The use of Ptychographical methods to analyse stacking faults

Javed Choudhoury

University College London

Department of Physics and Astronomy

31st March 2011

1st Supervisor: Prof. Ian Robinson

2nd Supervisor: Dr. Paul Warburton

Contents

0. Acknowledgments

I would like to thank Professor Ian Robinson for his supervision and guidance and Jesse Clark for his patience in assisting with various Matlab® issues.

1. Introduction

Nanowires have tremendous potential in the future to become part of the next generation of computing devices. Transistors based on nanowire technology have been shown to be three to four times speedier than their silicon counterparts [1].

Nanowires have typical diameters from 2nm up to 100nm. Gallium Phosphide nanowires were studied and were shown to develop stacking faults as they were synthesized [2]. The stacking faults within nanowires can drastically change their properties. Properties such as resistance and durability decrease as the number of stacking faults increase[3].

One way to look at these faults within nanowires is to use Coherent X-Ray Diffraction Imaging. This method makes use of coherent X-Rays to study nanoscale structures. These nanowires have been looked at previously and their structures have been modelled using parallel tempering[4]. This method has revealed various details about the structure of the Gallium Phosphide wire.

A problem with parallel tempering and other traditional phase retrieval algorithms is that the final solution is not unique[5]. Ptychography is the name given to an algorithm to reconstruct the object from its diffraction patterns. It provides a powerful solution to the problem of phase retrieval and uniqueness. It was looked at as a possible solution to finding a unique solution for a given set of nanowire diffraction patterns.

2 Theory

2.1 Diffraction

Typical crystal spacings are of the order 10^{-10}m or 1Å. Such distances and structures may be probed by X-Rays, whose wavelengths are of the same scale as these spacings (0.1-100 Å). Bragg diffraction is the term given to describe diffraction from a crystal in this manner. [6]

Incoming X-Ray radiation is either scattered by a lattice plane or passed through a layer where it may then be scattered by the next layer. Two scattered waves can interact and undergo constructive and destructive interference due to difference in path length between the planes.

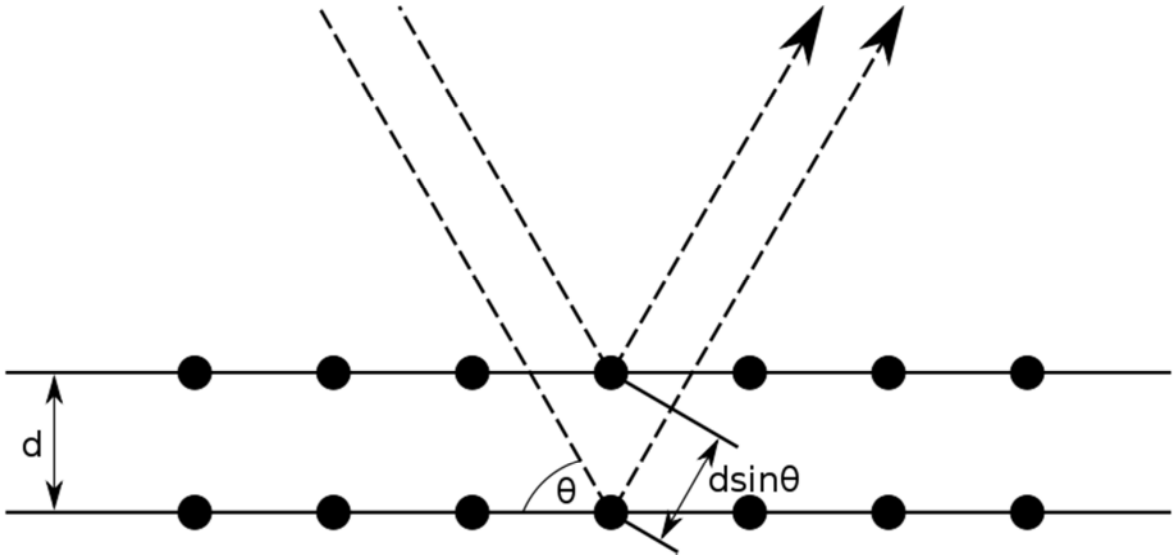


Figure 1: Bragg Diffraction[7]

For lattice planes separated by a distance, d, and an incoming radiation wavelength of λ at angle θ one can derive the Bragg condition. Constructive interference occurs when the path difference between the two beams is equal to an integer multiple of the incoming radiation. This is the basis of Bragg's law.

$$2d \sin\theta = n\lambda$$

Equation 1: Bragg's Law[6]

This law holds true even for larger and more complicated crystals. The diffraction pattern is related to the object under consideration via Fourier transform of the specific view.

2.2 Fourier Transform

The Fourier transform relates the object in real space to its diffraction pattern in reciprocal or momentum space. It is defined as[8]:

$$\tilde{A}(\vec{q}) = \int_V \rho(\vec{r}) e^{-2\pi i \vec{k}.\vec{r}} = FT[\rho(\vec{r})]$$

Equation 2: The Fourier Transform

$\tilde{A}(\vec{k})$ is the scattered complex amplitude. $\left|\tilde{A}(\vec{k})\right|^2 = I(\vec{k})$, the intensities and $\rho(\vec{r})$ is the electronic density or the 'shape' of the sample. The inverse operation is known as the inverse Fourier transform and is defined as:

$$\rho(\vec{r}) = \int_V \tilde{A}(\vec{q}) e^{2\pi i \vec{k}.\vec{r}} = FT^{-1}[\tilde{A}(\vec{q})]$$

Equation 3: The Inverse Fourier Transform

2.2.1 Discrete Fourier Transform

The Fourier transform is designed to operate on continuous functions and in order to analyse data using computers the data must be discrete and finite in length. The discrete Fourier transform is defined as the following:[9]

$$\tilde{A}(\vec{q}) = \frac{1}{N}\sum_{m=1}^{N} \rho(\vec{r}) e^{-\frac{2\pi i \vec{k}.\vec{r}}{N}} = DFT[\rho(\vec{r})]$$

Equation 4: The Discrete Fourier Transform

Like the Fourier transform, the Discrete Fourier transform has an inverse:

$$\rho(\vec{r}) = \sum_{m=1}^{N} \tilde{A}(\vec{q}) e^{\frac{2\pi i \vec{k}.\vec{r}}{N}} = DFT^{-1}[\tilde{A}(\vec{q})]$$

Equation 5: The Inverse Discrete Fourier Transform

The discrete transform views real space and Fourier space as periodic in nature

2.2.2 Sampling Requirements

The relationship between a real space step of Δx and a Fourier step size of Δq and the number of data points N is:

$$\Delta x \Delta q = \frac{2\pi}{N}$$

Equation 6: Relationship between sample step size, reciprocal step size and number of data points

If the number of data points is too low than aliasing can occur. This is a phenomenon where two samples are indistinguishable when they are sampled. If this occurs it is impossible to reconstruct the original sample. This can be avoided by sampling at an interval that is least twice the frequency of the highest frequency component in Fourier space[10]. This is known as the Nyquist–Shannon sampling theorem.

2.2.3 Examples of Fourier Transform Operations

A simple rectangle function with a width of 2a was simulated in the Matlab® software package.

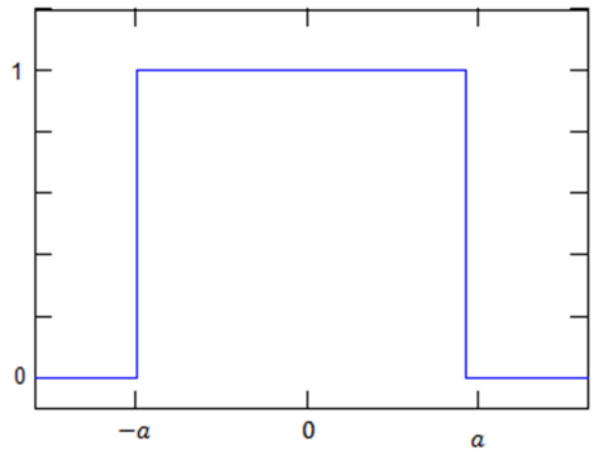


Figure 2: The rectangle function

Using equation 2 the expected Fourier transform can be calculated.

$$\int_{-a}^{a} \rho(x) e^{2\pi i \vec{k}.\vec{r}} = \int_{-a}^{a} 1.\, e^{ik.x} dx = \left[\frac{e^{ik.x}}{ikx}\right]_{-a}^{a} = \frac{\sin kx}{kx} = \operatorname{sinc} kx$$

Equation 7: Calculating the Fourier transform

The calculated form is indeed the Sinc (x) function.

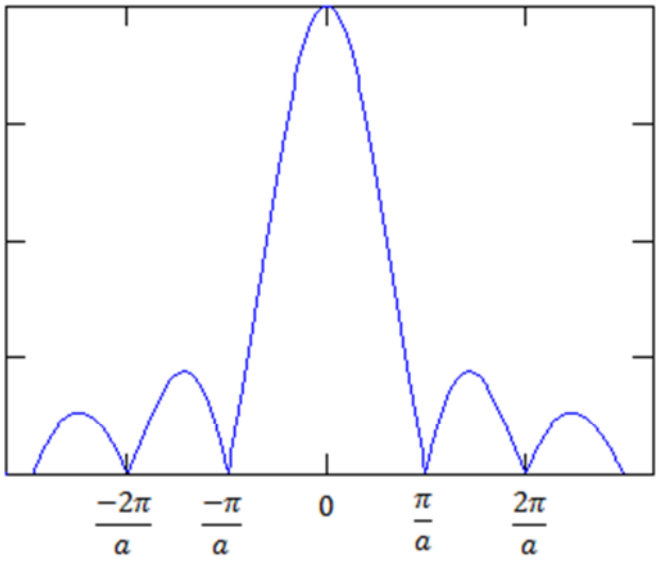


Figure 3: The Sinc (x) function

This Fourier transform is the same as the diffraction pattern that would be recorded from a single slit with a monochromatic source.

3. Experimental Methods

3.1 Nanowire Synthesis

The nanowires used in the experiment were composed of Gallium Arsenide (GaAs) at the top and Gallium Phosphide (GaP) at bottom and had an average diameter of 50nm. They were synthesised using the Vapour-Liquid-Solid (VLS) mechanism.

VLS is a mechanism used to grow one-dimensional structures from Chemical Vapour Deposition (CVD). VLS differs from the generally slow CVD by introducing a catalytic liquid alloy. This alloy absorbs the constituent semiconductor vapours to saturation and crystal growth occurs at the catalyst-substrate interface, the direction is dependent on the orientation of the substrate.

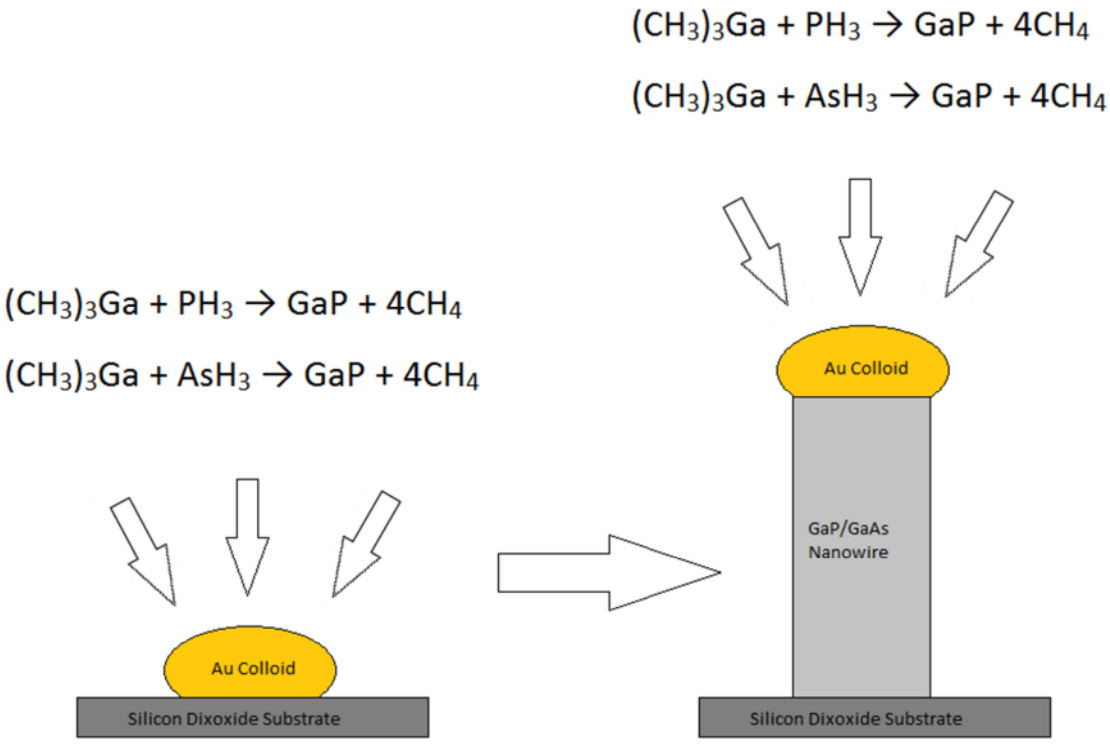


Figure 4: Schematic illustration of GaP/GaAs nanowire growth from reaction of constituent vapours

Initially gold colloids are dispersed onto the Silicon substrate. These colloids determine the radius of the resultant nanowire and so the diameters of the colloids are ~20nm. Without the gold catalyst the vapour mixtures must be heated to >800°C; the presence of the catalyst allows the reaction to proceed at ~420°C. The nanowires were synthesized by heating the substrate with Phosphine (PH_3) present, when the target temperature was reached of Trimethylgallium, $(CH_3)_3Ga$, was added to the mixture allowing the growth of a GaP section. When the desired length was reached, the Phosphide source was switched off and Arsine (AsH_3) was added to the mixture. This was alternated as required [2].

