

ASSESSING A MULTI-ELECTRON BEAM

APPLICATION APPROACH FOR

SEMICONDUCTOR PROCESS METROLOGY

By

Maseeh Mukhtar

Copyright © 2018

ASSESSING A MULTI-ELECTRON BEAM

APPLICATION APPROACH FOR SEMICONDUCTOR

PROCESS METROLOGY

A DISSERTATION PRESENTED

BY

MASEEH MUKHTAR

TO

THE COLLEGES OF NANOSCALE SCIENCE AND ENGINEERING

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE SUBJECT OF

NANOSCALE SCIENCE

STATE UNIVERSITY OF NEW YORK POLYTECHNIC INSTITUTE

ALBANY, NEW YORK

USA

2018

DISSERTATION COMMITTEE MEMBERS

Committee Chair: Bradley Thiel, Ph.D.

Head of the Nanoengineering Constellation

Professor of Nanoscale Science

SUNY Polytechnic Institute

Abner Bello, Ph.D.

Senior Section Manager (Advanced Module Engineering)

GlobalFoundries, Malta, NY

Alain C. Diebold, Ph.D.

Empire Innovation Professor of Nanoscale Science

SUNY Polytechnic Institute

Nathan Cady, Ph.D.

Associate Professor of Nanobioscience

SUNY Polytechnic Institute

Robert Geer, Ph.D.

Professor of Nanoscale Science

SUNY Polytechnic Institute

Woongje Sung, Ph.D.

Associate Professor of Nanoscale Engineering

SUNY Polytechnic Institute

ABSTRACT

Radical and disruptive technological approaches regularly require experimental prototypes be built, which is difficult to justify considering their oft-prohibitive requirements in terms of financial and/or time commitments. It is also frequently the situation that use cases for new technologies are not entirely worked out precisely which in turn make it even more difficult to build prototypes but the analysis of simulation data sets from virtual samples can be used to predict sensitivity to the devised signal, detection limits, and impact of design rules and material sets. The results can thus be used to guide prototype design. The aim of this work is to develop and demonstrate a predictive approach to technology assessment and prototype design. This work will focus on two such disruptive technology concepts: electron beam defect inspection and critical dimension measurement. These two concepts are based on the transfer from conventional process metrology technologies i.e., brightfield inspection and optical critical dimension scatterometry to multi-electron beam approaches.

Here, a multi-scale modeling approach is used to simulate data streams nominally generated by virtual tools inspecting virtual wafers. To this end, Java Monte Carlo Simulator for Secondary Electrons (JMONSEL) simulations are used to generate expected imaging responses of chosen test cases of patterns and defects with ability to vary parameters for beam energy, spot size, pixel size, and/or defect material and form factor. The patterns are representative of the design rules for aggressively-scaled FinFET-type designs. With these

simulated images and resulting shot noise, a signal-to-noise framework is developed, which relates to defect detection probabilities. Additionally, with this infrastructure the effect of detection chain noise and frequency dependent system response can be made, allowing for targeting of best recipe parameters for multi-electron beam inspection validation experiments. Ultimately, leading to insights into how such parameters will impact tool design, including necessary doses for defect detection and estimations of scanning speeds for achieving high throughput for high-volume manufacturing. Simulated images are also executed for measurement of critical dimensions of the abovementioned class of FinFETs. Similarly, validation experiments for multi-electron critical dimension measurements may use the information extracted for development of volume manufacturing metrology systems.

KEYWORDS: Critical dimension (CD), Defect detection, Dimensional metrology, Electron beam inspection (EBI), Java Monte Carlo simulator for Secondary Electrons (JMONSEL), Massively parallel, Multiple electron beam, Multi-column, Scanning electron microscopy (SEM), Wafer inspection

Dedicated to my brother, Edrees.

ACKNOWLEDGEMENTS

There are many people to whom I would like to express my appreciation for having assisted me during my graduate training. To be noted, this is by no means a comprehensive list.

Firstly, I would like to recognize my doctoral advisor, Prof. Brad Thiel for his invaluable support and guidance throughout the entirety of my graduate education. I'd like to thank Prof. Unni Pillai for his advisement during my Master's degree, Ben Bunday for his assistance and collaboration on portions of the work produced herein, and Matt Malloy for his assistance during my early days as a student research assistant for SUNY Poly SEMATECH. I'd like to extend thanks to the members of my committee for their review, suggestions, and critiques to strengthen this work especially Abner Bello for perceptive consultations on the direction of parts of this project as well as material support.

I would like to thank John Villarrubia of NIST for his writing and continued support of the JMONSEL SEM simulator code, and also for insightful discussions on its use, on edge detection and model-based algorithms, and CD metrology and simulation studies in general. As well, I would like to thank Bryan Barnes and Richard Sliver of NIST for their creation of the defect sensitivity analysis code employed herein. Also, I extend my gratitude to the entire MultiSEM team at Carl Zeiss for their instrumental support in making this work possible.

Sincere thanks to Feryan Ahmad, Kathryn Quoi, Philip Williams, James Dillon, Matt Strohmayr, Mark Altwerger, Paul David & the rest of the CNSE graduate student populace for their camaraderie and assistance.

Not to be forgotten, a special thanks to my parents and the rest of my family for their irreplaceable support throughout my life's endeavors.

CONTENTS

Abstract.....	vi
Acknowledgements.....	ix
Contents	xi
List of Figures	xv
List of Tables.....	xx
List of Equations	xxi
List of Symbols & Acronyms	xxiii
1 Introduction	25
2 State of the Art/Literature Review	30
2.1 <i>Multiple Electron Beam Systems</i>	31
2.1.1 MULTI-AXIS	32
2.1.2 SINGLE COLUMN BY MULTI-SOURCE	34
2.1.3 SINGLE COLUMN BY SINGLE SOURCE.....	35
2.1.4 MULTI-COLUMN BY SINGLE SOURCE	36
2.2 <i>Multi-Electron Beam System Development</i>	37
2.3 <i>Defect Inspection</i>	39
2.3.1 LIMITATIONS OF OPTICAL INSPECTION OF WAFER DEFECTS	39

2.3.2	INSPECTION USING ELECTRON BEAMS	44
2.3.2.1	THROUGHPUT CALCULATION	49
2.4	<i>Critical Dimension Measurement</i>	55
2.4.1	CURRENT/EXISTING HVM CD METROLOGY SYSTEMS	56
2.4.1.1	Constraints of Current CD Metrology Systems	57
3	Research Objectives and Approach	63
3.1	<i>Defect Inspection</i>	65
3.1.1	RESEARCH OBJECTIVES	65
3.1.2	RESEARCH METHODS	65
3.1.2.1	GENERATION OF IDEAL AND DEFECT STRUCTURE	67
3.1.2.2	APPLYING IMAGING CONDITIONS & CALCULATION OF ELECTRON EMISSION BEHAVIOR	71
3.1.2.3	CREATION OF VIRTUAL IMAGES	75
3.1.2.4	ANALYSIS OF DEFECT SIGNATURE	78
3.1.2.5	Zeiss MultiSEM System	83
3.2	<i>Critical Dimension Measurement</i>	87
3.2.1	RESEARCH OBJECTIVES	87
3.2.2	RESEARCH METHODS	88
3.2.2.1	GENERATION OF IDEAL AND PROCESS DEVIATED STRUCTURE	88
3.2.2.2	APPLYING IMAGING CONDITIONS & CALCULATION OF ELECTRON EMISSION BEHAVIOR	90
3.2.2.3	CREATION OF VIRTUAL IMAGES	91
3.2.2.4	ANALYSIS OF CRITICAL DIMENSION DEVIATION SIGNATURES	92
4	Results	98
4.1	<i>Defect Inspection</i>	98
4.1.1	Virtual Experiment: Effect of Detector Dark Current on Defect Sensitivity	101
4.1.1.1	SENSITIVITY ANALYSIS OF REAL SAMPLES (61-BEAM SYSTEM PERFORMANCE)	105

4.2	<i>Critical Dimension Measurement</i>	111
4.2.1	Virtual Experiment: Effect of Detector Dark Current on Measurement Sensitivity.....	114
5	Conclusions	125
5.1	<i>Summary</i>	125
5.2	<i>Future Directions</i>	126
	References	129
	Appendix	140
A.	<i>The JMONSEL Simulator</i>	140
	References	145
B.	<i>Backscattered Electron Simulations to Evaluate Sensitivity against Electron Dosage of Buried Features</i>	147
	Introduction.....	147
	Objectives	148
	Methods.....	150
	Analysis development	151
	Results.....	154
	Conclusions	161
	References	162
C.	<i>Signal to Noise Ratio (SNR) Producing Script</i>	163
D.	<i>JMONSEL Simulation Script</i>	180

LIST OF FIGURES

Figure 1.1 Illustrative roadmap for CMOS logic transistors. Application of strained silicon, high-k/metal gate, and non-planar devices continues to enable the pace of advances in performance (adapted from [9], [10]).	28
Figure 2.1 (Left to Right) Schematics (non-scale) of the four general categories of multi-beam systems: Multi-axis systems; Single column, multi-source; Single column, single source; Multi-column, single source.	32
Figure 2.2 Schematic presenting a cross section of a gun assembly for a miniature column electron beam system. The gun assembly all together is on the order 40 mm with optical column length on the order of ten millimeters while typical working distance is a few millimeters (adapted from [29], [30]).	33
Figure 2.3 Critical size for particles on bare wafers for recent past and future production years. Data includes critical defect sizes for the ITRS technology nodes and related stopgap half-nodes (process technology where chip size is reduced without need for restructuring the circuit architecture to be suitable for a reduced area.)	40
Figure 2.4 Relative scattering strengths of critical sized defects/particles by production year scaled relative to a 266 nm light scattering from an 8 nm particle ($K = 1$) for various wavelengths of optical inspection.	43
Figure 2.5 Illustration of signal, contrast, and noise from a line scan over a feature S_i on a background S_0 .	45
Figure 2.6 Schematic illustrating the shot noise limited secondary electron emission probability of reference and defect containing pixels [93].	46
Figure 2.7 Signal to noise ratio dependence on the electron dosage per pixel at given detector quantum efficiencies.	48
Figure 2.8 Chart showing along the y-axis the ITRS 2014 based requirements for inspection throughput (cm^2/hr) in various settings: Process R&D at 300, Yield Ramp-up at 1,200, and Volume Production at 3,000. The x-axis communicates the inspection system type application space with regards to the size of critical defects in nanometers.	51
Figure 2.9 Using Eqn. 13 with following assumptions to show the general trends of throughput versus the number of beams: $C = 50\%$, $I_p = 1 \text{ nAmp}$, $\delta = 1$, $\phi = 0.5$, $\text{SNR} = 2$ and pixel size = defect size as denoted on trend lines in the graph.	53
Figure 2.10 General trends showing the interdependence of throughput and dwell time (in nanoseconds) given a certain number of beams and pixel size	53

Figure 2.11 Left – Cross-section illustration perpendicular to a fin. Right – Illustration of fundamental unit of a FinFET with indications of twelve important process control parameters.	55
Figure 2.12 Schematic illustration of the principle of tilt beam imaging. After the electron beam has entered the electron optical column, the first deflector repels the beam away from the optical axis. The second deflector pushes the beam back towards the optical axis. Finally, the objective lens focuses the beam on a point where the beam will impinge on the surface of a target sample. The electron beam arrives at a tilted angle when landing on the surface.	59
Figure 2.13 Illustration of (and equation for determining) limitation of maximal imaging tilt angle given the feature aspect ratio of a sample.	60
Figure 3.1 General process flow of the generation of virtual data of ideal and defect structures.	66
Figure 3.2 Schematic illustration of a test case of FinFET links	68
Figure 3.3 Process flow of structure generation from concept design, schematic illustration, and the 3D rendering of the structures.	69
Figure 3.4 Sample set of 3D rendered designs of the defect tiles.	70
Figure 3.5 Schematic illustration of an IDA map.	70
Figure 3.6 Illustrated example of an IDA map with rows representing defect types and columns indicating size of defect.	71
Figure 3.7 Illustration of the key input components for a JMONSEL simulation. The sample topography, characteristics of the impinging beam, and the scattering models are all set before calculation of electron yield data can be commenced.	73
Figure 3.8 Example of simulated electron trajectories from JMONSEL. The red dashed arrow points to the location of the center of a designated pixel, and incident electrons enter the sample vertically in a Gaussian distribution of a designated spot size centered on that location.	75
Figure 3.9 Procedural flow for generation of virtual image data, from sample definition to tiled image to image post processing, yielding a realistic, quantitatively representative image for a multi-electron beam inspection tool.	76
Figure 3.10 Sample grayscale image generated by converting the JMONSEL SE yields of a test structure (3 nm pixel size, 3 nm probe 1σ , 500 eV, 8 nm Si island) into a grayscale image (third step in Figure 3.9). The central defect cell block is magnified for clarity.	76
Figure 3.11 Process flow of the defect sensitivity algorithm. The algorithm starts by creating a target ROI in the center (red square—top left image) and four reference shifts of same size. Next, it correlates back the shifted images to the target ROI and then subtracts the average image from among the images from the shifted locations from the target ROI. In this case, we multiply the central square four times and subtract each of the four shifts so that the defect is equally weighted in the calculation. Finally, any uncommon overlap is cropped, leaving the differential image to be processed by the threshold parameters.	81

Figure 3.12 Algorithm ran on an example ideal simulated image with central defect. Identified defect is shown by the red star indicated on the binary mask of the differential image.	82
Figure 3.13 Schematic representation of the multi-beam imaging setup. Primary electrons (blue dotted lines) are focused onto a sample and separated by a beam splitter from the secondary electrons (red dotted lines) that are detected simultaneously by a multi-detector.	83
Figure 3.14 Hexagonal arrangement of beam layout; shown for a 61-beam configuration. ...	84
Figure 3.15 Essential workflow of the Zeiss multi-beam system.	85
Figure 3.16 Demonstrative workflow illustrated on a 22nm wafer sample [85].	85
Figure 3.17 Illustrative FinFET example of currently produced device architecture. (TEM image courtesy of GlobalFoundries).....	87
Figure 3.18 General process flow of the generation of virtual data of ideal and deviated structures.	88
Figure 3.19 Schematic illustration of a test case of basic fin structures from top-down and cross-sectional views.	89
Figure 3.20 Process flow of structure generation from concept design, schematic illustration, and the 3D renderings of the structures process varied critical dimensions.	89
Figure 3.21 Illustration of a Fin array being struck by electron beams at successively different angles.	91
Figure 3.22 Sample grayscale images generated by converting the JMONSEL SE yields of a test structure into a grayscale image. Left – cross section view of Fin (10,000 electron trajectories per pixel). Right – 45 degree angle view of Fins (1,000 electron trajectories per pixel).	92
Figure 3.23 Simple rectangular shape is tilted counterclockwise by an angle θ generating an edgewidth x that can be measured. The height of the shape is then calculated by simple trigonometry.....	93
Figure 3.24 Operating method of the tilt-beam imaging: target feature is scanned twice with an electron beam at two different incident angles (α_1 & α_2) creating two images with different edgewidths (EW_1 & EW_2). Calculation of feature height and SWA can be accomplished by the relation of the tilt angles and edgewidths.....	94
Figure 3.25 Diagram of re-entrant trapezoidal model with corresponding geometrical relationships.	96
Figure 3.26 Dependence of sidewall angle on the edgewidth to height ratio.	97
Figure 3.27 Sensitivity of the edgewidth to feature height and tilt angle.	97
Figure 4.1 Left – Comparison between experimental (top) and simulated (bottom) images of X-bridge defect on 30-nm pitch IDA pattern. Both images represent approximately the same dose as well as pixel size and probe diameter (FWHM). Right – Comparison of line intensity profiles across the grating (top) and bridge defect (bottom).....	99

Figure 4.2 Left – Comparison between experimental (top) and simulated (bottom) images of Y-bridge defect on 30-nm pitch IDA pattern. Both images represent approximately the same dose as well as pixel size and probe diameter (FWHM). Right – Comparison of line intensity profiles across the grating (top) and bridge defect (bottom).....	100
Figure 4.3 Virtual imaging processing flow for the defect sensitivity versus dark current virtual experiment.....	102
Figure 4.4 Defect SNR for X-bridge defects versus added white noise, resulting from X-bridge defect images of three different sizes being continually drenched with excess white noise. The small images underneath the graph show the defect region with only shot noise added on the left, and then with white noise added in steps of 2σ going to the right, for all three defect sizes as marked. Final data points in a curves show the point where the defect is last detected, i.e., giving a defect SNR. After that, the noise drowns out the defect enough that it does not pass the threshold parameters set previously therefore giving no defect SNR.....	103
Figure 4.5 Defect SNR for various defect types of 10 nm size versus added white noise, resulting from the defect images being continually drenched with excess white noise. Defects included are silicon and copper islands, X-bridges with the bridge being aligned to either the center or end of a fin link, and also the case of a 6 nm square mousebite out of a Si fin link. Final data points in a curves show the point where the defect is last detected, i.e., giving a defect SNR. After that, the noise drowns out the defect enough that it does not pass the threshold parameters set previously therefore giving no defect SNR.....	104
Figure 4.6 Images of intentional defects from a 28nm half-pitch wafer sample. The sample was imaged using a 3.76 nm pixel size, 50 ns dwell time and 1.5 keV landing energy.	105
Figure 4.7 Top – SNR through programmed defect size for a bridge between two lines ending in the y-direction, with increasing x dimension in 15nm line/space HSQ patterns on silicon. Bottom – programmed defect example images.	106
Figure 4.8 Top – SNR through programmed defect size for a bridge between two line edges in the x-direction, with increasing y dimension in 28nm line/space etched silicon patterns. Bottom – programmed defect example images.	107
Figure 4.9 Top – SNR through programmed defect size for a “mousebite” in the side of a line, with increasing y dimension in 28nm line/space etched silicon patterns. Bottom – programmed defect example images.	108
Figure 4.10 SNR through programmed defect size for a bridge between two lines ending in the y-direction, with increasing x dimension in 15nm line/space HSQ patterns on silicon. The top regression line represents the SNR trend for the sample imaged at 400 ns dwell time whereas the bottom line represents the sample at a 50 ns dwell time.....	109
Figure 4.11 Sensitivity analysis of defects from real image data from the 61-beam tool with 2 nm pixels at 1.5 keV beam, imaging the 15 nm line/space IDA from SUNY Poly SEMATECH. Y-bridges result in the highest SNR when larger, whereas X-bridges show the greatest variance; mid-link gaps have consistent SNR (i.e., signature is about constant); mousebites have lowest SNR when small, and island defects are on the lower end of SNR overall.	110

