



POSITIONED INAs QUANTUM DOTS FROM SELECTIVE AREA GROWTH

Manufacturing, Characterization and Optical Measurements

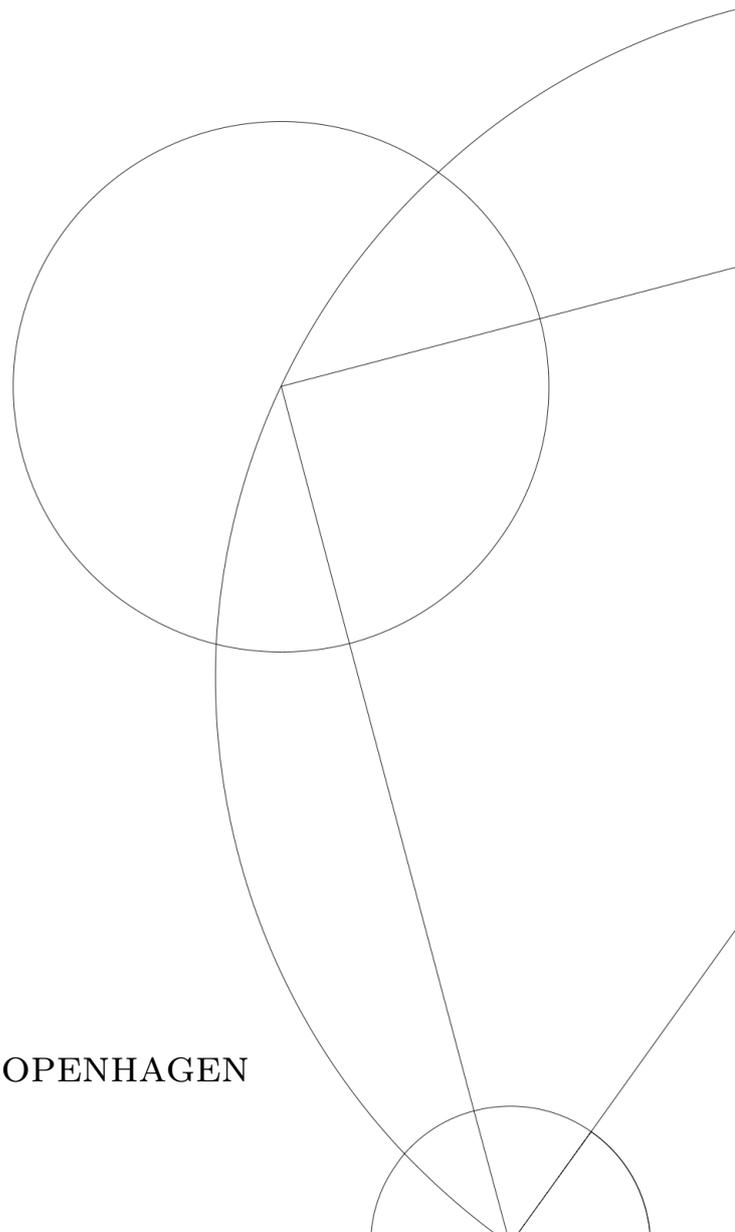
MASTER'S THESIS

Written by *Laurits Høgel*

Supervised by

Peter Lodahl and Peter Krogstrup

UNIVERSITY OF COPENHAGEN





UNIVERSITY OF
COPENHAGEN

NAME OF INSTITUTE: Niels Bohr Institute

NAME OF DEPARTMENT: Center for Quantum Devices and HyQ

AUTHOR(S): Laurits Høgel

EMAIL: qgh880@alumni.ku.dk

TITLE AND SUBTITLE: Positioned InAs Quantum Dots
from Selective Area Growth
- Manufacturing, Characterization and Optical Measurements

SUPERVISOR(S): Peter Lodahl and Peter Krogstrup

HANDED IN: 2021-10-15

DEFENDED: 2021-11-04

NAME _____

SIGNATURE _____

DATE _____

Abstract

Indium-Arsenide quantum dots are widely used in platforms for quantum photonics and serve as single photon sources and non-linear components. Commonly, Stranski-Krastanov quantum dots are used, but using instead MBE Selective Area Growth, the quantum dots can be positioned and tuned deterministically.

InAs quantum dots were grown on buffered GaAs (100) substrates, and characterized through AFM, SEM, TEM and PL. In parallel, simulations were done on MBE adatom dynamics, equilibrium crystal shapes, interface strain, and electronic confinement energy. It was observed that the GaAs buffer growth is highly dependent on a proximity effect. The InAs growth occurs exclusively on top facets. The task remains to grow elastically strained InAs quantum dots which requires utilisation of the structures near the lower MBE limit for spatial dimensions.

While the optimal calibration is yet to be reached, the proof of concept is established that buffered SAG can provide a deterministic path for InAs quantum dot manufacturing.

Resumé

Indium-Arsenid kvantepunkter bruges bredt i kvantefotonik-platforme og virker som enkelt-foton kilder og non-lineære komponenter. Normalvis bruges Stranski-Krastanov kvantepunkter, men ved at bruge MBE regio-selektiv krystalvækst, kan kvantepunkterne positioneres og frekvens-indstilles deterministisk.

InAs kvantepunkter er blevet groet på bufferet GaAs (100) substrater, og karakteriseret igennem AFM, SEM, TEM og PL. Sideløbende er simuleringer blevet kørt på MBE adatom densitet dynamik, ækvilibrium krystal former, grænseflade spænding og elektronisk begrænsnings energi. Det er observeret at GaAs buffer krystalvækst afhænger kraftigt af en nærhedseffekt. InAs væksten forekommer udelukkende på topfacetter. Tilbage står opgaven at gro elastisk spændte InAs kvantepunkter, hvilket kræver anvendelse af strukturer nær den nedre størrelsesgrænse for MBE.

Mens den optimale kalibrering stadig ikke er nået, er beviset på konceptetsuelighed etableret, at regio-selektiv krystalvækst kan give en deterministisk metode at fremstille InAs på.

Contents

1	Introduction	4
1.1	Abbreviations and Symbols	7
2	MBE and Selective Area Growth	8
2.1	Thermodynamics of SAG MBE	9
2.2	Growth Parameters	11
2.3	Adatom Dynamics	11
2.3.1	Adatom Density Simluations	12
2.4	Wulff Construction	14
2.4.1	Constrained Wulff Construction	15
2.5	Lattice Mismatch and Solid Mechanics	18
3	Quantum Dots and Electronic States	21
3.1	0-Dimensional Confinement	21
3.2	Spectrum Simulation	22
4	AFM Analysis, Tools and Scripts	26
4.1	AFM Data and Calibration	26
4.2	Structure Finding Algorithm	28
4.3	Extraction of Geometrical Values	28
4.4	Facet Analysis	30
4.5	SEM Analysis	33
5	Results and Discussion	35
5.1	Designs of Growth	35
5.2	Fabrication Process	35
5.3	Dimensions and Incorporation Rates of GaAs Growth	36
5.4	Kinetic and Thermodynamic Growth	40
5.5	Facet progression	40
5.6	InAs Growth, SEM and TEM	45
5.6.1	SEM	45
5.6.2	EDX	45
5.6.3	TEM	48
5.6.4	GPA	48

5.6.5	Electron Energy Loss Spectroscopy	50
5.7	Photoluminescence Experiments	50
6	Conclusion	56
6.1	Acknowledgements	56
	References	58
7	Appendix	61
7.1	Design	61
7.2	Fabrication Process	61
7.2.1	Spin Coating	61
7.2.2	Electron Beam Lithography	61
7.2.3	Development	61
7.2.4	HF Etching	64
7.2.5	Strip-off	64
7.3	List of Samples	64
7.3.1	QDev1072	65
7.3.2	QDev1099	65
7.3.3	QDev1104	65
7.3.4	QDev1121	67
7.3.5	QDev1132	67
7.3.6	QDev1142	67
7.4	Python Scripts	69

1 Introduction

The advent of quantum technology is approaching, as more and more utilizations seem within reach. Technologies like quantum internet, quantum key distribution and quantum computing are becoming still more embedded in our understanding of informatics of the future. But the enormous potential of encoding information into quantum states is limited by the manufacturing of the respective platforms[1].

Quantum dots (QDs) are small crystals of metal or semiconductor. Their dimensions are such, that the electronic wave function is confined in all three dimensions, i.e. a zero-dimensional quantum well. The band structure of the material turns into discrete energy levels, hence quantum dots are sometimes dubbed *artificial atoms*[2]. Like atoms, quantum dots have a spectrum and can absorb and emit photons. This makes them invaluable in quantum optics, as they serve as single photon sources and can be incorporated into large networks of many photon setups [3, 4]. For applications like quantum repeaters, the perfection of emitters is key, as many QDs must emit in the same frequency spectrum [5]. QDs also factor into the topological qubit hexon units, where the zero-modes couple to the state of an adjacent quantum dot[6]. The more intricate the system of interest, the more important the exact spectrum and coupling to surroundings become. This underlines the necessity for reproducible positioned QDs.

Quantum dots have very interesting electronic properties. Due to their discrete energy levels, by means of a gate, a QD can be tuned to conduct current or act as an insulator. This is in part due to the concept of *Coulomb Blockade*, where the adding of a single electron is blocked by the Coulomb repulsion of the electrons in the QD [7].

InAs quantum dots (InAs QDs) have been studied extensively for the last decades[8]. The 7.2 % lattice mismatch between GaAs and InAs allows for only a wetting layer of epitaxial growth, before the InAs self-assembles into *islands* or quantum dots. This is called Stranski-Krastanov Growth[9]. The downside of self-assembled QDs is that they cannot be positioned exactly and their sizes will have a stochastic distribution within some range.

Positioned InAs QDs on GaAs has also been studied [10, 11], but are not yet the common setup in quantum optical experiments, as the self-assembled QDs are easier to come by and are more readily integrated in devices. However, advances in positioned InAs QDs from *selective area growth* (SAG) might change the standard and open new doors in quantum optical experiments and eventually in our information infrastructure.

This work aims to grow and characterize positioned and buffered InAs QDs through

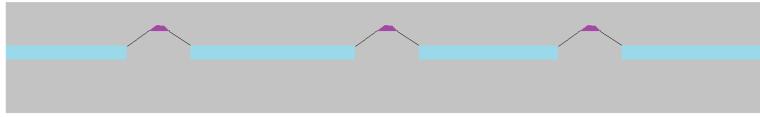


Figure 1: A cross section of the InAs QDs (purple) grown on and capped with GaAs (grey) with the remaining SAG SiO_x mask (light blue)

means of SAG growth in arrays with various dimensions. The buffer structures are the GaAs pedestals on which the InAs QD are grown. Their role is both to define a small growth site, and to accommodate elastic strain.

The overall scope will be to assess the viability and reproducibility of this method. To test how InAs QDs grow on GaAs buffer structures with large enough pitch to allow for single QD spectral measurements. To do so, this project combines experimental work and data analysis with simulations of crystal growth and QD photoluminescence measurements. The experimental part of the project contains several steps, and each is required to proceed. The specific tasks are as follows:

- Pregrowth fabrication of SiO_x masks on GaAs(100) samples and calibration of each step.
- MBE growth carried out by certified MBE user, postdoc Daria Besnasiuk.
- Characterization of the growth with AFM and SEM.
- Optimisation of GaAs buffer structure growth
- Subsequent InAs QD growth.
- Characterization of the QDs through SEM and TEM with help from the Zaragoza group.
- Growth of a capped sample to make the InAs QDs optically active.
- Measurements of photoluminescence of the InAs QDs.
- Connection between the growth to the observations from the photoluminescence.

This project aims to elucidate and increase the insight into the key aspects of InAs QD manufacturing and open up whole new line of experiments with spectrally and spatially coordinated QD systems.

1.1 Abbreviations and Symbols

AFM	Atomic Force Microscopy
EBL	Electron Beam Lithography
EELS	Electron Energy Loss Spectroscopy
EDX	Energy Dispersive X-Ray Spectroscopy
ECS	Equilibrium Crystal Shape
ERS	Equilibrium Reference State
GPA	Geometric Phase Analysis
HF	Hydrofluoric Acid
MBE	Molecular Beam Epitaxy
PL	Photoluminescence
QD	Quantum Dot
SAG	Selective Area Growth
SED	Surface Energy Density / Strain Energy Density
SEM	Scanning Electron Microscope
SK	Stranski-Krastanov
TEM	Tunneling Electron Microscope

2 MBE and Selective Area Growth

Molecular Beam Epitaxy (MBE) is a method of controlled crystal growth, that allows for bottom-up construction of structures in the nanometer-scale. MBE is one of a wide variety of similar means of carrying out crystal growth from an incident beam of atoms or molecules.

The basic principle of MBE growth is as such: A substrate is placed in a vacuum chamber while sources of pure elements are heated until they sublimate a flux of atoms and molecules onto the substrate. Some of the impinging atoms from the beam become adatoms on the surface and has a chance of incorporating into the substrate. The substrate is kept at a high temperature so that the incorporation is not immediate, but the adatom phase allows for some minimization of free energy. Hence the growth is said to be epitaxial, as the crystal structure of the substrate is continued layer by layer in the growth. This is the main difference between epitaxial growth and deposition, where the product is amorphous.

A particular type of material, studied extensively for MBE is III-V compounds, that is consisting of a group III and a group V elements of the periodic table. Examples are, GaAs, InAs, InP, GaP and ternary compounds within each group e.g. $\text{In}_{1-x}\text{Ga}_x\text{As}$

Selective area growth (SAG), is regioselective MBE growth defined by a mask on the substrate. By depositing and locally etching an oxide layer onto a substrate, it is possible to define precise locations and shapes for the epitaxial crystal growth as parasitic growth on the oxide is either completely or substantially suppressed. This is known as the selectivity window, a range of substrate temperature and beam flux, that allows for

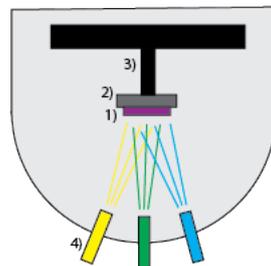


Figure 2: Schematic of an MBE growth chamber. 1) the substrate 2) sample holder 3) rotating arm 4) effusion cells



Figure 3: Different growth modes a) Frank-van der Merwe growth, layer-by-layer growth b) Stranski-Krastanov growth consisting of a wetting layer and island formation c) Volmer-Weber growth, only island formation

growth only on openings in the mask.

2.1 Thermodynamics of SAG MBE

Crystal growth in MBE is theoretically described by differential equations of adatoms and by phase transitions rates related to chemical potentials. This section is based on work by Peter Krogstrup et al.[12] which is used to describe the dynamics for vapor-liquid-solid growth.

First it is necessary to define a chemical potential for a given element i in phase p , $\mu_{p,i}$ and compare it to a equilibrium reference state (ERS) μ_i^{ERS} :

$$\delta\mu_{p-ERS,i} = \mu_{p,i} - \mu_i^{ERS} \quad (1)$$

where at the reference state, the two main phases are in equilibrium. This is necessary because the phases themselves change chemical potential with the changing morphology of the growth.

Together with the transition state energy $\delta g_{pq,i}^{TS,ERS}$, we can define an Arrhenius behaviour of a phase transition rate between the different phases p and q :

$$\Gamma_{pq,i} = \Xi_{pq,i} c_{p,i} \exp\left(-\frac{\delta g_{pq,i}^{TS,ERS} - \delta\mu_{p-ERS,i}}{k_b T}\right) \quad (2)$$

where Ξ is the *single atom prefactor*, c is the density of the initial adatom phase, and $\delta g_{pq,i}^{TS,ERS} - \delta\mu_{p-ERS,i}$ is the energy barrier. This treatment allows for incorporation of variables such as the shape of growth (growth front), with areas, angles etc through comparison to the ERS for the solid phase, s :

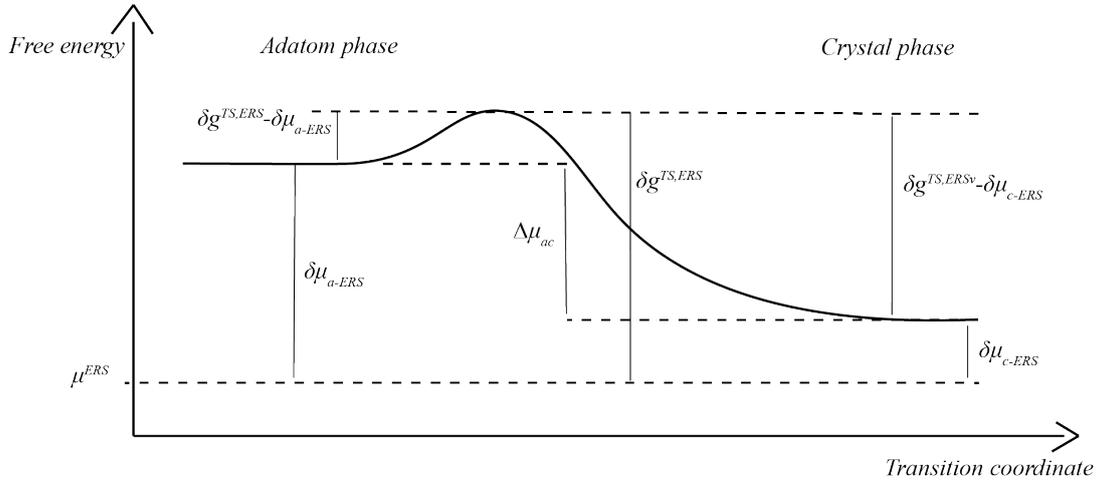


Figure 4: Transition diagram between adatom and crystal phase. All potentials are reported compared to an equilibrium reference state

$$\delta\mu_{s-ERS,III-V}^X = \sum_j \gamma_j \frac{\partial A_j}{\partial X} \frac{\partial X}{\partial N_{s,III-V}} + \Delta\epsilon_s \quad (3)$$

where γ_j is the surface energy density of the j 'th facet, A_j is a facet area, X is any geometric variable such as displacement or angle, $N_{s,III-V}$ is the amount of III-V pairs incorporated, and Δ_s is the added strain energy.

For the SAG growth, the first geometrical contributions to take into account are the mask-opening boundaries, and how the adatoms diffuse across. While the incorporation rate locally follows:

$$\Gamma_c \propto \rho \cdot \exp\left(\frac{\delta g}{kT}\right) \quad (4)$$

It has been reported [17] that the incorporation rate for long III/V nanowires follows the relation:

$$\Delta\Gamma = f_i + \frac{\Delta\Gamma_{a_m a_c}}{w} + \Delta\Gamma_{a_c v} \quad (5)$$

where f_i is the flux, $\Delta\Gamma_{a_m a_c}$ is the net adatom exchange between opening and mask, w is the nanowire width, and $\Delta\Gamma_{a_c v}$ is the evaporation of adatoms from the opening. The relation is based on mass conservation, and includes all the transitions an adatom at an opening can take. The effective boundary flux, inversely proportional to the width of

the nanowire, is even more detrimental for QDs, that have much higher circumference-to-area ratio. Here, the total growth can be written as:

$$I_{ac} = (f_i - \Gamma_{av}) \cdot \pi r^2 + (\Gamma_{ama_c} - \Gamma_{aca_m}) \cdot 2\pi r \quad (6)$$

so dividing by area gives the incorporation rate:

$$\Gamma_{ac} = f_i - \Gamma_{av} + (\Gamma_{ama_c} - \Gamma_{aca_m}) \frac{2}{r} \quad (7)$$

The consequences of this will be discussed further in Section 5.

2.2 Growth Parameters

The main parameters of MBE growth are substrate temperature, fluxes of the constituent elements, and the growth time. In GaAs, and III-V growth in general, a common setting is to have a saturation of As, i.e. the the V/III ratio is well above 1. Although the stoichiometry is 1:1, the higher ratio makes the growth more stable and makes it only depend effectively on the local concentration of one of the constituents. Therefore, fluxes are often reported in V/III ratio and nominal growth rate. The nominal growth rate is the 2D growth rate, i.e. the rate of growth when there is no mask and the whole chip acts as a growth site. Naturally, the growth rate differs from the nominal rate in intricate designs like in this project, since the adatoms are exchanged between the mask openings and the mask itself.

Finally, a growth surfactant can be used. A surfactant is a chemical species that does not incorporate as crystal growth, but modifies the chemical potentials and rates of the growth. It greatly affects the nucleation criteria and rate. A lot of effort has gone into understanding and modelling MBE growth with surfactants, and is still an active topic of research. The efforts ranges from thermodynamical analytical models to random walk Monte Carlo Simulations [14, 15].

2.3 Adatom Dynamics

Since the growth rate at any given point depends on the local adatom density, it is important to know how the adatoms diffuse on the sample. The sample is subjected to an approximately uniform flux of vapor phase atoms, but the adatoms will diffuse and evaporate differently from the mask and the openings. Therefore, the sizes of grown structures and what surrounds any given structure and with what distance are important.

The adatom density dynamics can be described by following equation:

$$\frac{\partial \rho}{\partial t} = F + \nabla \cdot (D \nabla \rho) - \Gamma_{ac} - \Gamma_{av} \quad (8)$$

where ρ is the adatom density, F is the flux. The second term is the Fickian diffusion, with $D \propto \lambda_L^2$, D and λ_L being the diffusion constant and the diffusion length respectively. The third term accounts for the adatoms being incorporated into the crystal phase, and the fourth term is the adatoms evaporating.

2.3.1 Adatom Density Simulations

The adatom density plays an important role in estimating an incorporation rate. For step-flow growth, incorporation rate is proportional to the adatom density, whereas for nucleation limited growth, it is proportional to the square of the density[16].

A publicly available *Matlab* script originally made for heat diffusion calculations, and then converted by Tobias Særkjær [17] to describe adatom diffusion, is used in this part, output seen in Figure 5. With estimations of diffusion coefficients, flux, and incorporation rate, the script calculates the steady state adatom density on the mask and openings. It solves equation 8 with two sets of constants for the mask and the openings, respectively: D_{SiOx} , $\Gamma_{av,SiOx}$ and D_{GaAs} , $\Gamma_{av,GaAs}$, $\Gamma_{ac,GaAs}$. $\Gamma_{ac,SiOx}=0$ since there is no growth on the mask.

The boundary conditions are as follows: the outer frame of the mask is set to have *Neumann Boundary Conditions*. This means that the derivative must go to zero, since the area represents a quadrant of a whole, thus every edge is a symmetry plane and there is no net flux across.