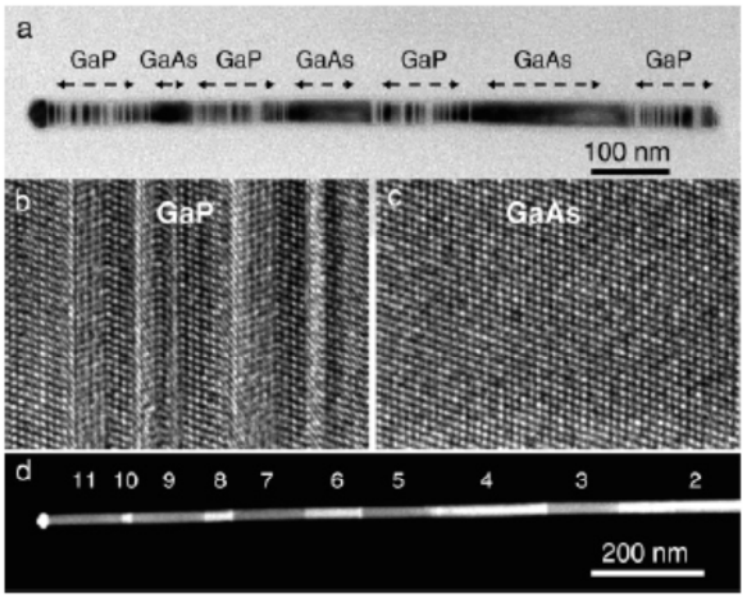


Figure 5: Transmission Electron Microscopy image [2].

Figure 5 shows the GaP sections of the wire have various stacking faults.

The Nanowires in this experiment consisted of GaP at the bottom and GaAs at the top and were grown on a Silicon Dioxide substrate.

3.1.1 Stacking Faults

Stacking faults in nanowires arise for a variety of reasons. One such reason is that the material in bulk has a different crystal structure to that in the nanowire regime[11]. In this case, GaAs in the bulk is most stable in the zincblende configuration, whereas in the nanowire regime it may take up zincblende or wurtzite[12]. When nanowires with stacking faults are subject to X-Ray diffraction they do not exhibit the expected diffraction patterns, which would be similar to a 2D version of the diffraction pattern shown in Figure 3. Instead a pattern resembling a bar code is seen[4].

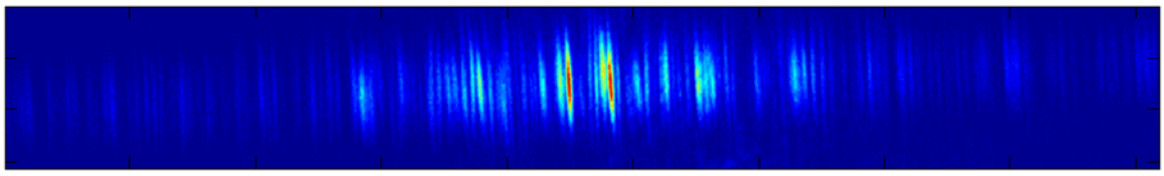


Figure 6: Bar-Code type pattern

3.1.2 Uses of nanowires

Nanowires have a variety of potential uses, potentially replacing traditional silicon CMOS devices[13]. Researchers at Harvard University have made Germanium/Silicon core/shell nanowires which had a performance advantage “of three to four times” over silicon devices.

These devices could be inserted into the next-generation of high-performance logic circuits as CMOS devices approach their limits.

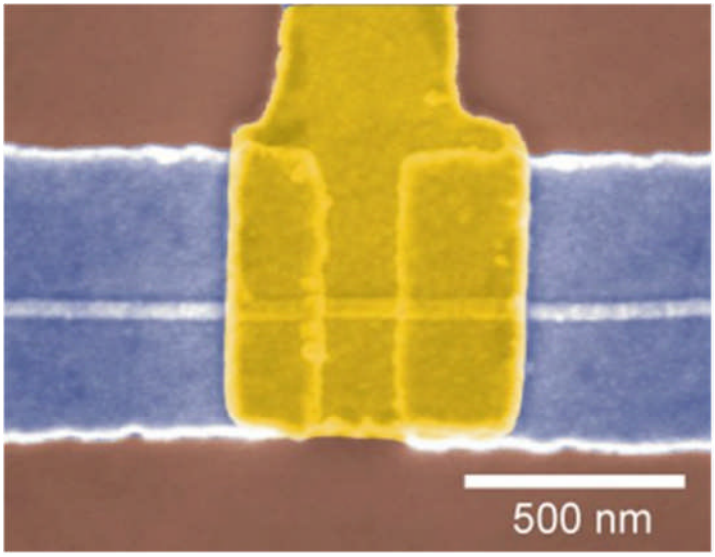


Figure 7: The nanowire rests on a SiO_2 (brown) substrate, with gate electrode (gold) in the middle and source/drain (purple) electrodes on the edges. Image courtesy of Harvard University

3.2 Coherent X-Ray Diffraction

Coherent X-Ray Diffraction is the use of Coherent X-Rays to image nanoscale objects. Such X-Rays are produced in Synchrotrons such as Diamond, Advanced Photon Source, Swiss Light Source and many more around the world.

The synchrotrons produce radiation when electrons travel at relativistic speeds and are forced to oscillate as they travel through undulators. The undulator consists of a series of alternating dipole magnets and as the electrons oscillate between the magnets they radiate energy.

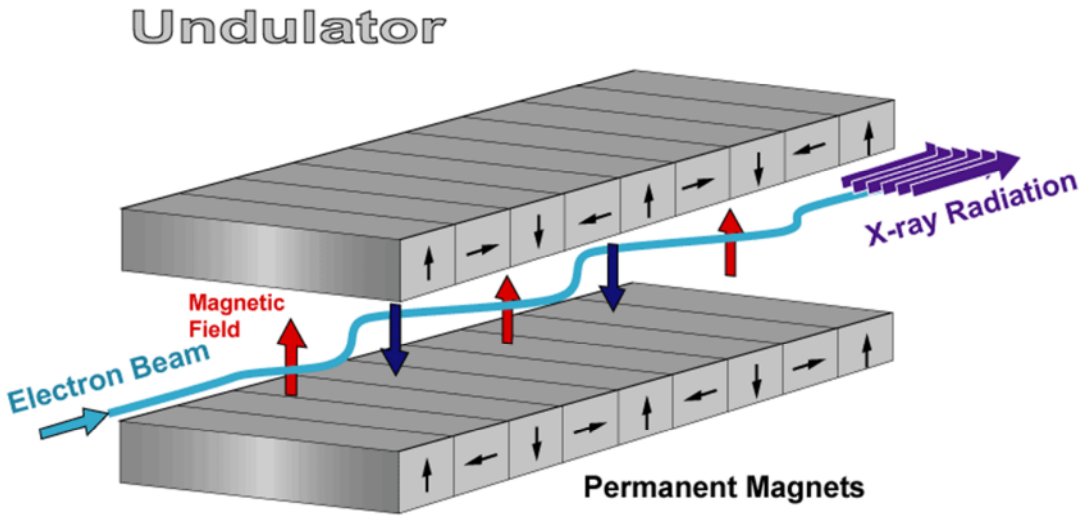


Figure 8: The Undulator[14]

The radiation emitted by one oscillation is in phase with the radiation emitted from an oscillation further on. The net effect is that the radiation emitted at each stage adds coherently. This

amplitude can then be squared to give the total intensity. X-Rays produced in this manner contain a range of energies and contain harmonics from the undulator. A monochromator is used to select a narrow bandwidth, suitable from X-Ray diffraction experiments. The X-Rays are then guided towards a beamline for use in experiments In order to observe coherent diffraction, the coherence length of the x-rays needs to be at least the size of the object. This is achieved by aperturing the X-rays with slits. The recorded pattern is then processed by phasing procedures.

3.2.1 Oversampling

As mentioned in section 2.2.2, samples must be taken at a frequency twice that of the highest frequency component. It was observed by Sayre that for intensity to be measured, it must be sampled at twice that of the sample rate for the amplitudes, a total of four times the highest frequency[10]. In practice this is accomplished by using a probe that is at least twice the size of the object under investigation and using a suitable detector. This is known as oversampling and is required for the phase retrieval algorithms (section 4) to work successfully.

3.3 Advanced Photon Source (APS)

The experiment was performed at the Advanced Photon Source (APS), Argonne National Laboratory in Illinois. The APS consists of 34 beamlines for conducting various scientific experiments; the data used in this experiment was collected from beamline 34-ID-C.

After X-Rays are generated from the undulator, they enter beamline 34 and enter the first optics enclosure. Here the beam is manipulated by various components to reduce the beam intensity, flux and to manipulate the coherence length of the beam. This is necessary to avoid damage to various components. After it is reduced the beam is split into two different branches, a “white” and a “pink” beam, using a beam splitting mirror. Where “white” refers to the fact that it is monochromatic and “pink” refers to the lack of monochromaticity.

The white beam is used for Microfocus Diffraction experiments and the pink beam is used for Coherent X-Ray Diffraction experiments. Despite the reduction in the flux, both experiments need beams that are even further reduced to avoid problems with cooling and saturation hence this will not cause any issues further down the line.

A schematic diagram of the layout is shown below.

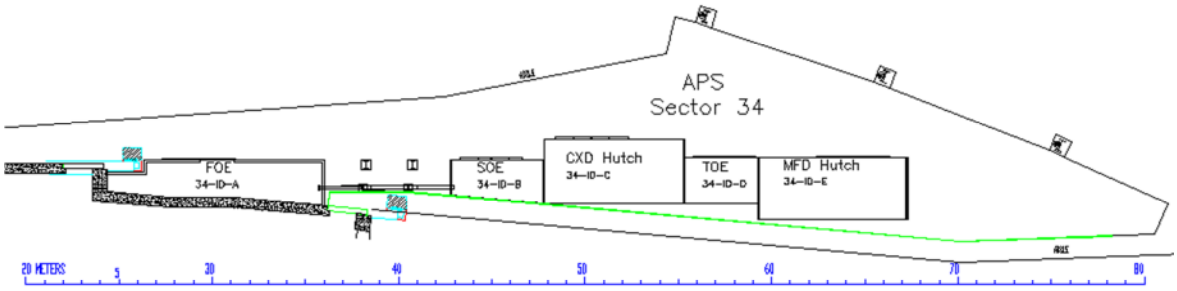


Figure 9: Floor Map of Beamline 34[15]. The CXD hutch is in the middle (34-ID-C)

Inside the CXD hutch (34-ID-C) is where the CXD experiments take place. All components are placed under Ultra High Vacuum (UHV) conditions. This is to ensure that a phase change in the X-Ray beam is not induced through interactions with particles in the air and to avoid scattering with air particles. The exception to this is the CCD sensor which is unable to detect changes in phase. A similar experimental set up exists in Diamond, Oxfordshire and the following images illustrate a typical experimental set up.

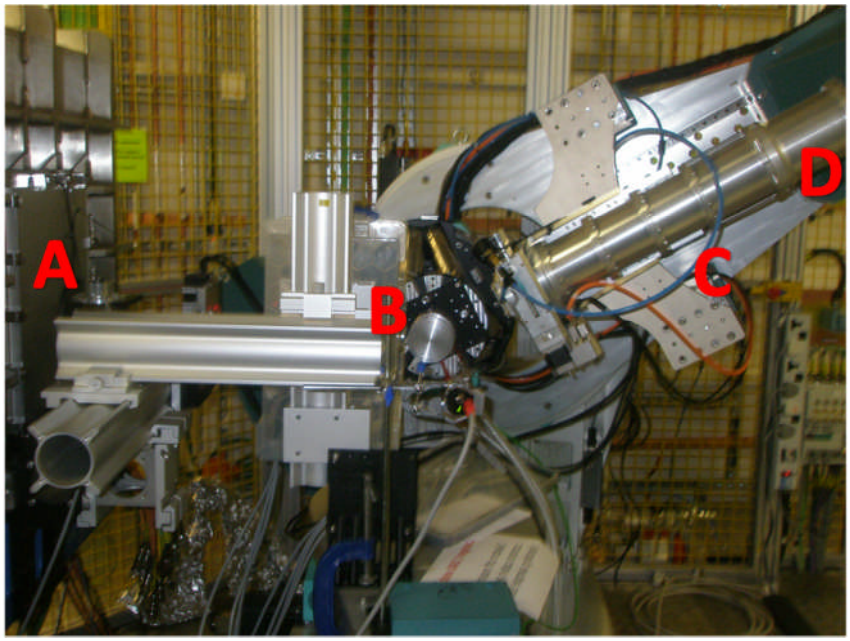


Figure 10: CXD experiment set up at Diamond

The X-Rays enter just beneath A and a set of slits aperture the beam. These slits can be adjusted to control the level of coherence. After the beam has been cut down, it is focused using a set of Kirkpatrick-Baez (KB) micro focusing mirrors[16]. These mirrors are composed primarily of silicon; they are housed in the plastic box behind point B.

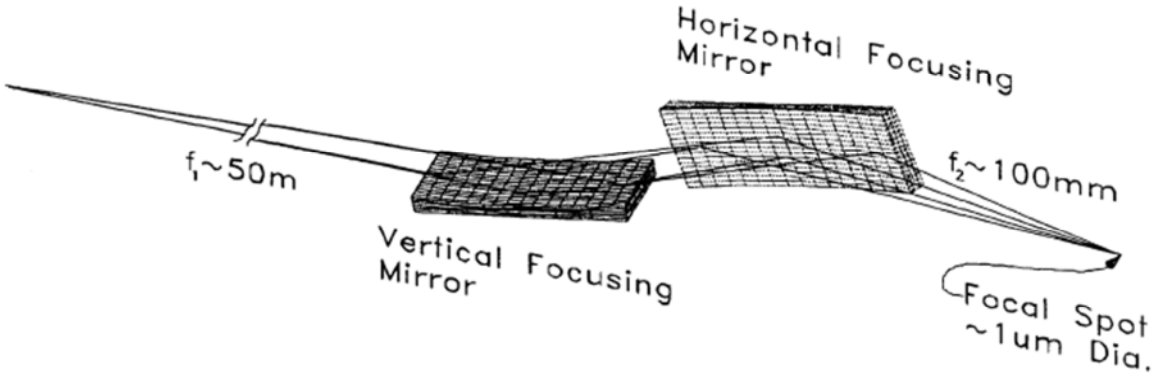


Figure 11: Arrangement of KB mirrors to focus X-Rays[16]

At B the sample under study is kept within an adjustable enclosure. The sample maybe be translated in the x, y or z directions and it can also be rotated in the theta, phi and psi directions.