Figure 4.12 Top – Real (CD-SEM and SEM) and simulated images of a 0.5 μm x 0.5 μm FOV of the Malter region, respectively. Bottom – normalized plot profiles of several gratings using the abovementioned methods.....	112
Figure 4.13 TEM image of a section of the Malter region.	113
Figure 4.14 Left – sample height measurements of fin gratings with various metrology methods. Right – sample SWA measurement for those same gratings. Simulation and SEM tilt images were taken at 3 and 6 degree tilt angles.....	113
Figure 4.15 Normalized grayscale average profiles for a Si fin of height 25nm at four different angles i.e. 0, 5, 8, and 15 degrees.	114
Figure 4.16 Secondary electron profile plot as related to a cross-sectional and top down view of simulated fin gratings, respectively.	115
Figure 4.17 Width of projected edge versus tilt angle (in degrees) for various heights (in nanometers). Final plot (bottom right) shows the effect of pixel size on measurement of edgewidth versus height. Lines represent expected widths while the various points are measured edgewidths.	116
Figure 4.18 Left – measurement of height from single edgewidth from a single tilt angle i.e., 5, 8, or 15 degrees. Right – measurement of height using both the tilt angle measurements i.e., 5 & 8 and 5 & 15 degrees. Simulations implemented 10,000 electrons per pixel at 1 nm pixels for 10 nm width structures.	117
Figure 4.19 Measured height versus input model height for a range of noise additives for 3 and 5 degree tilts, respectively.....	118
Figure 4.20 Measured height versus input model height for a range of noise additives using both 3 and 5 degree tilts.	119
Figure 4.21 Height measurement for fins versus added white noise, resulting from fin images of two different tilts being continually drenched with excess white noise. The small images underneath the graph show the fin region with only shot noise added on the left, and then with white noise added in steps of 2σ going to the right.	120
Figure 4.22 Height measurement for fins versus added white noise, resulting from fin images of two different tilts being continually drenched with excess white noise. The plots follow similar trends until the edgewidth reach sizes approximate to the interaction volume of the impinging electron beam.....	121
Figure 4.23 Top – measurement of height versus change in sidewall angle. Calculations were made using edgewidths from tilt angles of 3 & 5 degrees as well as for each individually (for the individual case assuming that the sidewall angle is the input value). Bottom – measured sidewall angle versus designed sidewall angle. Calculations were made using edgewidths from tilt angles of 3 & 5 degrees.	123

LIST OF TABLES

Table 1.1 Summary of transistor architecture and trends and possible future designs (adapted from [10]).....	27
Table 2.1 Advantages and disadvantages to using the two primary approaches being explored for employment in massively parallel electron beam inspection.....	38
Table 2.2 Summarized table of inspection tool performance specifications from the 2014 edition of the ITRS.....	52

LIST OF EQUATIONS

Equation 1 Cross section for scattering of light.....	41
Equation 2 Scattering strength of a particle	41
Equation 3 Lorentz – Lorenz equation relation of the polarizability per unit volume to the experimentally observable index of refraction	42
Equation 4 Scattering strength as a function of a particles size and wavelength	42
Equation 5 Definition of contrast between two signal regions	44
Equation 6 Rose Criterion.....	44
Equation 7 Rose Criterion for detectability with a minimum acceptable signal-to-noise ratio	45
Equation 8 The least number of electrons to contribute to an image pixel given a contrast level	45
Equation 9 Number of secondary electrons emitted n_{se} by irradiation with a probe current I_p in a dwell time of τ with secondary electron emission coefficient of δ and e as the electronic charge.....	47
Equation 10 Minimum beam current required to detect the contrast between two different regions as a function of scan rate	47
Equation 11 Net contrast from a defect size d within a pixel size p	49
Equation 12 Expression for throughput as a function of engineering and operational parameters	49
Equation 13 Throughput equation adjusting for when a defect/feature is smaller than the pixel size	50
Equation 14 Poisson probability mass function representing shot noise.....	77
Equation 15 White noise component described by a normal distribution.	78
Equation 16 Geometric relationship of height to edgewidths and tilt angles for a trapezoidal structure.....	93
Equation 17 Geometric relationship of sidewall angle to height and edgewidth & tilt angle from one image.	93

Equation 18 Equation for determining height of a trapezoidal structure from edgewidths and tilt angles.	94
Equation 19 Equation for determining sidewall angle of a trapezoidal structure from edgewidths and tilt angles.	94
Equation 20 Equation for determining height of a trapezoidal structure from edgewidths and tilt angles.	94
Equation 21 Three sigma precision ($3\sigma_h$) of the height measurement. In the formula, $3\sigma_{EW2}$ and $3\sigma_{EW1}$ are the 3σ edgewidth measurements for the tilt angle images. Note that the covariance term is included only if there is a reliable estimate otherwise it can be removed as in the final form of the equation.	95
Equation 22 Derived solution for SWA measurement from a re-entrant trapezoidal model...	96

LIST OF SYMBOLS & ACRONYMS

α_d	Dipole polarizability
α	Tilt angle
c	Speed of light
C	Contrast
δ	Secondary electron emission coefficient
D	Defect size
ϵ_0	Permittivity of free space
E	Electronic charge
EW	Edgewidth
F	Image frame time
h	Height
θ	Sidewall angle
I_p	Probe current
κ or SNR (S/N)	Signal-to-noise ratio
K	Optical scattering strength
λ	Wavelength
N	Electron counts in an image pixel
n_r	Refractive index
N	Noise
P	Number of pixels in an image
ρ	Density of dipoles
S	Signal from feature/defect
S_0	Signal from background
τ	Time constant for signal integration (dwell time)
T	Throughput
φ	Detector quantum efficiency
ω	Frequency
BSE	Backscattered Electron
CD	Critical Dimension
CD-AFM	Critical Dimension Atomic Force Microscopy
CD-SEM	Critical Dimension Scanning Electron Microscope
EBI	Electron Beam Inspection
eV	Electron volt
FIB	Focused Ion Beam
FinFET	Fin Field Effect Transistor

HVM	High Volume Manufacturing
IC	Integrated Circuit
ITRS	International Technology Roadmap for Semiconductors
keV	Kilo-electron volt
OCD	Optical Critical Dimension Scatterometry
SE	Secondary Electron
SEM	Scanning Electron Microscope
TEM	Transmission Electron Microscope

1 INTRODUCTION

Developments in nanoscale science has led to advancements of new material synthesis and technique production which are greatly affecting fabrication capabilities in numerous manufacturing industries [1]. These progressions are underwritten by the science of theoretical and practical measurement and its application; known as metrology. The significance of which is undoubtedly confirmed in the development of nanotechnology whereby the ability to measure at the nanoscale has categorically been a key factor. The constant and consistent challenge of metrology is the necessary improvement of techniques to allow measurement at the incessantly decreasing end of the size scale, thus facilitating improved technical knowledge of that space. Additionally, there is the challenge of creation of dependable infrastructure to secure the reliability of nanoscale measurement results in research and development as well as for commercial materials and products [2].

With integrated circuits (IC) being incorporated into an ever increasing number of products to replace old technologies and/or develop new ones, economic drivers for less costly, higher performing devices proceeds to grow. Design and manufacture of high performing ICs in a

manner that is financially beneficial and sustainable requires that the processes of manufacturing be meticulously controlled. Material properties and film thicknesses must be accurate and uniform, while structure linewidths and edge profiles must fall within tight limits, and devices must be free of yield-affecting defects. Wafer defect inspection and thin film metrology are crucial in control of the semiconductor manufacturing process. Structure linewidths, film properties, and defect levels must be measured to initially optimize the fabrication process then later to ensure that operation is under control [3].

As high volume manufacturing (HVM) semiconductor technology nodes progress further down the nanoscale regime, conventional process metrology systems continue to experience higher rates of diminishing sensitivity and efficiency to the point where some may become completely obsolete. Advanced wafer patterning techniques involve the usage of extreme ultraviolet (EUV) lithography, directed self-assembly (DSA) of block copolymers, and triple and quadruple patterning. Each of these produce particular challenges for defect detection, measurement of critical dimensions (CD), etc. The process metrology points in the fabrication line are there to ensure quality of manufacture in the production environment. Continued effective use of metrology systems is vital to fabrication yield.

Defect inspection is currently dominated by brightfield optical methods based on scattering of light to detect presence of defects, not to necessarily resolve them [4]. As feature sizes continue further down into the nanometer space, however, changes arise in the physics of light scattering which render the scattering technique no longer viable in effectively discerning defects. As such, light scattering-based defect inspection technologies used in HVM are seeing

exponentially diminishing sensitivity for present semiconductor technology nodes [5].

Transistors made with the use of planar architecture met acute performance limitations because of the undesirable short channel effects imposed by physical scaling down of features. In non-planar devices e.g., FinFETs (Fin Field Effect Transistor) or Tri-gates, multiple sides of the channels are surrounded by the gate, allowing higher drive current [6], improved electrostatic control (lower off-state current leakage), and lower supply voltage requirements than planar devices. In order to remain scaling with Moore's law, three dimensional (3D) device architectures were incorporated into manufacturing at the 22 nm node [7], [8]. Since then, non-planar transistor devices are standard complementary metal oxide semiconductor (CMOS) device architecture. Even still, their use poses challenges for metrology and these challenges will only further increase by the continual reduction of their 3D structure and a possible future evolution to gate-all-around (GAA) transistors (Figure 1.1) [9], [10].

Node (nm)	32/28	20/1x	14	10	7	5	3.5	2.5
Architecture	<i>Planar</i>	<i>Planar /FinFet Si</i>	<i>FinFet: Si/SiGe/Ge Channel</i>		GAA or III-V materials			

Table 1.1 Summary of transistor architecture and trends and possible future designs (adapted from [10]).

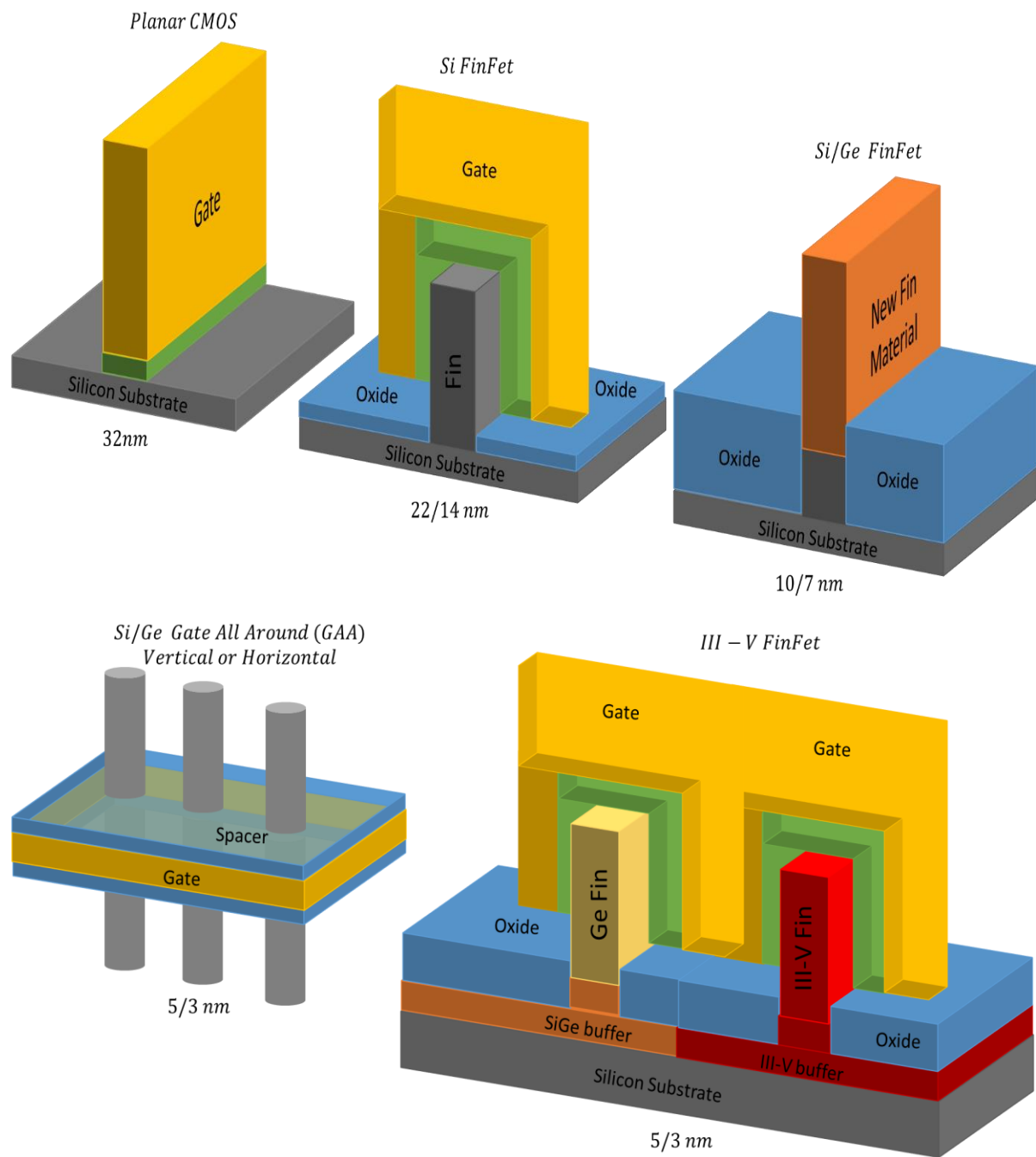


Figure 1.1 Illustrative roadmap for CMOS logic transistors. Application of strained silicon, high-k/metal gate, and non-planar devices continues to enable the pace of advances in performance (adapted from [9], [10]).

The transition from planar to non-planar device architectures brought about a substantial increase in the CD measurements necessary for proper process evaluation making it so that not one of the numerous CD metrology techniques available gives a complete solution to metrology needs. Implementation of mainstay techniques such as critical dimension scanning electron microscopy (CD-SEM) and optical critical dimension (OCD) scatterometry are extensive. These methods provide complementary information i.e., CD-SEM used for CD variation and roughness measurements while OCD employed for profile monitoring and average CD value determination. Also, these technology approaches are starting to more directly support one another through hybrid metrology [11]. Currently, optical critical dimension (OCD) scatterometry is the backbone technology used for the in-line (in fabrication environment) metrology of FinFETs [12]. Nevertheless as production moves from one design node to the next, there are challenges to the further use of OCD metrology tools. These systems rely on complex geometric models for optical CD measurements and require reference models for validation while typically working over tight process windows.

Both situations abovementioned require that the issues of resolution and throughput be realized and addressed in concordance with the demands of manufacturing environments. The use of a multi-electron beam approach to resolving these pertinent issues of sensitivity and output in relation to the abovementioned process metrology steps will be discussed in the forthcoming section.

2 STATE OF THE ART/LITERATURE REVIEW

When image resolution is required to reach into the nanometer range, electron microscopes provide highly capable solutions. Since their invention, the technology platform has developed into a pervasive tool throughout multiple science disciplines and research laboratory types. The prevailing form of this technology are scanning electron microscopes (SEMs), which have developed from large equipment systems (i.e., similar to early form computers) to automated, desktop-size or smaller tools. There have been vast advancements in other aspects along with miniaturizing the tool setup. Such advancements include increasing the imaging resolution limit, improved user experience by digitalization/software employment, the addition of extra detector tools (e.g., energy-dispersive x-ray spectrometry (EDX)) to take full advantage of the amount of information that can be extracted from a sample, etc. [13].

However, the promptness with which sample images are attained when keeping the resolution in nanometers and the signal-to-noise ratio (SNR) high has yet to be enhanced profoundly. Consequently, SEM images that are high in resolution from small areas are the only ones that can be acquired in practical amounts of time. Thus far, this has been a relatively benign

problem to deal with as need for high throughput, high resolution electron microscopy has been contained to a comparatively small number of application spaces. Though, with the changing conditions of wafer defect inspection, critical dimension measurement, etc. this limitation need be addressed with a sense of urgency.

2.1 MULTIPLE ELECTRON BEAM SYSTEMS

One approach to overcoming this throughput deficit is to employ multiple electron beams within a single platform. Introduced in the late 1960's, the multi-electron beam system concept is meant to increase the output of electron beam systems while avoiding source brightness limitations and Coulomb interactions [14], [15]. Since then, research and development of multi-beam systems continues with various approaches taken by many different teams internationally. Categorically, these systems can be thought of as Multi-Axis, Single Column by Multi-Source, Single Column by Single Source, and Multi-Column by Single Source (Figure 2.1) [16].

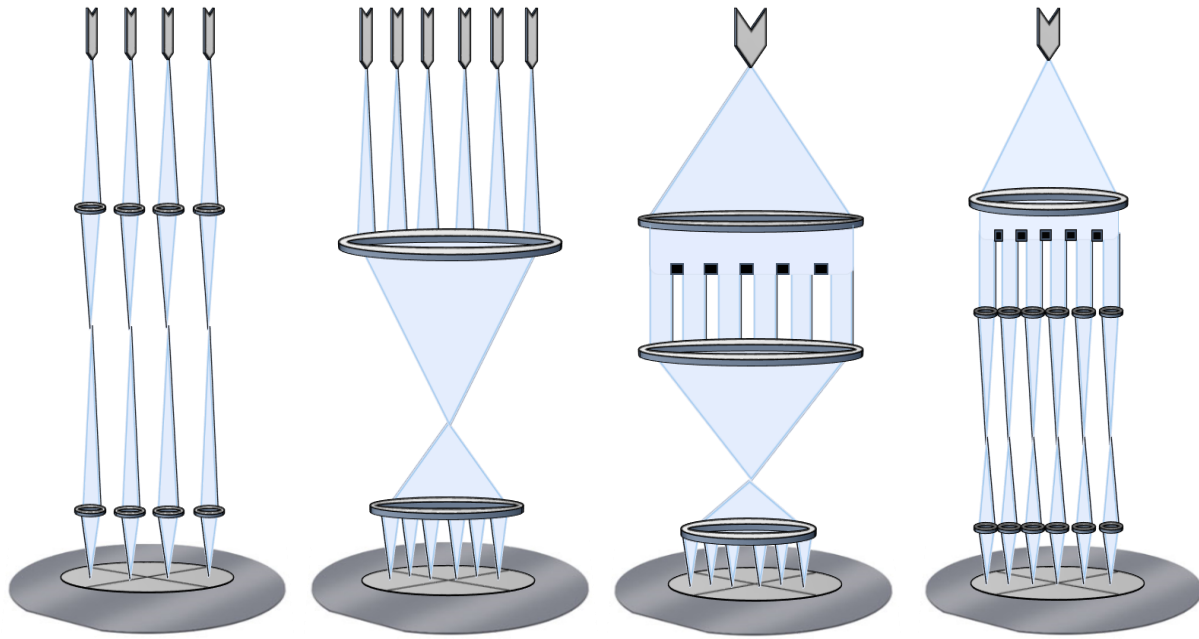


Figure 2.1 (Left to Right) Schematics (non-scale) of the four general categories of multi-beam systems: Multi-axis systems; Single column, multi-source; Single column, single source; Multi-column, single source.

2.1.1 MULTI-AXIS

The Multi-Axis system consists of multiple miniaturized single beam columns arrayed to emit multiple beams [17]–[28]. In contemporary designs, each miniature electron column comprises of a low-extraction-voltage thermal field emitter (TFE), optical components micromachined using IC and Micro Electro-Mechanical System (MEMS) fabrication technology, and a Microchannel plate (MCP) detector. Schematically, the optical apparatus of each column is comprised of four lens components which are the source (including the extractor, condenser, and blanker), the aperture, the dual octupole deflectors, and the objective (Einzel) lens. The optical column length is on the order of ten millimeters while typical working distance is a few millimeters. The gun assembly as a whole is on the order 40 mm [29], [30]. Further particulars on the fabrication and alignment procedures along with the electron optics design and

optimization of such a system have been expressed elsewhere [31], [32].

