UNIVERSITY OF NEW MEXICO

HONORS THESIS

The Effect of CS_2 on Diffusion and Light Yield in Optical Time Projection Chambers

---

Author:
Austin Vaitkus

Supervisor:
Dr. Dinesh Loomba

A thesis submitted in fulfillment of the requirements for the Departmental Honors Program

in the

Department of Physics and Astronomy

April 24, 2017

University of New Mexico

Abstract

Dr. Dinesh Loomba
Department of Physics and Astronomy

Departmental Honors

The Effect of CS_2 on Diffusion and Light Yield in Optical Time Projection Chambers

by Austin Vaitkus

Dark matter presents itself as one of the greatest mysteries to modern physics. The Weakly Interacting Massive Particle (WIMP) has been deemed a strong candidate for the makeup of this mysterious substance. Direct Detection of WIMPs involve the detection of nuclear recoils in Time Projection Chambers (TPCs) to determine the direction of incoming WIMP flux. TPCs fall victim to electron diffusion, making directionality difficult to determine at low recoil energies. In this thesis, we make use of the effect of CS_2 to reduce diffusion in the detector. We were able to reduce the width of alpha particle tracks by 75% while maintaining high light yield. From this research we determined the ideal ratio of CS_2 to CF_4 in the vessel to be about 3:150 Torr, or 2%. The ability for CS_2 to diminish diffusion is extremely powerful; the directionality of lower energy recoils can be better determined. With such a strong tool, we are one step closer to the direct detection of dark matter.

Acknowledgements

I would like to thank Dr. Dinesh Loomba for his strong mentorship throughout the process of my work. His passion for the subject, his patience, and his guidance enabled my success throughout the year. I am very grateful to have had the opportunity to participate in such fascinating research.

I would also like to the Undergraduate Committee for selecting my research for the Raeburn Fund for Undergraduate Research. Similarly, I would like to thank the Undergraduate Committee for the review my thesis.

Contents

Abstract i

Acknowledgements ii

1 Introduction 1
- 1.1 A Brief Introduction to Dark Matter . . . 1
- 1.2 The WIMP and Motivation for Directional Dark Matter Detection . . . 2
- 1.3 Our Detector . . . 2
- 1.4 Diffusion in the Detector . . . 4

2 Alpha Tracks 6
- 2.1 Why Use α Tracks . . . 6
- 2.2 The Algorithm . . . 7
  - 2.2.1 Initial Identification . . . 7
  - 2.2.2 Straight Segments and the Hough Transform . . . 8
  - 2.2.3 Rotation and Width Calculation . . . 10
- 2.3 Results for Diffusion from α Widths . . . 11

3 ^{55}Fe and X-Ray Tracks 14
- 3.1 Why Use ^{55}Fe . . . 14
- 3.2 The Algorithm . . . 14
- 3.3 Results for Light Loss with CS_2 . . . 15
- 3.4 Summary: the Ideal CS_2 Concentration . . . 17

4 Conclusions and Further Research 18

A Mathematics of the Hough Transform 19

B Matlab Codes 21
- B.1 Calibration . . . 21
- B.2 Alpha Codes . . . 24
- B.3 ^{55}Fe Codes . . . 34

Bibliography 37

Chapter 1

Introduction

1.1 A Brief Introduction to Dark Matter

The idea that there exists a type of matter completely invisible to the human eye has long been considered a fact of the Universe. Nineteenth century astronomers speculated the existence of "dark nebulae" and "dark stars", both of which carried mass and took up space but lacked the ability to emit light (Bertone and Hooper, 2016). This idea carried over into the work of Lord Kelvin and Jan Oort, who began making measurements of stellar velocity dispersions within the Milky Way in the early twentieth century. They concluded that the total mass of some arbitrary "dark matter", while not inconsequential, was probably much less than that of visible matter.

Dark matter did not begin building traction until the works of Fritz Zwicky in 1933. Zwicky, who studied the redshift of various galactic clusters, stumbled upon a kinematic phenomenon in the Coma cluster: the velocities of several galaxies were largely dissimilar, sometime differing by 1000 km/s (van den Bergh, 1999). These unexpectedly large velocity dispersions were not possible without the existence of especially high masses within the cluster. Zwicky set off to determine the observable mass of the system by multiplying the average mass of a galaxy (then estimated to be about 10^6 $M_\odot$) with the number of observed galaxies (Bertone and Hooper, 2016). His estimation indicated that the velocity dispersion within this particular cluster should not reach much higher than 80 km/s. The observational evidence that disagreed with theoretical predictions led to Zwicky's conclusion that, in contrast with Kelvin and Oort, dark matter is present in the Universe in much greater amounts than luminous matter (Zwicky, 1933).

Fritz Zwicky is often described as the pioneer of dark matter, but some of dark matter's most compelling evidence came in the form of galactic rotation curves. In 1970, Vera Rubin and Kent Ford made measurements of, with high degrees of certainty, the mass and rotational velocity of the Andromeda Galaxy (or M31) up to a radius of 24 kpc (Rubin and Ford, 1970). This led to a surprising result: the mass density due to luminous matter decreased at high radii, but the velocity did not; there is likely some additional form of matter in abundance that was contributing to the total gravitational potential, and thereby rotational velocity, of the galaxy.

Cluster velocity dispersions and galactic rotation curves confirmed that there was *a lot* of dark matter in the Universe, but until the era of precision cosmology the specific amount was poorly constrained. This changed in 2010. After 9 years of taking data, the Wilkinson Microwave Anisotropy Probe (WMAP) satellite-based experiment determined that the geometry of the Universe is flat (Komatsu, 2014). This fact tells us that the total energy density of the Universe is equal to its critical value, but also gives us an accurate value of the baryonic and dark matter densities in the Universe. In fact, it was determined that baryonic matter makes up about 4.6 percent of the contents of the Universe, while dark matter makes up about 24 percent. The remainder consists of 71.6 percent dark energy (Hinshaw et al., 2013).

Numerous theories exist today regarding the actual make-up of dark matter, however a few of its specific qualities are largely widespread: it should have mass, be neutrally charged, and lack any capability to interact electromagnetically. One of the leading candidates for dark matter is the Weakly Interacting Massive Particle, or, the WIMP.

1.2 The WIMP and Motivation for Directional Dark Matter Detection

As the name suggests, the WIMP is theorized to couple with ordinary matter through the weak interaction. If true, dark matter can be observed via its rare interactions with atoms in a detector.

There are three ways that have been proposed to detect the WIMP (Mount, 2017). First, using particle colliders to produce WIMPS at high energies. Here, WIMPS would show themselves as missing energy and momenta from the collision. Second, through Indirect Detection (ID) methods that look for WIMP-WIMP annihilation in locations in which dark matter tends to concentrate, such as the galactic center and in the Sun. These ID techniques assume the WIMP is its own antiparticle (a Majorana particle) and will have identifiable secondary particles, such as photons and neutrinos, produced in their annihilation. Finally, Direct Detection (DD) techniques search for the interaction between WIMPS and baryonic matter. This weak interaction produces traceable recoils in the atomic nuclei of a detector.

Direction detection techniques provide one very specific advantage over the others: nuclear recoils *should* have a directional signature attached to them. The Sun travels through the Milky Way with a velocity of about 230 km/s in the direction of the constellation Cygnus (Lewin and Smith, 1996). Galaxy formation theories and simulations indicate that dark matter does not co-rotate with the stars and gas in the galaxy. In this scenario the solar motion through the dark matter halo will result in a dark matter "wind" from the direction of Cygnus. Thus, if we can detect the tracks of nuclear recoils resulting from WIMP interactions, they should be directed opposite to Cygnus. Such a detection would provide an unambiguous signature for a WIMP discovery.

The concept of a directional WIMP wind is the motivation behind several DD experiments. In particular, the Directional Recoil Identification From Tracks (DRIFT) experiment makes use of a Time Projection Chamber (TPC) to search for and study the directional signature resulting from dark matter-based nuclear recoils (Daw, 2012). DRIFT is one of a number of directional dark matter experiments that use the gaseous TPC technology. The work described in this thesis also uses a TPC but, unlike DRIFT, it uses a scintillating gas (CF_4) and a CCD readout to detect recoil tracks.