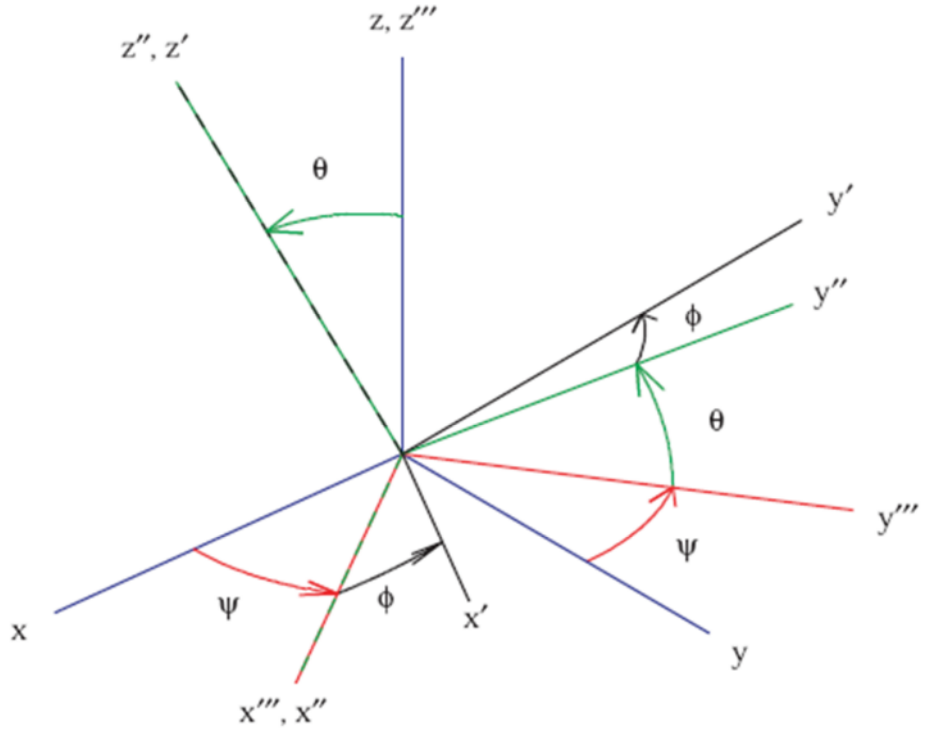


Figure 12: Illustrating the Tait-Bryan angles[17]

The detector may also be moved in a similar manner allowing the collection of diffraction patterns from a whole host of configurations. The diffracted X-Rays carry on through the pipe labelled C. This pipe is placed under vacuum conditions, rather than UHV to minimise scattering with air particles. Point D is where the scattered X-rays are recorded; the detector is just out of shot.

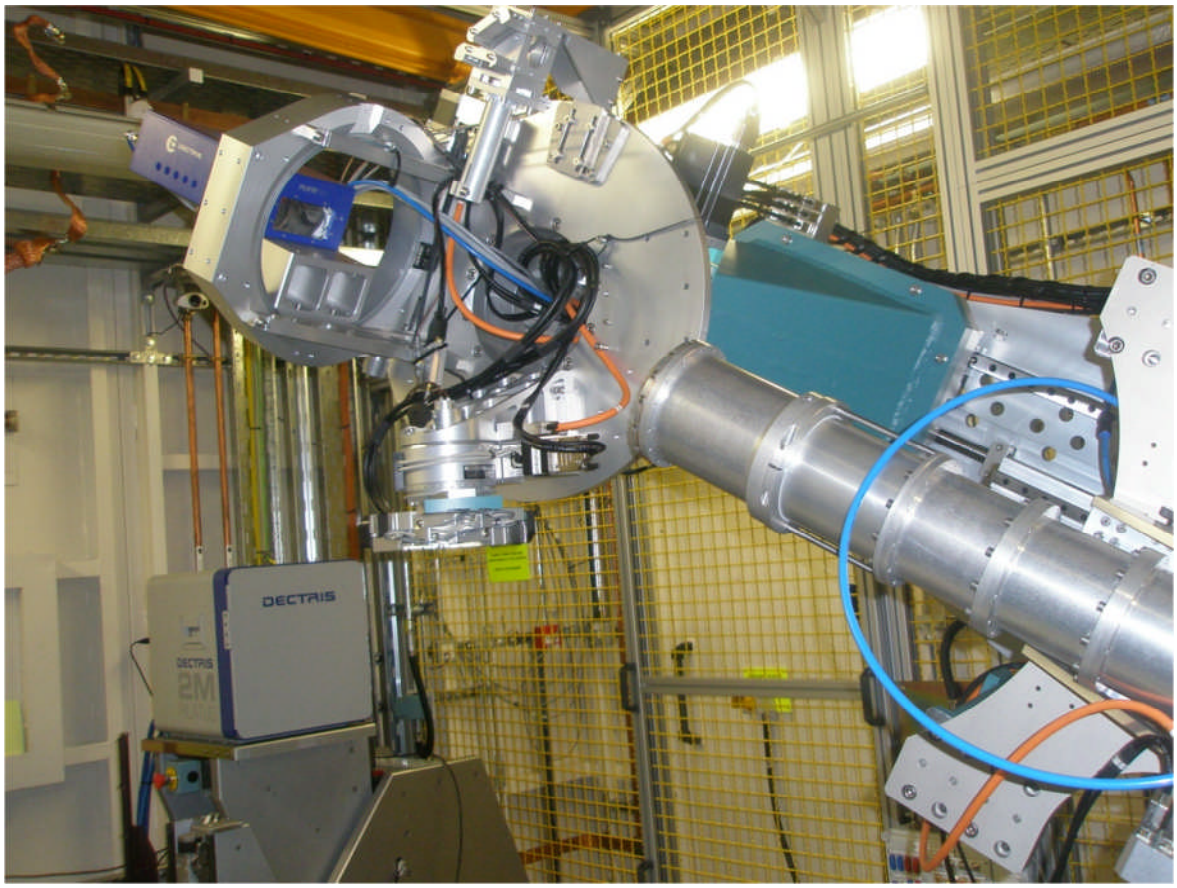


Figure 13: The X-Ray detector (blue box)

Once the diffraction pattern is recorded, it is processed to reconstruct the object. This is done using phase retrieval algorithms.

4. Phase Retrieval Algorithms

4.1 Phase Problem

Whenever a measurement is made, only the intensities can be recorded and the phase information is lost. The Phase problem is the name given to this loss of the phase when making a measurement. The Phases contain important information and have to be retrieved to reconstruct the object successfully. $I(\vec{k}) = \left|\tilde{A}(\vec{k})\right|^2$ and $\tilde{A}(\vec{k}) = \left|\tilde{A}(\vec{k})\right|e^{i\psi(k)}$ where $e^{i\psi(k)}$ represents the phases of the amplitudes.

To illustrate the importance of the phases, two images are presented here. Both images are Fourier transformed and their Fourier phases are interchanged. Upon returning to real space, it can be seen that the image that contributed to the phases has replaced the image responsible for the intensities.

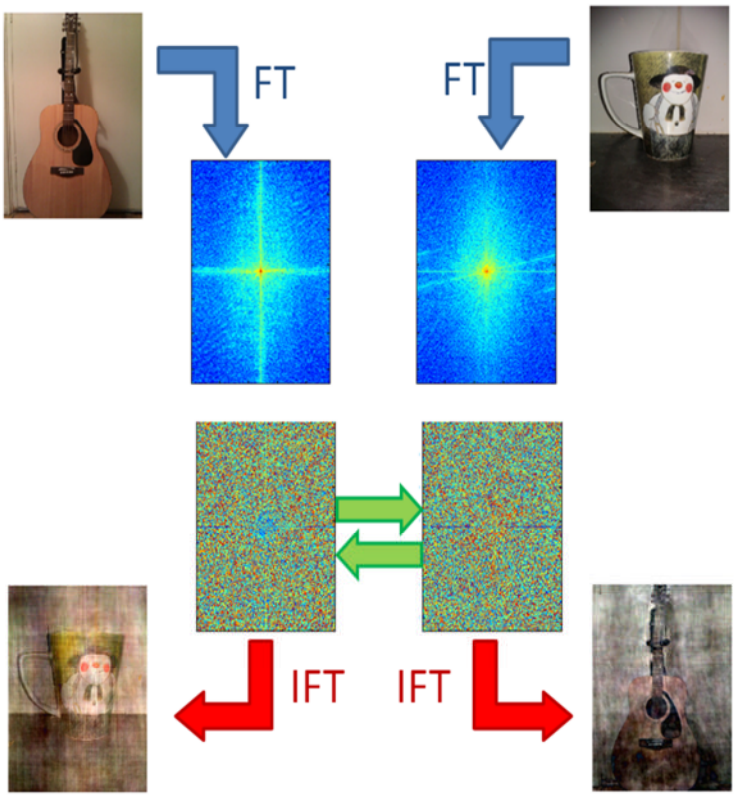


Figure 14: Illustration of the importance of Phases when reconstructing diffraction images. The upper images are the amplitudes (log scale), and the lower are the phases.

In order to recover phases several algorithms have been developed.

4.2 Error Reduction

The Error Reduction (ER) Algorithm is an algorithm developed by J.R. Fienup in 1978[18]. The algorithm works by applying certain constraints on the object and its diffraction pattern to reconstruct the object.

If the object is represented by $f(x)$ then it is connected to its diffraction pattern, $F(U)$ through its Fourier transform[18]:

$$F(u) = |F(u)|e^{i\psi(u)} = \mathcal{F}[f(x)] = \int\int_{-\infty}^{+\infty} f(x)e^{-i2\pi u.x}dx$$

Equation 8: Relationship between an object and its diffraction pattern

The constraints in real space are typically a support. A support is an area in which the object must exist and outside the support $f(x)$ is equal to zero. In Fourier or reciprocal space, the constraint is that the amplitudes of the guessed object must be changed to those obtained experimentally. This is outlined in the following flow chart.

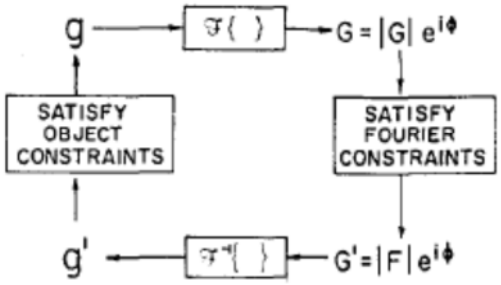


Figure 15: Error reduction flow chart[18]

Therefore there are four steps to the ER algorithm:

1. Calculate Fourier transform of initial guess. ($\mathcal{F}[g_k(x)] = |G_k(u)|e^{i\phi_k(u)} = G_k(u)$)
2. Replace guessed amplitudes with experimental ones. ($G'_k(u) = |F(u)|e^{i\phi_k(u)}$)
3. Return to real space through the inverse Fourier transform. ($\mathcal{F}^{-1}[G'_k(u)] = g'_k(u)$)
4. Apply support constraints to create a new guess and repeat as necessary. ($g_{k+1}(x) = \begin{cases} g'_k(x) & x \neq \gamma \\ 0, & x = \gamma \end{cases}$)

   Where γ represents all points outside the support.

As a demonstration of the preceding concepts, a simulation was created using Matlab®. Details of the code used to generate the following simulations are given in the appendix.

1. A simple square is generated, its dimensions are 56 by 56 pixels ($f(x)$):

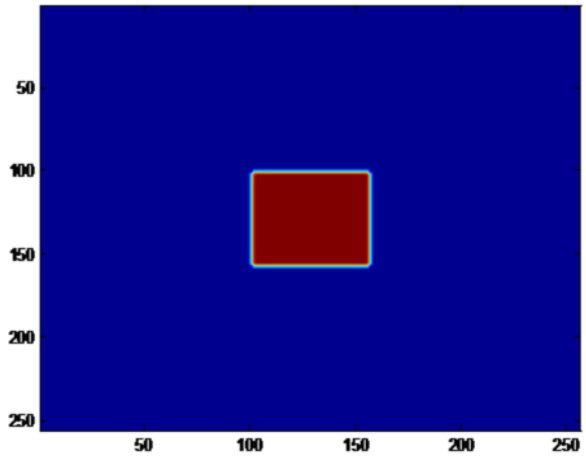


Figure 16: 56 by 56 square

2. A support of slightly larger dimensions is generated. (74 by 74)

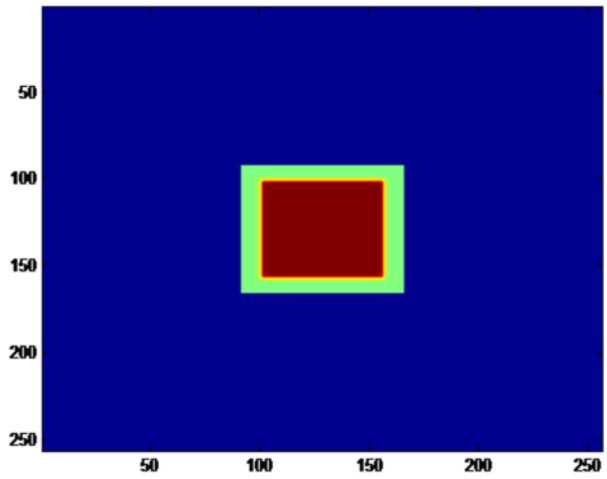


Figure 17: Support imposed upon object

3. The Fourier transform is computed. Although the phases are shown here, the phases as mentioned before, are lost and must be recovered to make an accurate reconstruction. It is at this stage that the calculated amplitudes are replaced with experimental ones.$|G_k(u)|e^{i\phi_k(u)} \rightarrow |F(u)|e^{i\phi_k(u)}$

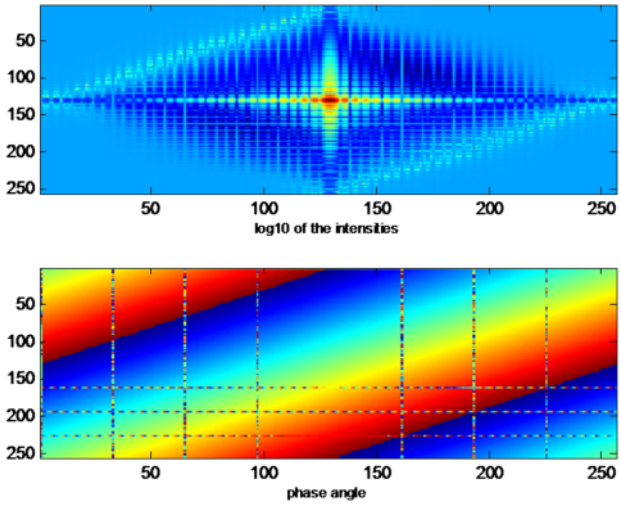


Figure 18: Fourier transform of square shown in figure 13

4. In order to simulate real life experimental conditions, phase information is removed (only retaining$|F(u)|$), it is transferred back to real space and the support constraint is applied. This object will serve as the initial guess ($g_{k+1}(x)$=$S[\ g'_k(x)]$). Where S is the support function.