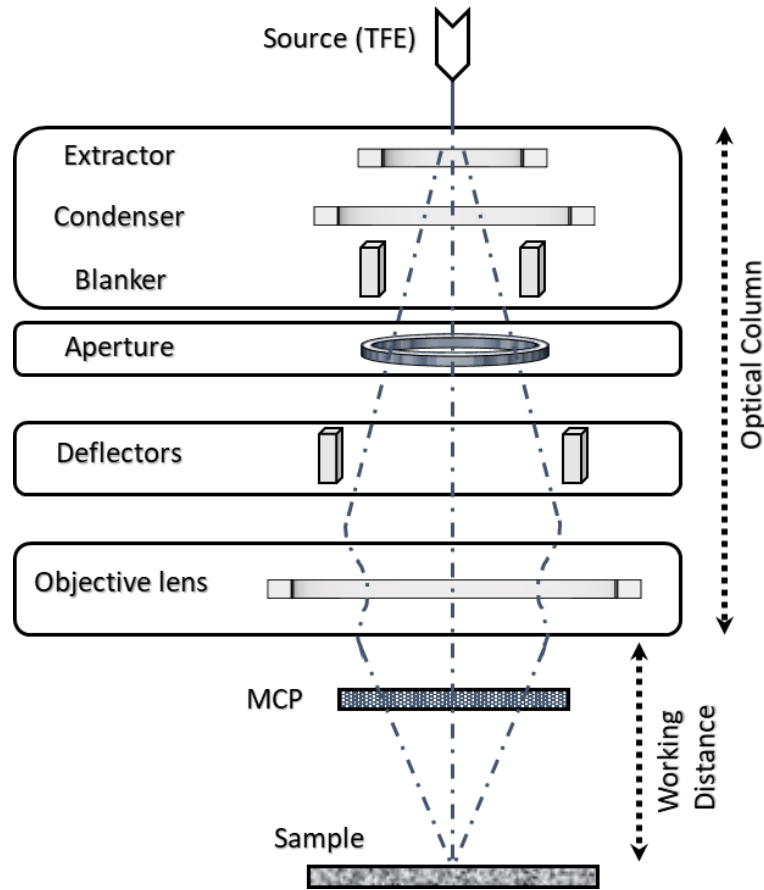


Figure 2.2 Schematic presenting a cross section of a gun assembly for a miniature column electron beam system. The gun assembly all together is on the order 40 mm with optical column length on the order of ten millimeters while typical working distance is a few millimeters (adapted from [29], [30]).

Advantages of an arrayed column system include the use of high brightness electron sources and the lack of Coulomb interactions or ‘crosstalk’ between adjacent beams as a result of the condensed electron interaction length. Although, it is a non-trivial task to either scale up or down the number of columns as required by the technology node in question because of the difficulty of making sure that each miniature column emits identical current using the same probe size at a particular location [16]. Though, there have been substantial strides made with recent advancements in design and development of miniature column systems to make them

much more viable contenders in the HVM inspection space moving forward [29], [33].

2.1.2 SINGLE COLUMN BY MULTI-SOURCE

The Single Column by Multi-Source system consists of beams emitted from a multi-source array, subsequently conveyed to a single column. As it is hard to guarantee the quality of each miniature column and that the throughput is limited by the number of those columns in a Multi-Axis system, a probable resolution would be to employ a shared projection lens for all the incoming beams. In Single Column by Multi-Source systems, multiple electron beams have been generated by use of several sources e.g., multiple laser-driven photocathodes or microfabricated emitter arrays [34]–[48]. In this system, because of the crossover point where all of the beams go through, electron-electron interactions may cause blur and beam displacement along with reduced beam pitch compared to that of the Multi-Axis system depending on the pitch of the arrayed source and system demagnification i.e., repulsion between beams may not be trivial. Coulomb interactions in this system may be reduced by modifying the source pitch, length of the column, and number of beams. Diligent design of the optical column may improve the Coulomb interactions along with other issues such as lens aberrations, system demagnification, etc. So, the main challenge for these systems is not the Coulomb interactions and off-axial aberrations which may be adjusted for, but fabrication of source arrays with the required high brightness, stability, low energy spread, and long lifetime [16].

2.1.3 SINGLE COLUMN BY SINGLE SOURCE

In order to bypass the large array of electron sources needed for the previous category of multi-beam systems the Single Column by Single Source approach was produced. In it, multiple beams are generated from a single emitting source and are subsequently conveyed to a single column [49]–[60]. Schematically these systems are usually comprised of an electron source, collimator lens, and an aperture array along with an individually focused blanker array. Electron beam sources have often been thermionic e.g., LaB₆ because of the adequate current with a high density they provide. The source emits a broad beam that is collimated then split by the aperture array. Subsequently, the blanker array is setup below which can switch on or off each beam. After which, projection lenses that provide a strong demagnification ratio are used to project the post aperture beams. In this such system, large Coulomb interactions may arise before the broad beam arrives at the aperture array, but unlike Coulomb interactions that occur at the crossover at projection lens these interactions do not reduce the resolution. They, on the other hand, cause lowering of source brightness and higher energy spread. So as to reduce these effects, a Schottky emitter electron beam source has been employed with modified optical column design e.g., a negative lens towards the bottom the column to lessen positive aberrations produced by the collimator lens [56]. Implementing a Schottky source is useful for high resolution use cases because of its high brightness as well as the current from a Schottky emitter result in Coulomb interactions that are less severe than that in other multi-electron beam sources [16]. The Carl Zeiss GmbH MultiSEM system employed in this work uses a Schottky emitter electron source.

2.1.4 MULTI-COLUMN BY SINGLE SOURCE

The Multi-Column by Single Source was developed to keep the advantageous use of a single electron beam source and to avoid the Coulomb interactions that persist in the single column design. In it, beams emitted from a single source are subsequently conveyed to multiple columns [61]–[75]. The MAPPER system that was first developed out of Delft University in the Netherlands is an example of such a technology [64], [68]. Primarily meant for maskless lithography, the system can also be expounded to inspection [65], [75]. In such a hybrid system, the upper column resembles more of multi-electron beam single source while the lower column resembles a Multi-Axis system. Unlike joint projection, each of the beams pass on to an individually focused projection lens without suffering from off-axial aberrations and since there is no crossover after beam splitting the Coulomb interactions are less of an issue [16]. Though, these systems are not without issues and still require numerous design considerations to be improved e.g., number of beams, pitch between the beamlets, etc. Roughly, a look at the electron optical column shows that the lens array needs to be designed in a way where each lens forms a stigmatic and well-focused image of the source free from the need to individually cater for focus or stigmatism. Next, demagnify the array onto the sample without introducing non-negligible off-axis aberrations from the imaging lenses. Also, signal from each individual beamlet must get an individual detector. Over the years there has been proposed improvements to design and considerable development to the whole of the Multi-Column by Single Source technology addressing most of these issues [65]. A series of commercial efforts have been demonstrated for the production of arrays of electron beams implementing the abovementioned categories of multi-beam systems [76].

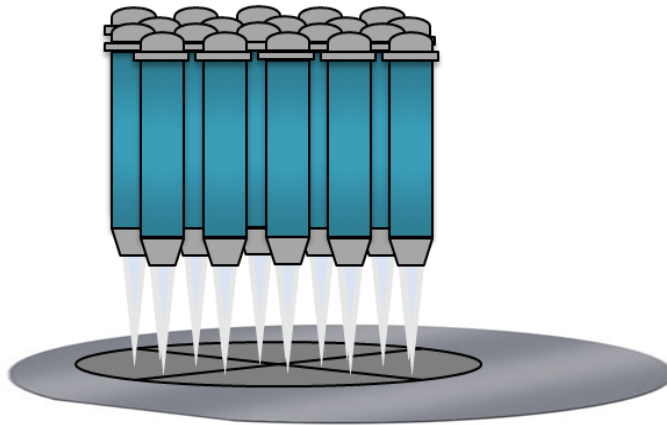
2.2 MULTI-ELECTRON BEAM SYSTEM DEVELOPMENT

As such, to overcome the throughput insufficiency within conventional electron beam systems, efforts have been taken to accelerate the development and commercialization of massively parallel multi-electron beam-based inspection technologies [5], [77]. The viability of designing and building a high-performance, high-throughput scanning electron microscope imaging system has already been demonstrated commercially, but geared more towards biological applications [78]–[80]. This technology could form the core of a revolutionary inspection technology, but it would need to be adapted and optimized for the specific applications of wafer or mask inspection. A major component of appropriating such a technology is to develop the specifications and design requirements for an HVM tool. Details of the tool development program and its objectives to facilitate this process are presented elsewhere as is a discussion of the relevant hardware [5], [77], [81]–[88]. In Table 2.1 below, the advantages and disadvantages of the two primary approaches to massively parallel electron beam inspection are laid out.*

* At the time of the study these two were the most developed opto-electronic systems with beam-splitting being even more developed than miniature columns hence the use of the Zeiss MultiSEM system technology in validation imaging experimentation [33].

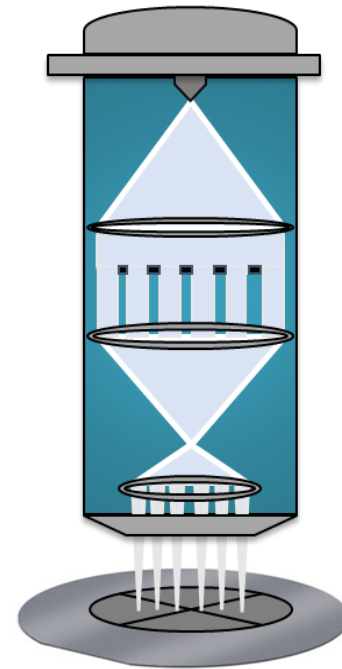
Miniature Columns

Multiple emitters/Multiple optics



Beam Splitter

Single emitter/ Single optic



Pros	<ul style="list-style-type: none"> • Direct control of each beam • Customizable positions by application • High current per beam • Throughput scaling with wafer size 	<ul style="list-style-type: none"> • Single optical path – more uniform performance • Single emitter • Rapid throughput in area being scanned
Cons	<ul style="list-style-type: none"> • Controlling uniformity and performance over individual beams • Requires individual emitters • Technology readiness – (column structure requires optimized redesign to meet HVM specifications [29]) 	<ul style="list-style-type: none"> • Number of beams limited by column design • Lower current per beam

Table 2.1 Advantages and disadvantages to using the two primary approaches being explored for employment in massively parallel electron beam inspection.

2.3 DEFECT INSPECTION

High-throughput inspection for critical defects on dense patterned wafers is a key metrology capability gap at upcoming nodes [89]–[91]. Brightfield inspection for critical defects, i.e., those larger than the half-pitch for the node, is already exhibiting compromised sensitivity for defects smaller than 20 nm. Conversely, electron beam inspection (EBI) tools are sensitive to defects which are 5 nm or smaller, but the throughputs of single-electron beam tools are several orders of magnitude too low to meet HVM requirements for defect inspection.

2.3.1 LIMITATIONS OF OPTICAL INSPECTION OF WAFER DEFECTS

Although the specific size that designates a defect as critical is process and application specific, for our purposes the generic definition that any defect greater than one-half of the process node is critical is used. For some applications, the critical size maybe as small as one-tenth of the node size. Consequently, for the 14 nm node process, locating defects as small as 7 nm is vital. Here, defects comprise nanometer-sized extraneous particles, defects printed during pattern lithography, surface scratches, and process residues. The trend for critical size defect specifications as identified in the International Technology Roadmap for Semiconductors (ITRS) is shown in Figure 2.3 [5], [92].

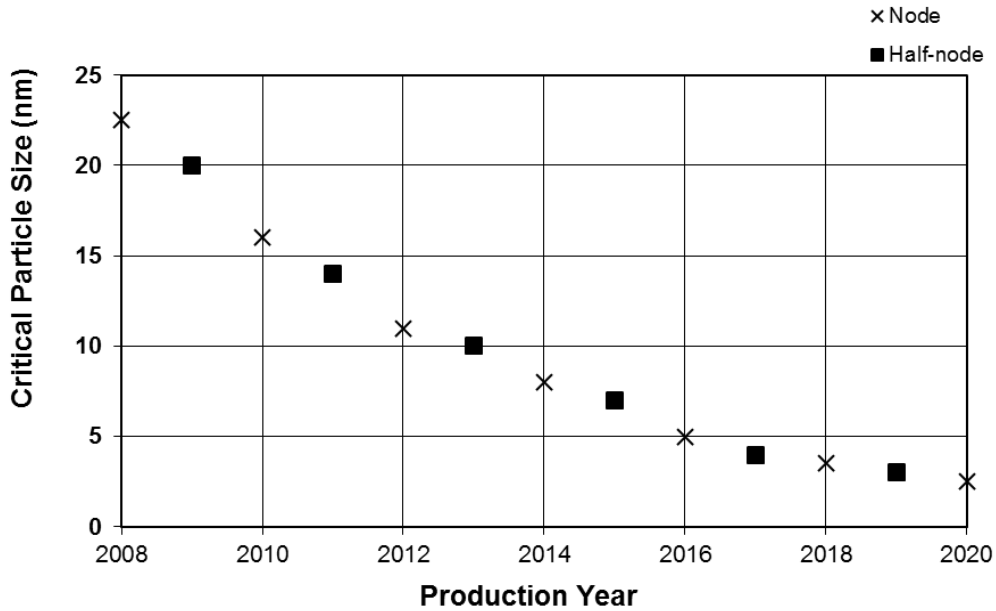


Figure 2.3 Critical size for particles on bare wafers for recent past and future production years. Data includes critical defect sizes for the ITRS technology nodes and related stopgap half-nodes (process technology where chip size is reduced without need for restructuring the circuit architecture to be suitable for a reduced area.)

The optical scattering methods that currently dominate defect inspection typically use wavelengths in the deep ultraviolet range [4]. As mentioned, with decreasing feature sizes further down the nanometer range, the physics of light scattering begins to change. The significance of the previously noted 20 nm defect size is that it represents a particle size that is less than a tenth of the commonly used UV wavelength (λ) of 266 nm. Objects larger than about one-tenth of the wavelength have light scattering that is well-described by the Mie solution to Maxwell's Equations, which has the property that strength of scattering scales with the square of the particle diameter i.e., with its area, and shows a very weak dependence on wavelength. Objects with diameter below one-tenth, however, are more accurately described by Rayleigh scattering. Rayleigh scattering treats the scattering object as a collection of a

collection of dipoles each with polarizability α_d . The cross section σ for scattering of light with frequency ω by a dipole is given by the Rayleigh equation (Equation 1).

$$\sigma = \frac{\omega^4 \alpha_d^2}{6\pi \epsilon_0 c^4}$$

Equation 1 Cross section for scattering of light

The speed of light is represented here by c and ϵ_0 is the permittivity of free space. Accordingly, the scattering strength K of a particle of diameter d containing a density of ρ dipoles per unit volume is

$$K = \frac{\omega^4}{6\pi \epsilon_0} \left(\frac{1}{6} \pi d^3 \rho \alpha_d \right)^2.$$

Equation 2 Scattering strength of a particle

For an isotropic dielectric substance[†], the Lorentz – Lorenz equation relates the polarizability per unit volume to the experimentally observable index of refraction n_r , by

[†] The polarizability and index of refraction for an anisotropic substance take the form of tensor properties, so the polarization of light needs to be accounted for. Nonetheless, as the values alongside the principle axes vary slightly by a few percentage points, the existing argument is unaltered.

$$\frac{\rho\alpha}{3\varepsilon_0} = \frac{n_r^2 - 1}{n_r^2 + 2}.$$

Equation 3 Lorentz – Lorenz equation relation of the polarizability per unit volume to the experimentally observable index of refraction

Bringing in Equation 3 into Equation 2 and then converting angular frequency into wavelength produces a formula that describes the scattering strength of particles as a function of size and wavelength:

$$K = \frac{2\pi^5 \varepsilon_0 d^6}{3 \lambda^4} \left[\frac{n_r^2 - 1}{n_r^2 + 2} \right]^2.$$

Equation 4 Scattering strength as a function of a particles size and wavelength

The above equation (Equation 4) provides an apt theoretical description for the analysis of brightfield inspection.

With devices becoming tinier, forecasts for prolonging brightfield technology are slight, mainly because Rayleigh dipole scattering decreases as the sixth power of the particle diameter. Optical inspection is executed with wavelengths ranging from 266 – 193 nm, thus at approximately a 20 nm feature size, Rayleigh scattering begins to dominate. Consequently, the sensitivity of defect inspection technologies based on light scattering used in the semiconductor manufacturing environment is decreasing exponentially for current technology nodes, as demonstrated by deteriorating defect capture rates. Whereas one ostensible solution would be to move to shorter wavelengths, scattering strength just increases with the fourth power of the inverse wavelength, thus the exponential loss in sensitivity will continue.

Moreover, optics, detection, and radiation damage all become more complex if more intense ultraviolet wavelengths are employed. Figure 2.4 shows the sensitivity of brightfield scattering for the critical size defects identified by the roadmap, scaled relative to a 266 nm light scattering from an 8 nm particle. It can be seen that the sensitivity of brightfield scattering to critical sized defects will decrease by an order of magnitude approximately every third year.

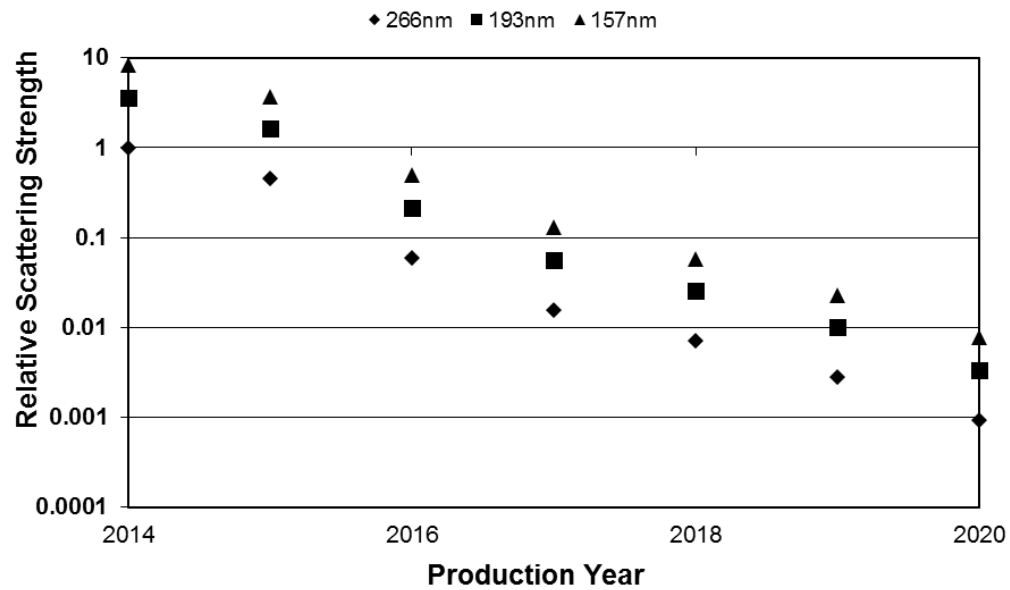


Figure 2.4 Relative scattering strengths of critical sized defects/particles by production year scaled relative to a 266 nm light scattering from an 8 nm particle ($K = 1$) for various wavelengths of optical inspection.

