

May 2022

Custom Calibration and Correction of Photoemission Electron Microscope Images Using Graphene

Henry Bell
Macalester College, hbell@macalester.edu

Follow this and additional works at: <https://digitalcommons.macalester.edu/mjpa>



Part of the [Astrophysics and Astronomy Commons](#), and the [Physics Commons](#)

Recommended Citation

Bell, Henry (2022) "Custom Calibration and Correction of Photoemission Electron Microscope Images Using Graphene," *Macalester Journal of Physics and Astronomy*. Vol. 10: Iss. 1, Article 2.
Available at: <https://digitalcommons.macalester.edu/mjpa/vol10/iss1/2>

This Honors Project - Open Access is brought to you for free and open access by the Physics and Astronomy Department at [DigitalCommons@Macalester College](mailto:DigitalCommons@MacalesterCollege). It has been accepted for inclusion in *Macalester Journal of Physics and Astronomy* by an authorized editor of [DigitalCommons@Macalester College](mailto:DigitalCommons@MacalesterCollege). For more information, please contact scholarpub@macalester.edu.

Custom Calibration and Correction of Photoemission Electron Microscope Images Using Graphene

Abstract

The Photoemission Electron Microscope (PEEM) is a full-field electron microscope that utilizes the photoelectric effect to image a surface. Due to a spatial resolution on the order of 10 nanometers and its ability to image both the morphology of a surface and its band structure, it is a useful tool for understanding the properties of materials for use in electronic devices. To correct for random sample misalignment and the experimental frame of reference in the spectroscopy mode of the PEEM, the 3D dataset must be rotated in both the momentum and energy coordinates which requires pixel calibration and energy alignment. I have created custom Python scripts to both automate this process and standardize the calibration and correction procedure to streamline data analysis for users of the PEEM. Graphene was utilized as an initial calibration material due to its distinct electronic band structure. The 6 Dirac cones of graphene were used as iso-energy points to align the frames on the energy axis and a series of matrix operations were utilized to rotate the image in the momentum axis to correct for sample misalignment. I used the corrected dataset to estimate the Fermi velocity and compared it to theoretical predictions.

Keywords

Photoemission Electron Microscope, Microscopy, Spectroscopy, Angle Resolved Photoemission Spectroscopy, Graphical User Interface, Computational, Graphene

Custom Calibration and Correction of Photoemission Electron Microscope Images Using Graphene

Henry Bell



Faculty Advisor: James Heyman
Department of Physics and Astronomy
Macalester College
Saint Paul, Minnesota
April 26th 2022

Acknowledgements

This project is the product of a remote research project that I worked on at the National Institute of Standards and Technology. I would like to thank my advisor, Sujitra Pookpanratana, for meeting with me every day of the summer to answer my questions, and providing thoughtful comments on drafts of this paper as well as mentoring me throughout the graduate school application process. I would also like to thank Falk Niefind and Ory Maimon for their help and comments on my research talk at NIST.

The physics and astronomy department at Macalester has been a great community to be a part of for the last four years. Thank you John Cannon for inspiring me to pursue physics and advising my class choices each semester. Thank you James Heyman and Tonnis ter Veldhuis for being amazing educators and listening and providing comments on my thesis talk and paper, as well as being part of my honors committee. I would also like to thank all of the professors and students of the physics and astronomy department for making my academic experience at Macalester fantastic.

Contents

1	Abstract	3
2	Introduction	4
2.1	Photoemission Electron Microscopy	5
2.1.1	Real Space Images	7
2.1.2	Momentum Space Images	9
2.2	Graphene	12
3	Computational Methods	14
3.1	General Correction	15
3.2	Real Space Correction	18
3.2.1	Thermal Drift Correction	18
3.3	Momentum Space Correction	21
3.3.1	Frame of Reference Correction	21
3.3.2	Energy and Momentum Calibration	25
3.3.3	Final Correction	29
4	Results and Discussion	32
4.1	Fermi Velocity	32
4.2	Graphical User Interface	34
4.2.1	Real Space	34
4.2.2	Momentum Space	36
4.3	Future Work	38
5	Cited Work	40

1 Abstract

The Photoemission Electron Microscope (PEEM) is a full-field electron microscope that utilizes the photoelectric effect to image a surface. Due to a spatial resolution on the order of 10 nanometers and its ability to image both the morphology of a surface and its band structure, it is a useful tool for understanding the properties of materials for use in electronic devices. To correct for random sample misalignment and the experimental frame of reference in the spectroscopy mode of the PEEM, the 3D dataset must be rotated in both the momentum and energy coordinates which requires pixel calibration and energy alignment. I have created custom Python scripts to both automate this process and standardize the calibration and correction procedure to streamline data analysis for users of the PEEM. Graphene was utilized as an initial calibration material due to its distinct electronic band structure. The 6 Dirac cones of graphene were used as iso-energy points to align the frames on the energy axis and a series of matrix operations were utilized to rotate the image in the momentum axis to correct for sample misalignment. I used the corrected dataset to estimate the Fermi velocity and compared it to theoretical predictions.

2 Introduction

The photoemission electron microscope (PEEM) at the National Institute of Standards and Technology (NIST) in the Nanoscale Device and Characterization division was built by SPECS and commissioned in 2019. No specialized software or instructions are included to perform correction and calibration of microscope images so these procedures had to be performed with custom scripts or non-specialized software each time. Correcting images in this way is undesirable because it is slow, tedious, and prone to errors. Additionally, some of the more specialized corrections are not feasible using current software. For example, the full 3D momentum space image on the PEEM was not able to be corrected using readily available software and had not yet been completed, only 2D frames of the dataset had been corrected. I had two interrelated goals for my project: to design image corrections to fully correct and calibrate a 3D momentum space image of graphene in energy, k_x and k_y . This correction procedure should also be applicable to other sample materials. My secondary goal was to code a specialized graphical user interface (GUI) to correct PEEM images for users at NIST. This GUI should be able to easily perform all needed corrections of real and momentum space images to transform them into publishable form. The GUI has a few main upsides over a manual correction strategy. It standardizes and streamlines the correction procedure which should reduce human error. It also decreases correction time significantly so that researchers can spend their time on other more important tasks. This paper will describe the different components that went into addressing the project goals and summarize results. I will first introduce and motivate the PEEM as an instrument and describe how it images a surface. I will then give background on the band structure of graphene and how it is grown. In the computational methods section I will describe the general corrections, real space corrections, and momentum space specific corrections in the order of how they are completed and then present the fully completed momentum space dataset. The results section will be focused on estimating

the Fermi velocity of the graphene sample using the corrected momentum space dataset and describing the functionality of the GUI. A discussion of potential future work is included at the end.

2.1 Photoemission Electron Microscopy

This section will be dedicated to describing the capabilities of the PEEM and why it is a useful tool for studying the properties of materials used in electronic devices. The later subsections will describe how the PEEM captures real space and momentum space images. The PEEM is an electron microscope whose mechanism of operation is the photoelectric effect. This refers to the electron's tendency to be ejected from a material that is illuminated with electromagnetic radiation and was first experimentally observed by Hertz in 1887. In 1905, the experimental results were accurately explained by theory formulated by Einstein using the idea that radiation is quantized. He derived the following equation relating the energy of photons to the kinetic energy of the emitted electrons.

$$h\nu = \Phi_0 + \frac{1}{2}m_e v^2 \quad (1)$$

Where ν is the frequency of radiation, Φ_0 is the work function of the material and m_e and v are the mass and velocity of the emitted electrons. In simple terms, the PEEM performs microscopy and spectroscopy by using this well-understood equation. Light of known frequency, ν , impinges on a sample which emits electrons. By analyzing the kinetic energy and momentum of those electrons we can experimentally determine spatial and electronic properties of the sample surface. A diagram of the path that the electrons take can be seen in Figure 1. First, a photon source produces radiation which impacts the sample. Electrons are emitted and accelerated by a large potential difference. The electrons continue through a series of lenses and an energy filtering apparatus and they are then projected onto a

detector. The microscope operates in an ultra high vacuum so that electrons interact minimally with the environment.

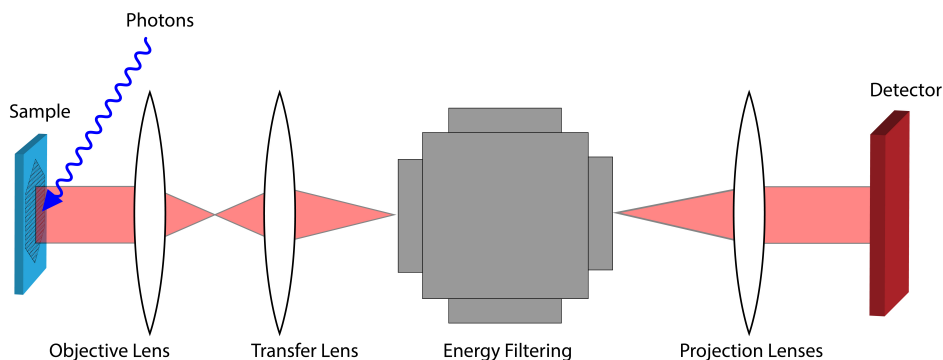


Figure 1: Simple diagram of the PEEM. Photons impinge on a sample which emits electrons that travel through a series of lenses and an energy filtering array. They are then projected onto a detector and recorded.

The PEEM has a relatively large spacial resolution compared to other microscopy techniques, around 10 nm. Other microscopes such as scanning electron microscopes or transmission electron microscopes have a resolution on the order of a nanometer. However, photoemission electron microscopy finds its use in distinguishing electronic differences in the surface of samples, unlike other microscopy methods that only distinguish topological features. PEEM is a full-field imaging technique, which makes it unique and ideal for imaging temporal processes. Additionally, many PEEMs are equipped with an energy filtering apparatus which allows angle resolved photoemission spectroscopy (ARPES) to be performed. Traditional ARPES images requires a uniform sample surface on the order of 1 mm to image, which is not feasible for most sample materials. Scanning tunneling microscopes can perform electronic structure measurements but restricted to scales of around 1 nm. The PEEM fills the length-scale gap between these extremes and can perform ARPES on a lateral length scale of 10's of nm to 100's of μm . This makes the PEEM an ideal instrument to image the electronic structure of nanoscale

electronic devices because they have functional electronic regions on the order of $30\ \mu\text{m}$ [6]. The overall goals for the PEEM at NIST is to advance PEEM-based metrology and to perform electronic measurements for use in advanced electronics.

