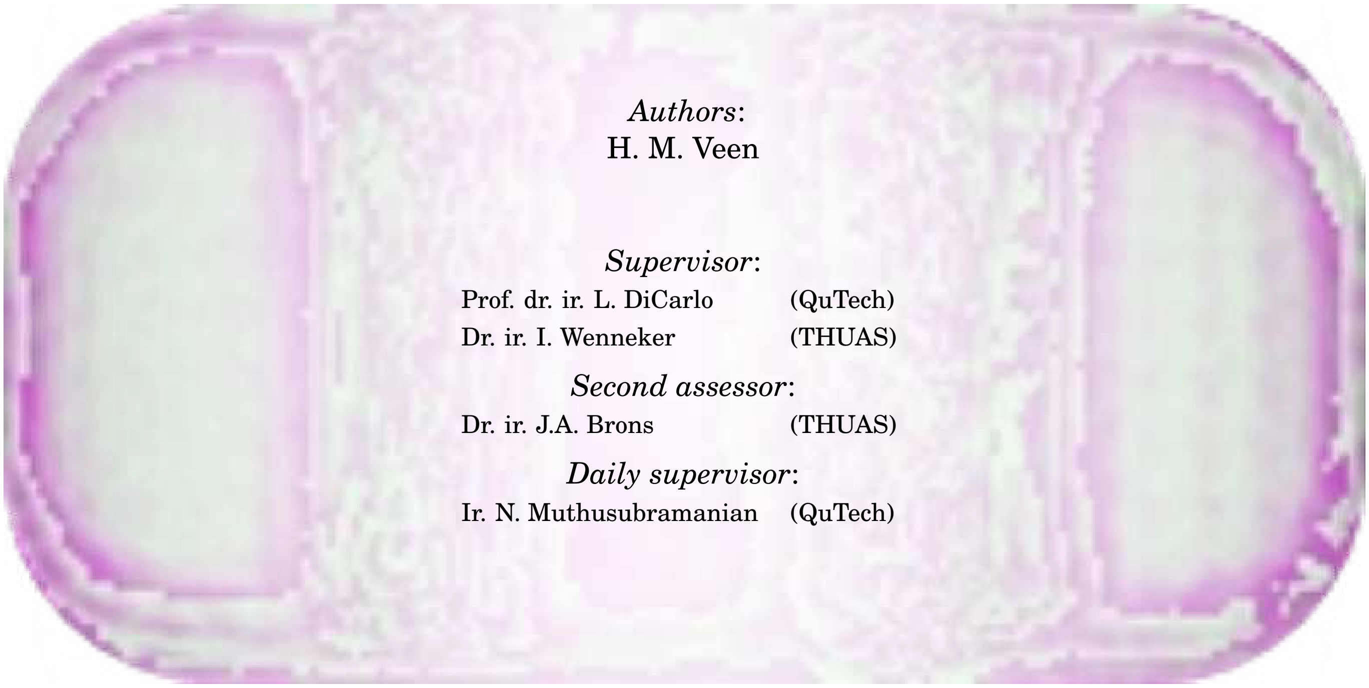
Pyclq: Image analysis suite for fabrication and metrology of superconducting quantum processors



Authors:
H. M. Veen

Supervisor:
Prof. dr. ir. L. DiCarlo (QuTech)
Dr. ir. I. Wenneker (THUAS)

Second assessor:
Dr. ir. J.A. Brons (THUAS)

Daily supervisor:
Ir. N. Muthusubramanian (QuTech)

Date: September 30, 2021

A wise person once said: 'Be thorough not fast.'
So I check everything thrice ever since!

THESIS
submitted in partial fulfilment of
the requirements for the degree of

BACHELOR IN APPLIED PHYSICS

Abstract

This report introduces a customized software tool to enable automation of optical inspection optical and SEM images of a superconducting quantum processor during its fabrication. This is achieved by implementing image processing algorithms using the Python package OpenCV. This suite consist of three components; `pyclq_base`, `pyclq_jj`, `pyclq_ab`. The first component will template match the base layer to its CAD design. The second component will be a validation for the airbridges. These have 3D components and therefore can not be matched to their CAD design. The results had about twice as many false positives for, broken bridges, and no false negatives. The third component measures the width and overlap area for Manhattan-style Josephson junctions using two different filtering methods; k-mean segmentation and thresholding. The results of these three components are used as a tool to understand the sources of spread in the conductance of Josephson junctions therefore optimizing the fabrication process.

Acknowledgements

First of all I want to acknowledge my daily supervisor Nandini. She got me more motivated and sharper than ever with critical questions and clear speech. It helped me keep my focus on the project and allowed me to look at problems from a different point of view; even at even at the middle of the night over a voice call... Second, Leonardo DiCarlo, for creating a team and environment were everyone can thrive in. I want to thank both of them for bringing me in and making this possible.

I want to thank Sean for his collaboration on the automated probe station and working with/next to me during my internship. Of course I could not have done it without the samples from Chistos and Matvey. I will bring more kruidnoten next time for everyone in the Dicarlo lab and am grateful for this internship.

Contents

1. Introduction

Superconducting qubits is one of the most promising platforms for developing quantum computing applications. A lot of research development is carried out by for example Google, IBM and Rigetti[1, 2, 3]. To progress to real world application in quantum computing it is important to scale the number of qubits and suppress noise in the computer. Similar to classical computing, quantum computing at its early stage suffers errors induced by bit-flips and phase-flips [4]. A solution is to increase the number of qubits with nearest neighbor interactions using surface code[5]. One of the key requirements needed to make superconducting quantum processors (SQP) fault-tolerant is to improve the fabrication efficiency yield and scalability. This project: `pyclq` is an attempt to solve these problems by introducing image processing tools to automate device inspection and metrology. This is a pioneering effort in the field of quantum computing and results in the following research question;

Is it possible to automate the optical inspection on SEM and microscope images of a superconducting quantum processor during fabrication?

This automation needs to contains a software package that measures the quality of key components from (SEM) images during fabrication of the the superconducting quantum processor . They will be measured in a software suite `pyclq` [6]. First the basic theory of the quantum hardware is given with some concepts in image processing, chapter 2.

The following chapters are separated into the three main components following the production processes of SQP. The three components are: the base layer in chapter 3, Manhattan style Josephson junctions in chapter 4 and air bridges in chapter 5. Each chapter starts with the fabrication process of the component, resulting in the need for automation. Then the method/pseudo code is given resulting in the results.

The last chapter, chapter 6, will conclude these results and give a final verdict on the possibility to automate the optical inspection. Jupyter notebook tutorials and the source code for `pyclq` can be found on the Dicarlo lab `pyclq` **Github** [6].

2. Theory

The semiconductor industry already applies image recognition to improve critical parts in their production. They are used to improve the number of components that can be used versus the number of parts that are produced, better known as **yield**. The yield can be improved when the critical areas, the areas that are most susceptible to defects, are inspected and possibly repaired. Moreover, an overall improvement can be made if a defect appears to be systematic instead of random by adjusting the fabrication method [7]. An example from IBM is given in figure 2.1 where they have produced different shapes and investigated the automation to find defects in these shapes.

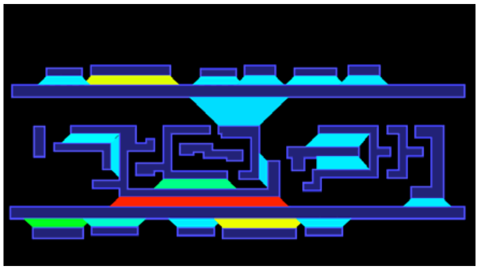


Figure 2.1: An image of a chip from IBM with the areas susceptible to shorts colored. Image is taken from [8]

More recent developments in image recognition and the use of machine learning this process are made on improving the yield of semiconductor quantum-dot qubits [9]. Similarly, `pyclq` is and attempts to apply image recognition during the fabrication of the quantum computers. In the case of `pyclq`, the first layer produced, the base layer, is the most susceptible to defects.

2.1 Harmonic resonators and coplanar waveguide

The transmission lines are coplanar waveguides used for controlling the qubits and connect the SQP to the external cables. A coplanar waveguide is a parallel plate capacitor able to capacitively couple to the components in the chip. The parallel plate capacitors act similar to a dielectric mirror passing electromagnetic frequencies with given wavelength dependent on the designed dimensions of the coplanar wave guide [10, 11, 12].

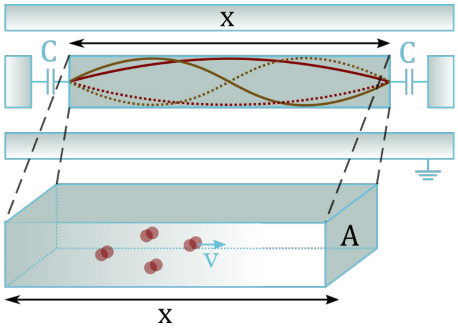


Figure 2.2: A schematic representation of a coplanar waveguide. It is a plate with an cross sectional area A, length x and is capacity coupled C to other resonators. The cooper-pairs have a velocity of v in the coplanar waveguide with a density of n_s. The ground frequency and first order frequency are drawn in the coplanar wave guide. The planes parallel to the coplanar waveguide are grounded to prevent interference. Any electrical components are only shown for ease of the viewer.

The coplanar waveguide can be represented as a harmonic oscillator with inductance L and capacitance C. A harmonic oscillator can be compared to an LC-circuit with an impedance given by Z_0;

$$Z_0 = \left(\frac{1}{i\omega L} + i\omega C\right)^{-1} \tag{2.1}$$

The ground frequency can be found when the capacitive and inductive resistances cancel out. Classically any higher order is will be multiple of the frequency of the ground state resulting in;

$$\begin{gathered} \omega_0 = \frac{1}{\sqrt{LC}} \\ \omega_n = \omega_0(n+1) \end{gathered} \tag{2.2}$$

To prevent higher harmonics airbridges are incorporated over the transmission lines, chapter 5. Adding this additional dimension of ground plane suppresses higher harmonic. The ground frequency and first harmonic frequency are drawn in figure 2.2.

The alternating current in a coplanar waveguide causes changes in the magnetic flux, which in itself induces a negative voltage opposite to the change of current flowing through it. Inductance can be interpreted as a property to counteract change of current in a conductor. The inertia of the electrons, current, in a cryogenic cooled superconductor will behave similarly, resulting in a kinetic inductance. [10, 12, 13] In an ideal situation the kinetic inductance for a coplanar waveguide on a non-magnetic substrate can be derived from its dimensions by;

$$L_k = \frac{m_e}{2n_s e^2}\frac{x}{A} \tag{2.3}$$

With m_e the mass of an electron, n_s the number of electrons in the wave guide that will form cooper pairs, A the area of the cross section of the waveguide and x the length of the waveguide.

2.2 Qubit

Classical computing works by encoding information as a binary digit either 0 or 1. Likewise, information is encoded in the quantum state of a qubit represented by Dirac notation as $|0\rangle$ and $|1\rangle$. These form the basis states which comprise a pure qubit state due to a linear quantum superposition.

The main benefit of the qubit is its quantum entanglement allowing a super position of different different binary strings. This makes it possible for a quantum computer to calculate with multiple binary strings at once reducing calculation time [14, 15] .
The transmon qubit is comprised of two capasitve pads and a superconducting quantum interference device (SQUID) containing two parallel Josephson junctions, figure 2.3. The SQUID allows tuning of the Josephson coupling energy [16, 17].

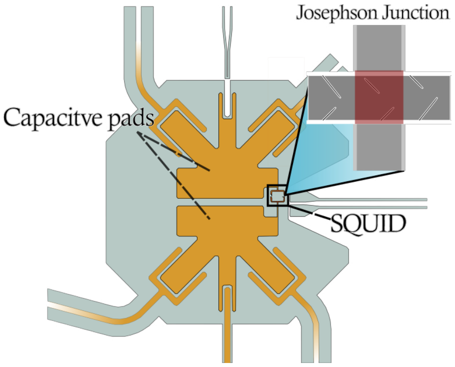


Figure 2.3: Transmon qubit designed by DiCarlo lab, nicknamed 'Starmon'; it contains two capacitve pads and a SQUID loop. In the squid loop are two Josephson junctions. The overlap area of the two electrodes of the Josephson junction is marked red.

To be able to measure if the qubit is in a $|0\rangle$ or $|1\rangle$ state the frequency difference of the the ground frequency to the first exited state can be measured [18]. This difference can be be found by obtaining the energy stored in sum of capacitive effects E_C and energy stored in the Josephson junction E_j;

$$f_{01} = \sqrt{8E_jE_C} - E_C \tag{2.4}$$

During fabrication the energy in the Josephson junction can be related by the room temperature resistance R_{RT} and material/geometrical properties of the aluminium and aluminium oxide M;

$$E_j = \frac{M}{R_{RT}} \tag{2.5}$$

By designing the geometrical properties of the junction accurate there is less need for adjustment and therefore error. `pycl_jj` will measure the overlap area to gain a better insight in these geometric properties.

2.3 Image processing

In `pyclq` grayscale SEM or microscopy gray scale images are processed to validate or measure components. Image processing contains multiple filters and functions to obtain the desired image. These filters are referred to as nodes and when then nodes are weighted by previous notes it is referred to as a (neural) network. The programming language used will be Python with the main modules of Numpy and OpenCV[19, 20].

Image processing at its core contains four nodes of which the second and third can be repeated multiple times to improve the result in the network. First the image needs be acquired. In this case using a camera or SEM. Second and third, an image is filtered and its desired features are separated from the image [21]. Finally, these segments are measured or validated. A simplified schematic representation is shown in figure 2.4.

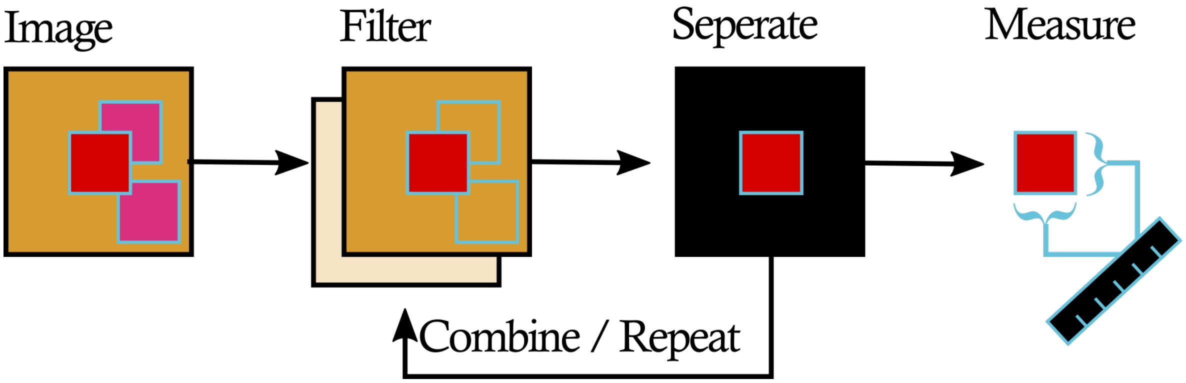


Figure 2.4: A simplified representation of image processing, first an image is obtained, then it is filters to make the desired feature more prominent. This feature is separated from the image and finally can be measured.

2.3.1 Separating an feature

To be able to detect features or measure an area, the feature or area of interest needs to be separated from the background. Some of the most common methods used in `pyclq` are thresholding, segmentation and masking [20].

Thresholding separates a image into two, one for pixels with a value above the threshold and one for pixels with a value below the threshold.

Segmentation is able to split an image into multiple images. Each image contains a range of pixel values known as bins. In addition, this can be combined with a k-means algorithm for automation and self learning[22, 23], chapter 2.3.3.

Masking is similar to a template used to add or remove material from a chip, with masking an area on an image can be kept whilst the rest will be masked by the mask. This mask could be the result of a previous filter over the image in the network as is applied in chapter 5.2.

After separating an image the desired feature can be measured, either by detecting the edges of the separated feature (hull detection), by counting pixels of the desired feature or by matching it to an another feature.

2.3.2 Convolution filters

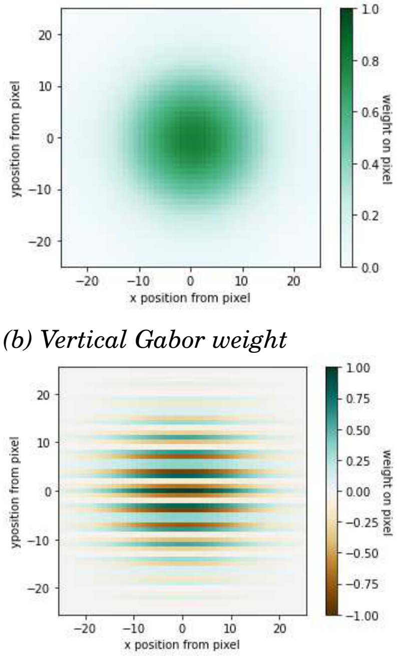
Convolution filters can be applied to improve the quality of the desired feature that will be separated. A convolution filter, when applied to an image, are known as blur filters as they often will blur out an image.

A convolution filter will add or subtract the neighboring values in an array or image. This is used to reduce noise or filter for a feature in the image. With the right filter it will be easier to separate a desired feature.

Four major convolution filters for image processing used in `pyclq` are Gaussian filters, Gabor filters and opening/closing.

Gaussian filters will blur an image, taking the original pixel value and following a Gaussian curve for the strength of the neighboring pixel. This can be utilised to reduce noise, however an image can become blurry if the filter is too large. An extreme example is visible in 2.5(a).

Gabor filters will detect directional patterns or lines by applying a Gaussian filter with alternating the strength of the neighboring values 2.5(b). An example of its use is found in chapter 4 where the charging effect of the electrodes will be reduced with this filter.



(a) Gaussian filter

(b) Vertical Gabor weight

Figure 2.5: A visual representation of the weights against the distance from a pixel in a Gaussian (a) and Gabor filter (b).

Opening/closing is used to highlight pixels or reduce noise in an image. With **dilation** the weight of the higher neighbouring pixel values will be increased in the convolution filter and therefore increase the amount of bright pixels. Similarly, with **erosion** the weight of the lower pixel values will be increased in the convolution filter and reduce the amount of bright pixels. Opening is dilation followed by erosion and closing is erosion followed by dilation. Opening and closing are used in chapter 5.

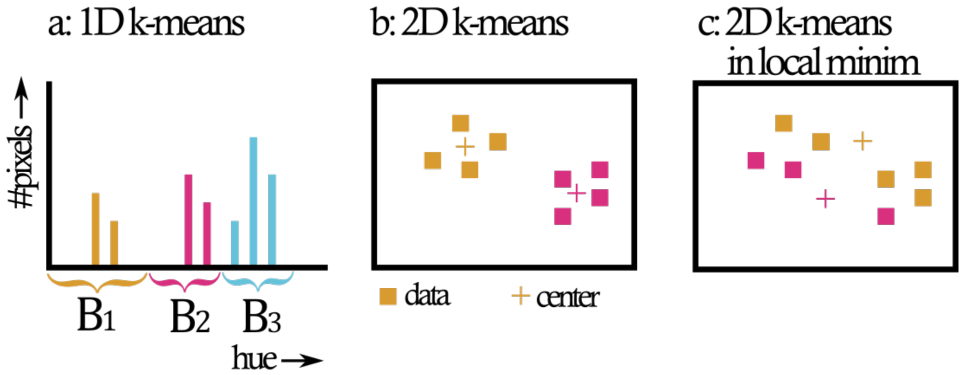


Figure 2.6: a visual representation of the k-means method wherein (a) all the hues (1D) of an image are separated by occurrence in an image (b) the a separation of two clusters from a scatter in a 2D data set and (c) an exaggerated example of wrong clustering stuck in a local minimum.

2.3.3 K-means algorithm

A k-means algorithm is a clustering algorithm that is able to separate data sets into different categories that are grouped together. By searching for the nearest point from an origin the data set can be separated into different bins. This is often used in image processing and machine learning as it is able to separate different colors or features in an image or multidimensional space [20, 24, 25]. The only downside is that a k-means algorithm can get stuck in a local minimum. This can be prevented by using either smart starting points or repetition of the algorithm with different starting points. The latter takes more calculation power and does not always give a desired result [22]. An example of the k-mean algorithm is given in figure 2.6.

In `pyclq` k-means will be used to separate different pixel values corresponding to edges, different material or heights in the chip. Making smart use of various starting points will prevent the algorithm from getting stuck in local minima.

2.3.4 Template matching

To find similarity between two images template matching can be applied. A correlation can be obtained by comparing the individual pixel between an image and template [26]. In `pyclq` template matching is applied with a normalized cross correlation to obtain similarities. From the OpenCV documentation [20] follows for the cross correlation R;

$$R(x,y) = \frac{\sum_{x',y'}\left(T'\left(x',y'\right)\cdot I'\left(x+x',y+y'\right)\right)}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}} \tag{2.6}$$

where,

$$T'\left(x',y'\right) = T\left(x',y'\right) - 1/(w\cdot h)\cdot \sum_{x'',y''} T\left(x'',y''\right)$$
$$I'\left(x+x',y+y'\right) = I\left(x+x',y+y'\right) - 1/(w\cdot h)\cdot \sum_{x'',y''} I\left(x+x'',y+y''\right)$$

On an image I with a template T. Whereas (x',y') and (x'',y'') are the relative location of the pixels on the template with a width w and height h at an given location on the image (x,y) and I' and T' are normalized pixel value. These values are normalized by subtracting the average value of the template from the pixel at T' or the average value from the image under the template at I'. These normalizing factors are not always necessary if the image have a similar brightness. This will be used as a part of the validation for finding defects in the base layer and for airbridges chapter 3.2 and 5.2.

If an image is larger than as the template multiple matches for the template can be made over the image. This results in a matrix of matches and can be used for feature detection or detecting a part in an image. The highest value in this matrix is the most likely location of the template on the image. An example is find Waldo in a 'Where's Waldo?' image [27].

3. Base layer

The base layer is the first step in the fabrication, covering the largest surface area of the chip. In this layer, a bulk of the functional components of a SQP are defined namely transmission lines namely feedlines, flux bias lines and microwave drive lines,readout resonators, Purcell resonators and superconducting buses comprising the resonating transmission lines and qubit capacitor pads. A Surface-7 layout is shown in figure 3.1, where each component is highlighted.

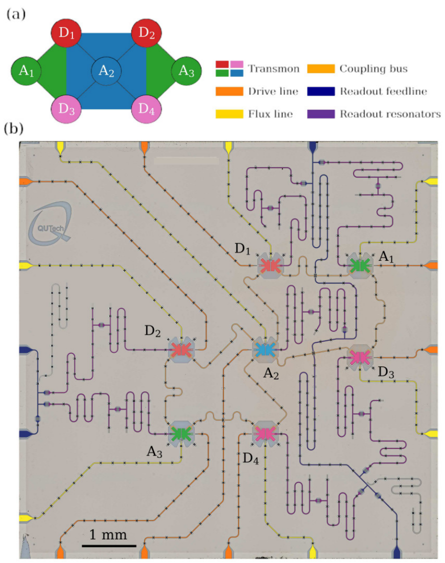


Figure 3.1: Surface-7 quantum processor; (a) Layout of nearest neighbor interaction in a Surface-7 device (b) Optical image of the quantum hardware with added false-color to emphasize different circuit elements. Figure taken from [5]

3.1 Fabrication

The base layer fabrication process is outlined in figure 4.2. The first step involves cleaning of a 100 mm diameter high-resistivity silicon wafer with organic solvents followed by inorganic treatments with hot piranha solution.

Prior to deposition of the superconducting base layer, the wafer is treated with buffered oxide etch (BOE) solution which removes native oxides from the silicon surface. The wafer is transferred to a physical vapor deposition system for wafer metallization with 200 nm of niobium titanium nitride (NbTiN). The pattern transfer of the CAD layout of the SQP is done using electron beam lithography followed by subtractive patterning of the base layer using a combination of dry and wet etching techniques. For the purpose of this hybrid etching process, the subsequent step after NbTiN deposition is the introduction of an inorganic 'sacrificial layer' which is necessary for the terminal wet etch step as shown from step 8.

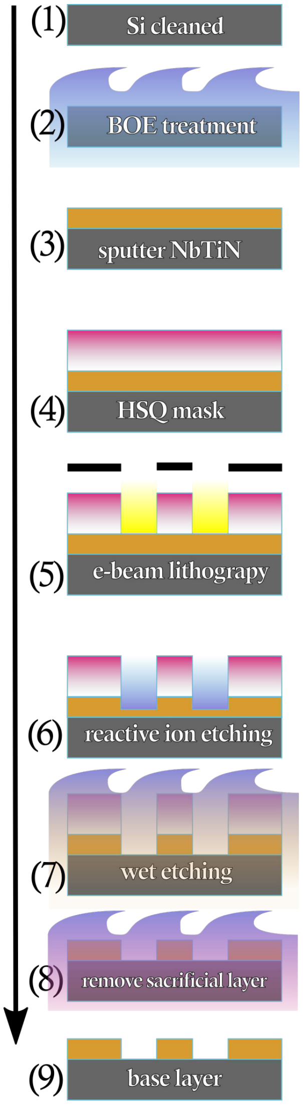


Figure 3.2: Fabrication process flow for the base layer.

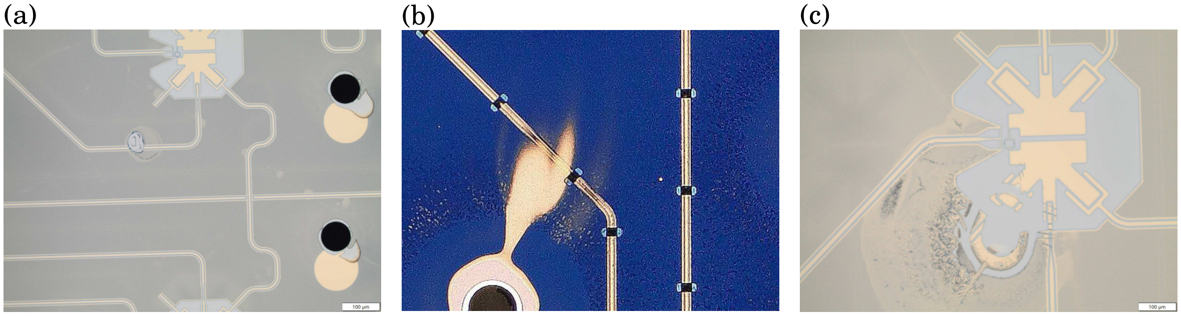


Figure 3.3: Examples of damage in the base layer, whereas; (a) discontinuities interrupting the transmission lines leading to open circuit,(b) a short due to not etching away around the transmission line and (c) damage to the qubit due to discontinuity of resist.