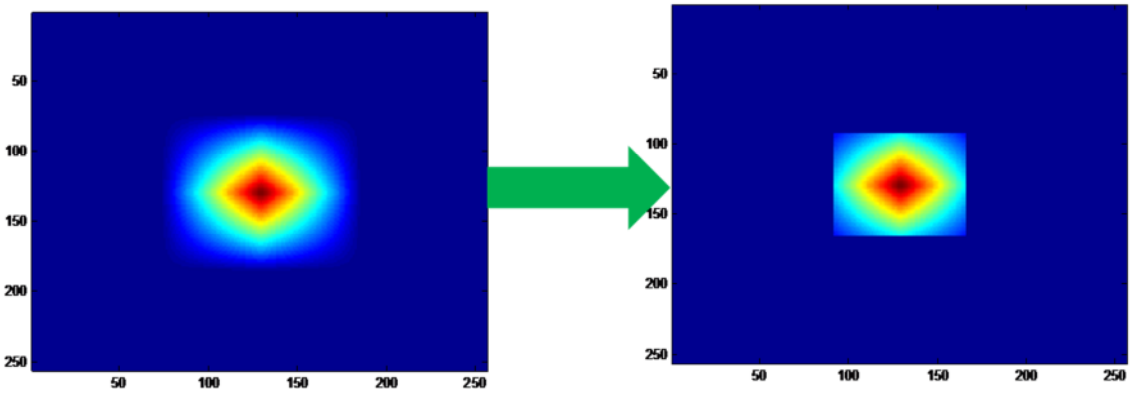


Figure 19: Setting all points outside the support to zero

5. The Fourier transform is calculated again and the resulting amplitudes are replaced with those of figure 15 whilst maintaining phases resulting from the application of a support. $(\mathcal{F}[g_{k+1}(x)] = |G_{k+1}(u)|e^{i\psi(u)} \rightarrow G'_{k+1}(u) = |F(u)|e^{i\psi(u)})$
6. The process is repeated until the desired result is obtained. In this case there were a total of 300 iterations. The resulting object is shown below.

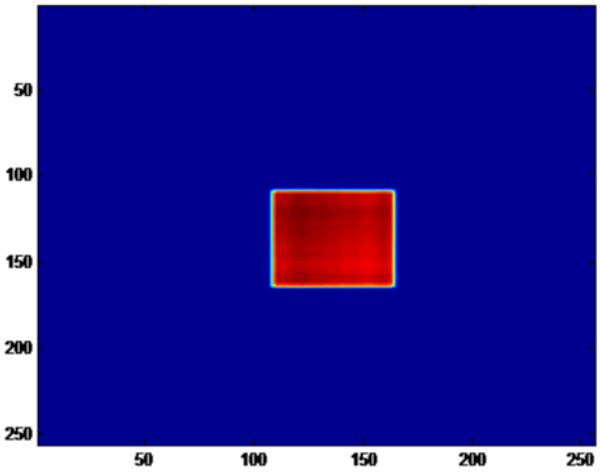


Figure 20: Recovered object

To monitor the progress of the algorithm the following equation shows the difference between experimental amplitudes and those calculated using recovered amplitudes, the sum squared error (SSE):

$$SSE = \frac{\sum[|G_k(u)| - |F(u)|]^2}{\sum|F(u)|^2}$$

Equation 9: SSE

The recovered object has an error of less than 10^{-6} which is low enough for an accurate reconstruction.

4.3 Hybrid Input/output

The Hybrid Input/output (HIO) algorithm is almost identical to the ER except it does not change values outside the support to zero. Instead values outside the support are used to generate new inputs. Step 4 becomes: $g_{k+1}(x) = \begin{cases} g_k(x), & x \neq \gamma \\ g_k(x) - \beta g'_k(x), & x = \gamma \end{cases}$

The β parameter controls the amount of feedback fed back into the next iterate and typical values are 0.9-1. Although the Error reduction algorithm has been shown to converge to the correct solution, there is no limit on the time taken to complete this convergence[19]. Typically both of these algorithms are used in conjunction to avoid problems with reaching local minima and convergence.

It is important to ensure a reasonable sized support is used and usually the support is generated from inverting its diffraction pattern which results in its autocorrelation function. To demonstrate this, Matlab® was used to create a large support and attempt to recreate the sample.

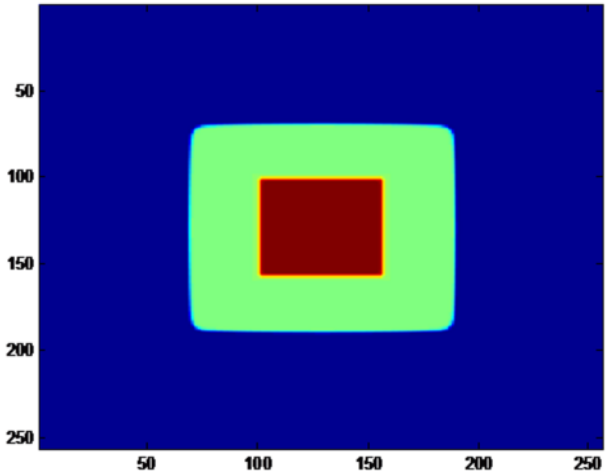


Figure 21: Support

Once more 300 iterations were performed:

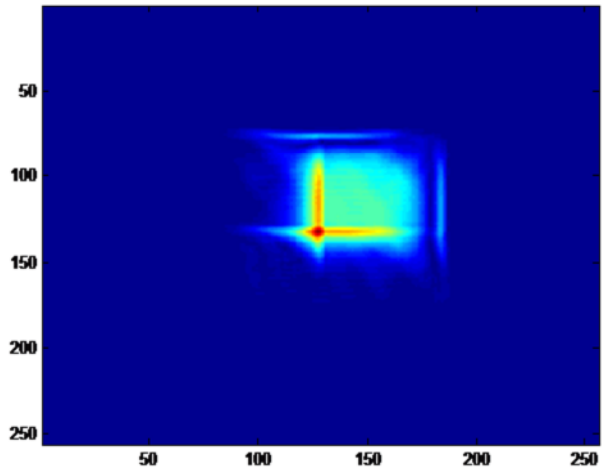


Figure 22

There was no visible improvement and only minuscule changes were observed between iterations. A solution to this problem is to implement a shrink wrap support. This uses a continuously updated support function based on a threshold. The support is typically updated every 10 iterations. The results of this are shown in the appendix.
Both of these algorithms suffer from the uniqueness issue[20]. This is where multiple images have the same Fourier amplitudes and it is very difficult to ascertain what the correct solution is. There are trivial solutions which differ from one another by reflection, conjugation, reflection, or an arbitrary phase offset. They are trivial as they all contain the sample and features of any interest. The issue arises when a solution also contains features from its conjugate for example, i.e. $f(x)$ and $f^*(-x)$ are both present in the image[21].

4.4 Ptychography

Ptychography is another algorithm that aims to solve the phase issue and retrieve phases unambiguously. It was invented by Professor Walter Hoppe in 1969[22]. The algorithm makes use of multiple overlapping regions to recover the phase and restore the object. The diffraction pattern consists of the Fourier transform of a convolution of the probe illumination and the object at a pre-determined position.
Ptychography (Greek for 'fold') works by making use of the interference in the overlap region to determine relative phase changes. This redundancy provides together with accurate knowledge of individual probe positions provides the information required to invert the image successfully.

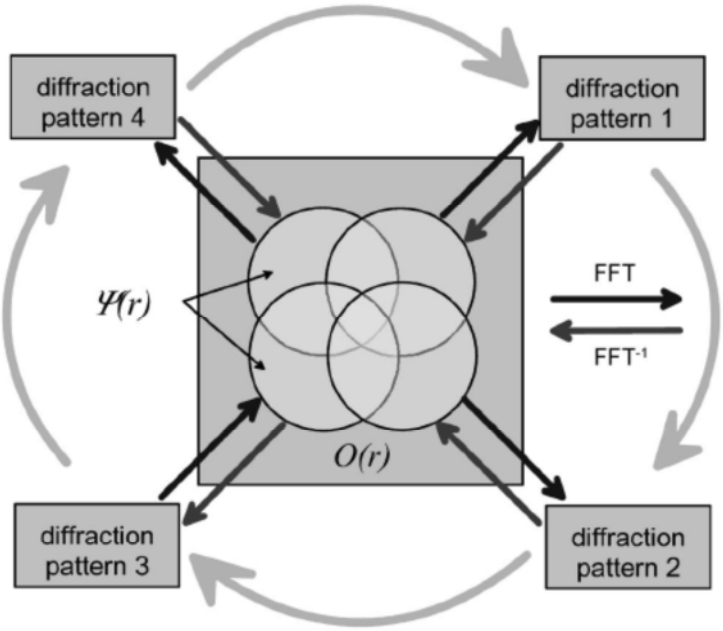


Figure 23: Outer arrows indicate position stepping between iterations. Inner arrows indicate FTs and IFTs and desired input-output information.[23]

If $O(\boldsymbol{r})$ represents the object function and $P(\boldsymbol{r})$ represents the probe illumination function, when the probe is incident upon the object the resulting exit wave, $\psi(\boldsymbol{r}, \boldsymbol{R}) = O(\boldsymbol{r})P(\boldsymbol{r} - \boldsymbol{R})$ is a convolution of the object and probe function at position $\boldsymbol{r} - \boldsymbol{R}$, where $\boldsymbol{R}$ is the relative shift between the probe and the object. This assumption is valid for thin objects[24].

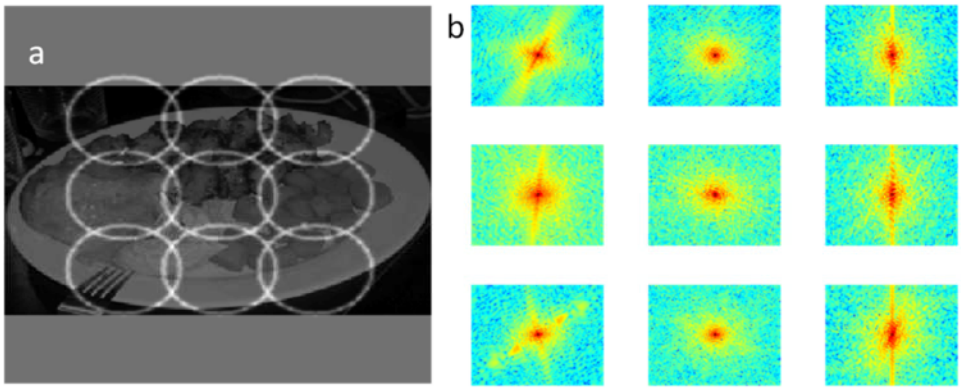


Figure 24: Testing the algorithm for nine overlapping positions. (a) is a sample image and the nine different probe positions can be seen. (b) are the respective Fourier transforms at this positions, the probe is a circular top hat (see appendix)

The algorithm is designed to find the modulus and phase of the Object function The starting point is an arbitrary guess of the object function $O_{g,n}(\boldsymbol{r})$, where g represents a guessed function at the nth iterate. From there the algorithm proceeds as follows:

1. Calculate guessed exit wave at position $\boldsymbol{R}$ by multiplying current object guess with probe at position R:

$$\psi_{g,n}(\boldsymbol{r},\boldsymbol{R}) = O_{g,n}(\boldsymbol{r})\, P(\boldsymbol{r}-\boldsymbol{R})$$

Equation 10

2. Apply the Fourier transformation operation to convert to reciprocal space:

$$\mathcal{F}\left[\psi_{g,n}(\boldsymbol{r},\boldsymbol{R})\right] = \Psi_{g,n}(\boldsymbol{k},\boldsymbol{R}) = \left|\Psi_{g,n}(\boldsymbol{k},\boldsymbol{R})\right| e^{i\theta_{g,n}(\boldsymbol{k},\boldsymbol{R})}$$

Equation 11

$\left|\Psi_{g,n}(\boldsymbol{k},\boldsymbol{R})\right|$and $i\theta_{g,n}(\boldsymbol{k},\boldsymbol{R})$ are the guessed amplitudes and phases respectively.

3. Replace guessed amplitudes with known experimental values:

$$\left|\Psi_{g,n}(\boldsymbol{k},\boldsymbol{R})\right| e^{i\theta_{g,n}(\boldsymbol{k},\boldsymbol{R})} \rightarrow |\Psi(\boldsymbol{k},\boldsymbol{R})| e^{i\theta_{g,n}(\boldsymbol{k},\boldsymbol{R})} = \Psi_{c,n}(\boldsymbol{k},\boldsymbol{R})$$

Equation 12

4. Perform the inverse Fourier transform to obtain an improved form of the exit wave function:

$$\mathcal{F}^{-1}\left[\Psi_{c,n}(\boldsymbol{k},\boldsymbol{R})\right] = \psi_{c,n}(\boldsymbol{r},\boldsymbol{R})$$

Equation 13

5. Update object at current probe position using the update function:

$$O_{g,n+1}(\boldsymbol{r}) = O_{g,n}(\boldsymbol{r}) + \frac{|P(\boldsymbol{r}-\boldsymbol{R})|}{|P_{max}|(\boldsymbol{r}-\boldsymbol{R})} \frac{P^*(\boldsymbol{r}-\boldsymbol{R})}{|P(\boldsymbol{r}-\boldsymbol{R})|^2+\alpha} \times \beta(\psi_{c,n}(\boldsymbol{r},\boldsymbol{R}) - \psi_{g,n}(\boldsymbol{r},\boldsymbol{R}))$$

Equation 14

6. Move to the next position in which the probe overlaps with that of the previous position and repeat the above steps until the error is sufficiently small.