2.1.1 Real Space Images

Microscope optics in the PEEM allow switching between real and momentum space imaging modes by tuning the electron lenses. These lenses can be activated or deactivated such that either the backfocal plane is projected onto the detector which forms a momentum space image or the image plane is projected onto the detector which forms a real space image. The photon source for real space images is usually a mercury arc lamp that produces light with energy of 5 eV. The detector of the PEEM that captures the final images comprises a phosphor screen that fluoresces when electrons impinge upon it. An image of this screen is then captured from behind by a camera such that each image has pixel width 1392 and height 1040 and contains over 1.4 million individual pixels. For each of these pixels, an intensity measurement is collected which is proportional to the number of electrons impacting the phosphor screen. An example real space image of epitaxial graphene taken by the PEEM before any correction can be seen in Figure 2. The contrast between the different features in this image are formed by both the differing work function and topography of the sample surface.

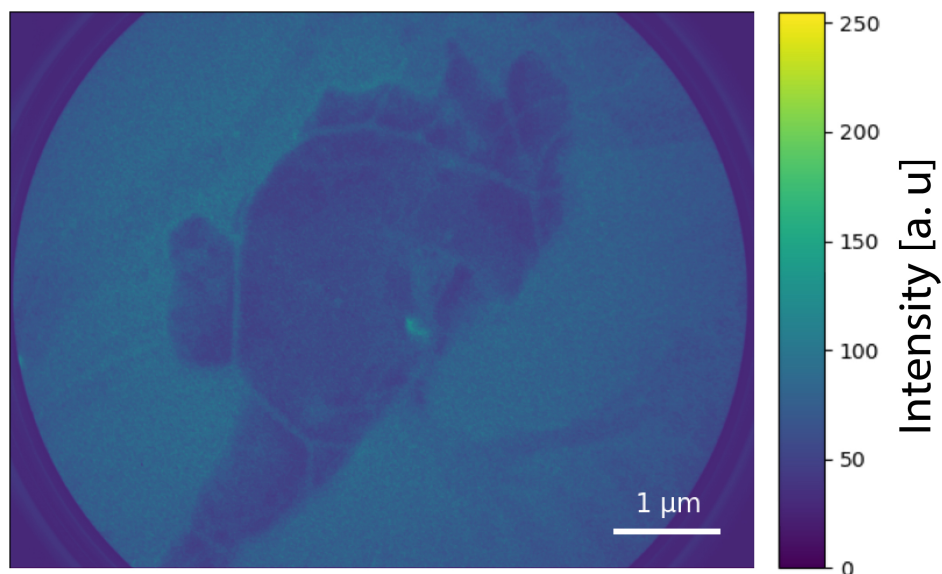


Figure 2: Raw PEEM image of epitaxial graphene with aperture diameter of 7 μm taken with a one second exposure time. Image is shown in a viridis color map where purple is low intensity, green is medium intensity and yellow is high intensity.

The real space imaging mode is used to both characterize the surface and find uniform regions of the surface to perform spectroscopy. The other main use for the real space mode is capturing the temporal changes in a sample.

2.1.2 Momentum Space Images

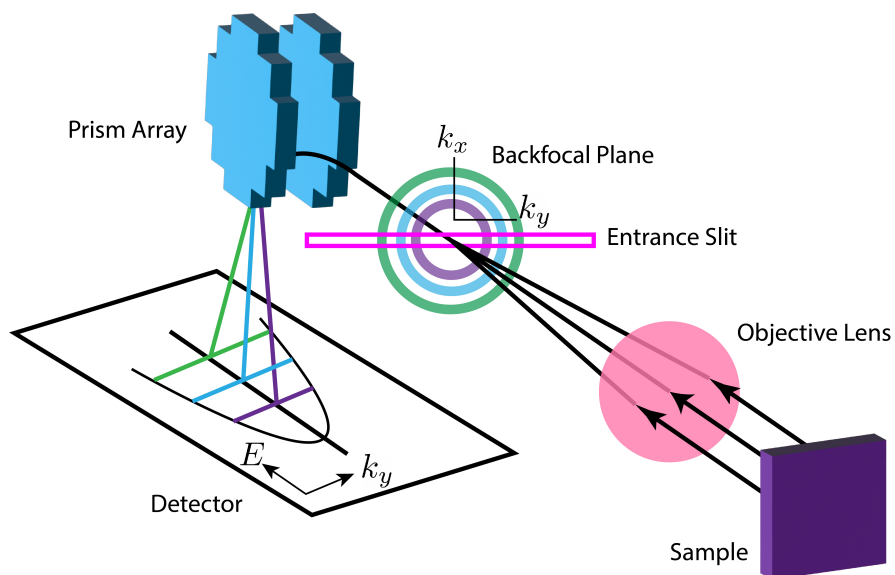


Figure 3: Image showing the path that electrons take in ARPES mode through the entrance slit and magnetic prism array. The electrons pass through an objective lens and are filtered in momentum using an entrance slit on the backfocal plane. The electrons then pass through the energy filter and one energy-momentum frame is formed on the detector. This Figure was created based on [11]

Momentum space images are captured using a helium arc lamp, a photon source which produces light with energy of either 21.2 eV or 40.8 eV. To produce an ARPES dataset, not only does the backfocal plane need to be projected onto the detector, the electrons need to be procedurally filtered with respect to their momentum and energy. Figure 3 shows the trajectory of electrons through the entrance slit and prism array which are utilized to filter the momentum and energy respectively and produce a 3D ARPES image. Electrons in the sample are excited by a photon source and are accelerated across large potential difference. They travel through an objective lens that focuses the electrons onto a backfocal plane. A narrow slit of width 1.5 μm

is placed in this plane which only allows electrons to pass with a certain k_x momentum. These electrons then pass through the magnetic prism array that disperses them with respect to energy [11, 12, 13]. The electrons are then projected onto the detector and recorded with the camera, producing one frame of the ARPES dataset. The lens mode of the microscope can be adjusted between the settings 5, 10 or 15 inverse angstroms which defines the range of momentum and energy projected on the detector. This procedure is repeated with the entrance slit at different vertical positions to form the full 3D dataset, six frames of which are shown in Figure 5.

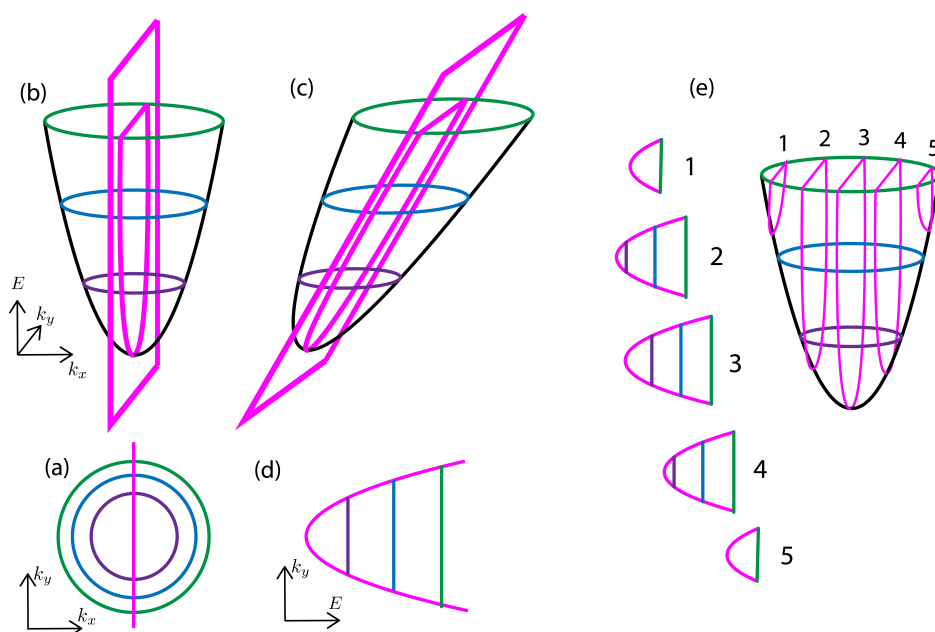


Figure 4: (a) shows the backfocal plane with the entrance slit placed over it to filter momentum. (b) shows the same image but also plotted in energy. (c) is the energy parabola after it has been dispersed by the energy filter and (d) is the image seen by the detector. (e) shows 5 different slit positions and their corresponding detector images. This figure is adapted from [11].

A more detailed description of how the electrons are filtered in momentum and energy in the PEEM is required for understanding how the ARPES

dataset is corrected. Illustrations in Figure 4 depict the electron's momentum and energy both in-plane and as a paraboloid at different stages of the electron optics shown in Figure 3. Consider three different energies of electrons, shown in green, blue, and purple which are listed by decreasing magnitude. In Figure 3a the distribution of electrons in k_x and k_y momentum is shown in the first backfocal plane immediately after the objective lens. The pink line depicts the slice of electrons that pass through the entrance slit. Figure 3b shows the same k_x and k_y distribution but plotted in a third axis, energy. An ARPES measurement aims to sample the distribution of electrons in this energy-momentum paraboloid. The electrons that pass through the slit are now represented by a plane in energy and k_y momentum. To discern the energy of the electrons, they are sent through the energy filter array. Dispersion occurs which can be seen in Figure 3c. The energy filter functionally projects energy onto the momentum axis perpendicular to k_y which allows energy to be imaged in the backfocal plane. The electrons are then focused onto the detector which forms the image in Figure 3d. Figure 3e shows five different slit positions on the energy paraboloid and the corresponding images formed on the detector. This shows that when multiple images at different slit positions are taken, the k_x , k_y , energy parabola can be constructed. In practice, 100 to 200 images are taken at evenly spaced slit positions of 1 to 2 mm to construct the ARPES datacube.