Defects in base layer

As with any manufacturing process, defects can accumulate at every fabrication step of the base layer. Some of the possible sources which introduce defects in the base layer are as follows:

- Introduction of particles at any or every fabrication step.
- Irregularities in resist during spinning.
- Peeling/cracking of the resist due to end of shelf-life and high/low dose during e-beam exposure.
- Non-uniform pattern development.

Any damage to the base layer could occur during or after this process. Examples of damage in the base layer are given in figure 3.3. Such unintentional changes to the geometry of the transmission lines may result in altered function and could even render it useless due to the lines being shorted or opened. Therefore `pyclq_base` will help detect defects in the transmission lines and base layer will be a module that will help catch such pattern transfer errors.

Effect of SQP scaling

As the device size increases from Surface-7 to Surface-17, the number of components in the SQP scales almost linearly. For example, the Surface-7 layout has 7 qubits with a dedicated flux bias line, readout resonator, and microwave drive line, 2 feed lines and 12 superconducting buses. Such scaling efforts is needed for realization of quantum fault-tolerance. However this also means that achieving full yield of device components becomes more challenging. This is where the need for automation becomes evident, necessitating the development of `pyclq_base`.

3.2 pyclq_base method

The code can be found at the Dicarlo lab pyclq **Github** [6] and a schematic representation is in the appendix. pyclq_base is a Python script to help to detect errors in the base layer. Using the CAD layout an image of a perfect chip is created and this is matched to the base layer to find any defects. Only the key components are inspected or it will take between 16 hours or 16 days to inspect an entire device due to frame-rate and the underlying software for acquiring the images. The method of the script will follow the following steps.

Replicating the chip from the CAD files of the masks for e-beam lithography to an image of the device. By choosing the same scale for pixels as the acquired image this replicated chip can be used for template matching.

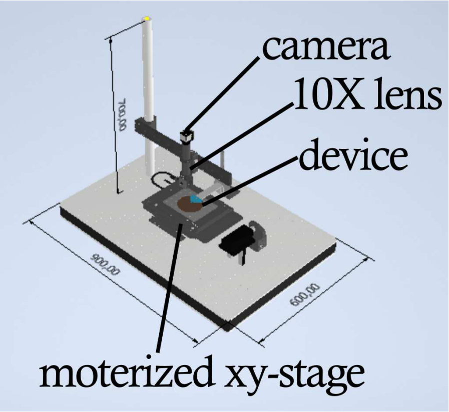


Figure 3.4: APS-LASIQ system, this is an automated room-temperature resistance measurement setup with the components for pyclq marked. The original image is taken from [28]

It should be noted that although the design concepts between pyclq_base and pyclq_AB are similar, the airbridges can't be matched to a template from the CAD design due to its 3D feature; therefore they have their own validation method, see chapter 5.

The CSV is created by dicing the replicate chip in a chessboard pattern any tile containing a key component is put in the CSV file. In addition the centers of the qubits are put in the CSV file as extra check as any error will be critical.

Image acquisition is achieved with the APS-LASIQ system, this is an automated room-temperature resistance measurement setup designed and built by members of the DiCarlo lab and is located in the Kavli Nanolab in TU Delft. A schematic of the APS-LASIQ system is shown in figure 3.4 .

For pyclq, the components which are used are the camera (Thorlabs Model:DCU224M), the 10X objective lens and the motorized XY stage (Thorlabs Model:MLS203-1). The various hardware components connected to the APS-LASIQ system are controlled by a LabView interface, developed by Dr. Matvey Finkel and Sean van der Meer [28].

The variable parameters of the camera in the software are the gain, closing time and frame rate. The closing time and frame rate is kept at a constant 200 ms and 5 Hz respectively. For manual inspection the gain is set to 400 as the human eye is better in catching bright features. However, to make use of the larger range of a gray scale image the gain is set to 300 for pyclq.

By sending the CSV file of chip coordinates of the critical locations to the Labview program, images of these locations can be obtained by the `pyclq` module. These images are stored and can be opened by `pyclq` with the same CSV file. The size of the images is 1280 x 1024 with 2.35 ± 0.05 pixel/μm for a 10X magnification.

Template matching is applied between a cutout of the replicated chip and the acquired images. By making the cutout smaller as the acquired image and matching at the highest cross correlation match any drift will be mitigated. Repeating this for multiple locations at the base layer results in a map where any differences from design are highlighted. This will help locate any defects in the base layer.

Validation is done by manually inspecting the SQP and comparing it to the highlighted areas created by `pyclq_base` any faults can be detected. By counting the images with false positives and false negatives and comparing it to the total amount of images taken from the SQP this results in a correctness in faulty detection in the base layer by `pyclq_base`.

3.3 Results

As pyclq_base is still in development there are no final results yet. However, template matching is already applied successfully in pyclq_ab, chapter 5.

In figure 3.5 an replication of a perfect chip is shown with all the different layers from the CAD design. The original image is at the same resolution as the camera, but due to compression the resolution is lowered. The full template match has not been finished. It is planned after this report.

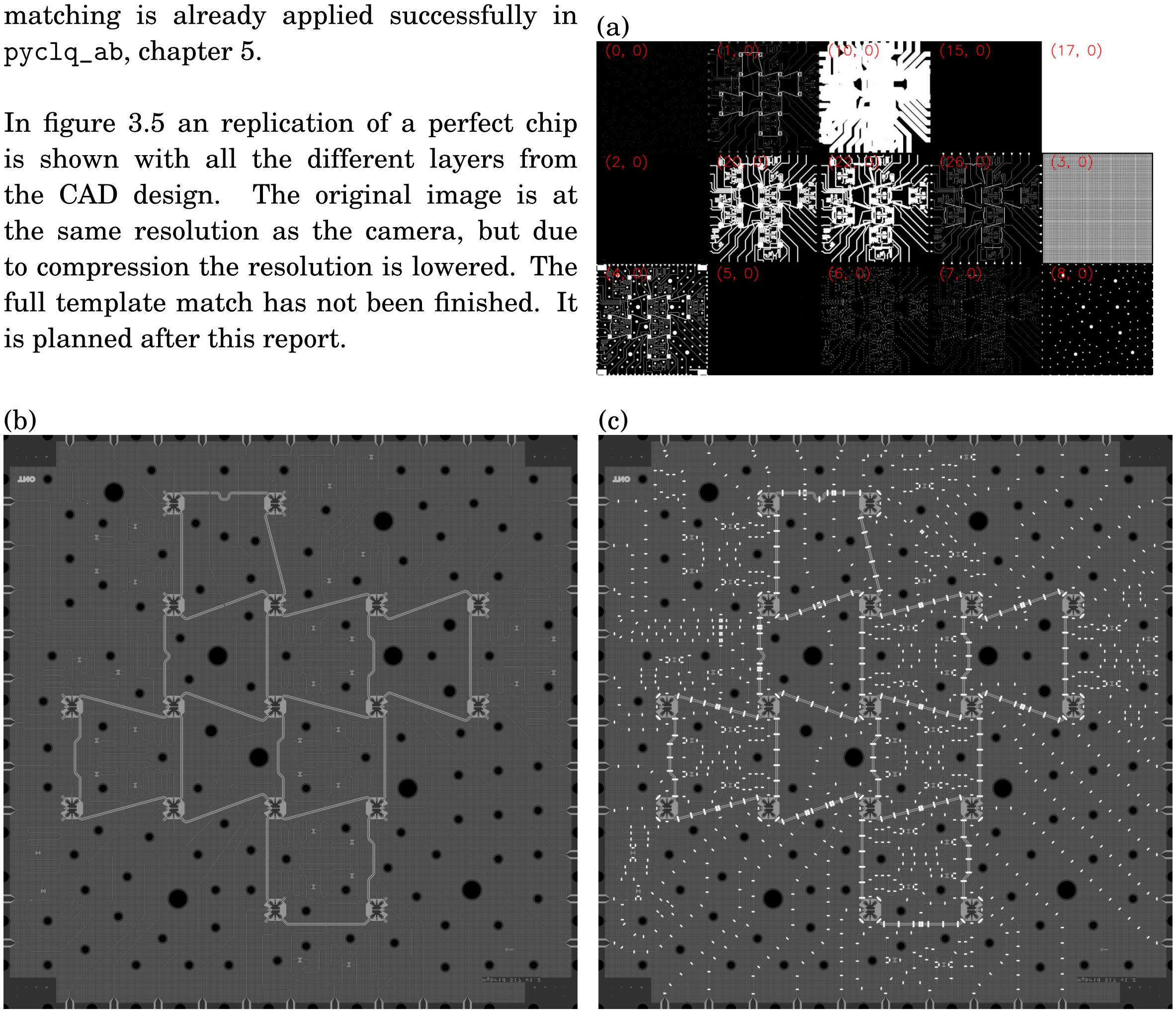


Figure 3.5: An image generated from the CAD design; (a) all the different layers and segments, (b) a of the base from these layers and (c) the full chip with 2D airbridges.

4. Josephson Junctions

Superconducting tunnel junctions, also known as Josephson junctions (JJ) consists of two overlapping electrodes separated by a thin insulating layer. One of the most widely used materials for fabricating Josephson junctions is aluminium, since a thin tunnel barrier of aluminium oxide can be grown in a controlled manner. Josephson junctions are contained within the transmon qubit, which is the fundamental unit of a quantum computer. The working principles of transmon qubit is similar to an LC circuit.The parallel metal pads contribute to the capacitance and the JJ contribute a highly non-linear inductance. This important difference from a linear oscillator such as resonators gives rise to the non-equal energy-level spacings. This is needed in order to limit the functioning of a qubit as a two-level system [29].

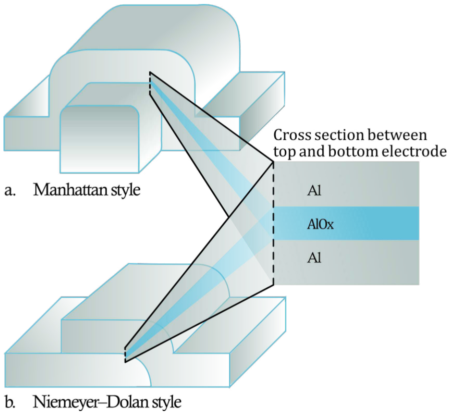


Figure 4.1: A schematic representation of (a) the Manhattan style junction and (b) Niemeyer-Dolan style junction. It shows a cross section how the aluminium and the aluminium oxide layer is stacked in these junctions. Schematic not to scale.

4.1 Design and fabrication

The fabrication process for JJs is shown in figure 4.2. Josephson junctions can be fabricated in different geometries depending on the 'shadowing' mechanism. The earliest technique developed for pattern transfer of JJs with sub-micron feature size is using the Niemeyer-Dolan bridge technique [30]. Such junctions fabricated using a resist bridge as the shadowing mechanism are called Dolan-bridge junctions, shown in figure 4.1 (a).

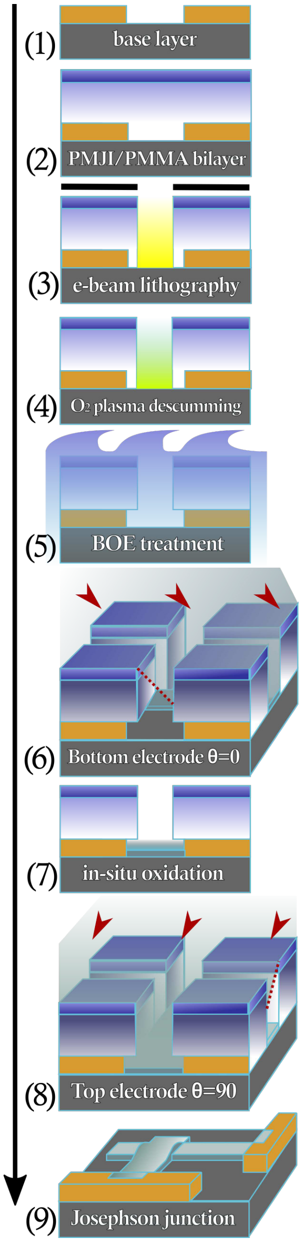


Figure 4.2: Fabrication process flow for the base layer.

Another fabrication process for JJs was developed by changing the deposition tilt angle and resist stack thickness [31]. These 'bridgeless' junctions have been termed in previous literature as Manhattan-style junctions due to the orthogonal layout of the Al contacts as shown in figure 4.3 [32, 33]. As mentioned in the theory section 2.2, the qubit frequency can be related to the room-temperature resistance of the JJs. R_N is further dependent on two factors, namely the overlap area of the electrodes and the tunnel barrier thickness.[34]. By sweeping the junction widths, it is possible to vary the range of JJ resistance. In the case of Manhattan-style junctions designed for this work, the widths of the bottom electrode is varied while the top electrode is kept constant. Prior works in wafer-scale fabrication revealed a systematic variation of junction conductance as a function of its position on the wafer. A geometric model has been proposed to linking the conductance variation to a variation in the overlap area due to shadowing of Al deposition by the resist. This may imply that when the junction pattern is at the center, the metal deposition is perpendicular. On the other hand, when the junction pattern is at the edge of the wafer the Al is deposited at an angle [17, 35]. However, more work is needed to quantify the extent of variation in the overlap area. `pyclq_jj` is a tool to automate the extraction of the overlap area from Scanning Electron Microscope (SEM) images.

4.2 `pyclq_jj` method

The code can be found at the Dicarlo lab `pyclq` **Github** [6] and a schematic representation is in the appendix. The SEM images were acquired by N. Muthusubramanian using the Hitachi Regulus 8230 system in the Kavli Nanolab. `pyclq_jj` measures three variables from the SEM images; the width of the top horizontal electrode (w_{top}, the width of the bottom vertical electrode w_{bottom} and the overlap area $p_{overlap}$. After importing the grey-scale images taken with SEM `pyclq_jj` will process them in the following steps:

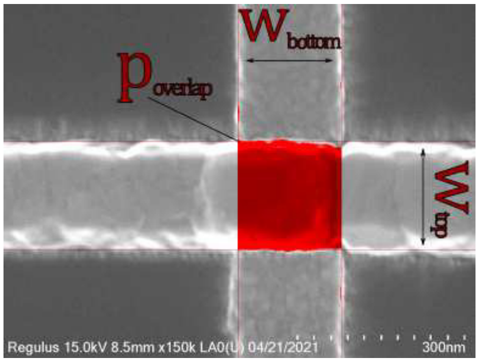


Figure 4.3: A example of an SEM image of blackbird 4, at the 5_5_right. Where the area measured is $p_{overlap}$ and the width of the horizontal top electrode and vertical bottom electrode.

Reading the image and its metadata from a user specified folder. The pixel size is obtained from the metadata.

Filtering the images is done using two techniques, segmentation or thresholding as described in table 4.1 below. Both methods return the edges of the electrodes as well as the widths of the electrode.

Table 4.1: A comparison of the separation method between thresholding and segmentation. In these methods the widths of the electrodes are measured as well.

Step	Thresholding	Segmentation
Pre-filter	An option for a Gaussian filter is possible but not necessary.	An option for a Gaussian filter is possible but not necessary. In addition, a horizontal and vertical Gabor filter is applied to reduce the halo effect caused by the charge of the SEM. The result of the Gabor filter will be subtracted from the image to reduce the halo.
Separation	The average pixel value of the image is obtained, this is multiplied by a set range of thresholds to detect the edge. This range of thresholds is used to filter the image and obtain the best edge.	The image is segmented into 6 bins with the use of a k-mean segmentation the amount of segmented bins can be changed and `pyclq_jj` will assign these segments automatically. The lowest two bins (or 30%) will be the background.
Top horizontal electrode	w_{top} is detected by measuring the sum of pixel values over the y-axis. The top horizontal electrode will be filled in to reduce noise to find the bottom electrode.	The top electrode will be the top two bins (or 30%). w_{top} is detected by averaging y coordinates of the edges of these bins.
Bottom vertical electrode	Similarly, w_{bottom} is found by summing value of the pixels over the x-axis. Within the range of thresholds, the widths closest to the center are used to estimated the overlap.	The leftover bins correspond to the bottom electrode. Remnant top electrode pixels are further separated by taking the edges in the same method as the thresholding method resulting in w_{bottom}.

The overlap $p_{overlap}$ will be obtained after the edges have been found. Then the pixels of w_{top} above w_{bottom} are counted and multiplied by the pixel size. Al the measured parameters are stored as .csv file together with the widths and filter settings.

A GUI inspection can be done for the thresholding method. A simple GUI with a fillable form for selecting the folders and changing the parameters is made. After measurement the overlap areas are shown in tiles and adjustments can be made with a drag-and-drop function where only the bottom electrode needs to be dragged for the thresholding method.

Validation is done by comparing data from `pyclq_jj` and manually extracted data using the image analysis software ImageJ. From the two measurement methods, the data $p_{overlap}$, w_{top} and w_{bottom} were obtained from different devices as described in table 4.2. The thickness of the AlO_x tunnel barrier is assumed to be a constant across all the junctions, therefore all conductance variations are attributed to geometric variations of the sampled junctions. In addition, the measured overlap areas by `pyclq_jj` have be marked and attached in the appendix.

4.2.1 Devices

The junctions deposited on five devices, these have been analyzed for this project. A description of the device layout is given below.

Blackbird 1 (BB1) was a device series fabricated on a NbTiN sputtered base layer consisting of 8 chips of size 17 x 17 mm. The junctions were deposited on each device separately. The device design has a SQUID layout, so from each location, 2 junctions were imaged, highlighted with a blue box as shown in figure 4.4. The designed width of the electrodes is; w_{top} = 160 nm and w_{bottom} = {84, 136, 196} nm.

Blackbird 2 (BB2) has the same layout as BB1, however it is a half-wafer layout where all the junctions were deposited in a single step. The same set of junctions have been imaged for this device as BB1. The designed width of the electrodes is; w_{top} = 160 nm and w_{bottom} = {84, 136, 196} nm.

Blackbird 4 (BB4) was a full-wafer dataset with the junctions arranged in a 35 x 35 square array. All the junctions in BB4 are of the same designed widths, with both w_{top} and w_{bottom} = 200 nm.

Jason was a Surface-17 device which was characterized in a dilution fridge. The widths of the electrodes in this device are; The designed width of the electrodes is; w_{top} = 140 nm and w_{bottom} = {140, 220, 276} nm.

Nighthawk 2 was a full-wafer dataset fabricated on a bare silicon substrate with both junctions and test pads made fully of aluminium and has the same layout as BB4, including the dimensions of the electrodes.

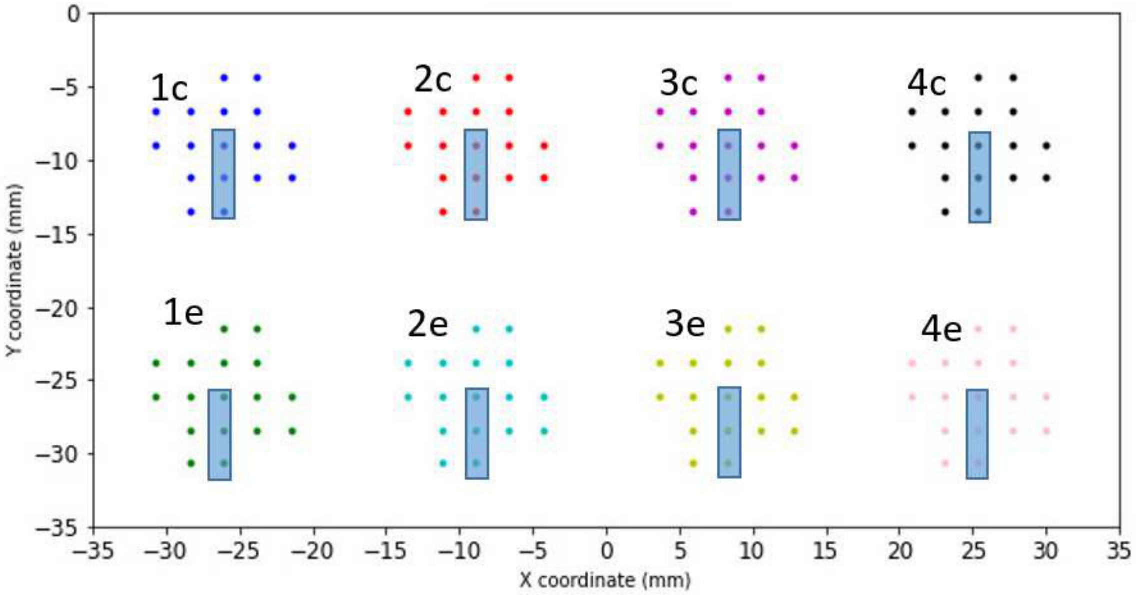


Figure 4.4: Device layout for Blackbird 1 and 2. The junctions analyzed are highlighted by the blue box. [35]

4.3 Results

Below the results of the scripts, first the areas and widths measured with `pyclq_jj` are compared to the manual measured areas and electrode widths in ImageJ. The results of the five datasets are shown in table 4.2.

Table 4.2: An comparison of the successful measurements of `pyclq_jj` and the found conductance over the area. The nighthawk dataset is left out of the average R^2 as it does not fit the data.

	ImageJ				Thresholding					Segmentation				
	failed	cond $\mu S/\mu m^2$		R^2	failed	range	cond $\mu S/\mu m^2$		R^2	failed	K	cond$\mu S/\mu m^2$		R^2
BB1	0/73	2992	± 90	0.73	11/73	[1.2-2.2]	3007	± 89	0.74	6/72	6	2990	±145	0.32
BB2	0/48	3339	± 124	0.72	6/48	[1.2-2.2]	2961	± 109	0.73	1/48	5	3081	± 112	0.73
BB4	0/34	2873	± 179	0.59	0/34	[1.1-2.2]	2890	±231	0.34	2/34	6	2489	± 223	0.20
Jason	0/34	2165	± 16	0.98	1/34	[0.9-1.8]	2145	±39	0.89	1/34	6	2008	± 40	0.86
Nighthawk	0/34	1993	± 120	0.06*	1/34	[1.2-2.2]	2208	± 176	-0.53*	1/34	6	1659	± 101	-0.04*
Average	100%			0.75	91%				0.67	96%				0.52

*left out of the average

The thresholding method had the most failed measurements, this was mostly due to the sharpness of the edges. This was mostly at the smaller overlaps of the Josephson junctions. The brighter and sharper the edge of the electrode the better the thresholding method works. As the thresholding method takes the brightest pixel values of the edge it was able to obtain the overlap area more accurate as the widths of the electrodes. This means a high contrast image with a small halo performs better as a low contrast image. Therefore, BB4 performed 100% and Nighthawk the worst for the Thresholding method.

The segmentation method was able to handle a wider variety of brightness and contrasts in the image. Therefore it was able to obtain more data from the Blackbird 1 image set and Nighthawk image set compared to the thresholding method. However, the halo effect was still too strong for the segmentation method and increasing the strength of the Gabor filter would remove the edge of the electrodes. This resulted in a larger perceived width of the electrodes using the segmentation method. Gaussian filters and opening had been applied, but with no success. As the halo effect is less prominent for smaller junctions and junctions with low contrast, segmentation performed better for these junctions as the thresholding method.

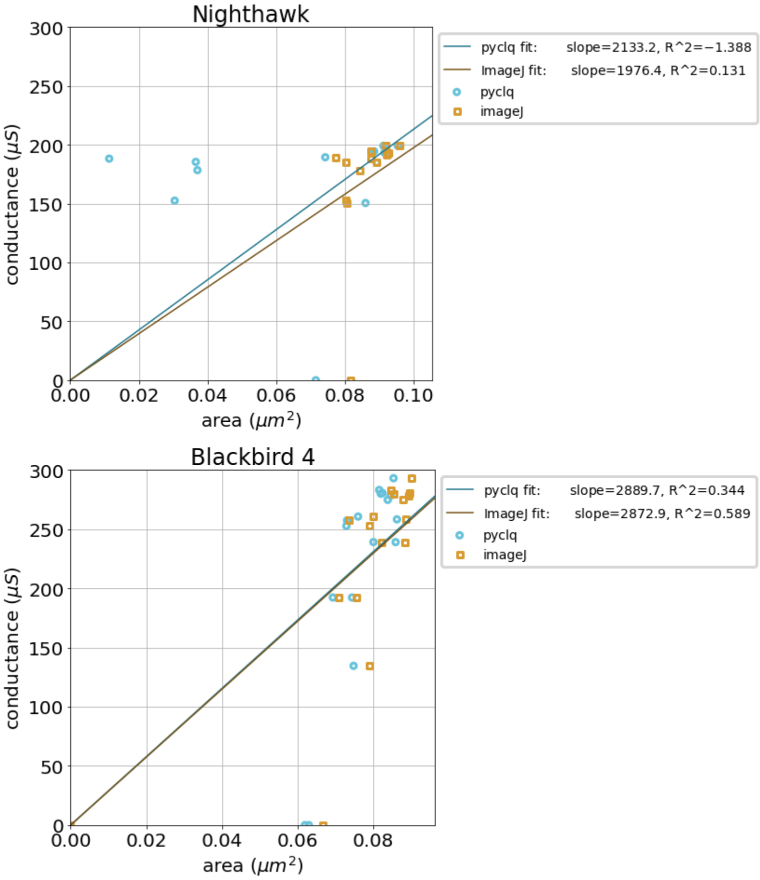


Figure 4.5: Fits of the conductance against the overlapping area for nighthawk and blackbird 4. Both measurements are shown from ImageJ and the `Pyclq_jj` thresholding method with the coefficient of determination, R^2. Blackbird 4 had a single junction at $0.08\mu m$ and two shorts around $0.06\mu m$, these shorts are omitted from the fit.

As a result the fit of the measured conductance against the surface area of the Josephson junction was plotted for the thresholding method together with the manual measured ImageJ fit. Any shorts are omitted from the fit and are shown as having 0 conductance. Moreover, the conductance per unit area was calculated individually and shown in figure 4.6. The spread and center of the histogram should correspond close to the slope of the fits as it is the same dataset.