The boundaries between mask and openings is set to have *Dirichlet Boundary Conditions*. This means that the value at the boundary coming from either side must be equal. The derivative is, however, not required to be continuous over a Dirichlet boundary.

Using estimate values it is simple to reproduce the overall concept of lower densities at outer openings consistent with experiments. Yet, the discrepancy between center and edge openings is too small to describe what we see experimentally. In table 1 some values are reported.

A particular parameter choice, that tends towards the experimental result, is to work with an effective flux on the oxide different from the flux on openings. While the gas phase concentrations remains approximately constant over the length scale of the design,

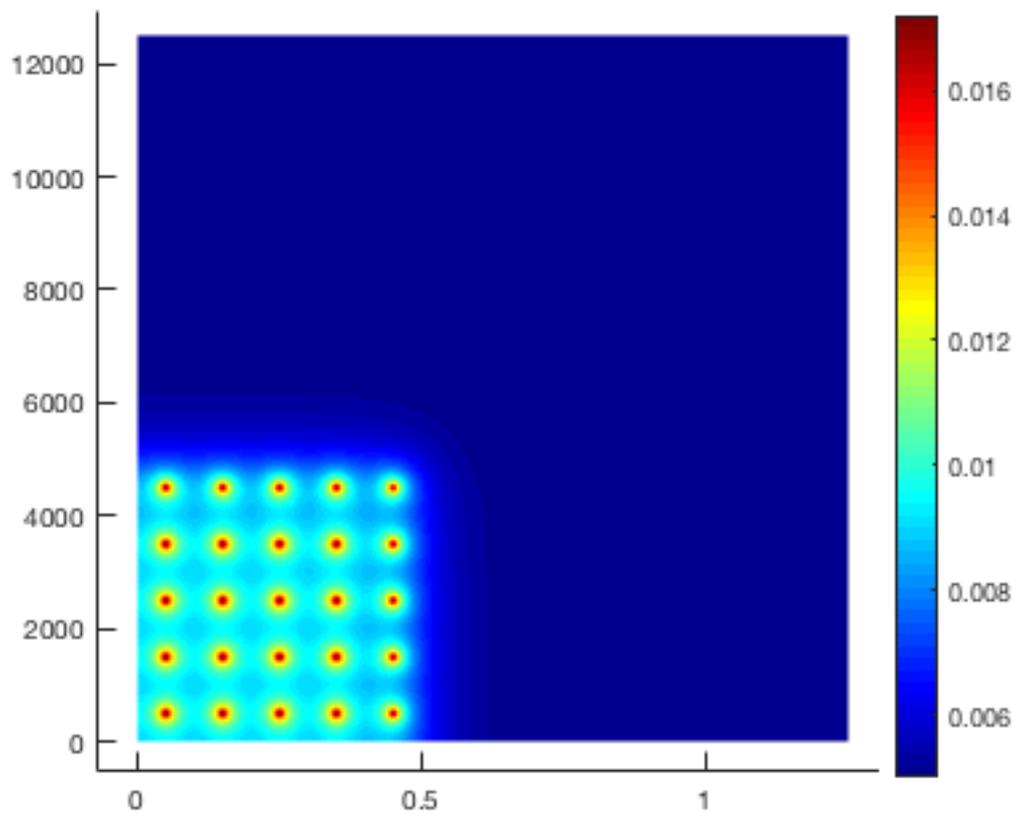


Figure 5: Adatom density at equilibrium shown for an array of buffer structures of 100nm width 750 nm pitch

Simulation of ρ							
#	Center Opening ρ	Corner Opening ρ	Oxide ρ	$\lambda_{SiOx}, \lambda_{GaAs}$	SiO _{av}	$\Gamma_{va,SiOx}$	$\Gamma_{va,GaAs}$
1	1.85	1.84	1.836	10 μ m	50	100	100
2	0.456	0.453	0.450	15 μ m	250	100	100
3	0.056	0.055	0.01	5 μ m	1000	1	100
4	0.0163	0.0150	0.006	15 μ m	1000	1	100

Table 1: Calculated adatom densities from four simulation. The input parameters that are varied are shown to the right, and the averaged densities to the left

the two fluxes could be justified by a slower rate $\Gamma_{va,oxide} < \Gamma_{va,GaAs}$, which is to say the impinging vapor atoms reflect more easily from the oxide.

The model still needs to be properly fitted to the experimental data, but once it has been, the calculations should be valuable, especially when working with less symmetric designs.

2.4 Wulff Construction

Wulff construction is a mathematical tool to calculate the shape with minimum surface energy of a crystal or a liquid crystal [18]. These shapes are known as the equilibrium crystal shape, ECS. The total surface free energy can be written as:

$$\Delta G_{surface} = \sum_i O_i \gamma_i \quad (9)$$

where O_i is the area of the i 'th surface and γ_i is the associated surface energy density. Wulff's theorem simply states, that for the equilibrium crystal shape, the distance of a facet from the center of the crystal equals the surface energy density of the given facet times a scaling constant.

$$h_i = \lambda \gamma_i \quad (10)$$

This means that the shape is entirely depending on a discrete set of facets and their surface energy densities.

The γ -value can also be generalized for any direction in space. Then the crystal facet orientation will be a local energy minimum in a γ -surface with Lorentzian characteristics. The γ -surface will be of the form:

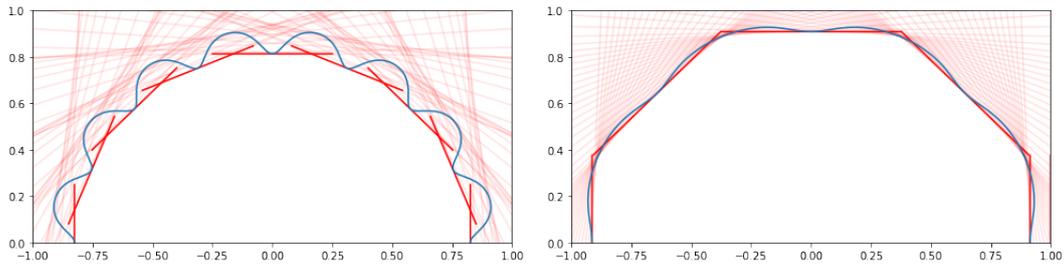


Figure 6: 2D Wulff shapes for two different crystals (both arbitrary). The blue curve is the *gamma*-surface, which is the directional surface energy density. The red lines, perpendicular to the center direction, outline the ECS's

$$\gamma(\theta, \psi) = 1 - \sum_i \frac{1}{\pi} \frac{\delta_i}{(\theta - \theta_i)^2 + (\phi - \phi_i)^2 + \delta_i^2} \quad (11)$$

Where i runs over all facets. In figure 6 two 2D representation of arbitrary γ -surfaces are shown for different amount of facets and different δ values. Two choices for δ are chosen, small and large, respectively. A large δ -value gives shallow dips in the γ -surface and introduces corner rounding.

The calculation becomes more involved when describing shapes with an interface. Winterbottom Construction is the interfaced counter part to Wulff Construction, and uses the ratio between the interface SED and the substrate SED to give a 'wetting angle', the angle between the substrate and the incident shape to minimize energy. This is also known as the *Young angle* and is described by:

$$\cos(\theta) = \frac{\gamma_{substrate} - \gamma_{interface}}{\gamma_{crystal}} \quad (12)$$

Winterbottom construction keeps the relative facet displacements of Wulff Construction. The ratio between $\gamma_{interface}, \gamma_{substrate}, \gamma_{crystal}$ determines the height of the crystal center in reference to the substrate plane. The extremes are: a high $\gamma_{substrate}$ and low $\gamma_{interface}, \gamma_{crystal}$ which leads to thin film formation. Oppositely, low $\gamma_{substrate}$ and high $\gamma_{interface}$ give closed crystal shapes with minimum interface area.

2.4.1 Constrained Wulff Construction

Finally, for constrained interfaces, like in the case for SAG, the ECS is no longer size invariant and the relative facet displacements are no longer bound by the SED ratios,

which was the case for Wulff and Winterbottom Construction. That means for every step of growth, there is a different ECS[19]. The calculation must be solved with numerical optimization. A such calculation is carried out in a *python* script in order to simulate the equilibrium shapes of the buffer structures, for different growth stages and opening shapes.

The script is constructed such, that the variables of the shape is the displacement from a center point for each facet. Then a function loops over all 3-plane-interfaces of facet planes with following equation, using $\mathbf{x}A_i + \mathbf{y}B_i + \mathbf{z}C_i = D_i$ for each plane. a, b, c is the normal vector to the plane, and d is displacement from the origo.

Initially two matrices are constructed; R_c and R_d :

$$R_c = \begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{bmatrix}, R_d = \begin{bmatrix} A_1 & B_1 & C_1 & D_1 \\ A_2 & B_2 & C_2 & D_2 \\ A_3 & B_3 & C_3 & D_3 \end{bmatrix} \quad (13)$$

Then, the rank of R_c must equal 3 and the rank of R_d must equal 4, in order to secure an actual three plane intersection, which is equivalent to the three normal vector spanning the full 3D space.

Then the coordinates of the intersection can be equated:

$$x = \frac{\begin{vmatrix} D_1 & B_1 & C_1 \\ D_2 & B_2 & C_2 \\ D_3 & B_3 & C_3 \end{vmatrix}}{Det}, y = \frac{\begin{vmatrix} A_1 & D_1 & C_1 \\ A_2 & D_2 & C_2 \\ A_3 & D_3 & C_3 \end{vmatrix}}{Det}, z = \frac{\begin{vmatrix} A_1 & B_1 & D_1 \\ A_2 & B_2 & D_2 \\ A_3 & B_3 & D_3 \end{vmatrix}}{Det}, \quad (14)$$

where Det is the determinant of R_c . Next step is then to remove all intersections that are cut of by any other facet plane. This is done by the same principle of simple Wulff Construction.

At this point, the shapes are generated. Now remains to optimize the surface energy. This is done by minimizing the surface energy summing over all facets:

$$E_{surface} = \sum_i A_i \cdot \gamma_i \quad (15)$$

while keeping the volume constant. That is done numerically with *ConvexHull.convex* in python. Then the two functions are set into *scipy.optimize*, a constraint optimizer - surface energy is the function to be optimized, volume is the constraint.

The solution space has dimensionality of the number of facet types plus one, as the top facet is not symmetrically equivalent to the outer {100} facets, and therefore needs

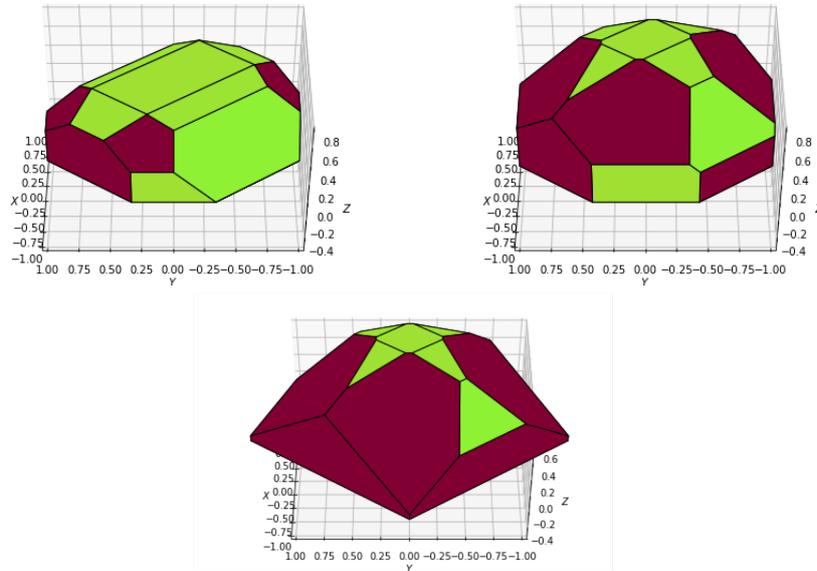


Figure 7: Constrained Wulff shapes. Above) Two different volumes with the same octagonal opening constraint. Below) a square opening shape

its own variable. That gives 7 displacement coefficients for facet types: $\{100\}$, $\{110\}$, $\{111A\}$, $\{111B\}$, $\{113A\}$, $\{113B\}$, and $[001]$. The volume and surface energy depends on the displacement coefficients in a non-linear fashion, with only piece-wise continuous derivative. Therefore, to keep the solutions within the right space, and to avoid elimination of particular facets, a penalty is added if a facet area subceeds 0.001 during optimization iteration. This still allows for facets practically being eliminated, as an area of 0.001 effectively has no contribution on the total surface energy

Also, as an emulation of actual crystal growth, the solution to each frame is set as initial conditions for the next incremental volume step, and the new solution is bound within some coefficient subspace close to the former solution, to avoid 'hopping' between different paths of meta-stable states.

In figure 7 two solutions for the ECS is shown for different volumes. In figure 8 the evolution of the top facet area is shown.

When comparing to the observed morphologies in SEM and AFM, particularly for small structures, the predicted top facet evolution is not in accordance. What the ECS model fails to include is the growth kinetics. For instance, the top facet is perpendicular to the effective flux and therefore growth on the top facet is quicker. There are different options, for modeling the growth kinetics. One is to use a phase-field method as done by

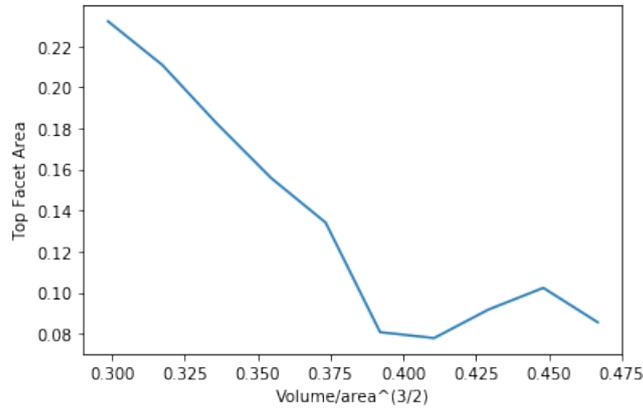


Figure 8: The top facet area as a function of volume for an octagonal opening. Both top face area and volume is normalised with the opening area.

Albani et al [20], where the modeled growth is depending on a multitude of parameters, including a local phase parameter, its gradient, the local crystal orientation and more.

2.5 Lattice Mismatch and Solid Mechanics

The strain in the crystal structures is important for two main reasons. First; an accumulation of strain can lead to plastic deformation and other crystal defects. Second; straining a semiconductor affects the band structure, as linear or planar strain result in symmetry breaking.

When InAs is grown on GaAs, the initial layer (the wetting layer) continues the GaAs lattice. But after some layers, at the critical height, the strain energy becomes high enough that islands are formed. This is known as Stranski-Krastanov growth mode, as seen in figure 3. However, this behaviour depends on the size of the InAs growth site. For small base area structures, like these buffered quantum dots, the strain will gradually decrease for the upper layers of the growth as the lattice return to the original lattice constant. So is the case for the GaAs buffers, strained oppositely, as they are more easily strained than a bare substrate. This means that there may not be any plastic relaxation for InAs QDs as long as the interface remains sufficiently small.

Total strain energy is measured by integrating over the strain energy density:

$$U = \frac{1}{2}(\sigma_x \epsilon_x + \sigma_y \epsilon_y + \sigma_z \epsilon_z + 2\sigma_{xy} \epsilon_{xy} + 2\sigma_{yz} \epsilon_{yz} + 2\sigma_{xz} \epsilon_{xz}) \quad (16)$$

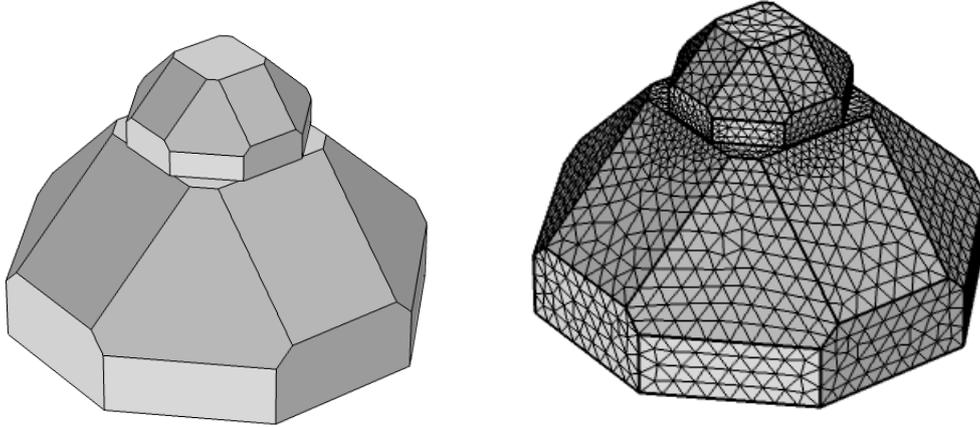


Figure 9: The QD-buffer structure drawn and meshed in COMSOL

where σ is the stress vector and ϵ is the strain vector. The two are related by the formula and inverse formula:

$$\sigma = \mathbf{S}\epsilon \quad \epsilon = \mathbf{C}\sigma \quad (17)$$

where \mathbf{S} is the *stiffness matrix* and its inverse \mathbf{C} is the *compliance matrix*. These matrices can be constructed from Young's Modulus and the Poisson ratio for the given material.

Using COMSOL, the strain energy is calculated for different sizes of QD-buffer structures in order to model the strain energy as a function of size. The 7.2 % lattice mismatch between GaAs, 5.65 Å, and InAs, 6.06 Å, is set as an initial strain in the InAs. That is equivalent to epitaxial growth of InAs on top of GaAs with no deformation. The QD-buffer structure then elastically relaxes until every point is in a force equilibrium, leaving strain centered at the interface. Due to the lower stiffness coefficients of InAs and due to the geometries and size differences, the strain will mainly be present in the QD, while the buffer will show minor deformations.

Figure 9 shows the COMSOL model of the buffer-QD structure. In figure 10 a simulation of the strain energy density of the buffer-QD structure is shown in a slice plot. Values for stiffness matrices from [21] has been used. It is clear how the strain is primarily located around the interface. This suggests that the total strain energy should be normalized with the area to show how the size of a structure affects the local conditions around the interface. In figure 11 the total strain is shown and it is shown that the strain

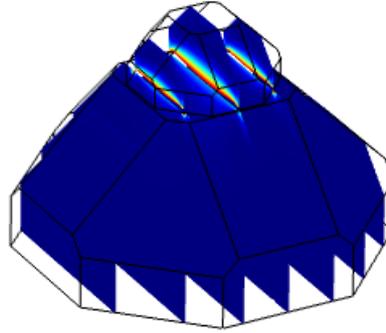


Figure 10: Slice plot showing the strain energy density of the elastically relaxed QD-buffer structure calculated in COMSOL

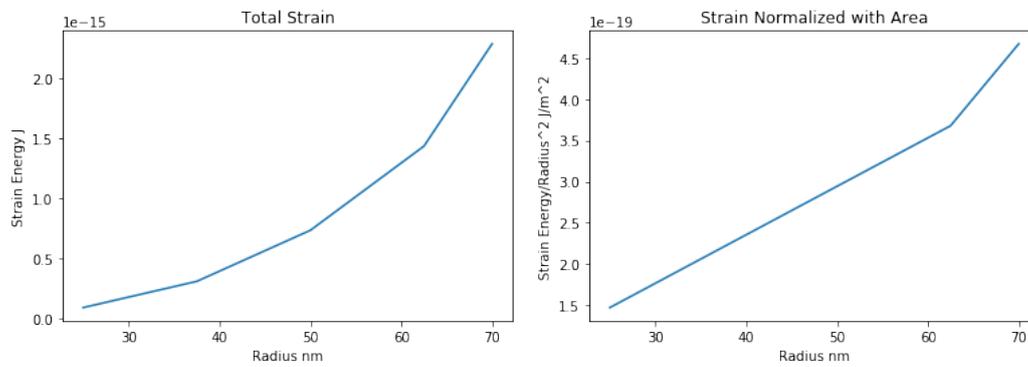


Figure 11: Left) the total strain energy as a function of the radius of the buffer structure. Right) The total strain normalized with the interface area

energy normalized with area is linear in the structure radius. From experimental data this gives an estimate of how much SED can be built up before plastic relaxation.

3 Quantum Dots and Electronic States

InAs QDs are the most common type of quantum dots used in quantum optics. Their spectra makes them excellent for interacting with light ranging from ≈ 900 nm, which is the normal range for optical experiments, towards wavelength of 1550 nm, which is the preferred wavelength for long distance signals, as it has the lowest loss in fiber optic cables.

3.1 0-Dimensional Confinement

We know that InAs has a bandgap of 0.35 eV, for the bulk material. The bandgap wavelength follows:

$$E = hf = hc/\lambda \quad \Rightarrow \quad \lambda = hc/E \quad (18)$$

which gives a bandgap wavelength of 3.8 μm for bulk InAs.

For any structure in the nanometer scale, the confinement will give rise to a further energy level separation. From the particle in a box example, we know that the electrons must have discrete energies to meet the boundary conditions. These energies follow:

$$E_n = \frac{\hbar^2(k_x^2 + k_y^2 + k_z^2)}{2m}$$

and the k_i 's must meet the requirement:

$$k_i = \frac{\pi n}{L_i} \quad n = 1, 2, 3\dots \quad (19)$$

where L_i is the side length of the box. This shows that in the case where one dimension, L_z is noticeably smaller than the other dimensions, the quantization of k_z will set the energy quantization, while k_x and k_y will form a quasi-continuum, that is, a further quantization but within a much smaller energy range.

The excitation energy E_{QD} and recombination energy is

$$E_{QD} = E_{bandgap} + E_{confinement} + E_{exciton} \quad (20)$$

where the second contribution consists of $E_{confinement} = E_{conf,e} + E_{conf,hh}$. The last energy contribution is from the relaxation of the the exciton. It is a negative energy correction, that originates from the relaxation of the excited state and the hole created. It accounts for the orbital relaxation that happens, because a valence state is unoccupied.