The β parameter serves a similar role to that in the HIO algorithm, controlling the feedback. The term

$$\frac{|P(\boldsymbol{r}-\boldsymbol{R})|}{|P_{max}|(\boldsymbol{r}-\boldsymbol{R})}$$

Equation 15

maximises the effect where $P(\boldsymbol{r}-\boldsymbol{R})$ is large and minimises the effect of less illuminated regions. The term

$$\frac{P^*(\boldsymbol{r}-\boldsymbol{R})}{|P(\boldsymbol{r}-\boldsymbol{R})|^2+\alpha}$$

Equation 16

Divides the probe function obtained in equation 9. The progress of the algorithm can be monitored using a variation of the SSE (See Equation 9).

$$SSE = \frac{\sum\left[|\Psi(\boldsymbol{k},\boldsymbol{R})|^2 - \left|\Psi_{g,n}(\boldsymbol{k},\boldsymbol{R})\right|^2\right]^2}{\sum|\Psi(\boldsymbol{k},\boldsymbol{R})|^2}$$

Equation 17: Modified SSE, the summations are over all the pixels in the array.

Clearly the algorithm relies heavily on probes which overlap to provide redundant information. Given a distance a between the centres of two overlapping probes and a probe radius r, the

absolute overlap is $o_{abs} = 2r - a$. Normalising this by dividing through by the diameter of the probe gives the relative overlap as[25]:

$$o = 1 - \frac{a}{2r}$$

Equation 18: Overlap parameter

The importance of this overlap parameter will be demonstrated in the following section.

4.4.1 Overlap Parameter

The same test image used in Figure 24 was used to demonstrate the importance of overlap. A total of 50 iterations were used to reconstruct the image for a variety of overlaps ranging from 0% to 100% in steps of 10%. The illumination function was chosen as a circular top hat.

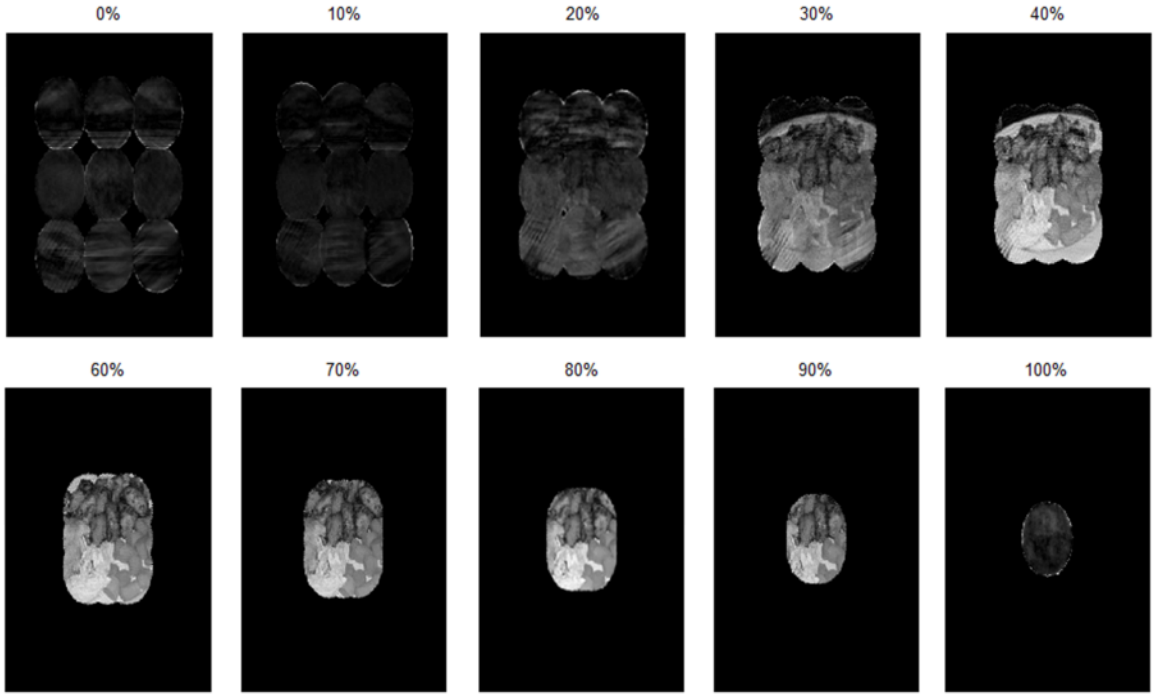


Figure 25: Reconstructed images for different overlaps. 50 iterations were used with nine probes in a 3x3 formation.

At the two extremes, 0% and 100% the Ptychography algorithm becomes very similar to the HIO algorithm; with 9 different regions solved simultaneously for 0% overlap and 9 identical degenerate positions containing no additional information for 100% overlap[25]. The differences between the original and the reconstructed images are very minimal for overlaps of 50% (not shown) to 90% when assed by the eye.

This is shown by the SSE error which is below 10^{-3} for 50%-90%. The respective errors show an initial rapid decay that quickly plateaus:

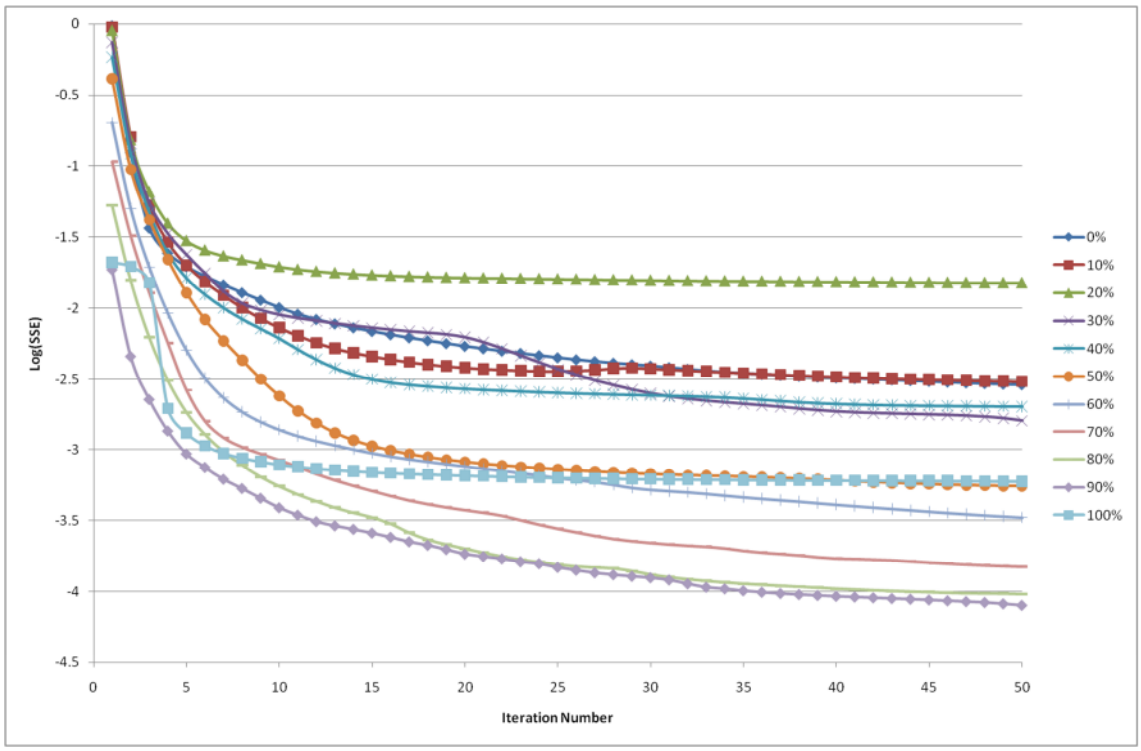


Figure 26: Log (SSE) error vs. number of iterations

Even though the $\log_{10}$ SSE for 100% is comparable to that of 50%, it is not discernible. The reason the error is comparable is that the summation for the error is over all nine probe positions, which reduces the SSE by a factor of ~9. If only one position was to be considered the error would be comparable to that of 0% overlap which would be the expected result as both scenarios are effectively equivalent to the HIO algorithm.

4.4.2 Probe retrieval

A modification to the algorithm allows simultaneous retrieval of the probe profile if this is also unknown. Usually the probe profile needs to be well known to allow accurate reconstructions, however this modification allows a guess to be made of both the probe function and object with the aim of retrieving both functions[26].

The modifications come in the form of a new update function:

$$\psi_j^{(n+1)}(\boldsymbol{r}) = \psi_j^{(n)}(\boldsymbol{r}) + p_F\left(2\hat{P}(\boldsymbol{r}-\boldsymbol{r}_j)\hat{O}(\boldsymbol{r}) - \psi_j^{(n)}(\boldsymbol{r})\right) - \hat{P}(\boldsymbol{r}-\boldsymbol{r}_j)\hat{O}(\boldsymbol{r})$$

Equation 19: Update function

Where p_F is the modulus constraint (Equation 12). $\hat{O}(\boldsymbol{r})$ and $\hat{P}(\boldsymbol{r})$ are defined as:

$$\hat{O}(\boldsymbol{r}) = \frac{\sum_j \hat{P}^*(\boldsymbol{r}-\boldsymbol{r}_j)\psi_j(\boldsymbol{r})}{\sum_j \left|\hat{P}(\boldsymbol{r}-\boldsymbol{r}_j)\right|^2}, \hat{P}(\boldsymbol{r}) = \frac{\sum_j \hat{O}^*(\boldsymbol{r}+\boldsymbol{r}_j)\psi_j(\boldsymbol{r}+\boldsymbol{r}_j)}{\sum_j \left|\hat{O}(\boldsymbol{r}+\boldsymbol{r}_j)\right|^2}$$

Equation 20: Object and Probe summations

5. Simulations

The nanowires as mentioned before contain stacking faults. There are four different possible configurations or domains, deformation and twinning faults within each of the Wurtzite and Zincblende sections. At the given angle the data was collected only the Zincblende domains contributed to the scattering. The effect these domains have in the scattered signal is a phase shift of $0, \frac{-2\pi}{3}$ or $\frac{+2\pi}{3}$.

To simulate these nanowires, a long rectangular object with missing domains and domains with different phases was created. The real nanowires have a diameter of roughly 50nm, to calculate how many pixels this corresponds to in the simulations the following equation must be used:

$$\Delta x = \frac{Z\lambda}{Nd}$$

Equation 21: Relationship between sample pixel size and detector pixel size

Where Δx is the sample pixel size, Z is the distance from the sample to the CCD camera (1.3m), λ is the wavelength of the incoming radiation (1.39Å), N is the number of pixels in the array (918) and d is the CCD pixel size (22µm). The calculation resulted in a pixel size of 8.95nm and so the wires are modelled to be 6 pixels in width.

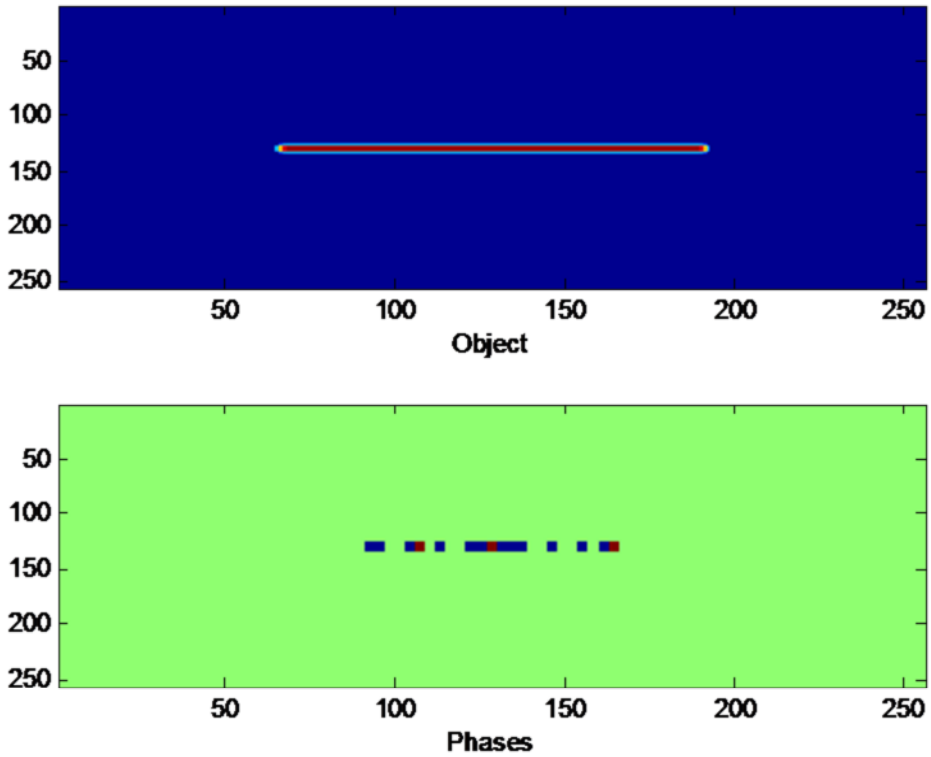


Figure 27: Simulated object, green is zero phase change, red is (2π/3) and blue is (-2π/3)

This wire has various domains ranging from 5 pixels to 10 pixels wide. First ER and HIO are evaluated to demonstrate problems these algorithms encounter when dealing with data that has such abrupt jumps in phase.