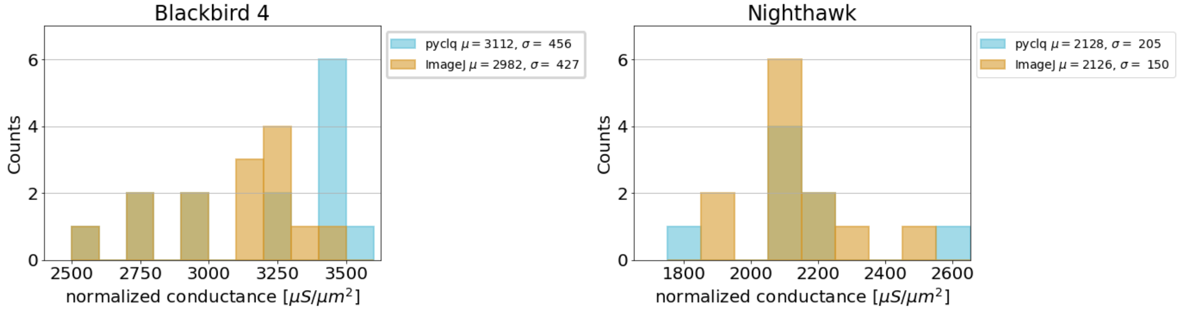


Figure 4.6: A histogram of the normalized conductance over the area measured with `pycl_jj` *and hand measured by ImageJ. The Blackbird 4 and Nighthawk data set are shown here, all histograms can be found in the appendix. For both histograms the binsizes are the same with a width of* $200\mu S/\mu m^2$ *where pyclq and ImageJ overlap the colors are mixed to a darker brown.*

5. Airbridges

The airbridges is the last step in the fabrication, placing 3D airbridges over all transmission lines on the SQP. The airbridges have two main purposes; helping to ground the device and suppressing higher harmonic oscillation in the transmission lines, chapter 2.1. Moreover, airbridges are used to make crossovers where two transmission lines meet making one transmission line pass another without physically touching each other. Damage to an airbridge could cause a short and similar problems as a broken transmission line, chapter 3.

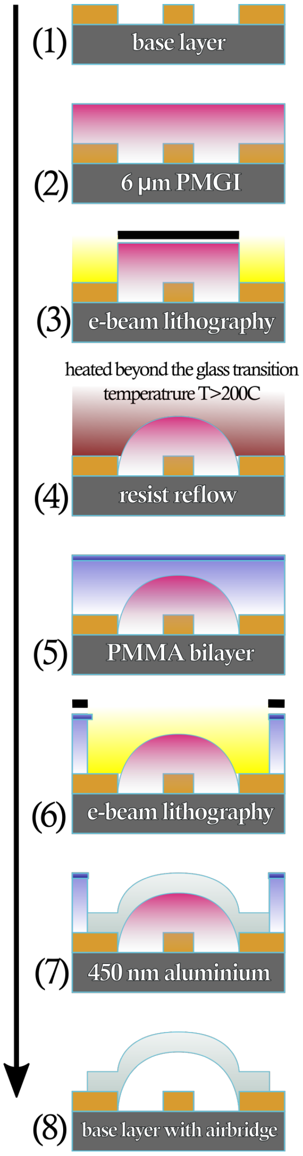


Figure 5.1: The different steps taken in order to make an airbridge on top of the base layer.

5.1 Fabrication

The airbridge fabrication process is outlined in figure 5.1. The first step is spinning of a 6 μ thick layer of polymethyl glutarimide (PMGI) resist. The first pattern transfer of the CAD layout for the aibridges is done using e-beam lithography followed by heating the device to 200°C for resist reflow, figure 5.1 (4). After the device has cooled down the second resist layer polymethylmethacrylate (PMMA). followed by a second pattern transfer with e-beam lithography. On this second mask 450 nm aluminium is deposited. When the PMMA and PMGA is removed only the airbridge is left behind.

`pyclq_ab` will significantly reduce the number of bridges that need to be inspected, reducing labor. As the size of the SQP scales so does the amount of airbridges. Surface-7 has over 500 airbridges, as a Surface-17 has over 1000 or even more than 1500 airbridges in the latest versions. Manual inspection becomes labor intensive and automation is needed.

5.2 `pyclq_ab` method

The code can be found at the Dicarlo lab `pyclq` **Github** [6] and a schematic representation is in the appendix. `pyclq_ab` is a tool to help detect faulty and broken airbridges. Due to the 3D nature the airbridges, the images can't be directly be matched to their template as the curves will reflect light in a gradient. Therefore a different method is needed to find defects in the air bridges as `pyclq_base`. In order to validate the airbridges the following steps are taken;

Image acquisition is done with the APS-LASIQ system in the same manner as acquiring images for the base layer, chapter 3.2. With two differences; the camera is centered on top of the bridge and the gain of the camera is set to 200. This to reduce glare and shape deformation due to the 3D nature of the aluminium from airbridge.

Image segmentation was applied twice with a k-mean segmentation, first to remove the glare and darkest features and a second time to separate the image. The top bins of both segmentation correspond to the aluminium edges of the airbridge. To filter any noise the result is closed and opened and finally the airbridge nearest to the center will be cut out from the original image for validation.

Cross section, sweep and path validation are three of the four methods a bridge is validated. These three harsh basic validation methods are used as a first line to filter the airbridges for their quality.

The middle cross section must follow a pattern according to obtained centers of the bins of the second segmentation. This gives a quality value $C \in \{0,1,2\}$.

Moreover, the cross section validation will be applied over the original image for all airbriges vissible. This results in the quality as a neighboring airbridge $N \in [0,2]$ with a weight depending on the average distance from their neighbors $N_w \in [0,1]$ and encounters $N_k \in \mathbb{Z}_{\geq 0}$.

The entire width of the bridge should follow a pattern similar to the middle cross section and results in a quality factor $S \in \{0,1,2\}$. The last validation before template matching will be searching for stray aluminium at near the edge of the airbridge $P \in \{0,1\}$.

For each of these methods the highest value is a pass and the lowest value a fail.

Template matching is applied to all airbridges. The template is made out of the average of all airbridge that passes all the harsh basic validations for a given device. All the airbridges will be matched to the perfect airbridge resulting in their cross correlation R as described in chapter 2.3. For semi-self-learning a perfect bridge with a weight can be given, guiding the template matching to a pre-set design.

Weights will of the template matching will be adjusted to the other validation methods. This adjustment will be capped at -1 resulting in a confidence of correctness of the bridge $R' = [-1,1]$;

$$R'(R,C',S,P) = R + (1-R)\frac{C'+S+P-4}{6} \tag{5.1}$$

with;

$$C' = \begin{cases} C & if \quad N_k < 2 \\ C(1-N_w)+N\cdot N_w & if \quad C=2 \quad \& \quad N_k > 5 \\ \frac{C(1-N_w)+N\cdot N_w\cdot N_k}{N_k+1} & else \end{cases}$$

Any airbridge will be labeled as bad if its confidence of correctness is below 0.50 and doubtful if below 0.75. The second value can be adjusted by the end user if they are more or less confident in the airbridges on the device.

Validation can be achieved by manually inspecting the images acquired. Both their quality and how they are labeled by `pyclq_ab` needs to be taken into account. No broken bridge should pass and, preferably, the least amount of unbroken airbridges should be labeled as bad or doubtful.

During the first measurement a drift was observed, this could either be human error by locating the markers or caused by the xy stage. To rule out the xy stage two stress test had been done.

First, to rule out fatigue, four airbridges at the corners will repeatedly be measured for 1028 rounds. Second, to rule out deviation due to small movements, four airbridges in close proximity on a straight line will repeatedly be measured for 124 rounds. If any of the stress tests see an change in drift over the measurements it will be the result of the xy-stage.

For the stress test we are interested in locating the airbridges accurately and not validating the airbridges, therefore the gain was set to 300, making it more accurate to locate the bright edge of the airbridge, but harder to validate its correctness.

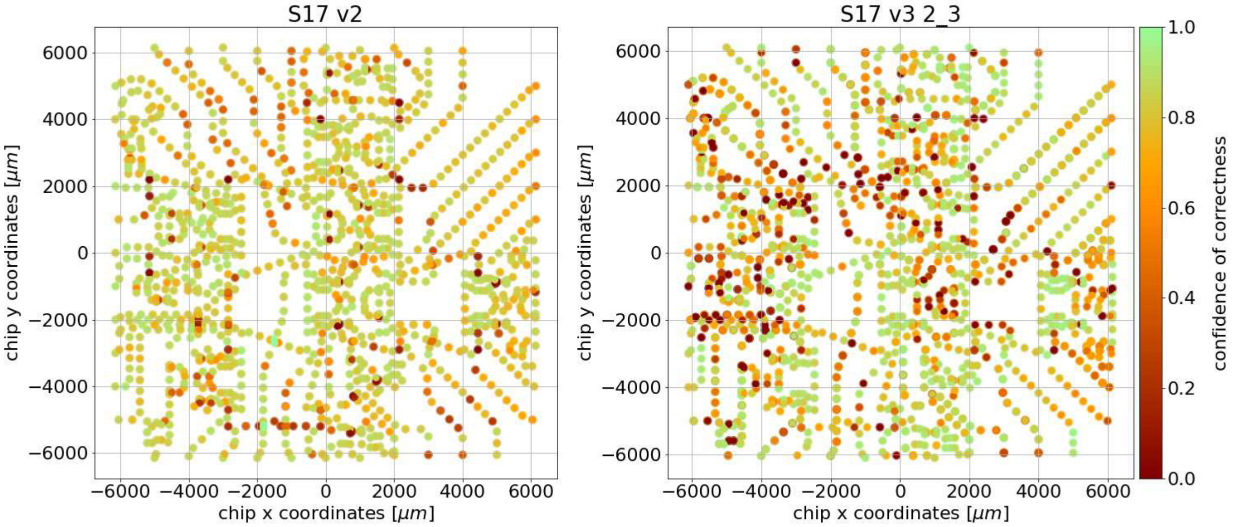


Figure 5.2: The confidence of correctness plotted on the locations of the aibridges corresponding to the design on the S17 v2 and S17 v3 2_3 device, both devices share the same colormap, x- and y-axis. The colormap is capped at 0 as everything below 0 will have a similar confidence as 0.

5.3 Results

In table 5.1 the result of bridges marked by pyclq_ab are shown. No broken bridge passed pyclq_ab and the majority of the defect airbridges will be marked as bad. A random assortment of airbridges marked bad from the S17 v3 2_3 is shown in figure 5.4. As can be seen some of them are blurred due to movement of the camera, some of them collapsed or had a dent in their reflection.

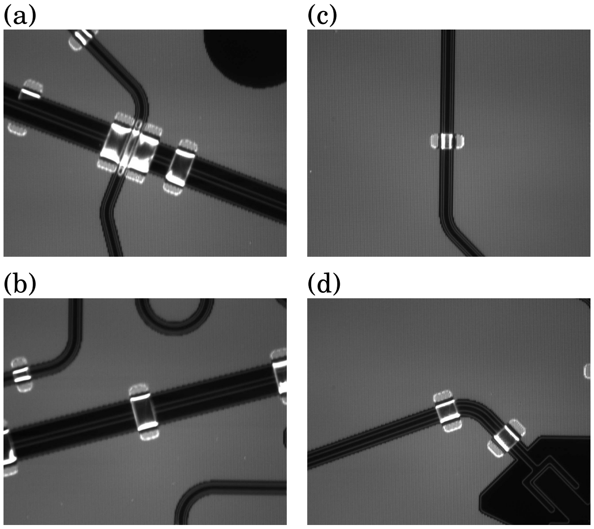


Figure 5.3: The four types of airbridges found, whereas; (a) is a crossover, (b) an airbridge of 40×116 μm, (c) the most common airbridge of 30×70μm and (d) an airbridge of 40×84 μm

The confidence of correctness capped at 0 $R' = [0,1]$ is plotted in figure 5.2 for every location on the device. All the different types of airbridges validated by pyclq_ab are shown in figure 5.3. The first few rows of the .csv file generated by pyclq_ab can be found in the appendix together with the images of these airbridges.

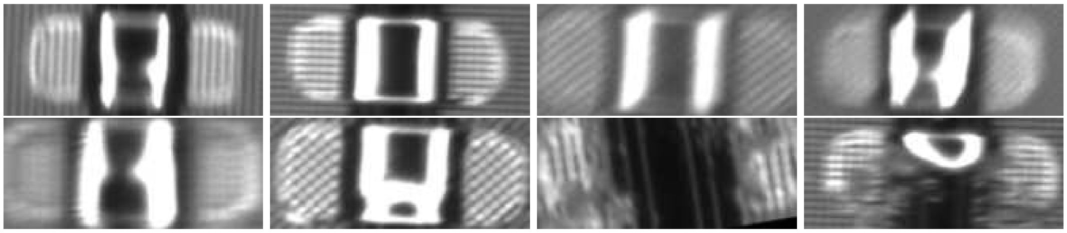


Figure 5.4: a random assortment of 30×70µm airbridges cut from the image by `pyclq_ab` marked as bad. The first two airbridges have failed to locate the edges well enough due to reflections of other aluminium features near the bridge, the third is due to movement of the camera, fourth and fifth are either collapsed or the camera moved and the last three collapsed.

Due to the lack of crossovers of the same size (< 15) the airbridge validation for the transmission line crossovers have been left out. The sample size to generate the 'perfect' feature have been to small and therefore the template matching will be non-sensible as they will either match to themselves, noting or only a single crossover. Moreover, due to the close proximity of the bridges at the crossover, the bridges at the crossover reflect of each other. This results in overexposure at crossover and made it even difficult to inspect manually with the current setup, figure 5.3 c.

Halve of the false positives are caused by the movement of the camera. The movement was likely caused by a combination of the architecture of Labvieuw and its communication to external drivers like the xy-stage and camera. All communication is pushed trough a single channel in the computer, causing desynchronisation in the delays what matches the camera movement and taking an image.

Table 5.1: For two different chip designs the amount of airbridges are shown, followed by the amount of airbridges marked as bad or doubtful by `pyclq_ab`. These bridges have been manually inspected and the actual bad arbridges have been counted in every marked category.

		marked as bad			marked as doubtful			marked as good		
device	bridges	marked	broken	unbroken	marked	broken	unbroken	marked	broken	unbroken
S17 v2	1197	73	51/73*	22/73	113	1/113	112/113	1084	0/1084	1084/1084
S17 v3 2_3	1549	312	267/312	45/312	278	159/278	119/278	1004	0/1004	1004/1004

* of which 20 are due to a difference in size between the CSV file and design

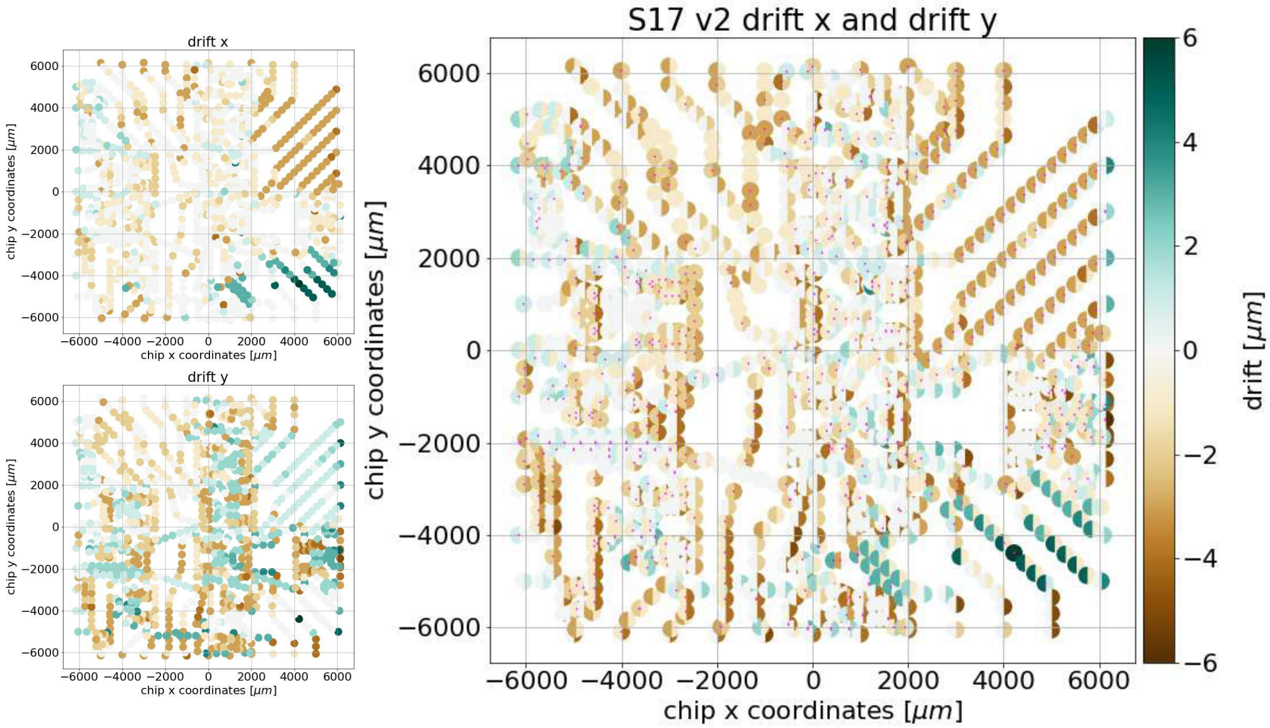


Figure 5.5: The capture of drift at every location, first plot is the the drift in the x direction, second is the plot of the drift in y direction and the last is drift in both x and y direction. Where the left half of the orb is drift in x direction and right half of the orb is the drift in y direction. All variables are in micrometers.

Drift and stress test

In figure 5.5 the locations of the airbriges in S17 v2 are shown with their drift. As can be seen this drift is quite significant resulting in the stress tests. Moreover, there seem to have been some design changes in the device, between the obtained .csv files and validating the airbridges, resulting in a perceived drift for some of the airbridges.

For the first stress tests there was no drift found, figure 5.6. This means the xy stage doesn't fatigue over time. Similarly, for the second stress test there was no change in drift found between the first and last point when moving small movements figure 5.7. This means the drift is caused by placing the markers in the automated room temperature resistance measurement unit whereas a deviation of 1 pixel already results in $1\mu m$ deviation depended on the location.

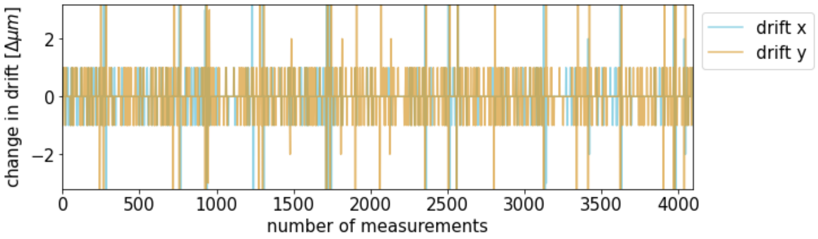


Figure 5.6: The change of the drift is shown for 4096 measurements of four airbridges near the corners taken repeatedly of a S17 V3_2. 1 μm drift deviations are caused by image processing due to opening and closing the image. Most of the outliers out of the 1μm drift are caused by unsharp images due to movement of the camera while taking the image.

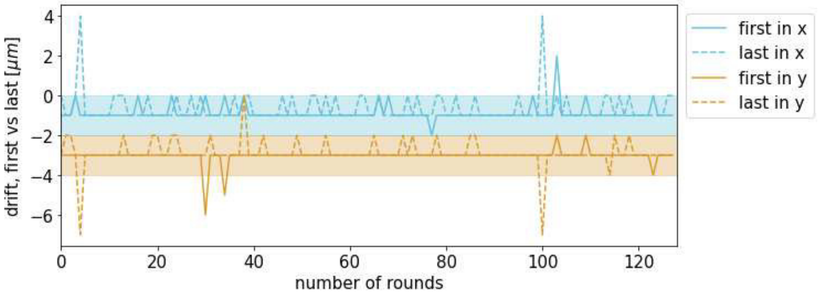


Figure 5.7: The drift of the first and last airbridge is shown for 124 rounds of short steps on a S17 V3_2 device. Most of the outliers out of the 1μm drift are caused by unsharp images due to movement of the camera while taking the image. The band shows the drift caused by image processing due to opening and closing the image.

6. Conclusion

The written software suite, pyclq, is able to automate parts of the optical inspection of a superconducting quantum processor during fabrication. This reduces the workload and is a first step in helping the fabrication process cope with the increased demands of device scaling using automation by image processing.

With three simple scripts the components of the chip can be inspected using convolution filters, segmentation, thresholding and template matching; automated by a k-means algorithm. For the measurements and yield it is important to keep in mind the geometry of 3D features, charging effects and the reflections dependent on the material.

pyclq_base helps tracing any defect in the base layer helping to quantify yield metrics. There have been no template matching results for pyclq_base yet, but the template matching method is expected to be easily applied for the base layer as well as it has been already applied for pyclq_ab. Whereas pyclq_base is not designed to detect airbridges due to their reflective 3D nature pyclq_ab is.

pyclq_jj is able to obtain the overlap area of the Manhattan-style Josephson junctions with two different methods. A k-means segmentation had the most successful measurements, however it overestimates the area even after reducing the halo from the SEM images. Hence thresholding method was the most accurate when compared to manual measurements with ImageJ. It should be noted that a k-means segmentation shines for low contrast images and smaller junctions as the halo effect is less prominent. These are the images the thresholding have difficulty with. By automating the extraction of overlap areas, errors introduced due to human bias is avoided and also reduces several hours of repetitive tasks.

pyclq_ab had some false positives, but no false negatives in detecting broken airbridges. This will ensure no mismanufactured airbridge will pass through the optical inspection. The only downside is a minimum requirement of airbridges > 15 what was one of the reasons pyclq_ab had a hard time validating the crossovers. The other reason is reflections from neighboring bridges due to close proximity, what makes it even difficult to validate those airbridges with the human eye under the APS-LASIQ automated probe station. Another possibility to improve validation of crossovers is to switch to a 20x objective lens for higher resolution.

Bibliography

[1] Rachel Courtland. Google aims for quantum computing supremacy [news]. *IEEE Spectrum*, 54(6):9–10, 2017.

[2] Matthias Steffen, David P DiVincenzo, Jerry M Chow, Thomas N Theis, and Mark B Ketchen. Quantum computing: An ibm perspective. *IBM Journal of Research and Development*, 55(5):13–1, 2011.

[3] Morten Kjaergaard, Mollie E. Schwartz, Jochen Braumüller, Philip Krantz, Joel I.-Jan Wang, Simon Gustavsson, and William D. Oliver. Superconducting Qubits: Current State of Play, May 2019. arXiv: 1905.13641.

[4] Diego Riste, Stefano Poletto, M-Z Huang, Alessandro Bruno, Visa Vesterinen, O-P Saira, and Leonardo DiCarlo. Detecting bit-flip errors in a logical qubit using stabilizer measurements. *Nature communications*, 6(1):1–6, 2015.

[5] JF Marques, BM Varbanov, MS Moreira, Hany Ali, Nandini Muthusubramanian, Christos Zachariadis, Francesco Battistel, Marc Beekman, Nadia Haider, Wouter Vlothuizen, et al. Logical-qubit operations in an error-detecting surface code. *arXiv preprint arXiv:2102.13071*, 2021.

[6] H.Martijn Veen. `pyclq`, September 2021. Accessed September 30, 2021, at: `https://github.com/DiCarloLab-Delft/PyCLQ`.

[7] Puneet Gupta and Evanthia Papadopoulou. Yield analysis and optimization. In *Handbook of Algorithms for Physical Design Automation*, pages 771–790. Auerbach Publications, 2008.

[8] Daniel N Maynard and Jason D Hibbeler. Measurement and reduction of critical area using voronoi diagrams. In *IEEE/SEMI Conference and Workshop on Advanced Semiconductor Manufacturing 2005.*, pages 243–249. IEEE, 2005.

[9] Antonio B Mei, Ivan Milosavljevic, Amanda L Simpson, Valerie A Smetanka, Colin P Feeney, Shay M Seguin, Sieu D Ha, Wonill Ha, and Matthew D Reed. Optimization

of quantum-dot qubit fabrication via machine learning. *Applied Physics Letters*, 118(20):204001, 2021.

[10] D.M. Pozar. *Microwave Engineering*. John Wiley and Sons, 2011.

[11] M. Göppl, A. Fragner, M. Baur, R. Bianchetti, S. Filipp, J. M. Fink, P. J. Leek, G. Puebla, L. Steffen, and A. Wallraff. Coplanar waveguide resonators for circuit quantum electrodynamics. *Journal of Applied Physics*, 104(11):113904, December 2008.

[12] Jurgen Dijkema. *Improvement of superconducting resonators for long-range coupling of spin qubits*. Masters thesis, TU Delft, May 2020.

[13] Martin Göppl, A Fragner, M Baur, R Bianchetti, Stefan Filipp, Johannes M Fink, Peter J Leek, G Puebla, L Steffen, and Andreas Wallraff. Coplanar waveguide resonators for circuit quantum electrodynamics. *Journal of Applied Physics*, 104(11):113904, 2008.

[14] Quantwiki, October 2015. Accessed September 30, 2021, at: `https://www.quantiki.org/wiki/qubit`.

[15] Anton Frisk Kockum and Franco Nori. *Quantum Bits with Josephson Junctions*, pages 703–741. Springer International Publishing, Cham, 2019.

[16] Jens Koch, M Yu Terri, Jay Gambetta, Andrew A Houck, David I Schuster, Johannes Majer, Alexandre Blais, Michel H Devoret, Steven M Girvin, and Robert J Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Physical Review A*, 76(4):042319, 2007.

[17] Pim Duivestein. Wafer scale fabrication of josephson junctions. Masters thesis, TU Delft, June 2020.

[18] Pranav S. Mundada, Gengyan Zhang, Thomas Hazard, and Andrew A. Houck. Suppression of Qubit Crosstalk in a Tunable Coupling Superconducting Circuit. *arXiv:1810.04182 [quant-ph]*, October 2018. arXiv: 1810.04182.