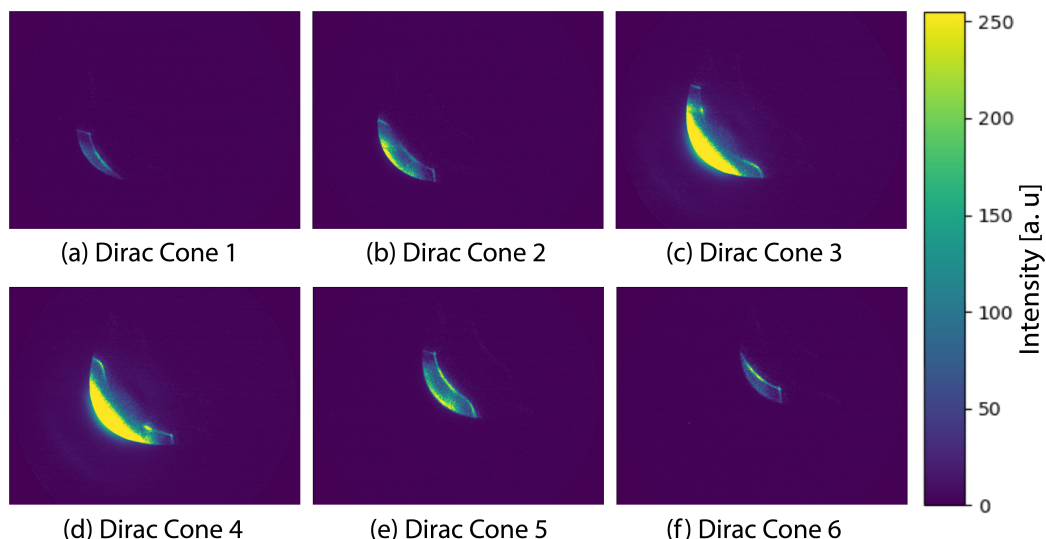


Figure 5: Six frames of raw 3D ARPES data of graphene. The six Dirac cones are shown at frame 23, 32, 63, 80, 120 and 129 of the 143 frame dataset. Each frame has been bright and dark field corrected to combat detector imperfection.

2.2 Graphene

Graphene is a 2D material consisting of carbon atoms arranged in a hexagonal honeycomb lattice. Graphene is of interest because it exhibits unusual charge transport properties, including an extremely high carrier mobility and vanishingly low effective mass. Graphene even exhibits a room temperature quantum Hall effect [8]. The interesting band structure of graphene that gives rise to these properties is well understood. A theoretical band dispersion diagram formulated using the tight binding model originally by Wallace [14] is shown in Figure 6 [2] along with a segment of a real space lattice of graphene and the first Brillouin zone in reciprocal space. The Brillouin zone is a hexagon centered at Γ and has vertices at K -points or Dirac points. The precise geometry of this band dispersion diagram is of importance to this paper as it will be used as a reference for correcting experimental results.

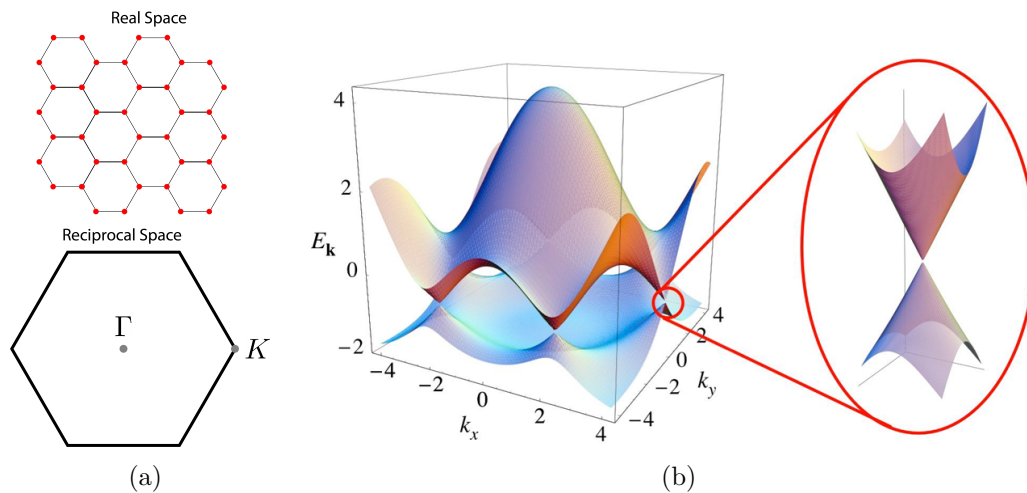


Figure 6: A section of the real space lattice of graphene and the corresponding reciprocal space unit cell with symmetry points marked is depicted in (a). The theoretical π band structure of graphene using tight binding approximation is shown in (b). Zoomed in to a K -point, a set of Dirac cones is shown. (b) is from [2] and was used with permission.

In Figure 6b, two surfaces can be seen: The conduction and valence bands which are the upper and lower surfaces respectively. Zoomed into a K -point, you can see the surfaces meet at the vertex of two opposing cones called Dirac cones. For a sample free of defects, the Fermi energy is located where the cones meet, the so called Dirac point. Due to this conical structure at K -points, there is an approximately linear dispersion relationship which requires that the Fermi velocity does not depend on momentum or energy. This means that charge carriers behave like relativistic massless particles. The Fermi velocity, a constant, is extremely large: About 10^6 m/s or 1/300 the speed of light [1] which results in a high carrier mobility.

Although graphene has been studied theoretically for over 70 years, first by Wallace [14], it was only physically realized less than 20 years ago in 2004 by Geim and Novoselov [7] who used the scotch tape method of fabrication. This method involves pulling apart graphene layers in crystalline graphite, which are only held together by weak Van der Waals bonds, with scotch

tape. A few years later, a different graphene fabrication method was discovered called graphitization of silicon carbide. When silicon carbide was first fabricated by inventor Edward Acheson, he discovered that when heated to a high temperature, the silicon atoms evaporate and graphite is created [1]. Later, it was noticed that by annealing the surface of silicon carbide repeatedly at temperatures up to 1400 °C, only the silicon at the surface evaporates and monolayer epitaxial graphene can be formed [10]. Other graphene growth methods have been discovered such as chemical vapor deposition [16]. The graphene that is imaged in this paper was fabricated using graphitization of silicon carbide in a custom vacuum furnace at NIST [15].

3 Computational Methods

The final goal of this project is to create a robust program that streamlines data correction for users of the PEEM. It should be able to complete any correction needed to transform a raw dataset into useable form. There are a few main computational challenges in this task. The script must calibrate each dataset such that each pixel of the detector is calibrated to a known unit amount. In a real space dataset, a ratio of pixels to length must be developed. In ARPES datasets, there are three pixel dimensions which must be calibrated with respect to energy and the two in-plane dimensions of electron momentum, k_x and k_y . Another challenge is transforming the ARPES dataset into a band dispersion diagram from its raw form. In the microscope, the electrons travel through an energy filtering array and designed to modify their trajectory based on their energy and momentum. These experimental transformations allow us to construct an ARPES image but corrections must be implemented to reformat the data into an image that can be compared to a band dispersion diagram. This requires the use of multiple 3D operations that must be applied in the correct order and requires manual input because each dataset must be corrected differently due to random misalignment. Fi-

nally, the implementation should be modular so that additional corrections can be added as needed.

We decided to implement this program in Python because I have the most experience in this language and it has a wide variety of modules that are useful for computation. I used the module Tkinter to create the graphical user interface (GUI), and NumPy and SciPy to complete matrix operations. The module matplotlib was used to visualize data and skimage was used to save data locally. In this section I will describe the corrections that are implemented in my GUI. This section is split into three main parts. First, general correction, which details corrections that are completed for any data collected in the PEEM. Namely, bright and dark field corrections, implemented similarly to [4]. Following that there are sections on real and momentum space corrections which contain information on corrections that are designed specifically for those data formats. My GUI is designed in a similar way, consisting of two separate windows to deal with real and momentum space images.

3.1 General Correction

For any detector, there will be inherent error. This error can be separated into two categories: random and systematic. Random error is error that shows no pattern between each subsequent measurement. In the PEEM, such error may be due to vibration of the system or noise due to low electron count. Typically, random error can be reduced during the experiment by using longer exposure times or averaging over many frames of data. Both of these strategies are utilized in the PEEM. For real space images, an exposure time on the order of a second is sufficient to reduce noise to acceptable levels. In ARPES measurements, the camera captures three frames at one-second exposure for each slit position. Each set of three frames is averaged to reduce random error.

Systematic error is a result of shortcomings in the experimental system

and can be seen in each measurement. In our case, the phosphor screen detector may produce non-uniform intensity due to defects or the phosphor screen-camera setup may have dead and bright pixels. To correct for this error, bright field images are taken with a zoomed-out lens mode. If the detector was perfect, a uniform and bright image would be seen. This allows us to distinguish detector imperfections so we can filter them out in surface images. An example bright field image can be seen in Figure 7a. In this image, a noticeable intensity gradient can be seen across the image as well as many detector imperfections such as the bright line in the lower left corner. A dark field image is also taken with the light source and phosphor screen turned off. A cropped dark field image is shown in 7b. This image allows us to filter out bright pixels.

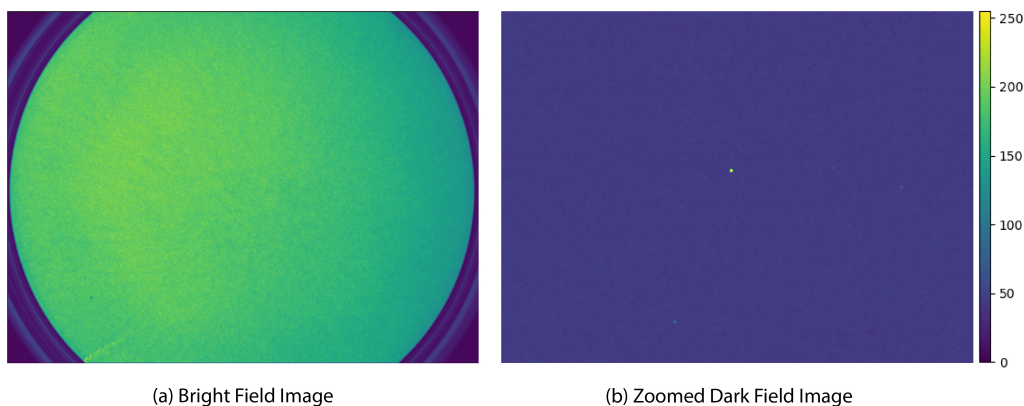


Figure 7: Example bright field (a) and dark field (b) images with a 2 and 10 second exposure time respectively. The dark field image is zoomed in to a 200 by 150 pixel rectangle and has a reduced contrast range from an intensity of 0 to 150 to highlight bright pixels.

For every image taken in the PEEM, real and momentum space, the following equation is applied using the bright and dark field images.

$$I_{i,j} = \frac{R_{i,j} - D_{i,j}}{B_{i,j}} \quad (2)$$

Where I is the resulting data, R is the raw image and D and B are normalized dark and bright field images that are represented as matrices with index i and j . We can see that if we input a dark field image as R , we get a completely dark image. If we input a bright field image, we get a uniform intensity image with removed bright pixels. These are the expected result of the bright and dark field images with a perfect detector. For a normal raw dataset, applying this equation should remove bright pixels and correct the contrast gradient of the detector. Applying Equation 2 to the raw data in Figure 2 yields the image in Figure 8.