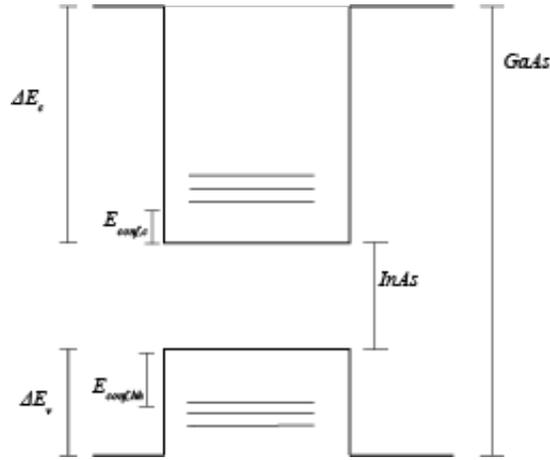


Figure 12: Band diagram of an InAs quantum well in GaAs. The potentials are the bandgaps of GaAs and InAs, the conduction band and heavy hole confinement energies, and the conduction band and valence band offset between GaAs and InAs

3.2 Spectrum Simulation

From the text book example, for a finite 1-D potential well the mathematical description is given as [22]:

$$\psi_1 = Ce^{\alpha x} \quad \psi_2 = A\sin(kx) + B\cos(kx) \quad \psi_3 = De^{-\alpha x} \quad (21)$$

We make the transformation:

$$u = \alpha L/2 \quad \text{and} \quad v = kL/2 \quad (22)$$

$$u^2 = u_0^2 - v^2 \quad \text{and} \quad u_0^2 = mL^2V_0/2\hbar^2 \quad (23)$$

From boundary conditions, and the requirement that the wave function and its derivative is continuous at all points, the main equations to solve is then:

$$\sqrt{u_0^2 - v^2} = v\tan(v) \quad \text{and} \quad \sqrt{u_0^2 - v^2} = -v\cot(v) \quad (24)$$

for symmetric and anti-symmetric solutions, respectively. For the ground state, in the *heavy holes* and the *conduction* bands, the equation to solve is the symmetric equation with

$$E_n = \frac{2\hbar^2v_n^2}{mL^2} \quad (25)$$

We use the effective masses for electrons in the conduction band, and for electron holes. For InAs

$$m_e = 0.023m_o \quad , \quad m_h = 0.41m_o \quad (26)$$

where m_o is the electron rest mass.

For $L=10\text{nm}$ we get two solutions for the excited electron:

$$v_1 = 1.15 \quad v_2 = 2.23 \quad \text{with} \quad E_1 = 0.00202\text{eV} \quad E_2 = 0.00759\text{eV} \quad (27)$$

and for the heavy hole there are nine solution the two lowest of which are:

$$v_1 = 1.44 \quad v_2 = 2.89 \quad \text{with} \quad E_1 = 0.00780\text{eV} \quad E_2 = 0.0312\text{eV} \quad (28)$$

An elaboration to this simplest approximation is then to include the different effective masses, m_* . The masses enter the time-independent Schrodinger equation as the proportionality between curvature and $E - V(r)$, so the spatial dependence of effective mass will modify the solutions' probability density relatively between the bound region, InAs, and the bordering regions, GaAs. For GaAs

$$m_e = 0.063m_o \quad , \quad m_h = 0.51m_o \quad (29)$$

This is no longer available for analytical solutions, so COMSOL is employed to calculate the eigen-states numerically. Figure 13 shows the 1D segment COMSOL model, representing the z-axis through the InAs QD and the first three solutions to the Schrodinger equation for electrons in the conduction band. Similar profiles occur for the heavy holes. Figure 14 show the first three eigen-energies as a function of QD thickness.

When these calculated energies are compared to those observed in InAs QDs from literature [24][25] the discrepancy is substantial. The missing factor is the strain. Pikus-Bir theory describes, among other, how the band structures of semiconductors is affected by strain. It is built on $\mathbf{k} \cdot \mathbf{p}$ perturbation theory, and is beyond the scope of this thesis. However, as a crude approximation, the conduction band deformation potential in the Pikus-Bir Hamiltonian, a_c , can be used [23]. a_c is -6.66 eV for InAs and it is the derivative of conduction band energy with respect to strain. So to approximate the order of magnitude, we can multiply by the lattice mismatch; $a_{c,InAs} \cdot -0.072 = 0.50\text{eV}$. A similar deformation potential lowers the energy of the split-off band, but since the main transition is between the conduction band and the heavy hole band at the Γ -valley, it suffices to look at the change in conduction band energy.

The strain induced spectral shift is an order of magnitude bigger than the confinement energy and shows that the key aspect of manufacturing small buffer structure top facet is to have a GaAs/InAs with no plastic relaxation.

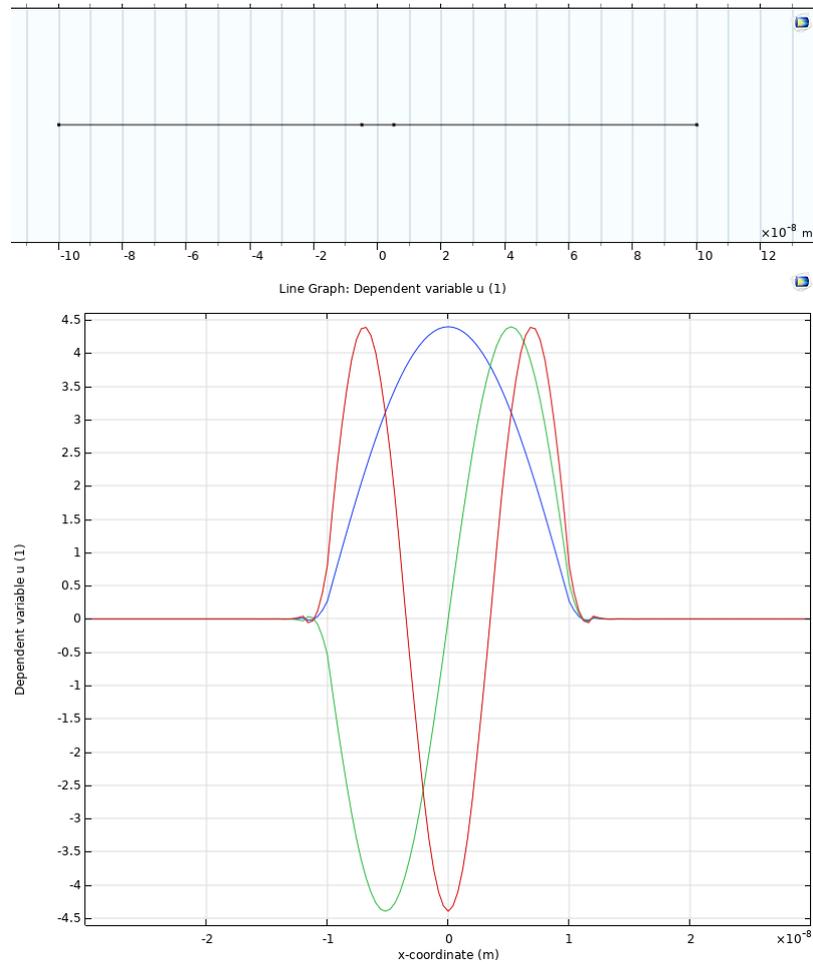


Figure 13: 1D quantum well calculations in COMSOL. Above) the 1D GaAs-InAs-GaAs segment. Below) The first three solutions to the Schrodinger equation for conduction band electrons

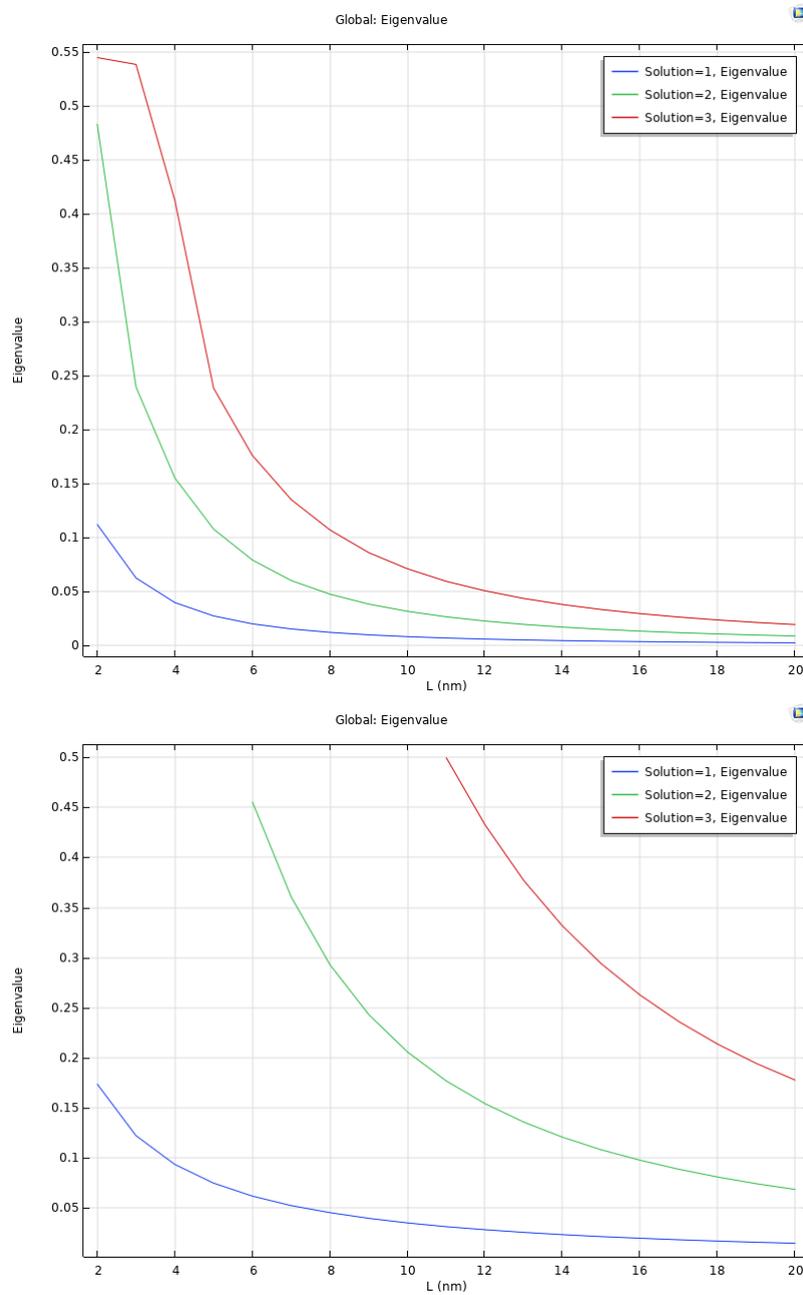


Figure 14: Calculated eigenvalues as a function of well length (QD thickness). Above) conduction band electrons. Below) heavy hole band. The difference in effective masses and band offset gives a higher confinement energy for the heavy hole band

4 AFM Analysis, Tools and Scripts

The main tool for characterization of the GaAs buffer structures is Atomic Force Microscopy (AFM). AFM measures a surface by probing with a tip of atomic sharpness as seen in Figure 15. AFM has many variations, working in different environments. It can measure topological, electronic, mechanical properties - for this study, only the topology is investigated.

In order to do a statistical analysis of the grown buffer structures, an analysis script was written. The aim was to develop a tool to statistically describe the morphology depending on opening dimensions and growth parameters. This script is built on the calibration steps of another script written by Anna Wulff Christensen.

Bruker has a package for python that converts the AFM-scan .smp data files into numpy arrays. Then all the different calibrations and extraction of physical quantities can be automatized with standard python array operations.

4.1 AFM Data and Calibration

The raw AFM image has several artefacts to correct for. This includes height offset, overall slope, line offsets, and curving of the slow axis start and finish.

Plane Leveling

The AFM image has a slope which the AFM cannot correct for itself. Therefore the mean plane is subtracted from the image to level it. Where some programs use a polynomial fit, here the mean plane is found by finding the maximum of a gradient histogram and the calculating the offset angle. Using gradients rather than mean planes, is more stable for non-symmetric AFM images.

Plane Zeroing

In order to place the surface of the SiO_x mask at zero, the height histogram peak is subtracted everywhere.

Line Levelling

Each time the AFM reads a new line of the sample, it picks up a slight offset from the previous line. This has to be corrected for as well by levelling each line individually. There are again several ways to go about that. As for the plane zeroing, the most effective way was to retrieve a height histogram of each line, then setting a range within

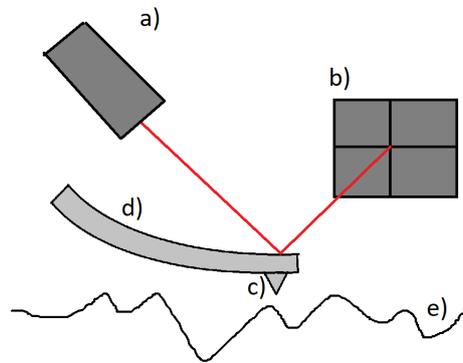


Figure 15: Schematic of AFM. a) A laser reflects off the cantilever b) the signal is collected and the offset is measured c) the AFM tip d) the cantilever e) the substrate surface

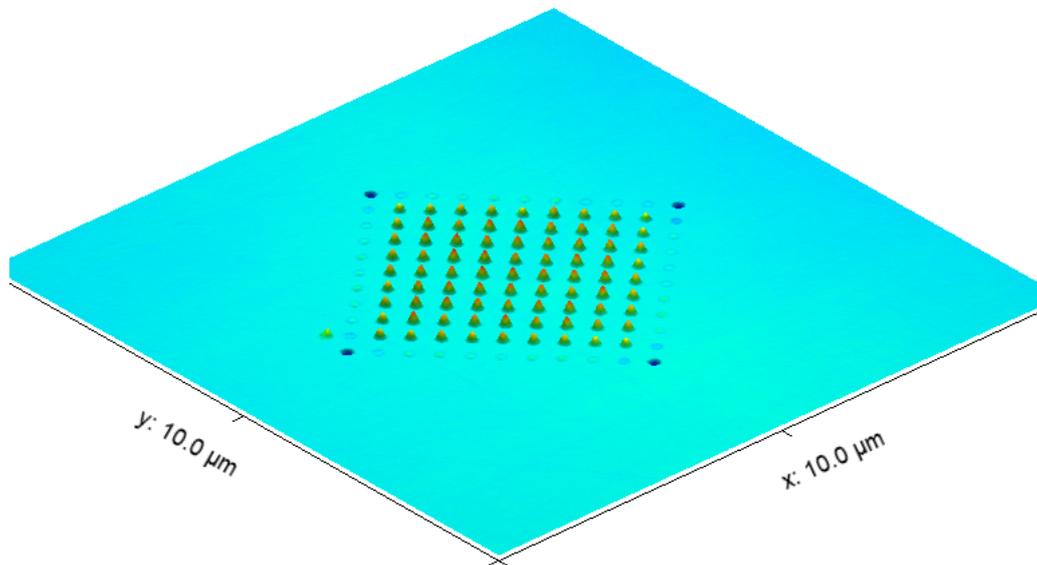


Figure 16: Color plot of an AFM scan.

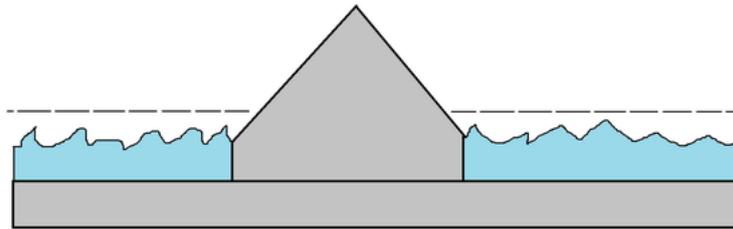


Figure 17: Schematic showing the data excluded by a threshold in the structure finding script

which the offset should be confined, and then finding the histogram peak, corresponding to the SiO_x surface.

Edge Cropping

At the edge of the AFM image, the image tend to be distorted so that a local slope is read. While this can also be corrected for using line levelling, it proved more stable to simply crop it out.

4.2 Structure Finding Algorithm

The package in SciPy, *regionproperties*, was used to find the buffer structures from AFM scans. A threshold value is set for the calibrated AFM image to select structures from background (SiO_x mask). Then each structure can be listed and its physical values assessed, such as position, area, volume, height, and shape. Figure 18 shows an AFM image of an array of GaAs buffer structures and the centroids of all the found structures

4.3 Extraction of Geometrical Values

A disadvantage with AFM is scan tip wear. In contact mode, the AFM tip is experiencing lateral forces from the roughness of the sample surface, and this will at some point make the tip blunt. Starting off, the tip is operational. By the end, the force feedback loop breaks down in lack of a stable force curve, when the tip is sufficiently damaged. However, in between, the tip can be either dented or blunted, resulting in a systematical error owing to the distinct shape of the tip. This can become obvious when the tip attains a particular shape that will then appear repeatedly in the scan. This can be seen

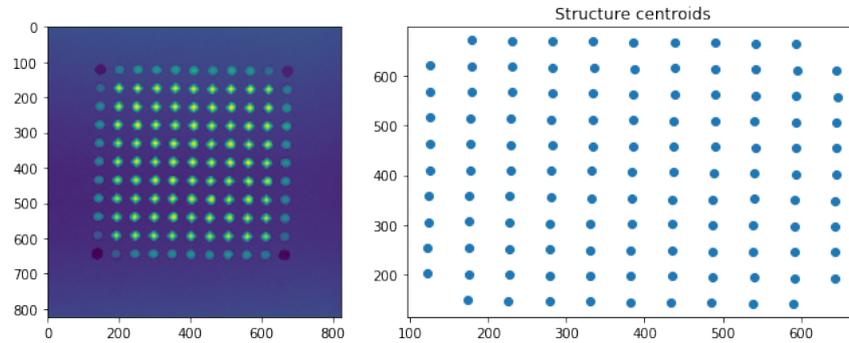


Figure 18: Left) an AFM image. Right) Centroids of the structure found

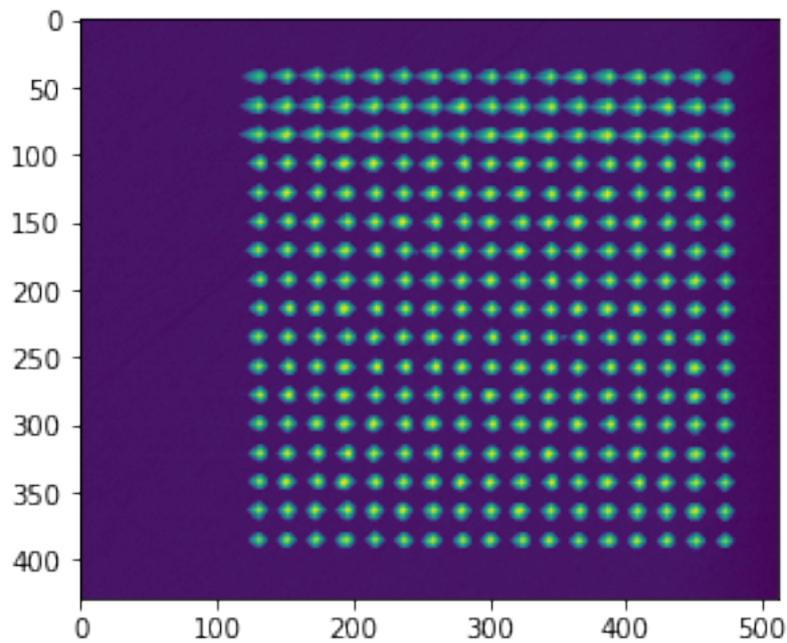


Figure 19: AFM scan of a buffer structure array. The tip is damaged during the scan and the upper structures attain an error matching the profile of the blunted tip

in Figure 19 where the tip gradually loses its sharpness leading to a systematic blurring in the x-axis. This means that the uncertainty of the extracted values using AFM is hard to estimate. While AFM has atomic resolution, the finite lifetime of a tip apex must be taken into account.

4.4 Facet Analysis

It is very important to know the facet composition of the grown buffer structures, and to get statistical estimates of facet areas. The 2D-histogram of the gradient of AFM scans is used to extract these values. The 2D-histogram of the buffer structures can be transformed into a stereographic projection and for every facet represented in the structures, there will be a distribution of counts around the corresponding point. Summing these peaks, respectively, will give a measure of the facet area. It is of course approximate, since each contribution to a particular facet, in general, need not be connected - but seeing that we are working with convex shapes, the peak should represent a connected facet.

In combination with the structure finding algorithm, the facets of each particular shape in an array can be assessed.

Figure 20 shows a stereographic project of all the lowest order facets of zinc-blende GaAs, (figure borrowed from [26]).

In Figure 21 a stereographic projection is shown for the slope distribution of a whole AFM image of an array of buffer structure. (Notice the 45° rotation coming from the [1-10],[110] orientation of the wafer)

The transformation is done by, transforming the 2D gradient into its normal vector in spherical coordinates:

$$\phi = \arctan\left(\frac{\nabla_y S}{\nabla_x S}\right) \quad \theta = \arctan\left(\frac{1}{\sqrt{\nabla_x S^2 + \nabla_y S^2}}\right) \quad (30)$$

where S is the surface as a function of x and y, and ∇ is a numerical gradient. Then it is transformed to the stereographic projection. The azimuthal angle, ϕ , remains the same while the zenith angle is replaced by a radius:

$$R = \cotan\left(\frac{\pi - \theta}{2}\right) \quad (31)$$

Figure 22 shows six gradient histograms for six individual GaAs buffer structures. The variety of shapes is apparent, ranging from pyramidal shapes consisting of {110} facets, to ECS-like shapes with a composition of facets close to and including the [001] facet.