Considering what may come ahead, an added complication arises as critical particle sizes are headed for quantum dimensions. There the refractive index of a nanoparticle starts to deviate from that of the bulk substance, adding another level of uncertainty to indirect particle size measurements. To sum up, optical inspection methods presently in usage are suffering an exponential deterioration in defect sensitivity, and the semiconductor manufacturing industry is probing for substitute inspection technologies.

2.3.2 INSPECTION USING ELECTRON BEAMS

Contrast C amid two regions, e.g., the signal from a region of interest S_i and from the background signal S_0 , is what determines the ability to differentiate between them. In this regard contrast is defined via:

$$C \equiv \frac{S_i - S_0}{S_0} = \frac{\Delta S}{S_0}.$$

Equation 5 Definition of contrast between two signal regions

By the Rose Criterion in order for a change in signal ΔS to be detectable with ~100% confidence, it's necessary that the change in signal should go above the noise level N by a factor of 5.

$$\Delta S > 5N$$

Equation 6 Rose Criterion

Since secondary electron emission is a normally distributed stochastic process, the root-mean-square (rms) noise is the standard deviation of the signal. Hence, if the number tally in a signal is n , the rms noise is square root of n , and the signal-to-noise ratio (SNR or S/N) is $n^{1/2}$. The concepts of signal, contrast, and rms noise are illustrated graphically in Figure 2.5.

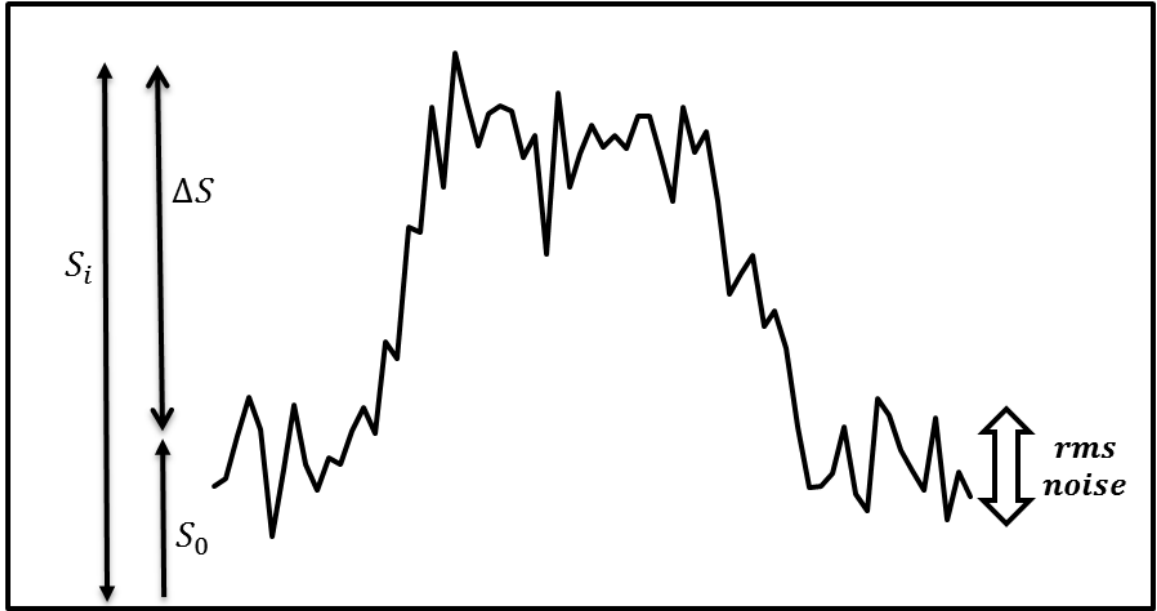


Figure 2.5 Illustration of signal, contrast, and noise from a line scan over a feature S_i on a background S_0 .

Linking the Rose Criterion with the definition of contrast, the Rose Criterion for detectability with a minimum acceptable signal-to-noise ratio can be stated as

$$C = \frac{\Delta S}{S_0} > \frac{5N}{S_0} = \frac{5\sqrt{n}}{n} = 5n^{-\frac{1}{2}} .$$

Equation 7 Rose Criterion for detectability with a minimum acceptable signal-to-noise ratio

Accordingly, given a contrast level, the least number of electrons contributing to an image pixel in order for the feature to be detected with >99% certainty has to be

$$n_{min} > \left(\frac{5}{C}\right)^2 .$$

Equation 8 The least number of electrons to contribute to an image pixel given a contrast level

Alternatively, it can be said that the Rose Criterion is equal to a 5σ measurement. A 20% contrast change, for example, requires 625 electrons to be observed at this level of certainty. The discrimination of features is based on signal threshold, so where circumstances produce poor contrast and large amounts of noise, a specific foray of large noise from the background level can be recorded as a false positive signal. Equally, a specific low signal from the object of interest could cause that feature to be confused with the background, causing a false negative. The following figure shows influence of contrast and noise on measurement certainty for defect detection, which for demonstrative purposes can be thought of as the statistical distribution of signals resulting from multiple measurements of a single region (e.g., background and feature from Figure 2.5) or the distribution of measurements from a large number of identical features.

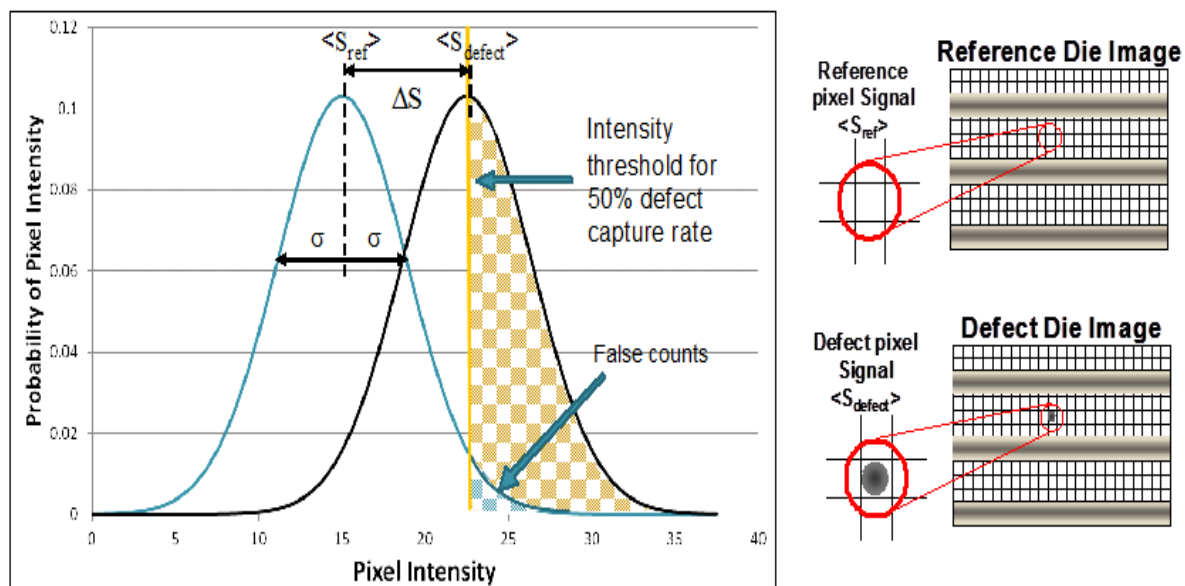


Figure 2.6 Schematic illustrating the shot noise limited secondary electron emission probability of reference and defect containing pixels [93].

Being that the Rose Criterion is resultant from statistical arguments purely, there is no need to be so unbending with its use. A value of $\Delta S/N = 2$ still provides a confidence level of >97% that the measured signal is real and true and not simply spurious noise. Moving ahead, κ will be used to signify the desired signal-to-noise ratio. For a material with a secondary electron emission coefficient of δ , the number of secondary electrons emitted n_{se} by irradiation with a probe current I_p in a dwell time of τ , where e is the electronic charge, is

$$n_{se} = \frac{\delta I_p \tau}{e}.$$

Equation 9 Number of secondary electrons emitted n_{se} by irradiation with a probe current I_p in a dwell time of τ with secondary electron emission coefficient of δ and e as the electronic charge.

Integrating the detector quantum efficiency (DQE) ϕ (equal to the square of an experimentally derived SNR divided by the theoretical SNR from a sample imaged) and setting Equation 8 and Equation 9 the same, solving for an expression giving the minimum beam current I_{min} required to detect the contrast between two different regions as a function of scan rate is direct.

$$I_{min} = \frac{\kappa^2 e}{C^2 \delta \phi \tau}$$

Equation 10 Minimum beam current required to detect the contrast between two different regions as a function of scan rate

In Figure 2.7, the relationship between SNR and n_{se} is shown. The equation for static SNR in electron beam settings is given as $SNR = \sqrt{6\phi n_{se}}$ [94]. The parallel lines represent the different levels of detector efficiencies with the range found among actual tools in use [95].

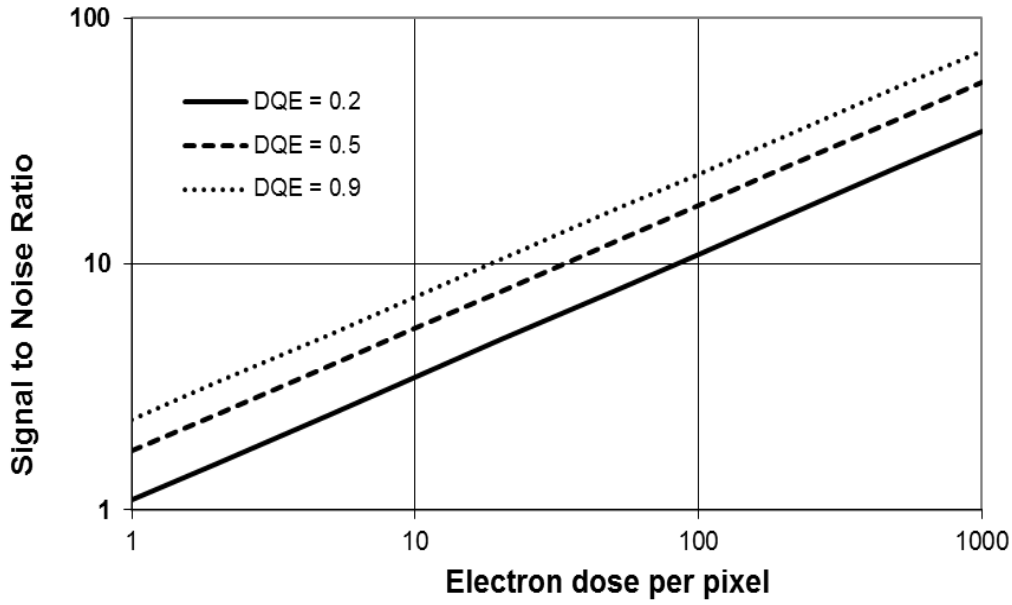


Figure 2.7 Signal to noise ratio dependence on the electron dosage per pixel at given detector quantum efficiencies.

The dwell time can also be thought of as the ratio of the frame-time F (inverse of scan rate) to the number of pixels in the image P . For a reference point, a characteristic set of operating parameters for a high quality CD-SEM image would be $I_p = 8$ pAmp, $P = 512 \times 512$ pixels, $\delta = 1$, $\phi = 0.33$, and $F = 2$ seconds. These conditions equate to a dwell time of $7.6 \mu\text{sec/pixel}$, and yield a minimum detectable contrast of 50% [92].

It has been implicitly taken under consideration that the feature size in the analysis above is much larger than the actual pixel size (as in Figure 2.5). This is what happens when attempting to image down to the information limit. When inspection is taking place a defect might actually be smaller in size than the pixel/probe diameter. Here, the detection standard should be based

on whether the contribution from the defect to the overall pixel intensity is sufficient enough to offer sufficient contrast between a pixel containing a defect and one free from defect. The way contrast is designated in Equation 7 is principally associated to the difference in intensity between the defect and non-defect features. When a pixel is larger than the defect, the variable of feature contrast needs to be scaled by the amount of area the defect takes in relation to the full area of the pixel. As such, a defect of size d within a pixel size p , leads to the net contrast C^* .

$$C^* = \frac{d^2}{p^2}$$

Equation 11 Net contrast from a defect size d within a pixel size p

The preceding derivation defined the performance requirements set out for a single electron beam to reach a given level of sensitivity. Subsequently, these must be scaled by throughput specifications for defect inspection to conclude the number of electron beams required.

2.3.2.1 THROUGHPUT CALCULATION

With the use of the minimum beam current formula (Equation 10) an expression for throughput T (in units of cm^2/hr) can be developed as a function of engineering and operational parameters, which include probe current in nAmp and pixel size p in nm, of a multi-electron beam tool having N_b beams.

$$p^2 * \tau^{-1} = \left[\frac{\kappa^2 e}{C^2 \delta \phi I_p} \right]^{-1} * p^2 \rightarrow T = 0.225 * N_b \frac{I_p C^2 \delta \phi}{\kappa^2} p^2 \left[\frac{\text{cm}^2}{\text{hr}} \right]$$

Equation 12 Expression for throughput as a function of engineering and operational parameters

Encompassed in the numerical pre-factor are the constants and conversion elements necessary to return units of square centimeters per hour. The probe diameter should preferably be the same as the pixel size so as to avoid undersampling. Equation 12 can appropriately be used for situations wherein the defects of interest are bigger than a single pixel. In practice, the pixel sizes used in electron beam inspection are a bit larger than the probe diameter. However, a defect feature could also be smaller than the pixel size/probe diameter, so the detection standard ought to be based on whether a defect's contribution to its pixel intensity is enough to provide sufficient contrast between a reference pixel and one with a defect. That is to say, where defects for inspection are smaller than the pixel size, Equation 11 should be implemented, adjusting the throughput equation for this case.

$$T^* = T * C^{*2} = 0.225 * N_b \frac{I_p}{p^2} d^4 \frac{I_p C^2 \delta \phi}{\kappa^2} \left[\frac{cm^2}{hr} \right]$$

Equation 13 Throughput equation adjusting for when a defect/feature is smaller than the pixel size

The above (Equation 12 & Equation 13) can be integrated by using probe current density as a parameter, but here it is appropriate to show throughput in terms of the operator settings. The two equations are also the same when the feature and pixel size are equal.

In order for any inspection system to be deemed worthy, it should follow along trends set out by the ITRS [90]. Figure 2.8 briefly explains those stipulations along with the target areas of application for inspection systems.

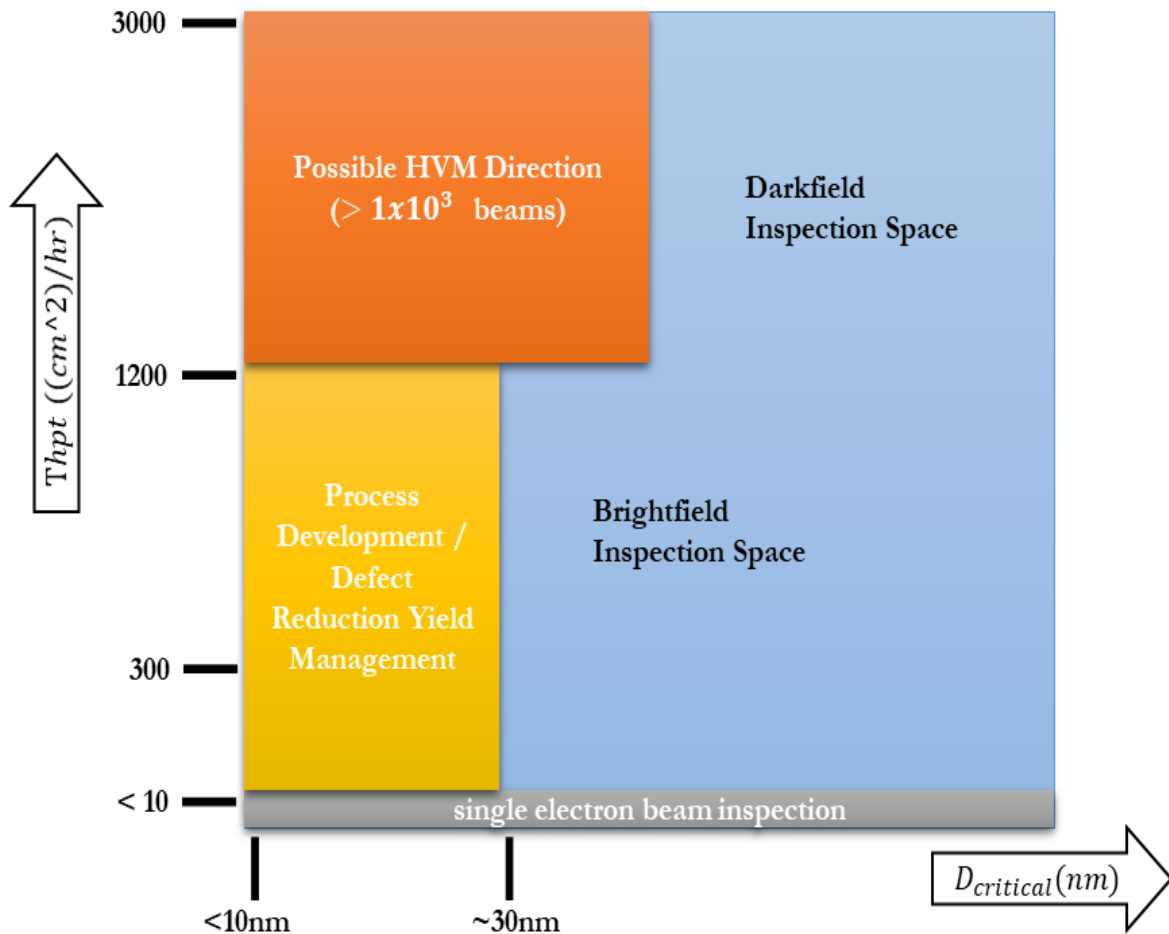


Figure 2.8 Chart showing along the y-axis the ITRS 2014 based requirements for inspection throughput (cm²/hr) in various settings: Process R&D at 300, Yield Ramp-up at 1,200, and Volume Production at 3,000. The x-axis communicates the inspection system type application space with regards to the size of critical defects in nanometers.

Summarized in Table 2.2 are the demarcated throughput rates as well as the efficiency rates that such a system would be minimally capable of.

Performance Metric	Production Stage	Target Value
Throughput	<i>R&D</i>	$300 \text{ cm}^2/\text{hr}$
Throughput	<i>Ramp-up</i>	$1,200 \text{ cm}^2/\text{hr}$
Throughput	<i>HVM</i>	$3,000 \text{ cm}^2/\text{hr}$
Capture Rate	<i>R&D</i>	50%
Nuisance Rate	<i>All</i>	< 5%
False Count	<i>R&D</i>	< 5%
False Count	<i>Ramp-up and HVM</i>	< 1%

Table 2.2 Summarized table of inspection tool performance specifications from the 2014 edition of the ITRS.

Using aggressive but realistic values for the parameters in Equation 13, the performance tradeoff in throughput versus the number of beams for detecting a range of defect sizes has been calculated, as shown in Figure 2.9.