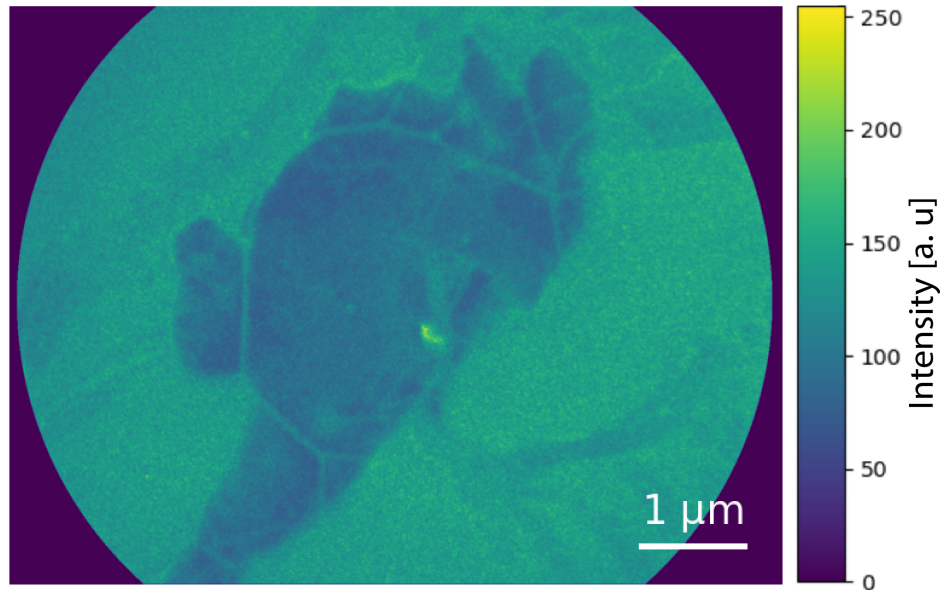


Figure 8: Corrected PEEM image of epitaxial graphene with aperture diameter of $7\ \mu\text{m}$ taken with 1 s exposure time.

Comparing the raw data in Figure 2 to the corrected data in Figure 8 we can see that the contrast gradient across the raw image has been corrected and a few dark and bright spots have been removed. This correction is implemented in my GUI and is the first operation that must be completed for each image taken on the PEEM, including ARPES images. Bright and

dark field images can be reused between measurements. However, new dark and bright field images are taken every few months because the detector accrues imperfections over that time scale.

3.2 Real Space Correction

The objective of this section is to describe the corrections that are applied to real-space images and are implemented in the real space window of the GUI. For most real space images, only a bright and dark field correction is sufficient. For other commonly collected real-space datasets, specifically 3D real space datasets where multiple frames are collected at different times, drift correction is a useful computational tool both for correction and visualization of temporal changes. Drift correction is implemented in my real space user interface and described below.

3.2.1 Thermal Drift Correction

An undesirable behavior that is present in real space images is thermal drift. This occurs when multiple images of the same region on a sample are taken over time, one of the main uses of the PEEM because it captures full-field images. In such a use case, the sample tends to drift with respect to the detector, a small but noticeable distance at time scales near a minute. This effect is due to random thermal movement of the surface and instability of the piezo motors that control the sample stage at distance scales near a μm which results in multi-pixel error in between features in subsequent images. For example, in Figure 9a, the first frame of a dataset is shown that contains 176 PEEM real space images taken each minute. In this dataset the sample is silver intercalated on epitaxial graphene which was heated from room temperature to 580 °K over the image collection time. Figures 9b, c, and d show the same zoomed in region of the detector containing a central silver feature at frame 1, 23, and 110, chosen arbitrarily to highlight drift.

Noticeable discrepancy, on the order of a few pixels, can be seen in the position of the short linear high intensity silver feature in the three zoomed in frames.

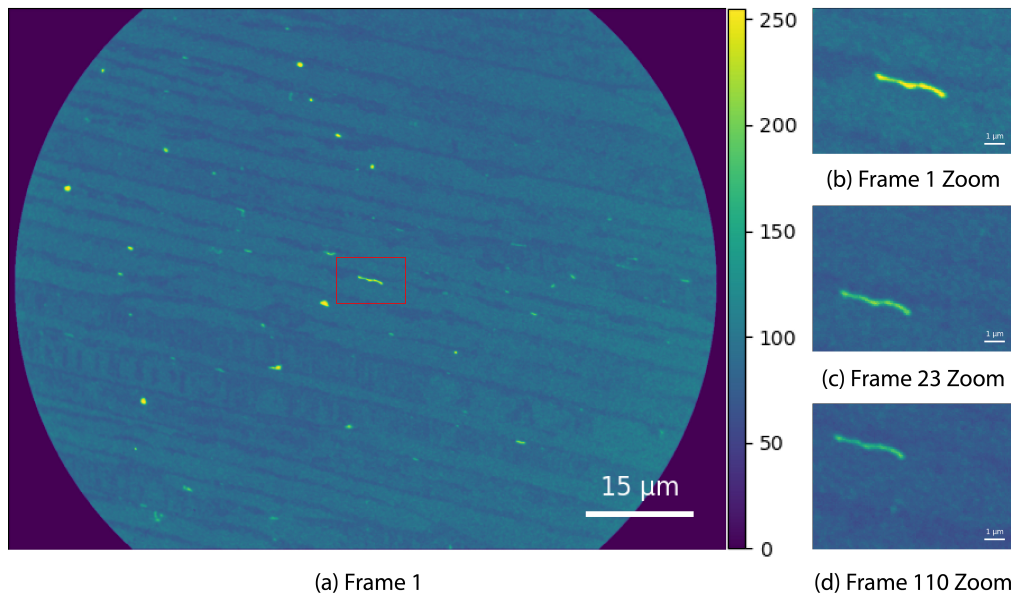


Figure 9: Shows PEEM images of silver intercalated on epitaxial graphene from a dataset of 176 images collected each minute while the sample is heated from room temperature to 580 Kelvin. In (a) the entire first frame of the dataset is shown. The images (b), (c) and (d) show zoomed in images on the red rectangle in (a) at frame 1, 23, and 110. All frames have been bright and dark field corrected.

Drift between images is a common problem that occurs in observational astronomy and multiple fields of microscopy as well as medical imaging [4]. This is a difficult computational problem because the goal of a temporal microscope experiment is to observe a change that the sample undergoes over time. For example, many of the small silver features in the dataset shown in Figure 9 drastically lose intensity when heated. The drift correction algorithm must detect the correct translation to align the images while the sample is temporally changing in proceeding frames. This problem is usu-

ally computationally resolved with image registration, where the root-mean squared error between a reference image and an image collected at a later time is translated in 2D and is minimized with respect to the 2D translation. Image registration is most successful when there are consistent features and only small changes in the two images. A reference image is chosen from the dataset by the user and is registered to every other image in the dataset. Each image is shifted accordingly to align with the reference image. An efficient algorithm that performs image registration with sub-pixel accuracy written by Manuel et al [5]. (2008) has been implemented in the Python module `skimage` and is used in my real space graphical user interface. In my graphical user interface, the user must select both a reference frame and an inner rectangular subsection of the PEEM image to perform the image registration. Only a subsection of each image is used for image registration both to avoid the algorithm aligning the circular outline of each image and to reduce computation time. In the GUI, the selection of the rectangular subsection can be specified with a rectangular slider for ease of use or by inputting the bounding indices. The dataset shown in Figure 9 is corrected using frame 1 as a reference and a large central subimage. Frame 1, 23, and 110 are shown in Figure 10 to compare to the same frames in Figure 9.

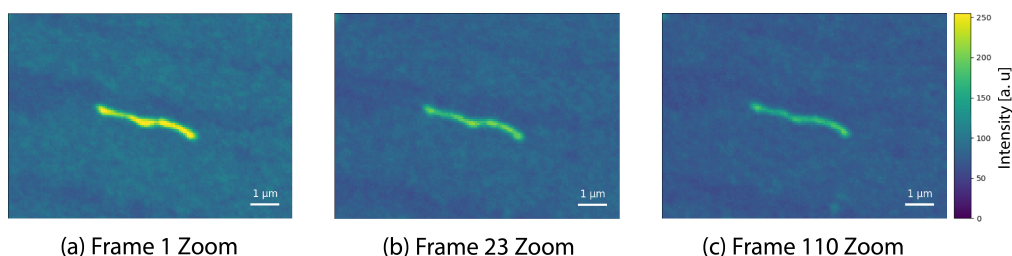


Figure 10: Shows PEEM images of silver intercalated on epitaxial graphene from a dataset of 176 images collected each minute while the sample is heated from room temperature to 580 Kelvin. The dataset is drift corrected and Frames 1, 23, and 110 of the corrected dataset are shown in (a), (b) and (c). The high intensity silver feature is at the same location in each frame, unlike the same frames of the uncorrected dataset shown in Figure 9.

3.3 Momentum Space Correction

The goal in this section is to procedurally describe how a raw ARPES dataset is corrected and calibrated. Six frames of the full 143 frame raw dataset are shown in Figure 5. In one frame, electrons with similar k_x values within a small range have reached the detector. In the next frame, electrons with different values of k_x are selected by moving the entrance slit a small amount that can be specified by the user. In this dataset, each frame corresponds to a $3\ \mu\text{m}$ change in the position of the slit. Each frame is a result from averaging 3 consecutive 1-second exposure images to increase the signal to noise ratio. Important features can be seen in Figure 5c and d: The π and σ bands sit above the bright secondary electron region which is bounded below by the characteristic energy momentum parabola. Secondary electrons are electrons that have inelastically scattered and are therefore do not reflect the band structure of the sample. The conduction band that can be seen in Figure 6 does not show up in the ARPES image because electrons do not occupy those states.

3.3.1 Frame of Reference Correction

Graphene has a distinct band structure that has been studied extensively both theoretically and experimentally. We need to perform corrections such that the final corrected image fits the form of the theoretical and experimental band dispersion diagrams. These corrections also need to make sense with respect to the physical experimental setup. For example, the energy momentum parabola in Figure 5 is rotated with respect to the pixel axis. In other words, the pixels that we know are at the same binding energy are not at the same height and electrons with the same k_y are not at the same horizontal position as we would want. This is a feature of the experimental setup. The camera is rotated with respect to the phosphor plate such that the path that the ARPES image traces out during an experiment is along the

longest region of the detector. To correct for this, a simple image rotation is applied to each iso- k_x frame. The result of this transformation can be seen below in Figure 11.