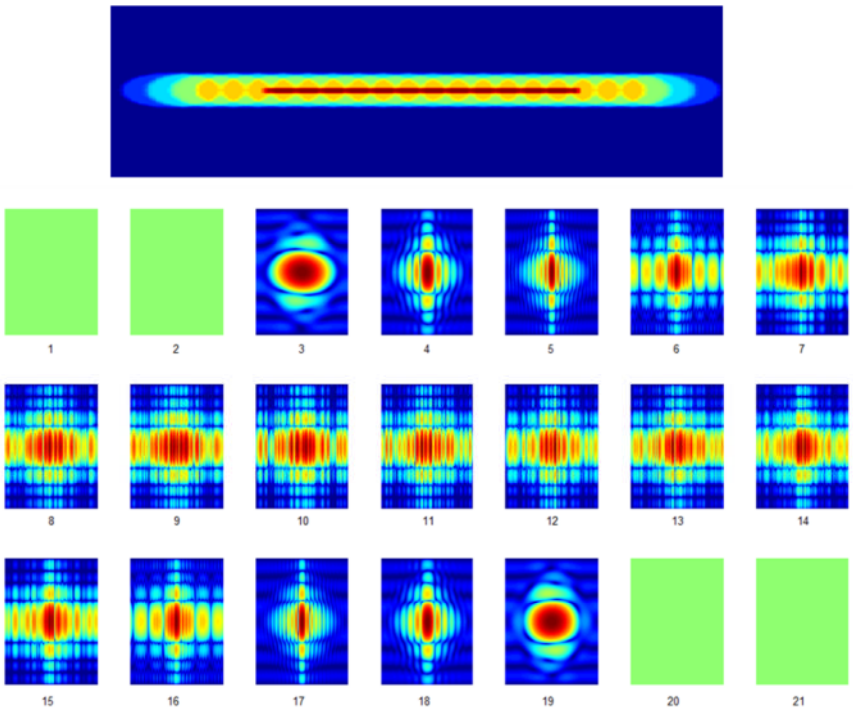


Figure 28: Diagram demonstrating probe positions over the wire. Counting from left to right the respective log10 intensities are also shown

5.1 Error Reduction/Hybrid Input/output

Position 11 was chosen as a demonstration for ER and HIO.

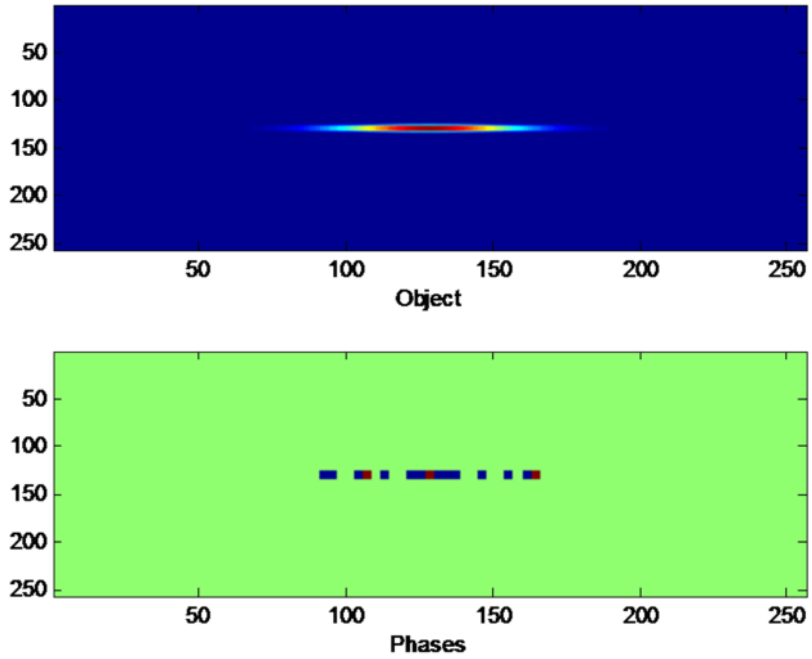


Figure 29: Portion of object "visible" from Probe position 11

Probe 11 acts as the support function. A total of 100 iterations were performed:

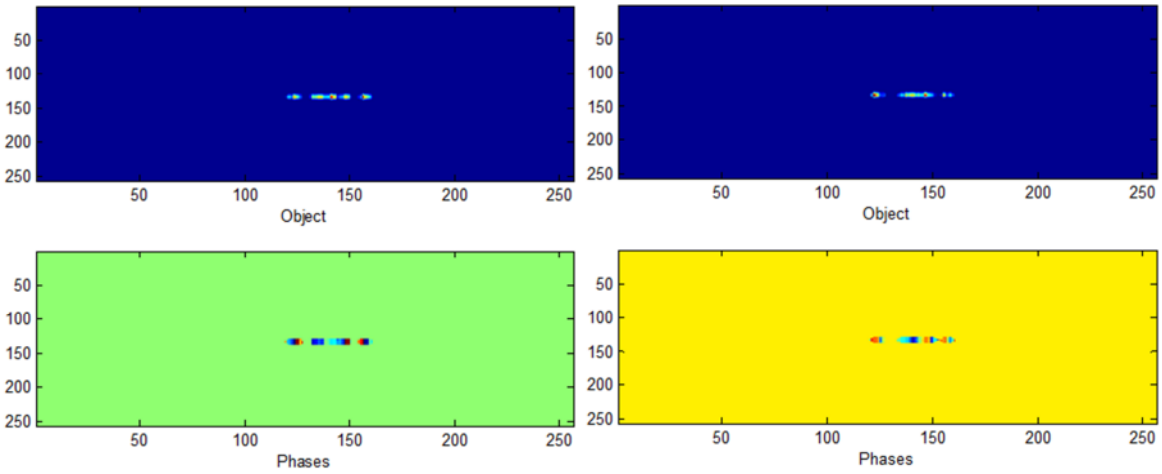


Figure 30: Retrieved Objects after 100 iterations from two different starting guesses

The errors obtained using Equation 9 were 0.0032213 and 0.0042343 respectively. This example demonstrates a problem with ER and HIO in that the solution is not necessarily unique[27]. Ignoring this problem, the size of the domains can still be estimated from this diagram. The different domains are (4±1) pixels, (3±1) pixels, (5±1) pixels, 6 pixels, (9±2) pixels, 2 pixels, 4 pixels and the final two are 2 pixels each. These values correspond to (35.8±8.95)nm, (26.85±8.95)nm, (44.75±8.95)nm, 53.7nm, (80.55±17.9nm), 17.9nm, 35.8nm and 17.9nm respectively.

5.2 Ptychography

The real data was obtained using 17 probe positions and 17 were used during these simulations.

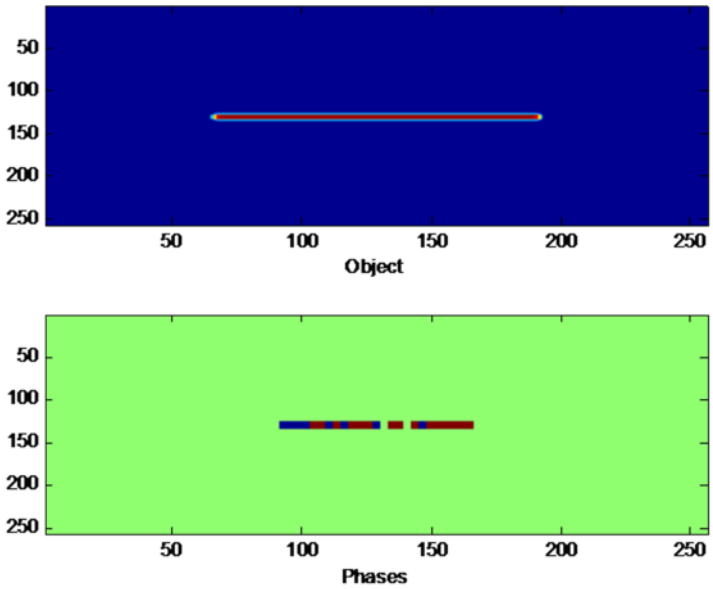


Figure 31: This simulated nanowire has the same dimensions as before, but slightly different phases as they are generated randomly

And the following diagram demonstrates the overlap:

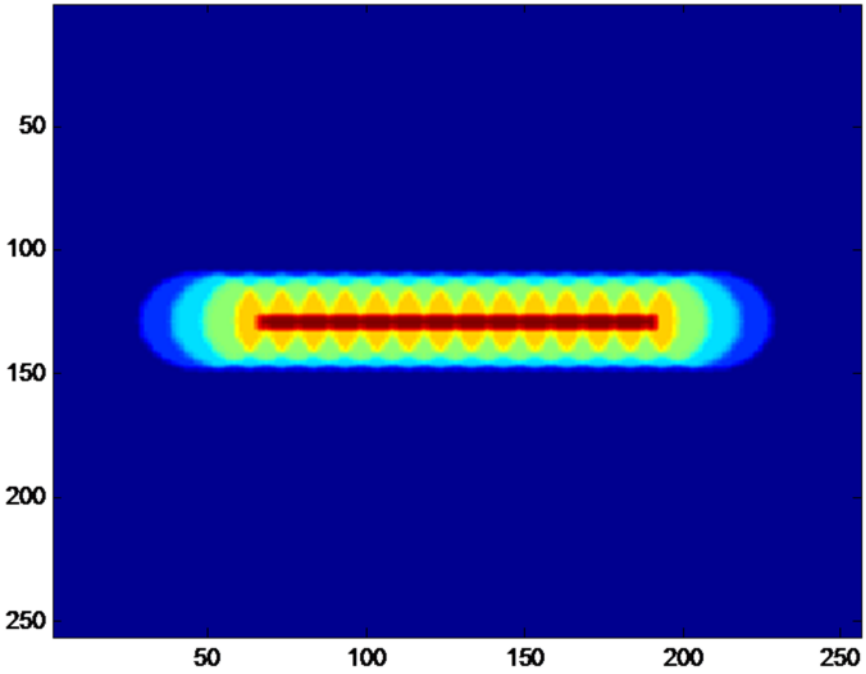
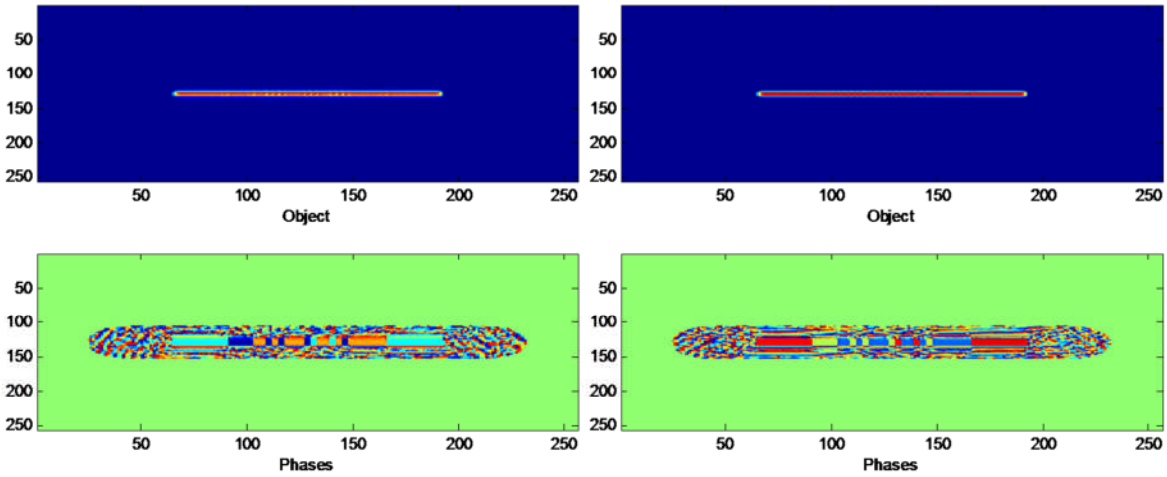


Figure 32: Probes overlapped upon the object

A total of 100 iterations were performed once more. The results after two different starting guesses is shown here:



Although the colour of the blocks has changed, the relative phase between the blocks remains constant. The problem is caused by the fact that two images with only a shift of phase between them have the same diffraction patterns and even with multiple overlapping regions it is not possible to distinguish the correct solution[21].

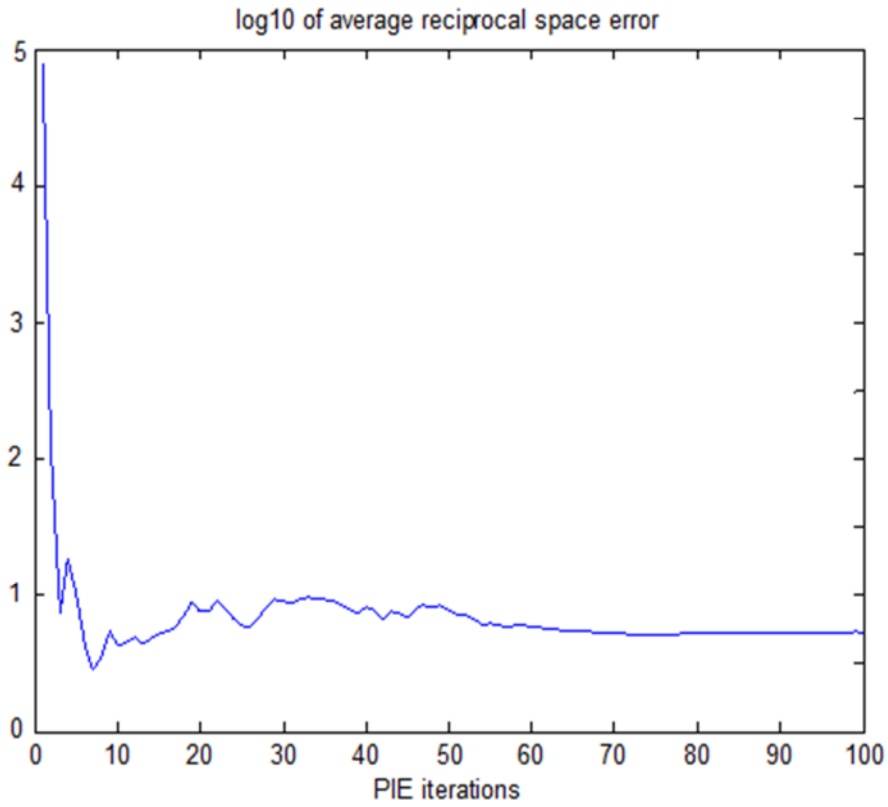


Figure 33: Error as a function of iteration number

6. Phasing of real data

Due to the nature of the diffraction patterns in the obtained data, it is not possible to use ER/HIO to decipher phasing information.

6.1 Ptychography

To save on computing time and to reduce the overall error nine central probe positions were used. The spacing between the probe positions was 0.5µm, which using Equation 21, corresponds to a spacing of 56 pixels. The initial probe function was modelled as a Gaussian with a FWHM of 1µm. A total of 200 iterations were performed and the probe was updated after 60 iterations. This was to ensure stability and avoid long convergence times[28].