[19] NumPy, documentation generated with `https://www.sphinx-doc.org/`, June 2021. Accessed September 30, 2021, at: `https://numpy.org/doc/stable/index.html`.

[20] OpenCV, documentation generated with `https://www.Doxigen.nl`, December 2020. Accessed September 30, 2021, at: `https://docs.opencv.org/4.5.1/`.

[21] Sreenivas Bhattiprolu. Python for microscopists, August 2021. Accessed September 30, 2021, at: `https://github.com/bnsreenu/python_for_microscopists`.

[22] Arvind Ramanathan, Laura L Pullum, Faraz Hussain, Dwaipayan Chakrabarty, and Sumit Kumar Jha. Integrating symbolic and statistical methods for testing intelligent systems: Applications to machine learning and computer vision. In *2016 Design,*

Automation & Test in Europe Conference & Exhibition (DATE), pages 786–791. IEEE, 2016.

[23] Fuqiang Zhong, Songping He, and Jiangwei Yi. A fast template matching method for led chip localization. In *MATEC web of conferences*, volume 34, page 04002. EDP Sciences, 2015.

[24] Joris Guérin, Olivier Gibaru, Stéphane Thiery, and Eric Nyiri. Clustering for different scales of measurement-the gap-ratio weighted k-means algorithm. *arXiv preprint arXiv:1703.07625*, 2017.

[25] SciPy 2016 Scikit-learn Tutorial, July 2016. Accessed September 30, 2021, at: `https://github.com/amueller/scipy-2016-sklearn`.

[26] Yasser H Anis, James K Mills, and William L Cleghorn. Visual measurement of mems microassembly forces using template matching. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 275–280. IEEE, 2006.

[27] W James MacLean and John K Tsotsos. Fast pattern recognition using normalized grey-scale correlation in a pyramid image representation. *Machine Vision and Applications*, 19(3):163–179, 2008.

[28] Sean van der Meer and Matvey Finkel. automated probe and laser annealing station `LASIQ`, September 2021. Accessed September 30, 2021, at: `https://github.com/DiCarloLab-Delft/LASIQ`.

[29] M. A. Nielsen and I. L. Chang. *Quantum Computation and Quantum Information, 10th Anniversary Ed.* Cambridge University, 2010.

[30] G. J. Dolan. Offset masks for lift-off photoprocessing. *Applied Physics Letters*, 31(5):337–339, September 1977.

[31] A Potts, PR Routley, Gregory J Parker, JJ Baumberg, and PAJ De Groot. Novel fabrication methods for submicrometer josephson junction qubits. *Journal of Materials Science: Materials in Electronics*, 12(4):289–293, 2001.

[32] Alexander Bilmes, Alexander K Neumann, Serhii Volosheniuk, Alexey V Ustinov, and Jürgen Lisenfeld. In-situ bandaged josephson junctions for superconducting quantum processors. *arXiv preprint arXiv:2101.01453*, 2021.

[33] Sergey K. Tolpygo, Vladimir Bolkhovsky, T. J. Weir, Alex Wynn, D. E. Oates, L. M. Johnson, and M. A. Gouker. Advanced fabrication processes for superconducting very large-scale integrated circuits. *IEEE Transactions on Applied Superconductivity*, 26(3):1–10, 2016.

[34] A Osman, J Simon, A Bengtsson, S Kosen, P Krantz, D P. Lozano, M Scigliuzzo, P Delsing, Jonas Bylander, and A Fadavi Roudsari. Simplified josephson-junction fabrication process

for reproducibly high-performance superconducting qubits. *Applied Physics Letters*, 118(6):064002, 2021.

[35] Nandini Muthusubramanian, Wilhelmus Duivestein, Chris Zachariadis, Matvey Finkel, Alessandro Bruno, and Leonardo DiCarlo. Fabrication parameters for frequency targeting in scalable superconducting quantum processors. *Bulletin of the American Physical Society*, 2021.

.1 Pseudo codes

Bellow are the flow diagrams representing the code. Each component shows what it does to the image and how it relates to the method obtaining the data.

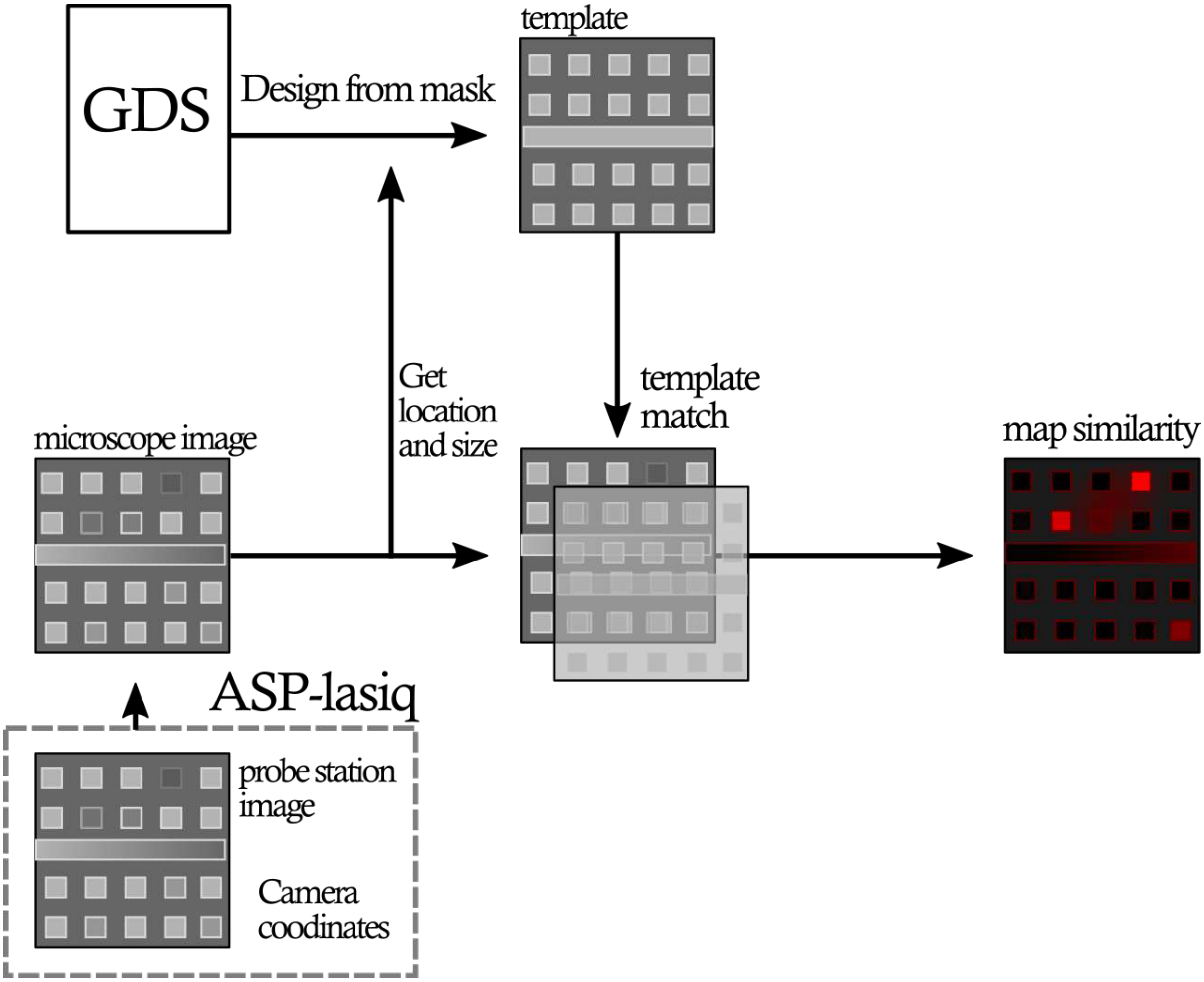


Figure 1: A perfect base layer is created from the file where an image is taken. The microscope image is obtained from ASP-Lasiq these two images are template matched for their individual locations resulting in a map with similarities between the two images. The more they vary the brighter the feature becomes what means that area needs inspection.

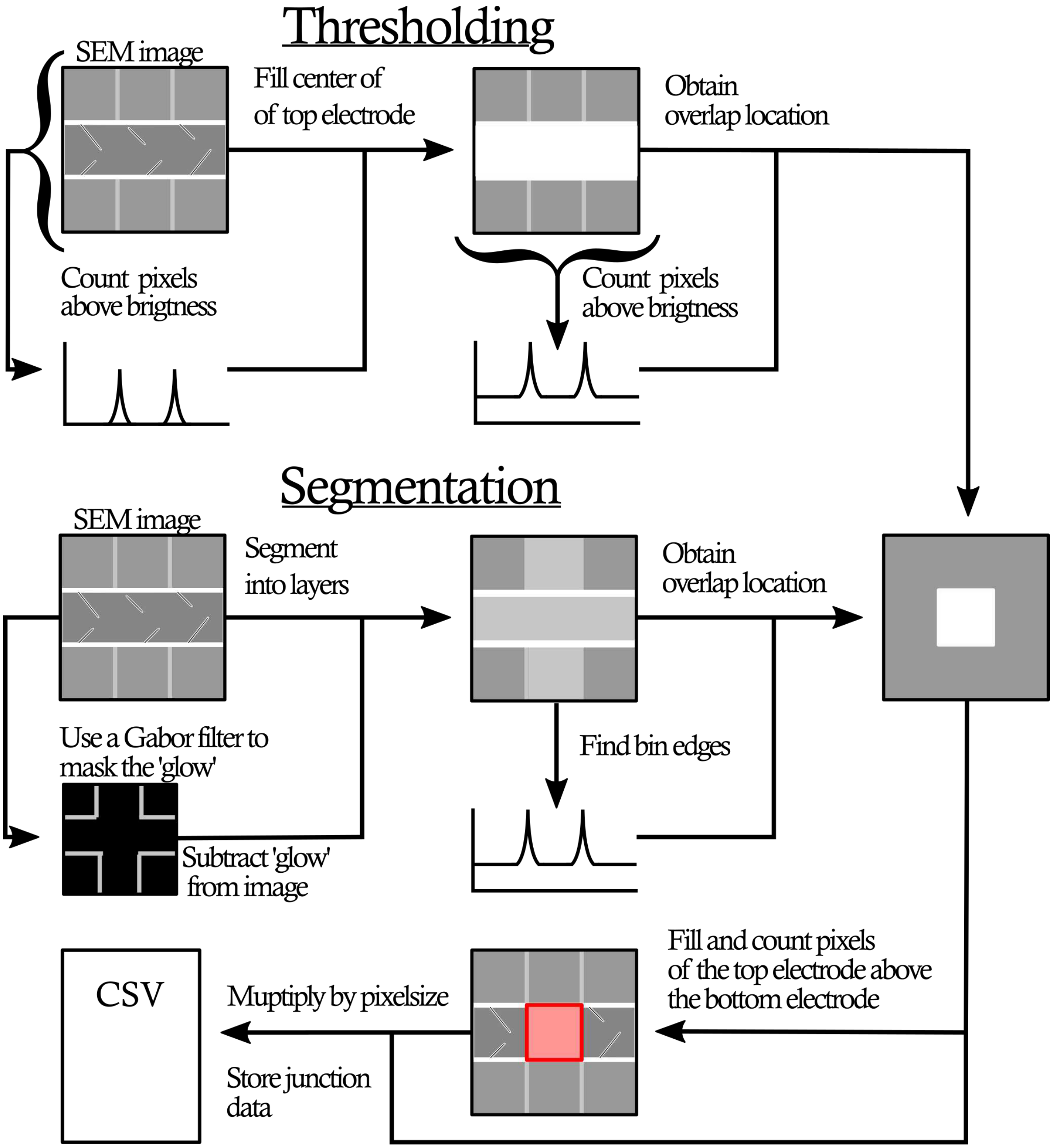


Figure 2: A visual representation of `pyclq_jj` at the top is the thresholding method, applying an sweep over the x- and y-axis to obtain the edges from the highes pixel value. This is repeated for multiple thresholds to improve accuracy. The segmentation method removes the 'glow' effect of the SEM first before segmenting into different layer of which the top layer(s) will be the edge from the top electrode. The bottom vertical electrode is obtained in a similar manner as the thresholding method. Both methods are filling the area of the top electrode that is overlapping the bottom electrode and storing it in a CSV file for future reference.

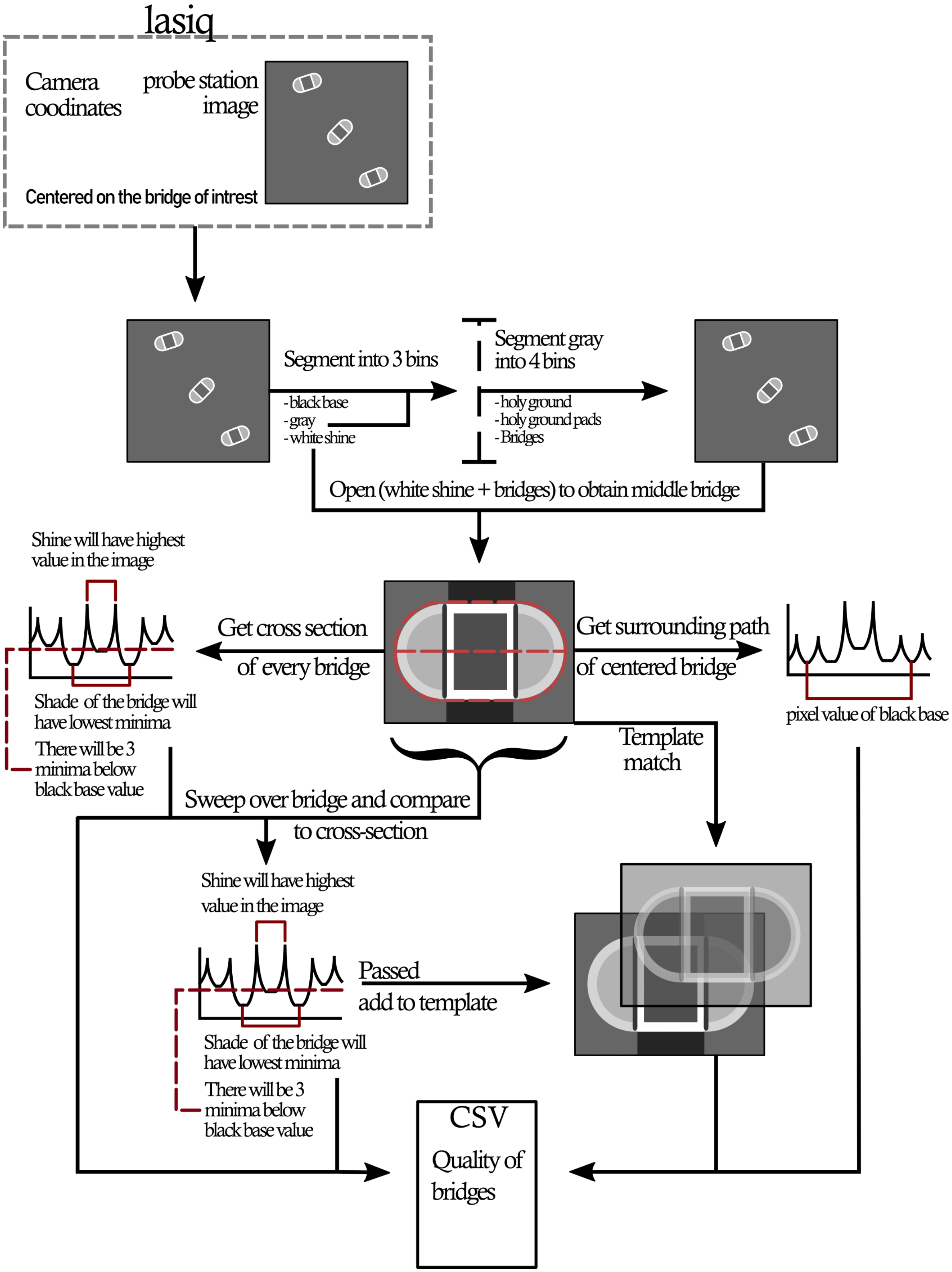

lasiq
Camera coodinates
probe station image
Centered on the bridge of intrest
Segment into 3 bins
- black base
- gray
- white shine
Segment gray into 4 bins
- holy ground
- holy ground pads
- Bridges
Open (white shine + bridges) to obtain middle bridge
Shine will have highest value in the image
Get cross section of every bridge
Shade of the bridge will have lowest minima
There will be 3 minima below black base value
Get surrounding path of centered bridge
pixel value of black base
Template match
Sweep over bridge and compare to cross-section
Shine will have highest value in the image
Passed add to template
Shade of the bridge will have lowest minima
There will be 3 minima below black base value
CSV
Quality of bridges

Figure 3: A visual representation of accusation and validation of bridges, fits the acquired image with ASP-lasiq is filtered and segmented in the different components of the chip. by opening, chapter **??**, *the white shine and bridge grey colors the bridges can be obtained (the top two bins). Next a surrounding pad and cross section validation made with expected values form the segmentation. From the the cross section validation is compared to a full scan over the width of the bridge to gain a third quality of the bridge, if passed with a high enough value it will be combined with other bridges of the same shape for template matching of all the airbridges for the last quality check.*

.2 Pyclq_AB

Below the first ten rows of airbridge validation by `pyclq_ab` of S17 v2 3_2.

Table 1: Below the first ten airbridges validated by `pyclq_ab` all widths and lengths are in μm and angles in degrees

Design						Found by `pyclq_ab`												
x	y	angle	transmission_line	width	length	drift_x	drift_y	width	length	C	S	P	R	N	N_w	N_k	R'
5944.437	1873.431	-135	Microwave drive line D1	30	70	-3	0	31	63	2	0	1	0.809584	0	0	0	0.777848
4376.569	305.5635	-135	Microwave drive line D1	30	70	-3	0	31	66	2	2	1	0.848954	0	0	0	0.874128
5720.455	1649.45	-135	Microwave drive line D1	30	70	-3	0	30	63	2	0	1	0.814407	0	0	0	0.783474
5496.474	1425.469	-135	Microwave drive line D1	30	70	-3	0	31	66	2	2	1	0.859434	0	0	0	0.882862
5272.493	1201.488	-135	Microwave drive line D1	30	70	-3	0	31	66	2	2	1	0.815088	0	0	0	0.845907
5048.512	977.5069	-135	Microwave drive line D1	30	70	-4	0	31	66	2	2	1	0.799738	0	0	0	0.833115
4824.531	753.5258	-135	Microwave drive line D1	30	70	-3	0	31	65	2	0	1	0.895164	0	0	0	0.877691
4600.55	529.5446	-135	Microwave drive line D1	30	70	-3	0	30	65	2	0	1	0.859872	0	0	0	0.836518
4250	42	-90	Microwave drive line D1	30	70	0	-6	20	26	0	0	1	-0.04892	0	0	0	-0.57339
5000	-5950	90	Microwave drive line D2	30	70	0	-3	29	64	2	2	1	0.899661	0	0	0	0.916384

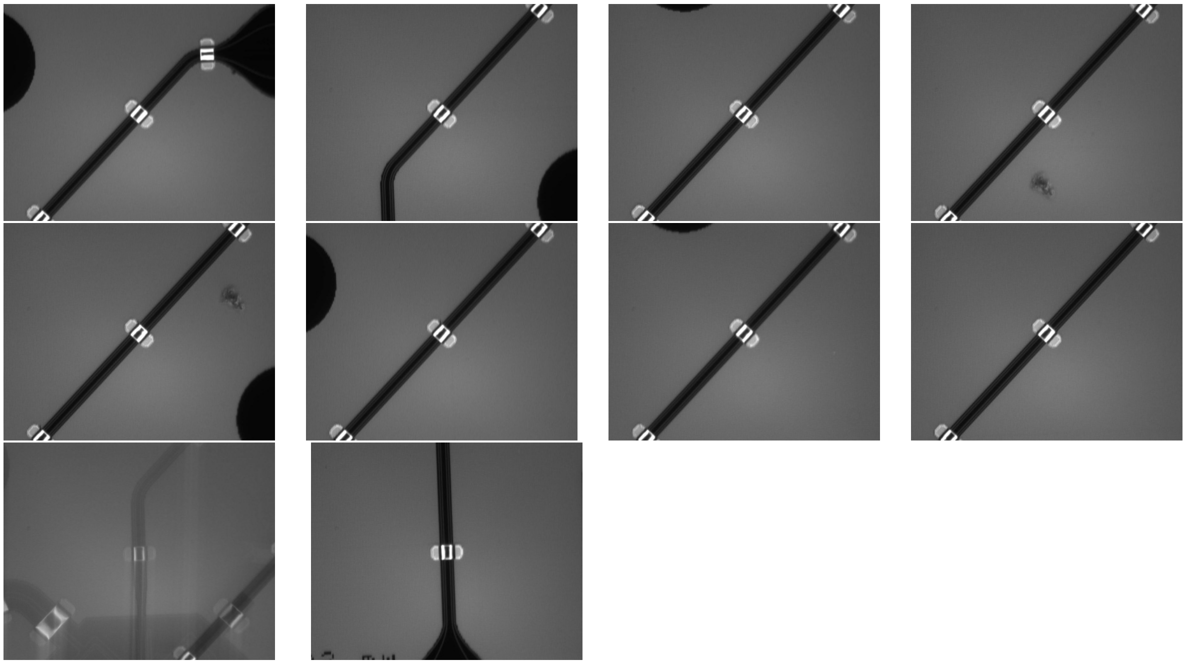


Figure 4: From left to right, top to bottom the airbridges validated in the table above, table 1. When taking the 9th, second last airbridge in figure and table there seem to be movement due to the communication between the computer and its external components.

.3 Pyclq_jj

.3.1 Plots

plots of conductance against area

Below all the fits of the conductance against the overlapping area measured with ImageJ and `Pyclq_jj` tresholding method are shown with the coefficient of determination, R^2. The numerical values and standaard deviation can be found in table **??**. This is followed up by a histogram of the conductance over the area for every measurements. The widths of all bins in the histograms have a bin size of $100\mu S/\mu m^2$

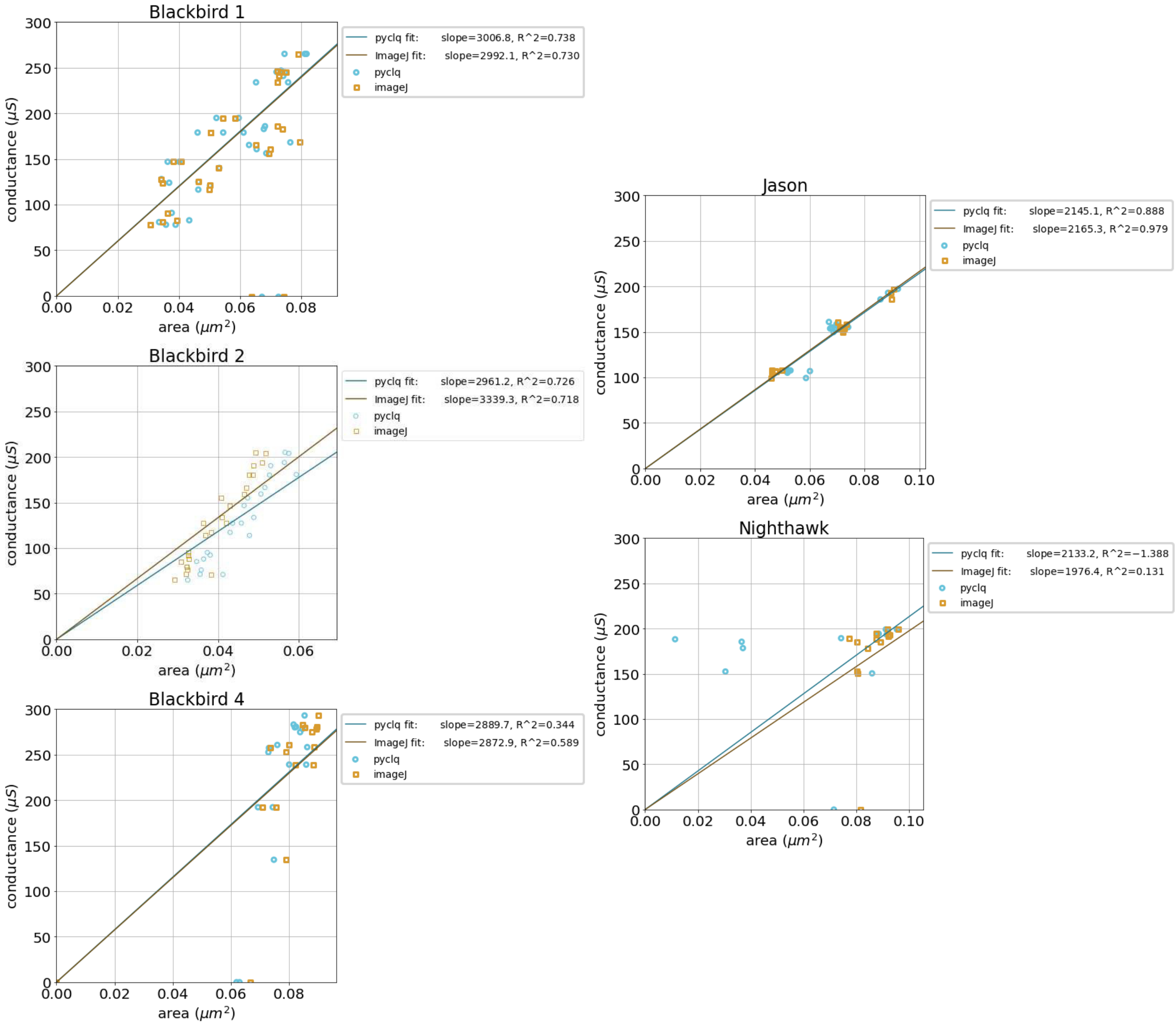


Figure 5: All the fits of the conductance against the overlapping area measured with ImageJ and `Pyclq_jj` *tresholding method are shown with the coefficient of determination, R^2.*