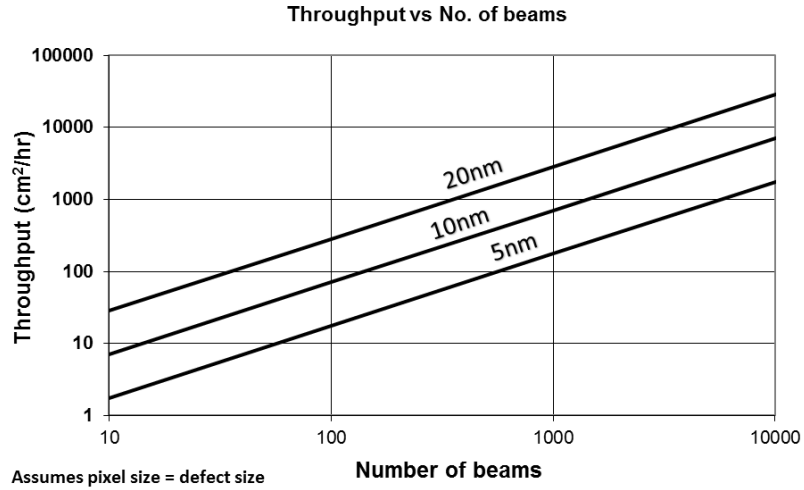


Figure 2.9 Using Eqn. 13 with following assumptions to show the general trends of throughput versus the number of beams: $C = 50\%$, $I_p = 1 \text{ nAmp}$, $\delta = 1$, $\phi = 0.5$, $\text{SNR} = 2$ and pixel size = defect size as denoted on trend lines in the graph.

In Figure 2.10, the general trends of throughput versus dwell time given a particular pixel size and number of beams are shown.

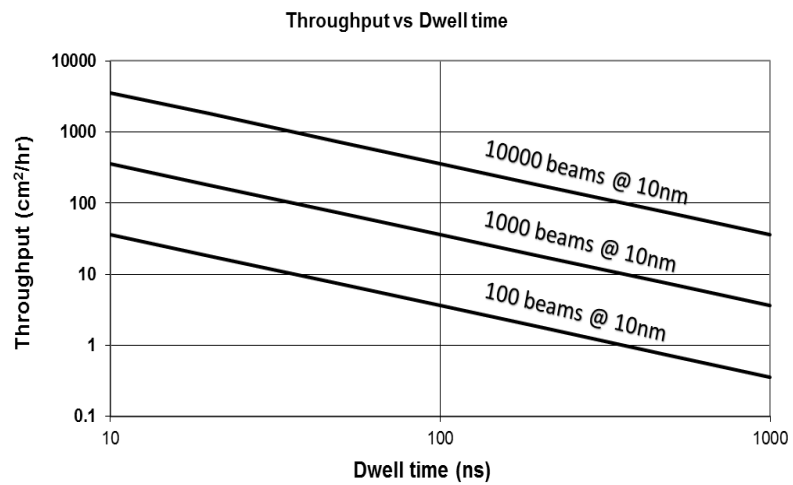


Figure 2.10 General trends showing the interdependence of throughput and dwell time (in nanoseconds) given a certain number of beams and pixel size

In order for a multi-electron beam approach to achieve viable throughput number, the density of beams needs to be rather high whether that be through a beam splitting or multi-column method.

2.4 CRITICAL DIMENSION MEASUREMENT

Accurate and timely measurement of minimum feature sizes is pertinent to maintaining productive wafer production lines. Mass production manufacturing necessitates process control technologies that more than simply meet the ITRS CD metrology uncertainty requirements [90]. They must also be automatic, high throughput, in-line and non-destructive i.e., that the sample is suitable for continued processing after examination and not require scrapping. Statistical significance of measurement sampling is also an important consideration, whereas some techniques measure individual/few features others report on averages over a larger number of features within the area of measurement. To this end, metrologists have employed a variety of tools to take these dimensional measurements i.e., height, width, spacing, etc. Moving from planar to 3D devices greatly complicated this work by going from six different CD measurements in planar device architectures to 12 or more different measurements as required by FinFETs [7], [8].

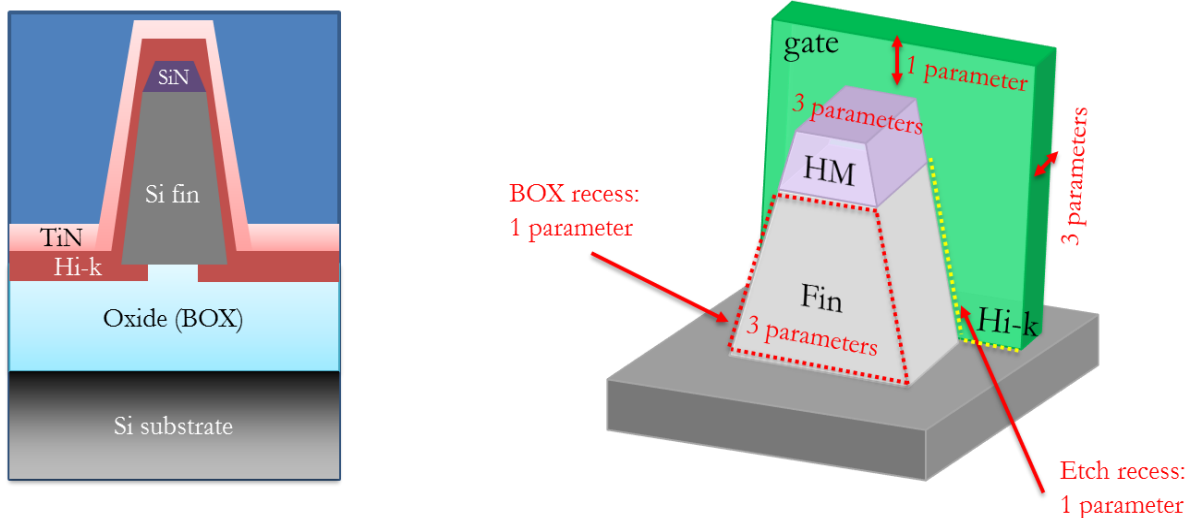


Figure 2.11 Left – Cross-section illustration perpendicular to a fin. Right – Illustration of fundamental unit of a FinFET with indications of twelve important process control parameters.

2.4.1 CURRENT/EXISTING HVM CD METROLOGY SYSTEMS

Under current conventions, in-line mainstay CD metrology setups are usually comprised of two imaging techniques along with one spectroscopic (light scattering and model-based) technique. The imaging techniques are those of CD-SEM and CD-AFM (Atomic Force Microscopy) while the scattering technique is OCD in one or more of its many forms e.g., spectral ellipsometry (SE), Mueller Matrix (MM), normal polarized reflectometry (NI-RP), normal incident plus oblique polarized reflectometry (NI+Obi RP), etc. Since none of these systems can fully address all issues of measurement, determination of which system is best is resolved on a use case basis. Data garnered from imaging techniques is complementary to that of spectroscopic techniques and vice versa. As a result, for HVM CD metrology to be efficacious it is vital for there to be at least one qualified imaging technique along with one spectroscopic technique available for usage for a given critical application step [7], [8].

Techniques based on imaging provide results for the measurement of isolated sample features located wherever on the target subject while affording the ability to determine roughness and variation among subsequent features. Alternatively, spectroscopic techniques deliver results for the measurement of feature profiles for 3D metrology use cases by producing average values with exceptional statistical confidence because of the concurrent sampling of larger target areas. Using a large set of differing techniques to assess and confirm the varying aspects of sample features and then combining their results for valuable hybrid results is an approach known as hybrid or holistic metrology [11]. Such an approach has already been used for provision of co-optimized hybrid results for the parameters incorporated in characterization of 3D features of advanced transistor architectures [96]–[98].

2.4.1.1 Constraints of Current CD Metrology Systems

The drive for continual forward progress of IC performance brings about critical challenges in the shrinking of device dimensions and increasing complexity of device geometry. Looking at the 1× nm node FinFET device process control, traditional metrology setups are confronting challenges in delivering measurements such as that of the gate & fin height, sidewall angle, profile, etc. Presently, in-line process control for 3D features of non-planar devices is mostly accomplished by use of OCD and CD-AFM. Nonetheless, OCD computer models take from weeks to months to optimize and are very much sensitive to frequent process changes at the development stage and are unable monitor in-die process variations. As well, CD-AFM is constrained by its long measurement time and the tip wear that it experiences during the measurement process making it so that chip sampling is extremely limited while in production fabrication [99].

As mentioned, OCD is being used for a greater part of CD measurements for the 3D devices. Though, CD-SEMs are still better suited for top-down imaging of structures. In order to make the CD-SEM more versatile and deliver in-die measurement for the non-planar (FinFET) device process control, tool providers have augmented systems to add a tilting capability to their instrument (Figure 2.12). First introduced in the year 2000, in this technique the electron beam is magnetically bent within the electron optical column so as to strike the sample target at a slight angle deviated from direct normal [100]. This angle tilt permits for imaging of both the top and bottom edges of many features so long as the aspect ratio of the space is such that it allows for the bottom of the raised feature to be visible. For example, a beam tilt angle of

10° implies that the space between the imaged feature and the adjacent feature must be no less than ~20% of the height of the latter (Figure 2.13). Implementing a tilted beam CD-SEM system for measurement does come with other benefits in contrast with OCD. There is already widespread use of CD-SEM for measurements in photolithography at various points of wafer processing. Also, electron microscopy systems have quicker and simpler primary setup of imaging recipes without need for substantial model-building and validation experiments, are indifferent to variations in film stacking as well as most material properties, and are able to swiftly gather data from non-periodic collections of features such as those in an IC [101]. In summary, the added capability allows for more than just top-down imaging, but also the ability to image device heights and take a sidewall angle measurement [99]–[104]. Even with the extra ability, CD-SEMs still cannot measure all remaining parameters [105].

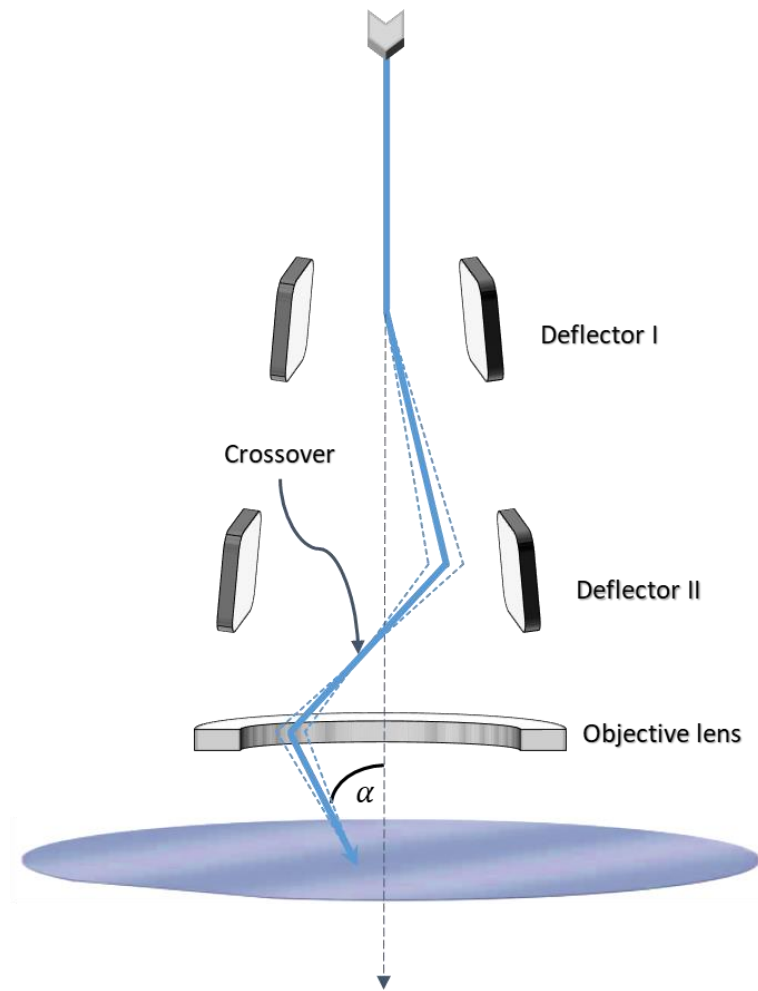


Figure 2.12 Schematic illustration of the principle of tilt beam imaging. After the electron beam has entered the electron optical column, the first deflector repels the beam away from the optical axis. The second deflector pushes the beam back towards the optical axis. Finally, the objective lens focuses the beam on a point where the beam will impinge on the surface of a target sample. The electron beam arrives at a tilted angle when landing on the surface.

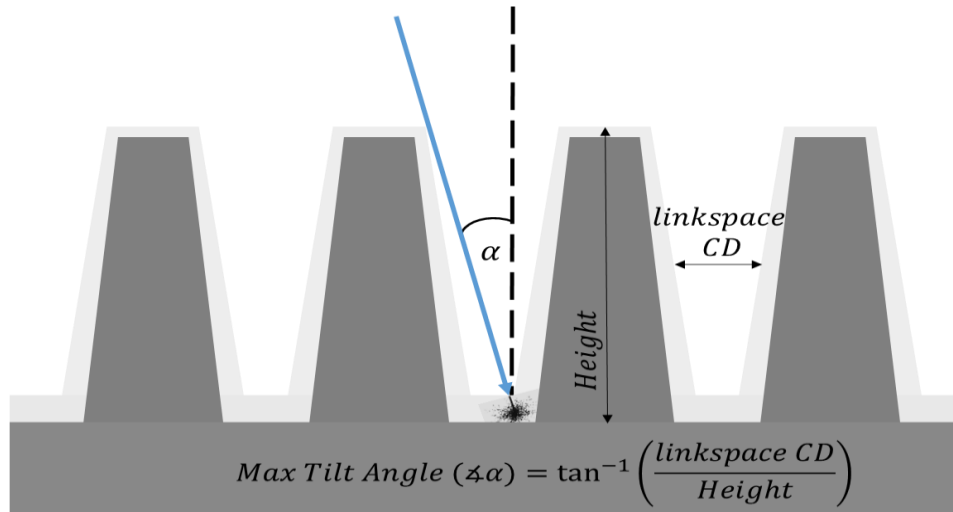


Figure 2.13 Illustration of (and equation for determining) limitation of maximal imaging tilt angle given the feature aspect ratio of a sample.

The innovative use of a multi-electron approach here may be able to expand upon the advancements made by use of tilted beam CD-SEM to capture the most important of necessary CD parameters needed in metrology and possibly advance upon examining of the remaining because of the extent to which it will be able to image structures directly at several different angles at once (depending on the architecture of system) allowing for greater degree and range of measurement while keeping up with the speed of other metrology techniques.

In order to bring such a system to fruition there are some key implications that need to be resolved. Firstly, tilting of the wafer sample itself during inspection may be too restrictive as even a moderated tilt would bring much of the viewing area of a multiple field of view system out of focus. As such, in this case, the imaging beams sizes would need be smaller and/or the sample features themselves more sharply distinct for such imaging to be useful. To practically overcome these issues and add extra speed to the process the beams itself may be tilted [100].

Previous works in the qualification of tilted beam measurement have shown promise for possible use in in-line fabrication. However, such works have focused on experimental tilt CD-SEM imaging [99]–[104]. They did not explore the use of first principles electron scattering theory models (e.g., JMONSEL) to assess the viability of extracting/fitting geometrical 3D feature parameters from tilted angle simulations. Preceding model based efforts concentrated on the simulation of features which were large enough where the edgewidths of the features were much larger than the excitation volume of the impinging electron beam and/or mainly strove to understand the topography of features from secondary electron waveform profiles of top-down views; primarily with the goal of the extraction of critical dimension width (top, mid, and bottom) measurements [106]–[109]. Here, some of the size dimensions explored will be much closer to the excitation volume of the material and comparable to the current proportions used in fabrication of high end IC wafers. Only cursory work has been published in extracting height measurement from tilt beam simulations itself in comparison to known model input parameters [110].

Besides the aforementioned, numerous alternative approaches of 3D feature reconstruction using SEM tilted sample imaging are available [111]–[115]. These methods rely on the well-established method of stereophotogrammetry to reconstruct 3D surfaces by acquiring images from several points of view by tilting the sample. These clustered images can be then be combined digitally to construct an image containing all three dimensions [116]. Several commercially available software packages have been produced using this methodology yet they have shown to be insufficient when it comes to reconstruction of smooth 3D linewidth arrays with uniform regions characteristic of patterned wafer fabrication for integrated circuits.

Dimensional software packages will only be valuable to the semiconductor industry if both smooth and rough surfaces can be reconstructed satisfactorily [110].

Another method for recovering the third dimension of samples imaged under SEM is use of the shape-from-shading technique [117]. In this method, samples are imaged using several detector channels. The shaded images generated from the variously placed channels are then utilized to produce 3D image reconstruction of the sample [118].

3 RESEARCH OBJECTIVES AND APPROACH

In formulating multi-beam system requirements, a crucial facet is to comprehend the ways various alterations to tool design and/or operating conditions effect throughput and sensitivity for a particular target application. In order to enable this objective, development of a suite of simulation programs intended to model tool performance and to forecast system sensitivity to various applications is vital. A robust set of simulation tools facilitates numerous objectives:

- ⊞ Simulations allow a large parameter space to be assessed economically.
 - Building a large range of machine components is not practical or financially viable.
 - Various critical system component configurations may be investigated without regard to the practical details of construction.
- ⊞ Optimal operating conditions for performing inspection can be assessed without regard to the limitations of physical hardware.
 - Particularly important for developing specifications, where there is an inherent trade-off between throughput and signature sensitivity.

- Assess impacts of operating parameters over a wider range of values than is readily accessible using lab tools
 - Ex: beam current, dwell time, landing energy, and detector efficiency can be assessed over many more orders of magnitude than could be achieved in a single prototype tool.
- ⊞ System performance can be predicted for advanced application spaces.
 - The ability to fabricate suitable test structures may not be available.
 - Virtual samples can be made for device architectures, material sets, or ultra-aggressive design rules.

In this work, the development and implementation of these capabilities is carried out. Hereafter, the descriptions of the intended objectives of this study and the methods employed to achieve these goals are relayed. The main themes are bifurcated to allow for a clearer, more concise route of understanding as each theme may stand on its own accord. The following is therefore split into a section concerning defect inspection and another related to the measurement of critical dimensions.

3.1 DEFECT INSPECTION

3.1.1 RESEARCH OBJECTIVES

To understand how different variations to tool design and/or operating conditions influence throughput and sensitivity to critical level defects of particular contemporary and future target applications it is key to develop a simulation set envisioned to model tool performance and to forecast system sensitivity to these defects. Physically grounded rigorous simulations allow for cost-effective exploration of optimal operating conditions for performing defect inspection without concern for the restrictions of physical hardware. Because of the innate imbalance of defect signature sensitivity and throughput this is very important for developing specifications of operation. In the case of defect inspection, it would be principally based on finding the optimum electron dosage and pixel size for performing inspection of patterned areas with defects. As well, system performance can be predicted for advanced application spaces. Here, that would entail simulations of patterns with aggressive design rules and a variety of defect types commonly and not as commonly seen during inspection.