This correction is implemented in my GUI with a preview plot of a single specified iso- k_x frame where the rotation angle can be updated with a slider. The user can then specify the correct rotation angle and apply this transformation to each frame. The angle of the camera with respect to the channel plate is fixed, so it can be reused for future experiments. This angle is approximately 49.75 degrees in the current experimental setup.

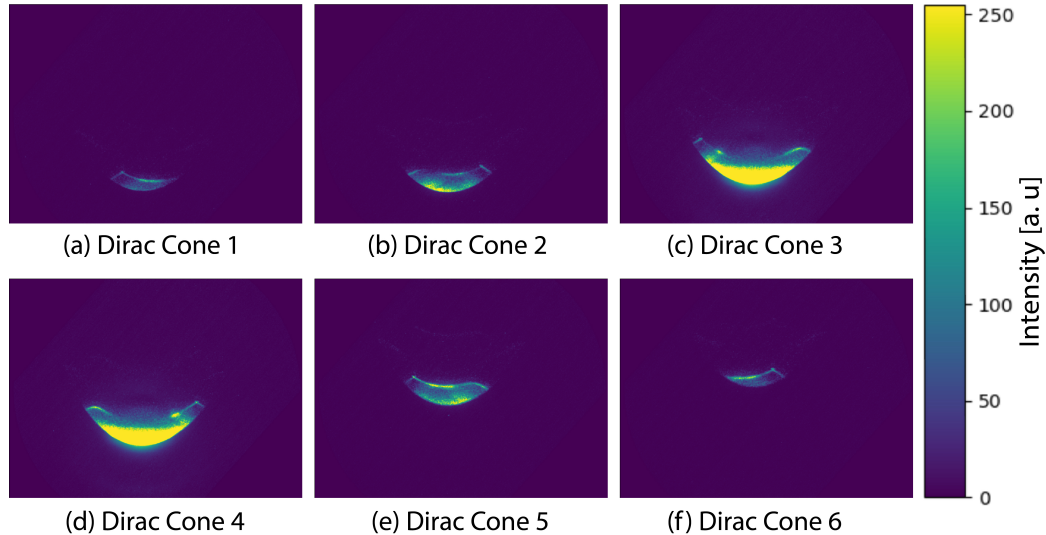


Figure 11: Six frames of 3D ARPES data of graphene. The six Dirac cones are shown at frame 23, 32, 63, 80, 120 and 129 of the 143 frame dataset. Each frame has been rotated counterclockwise by 49.75 degrees to correct for camera misalignment.

After the simple image rotation, the energy direction is parallel to the y-coordinate of each frame. However, there is another problem. According to theoretical band diagram predictions and extensive experimental studies of graphene, the six Dirac points in the first Brillouin zone of graphene are aligned at the same energy, which roughly equals the Fermi energy if un-

doped. In Figure 5 they are not aligned at the same y-pixel value. The Dirac cones slowly shift upwards in y-pixel value in increasing frames. For example, Figure 11a shows the Dirac point at a pixel value of 246 in the y-direction while Figure 11f shows a Dirac point at a y-pixel coordinate of 605, a large disparity that must be corrected for. This discrepancy from the theoretical prediction is due to the entrance slit shown in Figure 3. For each frame, the entrance slit is shifted vertically to allow for a different range of k_x electron momenta. Due to this, in different k_x frames the electrons take a different path through the prism array which changes their final position on the detector. This can also be seen in Figure 4e, where the 5 frames sampled from the energy paraboloid are not aligned with respect to energy. In essence, the green, blue, and purple constant energy lines are not aligned.

To calibrate each frame of the data cube such that energies are consistent, each frame would have to be shifted such that the Dirac points occur at the same y coordinate. For the six frames with Dirac points, it is simple to determine the amount that they should be shifted, but for the majority of frames there is no indication of where the Fermi energy is. However, the entrance slit is shifted vertically by a constant amount specified by the user on the order of a μm . Due to this, there should be a roughly linear relationship between the shift amount of each frame and its frame number. Graphing the frame number and energy pixel coordinate of each Dirac point on the x and y axis respectively yields the graph in Figure 12a. We decided to fit a 2-degree polynomial to the 6 points to interpolate the shift amount for each frame. In practice, we found that adding a second order coefficient on the interpolating polynomial better aligns the Dirac cones noticeably better than a linear curve. For each frame, the 2-degree polynomial is evaluated and the data is shifted accordingly. After this correction, the six Dirac points align at the same energy-pixel coordinate as desired. The result of this correction can be seen in Figure 12b, which shows a k_x and k_y cut of the data cube at the Fermi energy. The polynomial fit in Figure 12a has an R^2 value of

0.99902.

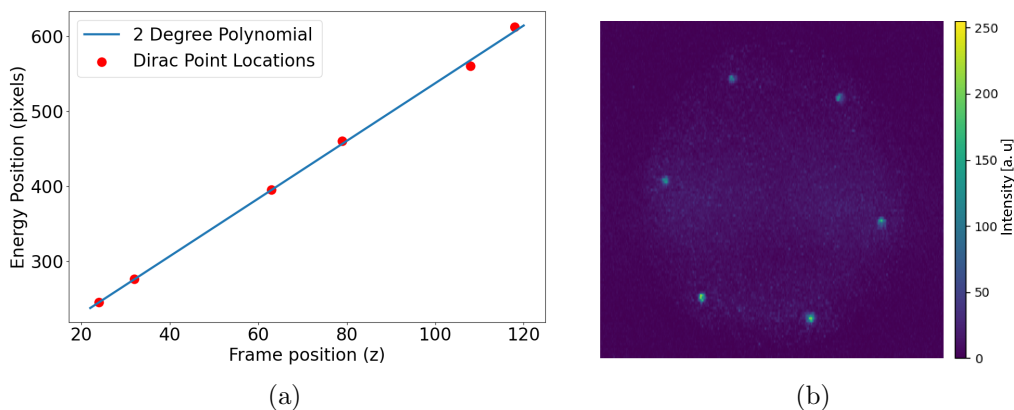


Figure 12: Dirac Cone Positions of the partially corrected dataset shown in Figure 11 are shown as red dots in (a). A 2-degree polynomial is fit to interpolate shift amount for frames without Dirac points. After each frame is shifted according to the polynomial, the Dirac cones line up at the same energy. This can be seen in (b) which shows an iso-energy cut of the corrected dataset at the Fermi Energy with aspect ratio defined by calibration calculations in section 3.3.2.

For graphene, this energy alignment step is a straightforward interpolation problem, but for other samples, it is not so simple. The band diagram of another sample of interest might not have consistent features that occur at the Fermi energy. Additionally, it is important that the correction procedure does not depend on previous knowledge of the sample's electronic properties. A different correction procedure must be implemented for other sample materials. One might think that the slope of the line in Figure 12 can be reused, but even in different ARPES measurements of the same graphene sample but on different days, the slope of the fitted polynomial varies by about 20%. The reason behind this is unknown, but it is clear that the shifting polynomial cannot be reused for subsequent ARPES measurement. This correction step in particular needs improvement and would be the focus of future work on the project. Possible solutions include taking a graphene

ARPES calibration measurement before each measurement of a new sample. In post, the shift polynomial would be calculated using the six Dirac points of the graphene image just like in Figure 12 and the same shift of each frame would be applied to the other sample.

3.3.2 Energy and Momentum Calibration

Now that we have generated a consistent energy axis for the data cube, the next step is calibration. There are three dimensions to the data cube: energy and momentum in two directions, k_x and k_y . A pixel to unit scale must be developed for each dimension.

For the energy calibration, we want to determine the number of pixels in the vertical direction of the detector that correspond to one eV. This can be calculated using a secondary excitation mode of the Helium arc lamp. For most ARPES measurements, including the one in Figures 5, the helium arc lamp is set to the He I excitation mode. In this excitation mode, light is produced with an energy of 21.22 eV or equivalently a wavelength of 58.43 nm which is sufficient for photoemission. In the He II excitation mode, radiation with energy 40.81 eV or a wavelength of 30.38 nm is also produced. These transitions have a known energy difference, 19.59 eV. Taking an ARPES image with the Helium arc lamp in He II mode produces two sets of dispersion curves which are separated by the difference in energy of the two transitions. One frame of a momentum space image of graphene is shown in Figure 13a at $k_x = 0$ that has been taken using the He II excitation mode. In this image, the y-coordinate corresponds to energy and the x-coordinate corresponds to k_y momentum. The π band of the He II transition is above the π band produced by the He I transition. By measuring the number of pixels between the two sets of π bands and using the known difference in energy between the helium transitions, a pixel to eV scale can be determined. Finding the pixel distance between the bands was done by taking multiple one dimensional slices of Figure 13a at different k_y indices near the Dirac points. One of

these one dimensional slices can be seen in Figure 13b. For this slice, the He I and He II π bands occur at energy pixel positions 498 and 263 respectively. Which results in a difference of 235 pixels and corresponds to a change of 13 pixels being equivalent to one eV using the know energy difference between the transitions. Taking six slices around the Dirac points and averaging the estimates, we calculated that 12.46 pixels corresponds to one eV.

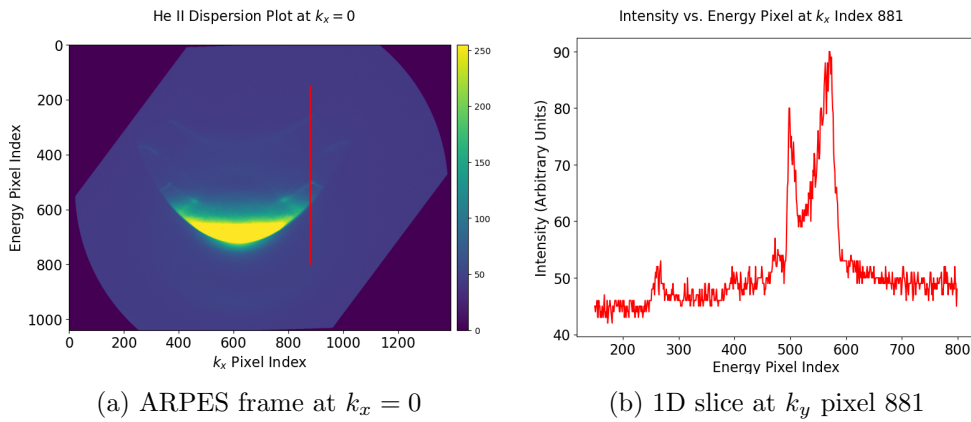


Figure 13: An ARPES image at $k_x = 0$ is shown in (a). The He I and faint He II π bands can be seen which are compared for energy calibration. In (b), a one dimensional slice of the dataset shown as the red line in (a). The small feature around energy pixel 275 is the intersection with the π band for the He II excitation and the peak around energy pixel 500 is the π band for the He I transition.