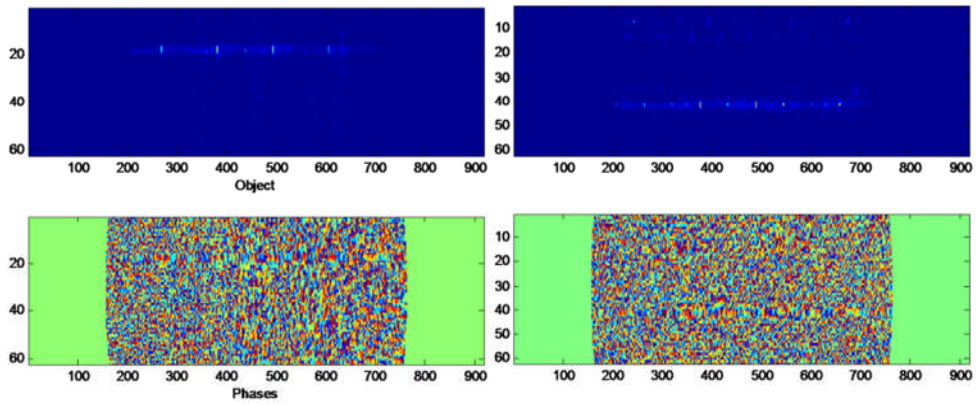


Figure 34: Recovered object

After 200 iterations a narrow wire like region is seen. The uniqueness issue is still present. The wires are smaller than the beam and a mirror image appears during the second reconstructions. The wire is six pixels in width corresponding to ~53.7nm; zooming into the phase data reveal discrete blocks.

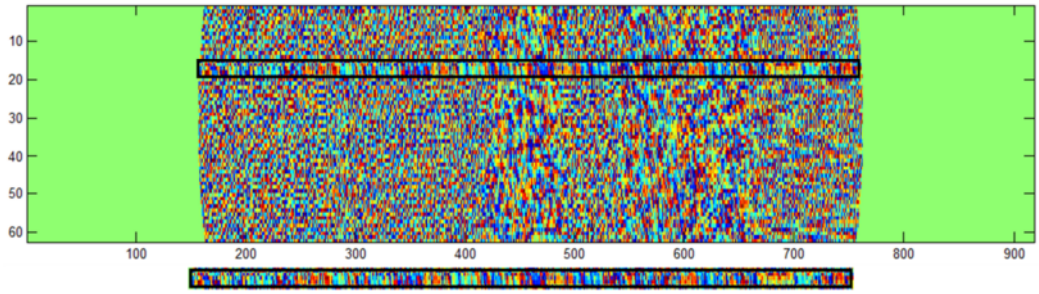


Figure 35

The width of the domains varies from ~5 pixels to ~60 pixels, corresponding to 44.75nm to 537nm. A comparison between the experimental intensities and ones that would be obtained from this object are shown in the appendix.

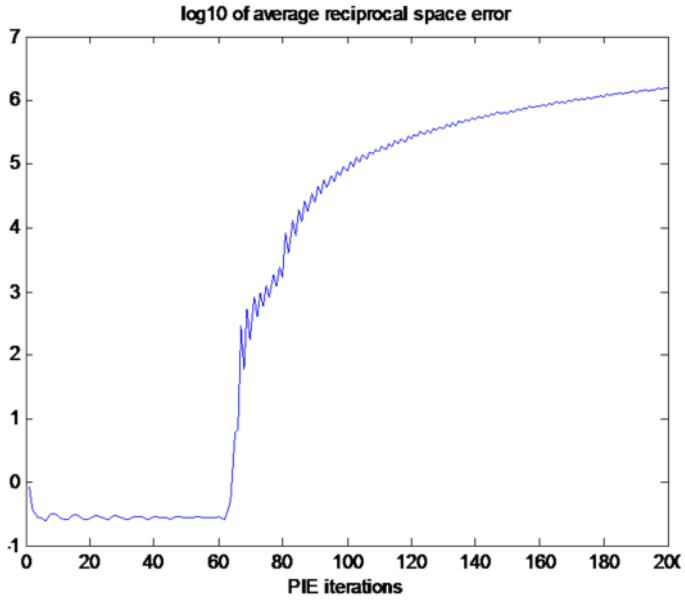


Figure 36: Error vs. Iteration number

The recovered phase profiles were not as clear as the simulations due to a variety of reasons.

- The data is not detailed enough; the data needs to be captured at a higher resolution as some of the more delicate features were unable to reconstruct.
- An inherent flaw in Ptychography when periodic objects are studied is that when a constant step size like that used in this data (0.5μm) is used, it is impossible to factorise the object function from the probe function. This leads to ambiguous results. The cure is to use irregular spacings to discriminate periodic features and features belonging to the probe.

7. Conclusions

It has been shown that Ptychographic methods can be used to invert various nanoscale objects to their original forms. ER and HIO, although they work well for non-complex objects, i.e. ones that only have transmissions and absorption properties with no phase shifting properties, they cannot be used for reconstructing faulted nanowires as the abrupt phase changes cause them to fail.
The Ptychography algorithm used has shown its strength is retrieving test images, test objects with abrupt phase jumps, and has shown potential in retrieving information about the nanowires.
The importance of overlap was demonstrated and it was shown that the optimal overlap was 80%-90%
The data acquired at APS was lacking in resolution and suffered from issues in periodicity, where all diffraction patterns share some common feature(s) making it impossible to factorise. Although Ptychography has shown promise and retrieved some structural data. In the future data could be collected at non-regular spacings to ensure there is no periodicity in the collected data.
The data was analysed using an Intel® Core™ 2 duo CPU @1.8GHz and 1GB of RAM, this was a limiting factor in the analysis of data, requiring long amounts of time to perform the calculations. The algorithm could be improved in the future to make the phase retrieval more efficient.
Possible future work in this area would be to look at different nanowires such as Indium Phosphide (InP), which has an even higher electron velocity than Gallium Arsenide[29]. An algorithm developed by Sicairos known as “Transverse translation diversity”[21] could also be tested in the future.

8. Appendix

8.1 Shape ER/HIO Code

```
%Square
square=ones([56,56]);
square=padarray(square,[100,100]);        % 'sample'


sample=convn(Hex,gauss_2D(7,7,1,1,0),'same');  %smooth the 'sample'
%support=padarray(ones([80,80]),[88,88]) ;       % create a support
int=abs(fftshift(fftn(fftshift(sample)))).^2;   %create the 'measured
intensity'
support=ifftshift(ifftn(fftshift(int)));
support=(support)./(support+0.01);
figure(1)
imagesc(sample)
pn=ifftshift(ifftn(ifftshift(int^0.5)));
figure(33);imagesc(abs(pn))
beta=0.9;         %guess at the sample
gk=support.*pn;
Gk=fftshift(fftn(fftshift(gk)));
Gdashk=(int.^0.5).*(Gk./abs(Gk));
gdashk=ifftshift(ifftn(ifftshift(Gdashk)));
gk=gdashk.*support;
figure(2)
imagesc(abs(gk))

for pp=1:1:1
for qq=1:1:30

%ER
for j=1:10

Gk=fftshift(fftn(fftshift(gk)));
Gdashk=(int.^0.5).*(Gk./abs(Gk));
gdashk=ifftshift(ifftn(ifftshift(Gdashk)));
%gk=gdashk.*support;
end


%%HIO
for j=1:50
Gk=fftshift(fftn(fftshift(gk)));
Gdashk=(int.^0.5).*(Gk./abs(Gk));
gdashk=ifftshift(ifftn(ifftshift(Gdashk)));
gk=gdashk.*support+((1-support).*(gk-beta.*gdashk));
end


for j=1:10
Gk=fftshift(fftn(fftshift(gk)));
Gdashk=(int.^0.5).*(Gk./abs(Gk));
gdashk=ifftshift(ifftn(ifftshift(Gdashk)));
gk=gdashk.*support;
end

%gk=(abs(gk)/max(max(abs(gk))));%.*exp(i*angle(gk));
```

```
nume=sum(sum(sum((abs(Gk)-(abs(int.^0.5))).^2)));
%calculating the error
denom=sum(sum(sum((abs(int.^0.5)).^2)));
error=nume./denom;
%kk=((qq));
ErrNume=nume;
ErrDenom=denom;
Error(pp,qq)=error;
figure(qq+2)
subplot(2,1,1);imagesc(abs(gk))
subplot(2,1,2);imagesc(abs(support))

support=shrink_wrap(support.*gk,.1,1,'gauss');
end
end
```

8.2 Ptychography example

```
probe_positions=[-25 -25 -25 0 0 0 25 25 25];
probe_positions(2,:)=[-25 0 25 -25 0 25 -25 0 25];

A=imread('INSERT IMAGE URL'); insert the url of the test image
qq=rgb2gray(A);
qq=double(qq);

n = 2^8; %define the array size
I = 1:n;
x = I-n/2;
y = n/2-I;
sample=qq;
figure(1)
colormap('gray')
imagesc(sample)

figure(1)
subplot(2,1,1);imagesc(abs(sample));xlabel('Object')
subplot(2,1,2);imagesc(angle(sample));xlabel('Phases')

[int,overlap,Object,Probes] = generate_diff(sample,probe_positions);
%generate diffraction patterns

qq=sum(Probes,3);
figure(45);colormap('gray');imagesc(50*qq+abs(sample)) %demonstrate probe
overlap


[ Probe,ObjectGuess1,Moves] = oldpy(int,probe_positions); %ptychography


figure(2) %final result
subplot(2,1,1);imagesc(abs(ObjectGuess1));xlabel('Object')
subplot(2,1,2);imagesc(angle(ObjectGuess1));xlabel('Phases')
```

generate_diff.m

```
function [int,overlap,Object,Probes] =
generate_diff(filename,probe_positions)

sample=filename;
%creat an array
nxyz=size(sample);
n=nxyz;
nx=nxyz(2);
ny=nxyz(1);
Ix = 1:nx;
x = Ix-nx/2;
Iy=1:ny;
y = ny/2-Iy;

%generate probes
probes=length(probe_positions);
%specify probe parameters
xfwhm=50;
yfwhm=50;
cparamx=xfwhm./(2.*((2.*log(2)).^0.5));
cparamy=yfwhm./(2.*((2.*log(2)).^0.5));
R=(xfwhm+yfwhm)/5;

Circ0= zeros(n);
[X0,Y0] = meshgrid(x,y);

if exist('probe_positions')==1;

    for xx=1:probes;
        Z=((X0-probe_positions(1,xx)).^2)+((Y0-probe_positions(2,xx)).^2)
<= R^2;
        Circ0(Z) = 1;
        Probe0=convn(Circ0,gauss_2D(7,7,1,1,0),'same');
        Probes(:,:,xx)=Probe0;
        Probe0= zeros(n);
        Circ0 = zeros(n);
    end
elseif exist('probe_positions')==0
    for qq=1:probes;
        pos = input('Please enter relative position from origin of probe
at position(qq):');
        Z = ((X0-pos).^2 + (Y0).^2 <= R^2);
        Circ0(Z) = 1;
        Probe0=convn(Circ0,gauss_2D(7,7,1,1,0),'same');
        Probes(:,:,qq)=Probe0;
        Circ0= zeros(n);
    end
end

for ww=1:probes
    Object(:,:,ww)=sample.*Probes(:,:,ww);
    int(:,:,ww)=(abs(fftshift(fftn(fftshift(Object(:,:,ww)))))).^2;
end
aaa=probe_positions;
yyy=aaa(1,2)-aaa(2,2);
xxx=aaa(1,1)-aaa(1,2);
%overlap=1-((yyy.^2+xxx.^2)^0.5)/(xfwhm+yfwhm);
a=abs(aaa(2,1)-aaa(2,2));
r=R;
overlap=1-(a./(2.*r));
```

```
end

%% this is just to create a 2d gauss for smoothing the square sample
function fxy = gauss_2D(nx,ny,sigx,sigy,angle )
%2d gauss, checks for even or odd length so as to center it correctly and
%avoid division by a 0 sigma.  0 sigma can be entered and it will return
%the correct gauss form, i.e repeated 1d gaussians

if isempty(sigy), sigy = sigx;end
if (nargin<5) || isempty(angle), angle = 0;end

fxy=zeros(nx,ny);


[x , y]=meshgrid( -(nx-1)/2:(nx-1)/2,-(ny-1)/2:(ny-1)/2);

if angle ~= 0
    theta=angle/180e0*pi;
    xd=x*cos(theta)-y*sin(theta);
    yd=x*sin(theta)+y*cos(theta);
    x=xd;
    y=yd;
    xd=0;
    yd=0;
end

if mod(nx,2) == 1
    if sigx == 0, gx=(x == 0);else gx=exp(-0.5.*x.^2./sigx^2);end
else gx=exp(-0.5.*x.^2./sigx^2);end

if mod(ny,2) == 1
    if sigy == 0, gy=(y == 0);else gy=exp(-0.5.*y.^2./sigy^2);end
else gy=exp(-0.5.*y.^2./sigy^2);end

fxy=gx.*gy;

fxy=fxy/sum(sum(fxy));

end
```

