters to tune. For the reconstruction, we have to tune the δ_θ and δ_v values. For the stain segmentation, manual thresholds were used to mask the stains, even on a white background. There are also parameters used in pruning the outliers of the stains using RANSAC.

6.2   Future Work

More controlled tests are needed to accurately estimate velocity based on stain density and angle, as described in Section 4.7. Also, more thorough validation of the whole method would also be necessary before using the presented algorithm in the field. In addition to accurately estimating the velocities, we could also learn δ_θ and δ_v which are essentially the standard deviation in the angle and speed estimates by comparing it with ground truth data.

It is not uncommon for forensic scientists to perform mock tests for a given scene to see if the result is plausible. If one were to want to see how blood travels through the air as the event occurs, there are various capture techniques to track this data. Corpetti et al. [6] present an optical flow technique which specifically tracks fluids using the divergence and curl properties of a fluid over time. There is also a method to capture fluids in motion using a stereoscopic camera with dyed water [17]. An additional mock test could be accurately simulating the blood with a fluid dynamics simulation. Simulating fluids is no easy task, and is an open area of research. R. Bridson's book on the subject [3] describes the approach and difficulties.

79

This is appealing as it is less messy than doing tests with actual blood and more reproducible.

Right now, we have only tested on planar data to ease with calibration, although with a more rigorous capture setup it could be possible to acquire a 3d textured model of a scene. To estimate the blood droplet angles from arbitrary surfaces would be very beneficial, by maybe using planar projections of localized mesh patches.

So far we have assumed one splatter event coming from a point source. If there were multiple events in the scene, or a blood trail, a new approach would be needed. Instead of calculating the probabilities by multiplication (an AND operation), we could use addition (an OR operation) and cluster using this data.

Another assumption we make is that there are only two forces acting upon an object, drag and gravity. However there may be a vector field such as for wind affecting the object as it passes through it. This may affect the outcome of a crime scene or another particle scene such as with shrapnel from an explosion.

Bibliography

[1] T. Bevel and R.M. Gardner. *Bloodstain pattern analysis: with an introduction to crime scene reconstruction*. CRC Press, 2008.

[2] K. Boonkhong, M. Karnjanadecha, and P. Aiyarak. Impact angle analysis of bloodstains using a simple image processing technique. 2010.

[3] R. Bridson. *Fluid simulation for computer graphics*. AK Peters Ltd, 2008.

[4] A. Carter. Backtrack images. http://www.physics.carleton.ca/carter/.

[5] AL Carter, J. Forsythe-Erman, V. Hawkes, M. Illes, P. Laturnus, G. Lefebvre, C. Stewart, and B. Yamashita. Validation of the Back-Track Suite of Programs for Bloodstain Pattern Analysis. *Journal of Forensic Identification*, 56(2):242, 2006.

[6] T. Corpetti, É. Mémin, P. Perez, and R.I. IRISA. Dense estimation of fluid flows. *IEEE Transactions on pattern analysis and machine intelligence*, 24(3):365–380, 2002.

[7] Deltasphere. Deltasphere 3000 - 3d scene digitizer. http://www.deltasphere.com/DeltaSphere-3000.htm.

[8] WG Eckert and S. James. Interpretation of bloodstain evidence at crime scenes. *No.: ISBN 0-444-01463-2*, page 367, 1989.

[9] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[10] A. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):476–480, 1999.

[11] R. Hartley and A. Zisserman. *Multiple view geometry*. Cambridge university press Cambridge, UK, 2000.

[12] MB Illes, AL Carter, PL Laturnus, and AB Yamashita. Use of the Backtrack Computer Program for Bloodstain Pattern Analysis of Stains From Downward-Moving Drops. *Canadian Society of Forensic Science Journal*, 38(4):6, 2005.

[13] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[14] A. Murta, S. Gibson, TLJ Howard, RJ Hubbold, and AJ West. Modelling and rendering for scene of crime reconstruction: A case study. In *Proceedings Eurographics UK*, pages 169–173, 1998.

[15] N. Rogers. Hematocrit–Implications for Bloodstain Pattern Analysis. 2009.

[16] AR Shen, GJ Brostow, and R. Cipolla. Toward Automatic Blood Spatter Analysis in Crime Scenes. In *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, pages 378–383, 2006.

[17] H. Wang, M. Liao, Q. Zhang, R. Yang, and G. Turk. Physically guided liquid surface modeling from videos. *ACM Transactions on Graphics (TOG)*, 28(3):90, 2009.

[18] A.Y. Wonder. *Blood dynamics*. Academic Pr, 2001.

[19] A.Y. Wonder. *Bloodstain pattern evidence: objective approaches and case applications*. Elsevier/Academic Press, 2007.

[20] J. Wright, A. Wagner, S. Rao, and Y. Ma. Homography from coplanar ellipses with application to forensic blood splatter reconstruction. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 2006.

[21] J. Yao, N. Kharma, and P. Grogono. A multi-population genetic algorithm for robust and fast ellipse detection. *Pattern Analysis & Applications*, 8(1):149–162, 2005.