The next step in calibration is to set the pixel unit spacing of the other two dimensions in our data cube. Both of these dimensions correspond to inverse angstroms, but they have different conversion ratios. The purpose of the rest of this section is calculating these conversion ratios, c_{k_x} and c_{k_y} . They are the ratio of the number of frames in k_x or pixels in k_y that correspond to an inverse angstrom.

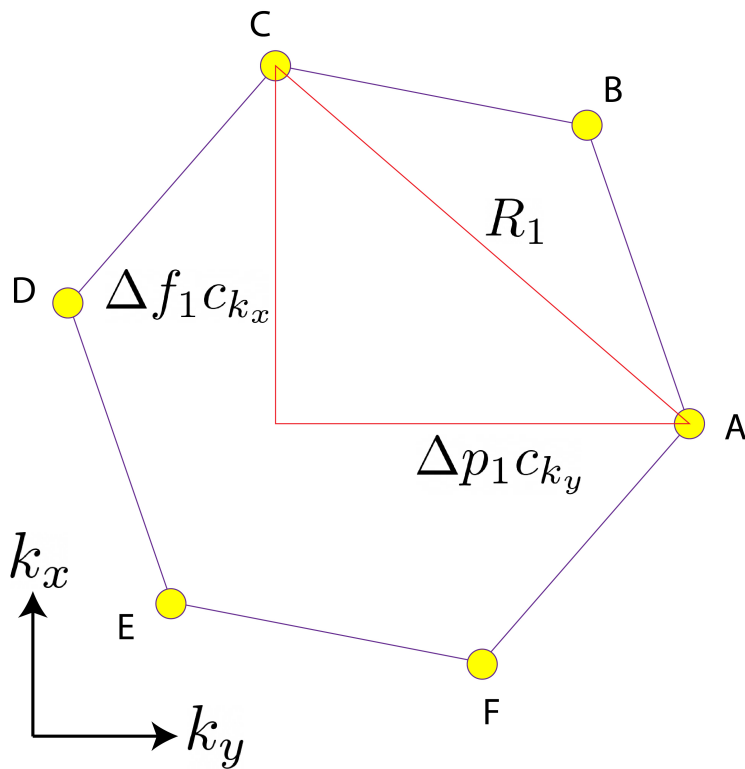


Figure 14: Diagram showing the hexagon formed by the Dirac cones at the Fermi energy which is rotated with respect to the image coordinates. A triangle can be formed for each set of two points where the sides of the triangle are the components of the hypotenuse in the image coordinates. This triangle is used to solve for the conversion ratios.

The slice of the data cube roughly at the Fermi energy shown in Figure 12 will be used for calibration. For graphene, the six Dirac cones form a regular hexagon with side length of 1.8\AA^{-1} [1, 2]. I have labeled each vertex of the hexagon from A to F as shown in Figure 14. For each set of two points in the hexagon, an equation relating c_{k_x} and c_{k_y} can be formulated. For example, points A and C are separated by a distance of R_1 , which is known to be 3.1\AA^{-1} . The different components of this line in the data cube coordinates can be drawn and are shown in Figure 14. The length of these lines are the change

in frames, Δf , or pixels, Δp , multiplied by the corresponding conversion factor. Simply applying the Pythagorean theorem to this right triangle yields the equation $(\Delta f_1 c_{k_x})^2 + (\Delta p_1 c_{k_y})^2 = R_1^2$. Formulating a second similar equation using a different set of points would give us a sufficient amount of information to approximate the two conversion factors. However, I decided to take an approach that utilizes all points. By estimating the position of each Dirac point in pixels and frames, I formed a set of equations describing each pair of points such as the example for point A and C. This set of equations can be represented by a matrix equation shown in equation 3.

$$M\vec{c} = \vec{r}, \quad \begin{bmatrix} \Delta f_1^2 & \Delta p_1^2 \\ \Delta f_2^2 & \Delta p_2^2 \\ \vdots & \vdots \\ \Delta f_n^2 & \Delta p_n^2 \end{bmatrix} \begin{bmatrix} c_{k_x}^2 \\ c_{k_y}^2 \end{bmatrix} = \begin{bmatrix} R_1^2 \\ R_2^2 \\ \vdots \\ R_n^2 \end{bmatrix} \quad (3)$$

Although there are 15 unique sets of two points in a hexagon, I chose to use only the 9 between adjacent points and points directly across from each other in the hexagon for simplicity. The above matrix equation does not have an exact solution in general due to human error in estimating the position of the points on the ARPES image and nonzero resolution. For this reason, the vectors in the A matrix are linearly independent of the \vec{r} vector. I used a least squares solution which minimizes $\|A\vec{c} - \vec{r}\|_2$ to solve for c_{k_x} and c_{k_y} . Three rows of the M matrix and values of \vec{r} are shown below in equation 4.

$$\begin{bmatrix} 4489 & 5929 \\ 63504 & 441 \\ \vdots & \vdots \\ 131044 & 12544 \end{bmatrix} \begin{bmatrix} c_{k_x}^2 \\ c_{k_y}^2 \end{bmatrix} = \begin{bmatrix} 3.24 \\ 3.24 \\ \vdots \\ 12.96 \end{bmatrix} \quad (4)$$

The least squares solution that I obtained was that approximately 142.45 pixels in k_y correspond to one inverse angstrom and 43.82 frames in k_x correspond to one inverse angstrom. with these values, the residual, $A\vec{c} - \vec{r}$, had

a 2-norm of 0.404. Figure 12b is plotted in the aspect ratio calculated in this section.

The result found for c_{k_x} and c_{k_y} can be reused for future measurements in the PEEM using the same lens mode and are taken as a constant in the experimental setup. However, the PEEM has multiple imaging modes. For example, in the dataset collected on graphene for calibration, the distance that the slit that restricts electrons within a small range of k_x shown in Figure 3 moves $2\ \mu\text{m}$ between each frame. If an experimenter wants more or less resolution in k_x , they might want a smaller or larger shift step size. If a different slit shift distance is used, the ratio of frames to \AA^{-1} can be linearly scaled and reused. Similarly, this image was collected using a lens mode of 10 inverse angstroms but using a different lens mode, 5 or 10 inverse angstroms, the ratio of pixels to k_y would scale linearly by 1/2 or 2 respectively. Recalibration using the same procedure might need to be performed every few months or years on the PEEM to maintain accuracy.

3.3.3 Final Correction

In general, when an image is taken on the PEEM, the image axis and the theoretically defined k'_x and k'_y axis are not aligned, differing by an arbitrary angle. The k'_x vector is usually taken to be arm-chair direction and the k'_y direction is defined as the zig-zag direction of the hexagonal lattice [9]. In Figure 12b, the hexagon formed by the Dirac cones and the theoretically defined basis vectors are off-angle with respect to the image. This could be avoided in the experimental setup by adding a rotation functionality to the sample stage to align the Brillouin zone. This is not a current feature of the PEEM but we can rotate it in post.

Image rotation is well studied and implemented in many programming languages. To rotate an image, each pixel is represented with a position vector, \vec{r} , and an intensity value. Each pixel's vector is then rotated by an angle θ by multiplying it with a rotation matrix, R_θ .

$$R_\theta = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}, \quad R_\theta \vec{r} = \vec{r}'$$

An interpolation algorithm is then used to approximate the pixel values on the original image's pixel grid because for an angle that is not a multiple of $\pi/2$ radians, the \vec{r}' grid does not coincide with the \vec{r} grid. Bilinear or spline interpolation is often employed [3]. This is the algorithm used to rotate each frame of data to go from Figure 5 to Figure 11.

However, applying this simple image rotation to the dataset shown in Figure 12b distorts the image and a regular hexagon of Dirac points is no longer maintained. The results of this basic rotation method can be seen in Figure 15a. This distortion is a result of the image having anisotropic resolution. For example, in the previous section, we determined that 142.45 pixels in the k_x direction correspond to an \AA^{-1} but in the k_y direction, only 43.82 pixels corresponds to an \AA^{-1} . With the above image rotation algorithm, a square lattice of pixels is used for \vec{r} . In this rotation, a rectangular non-square lattice of pixels correctly describes the data due to its anisotropic resolution. We can generalize our rotation procedure for this rotation step by multiplying by a different rotation matrix, $R_{\theta,c}$ which is the product of three matrices defined below. This matrix first scales one coordinate of each pixel vector by, c , the ratio of the \AA^{-1} calculations such that they form the correct rectangular lattice in reciprocal space. Then, the rotation vector R_θ is multiplied and finally each vector is scaled back to its original pixel spacing with the inverse of the first diagonal matrix.

$$R_{\theta,c} = \begin{bmatrix} 1 & 0 \\ 0 & 1/c \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & c \end{bmatrix}$$

This was implemented using scipy's Affine transform function. A spline interpolation algorithm is then used to calculate the intensity values on the original image pixel lattice. Using this generalized rotation algorithm yields

Figure 15b, which preserves the regular hexagonal geometry the Dirac points. This is the final correction that was implemented for an ARPES dataset.

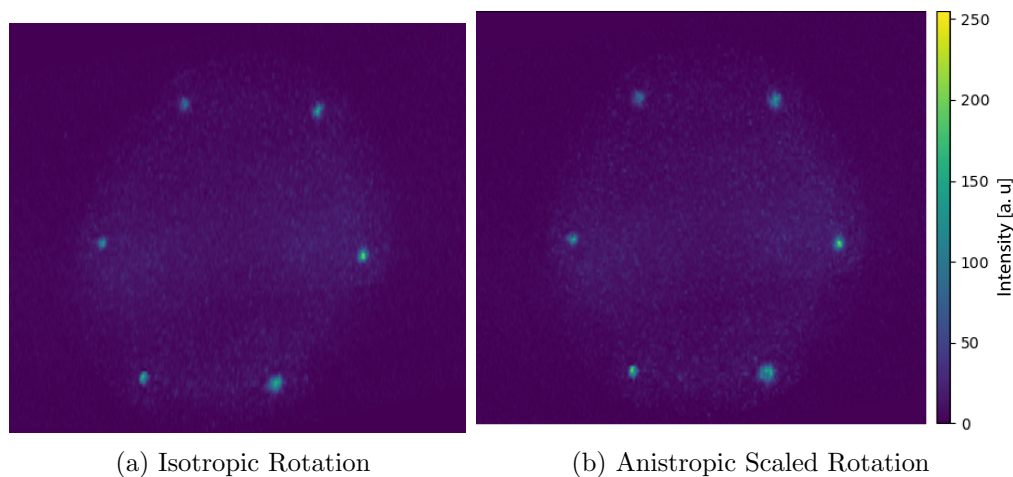


Figure 15: Two different image rotations from Figure 12b are shown. In (a), a naive image rotation is used where the pixel grid is incorrectly assumed to be square. Distortion occurs and the regularity of the Dirac point hexagon is lost. In (b), a more general image rotation algorithm is used that scales the pixels onto a rectangular grid before rotation. This method preserves the regular hexagon.