Ptych.m

```
function [ Probe,ObjectGuess1,M,Error] =
Ptych(int,probe_positions,linear,obj0)

positions=size(int,3);
%This define the number of different potiions used to generate each frame
%of the diffraction pattern

n=size(int);
nxyz=size(int);
nx=nxyz(2);
ny=nxyz(1);
Ix = 1:nx;
x = Ix-nx/2;
Iy=1:ny;
y = ny/2-Iy;
%generates an array of n-by-n, with co-ordinates of (-x,-y) to (x,y)
```

```
Circ0= zeros([n(1) n(2)]);
[X0,Y0] = meshgrid(x,y);

%probe parameters here must match up to those in generatediff
xfwhm=75;
yfwhm=75;
cparamx=xfwhm./(2.*((2.*log(2)).^0.5));
cparamy=yfwhm./(2.*((2.*log(2)).^0.5));
R=(xfwhm+yfwhm)/5;

    if exist('probe_positions')==1;

        for qq=1:positions;
            Z=(((X0-probe_positions(1,qq)).^2)+(Y0-
probe_positions(2,qq)).^2 <= R^2);
            Circ0(Z) = 1;
            Probe0=convn(Circ0,gauss_2D(7,7,1,1,0),'same');
            Probes(:,:,qq)=Probe0;
            Circ0= zeros([n(1) n(2)]);
        end

    elseif exist('probe_positions')==0
        for qq=1:positions;
            posx = input('Please enter relative position from origin of
probe at position(qq):');

            Z = ((X0-posx).^2 + (Y0).^2 <= R^2);
            Circ0(Z) = 1;
            Probe0=convn(Circ0,gauss_2D(7,7,1,1,0),'same');
            Probes(:,:,qq)=Probe0;
            Circ0= zeros(n);
        end
    end

%creates a matrix of probe positions for each frame


%Pytch

alpha=0.001;
beta=1;
%random guess at object
%its = input('Please enter the required number of iterations:');
its=50;

try
    ObjectGuess1=obj0;
catch
    ObjectGuess1 = ones([ny,nx]);
end

try
    linear=linear;
catch
    linear=0;
end
```

```
psi_n1=zeros([size(Probes)]);
for tt = 1:positions, psi_n1(:,:,tt)=Probes(:,:,tt).*ObjectGuess1;end
for ff=1:positions, objectdisplacement(:,:,ff)=circshift(ObjectGuess1,[0
-probe_positions(1,ff)]);end
conj_Probes=conj(Probes);
conj_object=conj(objectdisplacement);
for ww=1:its;

    fprintf(['PIE iteration ' num2str(ww,'%03d') ', error: '])

    %Using the conventions in Rodenburg's paper "A phase retrieval
algorithm
    %for shifting illumination"
    for ee=1:positions;
        psig(:,:,ee)=ObjectGuess1.*Probes(:,:,ee);
        PSIg(:,:,ee)=fftshift(fftn(fftshift(psig(:,:,ee))));
        PSIc(:,:,ee)=((int(:,:,ee)).^0.5).*(exp(i*angle(PSIg(:,:,ee))));
        psic(:,:,ee)=fftshift(ifftn(fftshift(PSIc(:,:,ee))));
        update_function=
conj(Probes(:,:,ee)).*abs(Probes(:,:,ee))./(conj(Probes(:,:,ee)).*Probes(
:,:,ee)+alpha);
        object_new = ObjectGuess1+update_function.*beta.*(psic(:,:,ee)-
psig(:,:,ee));

        ObjectGuess1 = object_new;
         figure(23)
         colormap('gray')
         subplot(2,1,1);imagesc(abs(ObjectGuess1))
         subplot(2,1,2);imagesc(angle(ObjectGuess1))
         M(:,:,ww)=ObjectGuess1;



        Num=sum(sum((abs(PSIg(:,:,ee))-int(:,:,ee).^0.5 ).^2));
        Den=sum(sum(int(:,:,ee)));



        if ww>300
            shiftpsic(:,:,ee)=circshift(psic(:,:,ee),[0 -
probe_positions(1,ee)]);
            compobjpsic(:,:,ee)=conj_object(:,:,ee).*shiftpsic(:,:,ee);

objectconjobj(:,:,ee)=objectdisplacement(:,:,ee).*conj_object(:,:,ee);
            Probest=(sum(compobjpsic,3))./(sum(objectconjobj,3));
            for tt=1:positions
                Probes(:,:,tt)=circshift(Probest,[0
probe_positions(1,tt)]);

            end
        end
    end

    if linear==1;
            for ee=positions:-1:1;
            psig(:,:,ee)=ObjectGuess1.*Probes(:,:,ee);
            PSIg(:,:,ee)=fftshift(fftn(fftshift(psig(:,:,ee))));

PSIc(:,:,ee)=((int(:,:,ee)).^0.5).*(exp(i*angle(PSIg(:,:,ee))));
            psic(:,:,ee)=fftshift(ifftn(fftshift(PSIc(:,:,ee))));
```

```
            update_function=
conj(Probes(:,:,ee)).*abs(Probes(:,:,ee))./...
            (conj(Probes(:,:,ee)).*Probes(:,:,ee)+alpha);
            object_new =
ObjectGuess1+update_function.*beta.*(psic(:,:,ee)-psig(:,:,ee));
            ObjectGuess1 = object_new;
            figure(23)
            subplot(2,1,1);imagesc(abs(ObjectGuess1))
            subplot(2,1,2);imagesc(angle(ObjectGuess1))
            Num=sum(sum((abs(PSIg(:,:,ee))-int(:,:,ee).^0.5 ).^2));
            Den=sum(sum(int(:,:,ee)));
            end
    else



    end


Error(ww)=(Num./Den);
disp(Error(ww))

ind=0;
end

 %% plot error
 figure(99);
 plot(log10(Error+0.0000000000001));
 title('log10 of average reciprocal space error')
 xlabel('PIE iterations')


Probe=Probes;
end
```

8.3 Restored shrink-wrap example

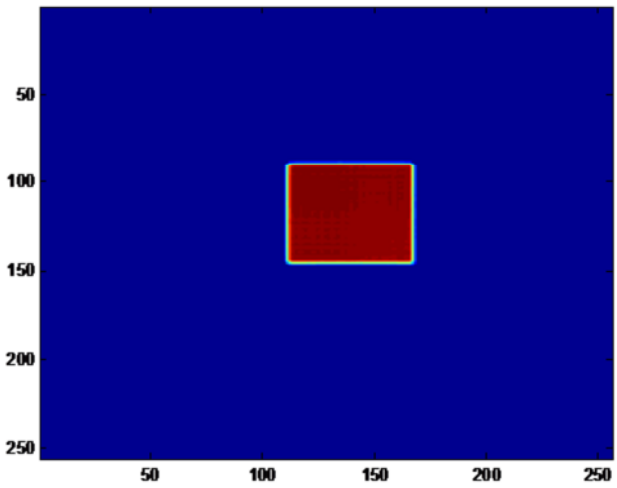


Figure 37: Restored Square

8.4 Comparison of experimental intensities and restored intensities

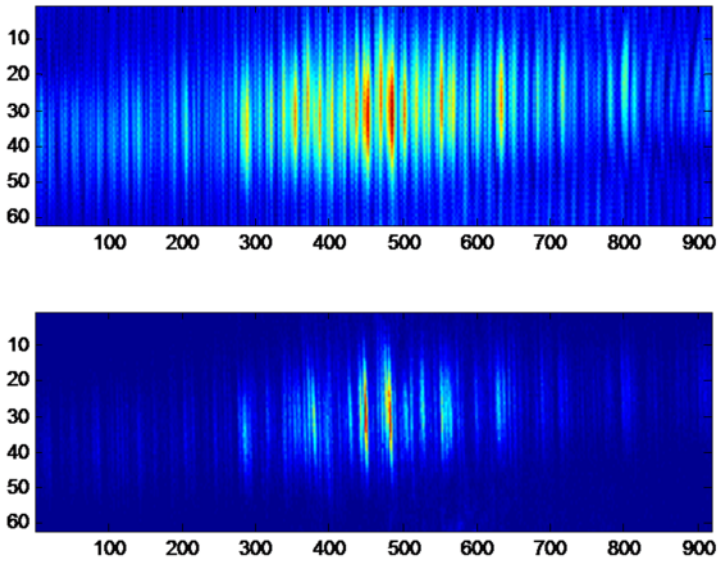


Figure 38: Comparison of experimental intensities and restored intensities

8.5 Top Hat Function

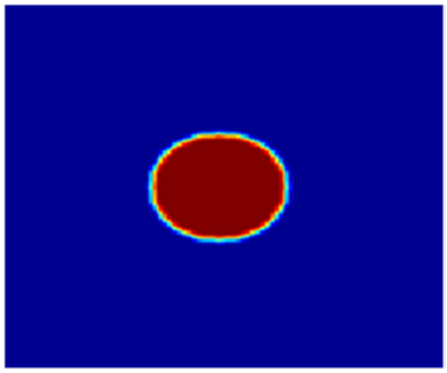


Figure 39: The two-dimensional top hat function

This was built in Matlab® using the following code:

```
%Probe
n = 2^8;
Circ= zeros(n);
I = 1:n;
x = I-n/2;
y = n/2-I;
[X,Y] = meshgrid(x,y);
R = 7;
A = ((X-3).^2 + (Y).^2 <= R^2);
Circ(A) = 1;
Probe=convn(Circ,gauss_2D(7,7,1,1,0),'same');
figure(2)
imagesc(Probe)
```

8.7 Recovered Probe Functions

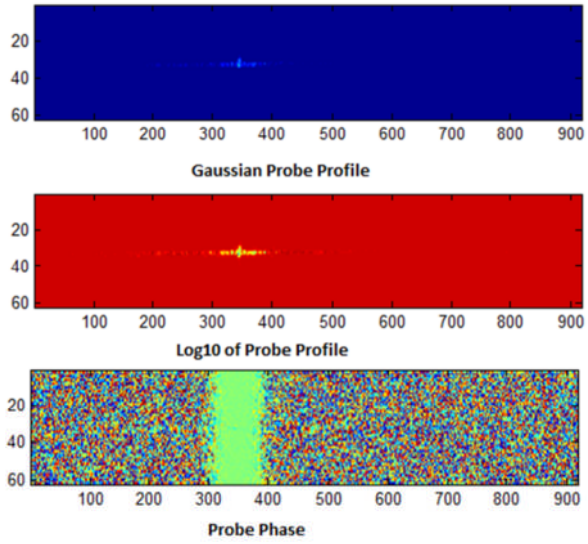


Figure 40: Recovered Probe Profile

9. References

1. Xiang, J., et al., *Ge/Si nanowire heterostructures as high-performance field-effect transistors.* Nature, 2006. 441(7092): p. 489-493.
2. Verheijen, M.A., et al., *Growth Kinetics of Heterostructured GaP–GaAs Nanowires.* Journal of the American Chemical Society, 2006. 128(4): p. 1353-1359.
3. Polák, J., *Electrical Resistivity and Thermoelectric Power of Stacking Faults in Gold.* physica status solidi (b), 1965. 11(2): p. 673-681.
4. Favre-Nicolin, V. and et al., *Analysis of strain and stacking faults in single nanowires using Bragg coherent diffraction imaging.* New Journal of Physics, 2010. 12(3): p. 035013.
5. Crimmins, T.R. and J.R. Fienup, *Ambiguity of phase retrieval for functions with disconnected support.* J. Opt. Soc. Am., 1981. 71(8): p. 1026-1028.
6. Cowley, J., *Diffraction physics*. 1981.
7. Furiouslettuce. *BraggPlaneDiffraction*. 2009; Available from: http://upload.wikimedia.org/wikipedia/commons/thumb/7/74/BraggPlaneDiffraction.svg/800px-BraggPlaneDiffraction.svg.png.
8. Bracewell, R.N., *The Fourier transform and its applications / Ronald N. Bracewell*. McGraw-Hill electrical and electronic engineering series. 1978, New York :: McGraw-Hill.
9. Champeney, D.C., *Fourier transforms and their physical applications*. Techniques of physics,. 1973, London, New York,: Academic Press. x, 256 p.
10. Sayre, D., *Some implications of a theorem due to Shannon.* Acta Crystallographica, 1952. 5(6): p. 843.
11. Kuo, L.H., *Nature and origins of stacking faults from a GaAs/GaAs interface.* Journal of vacuum science & technology, 1997. 15(4): p. 1241.
12. Glas, F., J.-C. Harmand, and G. Patriarche, *Why Does Wurtzite Form in Nanowires of III-V Zinc Blende Semiconductors?* Physical Review Letters, 2007. 99(14): p. 146101.
13. Liz Kalaugher, C.L. *Nanowire transistors outperform MOSFETs*. Available from: http://nanotechweb.org/cws/article/tech/24960.
14. Matsumoto, T.S.a.H. *What is the undulator?* ; Available from: http://www-xfel.spring8.or.jp/cband/e/FEL_PageFile/UndulatorConceptSmall.gif.
15. APS. Available from: http://groups.mrl.uiuc.edu/Robinson/~sector34/layout.html.

16. Peter J. Eng, M.N., Mark L. Rivers and Stephen R. Sutton, , *Dynamically figured Kirkpatrick Baez x-ray microfocusing optics.* Proc. SPIE 3449, 1998. 145
17. Diebel, J., *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors.* 2006.
18. Fienup, J.R., *Reconstruction of an object from the modulus of its Fourier transform.* Opt. Lett., 1978. 3(1): p. 27-29.
19. Bauschke, H.H., P.L. Combettes, and D.R. Luke, *Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization.* Journal of the Optical Society of America. A, Optics and image science, 2002. 19(7): p. 1334-1345.
20. Nikulin, A.Y., *Uniqueness of the complex diffraction amplitude in x-ray Bragg diffraction.* Physical review. B, Condensed matter, 1998. 57(18): p. 11.
21. Guizar-Sicairos, M. and J.R. Fienup, *Phase retrieval with transverse translation diversity: a nonlinearoptimization approach.* Opt. Express, 2008. 16(10): p. 7264-7278.
22. Hoppe, W., *Beugung im inhomogenen Primarstrahlwellenfeld. I. Prinzip einer Phasenmessung von Elektronenbeungungsinterferenzen.* Acta Crystallographica Section A, 1969. 25(4): p. 495-501.
23. Rodenburg, J.M., et al., *Hard-X-Ray Lensless Imaging of Extended Objects.* Physical Review Letters, 2007. 98(3): p. 034801.
24. Rodenburg, J.M. and H.M.L. Faulkner, *A phase retrieval algorithm for shifting illumination.* Applied physics letters, 2004. 85(20): p. 4795-4797.
25. Bunk, O., et al., *Influence of the overlap parameter on the convergence of the ptychographical iterative engine.* Ultramicroscopy, 2008. 108(5): p. 481.
26. Thibault, P., et al., *Probe retrieval in ptychographic coherent diffractive imaging.* Ultramicroscopy, 2009. 109(4): p. 338.
27. Millane, R.P., *Multidimensional phase problems.* J. Opt. Soc. Am. A, 1996. 13(4): p. 725-734.
28. Thibault, P., et al., *High-Resolution Scanning X-ray Diffraction Microscopy.* Science, 2008. 321(5887): p. 379-382.
29. Alves, J., et al., *Calculated atomic structures and electronic properties of GaP, InP, GaAs, and InAs (110) surfaces.* Physical Review B, 1991. 44(12): p. 6188.