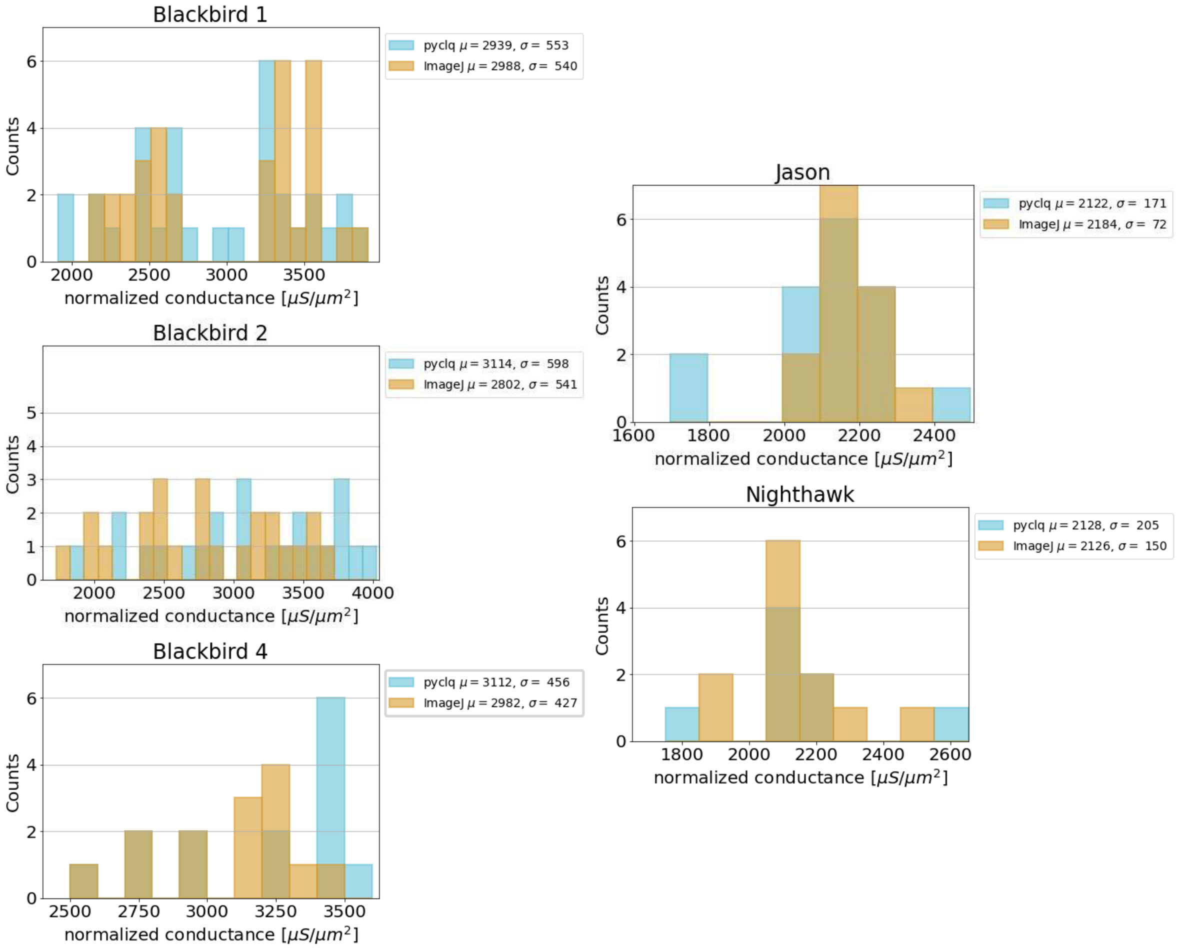
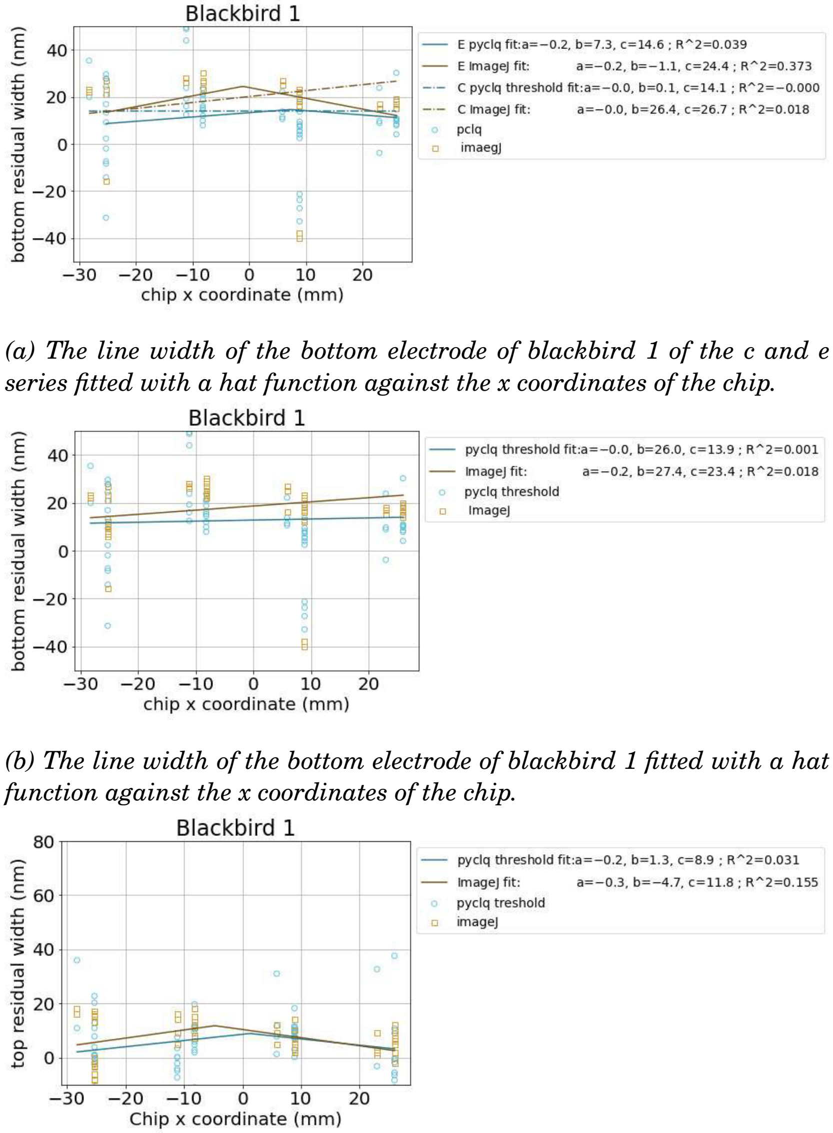


Figure 6: All points of conductance against the overlapping area measured with ImageJ and `Pyclq_jj` *tresholding method are shown in a histogram.*

Plots of widths

Below are the plots of the widths of the top horizontal and bottom vertical electrodes plotted against a hat function. For the locations of the Jason dataset the locations are unsuited to plot against x or y.

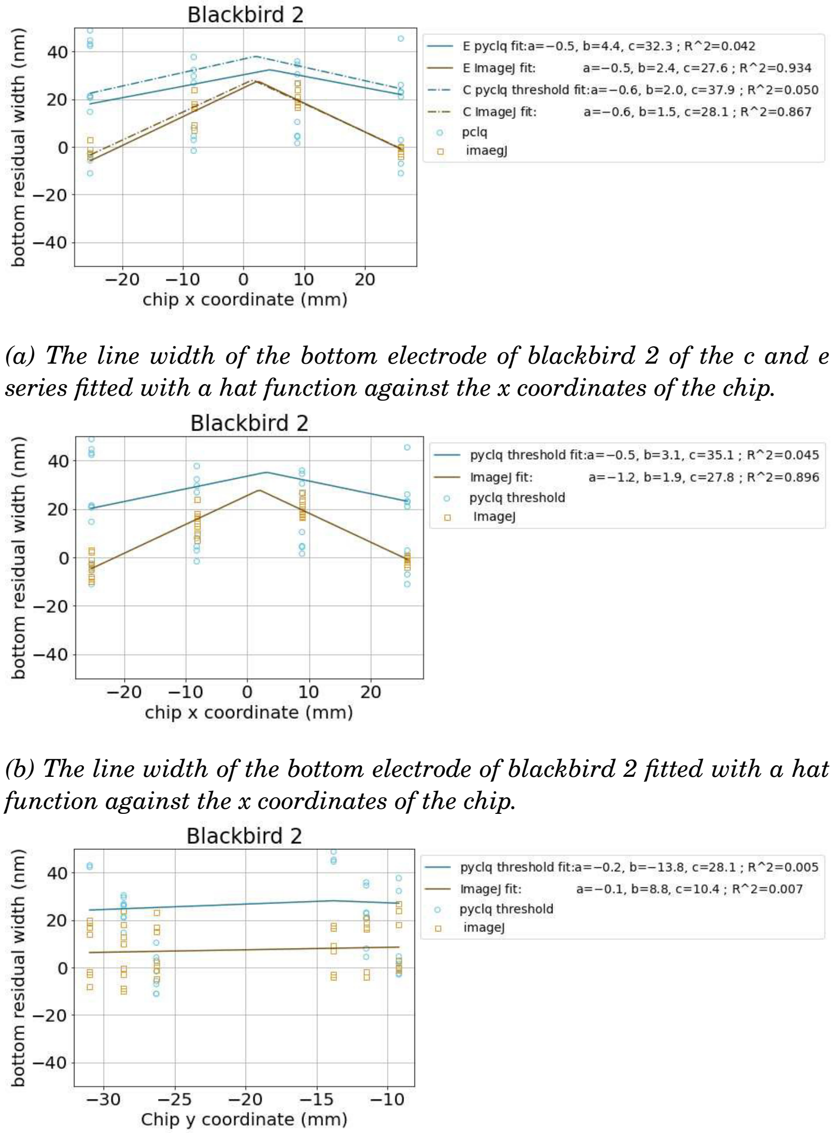


(a) The line width of the bottom electrode of blackbird 1 of the c and e series fitted with a hat function against the x coordinates of the chip.

(b) The line width of the bottom electrode of blackbird 1 fitted with a hat function against the x coordinates of the chip.

(c) The line width of the top electrode of blackbird 1 fitted with a hat function against the x coordinates of the chip.

Figure 7: The fits of the widths of the bottom horizontal electrode and top vertical electrode in blackbird 1 are shown above, in (a) the sets are fitted for the c and e series separately for the bottom horizontal electrode. The exact values are found in table .3.2.

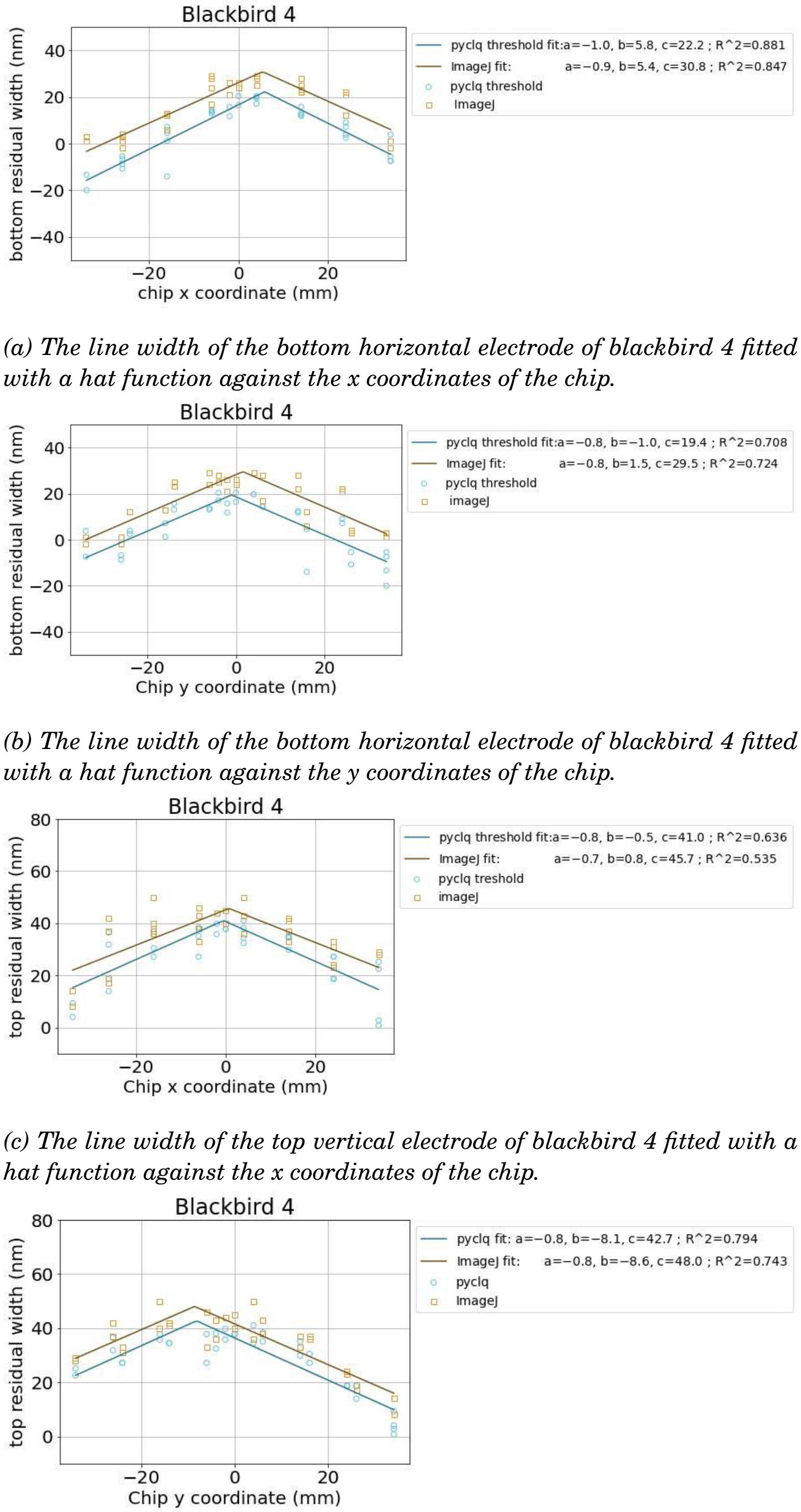


(a) The line width of the bottom electrode of blackbird 2 of the c and e series fitted with a hat function against the x coordinates of the chip.

(b) The line width of the bottom electrode of blackbird 2 fitted with a hat function against the x coordinates of the chip.

(c) The line width of the top electrode of blackbird 2 fitted with a hat function against the x coordinates of the chip.

Figure 8: The fits of the widths of the bottom horizontal electrode and top vertical electrode in blackbird 2 are shown above, in (a) the sets are fitted for the c and e series separately for the bottom horizontal electrode. The exact values are found in table .3.2.



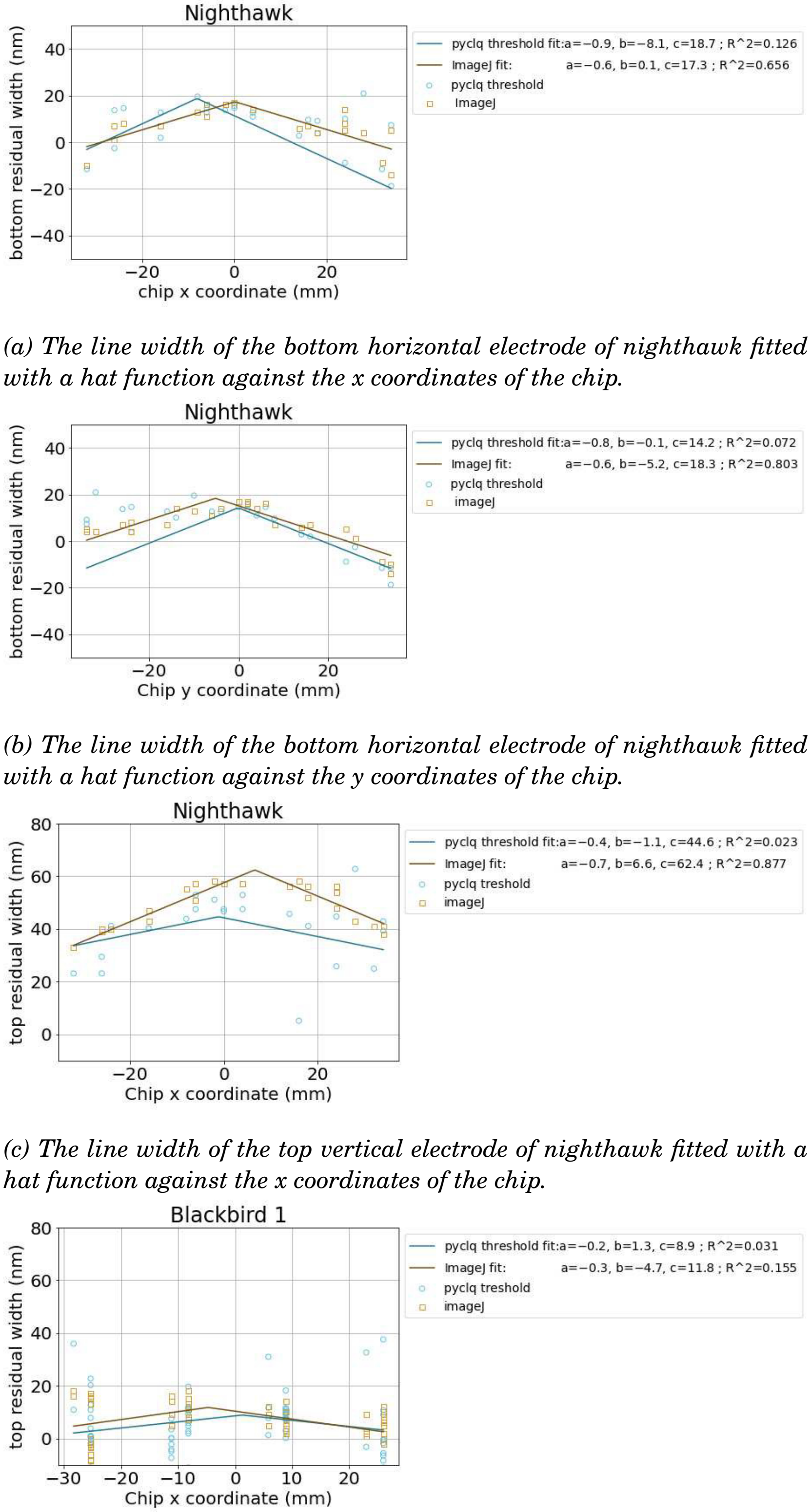
(a) The line width of the bottom horizontal electrode of blackbird 4 fitted with a hat function against the x coordinates of the chip.

(b) The line width of the bottom horizontal electrode of blackbird 4 fitted with a hat function against the y coordinates of the chip.

(c) The line width of the top vertical electrode of blackbird 4 fitted with a hat function against the x coordinates of the chip.

(c) The line width of the top vertical electrode of blackbird 4 fitted with a hat function against the y coordinates of the chip.

Figure 9: The fits of the widths of the bottom horizontal electrode (a,c) and top vertical electrode (b,d) in blackbird 4 are shown above, The exact values are found in table .3.2.



(a) The line width of the bottom horizontal electrode of nighthawk fitted with a hat function against the x coordinates of the chip.

(b) The line width of the bottom horizontal electrode of nighthawk fitted with a hat function against the y coordinates of the chip.

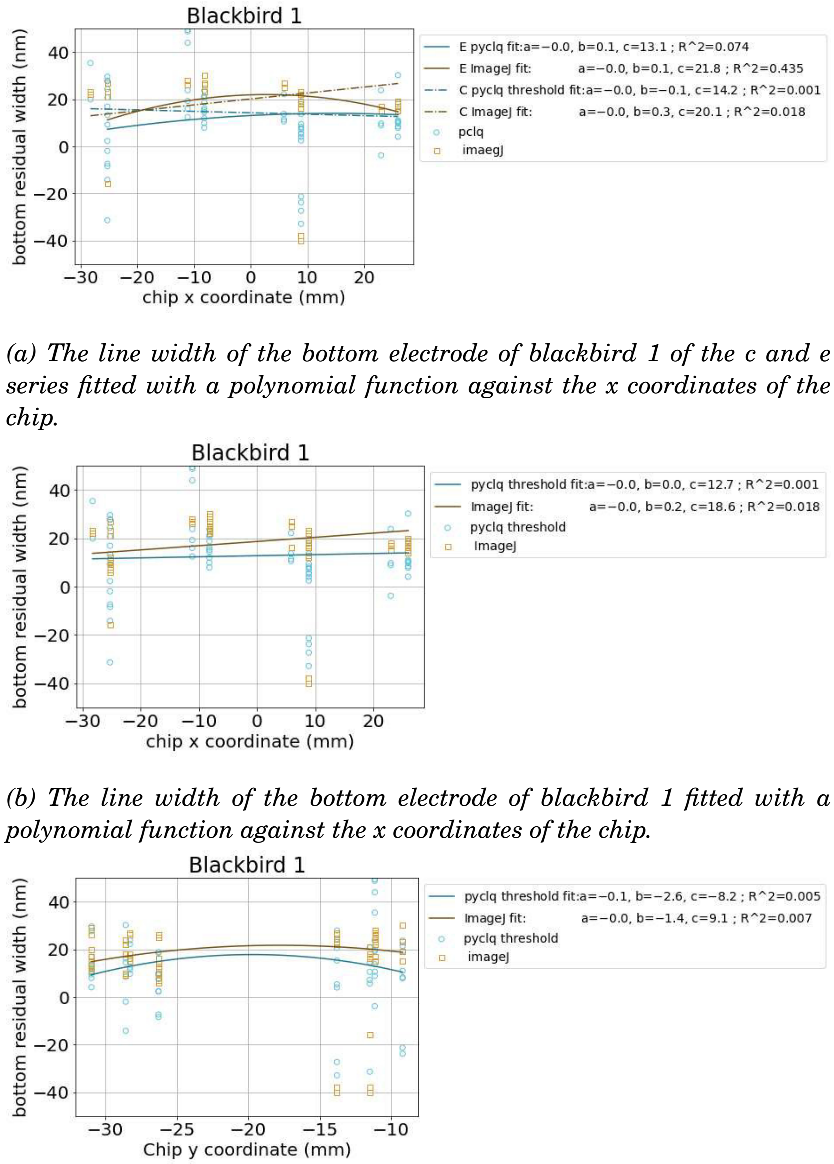
(c) The line width of the top vertical electrode of nighthawk fitted with a hat function against the x coordinates of the chip.

(c) The line width of the top vertical electrode of nighthawk fitted with a hat function against the y coordinates of the chip.

Figure 10: The fits of the widths of the bottom horizontal electrode (a,c) and top vertical electrode (b,d) in blackbird 4 are shown above, The exact values are found in table .3.2.

Plots of widths

Below are the plots of the widths of the top horizontal and bottom vertical electrodes plotted against a polynomial function. For the locations of the Jason dataset the locations are unsuited to plot against x or y.

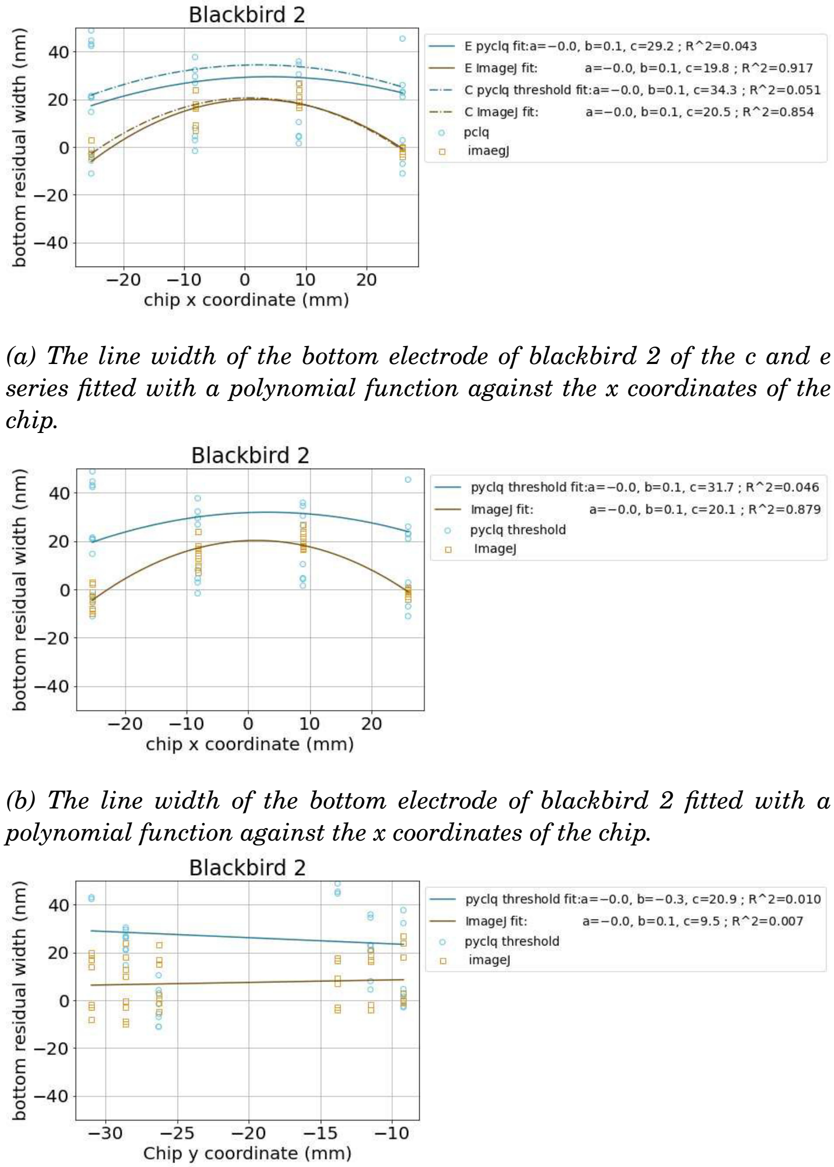


(a) The line width of the bottom electrode of blackbird 1 of the c and e series fitted with a polynomial function against the x coordinates of the chip.

(b) The line width of the bottom electrode of blackbird 1 fitted with a polynomial function against the x coordinates of the chip.

(c) The line width of the top electrode of blackbird 1 fitted with a polynomial function against the x coordinates of the chip.

Figure 11: The fits of the widths of the bottom horizontal electrode and top vertical electrode in blackbird 1 are shown above, in (a) the sets are fitted for the c and e series separately for the bottom horizontal electrode. The exact values are found in table .3.2.