1.3 Our Detector

A TPC is able to make 3D reconstructions of nuclear recoil tracks, which provides the ability to determine the directionality of any given recoil. To make measurements of

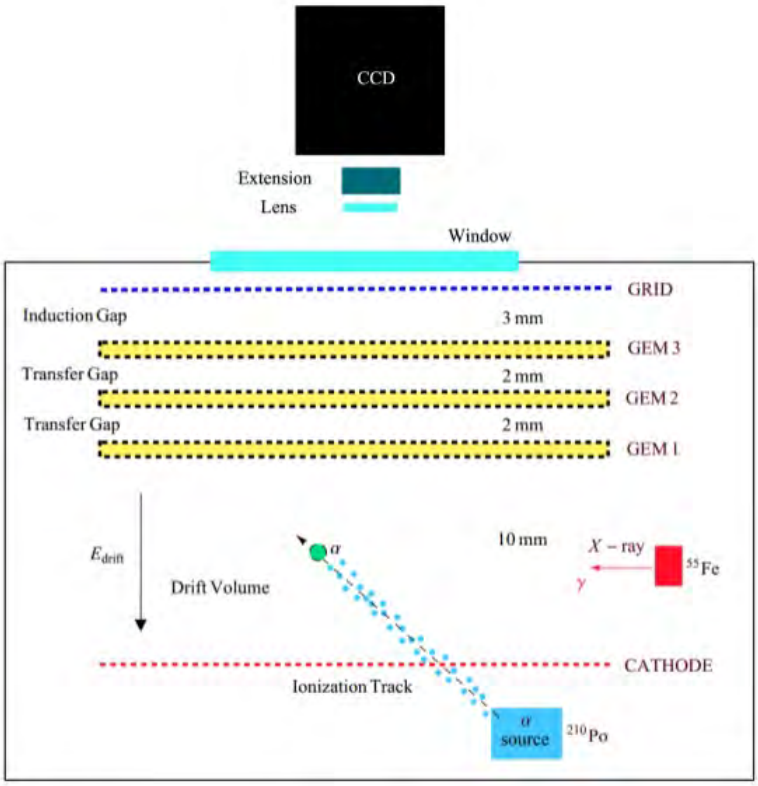

FIGURE 1.1: A schematic diagram of the prototype TPC. Not to scale. (Loomba, 2013)

nuclear recoils of various origin, we have made use of a prototype TPC developed by Nguyen Phan, a former graduate student at the University of New Mexico (Phan et al., 2016).

This detector resides in a cylindrical aluminum vacuum vessel 16 cm tall and 29 cm in diameter. The vessel was filled with 150 Torr of carbon tetrafluoride (CF_4). The detector, a representation of which can be seen in Figure 1.1, contains a cathode mesh, an anode wire grid, and three Gas Electron Multipliers (GEMs). The GEMs, which were designed and produced at CERN, are 7 cm x 7 cm and are made of 50 μm thick kapton foil that has 5 microns of copper cladding on both sides. Each GEM is covered in 50 μm holes, with a pitch of 140 μm. Above the detector is a charged-coupled device (CCD) camera with a 1024 $\times$ 1024 pixel sensor array. Attached to the CCD is a 58 mm f/1.2 lens on a 20 mm extension tube. A Polonium-210 (^{210}Po) and Iron-55 (^{55}Fe) source were placed inside the detector, each of which could be turned on or off on demand.

To fully understand the importance of the detector's components, it is first important to understand the function of a TPC. As an energetic particle enters the detector, it interacts and ionizes atoms along a track. The electrons emitted in this interaction are carried upwards towards the GEMs by an electric field produced between the cathode mesh and GEMs. Upon reaching the GEMs, the electrons undergo a massive stage of multiplication. The electrons are guided into the small holes in the GEMs, where an especially strong electric field ionizes the gas in the detector even further. This process, known as electron avalanche, occurs in each of the three GEMs. The term "gas gain" is used to describe the level of electron multiplication that occurs during the avalanche. In our detector we are able to reach an effective gas gain of over 10^5, meaning that for every one electron that reaches the first GEM, over 10,000 electrons leave the third.

With nuclear recoils of higher energy, the directionality is easy to discern: meaning the tracks are well resolved, clearly indicating the recoil direction. However, as the energy decreases, these tracks become short and diffusion limited, so directionality is lost; they appear as blobs rather than lines (see Figure 1.2).

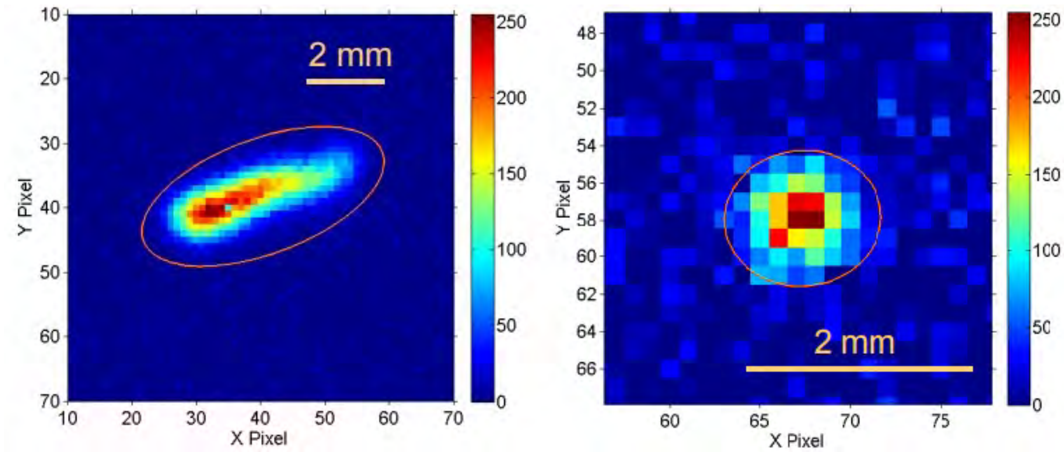

FIGURE 1.2: **Left**: a nuclear recoil track with a recoil energy of 302.4 keV. Here, directionality can be determined from the linear shape of the track. **Right**: a nuclear recoil track of 10 keV. The energy of this track is too low to easily determine directionality. (Loomba, 2013)

The gas in the chamber, CF_4, was originally chosen for preliminary testing due to its high capacity for scintillation. During the avalanche both electrons and scintillation photons are produced, which allows us to optically image the track with a lens and CCD camera. However, ionization that forms the track diffuses as it drifts along the electric field towards the anode. In the next section we discuss diffusion and how it can be minimized with appropriate gas mixtures. This is one of the key subjects of this thesis.

1.4 Diffusion in the Detector

One solution to the electron diffusion problem is to add an electronegative gas, such as CS_2, into the detector (Martoff et al., 2000). CS_2 has the benefit of having high electron affinity (Phan et al., 2016). CS_2 molecules capture the electrons initially produced by the recoiling particle (atom or electron), forming negative ions that drift towards the GEMs. The large mass (in comparison to electrons) of these molecules, allows for much lower drift velocity and also results in much lower diffusion, which occurs in the thermal regime. However, CS_2 lacks CF_4's strong scintillating properties.

Phan showed (Phan et al., 2016) that adding small quantities of CS2 to CF4 could provide the low diffusion benefits of CS2 without a large loss of scintillation. The goal of this thesis is to quantify this by measuring the reduction in both diffusion and scintillation light as a function of the partial pressure of CS2 in 150 Torr of CF4. The experiments, described below, will add CS2 in increments of about 0.5 Torr and make measurements of alpha particle and low energy 55Fe tracks to measure diffusion and scintillation light. We will show that we can achieve low diffusion, approaching the minimum thermal limit, without a too large loss of light.

Chapter 2

Alpha Tracks

2.1 Why Use α Tracks

The ^{210}Po source decays to emit an alpha (α) particle. We have chosen α tracks to measure diffusion and its reduction as a function of the CS_2 fraction in the dominant CF_4 gas. α particles rarely generate nuclear recoils in the detector. Instead they travel in mostly straight paths, strongly ionizing the gas and therby leaving a trail of ion-electron pairs. The tracks produced are long, straight, and bright (see 2.1). The intrinsic track should be very narrow, which broadens considerably due to diffusion when the electrons drift to the GEMs. Thus, measuring this width provides an excellent estimate of diffusion.

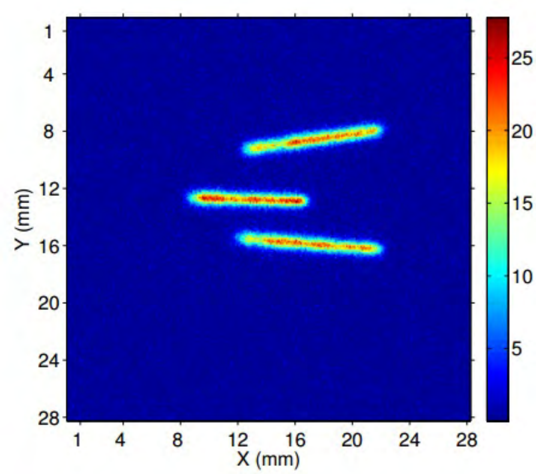

FIGURE 2.1: Three distinct α tracks observed in the prototype TPC. The color chart shows the high signal-to-noise of the particles. Here the qualities of being long, straight, and bright are apparent.

One key trait of α tracks is their uniform widths. Lengths vary depending on a variety of variables: The ^{210}Po source emits α particles in all directions, therefore we use a colimator to narrow the angular spread. Nevertheless, αs will have a distribution in direction and lengths as observed in our detector. Similarly, the loss of α particle energy in the detector is not constant due to energy loss fluctuations and variations in direction; particles with higher initial energy loss will be produce shorter tracks than those with smaller initial loss. Despite this, track widths remain fairly consistent.

This fact can be manipulated to quantify diffusion as a function of CS_2 fraction. Our procedure was to add CS_2 in increments of about 0.5 Torr to about 150 Torr of CF_4.

Measurements of diffusion with α tracks and light output with ^{55}Fe x-rays (next Section) were undertaken for each gas mixture. This will also allow us to see the moment of diminishing returns in the detector. We suspect that, at a certain ratio of CS_2 and CF_4, the low diffusion advantage of CS_2 will no longer increase with any addition of the gas.

Widths of several hundreds of tracks at each CS_2 increment will need to be acquired to achieve statistically significant results. Similarly, widths must not be calculated by eye as this can lead to bias and an overall incorrect result. The following section describes an algorithm that can correctly calculate a large volume of track widths without any human bias.