This rotation algorithm can be applied on any other ARPES dataset that have azimuthal energy misalignment with the detector if the ratio c is known. The constant c is identical when the PEEM is in the same imaging mode, and is only scaled by a constant if in a different imaging mode. The sample is, in general, randomly aligned by an angle θ , which changes for each dataset. For this reason, the angle must be specified by the user each time. In my graphical user interface, this is implemented in a popup window where a single energy frame is rotated with a slider. The user can select the correct angle of rotation either manually by aligning the image axis with the hexagon, or using trigonometric identities to calculate the correct angle.

This rotation is then applied to every energy frame, similar to how the first rotation described in Section 3.3.1 is implemented.

It is useful to visualize the final corrected graphene datacube from multiple axis. Figure 16 shows three images of the corrected dataset at the Dirac cones. we can see that the theoretical perpendicular k_x and k_y vectors align well with the corresponding images axis because two Dirac cones appear in each image.

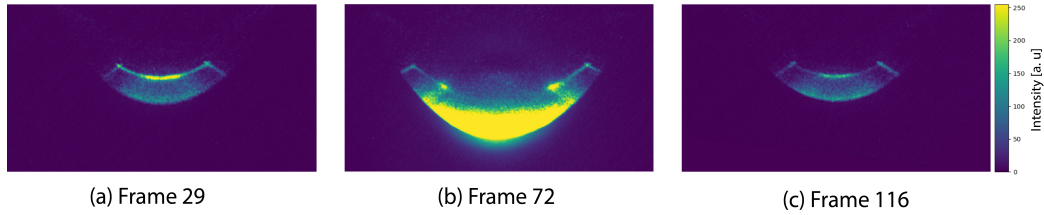


Figure 16: The corrected dataset is depicted with three frames of the datacube, each frame containing electrons with the same k_x momentum. Frame (b) shows the $k_x \approx 0$ inverse angstrom slice in the center of the Brillouin zone while (a) is at $k_x \approx -1.01$ inverse angstroms and (c) is at the $k_x \approx 0.99$ inverse angstrom frame.

4 Results and Discussion

4.1 Fermi Velocity

As a metric to check if the corrections that were performed on the ARPES dataset were accurate, I estimated the Fermi velocity of the epitaxial graphene sample from our results. The definition of Fermi velocity is shown in Equation 5 near the Fermi energy.

$$v_f = \frac{1}{\hbar} \frac{dE}{dk} \quad (5)$$

For graphene, there is a roughly linear dispersion relationship, which can be seen easily in Figure 16b. This results in a constant Fermi velocity.

According to theoretical predictions by the tight binding model, the Fermi velocity should be about 10^6 m/s [2]. To estimate the Fermi velocity for this graphene dataset, I used the frame of the corrected dataset at $k_x = 0$ because the Dirac cones in this frame are the most visible. The slope of the π bands around the Dirac point was estimated by placing four points following the π band on either side of each Dirac cone. For each set of four points I fit a linear line of best fit. The points and lines of best fit for the right Dirac cone can be seen in Figure 17. The slope of these lines were used to calculate the Fermi velocity twice at both of the Dirac cones in the image using Equation 5. Taking the average of the four estimated Fermi velocities, a value of $(1.25 \pm 0.2) \times 10^6$ m/s was obtained which roughly matches the theoretical prediction. Uncertainty values in Energy and k_y were propagated to calculate the uncertainty in the measurement.

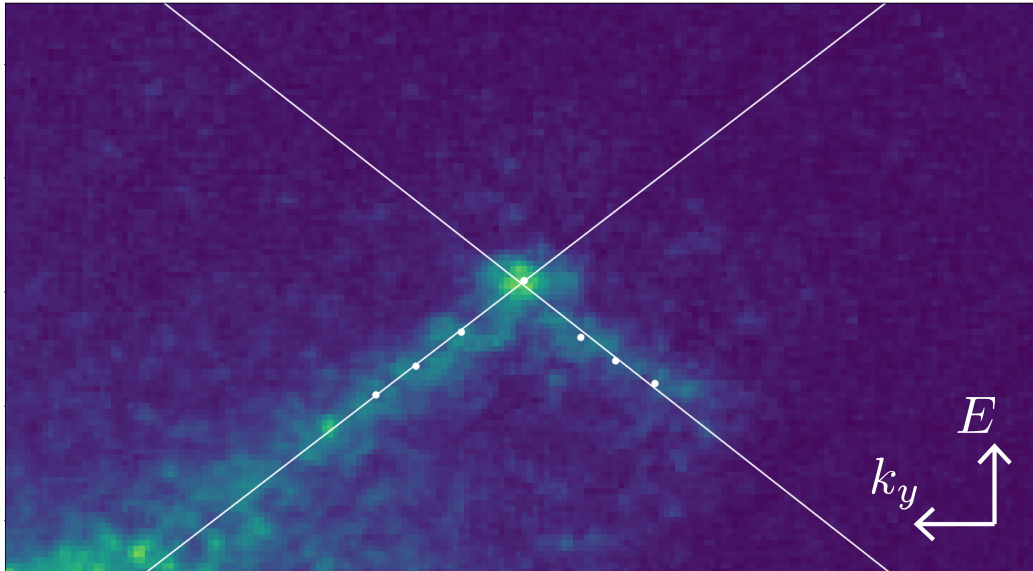


Figure 17: Momentum space frame at $k_x \approx 0$ zoomed into the right Dirac point. Four points including the Dirac point were used on either side of the Dirac cone to estimate the linear dispersion relation and calculate the Fermi velocity of graphene.

4.2 Graphical User Interface

The culmination of this project is the GUI that functions to correct data collected on the PEEM. This section will be dedicated to describing in more detail the various functionality of the GUI and how the correction procedures in the previous chapter interface with the user for streamlined correction of PEEM images.

4.2.1 Real Space

An image of the real space GUI is shown in Figure 18. In the top left, the user can select any number of real space images to correct. The GUI accepts .bmp and .tif files. Once they have been corrected, they can be saved, as .bmp or .tif files, to a specified folder with the button below. For each file that is saved, a metadata text file is also saved that contains information about the corrections that were performed for documentation purposes. For visualization, a large image is shown to the right. Other loaded images can be selected using the "next" and "last" keys above the visualization frame. Below the plot, zoom, and pan buttons are implemented to help with visualization. A save button is also included that saves the currently viewed figure as a .jpg image. In the scrolling window that shows each file name, the diameter of each image can be input to activate a scale bar, an example of which is shown in the image displayed in the bottom right of the PEEM image of Zinc Oxide in Figure 18.

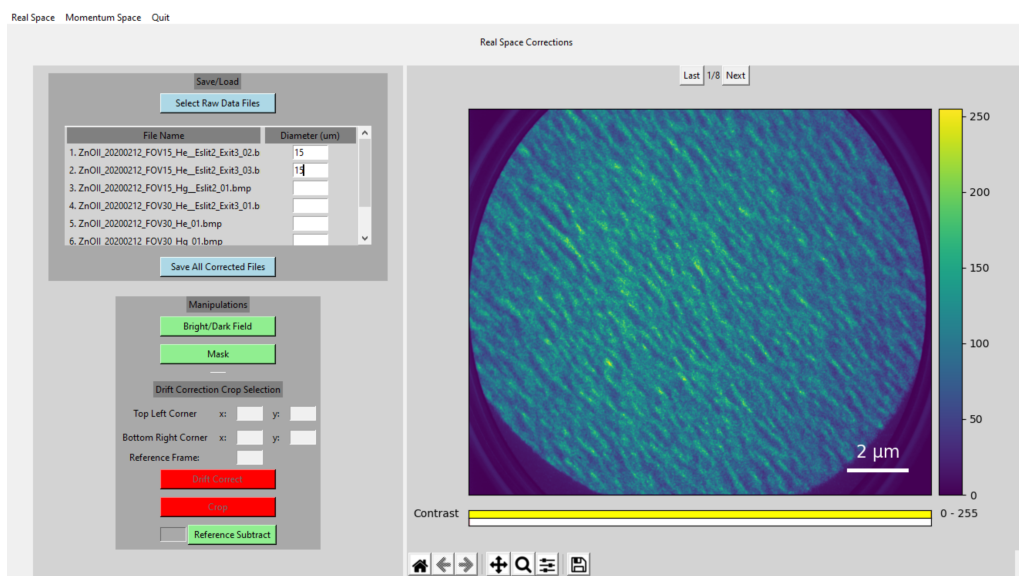


Figure 18: This figure shows a screenshot of the real space GUI. Real space images of Zinc Oxide have been loaded into the GUI for correction.

Correction procedures can be accessed in the "manipulation" window below the "save/load" window. The corrections are arranged in likely order of operations from top to bottom. With the first button, bright and dark field corrections can be applied to every loaded image. This correction uses bright and dark field images from directories specified by the user in a portion of the GUI not shown in Figure 18. A masking filter is often applied to the images which sets the pixels outside the circular window to zero intensity. At this point, most real space images are fully corrected and can be saved to a folder identified by the user. If the dataset must be drift corrected, that can be completed by selecting the subimage for image registration by dragging the cursor over the image. Once a reference frame is chosen, the drift correct button can be pressed which executes the algorithm defined in the thermal drift section. The crop button below cleans up the edges of the drift corrected dataset. The GUI was designed to be modular and more recently, a reference subtract feature was added that subtracts a specified

reference frame from every other frame in a dataset. This makes it easier to distinguish temporal changes in the dataset.