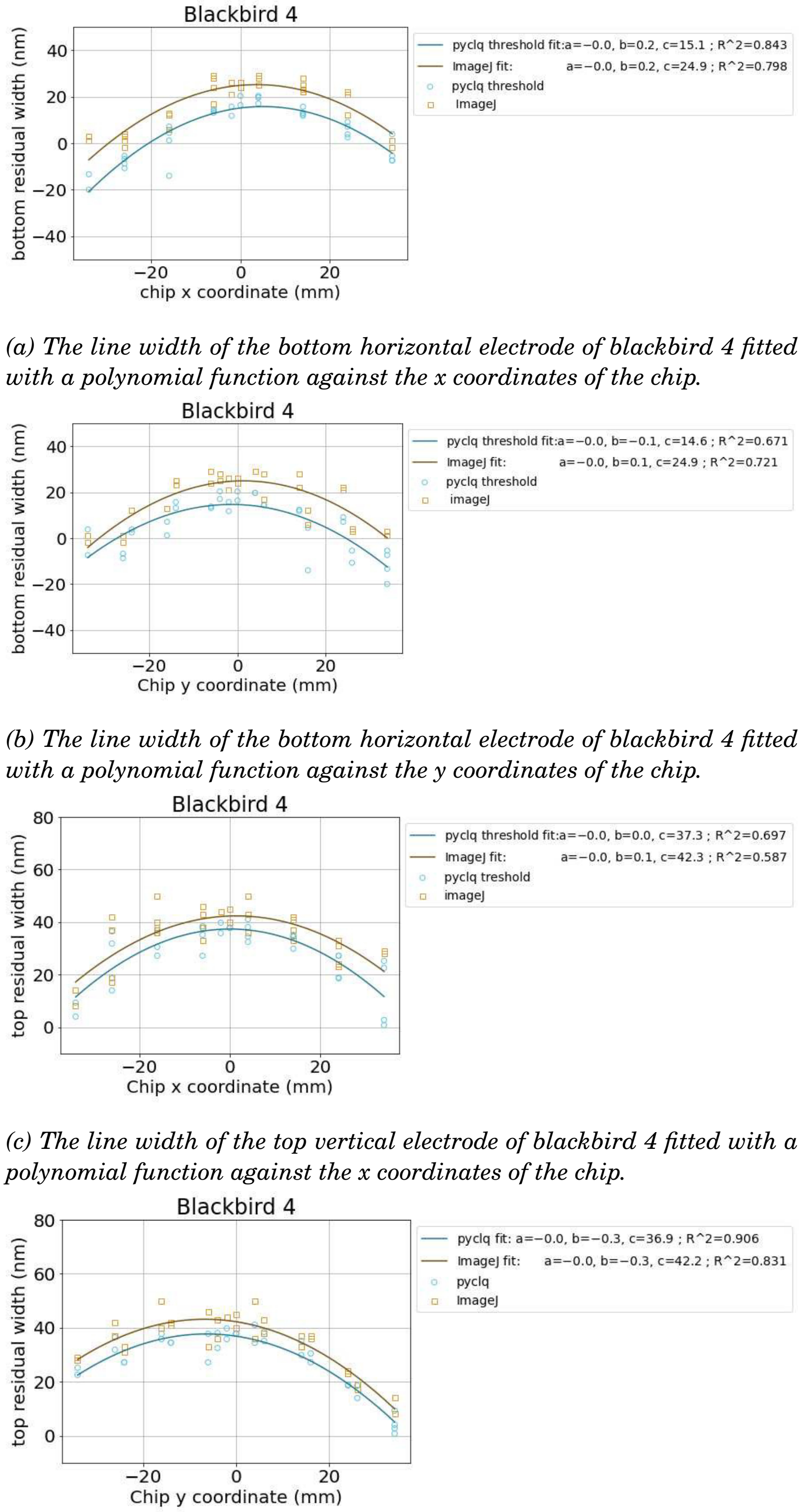


(a) The line width of the bottom electrode of blackbird 2 of the c and e series fitted with a polynomial function against the x coordinates of the chip.

(b) The line width of the bottom electrode of blackbird 2 fitted with a polynomial function against the x coordinates of the chip.

(c) The line width of the top electrode of blackbird 2 fitted with a polynomial function against the x coordinates of the chip.

Figure 12: The fits of the widths of the bottom horizontal electrode and top vertical electrode in blackbird 2 are shown above, in (a) the sets are fitted for the c and e series separately for the bottom horizontal electrode. The exact values are found in table .3.2.



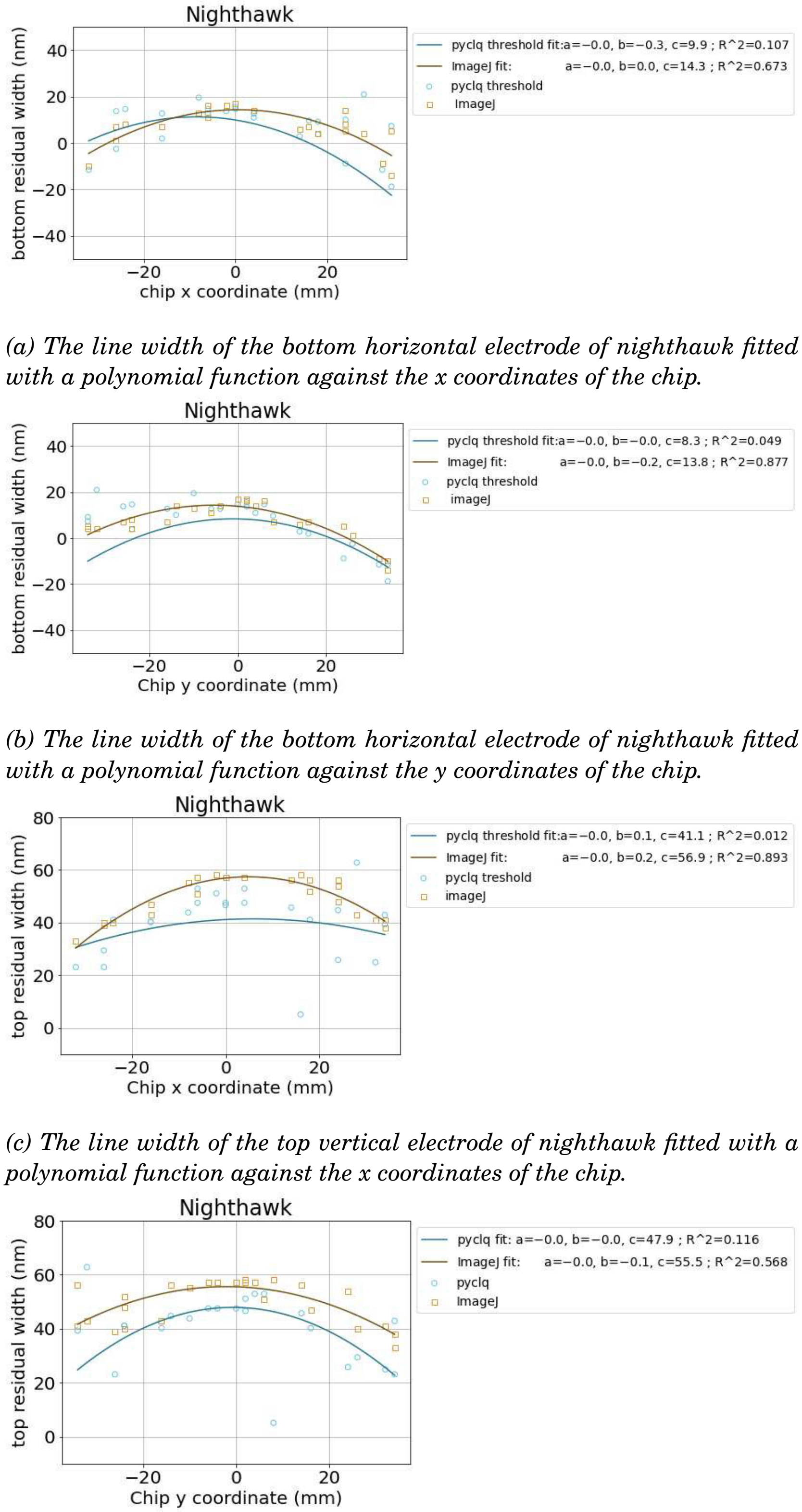
(a) The line width of the bottom horizontal electrode of blackbird 4 fitted with a polynomial function against the x coordinates of the chip.

(b) The line width of the bottom horizontal electrode of blackbird 4 fitted with a polynomial function against the y coordinates of the chip.

(c) The line width of the top vertical electrode of blackbird 4 fitted with a polynomial function against the x coordinates of the chip.

(c) The line width of the top vertical electrode of blackbird 4 fitted with a polynomial function against the y coordinates of the chip.

Figure 13: The fits of the widths of the bottom horizontal electrode (a,c) and top vertical electrode (b,d) in blackbird 4 are shown above, The exact values are found in table .3.2.



(a) The line width of the bottom horizontal electrode of nighthawk fitted with a polynomial function against the x coordinates of the chip.

(b) The line width of the bottom horizontal electrode of nighthawk fitted with a polynomial function against the y coordinates of the chip.

(c) The line width of the top vertical electrode of nighthawk fitted with a polynomial function against the x coordinates of the chip.

(c) The line width of the top vertical electrode of nighthawk fitted with a polynomial function against the y coordinates of the chip.

Figure 14: The fits of the widths of the bottom horizontal electrode (a,c) and top vertical electrode (b,d) in blackbird 4 are shown above, The exact values are found in table **??**.

.3.2 Tables

All fits for all chips are on the following page, followed by the tables of the the raw data itself for every chip. In these tables; seg is shortened for segmentation, thres is shortened for threshold, and the result of the fits are in nm. When a Josephson junction has shortened it is shown with a -.

		Fits:	$a\lvert x-b\rvert +$				$a\,x^2+b\,x+c$			
serie	methode	loc	a	b	c	R^2	a	b	c	R^2
bb1 c	thres	bottom x	-0.18 ± 0.19	7.32 ± 11.70	15 ± 3.89	0.039	-0.00424 ± 0.006	0.124 ± 0.092	13.056 ± 3.089	0.074
bb1 c	seg	bottom x	-0.21 ± 0.46	-27.45 ± 5.46E+08	24 ± 1.13E+08	0.008	-1.41E-16 ± 0.031	-0.207 ± 0.440	18.422 ± 14.792	0.008
bb1 c	imgj	bottom x	-0.46 ± 0.11	-1.14 ± 2.07	24 ± 2.20	0.373	-0.01363 ± 0.003	0.077 ± 0.045	21.838 ± 1.513	0.435
bb1 e	thres	bottom x	0.00 ± 0.81	0.06 ± 0.00	14 ± 13.89	0.000	-3.02E-23 ± 0.023	-0.061 ± 0.382	14.216 ± 9.541	0.001
bb1 e	seg	bottom x	0.00 ± 0.43	29.29 ± 0.00	12 ± 13.87	0.000	-1.12E-17 ± 0.026	-0.177 ± 0.425	11.996 ± 10.621	0.005
bb1 e	imgj	bottom x	-0.25 ± 0.37	26.39 ± 1.06E+09	27 ± 2.66E+08	0.018	-1.27E-21 ± 0.019	0.252 ± 0.311	20.098 ± 7.760	0.018
bb1	thres	bottom x	-0.05 ± 0.20	26.00 ± 7.92E+08	14 ± 3.57E+07	0.001	-3.64E-25 ± 0.013	0.045 ± 0.195	12.746 ± 5.586	0.001
bb1	seg	bottom x	-0.21 ± 0.51	-8.20 ± 28.42	18 ± 9.83	0.003	-2.43E-22 ± 0.019	-0.212 ± 0.297	14.960 ± 8.516	0.008
bb1	imgj	bottom x	-0.17 ± 0.16	27.41 ± 9.91E+08	23 ± 1.72E+08	0.018	-4.39E-26 ± 0.010	0.173 ± 0.157	18.625 ± 4.505	0.018
bb1	thres	bottom y	-0.05 ± 0.42	8.26 ± 2.03E+08	14 ± 1.02E+07	0.000	-0.06678 ± 0.122	-2.634 ± 4.902	-8.197 ± 40.812	0.005
bb1	seg	bottom y	0.00 ± 0.63	29.98 ± 0.00	15 ± 31.75	0.000	-3.50E-19 ± 0.181	-1.214 ± 7.308	-8.451 ± 60.843	0.055
bb1	imgj	bottom y	-0.18 ± 0.35	6.52 ± 1.39E+08	23 ± 2.51E+07	0.004	-0.03967 ± 0.099	-1.414 ± 3.984	9.059 ± 33.167	0.007
bb1	thres	top x	-0.23 ± 0.16	1.28 ± 5.98	9 ± 2.96	0.031	-0.00598 ± 0.005	-0.062 ± 0.073	7.143 ± 2.080	0.033
bb1	seg	top x	-0.74 ± 0.33	11.38 ± 6.64	19 ± 7.35	0.071	-0.03892 ± 0.014	0.351 ± 0.213	18.595 ± 6.096	0.143
bb1	imgj	top x	-0.30 ± 0.10	-4.72 ± 2.94	12 ± 1.76	0.155	-0.00823 ± 0.003	-0.015 ± 0.044	9.602 ± 1.265	0.112
bb1	thres	top y	0.00 ± 0.16	-2.50 ± 0.00	5 ± 2.97	0.000	-7.18E-19 ± 0.046	-0.096 ± 1.855	3.260 ± 15.441	0.006
bb1	seg	top y	-1.07 ± 0.48	2.61 ± 4.30E+07	29 ± 4.59E+07	0.071	-0.08058 ± 0.138	-2.171 ± 5.564	-0.548 ± 46.329	0.076
bb1	imgj	top y	-0.21 ± 0.10	8.02 ± 1.20E+08	13 ± 2.57E+07	0.069	-0.00691 ± 0.028	-0.064 ± 1.139	8.661 ± 9.480	0.069
bb2 c	thres	bottom x	-0.48 ± 0.55	4.39 ± 10.87	32 ± 10.59	0.042	-0.0141 ± 0.016	0.114 ± 0.248	29.171 ± 7.580	0.043
bb2 c	seg	bottom x	-0.15 ± 0.78	18.89 ± 87.15	29 ± 17.11	0.003	-0.00206 ± 0.032	0.116 ± 0.491	26.837 ± 15.001	0.003
bb2 c	imgj	bottom x	-1.21 ± 0.07	2.38 ± 0.52	28 ± 1.38	0.934	-0.03537 ± 0.002	0.130 ± 0.036	19.761 ± 1.106	0.917
bb2 e	thres	bottom x	-0.57 ± 0.55	2.02 ± 8.46	38 ± 10.54	0.050	-0.01659 ± 0.016	0.077 ± 0.247	34.273 ± 7.536	0.051
bb2 e	seg	bottom x	-0.14 ± 0.79	20.22 ± 101.47	35 ± 17.36	0.002	-0.00142 ± 0.033	0.112 ± 0.498	32.088 ± 15.219	0.002
bb2 e	imgj	bottom x	-1.18 ± 0.10	1.50 ± 0.74	28 ± 1.94	0.867	-0.0344 ± 0.003	0.053 ± 0.048	20.500 ± 1.455	0.854
bb2	thres	bottom x	-0.52 ± 0.38	3.11 ± 6.50	35 ± 7.25	0.045	-0.01534 ± 0.011	0.096 ± 0.170	31.722 ± 5.189	0.046
bb2	seg	bottom x	-0.15 ± 0.53	19.53 ± 64.23	32 ± 11.80	0.003	-0.00174 ± 0.022	0.114 ± 0.338	29.463 ± 10.343	0.003
bb2	imgj	bottom x	-1.19 ± 0.06	1.95 ± 0.45	28 ± 1.18	0.896	-0.03488 ± 0.002	0.091 ± 0.030	20.131 ± 0.909	0.879
bb2	thres	bottom y	-0.23 ± 1.74	-13.80 ± 50.02	28 ± 15.30	0.005	-1.13E-23 ± 0.101	-0.259 ± 4.075	20.940 ± 33.975	0.010
bb2	seg	bottom y	-0.52 ± 0.88	-30.00 ± 18.98	34 ± 10.40	0.010	-1.03E-19 ± 0.196	-0.687 ± 7.910	15.085 ± 65.944	0.019
bb2	imgj	bottom y	-0.10 ± 0.19	8.83 ± 2.86E+08	10 ± 2.99E+07	0.007	-3.75E-16 ± 0.050	0.105 ± 2.010	9.508 ± 16.760	0.007
bb2	thres	top x	-0.03 ± 0.04	26.63 ± 7.42E+08	2 ± 1.93E+07	0.008	-9.58E-17 ± 0.003	0.026 ± 0.042	1.145 ± 1.296	0.008
bb2	seg	top x	-0.06 ± 1.90	8.90 ± 354.60	174 ± 36.36	0.000	-7.35E-21 ± 0.056	-0.176 ± 0.851	172.683 ± 26.013	0.001
bb2	imgj	top x	-2.55 ± 2.53	26.00 ± 21.09	94 ± 53.87	0.098	-3.77E-25 ± 0.099	2.888 ± 1.523	31.608 ± 46.087	0.100
bb2	thres	top y	-0.32 ± 0.10	9.53 ± 1.38E+08	11 ± 4.42E+07	0.267	-0.07006 ± 0.019	-2.493 ± 0.749	-15.258 ± 6.247	0.443
bb2	seg	top y	-1.97 ± 2.19	-11.50 ± 11.88	189 ± 25.83	0.028	-3.51E-21 ± 0.491	2.074 ± 19.789	214.233 ± 164.987	0.028
bb2	imgj	top y	0.00 ± 16.64	-18.28 ± 0.00	35 ± 141.87	0.000	-2.34E-16 ± 1.183	-1.993 ± 45.584	-2.135 ± 358.222	0.009
bb4	thres	bottom x	-0.95 ± 0.06	5.84 ± 0.78	22 ± 1.34	0.881	-0.02398 ± 0.002	0.243 ± 0.040	15.106 ± 1.143	0.843
bb4	seg	bottom x	-0.75 ± 0.25	6.36 ± 3.84	16 ± 5.23	0.231	-0.01636 ± 0.007	0.197 ± 0.143	8.907 ± 4.037	0.176
bb4	imgj	bottom x	-0.87 ± 0.07	5.38 ± 0.95	31 ± 1.41	0.847	-0.02288 ± 0.002	0.165 ± 0.046	24.853 ± 1.217	0.798
bb4	thres	bottom y	-0.83 ± 0.10	-1.02 ± 1.36	19 ± 1.95	0.708	-0.02174 ± 0.003	-0.060 ± 0.059	14.583 ± 1.654	0.671
bb4	seg	bottom y	-0.65 ± 0.23	-5.29 ± 4.59	14 ± 4.91	0.206	-0.01294 ± 0.007	-0.229 ± 0.143	8.161 ± 3.999	0.192
bb4	imgj	bottom y	-0.83 ± 0.10	1.53 ± 1.25	30 ± 1.84	0.724	-0.02328 ± 0.003	0.061 ± 0.054	24.863 ± 1.429	0.721
bb4	thres	top x	-0.77 ± 0.10	-0.47 ± 1.57	41 ± 2.12	0.636	-0.02229 ± 0.003	0.003 ± 0.055	37.336 ± 1.543	0.697
bb4	seg	top x	-1.33 ± 0.59	8.10 ± 5.21	68 ± 12.42	0.146	-0.02634 ± 0.016	0.356 ± 0.337	54.068 ± 9.525	0.095
bb4	imgj	top x	-0.68 ± 0.12	0.76 ± 1.90	46 ± 2.26	0.535	-0.01999 ± 0.003	0.059 ± 0.064	42.302 ± 1.674	0.587
bb4	thres	top y	-0.78 ± 0.07	-8.14 ± 1.30	43 ± 1.61	0.794	-0.01999 ± 0.001	-0.255 ± 0.031	36.876 ± 0.859	0.906
bb4	seg	top y	-1.19 ± 0.53	-9.39 ± 6.32	66 ± 11.70	0.151	-0.01882 ± 0.016	-0.561 ± 0.332	52.456 ± 9.278	0.142
bb4	imgj	top y	-0.75 ± 0.08	-8.56 ± 1.39	48 ± 1.72	0.743	-0.02004 ± 0.002	-0.269 ± 0.041	42.236 ± 1.071	0.831
jason	thres	bottom x	-5.00 ± 4.10	-7.06 ± 1.44E+08	32 ± 7.19E+08	0.190	-5.21E-17 ± 1.275	-14.577 ± 3.801	-6.357 ± 13.830	0.334
jason	seg	bottom x	-5.00 ± 5.11	-7.26 ± 2.33E+08	31 ± 1.17E+09	0.195	-1.37E-19 ± 1.105	-11.095 ± 3.295	-7.838 ± 11.989	0.279
jason	imgj	bottom x	-3.72 ± 2.90	-5.03 ± 1.01E+08	108 ± 3.76E+08	0.117	-1.27281 ± 1.333	-7.247 ± 4.596	95.706 ± 10.664	0.175
jason	thres	bottom y	-5.00 ± 4.24	-15.73 ± 4.33E+08	52 ± 2.17E+09	0.133	-0.04043 ± 1.408	8.463 ± 74.302	250.000 ± 972.955	0.188
jason	seg	bottom y	-5.00 ± 3.81	-6.47 ± 1.44E+08	95 ± 7.18E+08	0.144	-3.79E-22 ± 1.180	8.846 ± 62.291	229.137 ± 815.683	0.177
jason	imgj	bottom y	-2.97 ± 8.64	-10.55 ± 3.87E+08	140 ± 1.15E+09	0.090	-0.21696 ± 0.789	-8.343 ± 41.184	25.548 ± 532.869	0.096
jason	thres	top x	-1.58 ± 2.07	5.78 ± 8.26E+07	25 ± 1.30E+08	0.020	-0.3707 ± 0.688	1.318 ± 2.052	18.493 ± 7.465	0.029
jason	seg	top x	-1.36 ± 1.96	-3.26 ± 5.00	20 ± 8.30	0.016	-9.22E-23 ± 0.533	-1.025 ± 1.590	15.209 ± 5.785	0.014
jason	imgj	top x	-0.21 ± 0.28	7.03 ± 4.88E+07	19 ± 1.02E+07	0.023	-1.39E-16 ± 0.084	0.208 ± 0.252	17.249 ± 0.917	0.023
jason	thres	top y	0.00 ± 2.02	19.96 ± 0.00	15 ± 93.87	0.000	-2.89E-14 ± 0.660	-3.680 ± 34.825	-81.756 ± 456.017	0.107
jason	seg	top y	-0.90 ± 1.63	4.54 ± 9.03E+07	43 ± 8.11E+07	0.011	-1.15E-14 ± 0.534	0.899 ± 28.187	39.249 ± 369.104	0.011
jason	imgj	top y	0.00 ± 0.25	8.59 ± 0.00	17 ± 8.68	0.000	-0.02596 ± 0.083	-1.694 ± 4.375	-9.249 ± 57.295	0.059
Nighthawk	thres	bottom x	-0.02 ± 0.02	-0.32 ± 0.35	9.86 ± 10.04	0.139	-0.92 ± 0.52	-8.10 ± 0.35	18.72 ± 12.32	0.118
Nighthawk	seg	bottom x	-0.02 ± 0.00	0.13 ± 0.07	41.40 ± 2.07	0.568	-0.74 ± 0.14	2.32 ± 0.07	45.59 ± 2.83	0.596
Nighthawk	imgj	bottom x	-0.02 ± 0.00	0.02 ± 0.05	14.28 ± 1.47	0.725	-0.60 ± 0.10	0.12 ± 0.05	17.33 ± 1.98	0.733
Nighthawk	thres	bottom y	-0.02 ± 0.02	-0.04 ± 0.32	8.27 ± 10.11	0.075	-0.76 ± 0.60	-0.11 ± 0.32	14.25 ± 12.79	0.051
Nighthawk	seg	bottom y	-0.02 ± 0.00	-0.26 ± 0.06	39.37 ± 1.73	0.661	-0.69 ± 0.11	-7.92 ± 0.06	45.74 ± 2.43	0.708
Nighthawk	imgj	bottom y	-0.02 ± 0.00	-0.17 ± 0.03	13.78 ± 0.88	0.810	-0.63 ± 0.07	-5.23 ± 0.03	18.32 ± 1.43	0.873
Nighthawk	thres	top x	-0.01 ± 0.02	0.09 ± 0.29	41.13 ± 8.37	0.026	-0.36 ± 0.51	-1.10 ± 0.29	44.63 ± 10.44	0.014
Nighthawk	seg	top x	-0.02 ± 0.01	0.21 ± 0.13	65.19 ± 3.66	0.309	-0.77 ± 0.26	6.16 ± 0.13	69.71 ± 5.27	0.387
Nighthawk	imgj	top x	-0.02 ± 0.00	0.19 ± 0.03	56.91 ± 0.82	0.906	-0.74 ± 0.06	6.60 ± 0.03	62.41 ± 1.20	0.912
Nighthawk	thres	top y	-0.02 ± 0.01	-0.03 ± 0.25	47.88 ± 7.73	0.105	-0.74 ± 0.47	-3.67 ± 0.25	51.74 ± 9.96	0.118
Nighthawk	seg	top y	-0.01 ± 0.00	-0.55 ± 0.06	61.62 ± 1.87	0.762	-0.81 ± 0.11	-17.72 ± 0.06	73.76 ± 2.72	0.840
Nighthawk	imgj	top y	-0.01 ± 0.00	-0.06 ± 0.05	55.51 ± 1.62	0.588	-0.53 ± 0.09	-1.76 ± 0.05	58.93 ± 2.01	0.559