2.2 The Algorithm

2.2.1 Initial Identification

I have developed an algorithm that correctly and efficiently analyzes a large data set of α tracks. This algorithm isolates particle tracks and measures their widths, the process of which is described in detail in this section.

Recall that the CCD camera is comprised of a 1024 x 1024 pixel sensor array, therefore a standard 1x1 pixel binning image will similarly be made up of 1024 x 1024 pixels. To improve signal-to-noise in the image, the image can be binned with a higher pixel number. "Binning" is the process of combining the charge of adjacent pixels, improving signal-to-noise but reducing resolution. A 4x4 binning image implies every square of 16 pixels is reduced to 1 pixel in the final image. Similarly, in an image of 1x1 binning, each pixel in the image corresponds to 29 microns of real space. Therefore each pixel in a 4x4 binned image corresponds to 116 microns.

We use 2x2 binned images to maintain high resolution while gaining the additional benefit of improved signal-to-noise. All images were imaged with a 1 second exposure time. Figure 2.2 shows a 2x2 image of α tracks after the image has been calibrated (see Appendix B.1).

Track identification is a key aspect to the algorithm. First, every pixel value is compared to the average value of the background. Pixels whose value are 3σ above the mean are chosen and exported to a second image (see Figure 2.3). This second, binary image is composed entirely of 1s and 0s: high intensity pixels and the background, respectively.

It is important to note that the binary image contains all high intensity pixels of the original image. Cosmic rays often create high intensity noise within CCDs. Free streaming cosmic particles pass directly through the silicon detector of the CCD, producing singular bright pixels within an image. Charge from these bright spots can spill over into adjacent pixels in the CCD, creating small trackless regions of unusually high energy (Howell, 2000). As such, the algorithm must be able to search for isolated, high intensity pixels due to cosmic rays.

The algorithm makes use of the Matlab function `bwconncomp`, which selects and records every group of adjacent 1s in the binary image. After looking at the total pixel number within each connected region, the algorithm cuts any "track" of less than a specific size. Once the actual tracks have been determined we can call upon each individual track stored by `bwconncomp`, allowing us to analyze multiple tracks per image. The algorithm cycles through every legitimate track, carrying out individual analysis for each one.

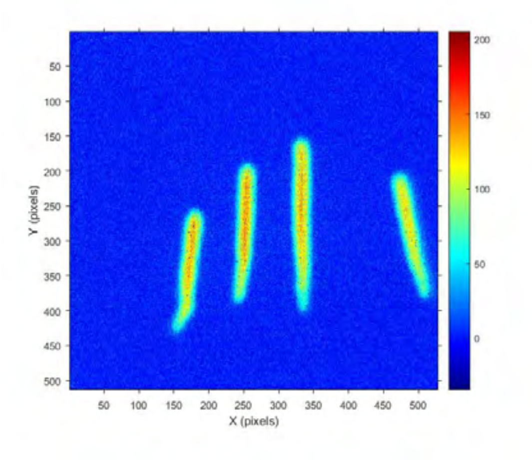

FIGURE 2.2: A 2x2 binned image with 4 distinct α tracks. The tracks stand out clearly from the background noise.

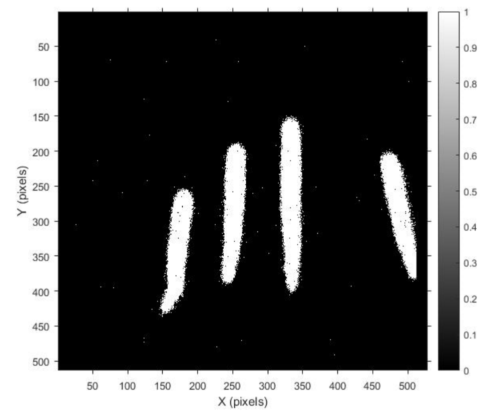

FIGURE 2.3: A binary replication of Figure 2.2. Here, all pixels 3σ above the background are given a value of 1, while others remain at 0.

2.2.2 Straight Segments and the Hough Transform

To obtain a significant measurement of the width of every α track, the algorithm must be able to determine the width over a straight segment of the track. This, however, raises the additional issue of requiring the track to be completely straight over the selected track segment (see Figure 2.4). The solution to this problem is found in the manipulation of the Hough Transform: an image analysis technique for detecting lines in pictures. A detailed explanation of the Hough Transform can be found in Appendix A

After selecting an individual track (see Figure 2.4), the algorithm bins the new binary image by a factor of 6 in the x direction. This post-imaging binning combines the pixel value of every 6 column-adjacent pixels. After following this process for every row, we are left with a track of equal length but with a width of only 4 or 5 pixels (see Figure 2.5).

Once the image has been binned in this way, the Hough Transform can be applied to find the two longest line segments detected in the track. The two detected lines follow the two longest portions of the track: the straight top segment, and the straight bottom segment that follows the curve. The transform outputs the x and y coordinates of the start and endpoints of each line. With these coordinates we can determine the equation of each line, and therefore find the point where the two lines intersect (see Figure 2.6).

The intersection coordinates of the two Hough lines could not be more valuable. The curvature of the track begins just below the y value of this point. We can carry the y value back to the original image and set all pixel values below this point to 0, leaving us with a completely straight segment of track (see Figure 2.7)

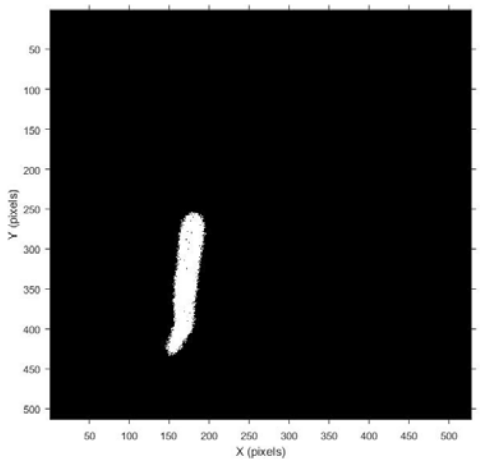

FIGURE 2.4: A single α track from Figure 2.2 in its binary representation. A curve is easily visible at its lowest point.

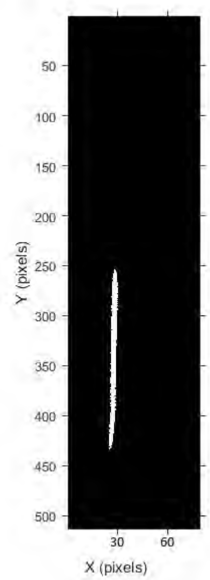

FIGURE 2.5: The same track with 6 times X-binning. The curve is still observable at the base of the track.

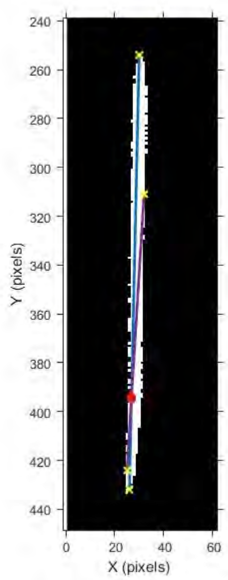

FIGURE 2.6: The Hough lines and their intersection point (in red).

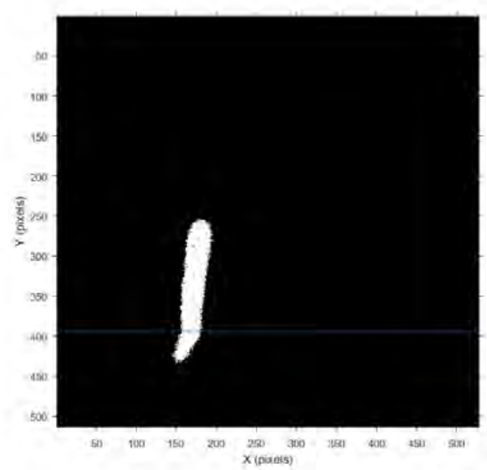

FIGURE 2.7: The original individual binary track with a line drawn over the y location found in the Hough Transform. All pixel values below this line will be set to 0.

2.2.3 Rotation and Width Calculation

Once a straight segment has been selected, the algorithm can begin to calculate the width of the track. To do this, the original image (Fig. 2.2) is rotated so the track is aligned with the y-axis. Image rotation is carried out using Matlab's `imrotate` function, which requires the angle by which the image should be rotated. To determine this angle, the algorithm finds the center (in (x,y) coordinates) of the top and bottom of the track. The total distance, as well as the difference in x-values, can then be calculated. Using basic trigonometry in conjunction with this information yields the angle needed to align the track with the y-axis. A better rotation method is the Radon Transform, which will be used in future iterations of this work (Kolouri, Park, and Rohde, 2016).

Upon rotation of the image, pixel values surrounding the track are removed, isolating the pixels belonging to the track (see Figure 2.8). With the track correctly rotated, the algorithm is ready to calculate its width. The pixels in the track are projected onto the x-axis. Pixels in every column are added so that we are left with a one dimensional vector of pixel intensities. The algorithm then plots this data against the x-axis. Due to the diffusion of the α tracks, the projection has a high intensity in the center that tapers off towards its edge; the plot takes the shape of a Gaussian distribution. The algorithm fits a Gaussian curve to this data (see Figure 2.9) and extracts the standard deviation (σ). Using σ, the full width at half maximum (FWHM) of the distribution is calculated. It is the FWHM of the Gaussian distribution that we use to quantify the diffusion of the α particle tracks.