3.1.2 RESEARCH METHODS

In order to produce realistic virtual samples with representative defects and subsequently analyze them numerous steps of computation and manipulation are necessary. In summary, the steps are:

- (1) Generating a pixelated virtual sample containing defects

- (2) Simulating the ideal electron emission behavior of each pixel i.e., the baseline emissivity
- (3) Simulating an image by modulating the ideal emission behaviors with dose-dependent shot noise and instrumental artifacts
- (4) Analyzing the resulting image to determine the signal strength of the defect signature.

This process flowchart is illustrated graphically in Figure 3.1, and the individual steps are further detailed hereafter.

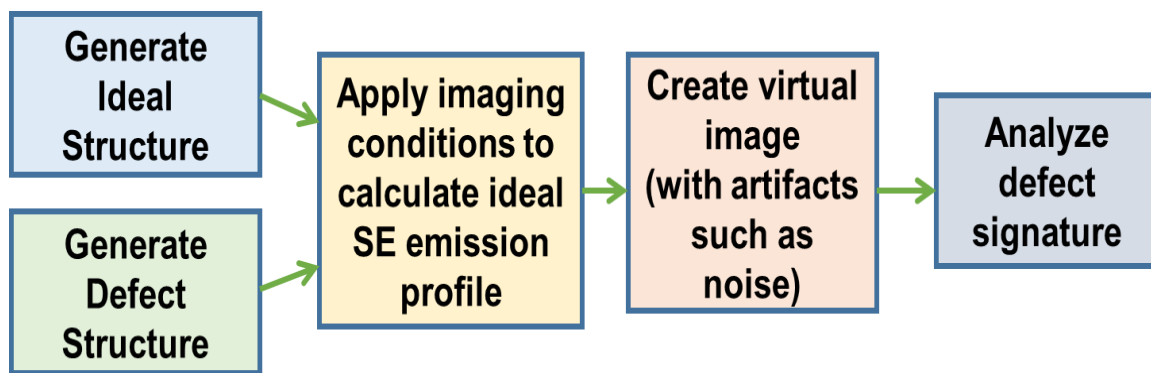


Figure 3.1 General process flow of the generation of virtual data of ideal and defect structures.

In HVM, defect densities are generally of low concentration (e.g., < 10 defects per 300 mm wafer). Throughout inspection, defects of interest comprise patterning errors, spurious particles, and residues/contamination, etc. To simulate full wafers with scant defects at nanometer resolution would be computationally exorbitant, necessitating management of numerous terabytes of data. To generate images in sizes analogous to those anticipated from the tool, in practical amounts of time, a unit cell of a repeating pattern structure is simulated in

order to include a precise match of the pixel size to the pitch of the structure, permitting smaller images to be tiled in a modular scheme. Defect cell blocks are created by varying an ideal block by adding a set type and size of defect. As such, tiling ideal and defect blocks are used to create a large scale virtual sample [119]. In this defect inspection case, each block is designed to be 32×32 pixels. Though, of greater importance is keeping an integer number of pixels to cover the periodicity in x and y directions of the defined background exactly and that all scatterings from a defect are contained well within tile boundaries.

3.1.2.1 GENERATION OF IDEAL AND DEFECT STRUCTURE

Prior to generating 3D models of test structures, the concept design and schematic understanding of the layout must first be resolved. Figure 3.2 shows an example setup of one such case. The schematic shows all pertinent size information including structure height, width, length, pitch, and material type.

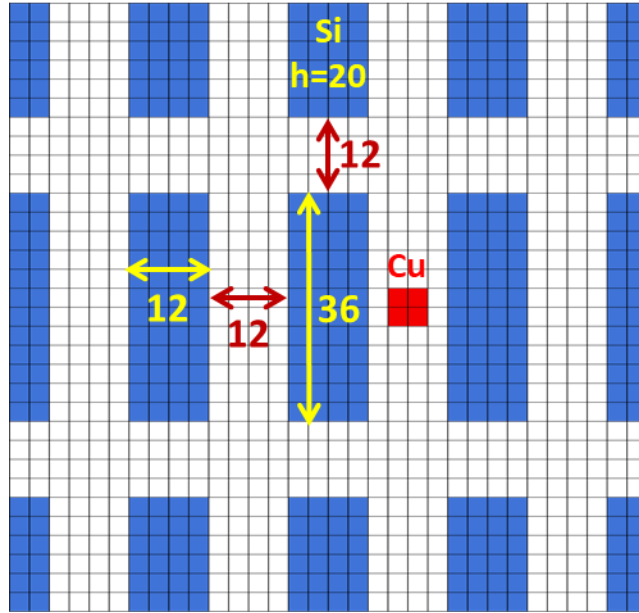


Figure 3.2 Schematic illustration of a test case of FinFET links

Using the information from the base schematic, a 3D model is then rendered. Figure 3.3 shows the flow of this task from concept design, schematic illustration, and the 3D rendering of the structures. The process is the same for the ideal case less defect inclusion.

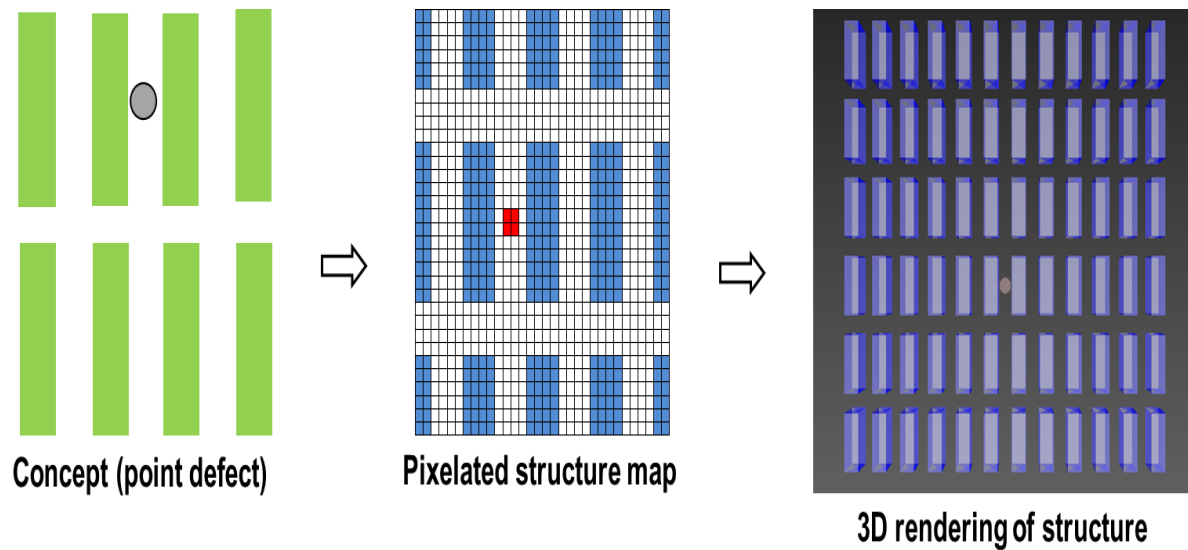


Figure 3.3 Process flow of structure generation from concept design, schematic illustration, and the 3D rendering of the structures.

In this work various defect types were simulated, examples of which are presented in Figure 3.4. To cover a thorough design of experiment (DOE) a specified defect type, e.g., a bridge, was replicated with the defect feature appearing in a succession of sizes and aspect ratios, each of which is relative to the design rule of the template structure. The set of defect classes replicated those present in the SUNY Poly SEMATECH's intentional defect array (IDA) test structure wafers used in experimental assessments of defect inspection technologies (illustratively represented in Figure 3.6). Pixel sizes were typically held equal to the full width at half maximum (FWHM) of the spot size. The outputs then were modular cell images containing and free from defects, which gave the expected defect signatures for further analysis.

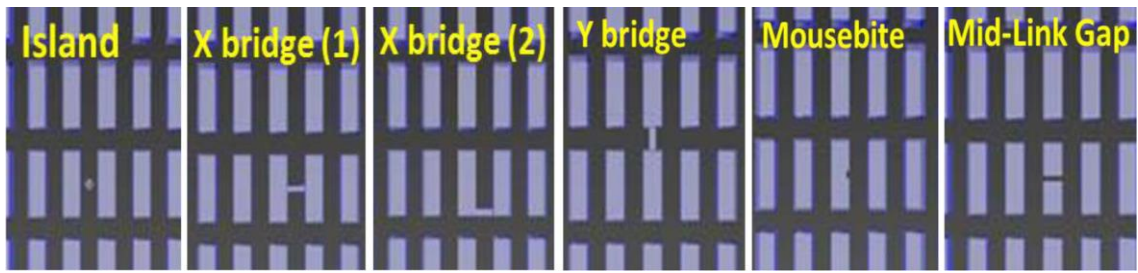


Figure 3.4 Sample set of 3D rendered designs of the defect tiles.

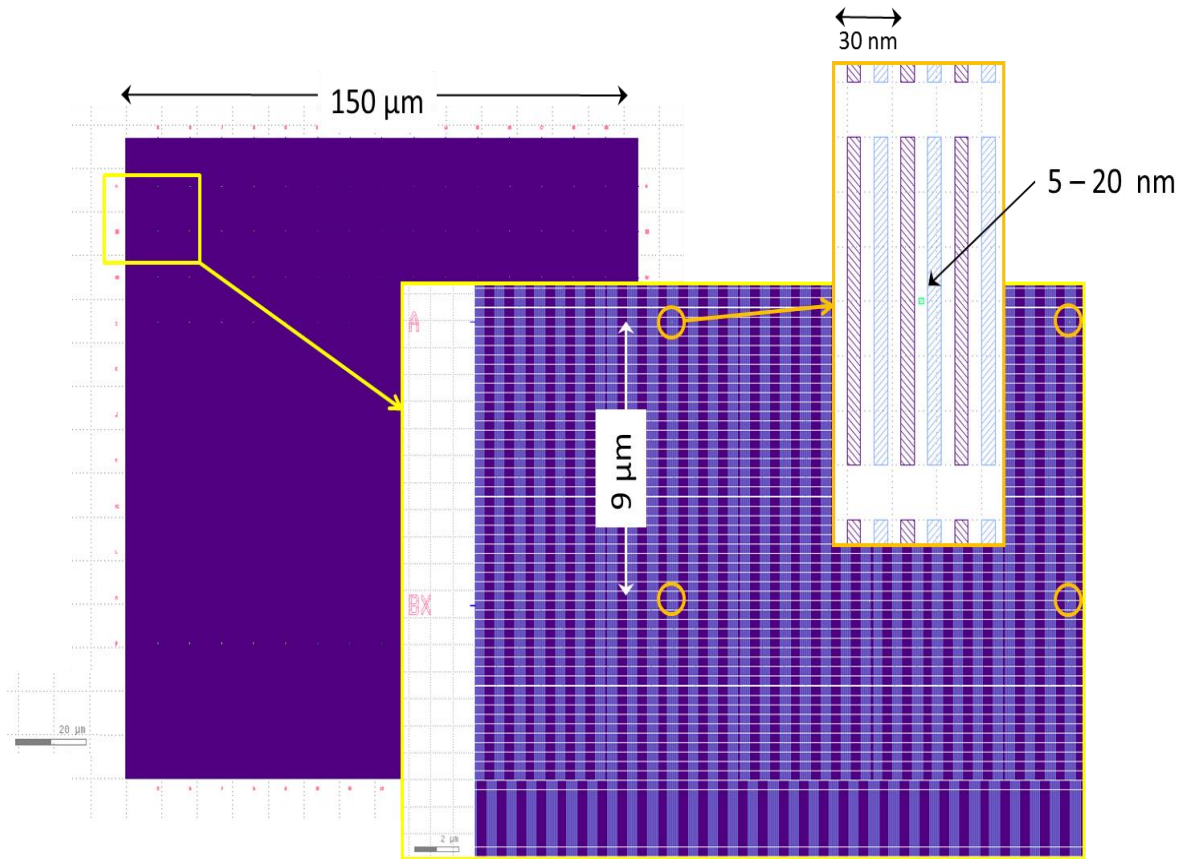


Figure 3.5 Schematic illustration of an IDA map.

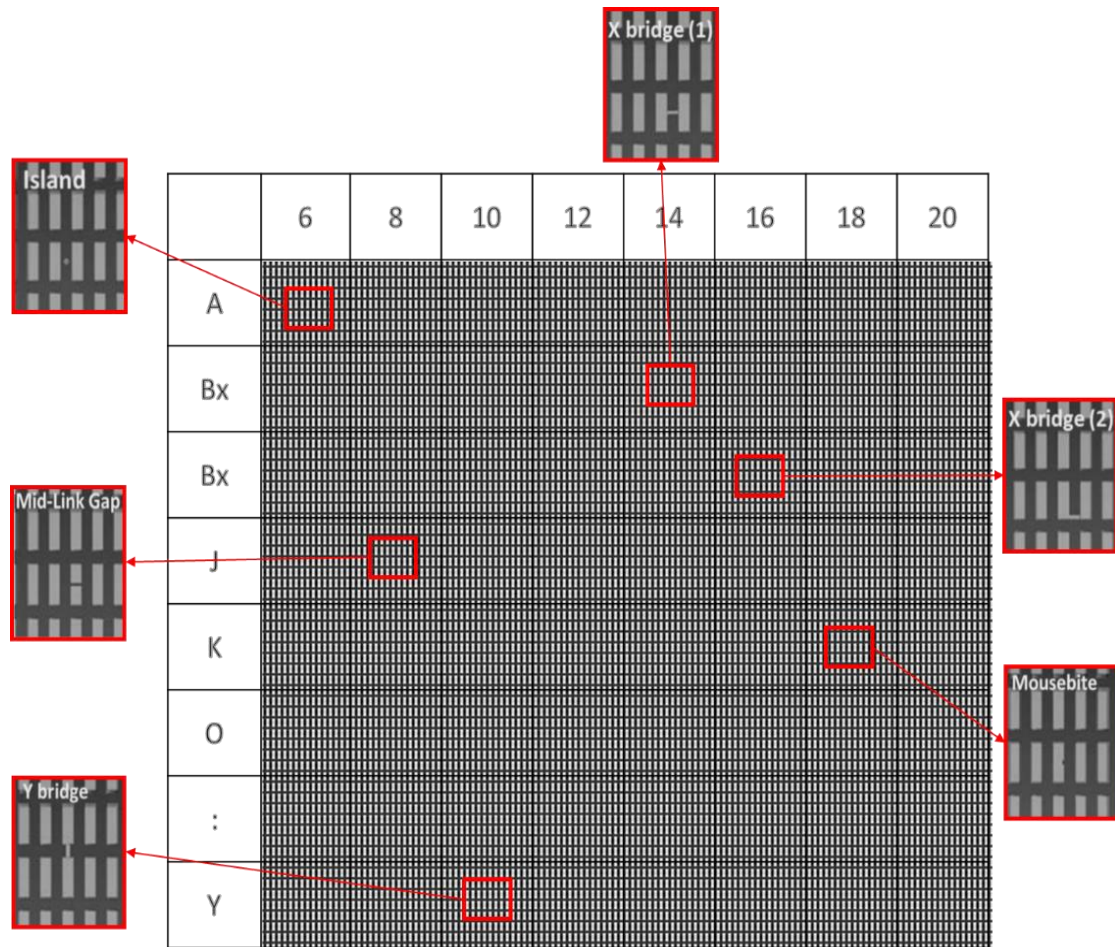


Figure 3.6 Illustrated example of an IDA map with rows representing defect types and columns indicating size of defect.

3.1.2.2 APPLYING IMAGING CONDITIONS & CALCULATION OF ELECTRON EMISSION

BEHAVIOR

Baseline emissivity calculations of secondary electrons (SE) were performed by the usage of Java Monte Carlo Simulator of Secondary Electrons (JMONSEL), a 3D SEM simulating program developed at the National Institute of Standards and Technology (NIST). The simulator software employs finite-element analysis to track emitted primary electrons as they enter a sample material, scatter, lose energy, and generate SEs and backscattered electrons

(BSEs) [120]–[126]. For purposes of this study, SEs are demarcated as those electrons reaching the detector with ≤ 50 eV energy, while BSEs are those with > 50 eV. By way of observing the electrons that depart the sample material and are caught by the detector, the electron yields can be established at any position chosen as a target pixel. Pixel intensities may then be represented as one-dimensional (1D) linescans or two-dimensional (2D) matrices of either SEs or BSEs. The physical models implemented in JMONSEL are the preeminent known models in the literature in the energy ranges used at this juncture, having complete transparency in their documentation, definition, and execution (Appendix: A).

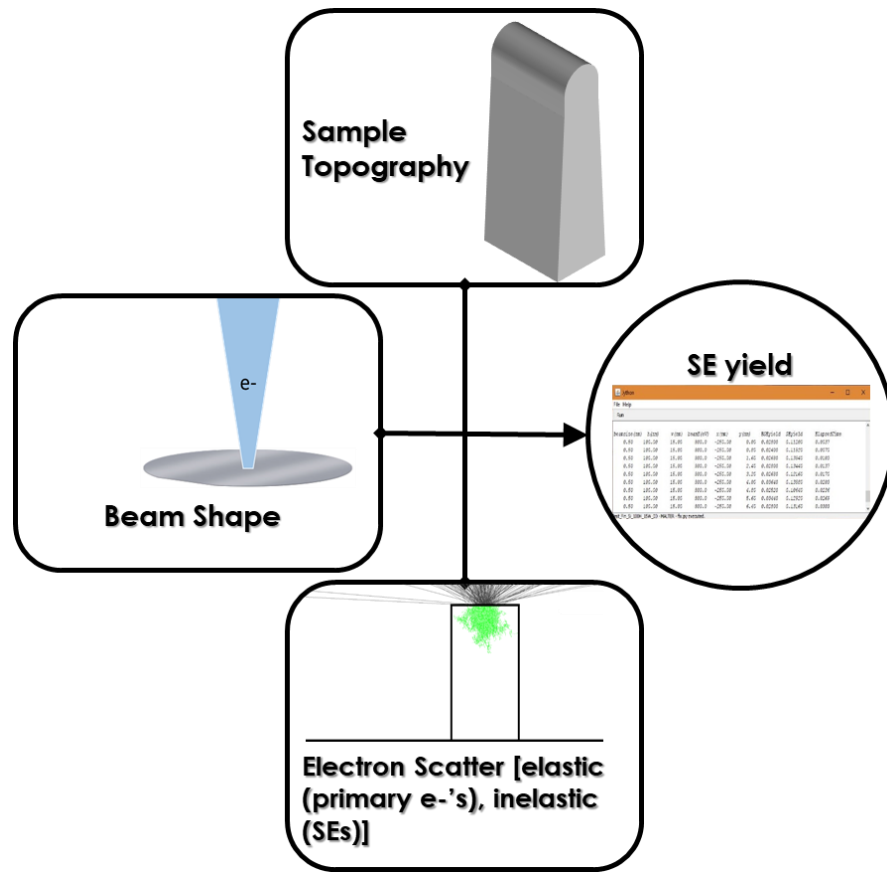


Figure 3.7 Illustration of the key input components for a JMONSEL simulation. The sample topography, characteristics of the impinging beam, and the scattering models are all set before calculation of electron yield data can be commenced.