4.2.2 Momentum Space

For the correction of momentum space images on the PEEM, a similar GUI was coded, a screenshot of which is shown in Figure 19. The $k_x \approx 0$ frame of the corrected dataset is displayed. The user can easily swap between the real and momentum space GUIs with buttons in the top left corner of the window. The layout of the momentum space GUI is very similar to its real space counterpart, with a large visualization window where the frame number and contrast can be adjusted surrounded by windows that load, perform correction or help visualize the data. In the top left, an ARPES image can be loaded into the GUI and once it has been corrected it can be saved with the button below. Similar to the real space GUI, a metadata text file is also saved that documents the exact corrections performed. In the top right, the lens mode of the ARPES image must be input by the user which is important to determine the pixel conversion ratios and the c constant for the final energy rotation. In the window below, the user can choose which axis to view the data: iso- k_x , iso- k_y or iso-energy. The large visualization image is controlled by the blue slider below to scrape through each frame of the 3D datacube. Zoom, pan, and save features are available below the contrast sliders.

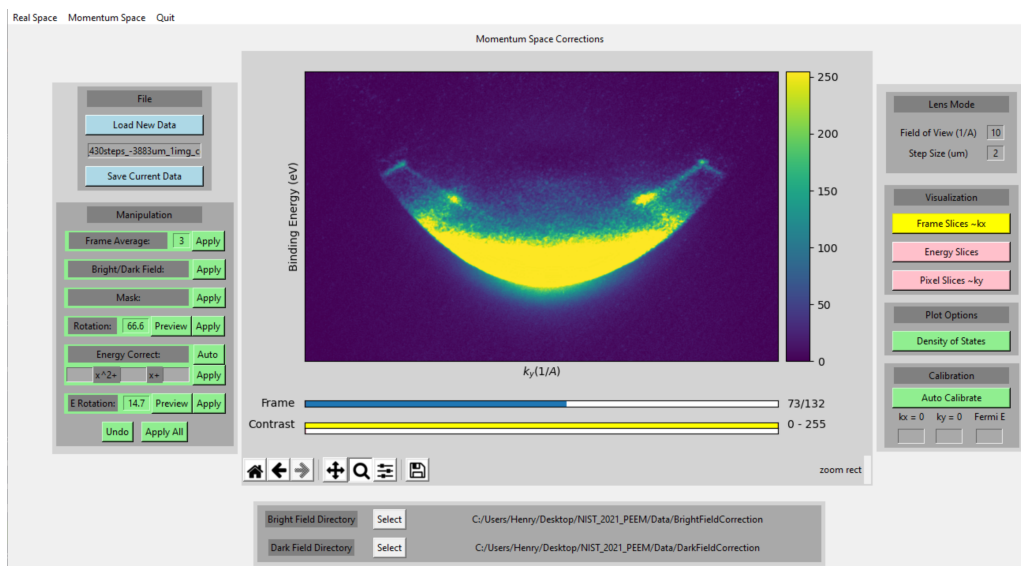


Figure 19: A screenshot of the momentum space GUI. The dataset loaded into the GUI is the corrected Graphene dataset displayed in Figure 16.

In the "Manipulations" window below the "File" window, various correction buttons are placed in procedural order. First, to increase the signal to noise ratio, a frame average button is implemented which averages every n images, where n is input by the user. Typically three images are taken at one second exposure time at the same frame position so $n = 3$ is used. The user can then apply dark and bright field corrections to each frame of data and mask the outer part of the image. The next correction to be performed is the rotation to correct for camera-screen misalignment. A preview button is available for this rotation that opens a separate window where a single frame of the dataset can be rotated by a slider. Once the frame is rotated to the correct angle, the apply button will rotate every frame accordingly. The energy alignment correction is available by pressing the next button. The user must specify the coefficients of the 2 degree shift polynomial to perform this correction. After the image is aligned in the energy axis, the final rotation step can be performed which is implemented with a separate window in

the same way as the first rotation. Additional helpful features are included in this window: An undo and apply all button. The undo feature reverts the dataset to its state before the last correction. The apply all button calls every corrections function in order from top to bottom.

4.3 Future Work

The GUI can fully correct an ARPES dataset of graphene in a few minutes of computation time. However, many advancements can be made to generalize the procedure to other materials' band structure. The energy correction step is a point of focus for future work because other samples do not, in general, have distinct features in their band structure that can be used to align the energy of the datacube. It would be ideal if the polynomial that is evaluated to determine the amount to shift each image was consistent between datasets and could be scaled linearly in different imaging modes. However, the coefficients of the polynomial differ to a small but noticeable extent even for two datasets of identical graphene samples collected in the same imaging mode but on two different dates. This means that it is unlikely that the same polynomial can be reused for every dataset so it must be generated each time. There are a few potential solutions to this problem. This could be solved experimentally by imaging a graphene sample immediately before each measurement of a new sample in the same imaging mode. By using the position of the six Dirac cones of the graphene image, the polynomial coefficients could be calculated. Once the graphene dataset has been corrected, the desired dataset could be loaded into the GUI and the same corrections could be applied through the apply all button. The staging in the PEEM does allow for this type of measurement procedure. Another approach would be a computational solution that does not rely on nice features in the band structure to align each frame. One reasonable idea would be to align the secondary electron paraboloid according to the theoretical equation, $E = \frac{\hbar^2 k^2}{2m}$. It would be simple to automatically detect the lower boundary of the bright

secondary electron region on a discrete k_x and k_y grid because the secondary electron region is near maximum intensity. This could be done by looping through the indices in energy from the bottom of the image at each k_x , k_y grid point and detecting the first pixel that has intensity above a tolerance. These automatically detected points could be used to find the coefficients of the shift polynomial with least-squares fitting. As far as computation time, the full correction procedure takes a few minutes to complete. The image rotation steps are by far the most computationally intensive part of the correction procedure. With the current implementation, each 2D frame of the dataset is rotated one at a time by the processor. Such an algorithm lends itself well to parallelization which would reduce computation time considerably and would be a useful feature to implement in the future.

5 Cited Work

- [1] Aaron Bostwick et al. “Experimental studies of the electronic structure of graphene”. In: *Progress In Surface Science* 84.11-12 (Nov. 2009), pp. 380–413. DOI: 10.1016/j.progsurf.2009.08.002.
- [2] A. H. Castro Neto et al. “The electronic properties of graphene”. In: *Rev. Mod. Phys.* 81 (1 Jan. 2009), pp. 109–162. DOI: 10.1103/RevModPhys.81.109. URL: <https://link.aps.org/doi/10.1103/RevModPhys.81.109>.
- [3] Per-Erik Danielsson and Magnus Hammerin. “High-accuracy rotation of images”. In: *CVGIP Graph. Model. Image Process.* 54 (1992), pp. 340–344.
- [4] T.A. de Jong et al. “Quantitative analysis of spectroscopic low energy electron microscopy data: High-dynamic range imaging, drift correction and cluster analysis”. In: *Ultramicroscopy* 213 (2020), p. 112913. ISSN: 0304-3991. DOI: <https://doi.org/10.1016/j.ultramic.2019.112913>. URL: <https://www.sciencedirect.com/science/article/pii/S0304399119302475>.
- [5] Manuel Guizar-Sicairos, Samuel Thurman, and James Fienup. “Efficient subpixel image registration algorithms”. In: *Optics letters* 33 (Feb. 2008), pp. 156–8. DOI: 10.1364/OL.33.000156.
- [6] F. Niefind et al. “Imaging and measuring the electronic properties of epitaxial graphene with a photoemission electron microscope”. In: *Journal of Applied Physics* 131.1 (2022), p. 015303. DOI: 10.1063/5.0076145. eprint: <https://doi.org/10.1063/5.0076145>. URL: <https://doi.org/10.1063/5.0076145>.
- [7] K. S. Novoselov et al. “Electric Field Effect in Atomically Thin Carbon Films”. In: *Science* 306.5696 (2004), pp. 666–669. DOI: 10.1126/science.1102896.

- [8] K. S. Novoselov et al. “Room-temperature quantum Hall effect in graphene”. In: *Science* 315.5817 (2007). DOI: 10.1126/science.1137201.
- [9] Peter Puschnig and Daniel Lüftner. “Simulation of angle-resolved photoemission spectra by approximating the final state by a plane wave: From graphene to polycyclic aromatic hydrocarbon molecules”. In: *Journal of Electron Spectroscopy and Related Phenomena* 200 (2015). Special Anniversary Issue: Volume 200, pp. 193–208. ISSN: 0368-2048. DOI: <https://doi.org/10.1016/j.elspec.2015.06.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0368204815001309>.
- [10] Thomas Seyller et al. “Epitaxial Graphene: A New Material”. In: *physica status solidi (b)* 245 (July 2008), pp. 1436–1446. DOI: 10.1002/pssb.200844143.
- [11] R M Tromp et al. “A simple energy filter for low energy electron microscopy/photoelectron emission microscopy instruments”. In: *Journal of Physics: Condensed Matter* 21.31 (July 2009), p. 314007. DOI: 10.1088/0953-8984/21/31/314007. URL: <https://doi.org/10.1088/0953-8984/21/31/314007>.
- [12] R.M. Tromp et al. “A new aberration-corrected, energy-filtered LEEM/PEEM instrument II. Operation and results”. In: *Ultramicroscopy* 127 (2013). Frontiers of Electron Microscopy in Materials Science, pp. 25–39. ISSN: 0304-3991. DOI: <https://doi.org/10.1016/j.ultramicro.2012.07.016>. URL: <https://www.sciencedirect.com/science/article/pii/S0304399112001866>.
- [13] R.M. Tromp et al. “A new aberration-corrected, energy-filtered LEEM/PEEM instrument. I. Principles and design”. In: *Ultramicroscopy* 110.7 (2010), pp. 852–861. ISSN: 0304-3991. DOI: <https://doi.org/10.1016/j.ultramicro.2010.03.005>. URL: <https://www.sciencedirect.com/science/article/pii/S0304399110000835>.

- [14] Philip R. Wallace. “The Band Theory of Graphite”. In: *Physical Review* 71 (1947), pp. 622–634.
- [15] Yanfei Yang et al. “Epitaxial graphene homogeneity and quantum Hall effect in millimeter-scale devices”. In: *Carbon* 115 (2017), pp. 229–236. ISSN: 0008-6223. DOI: <https://doi.org/10.1016/j.carbon.2016.12.087>. URL: <https://www.sciencedirect.com/science/article/pii/S0008622316311642>.
- [16] Yi Zhang, Luyao Zhang, and Chongwu Zhou. “Review of Chemical Vapor Deposition of Graphene and Related Applications”. In: *Accounts of Chemical Research* 46.10 (2013). PMID: 23480816, pp. 2329–2339. DOI: [10.1021/ar300203n](https://doi.org/10.1021/ar300203n).