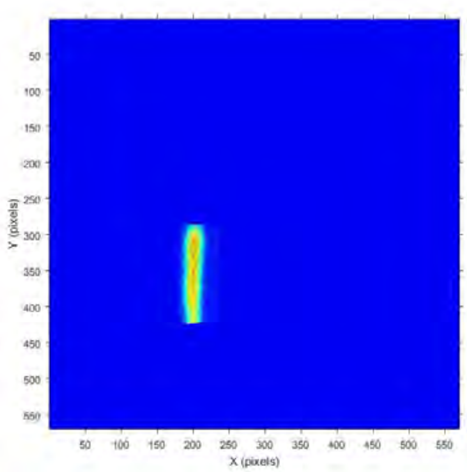

FIGURE 2.8: The straight segment of the track, rotated to be aligned with the y axis. All other pixel data has been removed.

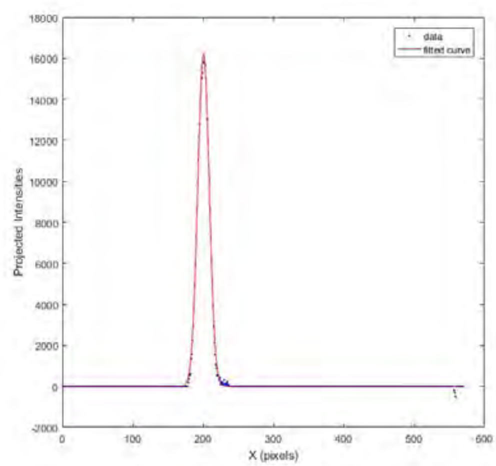

FIGURE 2.9: Gaussian distribution fit to the projected intensity. The FWHM extracted from this fit is what we use to characterize diffusion.

This entire process is repeated until every α track in every image has been analyzed, with the FWHM calculated for each track. The algorithm can be found in Appendix B.2.

2.3 Results for Diffusion from α Widths

Prior to any addition of CS_2, the algorithm ran through 300 images of α tracks allowing for the calculation of roughly 300 FWHMs (some images had 3 or 4 tracks, others had none). The resulting distribution is seen in Figure 2.10, whose median was calculated to be $FWHM_0 = 20.16$ pixels. One pixel corresponds to 29 microns, implying $FWHM_0 = 1169.3$ microns.

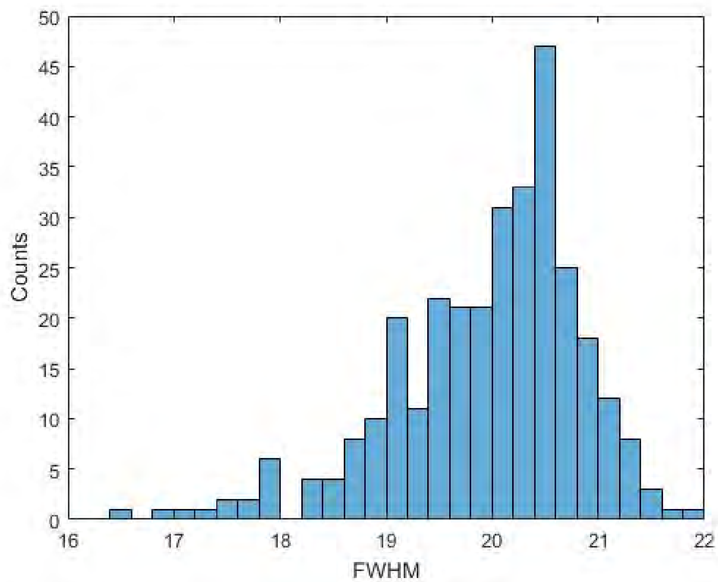

FIGURE 2.10: The FWHM distribution for 2x2 binning with pure CF_4 (ie. no CS_2.)

After initial calculations we incrementally added CS_2 into the prototype TPC. The effect of the CS_2 was immediate and very apparent; track width decreased immensely (see Figure 2.11 and Figure 2.12), even with just 0.5 Torr of CS_2. It is important to note that the pressure withing the detector increased as CS_2 was added, meaning no CF_4 was removed to maintain constant pressure.

This procedure was repeated 10 additional times, each with increasing increments of CS2 (see Table 2.1). The α widths, and therefore diffusion, no longer appeared to decrease after about 3.0 Torr of CS_2. However, we cannot rule out that this is due to pixelization or the GEM hole pitch. Decreased width is limited by the resolution of the CCD. With 2x2 binning, no track can be observed to be smaller than 2 pixels; there is a ”binning floor” in which the CCD camera interferes with our ability to observe increasing CS_2 benefit. This issue can temporarily be solved by imaging with 1x1 binning to improve resolution, which will be attempted in further iterations of this work.

It is important to note that the process of adding precise quantities CS_2 into the vessel was not without error. CS_2 can be absorbed by plastic components in the detector. As such, additional CS_2 was added to ensure the final outcome matched the increment values stated in the table above.

Despite the restraints due to absorption and resolution, one thing is for certain: CS_2 is clearly effective for reducing diffusion in TPCs. However, as stated earlier it also

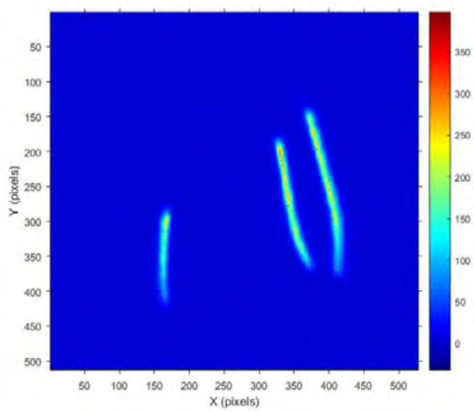

FIGURE 2.11: 2x2 binning image of α tracks in a mixture of 150 Torr CF_4 and 0.5 Torr CS_2. Track width is drastically reduced compared to 2.2.

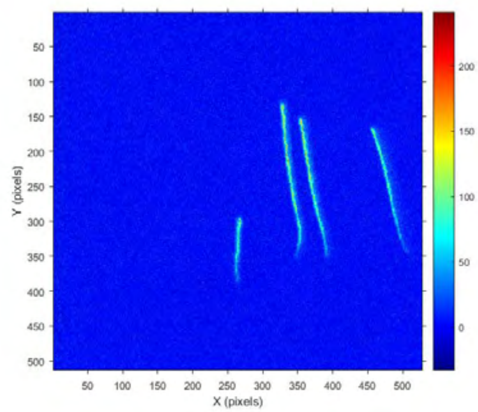

FIGURE 2.12: α track in mixture of 150 Torr CF_4 and 5.9 Torr CS_2. Here, tracks are even thinner. Improved resolution will likely improve results.

Torr of CS_2	Median FWHM (μm)	σ
0	1169.3	497.6
0.5	612.48	260.6
1.2	407.74	173.5
1.8	374.1	159.2
2.3	356.7	151.8
2.9	338.14	143.9
3.5	329.44	140.2
4.1	323.64	137.7
4.7	326.54	140
5.4	323.06	137.5
5.9	328.28	139.7

TABLE 2.1: FWHM and σ of α particle tracks calculated using 11 increments of CS_2.

lowers the scintillation light produced in CF_4. To find the ideal ratio of CS_2 to CF_4 it is now important to determine the scintillation light yield for each increment of added CS_2 using the Iron-55 (^{55}Fe) calibration source, which produces 5.9 keV x-rays. The electrons produced in the conversion of these x-rays in the gas leave short tracks that are detected similarly to the alpha tracks described above. A spectrum of intensities of ^{55}Fe tracks is used to characterize the light loss due to CS_2, as described in the next chapter.

Chapter 3

^{55}Fe and X-Ray Tracks

3.1 Why Use ^{55}Fe

^{55}Fe is often used in calibration for scintillation detectors due to its near-exclusive emission of constant energy x-rays (Schötzig, 2000). K-alpha x-rays, which are a result of electron capture within the ^{55}Fe nucleus, are emitted with an energy of 5.89 keV at high frequency. If placed into our prototype TPC, these x-rays are of high enough energy to produce ionization tracks (see Figure 3.1).

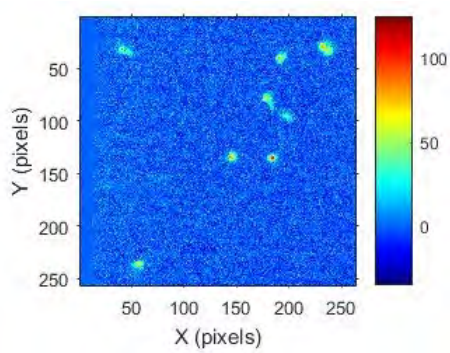

FIGURE 3.1: 4x4 binning image of 5.89 keV K-alpha x-ray tracks produced by an ^{55}Fe source. They are small and dim, but their intensities are constant.