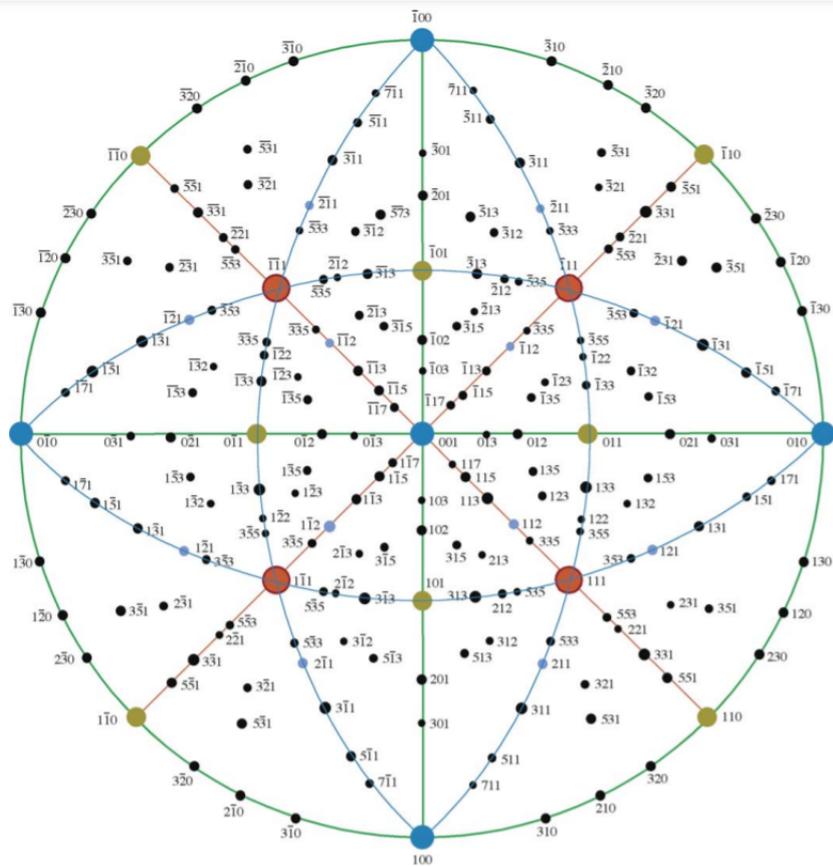


Figure 20: A stereographic projection of the lowest energy facets of zinc-blende GaAs. The bigger facet dots indicate the facets with lowest surface energy density

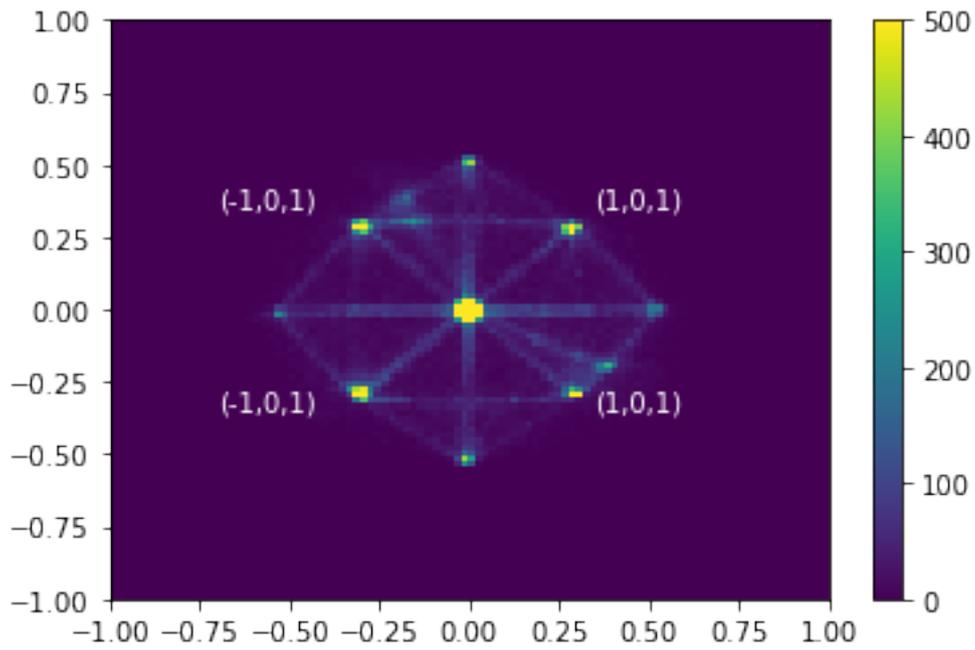


Figure 21: A stereographic projection of a slope histogram. The data is from a full AFM scan of an array of buffer structures

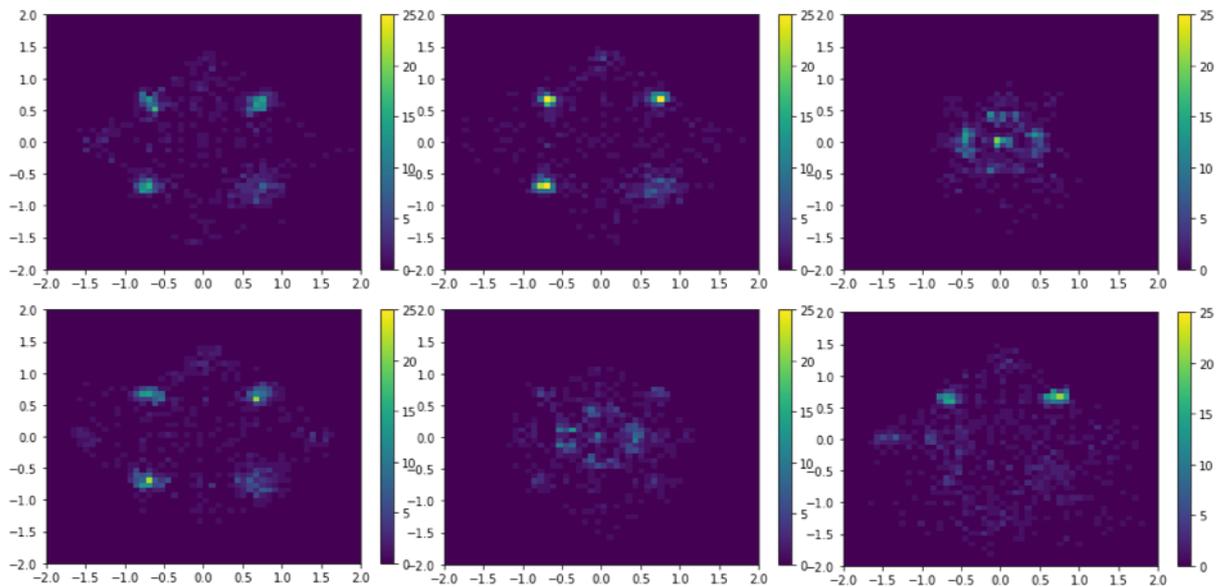


Figure 22: 2D gradient histograms for six specific buffer structures.

4.5 SEM Analysis

Scanning Electron Microscopy (SEM) is a useful tool to image samples. The basic principle is to focus a beam of electrons and move across a section of the sample and measure the scattered electrons with a detector placed either above or below the beam lens. The electron beam, consisting of high energy electrons, will be reflected or be partially transmitted through the sample surface, and the number of secondary scattered electrons will depend on the surroundings. That way, kinks and edges will appear more brightly, and outline the morphology of the sample.

While the resolution is high, the data itself is not as straight-forward to use for quantitative analysis, since each pixel of an image represents the intensity or amount of electrons collected by a detector. The scattering of the electron beam relies both on material, roughness, slope, and vicinity to edges.

A work in progress is to make an algorithm that automatically extracts facet areas and similar values of a structure from a SEM image. It relies on image blurring, curvature transformation, watershed segmentation, and labelling of each facet, which is seen in [Figure 23](#).

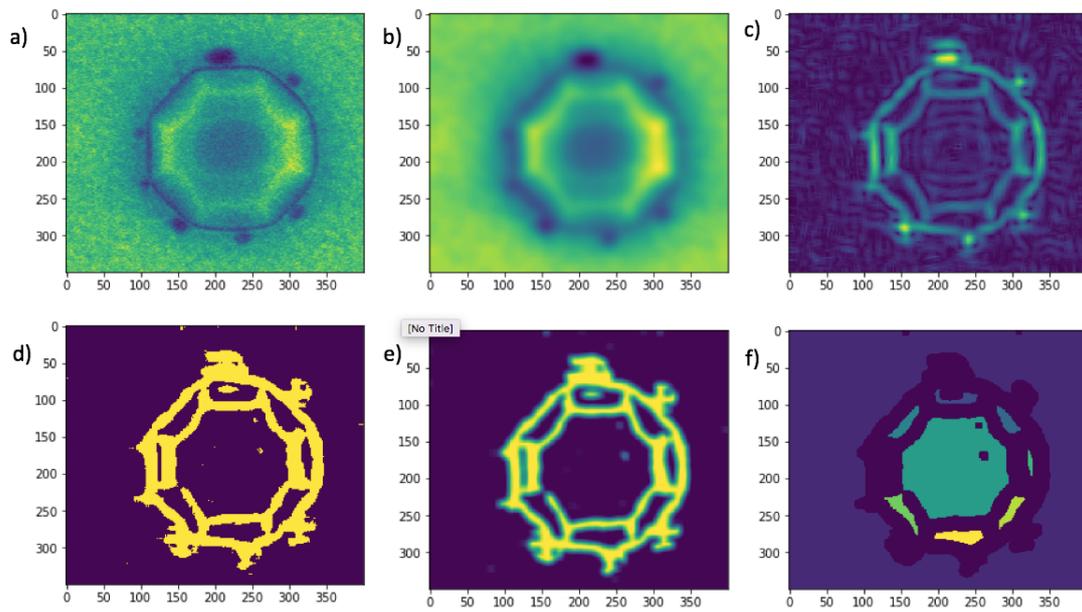


Figure 23: SEM analysis. a) The raw SEM image. b) Image blur to get rid of short scale noise. c) Curvature of the image. d) Watershed segmentation. e) Image blur. f) Facet labelling

5 Results and Discussion

This chapter reports the experimental results. It includes a statistical analysis of the GaAs buffer structure growth from AFM, then analysis of the InAs QD growth through SEM and STEM, and finally the PL experiments.

5.1 Designs of Growth

The mask opening size, shape, pitch, and coordination play a big role in how the growth and morphology progresses. A variety of different arrays is used to find the most effective and reproducible solution.

The samples contains six shapes: octagons, hexagons, squares, circles, rotated hexagons and rotated squares. An image of a design is shown in the appendix.

5.2 Fabrication Process

Fabrication describes the steps preparing oxide mask sample for MBE growth. A more detailed description is found in the appendix. Figure 24 shows an AFM scan of a prepared growth mask and a height histogram. A double Gaussian function is fitted to the height distribution. The oxide height and the roughnesses can be estimated from the means and the variations, respectively.

Both in EBL and in MBE growth there is a proximity effect to take into account. To test if the proximity effect from EBL is present, the correlation between areas of etched

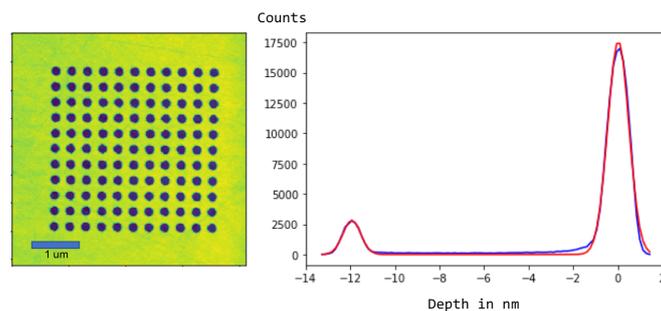


Figure 24: Left) an AFM scan of a prepared SAG mask. Right) A height histogram with a fitted double Gaussian from which mask thickness and roughness are extracted.

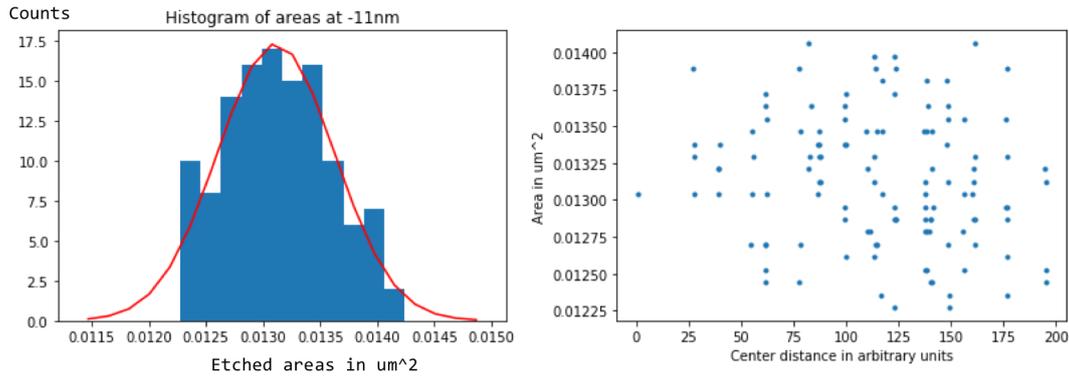


Figure 25: *Left*) Histogram of opening areas of a pre-growth substrate. The oxide thickness here is 12nm and the areas are found with threshold -11 nm. *Right*) Scatter plot of areas vs position to check for correlation.

openings and their array position were calculated, Figure 25. Their Pearson's correlation coefficient is -0.17 which is quite small suggesting no correlation.

5.3 Dimensions and Incorporation Rates of GaAs Growth

The GaAs buffer structures form the growth sites for the subsequent InAs growth. The distributions of geometrical values; peak height, base area, and volume are shown in Figure 26 - both as a function of placement in the 11x11 grid from sample (above) and as histograms. The data is from sample *QDev1109*.

In Figure 27 the same distributions are shown, with the edge structures highlighted. For the peak heights, the distributions suggests that, with the edge taken out, the remaining structures form a sharp distribution of heights. The distribution of areas is more scattered, partially due to threshold sensitivity of the structure finding algorithm.

The source-effect in SAG is of increasing importance as structures of interest become smaller. The incorporation rate depends approximately linearly on the pitch of the arrays of quantum dots (keeping width constant). This means that for these smallest structures, there is a maximum distance to the nearest neighbours, before openings start to desorb rather than grow. In Figure 28 12 AFM image analysed by the AFM analysis script establishes the connection between pitch and growth rate for 80 nm and 100 nm opening widths, respectively. The connection is approximately linear, and crosses into negative net growth at roughly 900 nm pitch.

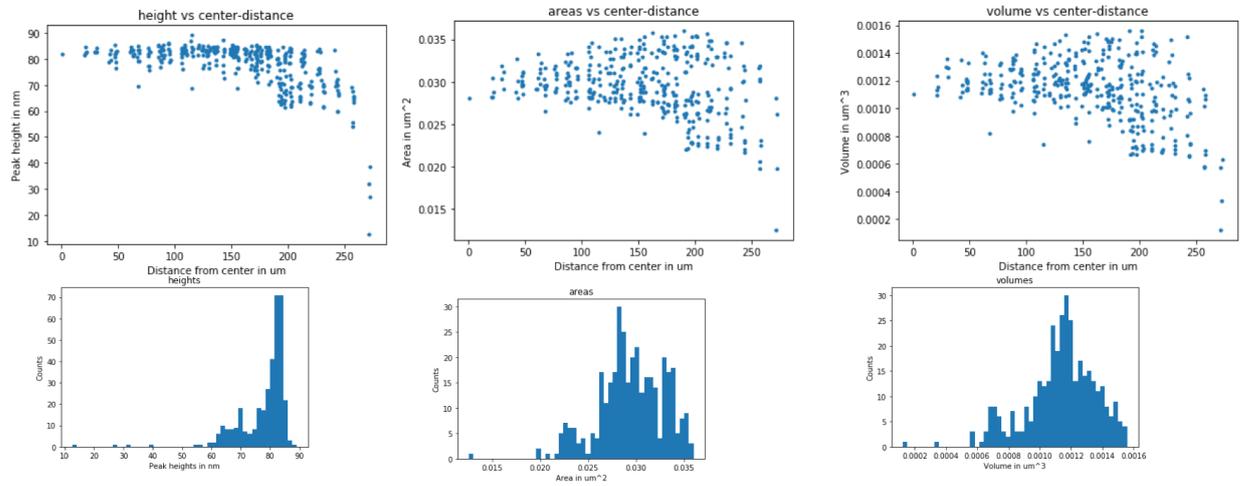


Figure 26: Above) Height, area and volume scatterplots of buffer structures as a function of position. Below) Height, area, and volume distributions

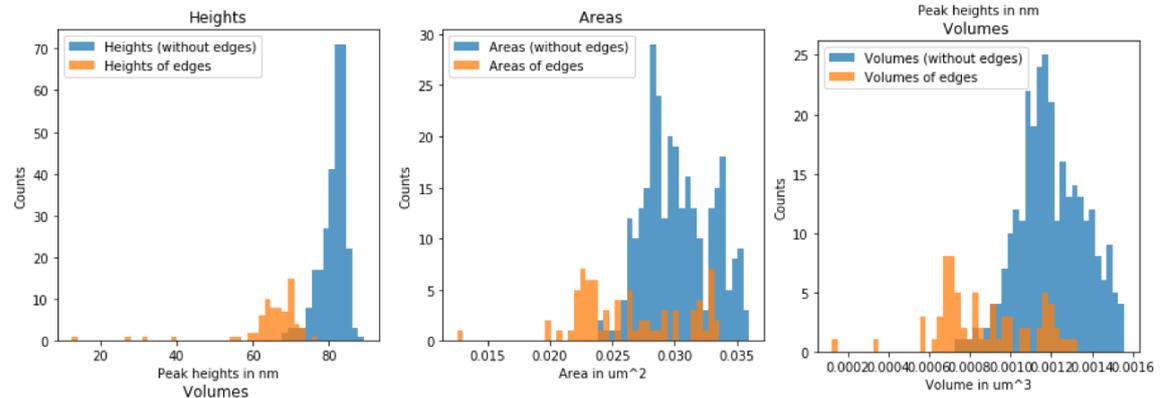


Figure 27: Height, area and volume distributions with edge structures labelled separately

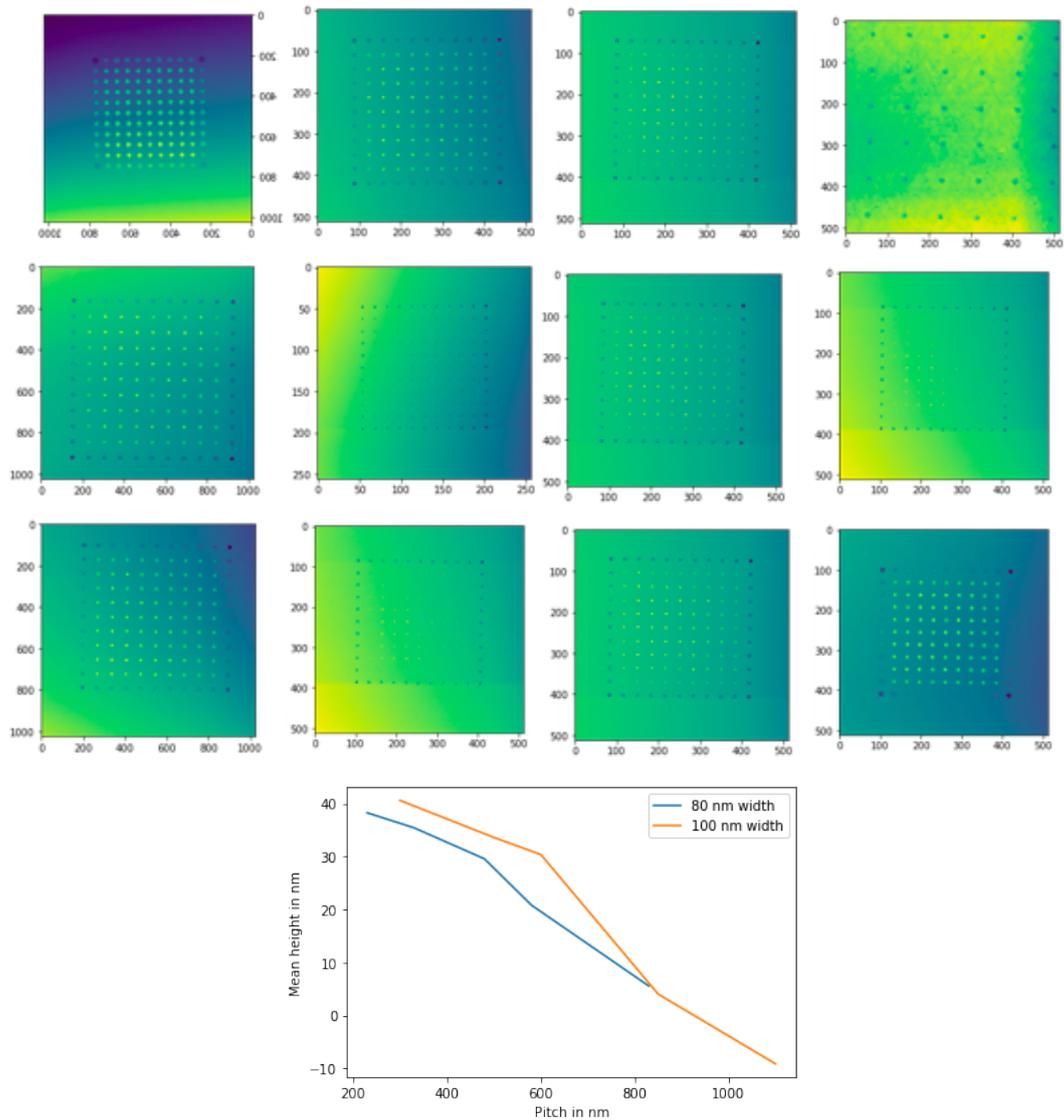


Figure 28: Above) 12 AFM scans of arrays with 80nm and 100nm widths (the displayed scans are shown before calibration). Below) The mean height of the buffer structure of the AFM scans as a function of pitch. The edge structures are excluded from the data.