Blackbird 1	chip coodinates (mm)		overlap Area (nm^2)			Total area (nm^2)			Bottom vertical width (nm)			Top horizontal width(nm)			Conductance	Conductanance/area(μS/μm^2)		
Image name	x	y	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	(μS)	ImageJ	Threshold	Segment
1c_xtype_D3_b3_L	-25.3	-13.8	19734.943	23750.33	17637.44	39287.61	43406.78	35454.24	111	111.894	158.746	173	170.872	135.044	82.8	2108.58	1908.48	2336.56
1c_xtype_D3_b3_R	-25.3	-13.8	19552.666	19656.45	17816.8				107	108.313	162.884	175.5	167.844	136.425	82.8	2108.58	1908.48	2336.56
1c_xtype_D6_b3_L	-28.3	-11.15	39953.055	39273.94	39496.8	79469.34	76472.74	76737.61	214	227.376	169.498	178	195.956	200.504	168.3	2117.18	2200.14	2192.55
1c_xtype_D6_b3_R	-28.3	-11.15	39516.287	37198.8	37240.82				215	211.885	164.537	176	170.924	195.13	168.3	2117.18	2200.14	2192.55
1c_xtype_X2_b3_L	-25.3	-11.5	29632.237	27067.99	26645.52	49951.17	46369.64	44736.45	157	156.738	161.803	177	160.831	162.184	116.5	2331.98	2512.10	2603.81
1c_xtype_X2_b3_R	-25.3	-11.5	20318.934	19301.65	18090.94				120	104.593	151.308	176	172.805	134.772	116.5	2331.98	2512.10	2603.81
1e_xtype_D3_b3_L	-25.3	-31	15337.333	17832.52	15318.13	30608.51	35790.63	30139.94	96	100.872	161.23	151.5	182.727	127.33	78.0	2548.31	2179.34	2587.93
1e_xtype_D3_b3_R	-25.3	-31	15271.18	17958.1	14821.81				95	97.2412	157.096	152	160.457	125.263	78.0	2548.31	2179.34	2587.93
1e_xtype_D5_b3_L_inver	-25.3	-26.3	32039.257	33850.92	35578.22	63898.14	67197.10	70930.17	198	194.241	164.537	158	157.374	189.342	-1.0	-15.65	-14.88	-14.10
1e_xtype_D5_b3_R_on S	-25.3	-26.3	31858.885	33346.17	35351.94				199	201.156	164.537	154	148.983	188.928	-1.0	-15.65	-14.88	-14.10
2c_xtype_D3_b3_L	-8.2	-13.8	20768.399	21121.79	21053.11	40897.78	40127.86	41581.16	111	104.575	181.074	178	179.702	148.001	147.0	3594.33	3663.29	3535.25
2c_xtype_D3_b3_R	-8.2	-13.8	20129.379	19006.07	20528.05				110	99.2686	176.304	175	170.489	145.393	147.0	3594.33	3663.29	3535.25
2c_xtype_D6_b3_L	-11.15	-11.15	40211.051	36118.65	39380.58	79052.66	74573.80	79338.83	218	211.238	172.805	176	155.87	199.677	265.2	3354.70	3556.18	3342.60
2c_xtype_D6_b3_R	-11.15	-11.15	38841.609	38455.15	39958.25				220	215.766	173.632	174	163.457	201.331	265.2	3354.70	3556.18	3342.60
2c_xtype_D6_d1_L	-11.15	-11.15	40211.051	40229.62	44176.94	79052.66	81223.40	89149.62	218	235.942	176.113	176	155.152	212.906	265.2	3354.70	3265.04	2974.75
2c_xtype_D6_d1_R	-11.15	-11.15	38841.609	40993.78	44972.69				220	241.424	174.459	174	160.163	215.8	265.2	3354.70	3265.04	2974.75
2c_xtype_D6_d2_L	-11.15	-11.15	40211.051	40556.12	44804.51	79052.66	81789.29	88199.38	218	246.369	172.805	176	152.63	215.387	265.2	3354.70	3242.45	3006.80
2c_xtype_D6_d2_R	-11.15	-11.15	38841.609	41233.17	43394.86				220	240.761	171.152	174	157.798	211.666	265.2	3354.70	3242.45	3006.80
2c_xtype_X2_b3_L	-8.2	-9.2	29284.132	29882.19	30251.71	58303.47	59591.77	60273.99	166	159.61	174.984	173	171.444	173.18	195.0	3344.57	3272.26	3235.23
2c_xtype_X2_b3_R	-8.2	-9.2	29019.339	29709.58	30022.28				159	157.378	176.788	171	171.817	172.729	195.0	3344.57	3272.26	3235.23
2e_xtype_D6_b3_L	-11.15	-28.3	38263.78	36879.5	40461.4	75180.66	71961.73	78958.73	218	207.974	174.459	169	167.564	203.398	245.4	3264.40	3410.42	3108.20
2e_xtype_D6_b3_R	-11.15	-28.3	36916.877	35082.23	38497.33				219	204.402	170.325	165	160.013	197.61	245.4	3264.40	3410.42	3108.20
3C_xtype_D3_b3_L	8.9	-13.8	17178.03	17400.04	17902.7	34226.93	34300.69	36054.72	106	89.304	172.488	163	169.9	137.761	128.0	3739.75	3731.70	3550.16
3C_xtype_D3_b3_R	8.9	-13.8	17048.898	16900.65	18152.01				106	87.9732	168.672	164	167.663	137.761	128.0	3739.75	3731.70	3550.16
3C_xtype_D5_b3_L	8.9	-9.2	36816.637	38867.67	475.9372	72202.27	75753.34	38287.32	215	211.164	174.082	167	169.973	87.9431	234.0	3240.90	3088.97	6111.68
3C_xtype_D5_b3_R	8.9	-9.2	35385.63	36885.68	37811.38				211	200.227	174.082	162	169.464	195.73	234.0	3240.90	3088.97	6111.68
3C_xtype_D6_b3_L	5.85	-11.15	36994.173	38262.96	42222.44	72289.50	73436.29	77830.06	219	205.724	173.632	169	171.671	207.532	247.0	3416.82	3363.46	3173.58
3C_xtype_D6_b3_R	5.85	-11.15	35295.33	35173.33	35607.62				217	202.577	167.844	165	161.302	190.582	247.0	3416.82	3363.46	3173.58
3C_xtype_X2_b3_L	8.9	-11.5	25214.055	27118.24	27187.81	50513.49	54516.16	54456.17	152	141.579	172.278	165	167.193	165.062	179.0	3543.61	3283.43	3287.05
3C_xtype_X2_b3_R	8.9	-11.5	25299.433	27397.92	27268.36				154	143.16	170.474	164	167.361	164.611	179.0	3543.61	3283.43	3287.05
3e_xtype_D3_b3_L	8.9	-31	17933.31	18619.96	18682.09	34705.58	36838.06	36655.28	97	97.6515	176.304	174	170.468	140.432	124.0	3572.91	3366.08	3382.87
3e_xtype_D3_b3_R	8.9	-31	16772.268	18218.1	17973.18				101	93.4632	170.198	170	171.374	137.38	124.0	3572.91	3366.08	3382.87
3e_xtype_D5_b3_L	8.9	-26.3	37583.782	38849.78	40135.75	73195.90	74712.08	78049.64	205	199.454	182.2	167	178.214	200.691	245.0	3347.18	3279.26	3139.03
3e_xtype_D5_b3_R	8.9	-26.3	35612.118	35862.3	37913.89				204	194.458	174.984	162	171.494	195.279	245.0	3347.18	3279.26	3139.03
3e_xtype_D6_b3_L	5.85	-28.3	35891.725	37506.06	36779.36	72718.80	74174.59	74429.68	208	214.146	171.979	172	190.995	192.236	241.0	3314.14	3249.09	3237.96
3e_xtype_D6_b3_R	5.85	-28.3	36827.074	36668.52	37650.31				208	203.398	173.632	169	167.768	194.716	241.0	3314.14	3249.09	3237.96
4c_xtype_D3_b3_L	26	-13.8	16996.79	17090.11	17401.84	34570.04	33548.95	34606.10	210.6	213.149	165.364	162	159.621	135.185	80.9	2339.88	2411.10	2337.44
4c_xtype_D3_b3_R	26	-13.8	17573.253	16458.84	17204.27				210	208.802	165.364	162	158.197	134.772	80.9	2339.88	2411.10	2337.44
4c_xtype_D5_b3_L	26	-9.2	35769.415	34641.92	36846.36	69369.58	68617.78	71979.54	211	202.969	163.71	160	159.381	193.889	156.3	2253.31	2278.00	2171.60
4c_xtype_D5_b3_R	26	-9.2	33600.164	33975.87	35133.18				207	199.804	159.576	165	154.476	188.928	156.3	2253.31	2278.00	2171.60
4c_xtype_D6_b3_L	23	-11.15	35295.403	32921.25	34988.25	69946.73	65445.90	69327.74	207	200.831	165.364	162	147.85	188.928	160.8	2298.76	2456.85	2319.28
4c_xtype_D6_b3_R	23	-11.15	34651.322	32524.65	34339.48				209	188.137	165.364	161	156.844	186.035	160.8	2298.76	2456.85	2319.28
4c_xtype_X2_b3_L	26	-11.5	25217.54	25694.73	25605.41	50116.80	50184.76	51074.10	154	150.613	159.576	168	148.967	159.163	121.0	2414.72	2411.45	2369.46
4c_xtype_X2_b3_R	26	-11.5	24899.256	24490.02	25468.68				153	146.522	162.057	158	146.161	159.163	121.0	2414.72	2411.45	2369.46
4e_xtype_D3_b3_L	26	-31	18493.69	19340.27	16971.15	36402.53	37684.83	33471.96	104	94.2575	164.537	170	170.738	135.598	91.0	2499.12	2414.09	2717.93
4e_xtype_D3_b3_R	26	-31	17908.843	18344.56	16500.81				99	88.0563	162.884	170	169.085	133.531	91.0	2499.12	2414.09	2717.93
4e_xtype_D5_b3_L	26	-26.3	36200.396	34345.17	34366.15	72171.30	68196.78	68382.27	208	210.517	159.576	172	148.166	188.102	185.9	2575.72	2725.83	2718.44
4e_xtype_D5_b3_R	26	-26.3	35970.904	33851.61	34016.13				206	201.695	158.749	167	153.573	188.102	185.9	2575.72	2725.83	2718.44
4e_xtype_D6_b3_L	23	-28.3	37230.38	35194.71	34495.35	73966.94	67800.50	69006.43	207	215.8	165.364	169	192.649	190.169	182.9	2472.78	2697.68	2650.54
4e_xtype_D6_b3_R	23	-28.3	36736.562	32605.8	34511.08				210	201.701	159.576	163	147.341	188.928	182.9	2472.78	2697.68	2650.54
4e_xtype_X2_b3_L	26	-28.6	27589.646	28358.39	26598.73	52988.18	53234.09	51507.52	154	166.191	162.884	166	197.61	162.47	140.2	2646.43	2634.20	2722.50

Blackbird 1	chip coodinates (mm)		overlap Area (nm^2)			Total area (nm^2)			Bottom vertical width (nm)			Top horizontal width(nm)			Conductance	Conductanance/area(µS/µm^2)		
Image name	x	y	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	(µS)	ImageJ	Threshold	Segment
4e_xtype_X2_b3_R	26	-28.6	25398.538	24875.7	24908.79				154	144.425	157.923	167	151.686	159.163	140.2	2646.43	2634.20	2722.50
1e_xtype_D3_b3_jj_left	-25.3	-31	15337.333	19715.93	15973.73	30608.51	38934.31	30789.39	96	111.621	164.537	151.5	180.247	129.811	78.0	2548.31	2003.37	2533.34
1e_xtype_D3_b3_jj_right	-25.3	-31	15271.18	19218.38	14815.66				95	113.56	155.442	152	146.985	124.436	78.0	2548.31	2003.37	2533.34
1e_xtype_D6_b3_jj_left	-25.3	-26.3	32304.487	31604.3	397.8734	65172.46	62968.10	812.15	200	183.572	165.364	160	161.077	80.4082	165.4	2537.38	2626.21	203615.86
1e_xtype_D6_b3_jj_right	-25.3	-26.3	32867.973	31363.81	414.2806				202	184.608	165.364	154	158.215	83.5088	165.4	2537.38	2626.21	203615.86
1e_xtype_X2_b3_jj_left	-25.3	-28.6	23063.693	22351.66	405.3934	46380.68	46422.57	799.85	145	121.775	164.537	156.5	163.835	81.8552	125.0	2695.09	2692.66	156279.57
1e_xtype_X2_b3_jj_right	-25.3	-28.6	23316.985	24070.91	394.4553				146	134.01	160.403	157	159.472	79.7881	125.0	2695.09	2692.66	156279.57
2e_xtype_D3_b3_jj_left	-8.2	-31	18646.95	18649.57	20970.39	38103.95	36343.70	43312.15	113	95.8988	170.325	178	162.772	145.934	147.0	3857.87	4044.72	3393.97
2e_xtype_D3_b3_jj_right	-8.2	-31	19457.004	17694.13	22341.75				110	91.8942	171.979	175	165.889	150.068	147.0	3857.87	4044.72	3393.97
2e_xtype_D5_b3_jj_left	-8.2	-26.3	37046.425	37295.81	40370.48	74415.09	72593.51	79557.59	218	201.889	171.979	171	169.392	202.571	-1.0	-13.44	-13.78	-12.57
2e_xtype_D5_b3_jj_right	-8.2	-26.3	37368.662	35297.7	39187.11				217	210.797	170.325	172	161.925	199.677	-1.0	-13.44	-13.78	-12.57
2e_xtype_X2_b3_jj_left	-8.2	-28.6	26996.79	27038.49	29853.5	54507.21	52294.39	58246.76	160	152.115	172.805	168	164.778	171.979	195.0	3577.51	3728.89	3347.83
2e_xtype_X2_b3_jj_right	-8.2	-28.6	27510.416	25255.91	28393.26				158	150.481	167.018	167	148.98	167.844	195.0	3577.51	3728.89	3347.83
3C_xtype_D5_a1_left_JJ	8.9	-9.2	36816.637	32928.46	34034.39	72202.27	65254.94	67573.62	215	170.678	171.647	167	169.002	184.049	234.0	3240.90	3585.94	3462.89
3C_xtype_D5_a1_right_J	8.9	-9.2	35385.63	32326.48	33539.23				211	168.125	170.655	162	171.16	182.561	234.0	3240.90	3585.94	3462.89
3C_xtype_X3_a1_left	8.9	-11.5	25214.055	23571.09	22967.53	50513.49	46139.98	45490.10	152	117.07	173.632	165	167.867	152.3	179.0	3543.61	3879.50	3934.92
3C_xtype_X3_a1_right	8.9	-11.5	25299.433	22568.88	22522.57				154	115.107	169.663	164	160.195	150.315	179.0	3543.61	3879.50	3934.92
3C_xtype_X3_c3_left	8.9	-13.8	25214.055	30487.52	31468.01	50513.49	61173.37	61743.88	152	159.077	173.632	165	171.358	176.608	179.0	3543.61	2926.11	2899.07
3C_xtype_X3_c3_right	8.9	-13.8	25299.433	30685.85	30275.87				154	164.641	170.655	164	162.959	173.135	179.0	3543.61	2926.11	2899.07

Blackbird 2	Chip coodinates (mm)		Overlap area (nm^2)			Total Area (nm^2)			Bottom vertical width (nm)			Top horizontal width(nm)			Conductance	Conductanance/area(µS/µm^2)		
Image name	x	y	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	(µS)	ImageJ	Threshold	Segment
1c_xtype_D3_b3_L	-25.3	-13.8	16322.9	16558.86	16731.87	32213.86	32247.46	32148.44	81.0	132.8	166.4	96.2	178.0	185.5	79.22	2456.68	2459.24	2464.25
1c_xtype_D3_b3_R	-25.3	-13.8	15655	15688.6	15589.58				81.0	128.6	163.3	93.9	174.0	180.9	79.22	2456.68	2459.24	2464.25
1c_xtype_D5_b3_L	-25.3	-9.2	31580.23	36161.39	35633.46	47649.68	52732.78	52529.96	195.0	189.0	165.1	170.5	164.0	388.3	180.22	3417.68	3782.27	3430.88
1c_xtype_D5_b3_R	-25.3	-9.2	31090.82	36173.91	35971.09				191.0	189.4	168.7	172.3	160.0	451.9	180.22	3417.68	3782.27	3430.88
1c_xtype_X2_b3_L	-25.3	-11.5	21666.26	33729.04	24797.55	36360.83	45750.81	41445.91	132.0	156.5	166.9	120.9	168.0	498.8	127.57	2788.31	3508.37	3077.92
1c_xtype_X2_b3_R	-25.3	-11.5	19801.96	29191.94	24887.04				132.0	156.9	164.2	116.4	162.0	447.4	127.57	2788.31	3508.37	3077.92
1e_xtype_D3_b3_L	-25.3	-31	12999.69	25329.69	15196.38	29255.53	32484.54	32231.15	81.0	126.3	158.0	83.2	153.0	379.3	65.06	2002.79	2223.84	2018.54
1e_xtype_D3_b3_R	-25.3	-31	12696.67	15925.68	15672.29				76.0	127.1	154.2	100.7	151.0	171.0	65.06	2002.79	2223.84	2018.54
1e_xtype_D5_b3_L	-25.3	-26.3	28456.29	33528.97	32456.48	46510.98	50600.22	51234.76	187.0	180.8	153.3	170.5	163.0	401.4	159.40	3150.25	3427.22	3111.24
1e_xtype_D5_b3_R	-25.3	-26.3	29952.11	34041.36	34675.89				194.0	186.3	163.3	171.4	158.0	305.8	159.40	3150.25	3427.22	3111.24
1e_xtype_X2_b3_L	-25.3	-28.6	21560.75	26845.15	24922.84	36895.69	47766.19	39561.68	127.0	157.4	171.4	113.6	151.0	431.1	113.97	2386.07	3089.08	2880.91
1e_xtype_X2_b3_R	-25.3	-28.6	20336.82	31207.32	23002.82				126.0	150.6	159.7	122.7	156.0	388.8	113.97	2386.07	3089.08	2880.91
2c_xtype_D3_b3_L	-8.2	-13.8	15922.51	20721.08	19953.23	32590.84	38031.36	36313.11	91.0	142.4	167.7	120.1	163.0	191.5	92.40	2429.58	2835.16	2544.55
2c_xtype_D3_b3_R	-8.2	-13.8	16031.98	21472.5	19754.24				93.0	141.6	165.1	127.2	159.0	196.6	92.40	2429.58	2835.16	2544.55
2c_xtype_D5_b3_L	-8.2	-11.5	33697.29	44662.42	38434.76	49243.86	56565.2	56057.78	209.0	196.5	168.7	192.5	163.0	460.4	205.17	3627.22	4166.50	3660.05
2c_xtype_D5_b3_R	-8.2	-11.5	32685	40006.33	39498.91				208.0	199.9	168.7	188.5	159.0	465.3	205.17	3627.22	4166.50	3660.05
2c_xtype_X2_b3_L	-8.2	-9.2	25512.54	29133.39	30544.59	40780.11	47315.4	45311.17	154.0	173.6	171.4	131.7	161.0	265.2	155.07	3277.38	3802.60	3422.35
2c_xtype_X2_b3_R	-8.2	-9.2	24221.25	30756.54	28752.3				160.0	168.2	167.8	134.4	156.0	373.0	155.07	3277.38	3802.60	3422.35
2e_xtype_D3_b3_L	-8.2	-31	14835.55	17234.36	18126.38	32080.95	35531.9	34441.18	98.0	136.2	159.5	80.1	155.0	260.3	71.32	2007.11	2223.01	2070.67
2e_xtype_D3_b3_R	-8.2	-31	15522.08	18973.03	17882.31				101.0	135.1	158.7	87.0	152.0	265.6	71.32	2007.11	2223.01	2070.67
2e_xtype_D5_b3_L	-8.2	-26.3	34472.85	43440	37826.03	48645.54	59359.68	52539.72	207.0	194.8	168.7	202.0	164.5	322.9	180.80	3045.90	3716.75	3441.27
2e_xtype_D5_b3_R	-8.2	-26.3	32086.68	42800.82	35980.85				209.0	190.3	162.4	198.4	158.5	389.7	180.80	3045.90	3716.75	3441.27
2e_xtype_X2_b3_L	-8.2	-28.6	24227.74	30533.69	26800.55	42088.45	43582.08	44212.85	146.0	162.8	165.1	140.7	163.0	173.2	127.44	2924.17	3027.94	2882.45
2e_xtype_X2_b3_R	-8.2	-28.6	25529.58	27023.22	27653.99				149.0	165.5	163.3	129.0	161.0	225.5	127.44	2924.17	3027.94	2882.45
3c_xtype_D3_b3_L	8.9	-13.8	16454.24	18146.2	21630.73	32595.86	37333.7	36290.05	101.5	148.4	174.0	78.6	167.5	360.2	95.26	2551.69	2922.58	2625.08
3c_xtype_D3_b3_R	8.9	-13.8	16037	20774.84	19731.18				100.5	142.0	167.1	85.5	164.0	351.1	95.26	2551.69	2922.58	2625.08
3c_xtype_D5_b3_L	8.9	-9.2	33774.86	40035.08	36759.44	51728.09	57481.7	54983.68	216.0	193.5	168.7	198.4	159.7	211.1	203.96	3548.29	3942.95	3709.49
3c_xtype_D5_b3_R	8.9	-9.2	35169.22	40922.84	38424.81				219.0	196.6	171.4	197.5	163.0	255.3	203.96	3548.29	3942.95	3709.49
3c_xtype_X2_b3_L	8.9	-11.5	27860.56	32353.34	29838.42	42903.32	46434.91	45987.24	155.0	171.8	174.1	134.4	165.7	465.0	146.71	3159.51	3419.59	3190.27
3c_xtype_X2_b3_R	8.9	-11.5	26344.45	29876.04	29428.38				157.0	170.5	167.8	135.3	162.0	467.2	146.71	3159.51	3419.59	3190.27
3e_xtype_D3_b3_L	8.9	-31	16201.72	19944.22	18253.37	32741.36	36412.47	35053.97	104.0	136.6	162.6	93.1	158.5	172.5	88.08	2418.94	2690.16	2512.68
3e_xtype_D3_b3_R	8.9	-31	16182.5	19853.61	18495.11				102.6	137.4	161.8	92.3	153.0	331.2	88.08	2418.94	2690.16	2512.68
3e_xtype_D5_b3_L	8.9	-26.3	35065.19	43924.22	40482.35	50870.2	56390.79	54377.57	215.0	202.4	170.7	198.4	165.0	458.4	194.01	3440.43	3813.80	3567.81
3e_xtype_D5_b3_R	8.9	-26.3	34311.34	39831.93	37818.7				209.0	196.2	169.6	198.4	157.0	274.7	194.01	3440.43	3813.80	3567.81
3e_xtype_X2_b3_L	8.9	-28.6	25350.22	31518.87	27955.82	40953.8	48830.65	43249.59	154.0	166.4	165.1	136.2	161.0	470.8	133.68	2737.66	3264.21	3090.94
3e_xtype_X2_b3_R	8.9	-28.6	24394.94	32271.79	26690.72				160.0	162.4	161.5	139.8	156.0	443.8	133.68	2737.66	3264.21	3090.94
4c_xtype_D3_b3_L	26	-13.8	13299.23	16165.65	18859.17	30824.62	34678.26	32122.23	80.0	141.2	175.5	899.8	168.6	173.3	85.36	2461.62	2769.37	2657.50
4c_xtype_D3_b3_R	26	-13.8	14265.76	18119.39	15563.36				81.0	129.4	166.4	108.4	166.0	196.9	85.36	2461.62	2769.37	2657.50
4c_xtype_D5_b3_L	26	-9.2	31226.36	37595.36	36597.54	48837.9	53012.69	54552.49	191.6	192.1	170.5	176.8	159.7	480.8	190.47	3592.93	3900.06	3491.52
4c_xtype_D5_b3_R	26	-9.2	32279.04	36453.83	37993.62				192.0	194.8	175.9	174.1	164.5	344.1	190.47	3592.93	3900.06	3491.52
4c_xtype_X2_b3_L	26	-11.5	23249.74	25390.83	25301.15	38304.35	41191.53	42058.59	132.0	158.7	173.2	112.7	169.0	472.6	71.11	1726.28	1856.40	1690.69
4c_xtype_X2_b3_R	26	-11.5	21745.49	24632.66	25499.73				134.0	159.1	166.4	116.8	162.0	356.0	71.11	1726.28	1856.40	1690.69
4e_xtype_D3_b3_L	26	-31	12997.75	32259.26	17439.43	32338.52	35813.98	35344.06	81.0	135.2	167.8	905.4	157.0	186.9	76.12	2125.41	2353.83	2153.67
4e_xtype_D3_b3_R	26	-31	15779.66	19255.11	18785.2				82.0	138.9	166.4	82.4	165.0	364.8	76.12	2125.41	2353.83	2153.67
4e_xtype_D5_b3_L	26	-26.3	30271.08	35780.81	34039.68	47045.03	51571.76	49054.39	191.0	184.9	164.2	177.7	159.7	292.7	166.57	3229.94	3540.72	3395.69
4e_xtype_D5_b3_R	26	-26.3	30486.16	35012.89	32495.53				192.7	180.8	163.3	178.6	157.0	163.3	166.57	3229.94	3540.72	3395.69
4e_xtype_X2_b3_L	26	-28.6	22413.77	29530.87	26311.6	38303.85	42970.81	41432.89	133.0	161.9	166.9	119.1	163.0	465.4	117.33	2730.41	3063.08	2831.75
4e_xtype_X2_b3_R	26	-28.6	21744.99	26411.94	24874.03				135.7	156.9	166.0	110.9	158.5	388.8	117.33	2730.41	3063.08	2831.75