Due to their low energy and low energy loss, ^{55}Fe tracks are not nearly as bright as α particle tracks. However, creating x-ray energy spectra at various increments of CS_2 provides a strong method to determine decreased light yield as a function of CS_2 percentage, as the peak of these spectra will always correspond to 5.89 keV. Intensities of a large number of tracks must be acquired to produce such spectra. As such, a second algorithm was written that detects ^{55}Fe tracks and produces a spectrum of their intensities. It is this spectrum that is used to quantify the light loss due to CS_2.

3.2 The Algorithm

I have developed a second algorithm that, much like the first, begins by creating a binary image of all x-ray tracks in the image. Pixels whose intensity surpasses a certain multiple of the standard deviation of the background are selected as potential tracks.

In the case of the x-ray track images this value is lower than that of the α tracks, as the dimmer x-ray tracks are closer in intensity to the background: For 16x16 binning, I chose that a pixel must have an intensity of 1.5σ above the background to be a candidate for a track. An example of this binary image can be seen in Figure 3.2. A low σ value insures the entire track is selected.

Signal-to-noise of these tracks are low enough to allow many pixels with no track affiliation into the binary image. Matlab's `bwconncomp` function is used, once again, to record connected groups of pixels; any collection fewer than 7 pixels are discarded with the remainder characterized as tracks. All remaining pixel groups are cycled through individually (see Figure 3.3).

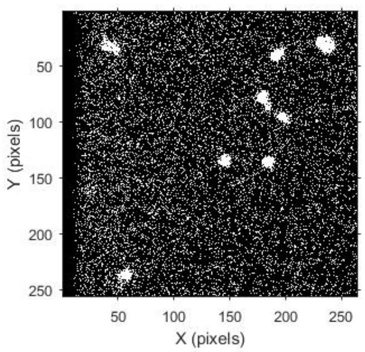

FIGURE 3.2: A 16x16 binary image of every pixel 1.4σ above the background average.

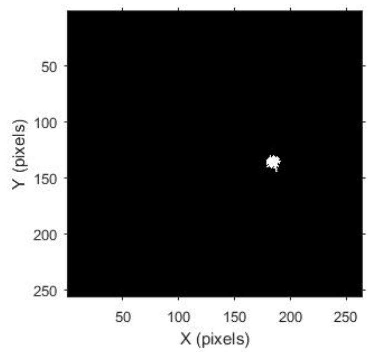

FIGURE 3.3: Selection of an individual track allowing for further analysis.

Once a track has been selected, the intensity values of the corresponding pixels in the original image are summed, giving us the total intensity of the x-ray track. This process is repeated for every track in the image, and every image taken for a particular increment of CS_2.

3.3 Results for Light Loss with CS_2

After determining all individual track intensities for 0 CS_2, an energy spectrum can be created. This is merely a histogram of the aforementioned intensities (see Figure 3.4).

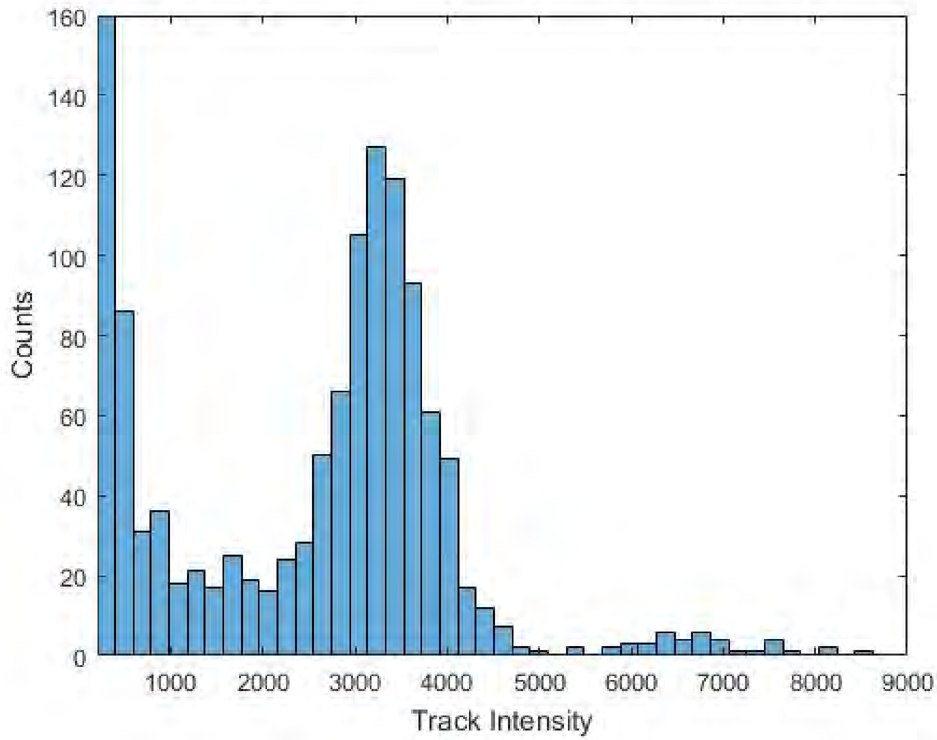

FIGURE 3.4: Energy spectrum of 5.9 keV x-rays at 16 by 16 binning. A peak lies clearly around 3170 ADUs. The high number of low energy tracks are a result of non-track pixels that escaped elimination. The second peak around 6200 ADUs is a direct result of overlapping tracks.

This process was repeated with increasing fractions of CS_2, in increments of about 0.5 Torr. As signal-to-noise is more important than resolution, 16 by 16 binning was chosen to guarantee x-ray tracks would still be visible at high levels of the gas. Peak intensities were recorded at each CS_2 increment resulting in the following data:

Torr of CS_2	Peak Intensity (ADUs)
0	3175
0.5	1115
1.2	995
1.8	895
2.3	625
2.9	595
3.5	495
4.1	485
4.7	445
5.4	385
5.9	365

TABLE 3.1: Peak intensities (in ADUs) determined at every increment of CS_2.

3.4 Summary: the Ideal CS_2 Concentration

The decreasing trend in light yield was apparent in the ^{55}Fe data. This information can can be coupled with the data acquired of α track widths to find the ideal ratio of CF_4 and CS_2 (see Figure 3.5).

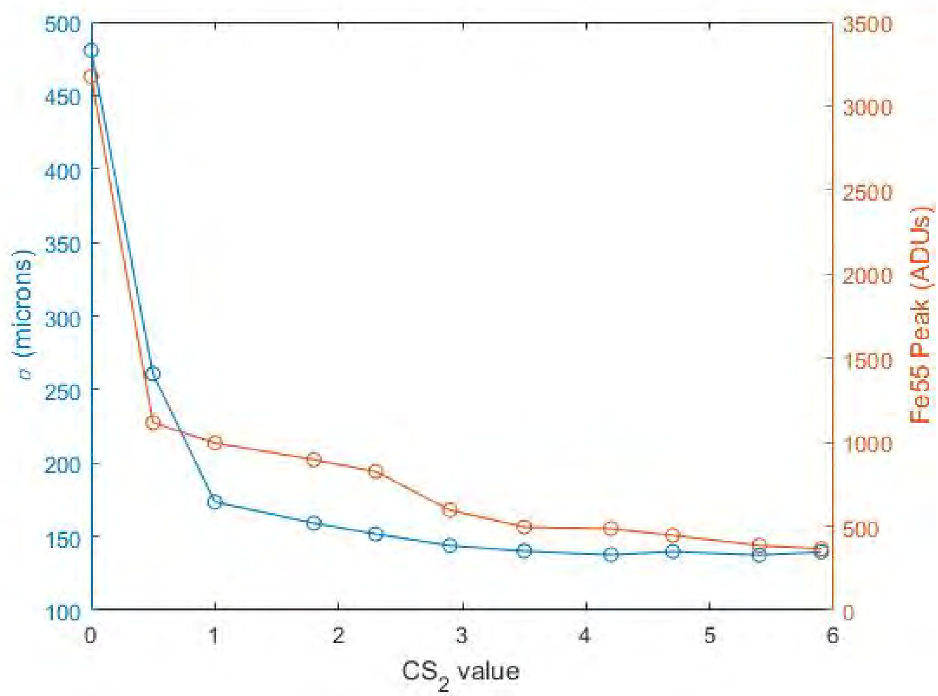

FIGURE 3.5: Here, the remarkable effect of CS_2 is easy to see, while the constant decrease in light yield is similarly visible. The ideal amount of CS_2 occurs between 3-4 Torr, after which the α width appears to reach a minimum.

From Figure 3.5 it is easy to see that the decreased width of α tracks plateaus around a σ of 150 microns. However, light yield continues to decrease steadily. The α plateau occurs around 3 Torr of CS_2. It can therefore be concluded, assuming that this plateau is truly the point of diminishing returns and *not* a result of poor resolution, that the ideal ratio of CS_2 to CF_4 is approximately 3:150, or 2%.

Chapter 4

Conclusions and Further Research

Changes can be made to improve the results of this thesis. As stated previously, 1x1 binning images should be taken at all increments of CS_2. This will show a further decrease in width assuming pixelization from the 2x2 images resulted in a hard limit of width reduction. Light yield was not an apparent issue at 2% CS_2, so increasing the ratio above 2% is certainly an option. Similarly, the use of a radon transform to rotate the images will remove some unfortunate blemishes created in the images due to naive image rotation techniques. This can show, more accurately, the true effect of the CS_2.