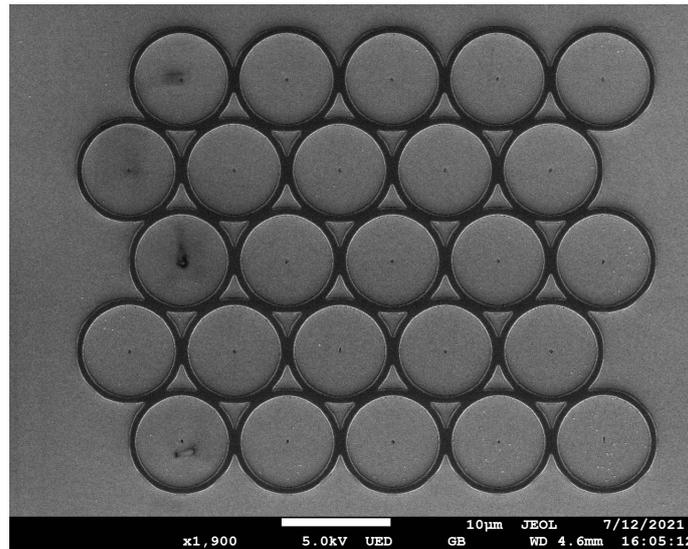


Figure 29: SEM image of ring structures, made to apply a maximum long-range sourcing effect

Structures were also designed to test how strongly the adatom sourcing can be. A ring opening surrounds a single small opening, of dimensions normally below growth conditions. This helps setting a minimum size, at maximum adatom density, given some radius of mask around, [Figure 29](#)

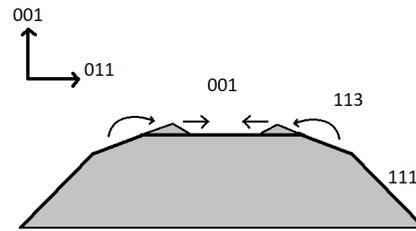


Figure 30: Schematic of the mechanism leading to fast top facet elimination. Adatoms form bordering facets lead to more nucleations at the edge of the top facet

5.4 Kinetic and Thermodynamic Growth

Crystal growth and formation is governed by both growth kinetics and thermodynamics. The balance between these two contributions is important to the shape and faceting of the growth. Kinetic growth shapes form after a growth front incorporating adatoms, and this growth front depends on orientation and facet. The thermodynamic growth allows for more exchange and relocation of crystal atoms and minimizes the free energy of the shape. A clear example can be seen from the larger features on the chips, that are comparable to 2D growth. For lower flux, the adatoms have a longer diffusion length, and moves around more before incorporating. This makes the surface reorganise into pyramidal shapes with low energy facets.

Another feature of thermodynamic growth is the exchange of adatoms across different facets. Faster top facet elimination for smaller structures occur as these edge effect become more important, Figure 30. This effect can be seen from looking at the edge of a $10 \mu\text{m}$ wide guide mark Figure 31. There are multiple factors at play, such as facet-dependent adatom mobilities or evaporation rates.

5.5 Facet progression

As the structures grow in the openings, the morphology will change, guided by growth kinetics and ECS. The different facets will have different areas through the growth, which in this project creates an important link between growth time and top facet area.

The different growth rates for bulk and edge structures, provides a range of different growth stages within an array. In Figure 32 volumes and top facet areas are plotted against distance from array center. The top facet area goes down approximately linearly as seen in Figure 33.

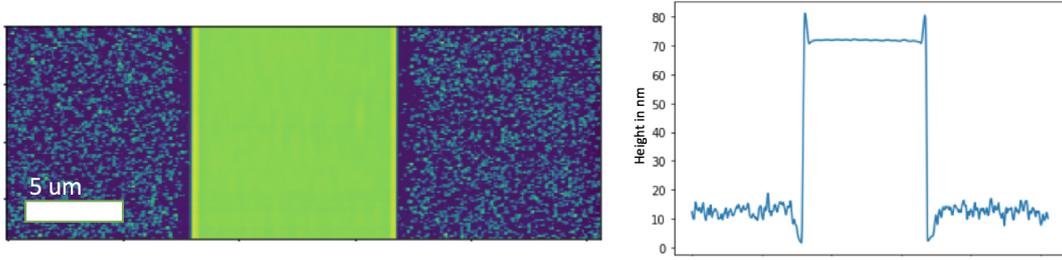


Figure 31: Left) AFM image of a section of a 10 μm rectangular guide mark on QDev1099. Right) an average profile showing increased growth at the edge of the 001-facet

Approximating the shape to a conical frustum 'filling up' a cone shape layer by layer, mathematically the top facet area is related to the height by:

$$A(h) = \pi \cdot \left(r \cdot \frac{h_{max} - h}{h_{max}} \right)^2 \quad (32)$$

and the volume is related to height as:

$$V(h) = \frac{\pi r^2}{3} \left(h_{max} - \frac{(h_{max} - h)^3}{h_{max}^2} \right) \quad (33)$$

The growth is similar to the top facet dependence of the volume of the frustum as seen in Figure 34 where this dependency plotted for four different height-to-radius ratios. The speed, with which the top facet disappears with volume, accelerates, underlining again the importance of a precise GaAs growth time accuracy and optimisation.

To better compare similar shapes for different opening sizes, a normalization is made such that both volume and areas are scaled such that they are unitless. This is done by reporting areas as a fraction of the opening area, and volumes are scaled with $A_{opening}^{3/2}$:

$$\eta_A = \frac{A}{A_{opening}} \quad \eta_V = \frac{V}{A_{opening}^{3/2}} \quad (34)$$

This formalism helps addressing the size-dependent effects that govern crystal growth and faceting. The ECS simulation figure is also following this formalism. For the two smaller opening sizes, 80 nm and 100 nm the η_A vs η_V is plotted with fitted piecewise linear curves in Figure 35. The slope gives a metric of how sensitive the top facet area is to incorporation rate. The top facet area goes directly to zero, but comparing with the simulated ECS in section 2 we see that for the lowest free energy, the top facet

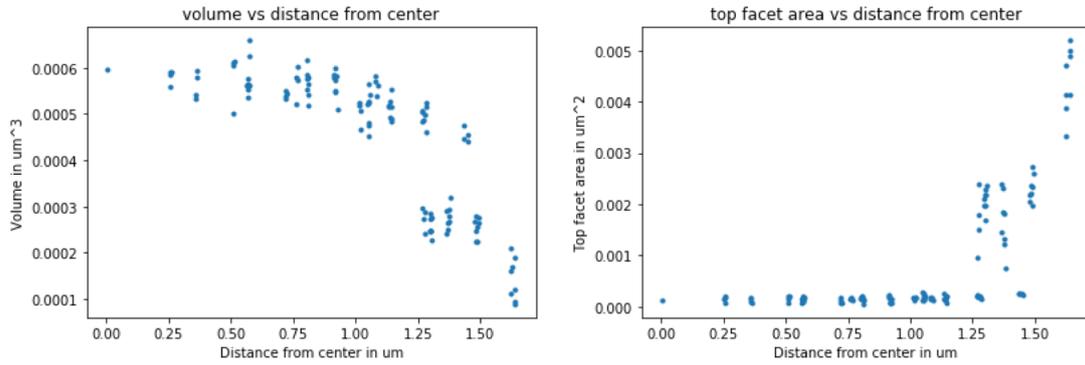


Figure 32: Left) Buffer structure volumes as a function of center distance. Right) Top facet area as a function of center distance. The edge structure do not have terminated top facets due to the slower growth at edges

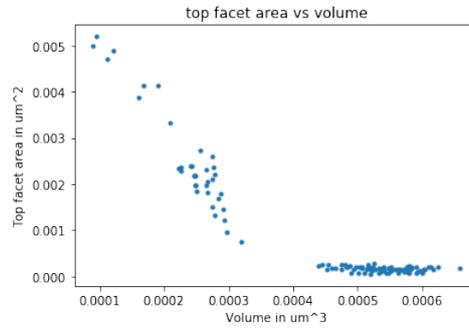


Figure 33: Top facet area as a function of volume for the GaAs buffer structures. The dependency is uniform and approximately linear

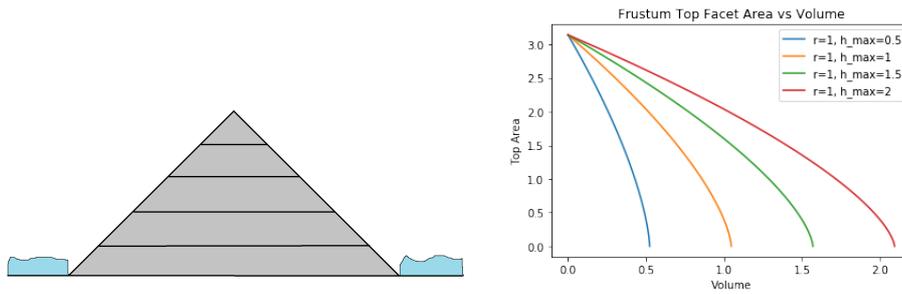


Figure 34: Left) A cross section of a conical frustum growing layer by layer. Right) Calculated top facet area as a function of volume for conical frustums, growing layer by layer with four different h/r ratios. The dependency is similar to the observed GaAs buffer structure

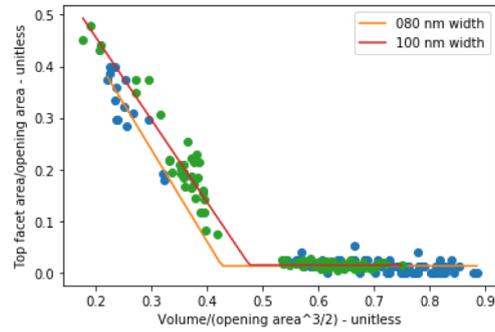


Figure 35: Normalised top facet area as a function of normalised volume shown for 80nm and 100nm width buffer structures

area converges at a finite value. In Figure 36 the distribution and progression of all the $\{110\}$ -facets are shown. Their dependence on growth stage is not as uniform, as it is for top facets. It was also observed, that for later growth stages, the four $\{110\}$ -facets rarely have same area. Rather, asymmetric shapes are more common.

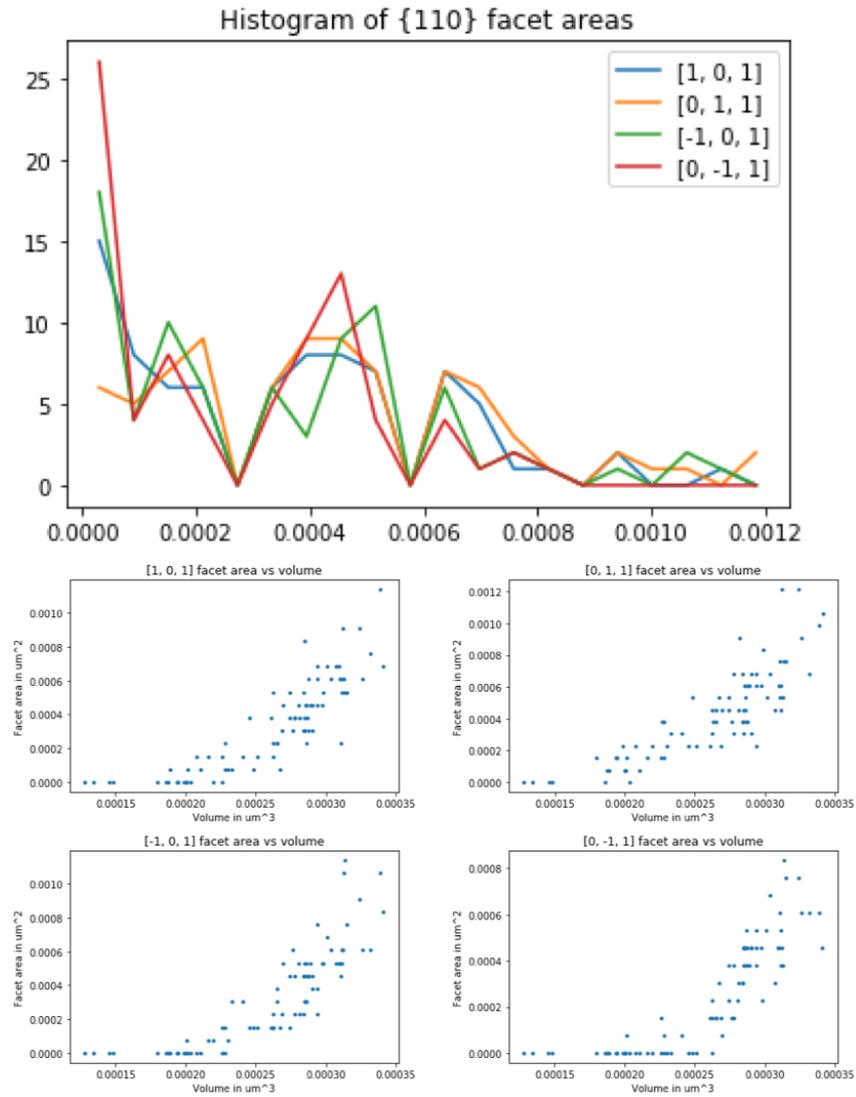


Figure 36: Above) distribution of $\{110\}$ facet areas of structures in an array. Below) $\{110\}$ facet areas as a function of volume. The scatter plots show how the $\{110\}$ facet areas are much more disperse than the top facet.

5.6 InAs Growth, SEM and TEM

The InAs is grown in the same growth session. The temperature is lowered, and the effusion cells are switched.

The InAs QDs were characterised initially using SEM EDX and subsequently through High-Angle Annular Dark Field Scanning TEM (HAADF-STEM) imaging of a lamella cut. Two arrays were chosen for STEM: an 11x11 grid of 125 nm width 1.5 μm pitch and 150 nm width 2 μm pitch. Although the aim of the project revolves around the smaller structures, these were chosen, since they exhibited the most reproducible InAs growth from SEM imaging, and to make sure that the FIB (Focus Ion Beam) lamella cut were able to coincide accurately with the structures.

5.6.1 SEM

The InAs growth was initially characterized with SEM. The assumed nature of the InAs, i.e., top facet growth is confirmed, as can be seen in Figure 37. It is clear that for the buffer structures with a well defined top facet, the InAs nucleates and accumulates there at the top facet. Some of the InAs QDs shapes suggest that the nucleation takes place around the edge between [001] and the bordering {113},{213} facets, as two or more domains merge across the top facet, leaving a non-convex shape. Conversely, when there is no top facet, which is in general the case for the smaller buffer sizes, the InAs will grow on the {110} facets, as can be seen in Figure 38. The sink effect during InAs growth has the effect of suppressing growth in areas close to better sites for nucleation, and this explains the exclusive nature of on-{110} InAs growth.

5.6.2 EDX

EDX is a method in SEM that detects the X-ray spectrum emerging from intra-atomic electronic transitions caused by the high energy electron of the primary beam. The detected X-ray energies then show the chemical composition and quantifies the abundances.

EDX was used to confirm that the InAs growth has been confined to the buffer structure top facet. Figure 39 shows an EDX line scan across an InAs QD. Although subtle, we see a dip in Ga and a small peak in In. The subtlety is due to the vast signal of the GaAs of the substrate.

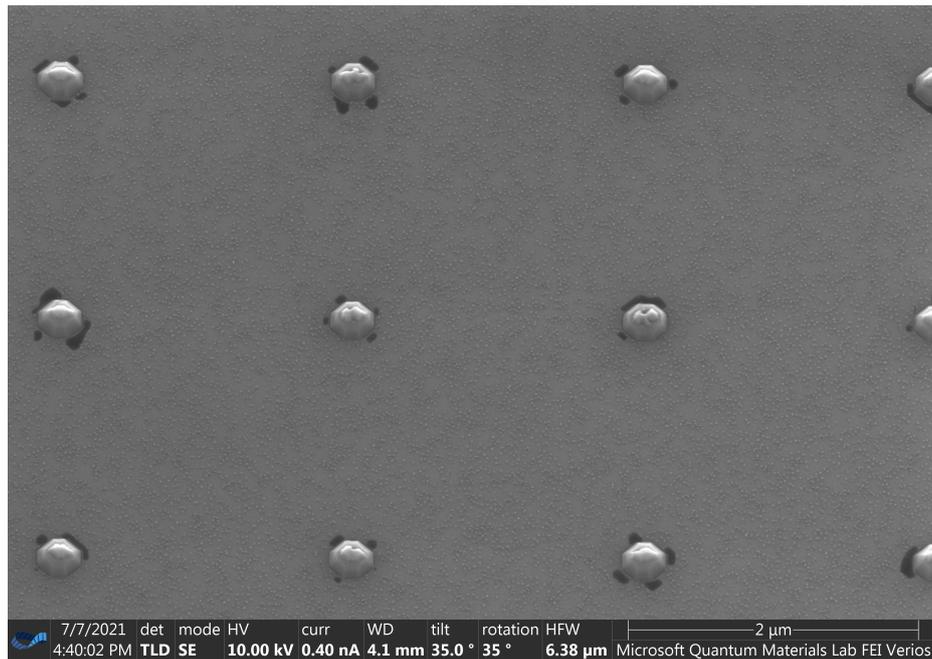


Figure 37: SEM image of buffered InAs quantum dots

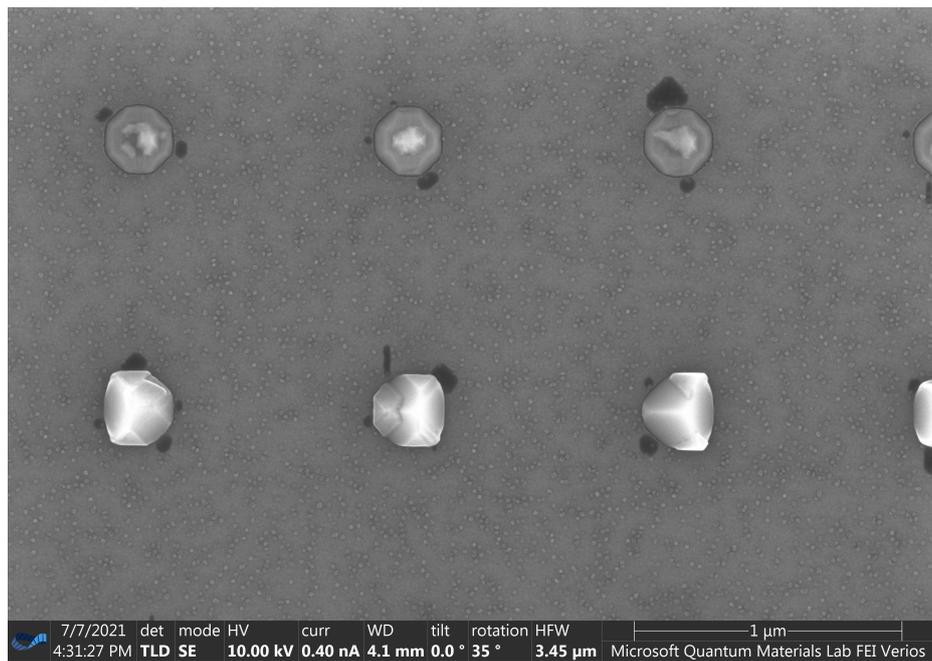


Figure 38: SEM image of the buffered InAs QDs. The upper row have InAs on top facets. The lower row of buffer structures have terminated top facets, and has InAs growth on $\{110\}$ facets

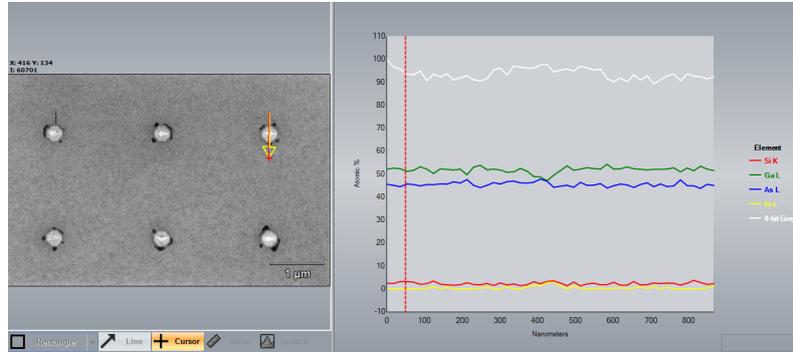


Figure 39: SEM image and an EDX line scan of an InAs QD. The measurement was carried out at MQML.

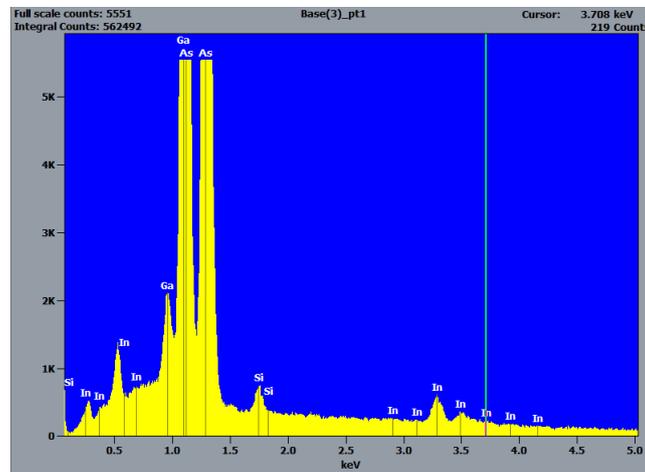


Figure 40: Point-scan EDX of an InAs QD

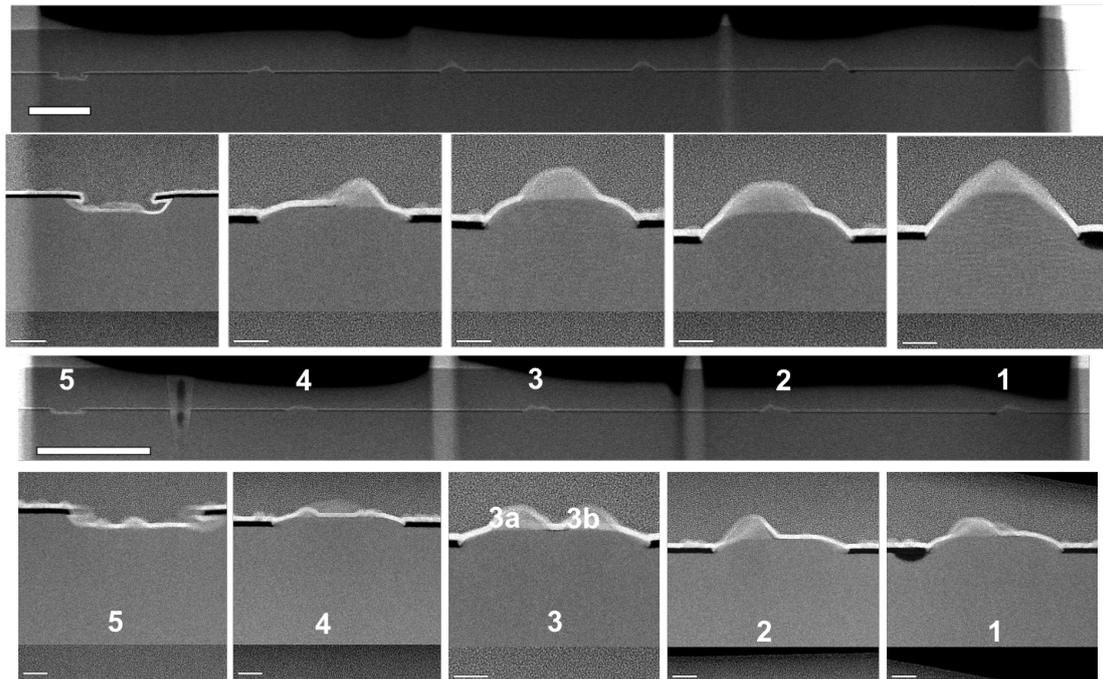


Figure 41: TEM images of the two lamella cuts. Five QD-buffer structures cross sections are shown from each lamella cut, with close-ups of each structure

5.6.3 TEM

By performing a FIB lamella cut, it is possible to gain insight into the crystallography of the MBE growths. This is then analysed in HAADF-STEM. To stabilise the sample during FIB, a capping layer of HfOx layer is deposited. In figure 41 TEM images are shown of one of two lamella cuts of the InAs QD.