Here, JMONSEL is taken as a virtual SEM whereby the operator stipulates the sample and feature geometry along with the material sets. Unless roughness is explicitly incorporated, the structures are idealized, having zero roughness and do not replicate process variations. Subsequent to describing the target, the exact parameters of imaging of the model structures needs are defined. These include imaging angle, scan origination point, beam energy (in electron volts) and size (in nanometers), number of incident electrons per pixel, pixel size, and number of pixels to be imaged. From here, the implementation of the simulation technique may commence. Calculating the SE and BSE yield values with small pixels allows for the

Research Objectives and Approach |73

emission profiles of larger pixels to be readily obtained by appropriate averaging. JMONSEL can also monitor charging phenomena in and around the sample, though these effects were negligible for the samples considered here under the conditions modeled.

For each pixel, N incident electrons are simulated and the numbers of SEs and BSEs reaching the detector are registered. The ratio of electrons arriving at the detector to the number incident at the sample is considered the yield. By way of SE being a normally stochastic process, the uncertainty in the projected yield scales as the inverse square root of the number of incident electron trials. Therefore, $N = 12,000$ produces an emission coefficient with roughly 1% uncertainty in the predicted value (suitable for rapid screening), while $N = 100,000$ produces an estimate with an uncertainty of about 0.3%. Even though using low numbers of doses in the simulations would imitate the effect of shot noise in a real image, large values of N were used in these trials to generate ideal cell images. The effects of dose-limited shot noise are incorporated afterwards. An example rendering of the script is provided below (Appendix: D).

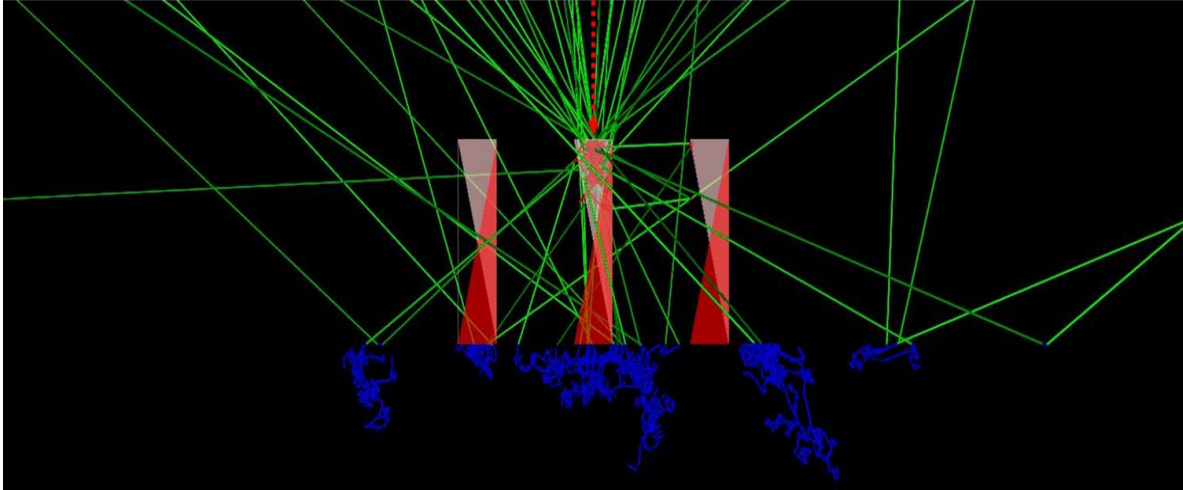


Figure 3.8 Example of simulated electron trajectories from JMONSEL. The red dashed arrow points to the location of the center of a designated pixel, and incident electrons enter the sample vertically in a Gaussian distribution of a designated spot size centered on that location.

3.1.2.3 *CREATION OF VIRTUAL IMAGES*

The entire flow of computation and tiling to generate a realistic image is shown in Figure 3.9. The ideal simulated yields are multiplied by a nominal dose to replicate an SE intensity map for a given combination of probe current and pixel dwell time. These pixel intensities are then scaled to an 8-bit grayscale image format. The ideal image cells are then tiled to generate a virtual image with dimensions characteristic of a real SEM image e.g., 1024×1024 pixels. A cell containing an intentional defect is located at the center of the image. Post processing is then executed to mimic the effects of imaging with a real tool i.e., including noise, Fourier masks representative of the tool's contrast transfer function, and other artifacts, such as detector dark current.

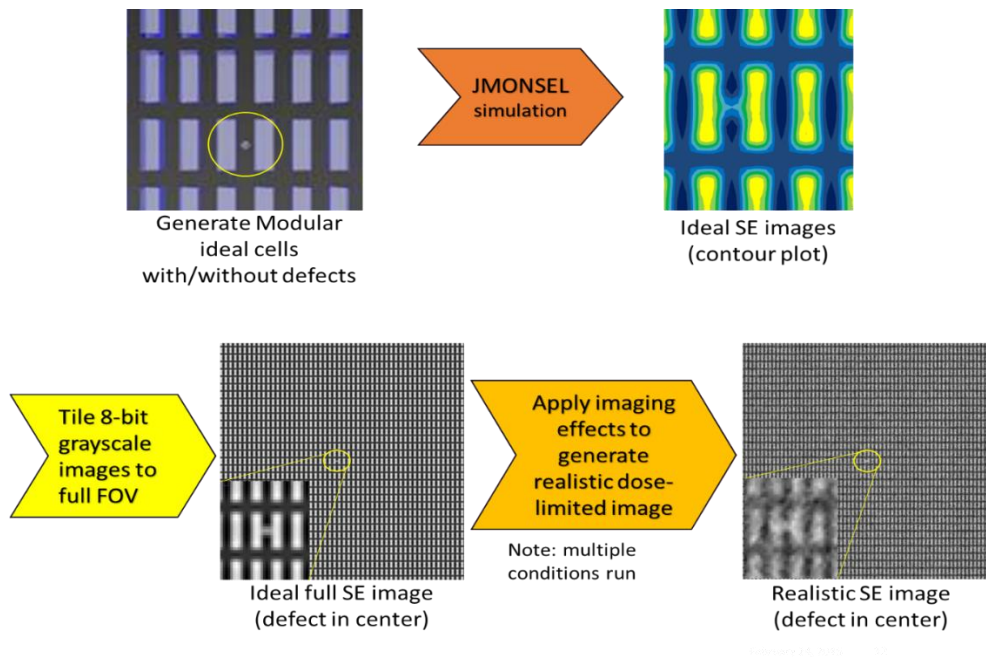


Figure 3.9 Procedural flow for generation of virtual image data, from sample definition to tiled image to image post processing, yielding a realistic, quantitatively representative image for a multi-electron beam inspection tool.

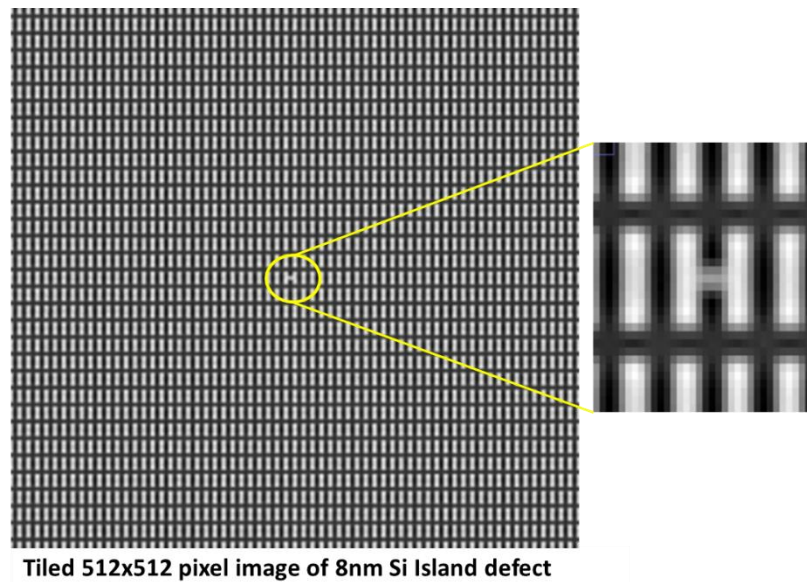


Figure 3.10 Sample grayscale image generated by converting the JMONSEL SE yields of a test structure (3 nm pixel size, 3 nm probe 1σ , 500 eV, 8 nm Si island) into a grayscale image (third step in Figure 3.9). The central defect cell block is magnified for clarity.

The two key sources of noise incorporated into this study are (1) the dose-dependent shot noise of electron emission from the sample, which modulates the ideal SE emission profile, and (2) the uncorrelated electronic noise (i.e., dark current) present in the detection system. Each factor has a significant impact on the sensitivity to defect of a system.

Shot noise arises in the image because of the fact that SE emission is a stochastic process that can be described with Poisson statistics. For a high electron dosage N , the process follows a normal distribution and the signal-to-noise ratio (SNR) is given by $SNR = N/\sqrt{N}$. However, in defect inspection maximizing throughput is vital, meaning that a system will always operate at the minimally feasible dose. Hence, the appropriate model for SE emission in this case is a Poisson distribution. The effect of shot noise on the virtual images is incorporated by taking the value for the SE emission coefficient of each pixel (x, y) projected by JMONSEL and multiplying by the nominal electron dose to produce a mean emission count of α . Finally, the integer emission intensity value k assigned to that pixel is then selected from a randomly generated Poisson probability distribution P_k according to

$$P_k(x, y) = \frac{\alpha_{x,y}^k}{k!} e^{-\alpha_{x,y}}.$$

Equation 14 Poisson probability mass function representing shot noise.

The effect of detector noise is modeled as an uncorrelated white noise with a user-determined amplitude μ . The root-mean-square (rms) of the noise σ is set to be the square root of the amplitude. Therefore, the final virtual image is attained by taking each shot-noise modulated

pixel and then adding a white noise component $n(x, y)$ selected from a normal distribution given by

$$n(x, y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\mu^2}{2\sigma^2}}.$$

Equation 15 White noise component described by a normal distribution.

The contrast transfer function (CTF) of an instrument describes the fidelity of information in a signal stream of an SEM and is embodied in the final image as a function of spatial frequency [127]–[129]. Generally, all imaging instruments experience a decaying response function as spatial frequencies approach the Nyquist limit of the system and as the signal intensity falls below the noise floor. This effect can be duplicated through the use of an appropriate Fourier mask applied to the image. Optical aberrations as well astigmatisms can be represented in this function.

3.1.2.4 ANALYSIS OF DEFECT SIGNATURE

In this study, the defect sensitivity analysis provides a quantitative means for analyzing the relative strength of the defect signature in an image, real or virtual. Broadly, a differential image is obtained by subtracting two nominally identical images – a reference image of a defect free region and a test image containing a defect – except for the noise on each. Here, the defect signature is the SNR of the differential image, explicitly, the residual image of the defect relative to the noise differential. The SNR can be utilized as a figure of merit to evaluate the impact of defect size, material contrast, image dose, tool performance, detector noise, etc. It ought to be noted that this is not a defect inspection algorithm: this process requires knowing

the precise location of the intentional defect beforehand. It should not be confused with the defect inspection algorithms, employed by manufacturers of inspection tools, which compare images to detect defects, such as with die-to-die (random mode), cell-to-cell (array mode), or die-to-database type inspections. Likewise, the SNR values produced by this algorithm are only an internal metric and cannot be equated to the SNR values for images generated by an actual tool. Nonetheless, the “trends” forecast by this analysis should mirror the influence of sample differences, tool design, or operating conditions on the ability of an actual inspection algorithm to detect defects.

3.1.2.4.1 National Institute of Standards and Technology Defect Sensitivity Analysis Algorithm

The defect sensitivity algorithm utilized in this study was developed as a joint effort between SUNY Poly SEMATECH and the National Institute of Standards and Technology (NIST) for the purpose of assessing pattern defects technologies and test structures. The NIST approach is to use the singular input image upon itself to locate and identify the defect in question. First, Fourier filters are applied to the raw image so that spurious high and low frequency features are removed (standard 0.01 high pass and 0.99 low pass). Next, a central region of interest (ROI) with the defect is demarcated – this region will be the target defect test case. In the present case, the defect is centrally located in the image. Without a second reference image available, shifts in the single standard image are used to generate a “reference” for use in making differential images. For this work, four similar reference regions are used, as illustrated in Figure 3.11. The reference regions are the same size as the test region, but shifted so that there is no overlap with the central ROI.

By means of the enhanced correlation coefficient algorithm [130], all shifts are correlated back onto the central ROI. Not all periodicity results in direct mirror images of shifted regions. As such, correlation back to the target ROI will require use of an offset affine transformation. As can be seen from Figure 3.11, there is no direct overlap of the regions. Excess pattern is cropped off. The average of the reference shifts after correlation is subtracted from the target ROI to produce the differential image so that the defect signature is more prevalent. Size (measured in contiguous pixels) and intensity thresholds are applied to the differential image to identify the defect signature. Finally, the SNR is calculated by subtracting the mean intensity of the noise pixels from the mean intensity of the defect signature and scaling to the standard deviation of the noise pixel intensity. The intermediate steps of this process are shown in the example given in Figure 3.12 which shows the analysis of a virtual image with a mid-fin bridge defect (Appendix: C).

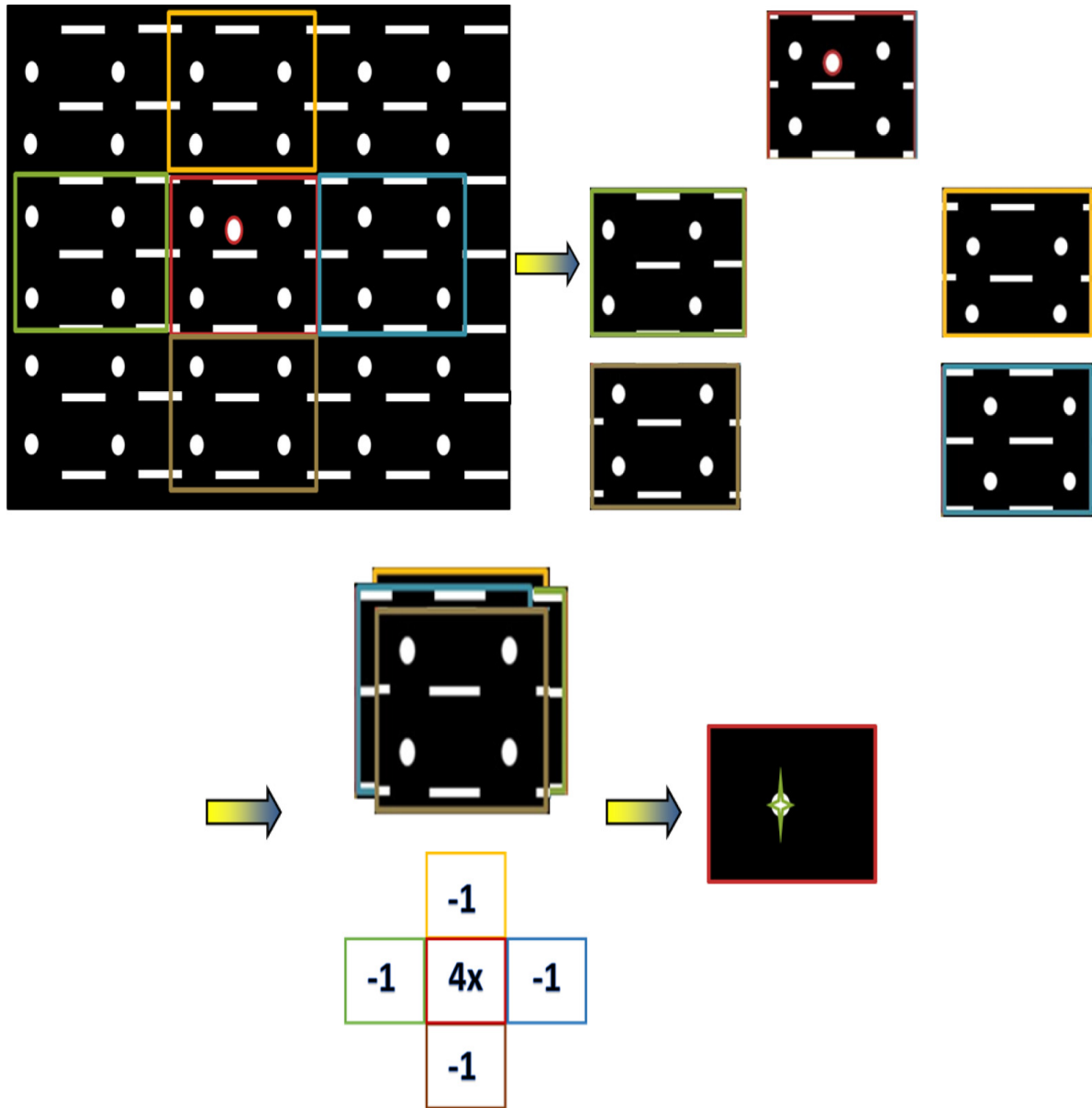


Figure 3.11 Process flow of the defect sensitivity algorithm. The algorithm starts by creating a target ROI in the center (red square—top left image) and four reference shifts of same size. Next, it correlates back the shifted images to the target ROI and then subtracts the average image from among the images from the shifted locations from the target ROI. In this case, we multiply the central square four times and subtract each of the four shifts so that the defect is equally weighted in the calculation. Finally, any uncommon overlap is cropped, leaving the differential image to be processed by the threshold parameters.