An important aspect of CS_2 is its toxicity. The National Center for Biotechnology Information's database on chemical molecules PubChem lists CS_2 as a dangerous chemical. The chemical is not only flammable, but also a powerful neurotoxin and can result in reproductive toxicity. These dangers have kept the gas from widespread use. Fortunately, there exists a second electronegative gas with nearly identical properties to CS_2 called sulfur hexaflouride (SF_6). SF_6 has the added benefit of being entirely safe breath, and therefore lack any safety concerns in a laboratory setting. This strong replacement requires the same analysis as done in this thesis, but similar, or identical, results are expected.

A roughly 75% reduction in width increase due to diffusion is remarkable; clearly CS_2 is a powerful tool to combat diffusion within TPCs. Although the advantages of CS_2 are obvious, its true potential has yet to be determined. Direct dark matter detection experiments must be able to resolve nuclear recoil tracks of very low energies. CS_2 will vastly improve upon the lower limit of recoil energies set by pure CF_4 detectors. In future iterations of this work we will determine the lowest energy of nuclear recoil in which directionality can be determined using the new ideal ratio of 2% CS_2. Given the success of our results thus far, we expect that value to drop significantly, bringing us one step closer to detecting the WIMP.

Appendix A

Mathematics of the Hough Transform

The Hough Transform is a powerful line detection tool in modern image analysis. The following describes the mathematic process of this transform. In Cartesian coordinates, a line has the equation:

$$y = mx + b \tag{A.1}$$

where m and b are the parameters giving slope and y-intercept, respectively. Points on the line (x_i,y_i) similarly follow the equation:

$$y_i = mx_i + b \tag{A.2}$$

Every single point on the line is related to every other point by this equation; each (x_i,y_i) are related by the same m and b values.Therefore, the Cartesian equation of a line can be mapped onto a new, slope-intercept coordinate system. Here, every line follows the equation:

$$b = m(-x) + y \tag{A.3}$$

The points (x_i,y_i) can be plotted as lines in the this new coordinate system. These new lines will all intersect at a single point (m,b), which are the slope and y-intercept of the original line. This idea (shown in Figure A.1) forms the basis of the Hough Transform.

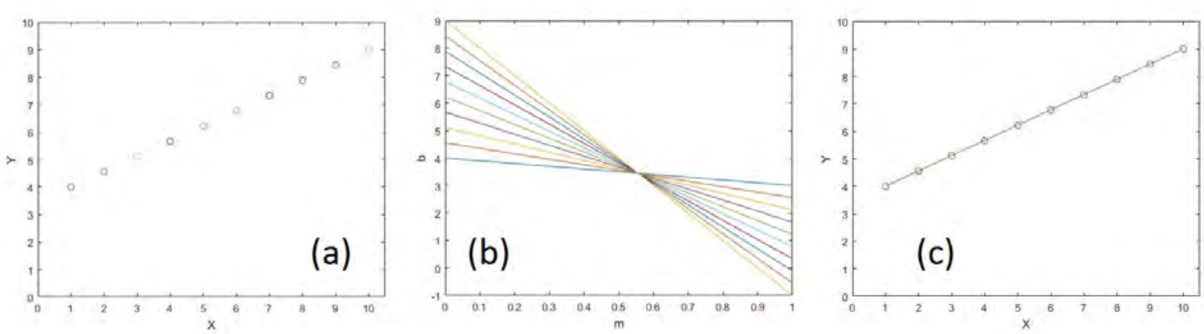

FIGURE A.1: **(a)** shows several points (x_i,y_i). Together, these points form a line. **(b)** shows the lines created using Equation A.3 intersecting at one point: (m,b). **(c)** shows the line created using the slope and intercept (m,b) found in **(b)**

This method runs into issues when the slope nears or becomes infinity (vertical lines). To solve this issue, the lines are instead written in trigonometric form:

$$x \cos\theta + y \sin\theta = \rho \tag{A.4}$$

Where ρ is the distance from some origin and θ is the angle between the y axis and $\vec{\rho}$ (as seen in Figure A.2).

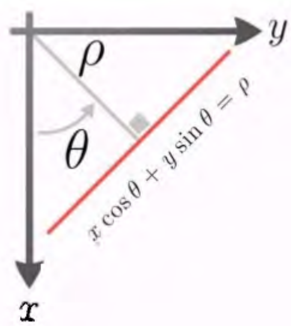

FIGURE A.2: A line parameterized by ρ and θ.

The Hough Transform takes a binary image and plots every illuminated "1" pixel as a curve in the new parameter space. A line in the image is then detected by looking at (ρ,θ) point in which the highest number of intersections occur.

For further review of the Hough Transform, see Illingworth and Kittler, 1988 and Shehata Hassanein et al., 2015.

Appendix B

Matlab Codes

B.1 Calibration

Correct calibration of an image is the key to strong image analysis. Without it, data is flawed and end results will be incorrect. Fortunately, the general process for correct image calibration is widely understood: removal of the pixel value pedestal using dark frames, and correction of optical anomalies using flat frames. However, more intricate methodology must be put in place to achieve best calibration.

The following algorithm makes a master dark frame using a combination of multiple individual frames. This algorithm also records any bad pixels found in any of the dark frames. A ”bad pixel” refers a pixel who has higher than average pedestal value. These are likely issues with the CCD itself and must be accounted for in every image taken (even data frames).

The master dark frame is subtracted from every light image used for analysis. No flat frame was created for this thesis, but would likely improve calibration.

```
ChooseBin=input('Enter # for #x# binning images:\n');
Num_C=input('Enter the number of Dark Images desired:\n');
DarkCal=cell(1,Num_C);

if ChooseBin==2
for ii = 1:Num_C
DarkCal{ii}=fitsread(sprintf('alpha-%03ddark.fit',ii));
end
elseif ChooseBin==4
for ii = 1:Num_C
DarkCal{ii}=fitsread(sprintf('alpha-%03ddark2.fit',ii));
end
elseif ChooseBin==6
for ii = 1:Num_C
DarkCal{ii}=fitsread(sprintf('alpha-%03ddark3.fit',ii));
end
else
fprintf('That is not an acceptable binning number! Try Again!\n')
end

%------------------------------Form Master Dark------------------------------------
```

```
[N_x, N_y] = size(DarkCal{1});
HH=6;
D_master_sum=zeros(N_x,N_y);
for ii=1:Num_C
D_master_sum=D_master_sum+DarkCal{ii};
end
D_master=D_master_sum./Num_C;

D_STD=std(D_master(:));
D_Mean=median(D_master(:));
D_alpha=1;
B_pix=zeros(N_x,N_y);

% The following two loops look for pixels in each individual callibration
    image to
% verify the presence of bad pixels, then removes them/notes their
% location.
for ii=1:N_x
for jj=1:N_y
if D_master(ii,jj)>D_Mean+D_alpha*D_STD
B_pix(ii,jj)=D_master(ii,jj);
%Number=Number+1;
end
end
end
for qq=1:Num_C
Mean_cur=mean(DarkCal{qq}(:));
STD_cur=std(DarkCal{qq}(:));
for ii=1:N_x
for jj=1:N_y
if B_pix(ii,jj)>0
if DarkCal{qq}(ii,jj) > Mean_cur+D_alpha*STD_cur
DarkCal{qq}(ii,jj)=0;
end
end
end
end
end

Avg_num=0;
Avg_tot=0;
S_mean=0;
D_STD=0;

% For every Cal image individually, the following searces for semi bad
% pixels, calculates the mean/std without their input, and makes note of
% their values.
```

```
for qq=1:Num_C
for ii=1:N_x
for jj=1:N_y
if DarkCal{qq}(ii,jj)~=0
Avg_tot=Avg_tot+DarkCal{qq}(ii,jj);
Avg_num=Avg_num+1;
end
end
end
MeanD=Avg_tot/Avg_num;
for ii=1:N_x
for jj=1:N_y
if DarkCal{qq}(ii,jj)~=0
S_mean=S_mean+(DarkCal{qq}(ii,jj)-MeanD)^2;
end
end
end
D_STD=sqrt(S_mean/Avg_num);
for ii=1:N_x
for jj=1:N_y
if DarkCal{qq}(ii,jj)>MeanD+D_alpha*D_STD
DarkCal{qq}(ii,jj)=MeanD;
end
end
end
for ii=1:N_x
for jj=1:N_y
if DarkCal{qq}(ii,jj)==0
DarkCal{qq}(ii,jj)=MeanD-HH*rand(1,1)+(HH.*rand(1,1)+HH.*rand(1,1).*rand
    (1,1));
end
end
end
Avg_num=0;
Avg_tot=0;
S_mean=0;
D_STD=0;
end

D2_alpha=1.5;
for qq=1:Num_C
for ii=1:N_x
for jj=1:N_y
if DarkCal{qq}(ii,jj)~=0
Avg_tot=Avg_tot+DarkCal{qq}(ii,jj);
Avg_num=Avg_num+1;
end
```

```
end
end
MeanD=Avg_tot/Avg_num;
for ii=1:N_x
for jj=1:N_y
if DarkCal{qq}(ii,jj)~=0
S_mean=S_mean+(DarkCal{qq}(ii,jj)-MeanD)^2;
end
end
end
D_STD=sqrt(S_mean/Avg_num);
for ii=1:N_x
for jj=1:N_y
if DarkCal{qq}(ii,jj)>MeanD+D2_alpha*D_STD
DarkCal{qq}(ii,jj)=MeanD-HH*rand(1,1)+(HH.*rand(1,1)+HH.*rand(1,1).*rand(1,1));
end
end
end
Avg_num=0;
Avg_tot=0;
S_mean=0;
D_STD=0;
end

% The following creates a final averaged dark callibration frame
D_master_sum=zeros(N_x,N_y);
for ii=1:Num_C
D_master_sum=D_master_sum+DarkCal{ii};
end
D_master=D_master_sum./Num_C;
```