5.6.4 GPA

Geometric Phase Analysis is a tool in high-resolution TEM that quantifies crystalline strain, both elastic and plastic. The main idea is to use a fast-Fourier Transform, FFT, of TEM imaging where each crystal atom is visible. Any displacement in the crystal will change the phase of the frequency-representation, so looking at a local phases will reveal any crystal deformations. The GPA is finally reported as a dilatation map, D_{xy} , and a rotation map, R_{xy} . The dilatation map shows the relative shift in periodicity between domains while the rotation map shows bending of the crystal lattice. In Figure 42 HAADF-STEM images, dilatation maps, and rotation maps are shown for the lamella

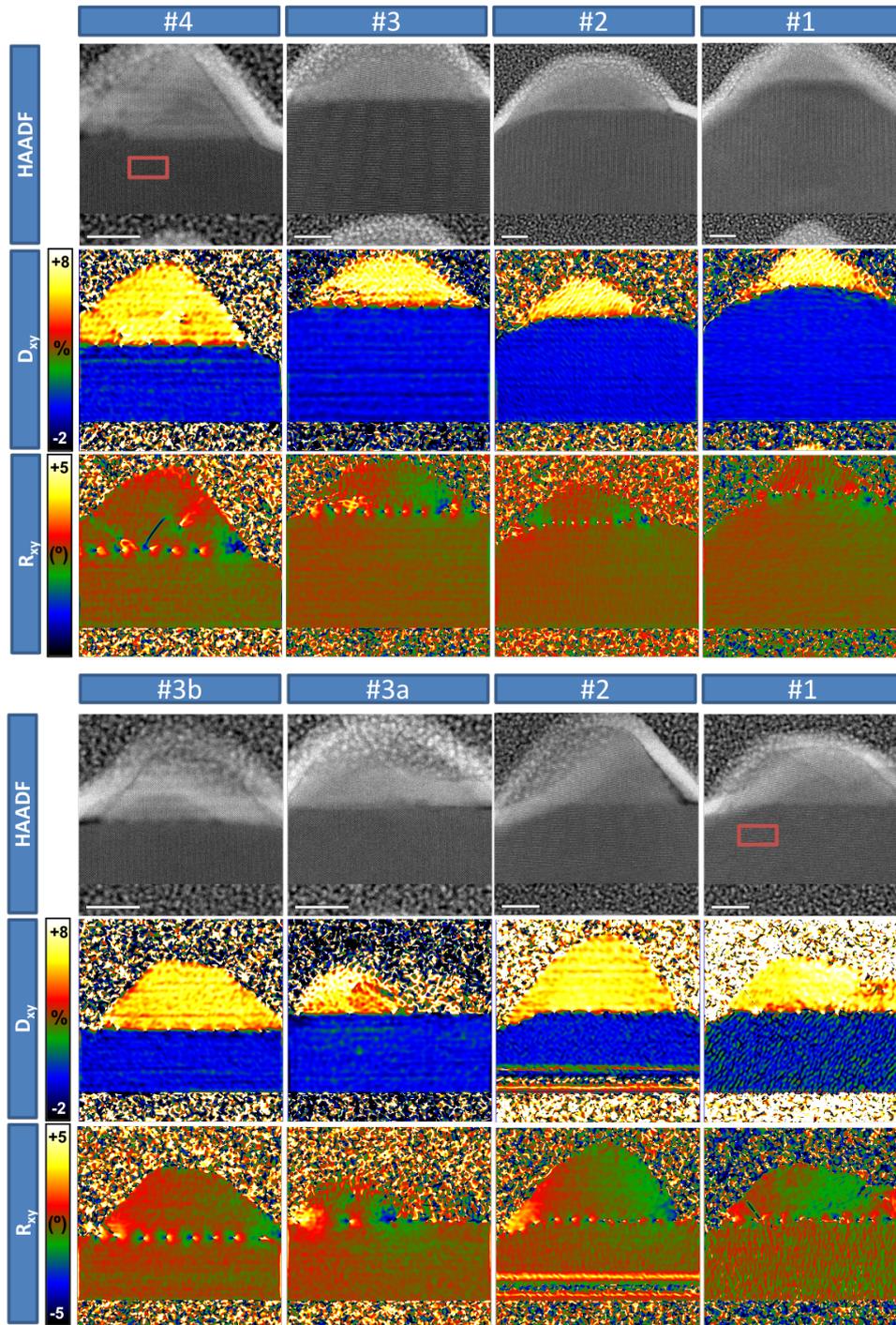


Figure 42: GPA plots of a selection of QDs from two lamella cuts. Above) HAADF images. Middle) GPA dilatation maps. Below) GPA rotational maps

cuts.

The HAADF-STEM images provide insight into the 3D shapes as the detected scattering electrons interact with crystal atoms through the lamella.

The dilatation maps show that the InAs is plastically relaxed, visible from the relative dilatation of the 7.2 % lattice mismatch. When fully plastically relaxed, the lattice mismatch causes array dislocations of $\frac{100\%}{\text{lattice-mismatch}} \cdot \text{lattice-constant} = \frac{100\%}{7.2\%} \cdot 0.56\text{nm} = 7.8\text{nm}$ spacing, consistent with the dilatation map. This tells us that the specific sizes investigated with TEM are well above the threshold under which plastic relaxation can be avoided.

The rotation maps also show the periodic dislocations very clearly, and reveal elastic rotation at the edges of the InAs QDs

5.6.5 Electron Energy Loss Spectroscopy

Electron Energy Loss Spectroscopy (EELS) is a method in TEM that characterizes chemical composition in materials by looking at the energy loss spectrum of a electron beam. Similarly to EDX, EELS can detect core-electron ionization potentials and thereby estimate chemical compositions.

Figure 43 shows the EELS measurements of the lamella cut QDs, from two different arrays. Overall the interface between GaAs and InAs is quite sharp. That is seen from the well-defined color boundary of Ga and In. In migration is known from literature[27], and is an important concept to monitor. The relative atomic composition figures show a brief transition from GaAs into InAs.

Another important aspect is that InAs growth is how the InAs is confined to a single bulk shape. Compared to SK growth, where a wetting layer remains, the strong sink effect during SAG, due to a depletion of higher energetic growth site-adatoms, is confirmed through EELS.

5.7 Photoluminescence Experiments

The spectra of the InAs QDs can be measured with photoluminescence. Unfortunately, the photon detectors mounted on the spectrometer, existing in our lab, are Si which is unable to detect wavelengths above 1000nm. The dimensions and strain conditions, mentioned in chapter 3 and 5 lead to much longer wavelengths than of QDs commonly

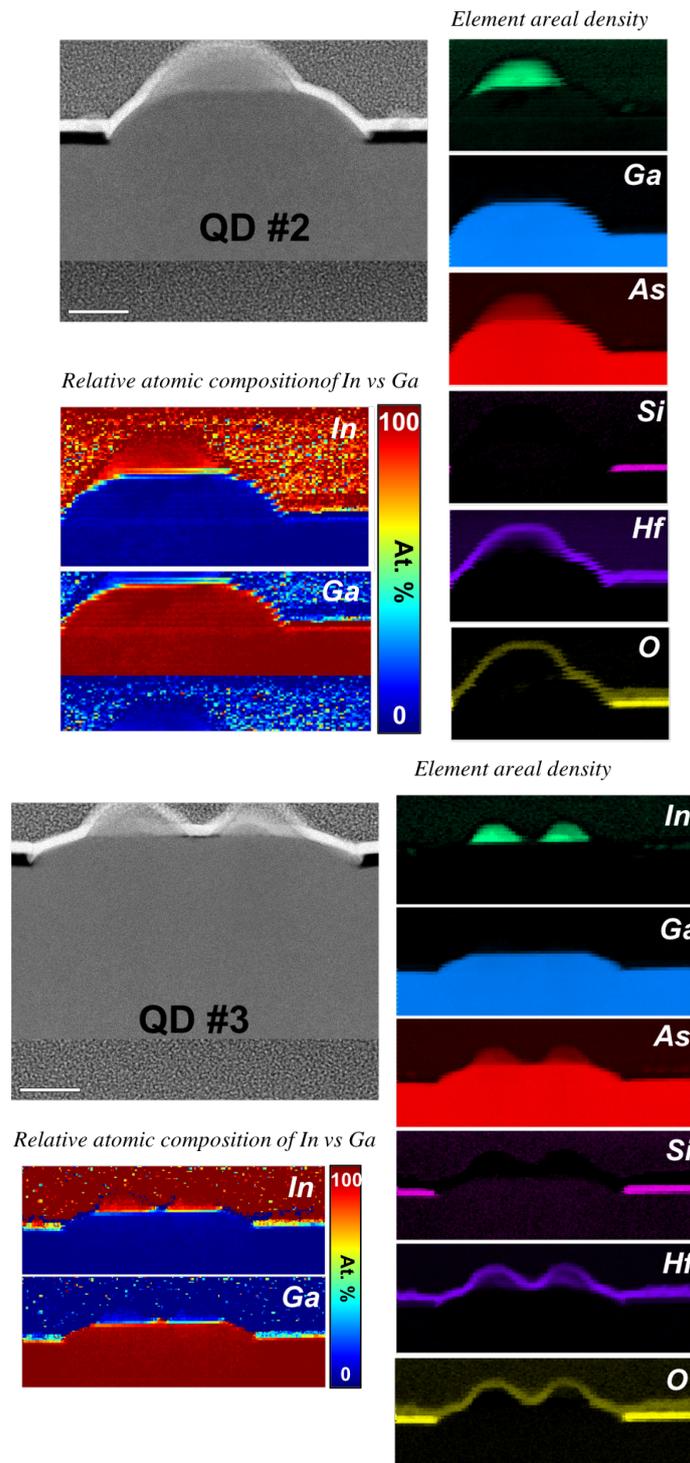


Figure 43: Caption

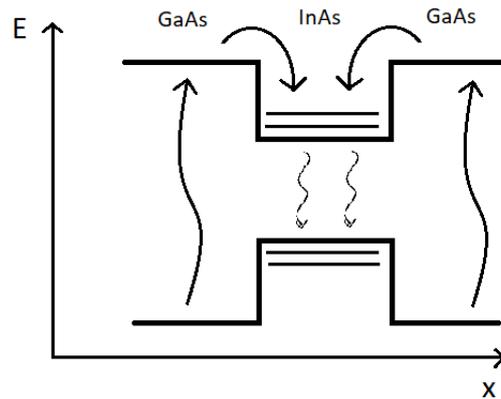


Figure 44: Above band excitation. Excited electron from the surrounding GaAs vibrationally relaxes to the QD and recombines.

used in quantum photonics ≈ 930 nm. Therefore the only results based on the optical experiments are the fluorescence seen in the CCD camera of the setup, which can detect some higher wavelengths than the spectrometer.

Before having exact knowledge of the resonance frequency of the QDs, above-band excitation can be used. Here a laser, with a frequency with energy above the GaAs band gap, is applied. Electrons in the substrate is excited from the valence band to the conduction band as in figure 44. Through vibrational relaxation, the excited electrons will flow to the lowest energy states in the conduction band of the InAs QD, and accordingly the holes in the valence band will 'bubble up' to the highest energy level. That way, an InAs QD will locally be a main source of radiative decay, since the the resonance energy is lower than the bandgap of the surrounding GaAs.

The setup used to carry out the photoluminescence experiments is shown in figure 45, which includes exciting quantum dots, collecting and detecting the emitted photons. The left and upper part of the figure 45 shows the excitation path and the according components as follows: :

- Laser. Both a Tsunami Laser and a continuous tunable laser (CTL) is required. The CTL laser is applied to align the setup.
- Signal is split by a Fiber beam splitter

- Wavemeter and PID. A wavemeter is used to measure the signal wavelength. A PID power controller is employed to apply a stable power in the path.
- The light is coupled out of the optical fiber into the free space path
- Input Mirrors. A pair of mirror is implemented to guide the light on targeting sample surface.
- Power meter (PM). Measure the applied power.
- Optical filter
- A focusing lens widens the signal profile
- A beam splitter introduces the white light source for imaging.
- Another focusing lens focuses the signal to the focal plane of the sample surface

The right side in the figure 45 displays the excitation path and the according components as follows:

- Another beam splitter to separate the excitation and collection paths
- Optical filter
- A beam splitter to guide part of the emitted photons into a CCD camera.
- A focusing lens
- Optical filter
- Output mirrors operating the emitted photons into detection channels.
- Fiber-coupling to spectrometer
- Spectrometer

In Figure 46 two CCD images of an array of 400nm width structures with 1 μm pitch, first with a white light source, then with a pumping laser. We can see the InAs fluorescing with various intensities in the array.

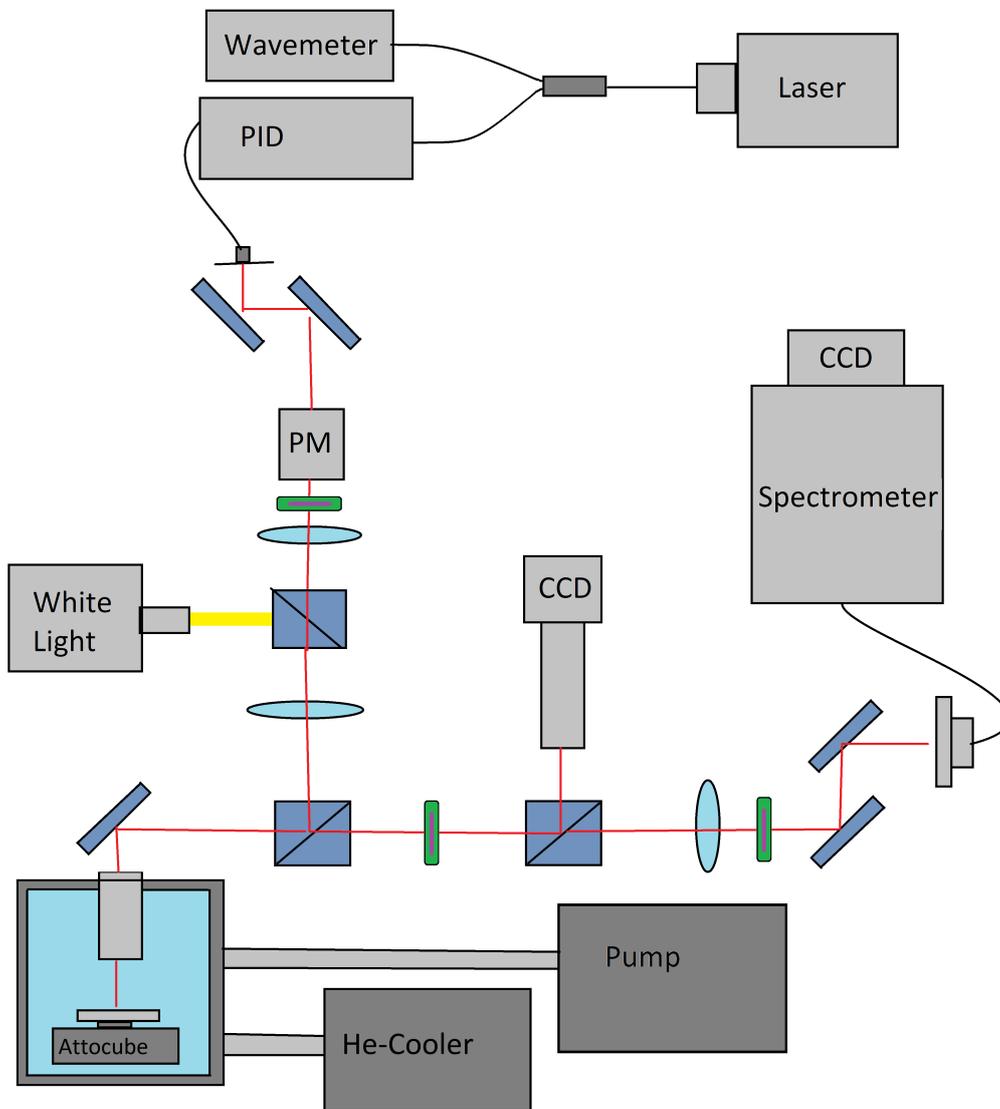


Figure 45: Schematics of the optical setup for PL measurements.

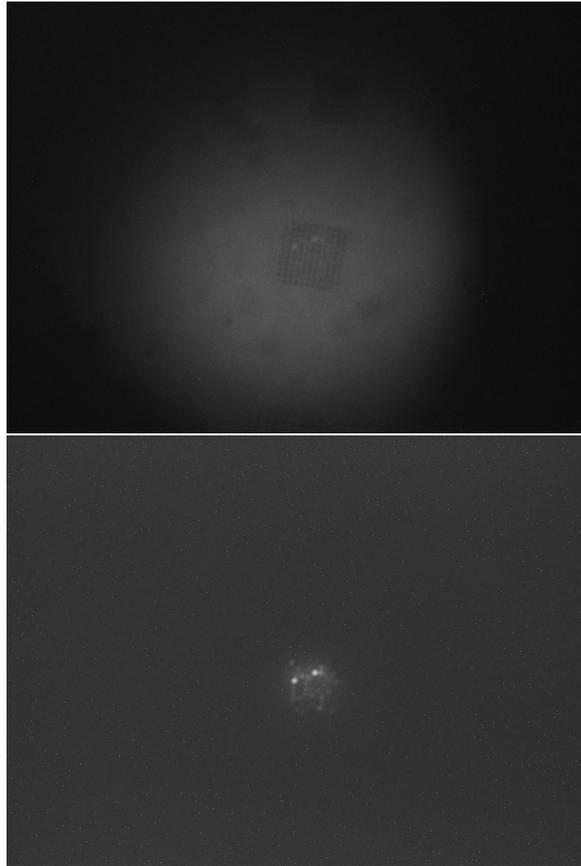


Figure 46: CCD images of an array of InAs QDs in the PL setup (growth openings have 400 nm width, 1000 nm pitch). Above) The array is seen with a white light source. Below) The same array is seen with the excitation laser passed through a low-pass filter

6 Conclusion

This report contains a experimental approach and a simulated treatment of the physics that go into making positioned InAs QDs. While the final product does not set any full stop in this regard, the combined physical data, the conceptual considerations, and theoretical treatment should help showing the way to positional and spectrally tailored buffered InAs QDs.

The major challenge was to ensure GaAs growth in small mask openings with high pitch. The small openings serve to create structures with a small top facet, and the high pitch is important to do single QD excitation. The growth rate has a linear decreasing dependency on array pitch, with the growth parameters used, and for 80 nm buffer diameters it becomes negative around 900nm pitch. The optimization of growth time remains. The importance lies in reproducibly creating top facets with diameters 4-5 times smaller than the opening. That way, the InAs will remain elastically strained, unlike what is seen with GPA for larger structures. With the limited amount of growths - the perfect growth time, temperature, size, and pitch setting is yet to be set, however, we have moved in on the optimal solution.

The initial assumption, that InAs growth would be confined to top facets, was confirmed by EELS. When a top facet is available for nucleation, the sink effect inhibits growth on the {110} facets. The GaAs-InAs interface is sharp, and TEM shows clean, connected domains of InAs. The photoluminescence setup did not allow for spectral measurements, and that leaves the question of spectral reproducibility unanswered for now. It is, however, clear that the InAs QDs are optically active, as can be seen by the CCD in the PL setup.

Assuming all these parameters optimized, the next challenge becomes to integrate the InAs QDs into a photonic device. An interesting study would be to try to remove the SiO_x mask or to assess the capping layer of the final growth as a wave guide.

6.1 Acknowledgements

The work of this thesis is the product of a collaboration between QDev/MQML with supervisor prof. Peter Krogstrup and HyQ with supervisor prof. Peter Lodahl. It has been an exciting and enlightening task to bridge these two scientific and intellectual capacities, and a great gratitude is extended for making it possible.

The MBE growths, that are the axis of this project, was carried out by postdoc Daria Beznasyuk, who deserves great appreciation for her work and expertise. Characteriza-

tion, fabrication and PL experiments was assisted by Gunjan Nagda, Martin Espineira, Charalampos Lampadaris, Somya Gupta, Jordi Arbiol, Sara Marti, Ying Wang, Alexey Tiranov, and Vasiliki Angelopoulou.

A special thanks is owed to Leonardo Midolo and Tobias Særkjær for help with simulations and theoretical background and to Anna Wulff for help with data analysis.

Finally a mentioning is owed to Kamal Maharajan for feedback and support, and to Charline Kirchert who shared the same project and has been a close colleague these last two years.