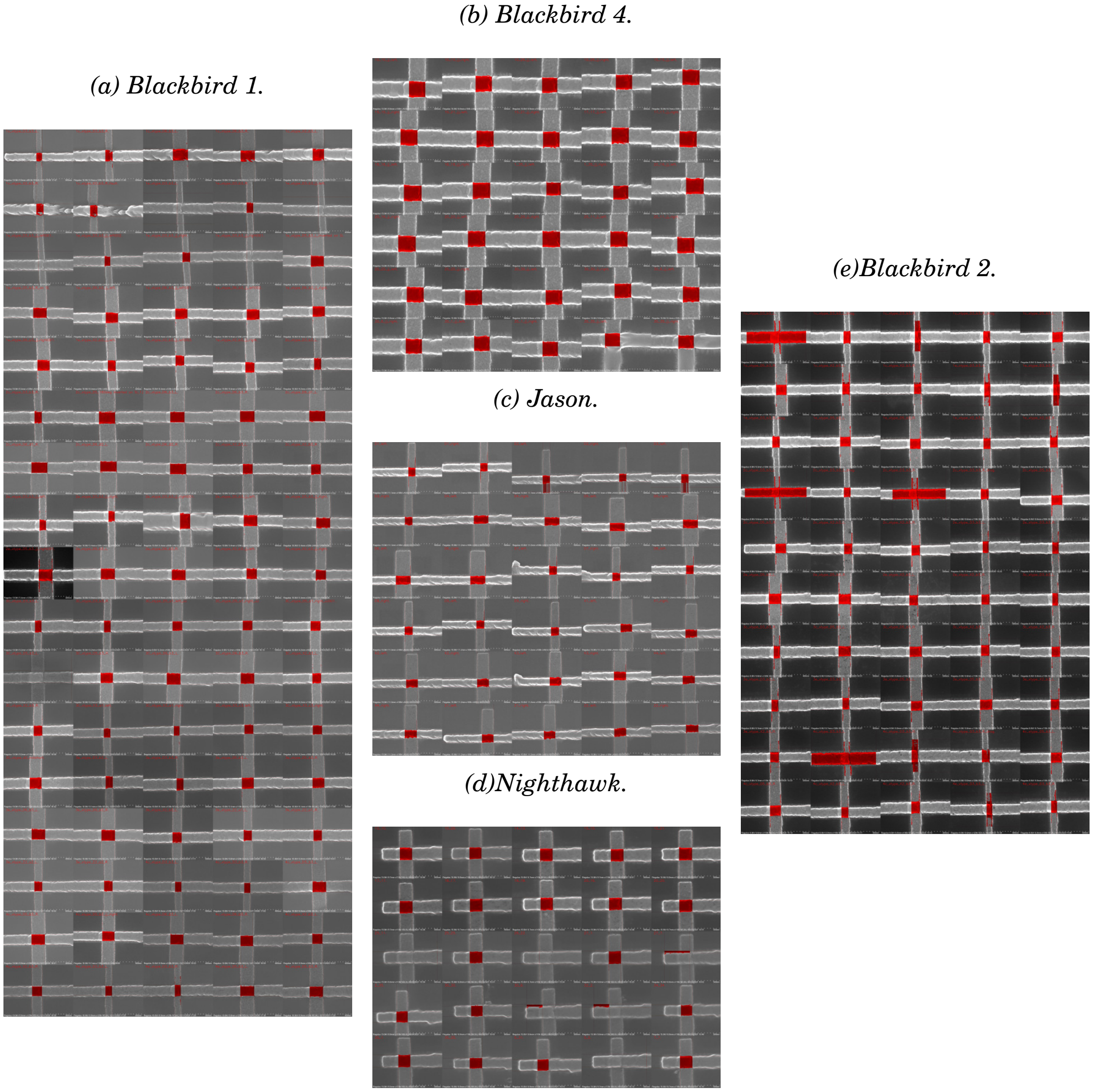
Blackbird 4	Chip coodinates (mm)		Overlap area (nm^2)			Total Area (nm^2)			Bottom vertical width (nm)			Top horizontal width(nm)			Conductance	Conductanance/area(µS/µm^2)		
Image name	x	y	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	(µS)	ImageJ	Threshold	Segment
10_10_jj_left	-16	-16	42673.130	39242.6	51616.84	82146.8	79983.3	100278.7	205.0	193.1	210.3	210.0	197.8	227.5	239.15	2911.22	2989.97	2384.83
10_10_jj_right	-16	-16	39473.684	40740.72	48661.85				205.0	199.1	136.1	200.0	195.8	146.1	239.15	2911.22	2989.97	2384.83
10_26_jj_left	-16	16	38585.093	34918.2	48661.85	78827.5	72839.6	49041.2	198.0	177.9	136.1	196.0	190.5	146.1	252.72	3205.98	3469.53	5153.20
10_26_jj_right	-16	16	40242.382	37921.4	379.3265				204.0	196.5	195.1	197.0	187.2	96.1	252.72	3205.98	3469.53	5153.20
15_15_jj_left	-6	-6	41097.645	40390.08	47105.61	85334.1	81920.8	96085.5	221.0	205.7	197.8	193.0	187.2	217.6	280.03	3281.56	3418.29	2914.37
15_15_jj_right	-6	-6	44236.496	41530.77	48979.93				216.0	205.0	206.4	206.0	197.8	221.9	280.03	3281.56	3418.29	2914.37
15_21_jj_left	-6	6	44293.629	42116.16	48347.28	87784.4	83755.2	100982.7	209.0	206.4	205.7	198.0	195.1	219.9	274.79	3130.28	3280.87	2721.16
15_21_jj_right	-6	6	43490.738	41639.05	52635.37				220.0	206.4	208.4	203.0	198.4	229.9	274.79	3130.28	3280.87	2721.16
17_17_jj_left	-2	-2	45083.102	42026.33	51447.96	89319.6	84532.4	99073.8	218.0	203.7	205.7	204.0	199.8	227.5	278.38	3116.65	3293.15	2809.80
17_17_jj_right	-2	-2	44236.496	42506.08	47625.82				213.0	207.7	203.7	204.0	195.8	218.6	278.38	3116.65	3293.15	2809.80
18_18_jj_left	0	0	44703.515	42830.1	48992.62	88413.3	85837.7	97872.4	218.0	212.3	206.4	205.0	197.8	221.6	239.08	2704.14	2785.27	2442.79
18_18_jj_right	0	0	43709.747	43007.64	48879.74				216.0	208.4	207.0	200.0	197.8	220.9	239.08	2704.14	2785.27	2442.79
1_35_jj_left	-34	34	33231.475	29582.92	44207.06	66558.6	61898.5	85196.6	193.0	172.0	179.3	174.0	164.0	213.0	-			
1_35_jj_right	-34	34	33327.13	32315.59	40989.57				195.0	178.6	178.6	168.0	169.3	204.7	-			
20_16_jj_left	4	-4	44623.009	42517.49	49406.94	90008.2	85284.6	99795.2	220.0	212.3	205.0	196.0	192.5	223.2	293.09	3256.31	3436.66	2936.96
20_16_jj_right	4	-4	45385.215	42767.15	50388.29				217.0	209.0	206.4	203.0	198.1	224.9	293.09	3256.31	3436.66	2936.96
20_20_jj_left	4	4	44210.526	42398.68	48131.15	88544.4	86187.2	98724.6	221.0	211.7	203.7	196.0	194.5	219.6	258.33	2917.50	2997.29	2616.66
20_20_jj_right	4	4	44333.882	43788.52	50593.49				221.0	211.7	211.0	210.0	201.1	225.2	258.33	2917.50	2997.29	2616.66
25_11_jj_left	14	-14	44672.784	40938.92	49153.18	89631.7	82373.3	100067.8	217.0	205.0	204.4	202.0	194.5	221.9	280.59	3130.49	3406.33	2804.01
25_11_jj_right	14	-14	44958.882	41434.41	50914.62				215.0	207.7	209.7	201.0	194.5	226.2	280.59	3130.49	3406.33	2804.01
25_25_jj_left	14	14	40413.348	40129.6	46455.46	84504.8	81501.0	94834.7	214.0	203.7	195.8	193.0	189.8	216.0	283.14	3350.61	3474.11	2985.65
25_25_jj_right	14	14	44091.499	41371.4	48379.22				220.0	204.4	200.4	197.0	195.1	220.3	283.14	3350.61	3474.11	2985.65
30_30_jj_left	24	24	39622.576	36998.51	48695.98	79042.6	74738.9	92294.0	214.0	199.1	186.5	183.0	178.9	223.2	134.57	1702.56	1800.60	1458.11
30_30_jj_right	24	24	39420.014	37740.35	43598.04				213.0	201.1	187.2	184.0	178.6	210.0	134.57	1702.56	1800.60	1458.11
30_6_jj_left	24	-24	40057.999	38266.58	45325.36	79968.4	75917.9	92872.4	204.0	195.8	197.1	191.0	187.2	213.0	260.71	3260.20	3434.15	2807.22
30_6_jj_right	24	-24	39910.405	37651.3	47547.06				204.0	194.5	198.4	193.0	187.2	218.6	260.71	3260.20	3434.15	2807.22
35_1_jj_left	34	-34	36340.028	37616.81	42087.3	73567.3	73026.7	86330.2	190.0	195.8	193.8	189.0	182.6	205.7	257.42	3499.04	3524.95	2981.75
35_1_jj_right	34	-34	37227.32	35409.93	44242.94				193.0	184.5	194.5	188.0	185.2	210.3	257.42	3499.04	3524.95	2981.75
35_35_jj_left	34	34	-	32069.59	51668.9	-1.0	62911.5	52005.4	-1.0	184.5	191.2	-1.0	160.7	230.8	-			
35_35_jj_right	34	34	-	30841.96	336.4499				-1.0	186.5	174.0	-1.0	162.7	83.8	-			
5_31_jj_left	-26	26	34543.369	35001.9	44043.87	70817.6	69266.0	44410.9	196.0	186.5	181.2	177.0	174.0	211.7	192.29	2715.35	2776.17	4329.89
5_31_jj_right	-26	26	36274.238	34264.15	367.076				195.0	181.2	186.5	179.0	178.6	92.9	192.29	2715.35	2776.17	4329.89
5_5_jj_left	-26	-26	36871.105	36450.21	46292.7	75599.9	74331.4	94812.4	190.0	183.2	202.4	197.0	191.8	215.3	192.30	2543.60	2587.00	2028.17
5_5_jj_right	-26	-26	38728.792	37881.21	48519.66				193.0	185.2	204.4	202.0	196.5	220.3	192.30	2543.60	2587.00	2028.17

Jason	Chip coodinates (mm)		Overlap area (nm^2)			Total Area (nm^2)			Bottom vertical width (nm)			Top horizontal width(nm)			Conductance	Conductanance/area(µS/µm^2)		
Image name	x	y	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	(µS)	ImageJ	Threshold	Segment
D1_left	4.25	-13.35	23095	26361.2	27804.98	46107.3	51817.8	56899.5	139.0	125.0	165.7	167.0	157.8	163.7	105.2	2281.6	2030.1	1848.8
D1_right	4.25	-13.35	23012.33	25456.55	29094.57				142.0	130.0	170.2	161.0	151.8	161.7	105.2	2281.6	2030.1	1848.8
D2_left	1.95	-15.65	23494.58	34833.34	3463.194	45797.3	58604.1	28570.9	144.0	128.0	85.8	158.0	304.6	20.8	99.4	2170.5	1696.2	3479.2
D2_right	1.95	-15.65	22302.73	23770.78	25107.66				138.0	127.0	157.3	158.0	142.9	155.8	99.4	2170.5	1696.2	3479.2
D3_left	-0.35	-17.95	24158.91	35990.6	26160.99	47582.6	60045.8	52215.7	146.0	129.0	160.7	160.0	239.1	159.7	107.0	2248.7	1781.9	2049.1
D3_right	-0.35	-17.95	23423.72	24055.25	26054.67				145.0	133.9	160.2	156.0	148.8	159.7	107.0	2248.7	1781.9	2049.1
D4_left	1.95	-11.05	45405.6	42738.26	48662.89	89877.2	85779.0	95851.1	213.0	261.9	230.2	153.0	148.8	165.7	185.8	2067.3	2166.1	1938.5
D4_right	1.95	-11.05	44471.61	43040.71	47188.23				215.0	265.9	228.2	154.0	149.8	156.8	185.8	2067.3	2166.1	1938.5
D5_left	-0.35	-13.35	44278.71	43349.04	48774.13	89474.7	88545.4	97322.8	220.0	265.9	231.2	160.0	145.9	161.7	193.0	2157.2	2179.8	1983.2
D5_right	-0.35	-13.35	45195.97	45196.36	48548.7				219.0	267.9	230.7	159.0	151.8	161.7	193.0	2157.2	2179.8	1983.2
D6_left	-2.65	-15.65	45280.61	46938.51	49498.66	90535.6	92146.1	98373.2	215.0	265.9	231.7	150.0	156.8	166.7	197.3	2179.4	2141.3	2005.8
D6_right	-2.65	-15.65	45255.02	45207.6	48874.54				222.0	269.9	231.2	154.0	151.8	162.7	197.3	2179.4	2141.3	2005.8
D7_left	4.25	-15.65	23842	26586.5	28424.18	46065.0	52277.6	55286.1	145.0	134.9	167.7	156.0	153.8	162.7	108.0	2344.6	2066.0	1953.5
D7_right	4.25	-15.65	22223.01	25691.11	26861.9				148.0	137.9	162.7	160.0	148.8	165.7	108.0	2344.6	2066.0	1953.5
D8_left	-0.35	-15.65	24831.11	26836.22	29193.99	49854.1	52942.4	57491.2	147.0	133.0	170.2	160.0	155.8	162.7	107.7	2160.3	2034.3	1873.3
D8_right	-0.35	-15.65	25023.03	26106.21	28297.19				152.0	140.9	167.7	155.0	148.8	160.7	107.7	2160.3	2034.3	1873.3
D9_left	-0.35	-11.05	23518.2	25801.39	27163.13	47813.9	51805.8	55403.2	146.0	138.9	164.2	150.0	144.9	153.8	106.2	2221.2	2050.0	1916.9
D9_right	-0.35	-11.05	24295.71	26004.36	28240.09				146.0	139.9	167.2	165.0	151.8	157.8	106.2	2221.2	2050.0	1916.9
X1_left	-0.35	-8.75	35264.51	33533.06	37138.27	70906.0	67405.3	76177.5	278.0	206.4	196.5	154.0	140.9	153.8	153.8	2169.1	2281.7	2019.0
X1_right	-0.35	-8.75	35641.46	33872.28	39039.19				275.0	206.4	201.9	157.0	145.9	157.8	153.8	2169.1	2281.7	2019.0
X2_left	-2.65	-11.05	35961.32	34567.05	38337.3	72137.2	68637.4	77883.5	274.0	209.3	198.9	150.0	145.9	160.7	149.9	2078.0	2184.0	1924.7
X2_right	-2.65	-11.05	36175.87	34070.39	39546.17				275.0	210.3	202.9	154.0	142.9	160.7	149.9	2078.0	2184.0	1924.7
X3_left	-4.95	-13.35	34057.89	35440.31	38797.02	70958.1	73996.6	79303.0	276.5	208.4	200.9	157.0	149.8	158.7	155.3	2188.7	2098.8	1958.4
X3_right	-4.95	-13.35	36900.24	38556.33	40505.98				277.0	217.3	205.9	159.0	153.8	158.7	155.3	2188.7	2098.8	1958.4
X4_left	-4.95	-11.05	37292.93	34542.03	38692.67	72606.6	69182.9	77289.9	225.0	216.3	200.9	160.0	141.4	157.8	153.7	2116.9	2221.7	1988.7
X4_right	-4.95	-11.05	35313.72	34640.85	38597.18				222.0	216.3	201.4	157.0	141.9	155.8	153.7	2116.9	2221.7	1988.7
Z1_left	1.95	-13.35	35193.65	34750.68	36975.84	70289.9	66962.3	73319.7	213.0	207.4	196.0	155.0	146.8	152.8	161.0	2290.6	2404.4	2195.9
Z1_right	1.95	-13.35	35096.21	32211.66	36343.84				213.0	205.4	193.5	156.0	128.5	157.8	161.0	2290.6	2404.4	2195.9
Z2_left	-2.65	-17.95	36758.51	36302.61	36718.91	73287.7	69703.1	74468.5	217.0	212.3	194.5	161.0	157.8	159.7	158.4	2161.4	2272.5	2127.1
Z2_right	-2.65	-17.95	36529.2	33400.47	37749.6				225.0	211.3	197.9	160.0	138.9	156.8	158.4	2161.4	2272.5	2127.1
Z3_left	1.95	-8.75	35598.15	34267.72	37047.71	71093.9	68023.4	73771.5	215.0	206.4	195.0	158.0	148.8	158.7	155.0	2180.1	2278.5	2101.0
Z3_right	1.95	-8.75	35495.8	33755.69	36723.83				219.0	208.4	194.5	154.0	143.9	159.7	155.0	2180.1	2278.5	2101.0
Z4_left	-2.65	-13.35	36651.24	35618.88	40020.66	72164.7	69368.1	77706.3	221.0	210.3	204.4	157.0	151.8	160.7	152.1	2107.6	2192.5	1957.3
Z4_right	-2.65	-13.35	35513.51	33749.23	37685.61				219.0	209.3	197.9	159.0	144.9	158.7	152.1	2107.6	2192.5	1957.3

Nighthawk	Chip coodinates (mm)		Overlap area (nm²)			Total Area (nm²)			Bottom vertical width (nm)			Top horizontal width(nm)			Conductance	Conductanance/area(µS/µm²)		
Image name	x	y	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	ImageJ	Threshold	Segment	(µS)	ImageJ	Threshold	Segment
10_10	-16.00	-16.00	43714.6	44240.3	52507.7	87429.2	88480.6	105015.3	199.0	204.7	228.2	203.0	200.2	222.8	194.4	2223.6	2197.2	1851.3
10_26	-16.00	16.00	42459.5	40940.0	47666.1	84918.9	81880.0	95332.3	199.0	193.9	218.3	207.0	200.2	204.7	199.2	2345.3	2432.3	2089.1
14_13	-8.00	-10.00	46391.4	45740.9	55976.7	92782.8	91481.7	111953.5	205.0	211.5	235.9	215.0	203.8	227.3	193.4	2083.9	2113.5	1727.1
15_15	-6.00	-6.00	46154.4	45646.5	53515.7	92308.9	91293.0	107031.4	203.0	204.7	230.5	217.0	207.5	222.8	190.4	2062.5	2085.5	1778.8
15_21	-6.00	6.00	48014.5	47677.8	55599.2	96028.9	95355.7	111198.5	208.0	206.6	235.9	211.0	212.9	221.9	199.1	2073.5	2088.2	1790.7
17_17	-2.00	2.00	45537.4	45720.1	55013.5	91074.8	91440.2	110026.9	208.0	205.7	233.8	218.0	211.1	232.7	190.5	2091.6	2083.2	1731.3
18_17	0.00	2.00	46055.0	46363.9	57551.8	92109.9	92727.8	115103.6	209.0	207.5	239.9	217.0	206.6	229.1	191.5	2078.7	2064.9	1663.5
18_18	0.00	0.00	45465.4	44566.6	53762.2	90930.9	89133.2	107524.4	209.0	206.6	231.8	217.0	207.5	220.1	200.3	2202.2	2246.6	1862.4
20_16	4.00	-4.00	45902.9	45582.1	55100.5	91805.9	91164.1	110201.0	206.0	204.7	234.5	217.0	207.5	218.3	199.1	2169.1	2184.4	1807.1
20_20	4.00	4.00	45655.5	46315.9	53161.8	91310.9	92631.7	106323.6	206.0	202.9	230.5	217.0	212.9	216.5	199.5	2185.2	2154.0	1876.6
25_11	24.00	-14.00	44616.3	18285.7	51413.4	89232.6	36571.3	102826.8	206.0	202.0	225.5	216.0	204.7	226.4	185.5	2078.8	5072.3	1804.0
25_25	14.00	14.00	43895.7	42550.4	49158.2	87791.5	85100.8	98316.4	198.0	194.8	221.9	216.0	205.7	206.6	183.3	2088.4	2154.4	1864.8
26_22	16.00	8.00	43771.2	5684.6	52578.5	87542.4	11369.2	105156.9	199.0	201.6	228.7	218.0	165.1	216.5	188.3	2150.4	16557.9	1790.2
27_1	18.00	-34.00	43281.9	24031.2	51701.4	86563.8	48062.3	103402.9	196.0	201.1	226.4	216.0	99.2	230.0	168.0	1940.4	3494.8	1624.4
27_7	18.00	-24.00	42093.1	18526.4	52678.5	84186.2	37052.8	105357.0	196.0	195.7	227.3	212.0	201.1	234.5	178.5	2119.9	4816.6	1694.0
2_35	-32.00	34.00	34470.3	36688.3	40836.2	68940.5	73376.5	81672.5	182.0	180.4	202.5	193.0	183.1	186.7	177.4	2573.1	2417.5	2172.0
30_30	24.00	24.00	40797.6	35746.1	47069.0	81595.2	71492.3	94137.9	197.0	183.1	216.9	214.0	185.8	198.4	-			
30_6	24.00	-24.00	41888.5	13660.0	50095.4	83777.0	27320.0	100190.9	200.0	37.9	222.3	208.0	259.8	230.9	-			
32_2	28.00	-32.00	40090.0	15210.7	57586.0	80179.9	30421.4	115171.9	196.0	212.9	239.5	203.0	222.8	230.9	152.6	1903.0	5015.5	1324.8
34_34	32.00	32.00	35997.9	35081.0	43843.2	71995.8	70162.1	87686.4	183.0	180.4	209.7	201.0	184.9	193.9	179.0	2486.3	2551.2	2041.4
35_1	34.00	-34.00	40201.6	42983.2	50890.3	80403.1	85966.5	101780.6	197.0	199.3	224.6	201.0	199.3	223.7	150.6	1872.6	1751.4	1479.3
35_35	34.00	34.00	33613.8	38426.4	40387.1	67227.7	76852.9	80774.3	178.0	173.2	201.1	198.0	202.9	182.2	153.7	2286.0	1999.7	1902.6
5_31	-26.00	26.00	38631.9	37120.7	46809.4	77263.7	74241.4	93618.9	193.0	189.4	216.0	200.0	189.4	201.1	189.4	2451.6	2551.4	2023.3
5_5	-26.00	-26.00	40427.2	15593.6	46811.1	80854.4	31187.3	93622.1	199.0	205.7	215.1	199.0	183.1	214.7	179.7	2222.4	5761.6	1919.3
7_7	-24.00	-24.00	40152.2	44617.6	49761.9	80304.5	89235.3	99523.8	200.0	206.6	223.2	200.0	201.1	212.0	184.8	2301.6	2071.3	1857.1

.3.3 Images

Below the SEM images where the overlap area is marked red. Four red lines are drawn where the edges have been found for the thresholding method. The next page contains the segmentation method.



(a) Blackbird 1.

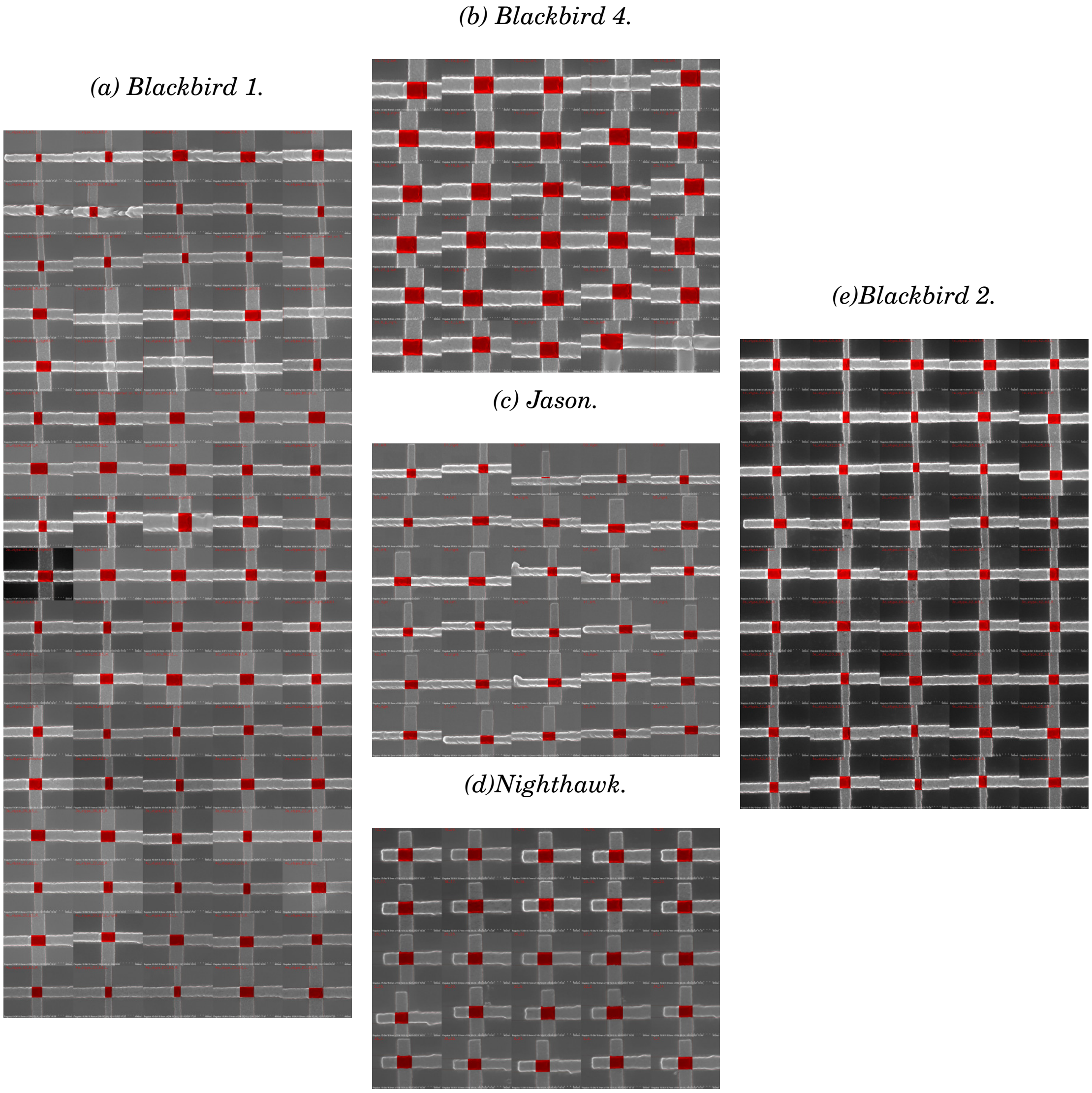
(b) Blackbird 4.

(c) Jason.

(d)Nighthawk.

(e)Blackbird 2.

Figure 15: All the junctions measured with the thresholding method stitched together for each data set. Any remeasurement with the GUI is represented next to the original measurement in the stitched image.



(a) Blackbird 1.

(b) Blackbird 4.

(c) Jason.

(d)Nighthawk.

(e)Blackbird 2.

Figure 16: All the junctions measured with the segmentation method stitched together for each data set.