B.2 Alpha Codes

The following are the Matlab codes used for track detection and analysis for α particle tracks in the prototype TPC. The method of which is described in the text.

```
GG=0;
Im_max=input('Enter how many light images you have:\n');
for gg=1:Im_max
A=fitsread(sprintf('alpha-%03dlight.fit',gg));
CC=fitsread('alpha-001dark.fit');
%-----------------------------------------------------------------------%

[N_x, N_y] = size(A);
A=A-D_master;
```

```
CCD=CC-D_master;
STD=std(CCD(:));
Mean=mean(CCD(:));
alpha=3.3;
CR=23;
BB = zeros(N_x,N_y);
XsYs = zeros(N_x,N_y);

for ii=1:N_x
for jj=1:N_y
if B_pix(ii,jj)>0
A(ii,jj)=-20 + (20+20).*rand(1,1);
end
end
end

% The following displays a scaled color image
%figure
%imshow(A,[],'Colormap',jet(255))


%------------------------------ Track Identification ------------------------------

% Creates a binary image of all pixels above a threshold value
for ii = 2:N_x-1
for jj = 2:N_y-1
if A(ii,jj)>Mean+alpha*STD
XsYs(ii,jj)=1;
else
XsYs(ii,jj)=0;
end
end
end

% Remove the edges from the image (we don't want any track that appears too
% close to the edge).
EdgeR=15;
for ii=1:N_x
for jj=1:EdgeR
XsYs(ii,jj)=0;
end
end
for ii=1:N_x
for jj=N_y-EdgeR:N_y
XsYs(ii,jj)=0;
end
```

```
end
for ii=1:EdgeR
for jj=1:N_y
XsYs(ii,jj)=0;
end
end
for ii=N_x-EdgeR:N_x
for jj=1:N_y
XsYs(ii,jj)=0;
end
end

% Find all connected pixels

CC = bwconncomp(XsYs);
XsYs2=XsYs;
A2=A;
numPixels = cellfun(@numel,CC.PixelIdxList);

% Now cycle through each connected pixels (tracks) to find information
% about each track.

for ww=1:length(numPixels)
XsYs=XsYs2;
A=A2;
B=0;
idx=ww;
for ii=1:length(numPixels)
if ii == idx
continue
end
XsYs(CC.PixelIdxList{ii}) = 0;
end

% Ignore small tracks
if sum(XsYs(:))<400
continue
end
% Ignore overlapping tracks
if sum(XsYs(:))>6000
continue
end
%imshow(XsYs)------

% Remove any isolated pixels
for ii = 2:N_x-1
for jj = 2:N_y-1
```

```
if XsYs(ii+1,jj)==0 && XsYs(ii,jj+1)==0 && ...
XsYs(ii-1,jj)==0 && XsYs(ii,jj-1)==0
XsYs(ii,jj)=0;
end
end
end


% ------------------------------------Intensity----------------------------------------------

% For all values in XsYs that a marked one, a correpsonding matrix assigns
% pixels values of A to itself: gives TOTAL intenstiy of the track.
Intense = zeros(N_x,N_y);
for ii = 1:N_x
for jj = 1:N_y
if XsYs(ii,jj)==1
Intense(ii,jj)=A(ii,jj);
else
Intense(ii,jj)=0;
end
end
end

% The total pixel value of the image!
Intens_tot = sum(Intense(:));
Int_end=1000;
if Intens_tot<Int_end
continue %error('No Valid Track')
end
GG=GG+1;

%--------------------------------Find Curves In Track----------------------------------
% Bin the image order get the width of track to ~1 pixel, to easily show
% curve in track. Bin only in the x direction to maintain length.
[N_x, N_y] = size(A);
Xbin=6;
for ii = 1:N_x
qq=1;
for jj = 1:Xbin:N_y
BB(ii,qq)=XsYs(ii,jj)+XsYs(ii,jj+1)+XsYs(ii,jj+2)+XsYs(ii,jj+3)+XsYs(ii,jj
    +4)+XsYs(ii,jj+5);%+A(ii,jj+6)+A(ii,jj+7);
%+A(ii,jj+8)+A(ii,jj+9)+A(ii,jj+10)+A(ii,jj+11);
qq=qq+1;
end
end

[BB_x, BB_y] = size(BB);
```

```
for ii = 1:BB_x
for jj = 1:BB_y
if BB(ii,jj)>0
BB(ii,jj)=1;
end
end
end

% Use a Hough Transform to identify lines on the binned image, as well as
% the start/end points of each line segment.

[H,theta,rho] = hough(BB);
P=houghpeaks(H,2,'threshold',ceil(0.3*max(H(:))));
x = theta(P(:,2));
y = rho(P(:,1));
lines = houghlines(BB,theta,rho,P,'FillGap',5,'MinLength',7);
%imshow(BB), hold on
max_len=0;

% Remove tiny line that messes everything up
if length(lines)>2
for kk=1:length(lines)
ybad(kk)=lines(kk).point2(2);
end
Ybad=min(ybad);
for kk=1:length(lines)
if lines(kk).point2(2)==Ybad
break
end
end
lines(kk)=[];
end

if length(lines)==1
GG=GG-1;
continue
end

for k = 1:length(lines)
xy = [lines(k).point1; lines(k).point2];

% plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
% Plot beginnings and ends of lines
% plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
% plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');
end

```

```
hold off

if length(lines)==0
GG=GG-1;
continue
end
xy_3 = [lines(1).point1; lines(1).point2];
xy_4 = [lines(2).point1; lines(2).point2];
xy_1 = [xy_3(1) xy_4(1); xy_3(2) xy_4(2)];
xy_2 = [xy_3(3) xy_4(3); xy_3(4) xy_4(4)];

% With the start/end points of both lines - we can easily find the point at
% which the lines intersect.
dx = diff(xy_1);  %# Take the differences down each column
dy = diff(xy_2);
den = dx(1)*dy(2)-dy(1)*dx(2);  %# Precompute the denominator
ua = (dx(2)*(xy_2(1)-xy_2(3))-dy(2)*(xy_1(1)-xy_1(3)))/den;
ub = (dx(1)*(xy_2(1)-xy_2(3))-dy(1)*(xy_1(1)-xy_1(3)))/den;

xi = xy_1(1)+ua*dx(1);
yi = xy_2(1)+ua*dy(1);

%imshow(A,[],'Colormap',jet(255))
%hold on
%plot(xi,yi,'*')

x = [0 length(A)];
y = [yi yi];
line(x,y)

% The intersection point of the two lines are taken to be the point that a
% substantial curve is detected. Remove all data below this point.

[N_x, N_y] = size(A);
YBot=floor(yi);
if yi==Inf
GG=GG-1;
continue
end
if isnan(yi)==1
GG=GG-1;
continue
end
if yi<0
[row, colomn] = find(XsYs);
YBot = max(row);
```

```
end
if yi>length(A)
[row, colomn] = find(XsYs);
YBot = max(row);
end
if yi>size(XsYs,1)
[row, colomn] = find(XsYs);
YBot = max(row);
end

for ii = YBot:N_x
for jj = 1:N_y
XsYs(ii,jj)=0;
end
end
for ii = YBot:N_x
for jj=1:N_y
A(ii,jj)=0;
end
end
if yi==1
yi=2;
YBot=2;
end
Bottom=0;
for ii=1:N_y
if XsYs(YBot-1,ii)==1
Bottom=Bottom+1;
end
end
BotNum=round(Bottom/2);
for ii = 1:N_y
if XsYs(YBot-1,ii)==1
break
end
end
BotMid=BotNum+ii-1;

% Find the highest y value
[row, colomn] = find(XsYs);
YTop = min(row);
YTop=YTop+8;
Toptop=0;
for ii=1:N_x
for jj=1:YTop
XsYs(jj,ii)=0;
end
```

```
end
YTop=YTop+1;
for ii=1:N_y
if XsYs(YTop,ii)==1
Toptop=Toptop+1;
end
end
TopNum=round(Toptop/2);
for ii = 1:N_y
if XsYs(YTop,ii)==1
break
end
end
TopMid=TopNum+ii-1;

%imshow(XsYs)
hold on
% plot(BotMid,YBot,'*')
% plot(TopMid,YTop,'*')
hold off

% We now have the (x,y) values of the middle of the top and bottom of the
% track. With this information, we can rotate the image so the track
% alligns with the y axis.

if sum(XsYs(:))==0
GG=GG-1;
continue
end
% Find distance between two points
Length=sqrt((TopMid-BotMid)^2+(YTop-YBot)^2);
if length(Length)==0
GG=GG-1;
continue
end
% Find X distance between the points
X_dist=abs(BotMid-TopMid);
% Find angle between the points
Ang = asin(X_dist/Length);

%rotate the image (Both XsYs2 and A)
Deg=57.2958*Ang;
if BotMid-TopMid<0
B = imrotate(A,Deg);
elseif BotMid-TopMid>0
B = imrotate(A,-Deg);
else
```

```
B=imrotate(A,Deg);
end
if BotMid-TopMid<0
BXsYs = imrotate(XsYs,Deg);
elseif BotMid-TopMid>0
BXsYs = imrotate(XsYs,-Deg);
else
BXsYs=imrotate(XsYs,Deg);
end
[B_x,B_y] = size(B);

% Now remove all data except a square in the image that comtains the track.
for ii=1:B_x
BBB=0;
GGG=0;
for jj=1:B_y
BBB=BXsYs(ii,jj)+BBB;
if BBB>0
GGG=ii;
break
end
end
if GGG==ii
break
end
end
BYTop=ii;
BToptop=0;
for ii = 1:B_y
if BXsYs(BYTop,ii)==1
BToptop=BToptop+1;
end
end
BTopNum=round(BToptop/2);
BTopFind=0;
for ii = 1:B_y
if BXsYs(BYTop,ii)==1
BTopFind=BTopFind+1;
end
if BTopFind==BTopNum
break
end
end
BTopMid=ii;
BOXtop=5;
BOXside=30;
```

```
for ii = 1:BYTop-BOXtop
for jj=1:B_x
B(ii,jj)=0;
end
end
for ii = 1:B_y
for jj = 1:BTopMid-BOXside
B(ii,jj)=0;
end
end
for ii = 1:B_y
for jj = BTopMid+BOXside:B_x
B(ii,jj)=0;
end
end

% Add all values along the columns to find the width of the track.
Tot=sum(B);
Xs=1:length(Tot);
% Fit a Gaussian to the data
f = fit(Xs.',Tot.','gauss1');
%plot(f,Xs,Tot)
%pause

% Calculates FWHM
FitCoef=coeffvalues(f);
c_i=FitCoef(3);
sigma=c_i/sqrt(2);
width=2*sqrt(2*log(2))*sigma;

% Store some data: this will be used in scatter plots later on. Note, here
% Length is the length of the cut track (curved removed).
FWHM=width';
Intensity=Intens_tot';

Length_list(GG)=Length;
Int_list(GG)=Intensity;
Ang_list(GG)=Deg;
FWHM_list(GG)=FWHM;


if FWHM > 18
GG=GG-1;
continue
end

if Length < 20
```

```
GG=GG-1;
continue
end
if Deg>25
GG=GG-1;
continue
end

end
end
```