References

- [1] MacFarlane, A. G. J, Jonathan P Dowling, and Gerard J Milburn. Quantum Technology: The Second Quantum Revolution. *Philosophical transactions of the Royal Society of London. Series A: Mathematical, physical, and engineering sciences* 361.1809 (2003): 16551674. Web.
- [2] Girard, JM & Robert, Ikwera & Moreau, Eric & Gayral, Bruno & Abram, I. (2003). InAs quantum dots: artificial atoms for solid-state quantum optics. 11-20.
- [3] Peter Lodahl. *Quantum-Dot Based Photonic Quantum Networks*. N.p., 2018. Print.
- [4] Lodahl, Peter, Sahand Mahmoodian, and Soren Stobbe. Interfacing Single Photons and Single Quantum Dots with Photonic Nanostructures. *Reviews of modern physics* 87.2 (2015): 347400. Web.
- [5] Borregaard, J et al. One-Way Quantum Repeater Based on Near-Deterministic Photon-Emitter Interfaces. *Physical review. X* 10.2 (2020): 021071. Web.
- [6] Karzig, Torsten et al. Scalable Designs for Quasiparticle-Poisoning-Protected Topological Quantum Computation with Majorana Zero Modes. *Physical review. B* 95.23 (2017): n. pag. Web.
- [7] Ihn, Thomas., and Thomas. Ihn. *Semiconductor Nanostructures: Quantum States and Electronic Transport*. Oxford: Oxford University Press, 2010. Print.
- [8] Schaffer, William J. et al. Nucleation and strain relaxation at the InAs/GaAs(100) heterojunction. (1983)
- [9] Stranski, I. N, and L Krastanow. Berichtigung zur Arbeit: Zur Theorie der orientierten Ausscheidung von Ionenkristallen aufeinander. *Monatshefte für Chemie* 72.1 (1939): 7676. Web.
- [10] Holewa, Pawe et al. Optical Properties of Site-Selectively Grown InAs/InP Quantum Dots with Predefined Positioning by Block Copolymer Lithography. *Materials* 14.2 (2021): 391. Web.
- [11] Ozaki, N et al. InAs Quantum Dots Grown on Selective Areas with a Metal Mask for Photonic-Crystal-Based Ultra-Small and Ultra-Fast All Optical Devices. 2007 IEEE 19th International Conference on Indium Phosphide Related Materials. IEEE, 2007. 493496. Web.

- [12] Krogstrup, Peter et al. Advances in the Theory of III-V Nanowire Growth Dynamics. *Journal of physics. D, Applied physics* 46.31 (2013): 313001. Web.
- [13] Yeu, In Won et al. Equilibrium Crystal Shape of GaAs and InAs Considering Surface Vibration and New (111)B Reconstruction: Ab-Initio Thermodynamics. *Scientific reports* 9.1 (2019): 11271127. Web.
- [14] Malarz, K, and A. Z Maksymowicz. A Simple Solid-on-Solid Model of Epitaxial Thin Films Growth: Surface Roughness and Dynamics. (1999): n. pag. Web.
- [15] Malarz, K, and A. Z Maksymowicz. A Simple Solid-on-Solid Model of Epitaxial Thin Films Growth: Surface Morphology Anisotropy. (1999): n. pag. Web.
- [16] Markov, Ivan V. *Crystal Growth for Beginners Fundamentals of Nucleation, Crystal Growth and Epitaxy*. 2nd ed. Singapore;: World Scientific, 2003. Print.
- [17] Cachaza, Martin Espiñeira et al. Selective Area Growth Rates of III-V Nanowires. (2021): n. pag. Web.
- [18] Einstein, Theodore L. *Equilibrium Shape of Crystals*. N.p., 2015. Web.
- [19] Wong, Ping-Show et al. Controlled Formation and Dynamic Wulff Simulation of Equilibrium Crystal Shapes of GaAs Pyramidal Structures on Nanopatterned Substrates. *Crystal growth design* 10.6 (2010): 25092514. Web.
- [20] Albani, Marco et al. Competition Between Kinetics and Thermodynamics During the Growth of Faceted Crystal by Phase Field Modeling. *physica status solidi (b)* 256.7 (2019): 1800518n/a. Web.
- [21] Zhou, Shuai et al. The Formation Site of Noninterfacial Misfit Dislocations in InAs/-GaAs Quantum Dots. *Journal of nanomaterials* 2014 (2014): 15. Web.
- [22] Hall, Brian C. (2013), *Quantum Theory for Mathematicians*, Graduate Texts in Mathematics, 267, Springer
- [23] Pryor, Craig. Eight-Band Calculations of Strained InAs/GaAs Quantum Dots Compared with One-, Four-, and Six-Band Approximations. *Physical review. B, Condensed matter* 57.12 (1998): 71907195. Web.
- [24] Nedzinskas, R et al. Energy Spectrum of InAs Quantum Dots in GaAs/AlAs Superlattices. *Acta physica Polonica, A* 113.3 (2008): 975978. Web.

- [25] Rakhlin, M V et al. InAs/AlGaAs Quantum Dots for Single-Photon Emission in a Red Spectral Range. *Scientific reports* 8.1 (2018): 52996. Web.
- [26] Williams, David B., and C. Barry. Carter. *Transmission Electron Microscopy A Textbook for Materials Science*. 2nd ed. 2009. New York, NY: Springer US, 2009. Web.
- [27] Tongbram, B et al. Optimization of InAs Quantum Dots through Growth Interruption on InAs/GaAs Quantum Dot Heterostructure. *Journal of luminescence* 192 (2017): 8997. Web.
- [28] Lopez, Gerald G. Ph.D. and Azadi, Mohsen, "PMMA A2 and A4 Spin Curves", *Protocols and Reports*. Paper 34

7 Appendix

7.1 Design

Figure 47 shows the design for EBL. Array of different size, pitch and coordination are drawn to investigate their effect on growth.

7.2 Fabrication Process

The preparation of a pre-growth substrate includes the multiple fabrication steps. It is demanded to optimize each single step and develop a reproducible recipe for fabrication.

7.2.1 Spin Coating

To obtain a uniform film of resist on the GaAs/SiO_x chip, the resist is dispersed on the sample and then spun at 4000 rpm for 45s and then baked at 185° to obtain a 200 nm (dry) resist. Figure 49 shows the film thickness depending on the spin speed from a report on EBL resists [28].

During the project, three different resist types have been used: EL9, A4 and CSAR.

7.2.2 Electron Beam Lithography

The 100 kV Elionix at H.C Ørsted Institue is used for EBL. In order to get a high resolution, low currents are used - 100 pA with aperture size of 40 μm. The different resist types requires different charge per area, and calibrations were made through dose tests. Figure 50 show two guide marks for two different EBL charges, and how overexposure widens the shape.

7.2.3 Development

The development is done using MIBK developper (Methyl Isobutyl Ketone). Positive electron beam resist has been used, and the exposed area will be dissolved in the developer.

To monitor this intermediate step before etching, gold was deposited using an AJA, shown in Figure 51. That way more insight can be gained into which part of fabrication is the resolution limiting step.

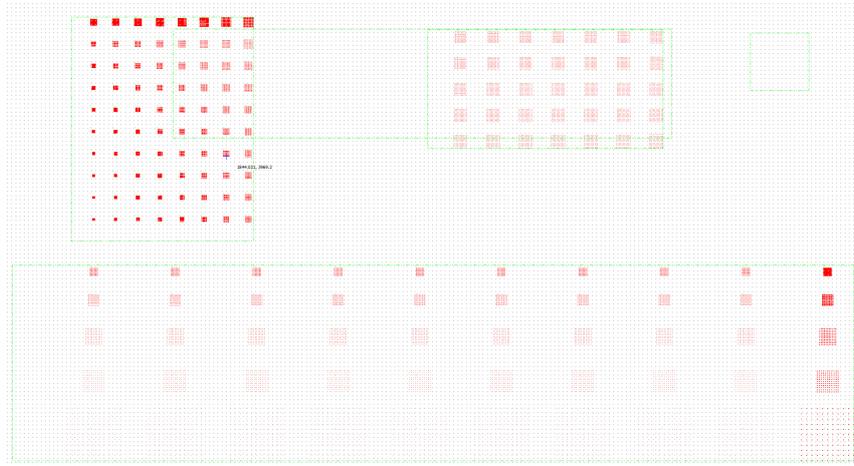


Figure 47: Design of the SiOx mask through EBL

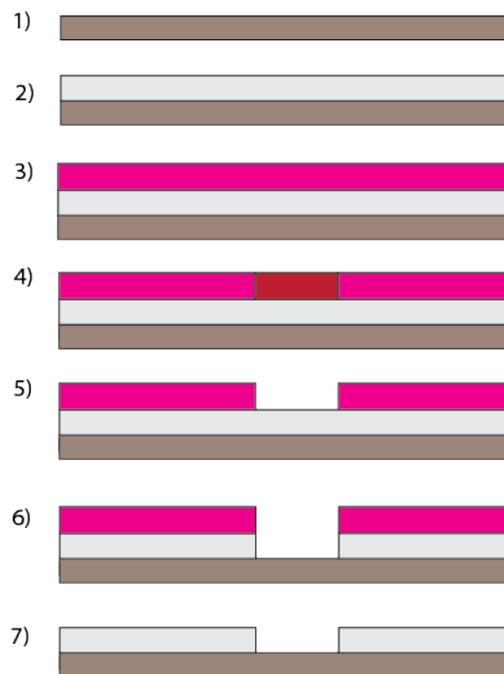


Figure 48: Fabrication Process. 1) The bare GaAs (100) 2) 10nm layer of SiOx deposited. 3) Spin-resist 4) EBL 5) Development 6) Etching 7) Strip-off

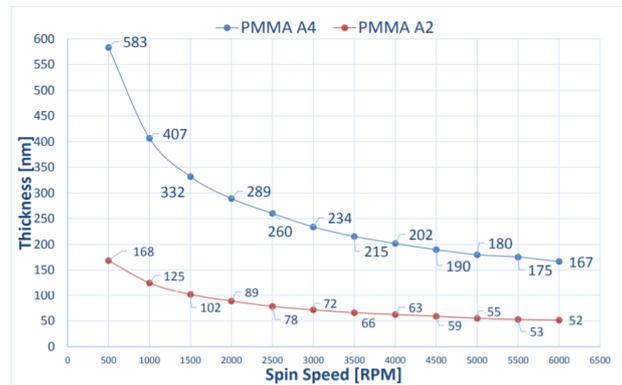


Figure 49: Graph showing the resist thickness of PMMA A4 and PMMA A2 for different spin speeds.

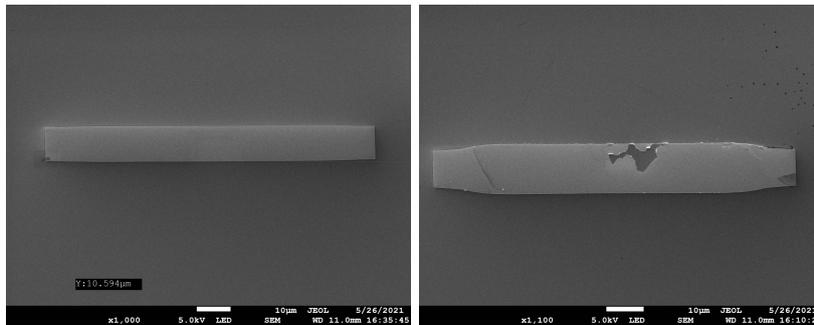


Figure 50: SEM images of Gold deposition guide marks. Left) $600 \mu\text{C}/\text{cm}^2$ was applied. Right) $800 \mu\text{C}/\text{cm}^2$ was applied, and the shape is overexposed

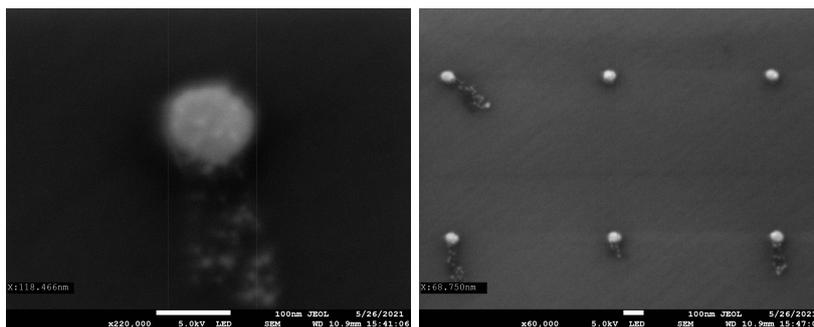


Figure 51: Metal deposited after development, to intermediately assess the resolution before wet etching

7.2.4 HF Etching

To make the openings in the deposited SiO_x and define the design, hydrofluoric acid, HF, is used. HF does not etch GaAs (or only with a miniscule etch rate), so it is an ideal etchant to use for the SiO_x mask. A disadvantage is that it is an isotropic etch, which results in even the unexposed SiOx will be etched slightly. So the openings will be etched laterally as well.

In order to calibrate etch time, the etch rate is estimated by measuring the thickness of the the deposited SiO_x and comparing thickness before and after in an ellipsometer. An obstacle occurred here, as the oxide height measured with the ellipsometer did not agree with the selected deposition and with the verification in AFM. In general: the ellipsometer would measure 20% higher. Furthermore, when two different etch times were compared, they would give different etch rates. To obtain the best estimate, some assumptions were made. First: the ellipsometer is off by a constant factor $h_{SiO_x,AFM} = h_{SiO_x,Ellipsometer} \cdot k_1$. Second: the error grows with longer etch times, since the SiOx layer approaches a critical point, so from two etch rate measurements with different etch times, an 'instantaneous' etch rate can be approximated: $\epsilon_0 = \epsilon_1 - \frac{\Delta\epsilon_{12}}{\Delta t}$. HF-etching has the advantage of H-passivisation. Which means that the dangling bond of the exposed GaAs openings, will be passivized by hydrogen bonds, and will not form native oxide right away.

7.2.5 Strip-off

After etching, the spin resist is removed through what is called strip-off. Solvents used includes acetone, isopropane alcohol, and MQ water sequentially inserted in a sonicator bath. Then 4 minutes of plasma ashing to remove any residing resist from the sample. Although this part seems straightforward, an enormous amount of time was spent dealing with impurities on samples. At some point the oxidizer *Piranha* was tried, as numerous samples came out with inexplicable and solvent-resistant impurities.

7.3 List of Samples

During this master's project, a number of GaAs samples have been used. For growth and to optimize fabrication processes. In total, six growths were carried out. Each succession

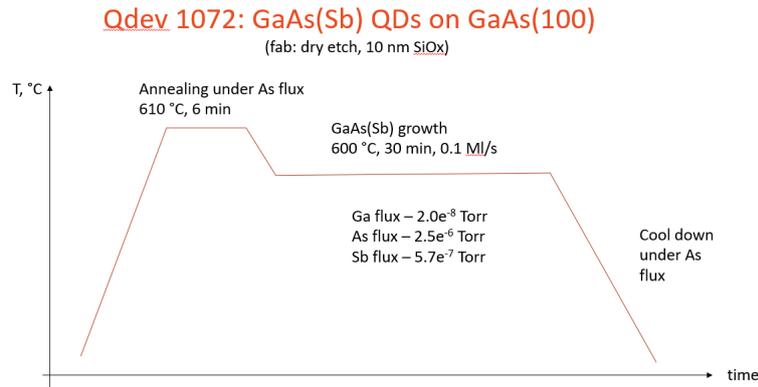


Figure 52: Grow schedule for QDev1072

allowed for a tweaking of growth parameters to obtain the desired small-opening-high-pitch growth. It needs to be stressed, that many more growths are needed to really attain the optimal conditions. The nomenclature for the different samples set by Daria Besnasiuk, was used.

7.3.1 QDev1072

While still working on producing a clean pre-growth fabrication, a first growth was included in a chip from Daria Besnasiuk. The fabrication of this chip was entirely carried out by Daria.

7.3.2 QDev1099

For this sample, the EBL was carried out by Daria Besnasiuk, since the pre-growth fabrications still had too many impurities for MBE growth. The sample was wet-etched with HF. QDev1099 is the main sample for GaAs growth analysis.

7.3.3 QDev1104

In this sample the temperature was reduced to increase growth in sites that would otherwise be desorption sites. In particular, the widths of 80 and 100nm with pitch around 1 μm was in focus. The reduced temperature caused the growth to leave the SAG window and allow for oxide growth, Figure 54.

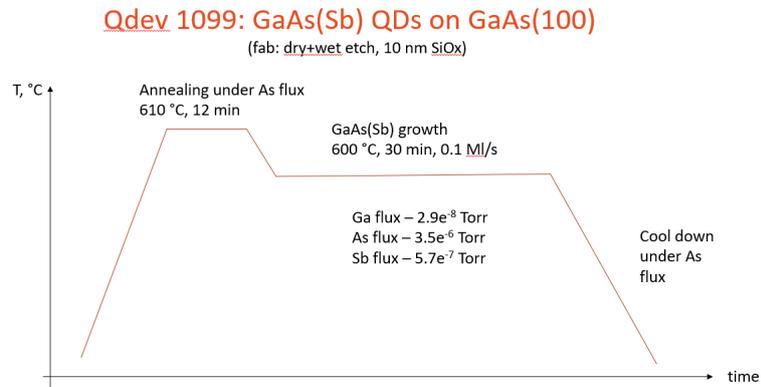


Figure 53: Grow schedule for QDev1099

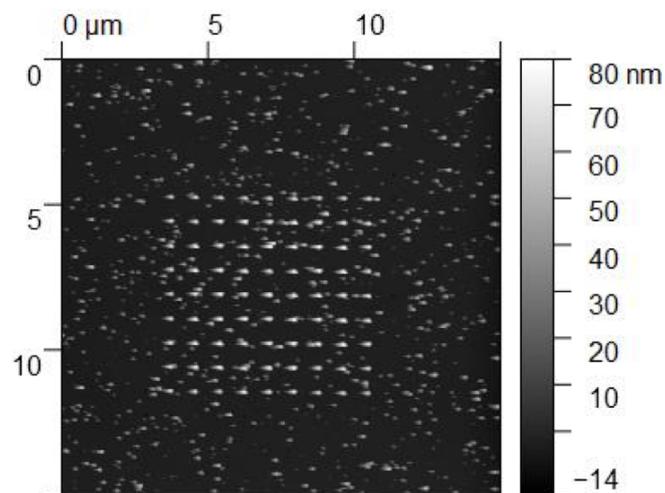


Figure 54: AFM image of an array, where GaAs has nucleated at the oxide

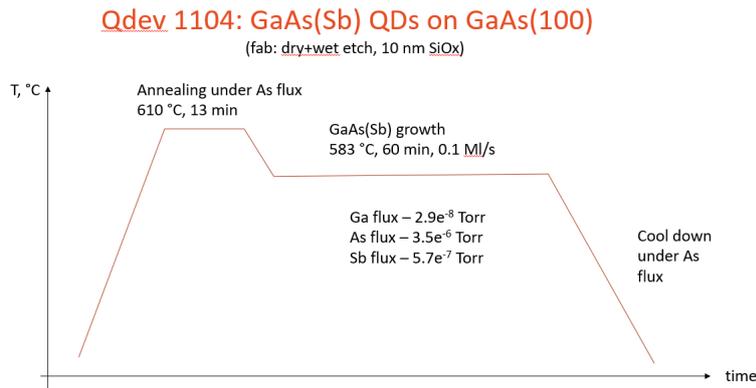


Figure 55: Grow schedule for QDev1104

7.3.4 QDev1121

For this sample, the slow growth was performed without leaving the SAG window. The limit for small openings does not change much compared with QDev1099, however. The growth for the smaller structures have even higher source effects. This is visible from the outer rows of openings in the arrays, where the openings, desorb material and leave pits $>50\text{nm}$ deep. In this sample, the effect stretches over the whole 11×11 array, and does not have stable center of structures. It also does not have many intermediately growths - the openings either give rise to overgrowth or desorption.

Another feature of the slower growth is that the big guide marks in the sample facet into pyramidal shapes. This is a sign that it is closer to an ECS from the increased time for energy minimization as seen in Figure 57.

7.3.5 QDev1132

This sample is the first InAs growth. The GaAs settings from QDev1121, though not optimised, has been kept the same. 12.5 monolayers, (or 7.5 nm) of InAs is grown.

7.3.6 QDev1142

This last sample has the same growth parameters as QDev1132 and is then capped with 100 nm GaAs. The capping layer serves to protect the InAs QDs from oxidation or proximity to surface impurities that can quench radiative recombination.

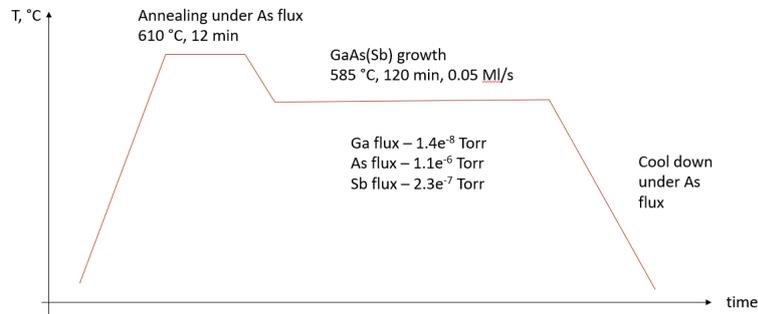
Qdev 1121: GaAs(Sb) QDs on GaAs(100)(fab: wet etch, 10 nm SiO_x)

Figure 56: Grow schedule for QDev1121

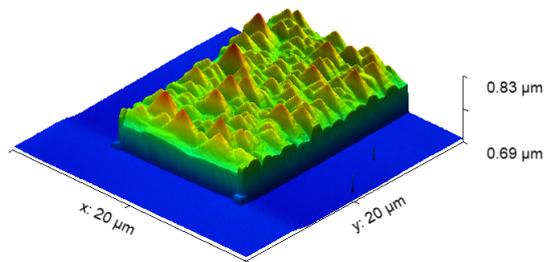


Figure 57: AFM of a rectangular guidemark of QDev1121. The slower growth enabled faceting of the surface into pyramidal shapes

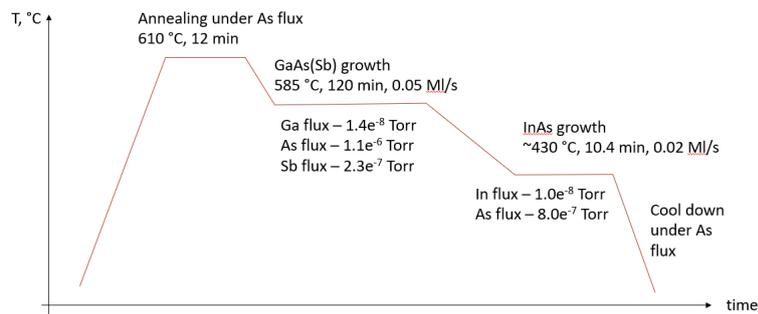
Qdev 1132: InAs/GaAs(Sb) QDs on GaAs(100)(fab: wet etch, 10 nm SiO_x)

Figure 58: Grow schedule for QDev1132

Qdev 1142: InAs/GaAs(Sb) QDs on GaAs(100) with GaAs capping layer

(fab: wet etch, 10 nm SiO_x)

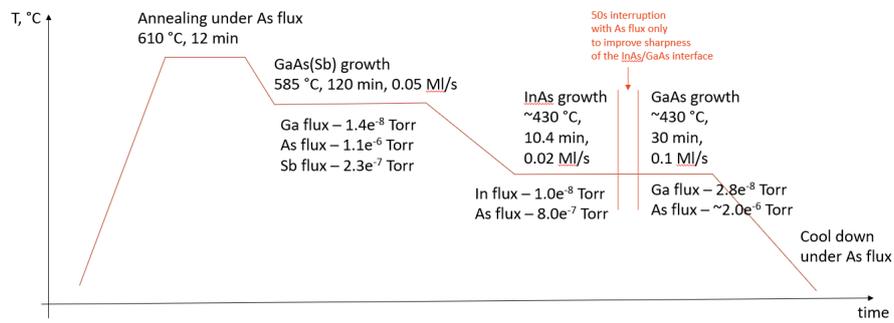


Figure 59: Grow schedule for QDev1142

7.4 Python Scripts

In this section, the most important scripts are reported. This includes the constrained Wulff simulations, the central part of the AFM analysis, and the SEM analysis script.