B.3 ^{55}Fe Codes

The following are the Matlab codes used for track detection and analysis for ^{55}Fe x-ray tracks in the prototype TPC. The method of which is described in the text.

```
NumFE=input('How many Fe55 images are there?\n');
Intens_tot=zeros(1,1);
SIZE = 20;
for gg=1:NumFE
A=fitsread(sprintf('fe55-%03dlight1.fit',gg)); %200 rn
CC=fitsread('fe55-001dark1.fit');
%-------------------------------------------------------------------------%
[N_x, N_y] = size(A);
A=A-D_master;
CCD=CC-D_master;
STD=std(CCD(:));
Mean=mean(CCD(:));
alpha=1.7;
XsYs = zeros(N_x,N_y);
XsYs2= zeros(N_x,N_y);
for ii=1:N_x
for jj=1:N_y
if B_pix(ii,jj)>0
A(ii,jj)=0; %0
end
end
end

% The following displays a scaled color image
%imshow(A,[],'Colormap',jet(255))
for ii = 2:N_x-1
for jj = 2:N_y-1
if A(ii,jj)>Mean+alpha*STD
XsYs(ii,jj)=1;
else
```

```
XsYs(ii,jj)=0;
end
end
end
EdgeR=4;
for ii=1:N_x
for jj=1:EdgeR
A(ii,jj)=0;
end
end
for ii=1:N_x
for jj=N_y-EdgeR:N_y
A(ii,jj)=0;
end
end
for ii=1:EdgeR
for jj=1:N_y
A(ii,jj)=0;
end
end
for ii=N_x-EdgeR:N_x
for jj=1:N_y
A(ii,jj)=0;
end
end

% Remove any isolated pixels
for ii = 2:N_x-1
for jj = 2:N_y-1
if XsYs(ii+1,jj)==0 && XsYs(ii,jj+1)==0 && ...
XsYs(ii-1,jj)==0 && XsYs(ii,jj-1)==0
%XsYs(ii+1,jj+1)==0 && XsYs(ii-1,jj+1)==0 &&
%XsYs(ii+1,jj-1)==0 && XsYs(ii-1,jj-1)==0
XsYs(ii,jj)=0;
end
end
end
Ilabel = bwlabel(XsYs);
stat = regionprops(Ilabel,'centroid');
XXX=zeros(length(stat),2);
%imshow(A,[]); hold on; %--
for x = 1: numel(stat)
%plot(stat(x).Centroid(1),stat(x).Centroid(2),'ro'); %--
XXX(x,:)=[stat(x).Centroid(1),stat(x).Centroid(2)];
end
%hold off
%imshow(XsYs)
```

```
%pause
Intensities=zeros(length(XXX),1);
B=zeros(size(A));
for ii = 1:length(XXX(:,1))
XsYs2=bwselect(XsYs,XXX(ii,1),XXX(ii,2),4);
if sum(XsYs2(:)) < SIZE
continue
end
for jj = 1:N_x
for qq = 1:N_y
if XsYs2(jj,qq)==1
B(jj,qq)=A(jj,qq);
else
B(jj,qq)=0;
end
end
end
if sum(B(:))==0
continue
end
if sum(B(:))>20000
continue
end
Intensities(ii)=sum(B(:));
Intens_tot(end+1)=Intensities(ii);
if Intensities(ii)>1500
%imshow(XsYs2)
%figure
%imshow(A,[])
%pause
% Use this to show that most tracks above a certain int. value is in
% fact two overlapping tracks.
end
end
end

figure
histogram(Intens_tot,60)
```

Bibliography

Bertone, G. and D. Hooper (2016). "A History of Dark Matter". In: *ArXiv e-prints*. arXiv: 1605.04909.

Daw, E. et al. (2012). "The DRIFT Directional Dark Matter Experiments". In: *EAS Publ. Ser.* 53, pp. 11–18. DOI: 10.1051/eas/1253002. arXiv: 1110.0222 [physics.ins-det].

Hinshaw, G. et al. (2013). "Nine-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Parameter Results". In: 208, 19, p. 19. DOI: 10.1088/0067-0049/208/2/19. arXiv: 1212.5226.

Howell, Steve B. (2000). *Handbook of CCD Astronomy*. 1st ed. Cambridge University Press. ISBN: 052164834.

Illingworth, J. and J. Kittler (1988). "A survey of the hough transform". In: *Computer Vision, Graphics, and Image Processing* 44, pp. 87–116. DOI: 10.1016/S0734-189X(88)80033-1.

Kolouri, S., S. R. Park, and G. K. Rohde (2016). "The Radon Cumulative Distribution Transform and Its Application to Image Classification". In: *IEEE Transactions on Image Processing* 25, pp. 920–934. DOI: 10.1109/TIP.2015.2509419. arXiv: 1511.03206 [cs.CV].

Komatsu, Eiichiro et al. (2014). "Results from the Wilkinson Microwave Anisotropy Probe". In: *PTEP* 2014, 06B102. DOI: 10.1093/ptep/ptu083. arXiv: 1404.5415 [astro-ph.CO].

Lewin, J. D. and P. F. Smith (1996). "Review of mathematics, numerical factors, and corrections for dark matter experiments based on elastic nuclear recoil". In: *Astroparticle Physics* 6, pp. 87–112. DOI: 10.1016/S0927-6505(96)00047-3.

Loomba, D. (2013). "RD with Implications for a Directional Low Mass Dark Matter Search". In: *CYGNUS 2013 Toyama*.

Martoff, C. J. et al. (2000). "Suppressing drift chamber diffusion without magnetic field". In: *Nuclear Instruments and Methods in Physics Research A* 440, pp. 355–359. DOI: 10.1016/S0168-9002(99)00955-9.

Mount, B. J. et al. (2017). "LUX-ZEPLIN (LZ) Technical Design Report". In: arXiv: 1703.09144 [physics.ins-det].

Phan, N. S. et al. (2016). "GEM-based TPC with CCD Imaging for Directional Dark Matter Detection". In: *Astropart. Phys.* 84, pp. 82–96. DOI: 10.1016/j.astropartphys.2016.08.006. arXiv: 1510.02170 [physics.ins-det].

Rubin, V. C. and W. K. Ford Jr. (1970). "Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions". In: 159, p. 379. DOI: 10.1086/150317.

Schötzig, U. (2000). "Half-life and X-ray emission probabilities of 55Fe". In: *Applied Radiation and Isotopes* 53, 469–472. DOI: 10.1016/S0969-8043(00)00166-4.

Shehata Hassanein, A. et al. (2015). "A Survey on Hough Transform, Theory, Techniques and Applications". In: *ArXiv e-prints*. arXiv: 1502.02160 [cs.CV].

van den Bergh, S. (1999). “The Early History of Dark Matter”. In: 111, pp. 657–660. DOI: 10.1086/316369. eprint: astro-ph/9904251.

Zwicky, F. (1933). “Die Rotverschiebung von extragalaktischen Nebeln”. In: *Helvetica Physica Acta* 6, pp. 110–127.