PythonScripts

October 14, 2021

```
[ ]: #constrained Wulff Construction

#script to implement SLSQP

from numpy.linalg import matrix_rank
import numpy as np
import copy
import math
import matplotlib.patches as patches
import pylab
import pylab as pl
import time

from scipy.spatial import ConvexHull
from scipy.optimize import minimize

import mpl_toolkits.mplot3d as a3
import matplotlib.colors as colors
from mpl_toolkits import mplot3d
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

#octagon opening width
opening=1
#facet index under opening constraint; 1,2,3,4 are 100 - 6,7,8,9 are 110 of the
↪ opening
boundary_ind=[1,2,3,4,6,7,8,9]

#http://www.ambrsoft.com/TrigoCalc/Plan3D/3PlanesIntersection_.htm

ff100=[[0,0,1],[0,0,-1],[1,0,0],[-1,0,0],[0,1,0],[0,-1,0]]

ff110=[[1,1,0],[-1,1,0],[1,-1,0],[-1,-1,0],[1,0,1],[-1,0,1],[0,1,1],[0,-1,1]]
for i in range(len(ff110)): ff110[i]=ff110[i]/np.sqrt(2)

ff111A=[[1,1,1],[-1,-1,1]]
```

```

ff111B=[[1,-1,1],[-1,1,1]]
for i in range(len(ff111A)): ff111A[i]=ff111A[i]/np.sqrt(3)
for i in range(len(ff111B)): ff111B[i]=ff111B[i]/np.sqrt(3)

ff113A=[[1,1,3],[-1,-1,3]]
ff113B=[[-1,1,3],[1,-1,3]]
for i in range(len(ff113A)): ff113A[i]=ff113A[i]/np.sqrt(11)
for i in range(len(ff113B)): ff113B[i]=ff113B[i]/np.sqrt(11)

SED_ff100=1.14
SED_ff110=1
SED_ff111A=1.06
SED_ff111B=1.25
SED_ff113A=1.12
SED_ff113B=1.13
#SED_list=np.
↳array([SED_ff100,SED_ff100,SED_ff110,SED_ff111A,SED_ff111B,SED_ff113A,SED_ff113B])↳
↳#100 twice due to free top facet

vector=[]
SED=[]
for i in range(len(ff100)): vector.append(ff100[i]); SED.append(SED_ff100)
for i in range(len(ff110)): vector.append(ff110[i]); SED.append(SED_ff110)
for i in range(len(ff111A)): vector.append(ff111A[i]); SED.append(SED_ff111A)
for i in range(len(ff111B)): vector.append(ff111B[i]); SED.append(SED_ff111B)
for i in range(len(ff113A)): vector.append(ff113A[i]); SED.append(SED_ff113A)
for i in range(len(ff113B)): vector.append(ff113B[i]); SED.append(SED_ff113B)

SED_interface=-1.14
SED.append(SED_interface)

def area2(poly0):
    if poly0=='NoFacet' or len(poly0)<3:
        return 0
    poly=np.copy(poly0)
    area=0
    for i in range(len(poly)-2):
        v1= np.array(poly[i+1])-np.array(poly[0])
        v2= np.array(poly[i+2])-np.array(poly[0])
        area+=np.linalg.norm(np.cross(v1,v2))/2
    return area

def proj(a,b): #a onto b
    return np.dot(a,b)*np.array(b)/np.dot(b,b)

def columnswap(u,m0,col):
    k=copy.deepcopy(u); m=np.copy(m0)

```

```

    #print(u); print(m); print(col)

    for s in range(3):
        k[s][int(col)]=m[s]
    return k

def zeroremove(x): #removes remaining [0,0,0]'s in the array
    u=[]
    for i in range(len(x)):
        zeros='true'
        for j in range(3):
            if x[i][j]!=0:
                zeros='false'
        if zeros=='false':
            u.append(x[i])
    return np.array(u)

def pointsorter(pp,b): #orders the point into a convex shape,'pp' is the array
    ↪ of points, b is the normal vector.
    if b[0]!=0: a=0
    elif b[1]!=0: a=1
    elif b[2]!=0: a=2
    if len(pp)!=0:
        cent=(sum([p[(1+a)%3] for p in pp])/len(pp),sum([p[(2+a)%3] for p in
    ↪ pp])/len(pp))# compute centroid
        pp.sort(key=lambda p: math.atan2(p[(2+a)%3]-cent[1],p[(1+a)%3]-
cent[0])) # sort by polar angle

def calcSurfaceEnergy(x): #input is the 'faces' 3rd order matrix

    surface_energy=sum(area2(x[i])*SED[i] for i in range(len(x)))
    print('calcSurfaceEnergy',surface_energy, 'interface area:
    ↪ ',area2(x[-1]),'other:
    ↪ ',area2(x[0]),area2(x[1]),area2(x[2]),area2(x[3]),area2(x[4]),area2(x[5]),area2(x[6]))
    return surface_energy

def calcVolume(x):
    vertices=[]
    for i in range(len(x)):
        if x[i]!='NoFacet':
            for j in range(len(x[i])):
                vertices.append(x[i][j])
    volume=ConvexHull(vertices).volume
    print('calcVolume',volume)
    return volume #tolist()

```

```

#all 3-plane intersections
counter=0.
def generate_shape(DispL_0): #the shaping becomes a function of the vector of
↳facet family displacements from center
    global counter; print(counter);counter+=1
    DispL=[]
    DispL.append(DispL_0[0])
    for i in range(1,len(ff100)): DispL.append(DispL_0[1])
    for i in range(len(ff110)): DispL.append(DispL_0[2])
    for i in range(len(ff111A)): DispL.append(DispL_0[3])
    for i in range(len(ff111B)): DispL.append(DispL_0[4])
    for i in range(len(ff113A)): DispL.append(DispL_0[5])
    for i in range(len(ff113B)): DispL.append(DispL_0[6])

    DispL=np.array(DispL)

    print(tuple(DispL_0))

    grandlist=np.zeros((len(vector),len(vector)**2,3))
    for k in range(len(vector)):
        for i in range(len(vector)):
            for j in range(i,len(vector)):
                #checking for point intersection
                #if (np.dot(vector[i],vector[j])>=-0.001 or i==1 or j==1) and
↳(np.dot(vector[i],vector[k])>= -0.001 or i==1 or j==1) and (np.
↳dot(vector[k],vector[j])>=-0.001 or i==1 or j==1) and np.linalg.
↳det([vector[k],vector[i],vector[j]])!=0 and
↳matrix_rank([vector[k],vector[i],vector[j]])==3 and matrix_rank([np.
↳append(vector[k],DispL[k]),np.append(vector[i],DispL[i]),np.
↳append(vector[j],DispL[j])])==3: #endret 7. maj
                if (    (np.dot(vector[i],vector[j])>=0.001 or i==1 or j==1 or
↳k==1)
                    and (np.dot(vector[i],vector[k])>= 0.001 or i==1 or j==1 or
↳k==1)
                    and (np.dot(vector[k],vector[j])>=0.001 or i==1 or j==1 or
↳k==1)
                    and np.linalg.det([vector[k],vector[i],vector[j]])!=0
                    and matrix_rank([vector[k],vector[i],vector[j]])==3
                    and matrix_rank([np.append(vector[k],DispL[k]),np.
↳append(vector[i],DispL[i]),np.append(vector[j],DispL[j])])==3 ):

                    R_c=np.array([vector[k],vector[i],vector[j]]);

```

```

        Det=np.linalg.det(R_c)
        d_vector=np.array([DispL[k],DispL[i],DispL[j]])

        xR_c=columnswap(R_c,d_vector,0); x_coord=np.linalg.
↪det(xR_c)/Det  #x coordinate
        yR_c=columnswap(R_c,d_vector,1); y_coord=np.linalg.
↪det(yR_c)/Det  #y coordinate
        zR_c=columnswap(R_c,d_vector,2); z_coord=np.linalg.
↪det(zR_c)/Det  #z coordinate

        intersection=[x_coord,y_coord,z_coord]
        #checking if intersection is in the Wulff shape
        actualvertex = 'true'
        for g in range(len(vector)):
            if np.dot(intersection,vector[g])>0 and np.linalg.
↪norm(proj(intersection,vector[g]))-0.01>DispL[g]:
                actualvertex = 'false'
        if actualvertex == 'true':
            grandlist[k][i*len(vector)+j]=intersection

faces=[]
interface=[]
grandlist0=np.copy(grandlist)
for p in range(len(vector)): #Winterbottom part
    face=zeroremove(grandlist0[p]).tolist()
    if len(face)==0:
        faces.append('NoFacet')
    if len(face)!=0:
        above='false'
        below='false'
        for i in range(len(face)):
            if face[i][2]<0: below='true'
            if face[i][2]>0: above='true'
        if above=='true' and below=='false':
            pointsorter(face,vector[p]); faces.append(face)
        elif above=='false' and below=='true':
            faces.append('NoFacet')
        elif above=='true' and below=='true':
            pointsorter(face,vector[p])
            winterbottomface=[]
            for i in range(len(face)):
                if face[i][2]>0:
                    winterbottomface.append(face[i])
                elif face[(i+1)%len(face)][2]>0:
                    interfacevertex=face[i]-(np.
↪array(face[(i+1)%len(face)])-np.array(face[i]))*face[i][2]/
↪(face[(i+1)%len(face)][2]-face[i][2]))

```

```

        winterbottomface.append(interfacevertex); interface.
↪append(interfacevertex.tolist())

        elif face[(i-1)%len(face)][2]>0:
            interfacevertex=face[i]-(np.
↪array(face[(i-1)%len(face)])-np.array(face[i]))*face[i][2]/(face[(i-
1)%len(face)][2]-face[i][2])
            winterbottomface.append(interfacevertex); interface.
↪append(interfacevertex.tolist())
        faces.append(winterbottomface)

    pointsorter(interface,vector[0])
    faces.append(interface)
    return faces
#https://en.wikipedia.org/wiki/Line%E2%80%93plane_intersection

def objective(x):
    faces_o=generate_shape(x)
    return calcSurfaceEnergy(faces_o)

startguess=[1.07716505, 1 ,          1 ,          1.00689432, 1.2247463, 1.
↪07015091, 1.10040266] #solution to V=2.5
Volumes=np.arange(2.5,1.9,-0.1)
lower=0.9; upper= 1.5 #set it by half min SEDlist and double max SEDlist
bnds = ((0.01,upper), (lower,1), (lower,1), (lower,upper), (lower,upper),
↪(lower,upper),(lower,upper))
gstep=0.1 #the growth step, and the bounds for next size

solution_array=[]

for Volume in Volumes:
    def constraint(x):
        faces_=generate_shape(x)
        return calcVolume(faces_)-Volume

    cons = ({'type': 'eq', 'fun':constraint})

    sol =
↪minimize(objective,startguess,method='SLSQP',constraints=cons,bounds=bnds,options={'disp':
↪True,'ftol': 1e-03, 'maxiter': 50 }) #maybe change to 30
    vectorOpt = sol.x
    SEOpt=sol.fun
    globals()['displacement'+str(round(Volume*100))[0:3]]=vectorOpt
    solution_array.append(vectorOpt)

```

```

startguess=vectorOpt #update the following startguess
bnds_arr=np.transpose(np.array([np.array(vectorOpt)*(1-gstep),np.
↪array(vectorOpt)*(1+gstep)]));
bnds_arr[1][1]=bnds_arr[2][1]=1 #last modification to meet opening bounds
bnds=tuple(map(tuple,bnds_arr))

np.save('WulffSolutions'+time.strftime("%Y-%m-%d-%H:%M"),np.
↪array(solution_array)) #save the solutions to file

```

```

[ ]: #AFM calibration and Analysis

def find_rotation(img, center, search_range):
    thetarange = np.linspace(center-search_range, center+search_range, 20)
    radTransform = skimage.transform.radon(img, theta=thetarange)
    argmax = np.argmax(np.var(radTransform, axis=0)) # Maximal variance
    ↪indicates dominant axis
    detected_center = thetarange[argmax]

    if search_range < 0.01:
        return (detected_center, 2*search_range/10)
    else:
        return find_rotation(img, detected_center, 2*search_range/10)

def zero_plane(data, sanity_check=True):

    data_rough_shift = zero_plane_rough(data)
    data_fine_shift, std = zero_plane_fine(
        data_rough_shift, sanity_check=sanity_check)

    return data_fine_shift, std

def zero_plane_rough(data):
    bin_size = 0.1
    hist, bins = np.histogram(data, bins=np.arange(
        np.amin(data), np.amax(data), bin_size), normed=True)

    bin_center = (bins[:-1] + bins[1:]) / 2

    # Define zero as histogram maximum. This assumes that the substrate is the
    ↪dominant feature in the image, and there are not TOO many crystallites
    # Find histogram maximum position
    mean_offset = bin_center[np.argmax(hist)]

    # Initial shift

```

```

shifted_data = data - mean_offset

return shifted_data

def zero_plane_fine(data, sanity_check=True):
    bin_size = 0.1
    zero_bound = 10

    zdata = data[(data < zero_bound) & (data > -zero_bound)]

    hist, bins = np.histogram(zdata, bins=np.arange(
        np.amin(zdata), np.amax(zdata), bin_size), normed=True)

    bin_center = (bins[:-1] + bins[1:]) / 2

    dbl_gauss, params = prep_double_gaussian()

    fit = dbl_gauss.fit(hist, params, x=bin_center)

    if sanity_check is True:
        fig, gridspec = fit.plot(data_kws={'markersize': 1})
        plt.show()

    # Choose appropriate gaussian by scale factor
    if fit.best_values['g1_amplitude'] > fit.best_values['g2_amplitude']:
        corr_data = data - fit.best_values['g1_center']
        return corr_data, fit.best_values['g1_sigma']
    else:
        corr_data = data - fit.best_values['g2_center']
        return corr_data, fit.best_values['g2_sigma']

def line_level3(img,ldr=5): #good for fab analysis
    #ldr is line defect range, and must exceed the line defects of the scan
    img_copy = img.copy()

    counts,binedges=np.histogram(img_copy.flatten(),bins=50); bins=(binedges[1:
↵] + binedges[:-1])/2

    peak=bins[np.argmax(counts)];

    for i in range(img.shape[0]):
        line = img[i, :]

```

```

n_bins=50
counts,binedges=np.histogram(line,bins=n_bins,range=(peak-ldr,peak+ldr))
bins=(binedges[1:] + binedges[:-1])/2

peak_index=np.argmax(counts)
bin_size= ( line.max()-line.min() )/n_bins
bin_range=int(round(2/bin_size)) #choose a range of 2 nm around peak

lower_range=max(peak_index-bin_range,0) ;
↪upper_range=min(peak_index+bin_range,n_bins-1)

offset=sum(bins[lower_range:upper_range]*counts[lower_range:
↪upper_range])/sum(counts[lower_range:upper_range])
levelled_line = line - offset

img_copy[i, :] = levelled_line

return img_copy

def dot_selection3(image, cfactor, threshold=2.5,ox_depth=10.5): #including
↪openings without growth, and removes small regions

    #oxide depth included, check up later !!!

    zeroed_bkg=np.copy(image)
    zeroed_bkg[abs(zeroed_bkg)<threshold]=0

    s = generate_binary_structure(2,2)
    labeled_array,n_features=label(zeroed_bkg,structure=s)
    r_prop = regionprops(labeled_array, intensity_image=image)

    area_thresh=max([r_prop[r].area for r in range(len(r_prop))])/2 # changed
↪from median area to max area, in order to filter

    num=np.arange(len(r_prop)); inds=[j for j in num if r_prop[j].
↪area>area_thresh ]
    #and r_prop[j].area<5 potential limit

    dot_count = len(inds)
    areas = [r_prop[r].area*(cfactor**2) for r in range(len(r_prop)) if r in
↪inds]
    # WARNING: unclear if max_intensity is scaled from [0-1]. If so we need
↪scale factor
    heights = [r_prop[r].max_intensity+ox_depth for r in range(len(r_prop)) if
↪r in inds]

```

```

    local_heights = [util.local_height_map(image, r_prop[r]) for r in
↪range(len(r_prop)) if r in inds]
    # TODO: Does this even make sense
    volumes = [np.nansum(a)*(cfactor**2) for a in local_heights]
    for i in range(len(areas)):
        volumes[i]+= areas[i]*ox_depth

    centroids = [r_prop[r].centroid for r in range(len(r_prop)) if r in inds]

    return dot_count, centroids, areas, heights, volumes, labeled_array, inds

```

[]: *#Facet Analysis*

```

def facet_distribution(img,cfactor,dot_count,hist_range=2):
    xgrad,ygrad=np.gradient(img); xgrad/=1000*cfactor; ygrad/=1000*cfactor
    threshold=5
    zeroed_bkg=np.copy(img)
    zeroed_bkg[zeroed_bkg<threshold]=0
    s = generate_binary_structure(2,2)
    labeled_array,n_features=label(zeroed_bkg,structure=s)
    print(n_features)
    #region_properties = regionprops(labeled_array, intensity_image=xgrad) #is
↪this used?

    structure_xgrad=[]
    structure_ygrad=[]
    for i in range(dot_count):
        globals()['xlist'+str(i).zfill(3)]=[]
        globals()['ylist'+str(i).zfill(3)]=[]

    for i in range(len(img)):
        for j in range(len(img[i])):
            if labeled_array[i][j]!=0:
                globals()['xlist'+str(labeled_array[i][j]-1).zfill(3)].
↪append(xgrad[i][j])
                globals()['ylist'+str(labeled_array[i][j]-1).zfill(3)].
↪append(ygrad[i][j])
                structure_xgrad.append(xgrad[i][j])
                structure_ygrad.append(ygrad[i][j])
    top_facet_area_list=[]

    facet101_area_list=[]; facet011_area_list=[];facet_101_area_list=[];
↪facet0_11_area_list=[]
    ff110=[facet101_area_list,
↪facet011_area_list,facet_101_area_list,facet0_11_area_list]

```

```

    for i in range(dot_count):
        bins,xedges,yedges=np.histogram2d(globals()['xlist'+str(i)].
↪zfill(3)],globals()['ylist'+str(i)].zfill(3)],bins=50,range=[[-hist_range,
↪hist_range], [-hist_range, hist_range]])
        '''
        if i in np.arange(40,50): #show the individual structures
            plt.hist2d(globals()['xlist'+str(i)].
↪zfill(3)],globals()['ylist'+str(i)].zfill(3)], vmax=25,bins=50,range=[[-
hist_range, hist_range], [-hist_range, hist_range]])
            plt.colorbar()
            plt.show()'''

        xpos=25#unravel_index(abs(xedges).flatten().argmin(), xedges.shape)
        ypos=25#unravel_index(abs(yedges).flatten().argmin(), yedges.shape)
        #print(xpos,ypos)
        sum_r=2 #radius of summed bins (0 radius -> 0 bins, 2 radius -> 16 bins
↪etc)

        top_facet_area_list.append(np.sum(bins[xpos-sum_r:xpos+sum_r,ypos-sum_r:
↪ypos+sum_r]))

        ind110=np.array([[ -1, -1], [1, -1], [1, 1], [ -1, 1]])
        for j in range(4): # there is 2 / 25 slope pr bin and 25 / 2 bin pr
↪slope
            xpos=int(round(25+ind110[j][0]/np.sqrt(2)*25/2 ));
            ypos=int(round(25+ind110[j][1]/np.sqrt(2)*25/2 )) # start 25+
↪sign*slope*range/n_bins

            ff110[j].append(np.sum(bins[xpos-sum_r:xpos+sum_r,ypos-sum_r:
↪ypos+sum_r]))

    return top_facet_area_list, ff110, [structure_xgrad,structure_ygrad]

```

```

[ ]: import cv2
import numpy as np
import matplotlib
from matplotlib.pyplot import imshow
from matplotlib import pyplot as plt

from scipy.ndimage import label, generate_binary_structure

#img= cv2.imread('200nm600nmp.03.bmp',0)
img= cv2.imread('2.bmp',0)

```

```

img=img[0:int(len(img)*0.9),:]
img=img[50:400,250:650]
plt.imshow(img);plt.show()
print('input',img[0][0])

k_size=10
kernel=np.ones((k_size,k_size))/k_size**2
for i in range(5):
    img = cv2.filter2D(img,-1,kernel)

plt.imshow(img);plt.show()

ygrad,xgrad=np.gradient(img)

ygrad,xgrad = cv2.filter2D(ygrad,-1,kernel),cv2.filter2D(xgrad,-1,kernel)

yxgrad,xxgrad=np.gradient(xgrad)
yygrad,xygrad=np.gradient(ygrad)
curvature=np.sqrt(xxgrad**2 + yygrad**2)

plt.imshow(curvature);plt.show()

plt.hist(curvature.flatten());plt.show()
curvature=np.array(curvature)
print(len(curvature.flatten()))
thr=0.09
curvature[curvature>thr]=1000
curvature[curvature<thr]=1
plt.imshow(curvature);plt.show()

curvature=cv2.filter2D(curvature,-1,kernel)
plt.imshow(curvature);plt.show()

curvature[curvature>100]=0
s = generate_binary_structure(2,2)
labeled_array,n_features=label(curvature,structure=s)
plt.imshow(labeled_array);plt.show()
print(n_features)

```