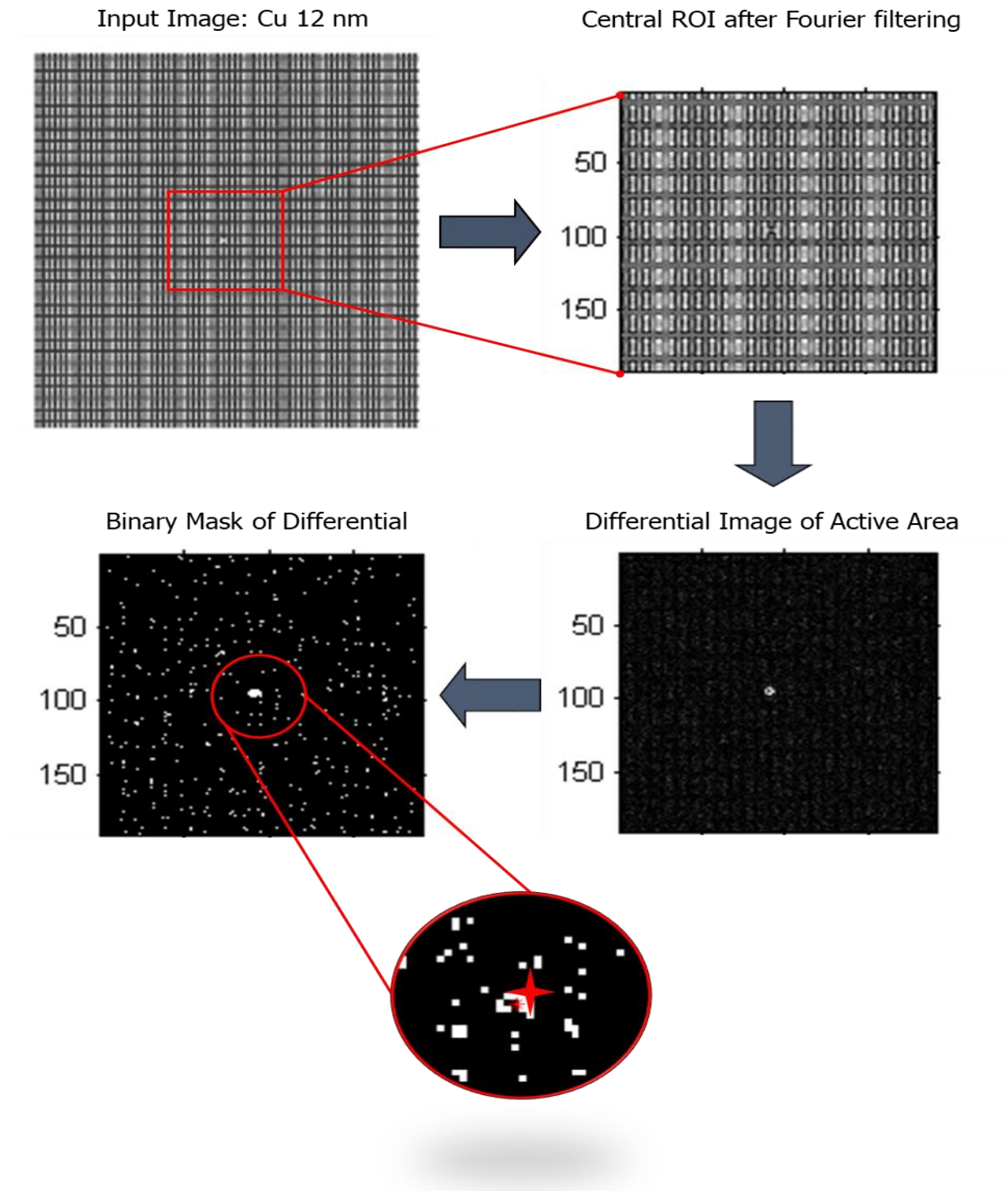


Figure 3.12 Algorithm ran on an example ideal simulated image with central defect. Identified defect is shown by the red star indicated on the binary mask of the differential image.

3.1.2.5 Zeiss MultiSEM System

The real imaging data was collected from a 61-parallel beam Zeiss MultiSEM system [77]. This is an engineering tool located in Oberkochen, Germany. Here, a brief outline of the multi-beam system operating principles is provided (for a more detailed description of this multi-beam technology see [82]–[84], [131]). In recent years the multi-beam technology has been developed for brain mapping applications in neuroscience [78]–[80]. A schematic diagram of the multi-beam system is illustrated in Figure 3.13.

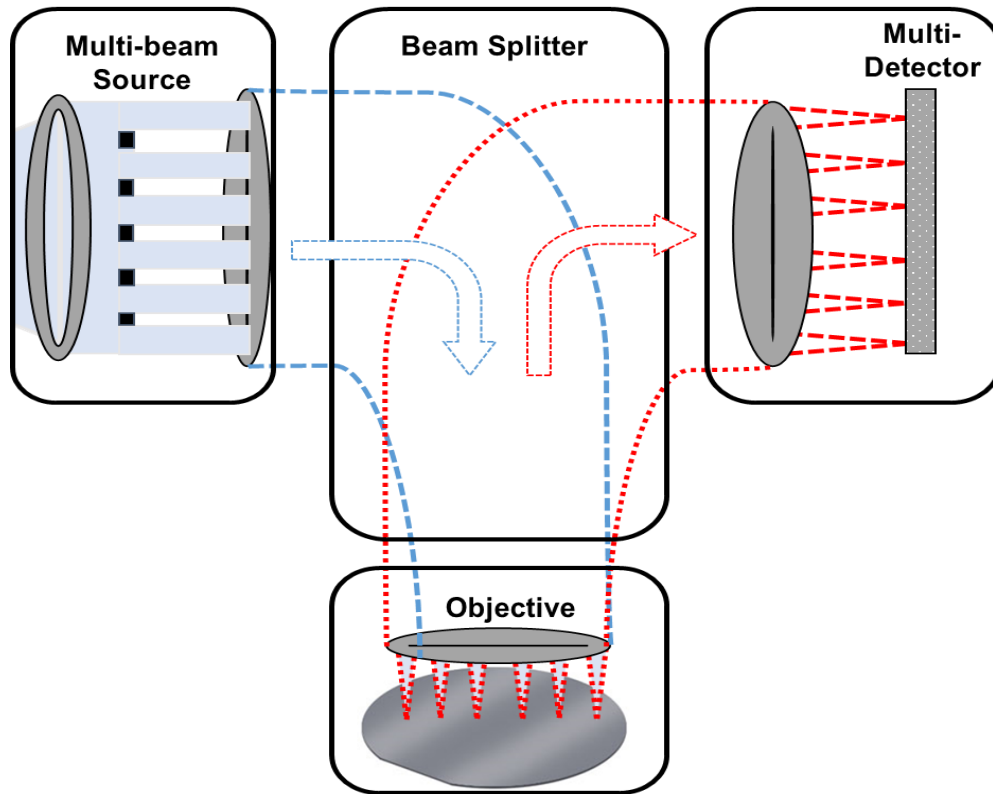


Figure 3.13 Schematic representation of the multi-beam imaging setup. Primary electrons (blue dotted lines) are focused onto a sample and separated by a beam splitter from the secondary electrons (red dotted lines) that are detected simultaneously by a multi-detector.

The system utilizes multiple beams in one electron optical column and one detector for each beamlet. Firstly, a multi-beam electron source creates an array of electron beams that are consequently focused onto a sample. Primary electron beams are organized in a hexagonal configuration as in Figure 3.14 to minimize electron optical aberrations. The secondary electrons that emanate from each primary electron spot are relayed onto a multi-detector which has one detection unit for each beam. A magnetic beam splitter is used to separate primary and secondary electron beams.

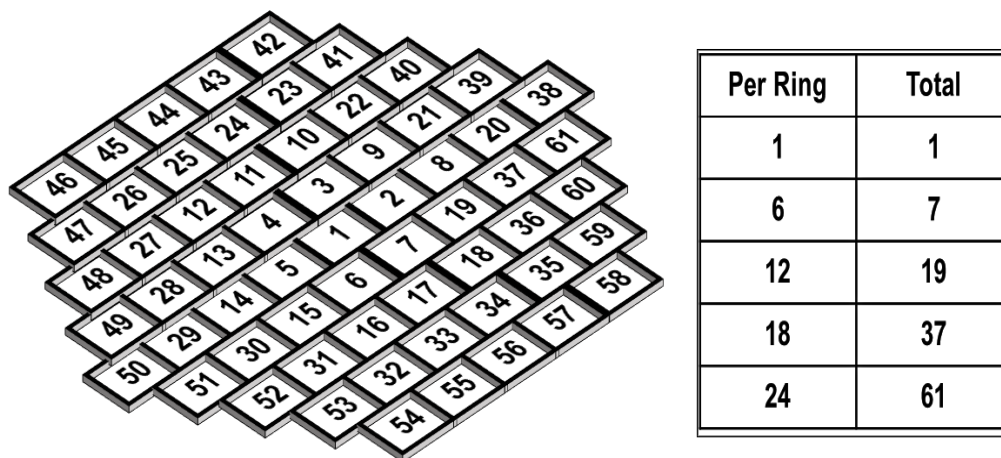


Figure 3.14 Hexagonal arrangement of beam layout; shown for a 61-beam configuration.

As illustrated in Figure 3.15, each imaging beam produces a single field of view (FOV) image of a defined area. These images are joined together to construct an aggregate hexagonal image that is known as the multi-beam field of view (mFOV). Subsequently, the imaging process is reiterated at the ensuing location once the sample stage is moved. The process is repeated until the required sample space is imaged. Figure 3.16 demonstrates this process on a wafer sample.

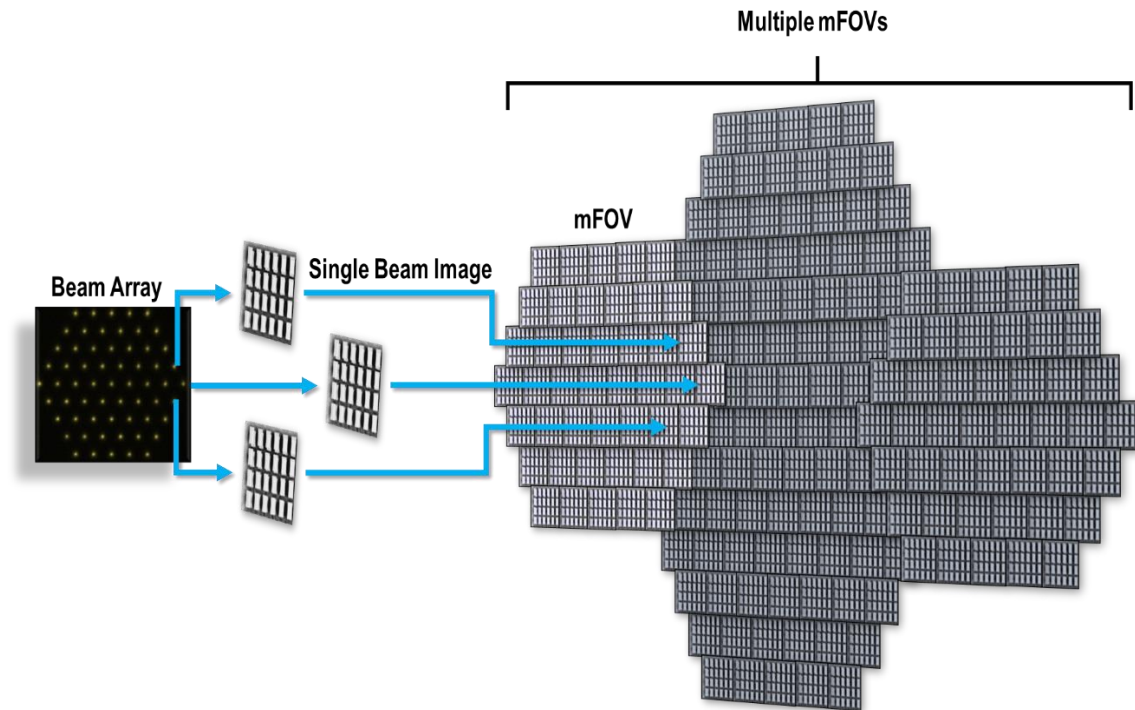


Figure 3.15 Essential workflow of the Zeiss multi-beam system.

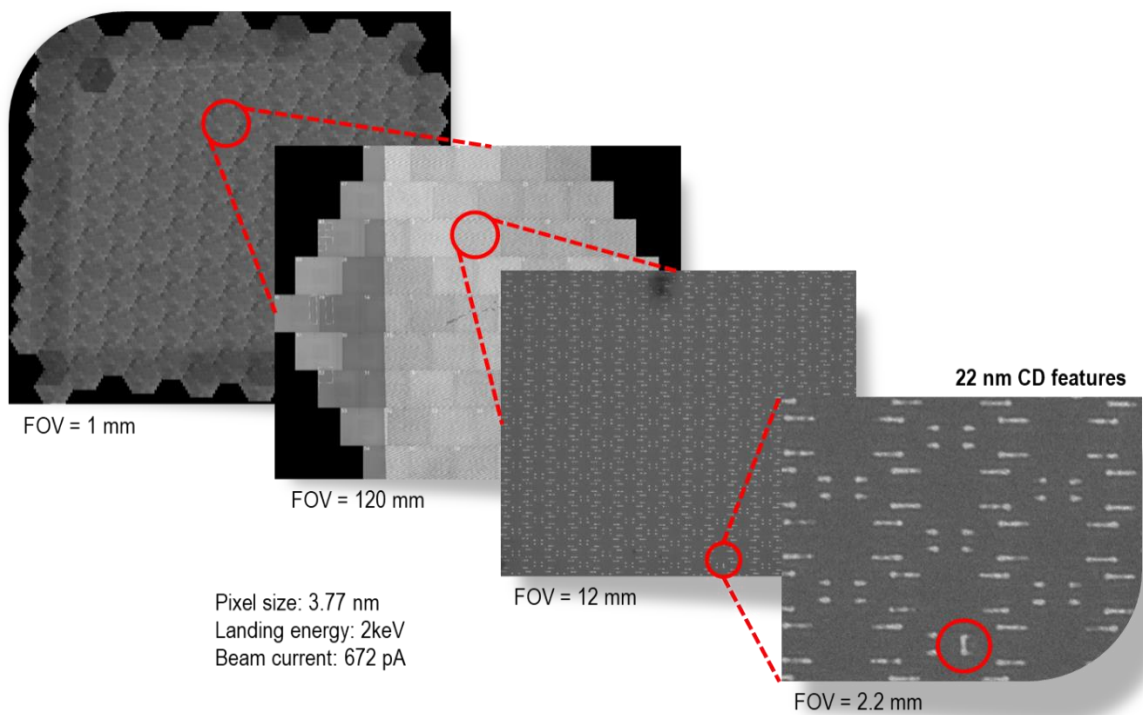


Figure 3.16 Demonstrative workflow illustrated on a 22nm wafer sample [85].

SUNY Polytechnic SEMATECH explored a wide range of electron beam inspection technologies. Subsequent to exploration, later focus was strictly placed upon massively parallel electron beam options. The primary considerations here included: throughput, sensitivity, technology readiness, performance results, and ability to scale to HVM. Based on evaluation, the Carl Zeiss GmbH technology was selected as best the prospect and the reason images of real defect wafer and other samples were produced by its use.

3.2 CRITICAL DIMENSION MEASUREMENT

3.2.1 RESEARCH OBJECTIVES

As mentioned above, vigorous simulations facilitate many advantageous aspects of virtual experimentation. In creating simulated images of non-planar (e.g., FinFET) designs, to investigate the use of first principles electron scattering theory simulations, in order assess the viability of extracting/fitting 3D feature parameters from tilted angles much of the same process described above will apply here as well. Though, design parameters, features shapes, imaging angles, and material compositions may be different. Structures will be based on currently researched and produced FinFETs in industry (Figure 3.17).

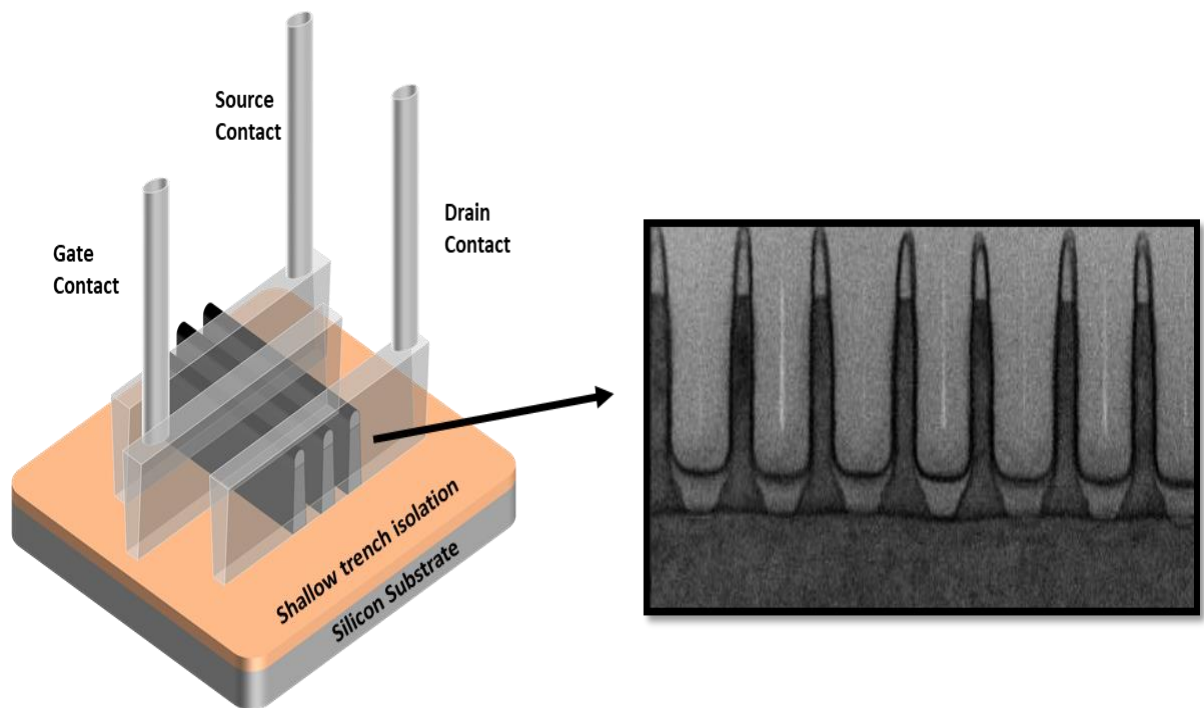


Figure 3.17 Illustrative FinFET example of currently produced device architecture. (TEM image courtesy of GlobalFoundries)

3.2.2 RESEARCH METHODS

Largely, the steps used previously are reiterated here: (1) generate a pixelated virtual sample containing features, (2) simulate the ideal electron emission behavior of each pixel i.e., the baseline emissivity, (3) simulate an image by modulating the ideal emission behaviors with dose-dependent shot noise and instrumental artifacts, and (4) analyze the resulting image to determine critical dimension variation from modeled inputs for feature parameters as a function of the imaging conditions. To be noted, the imaging here takes place from several different viewpoints i.e., angle tilts of the sample to capture the top and bottom of the structure. This process flowchart is illustrated graphically in Figure 3.18, and the individual steps are further detailed henceforth.

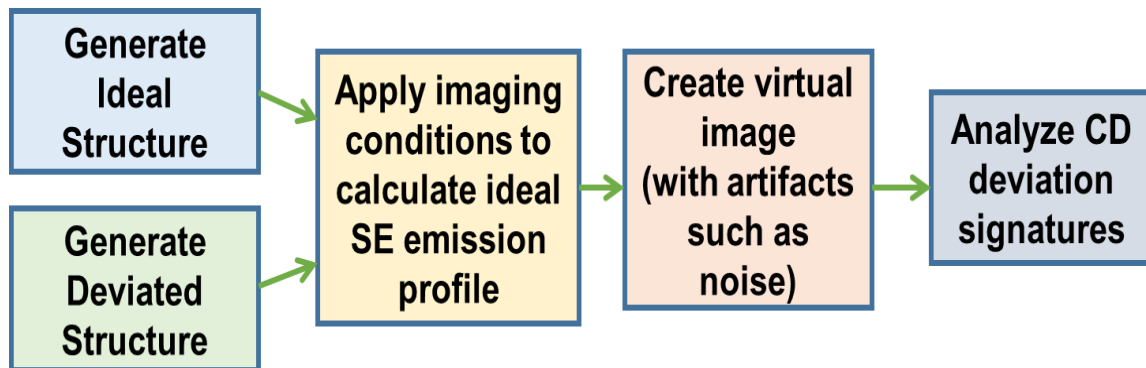


Figure 3.18 General process flow of the generation of virtual data of ideal and deviated structures.

3.2.2.1 GENERATION OF IDEAL AND PROCESS DEVIATED STRUCTURE

Firstly, the concept design and schematic understanding of the layout must be determined prior to generation of 3D models of the test structure. Figure 3.19 shows an example setup of one

such case. The schematic shows all pertinent size information including structure height, width, pitch, and material type.

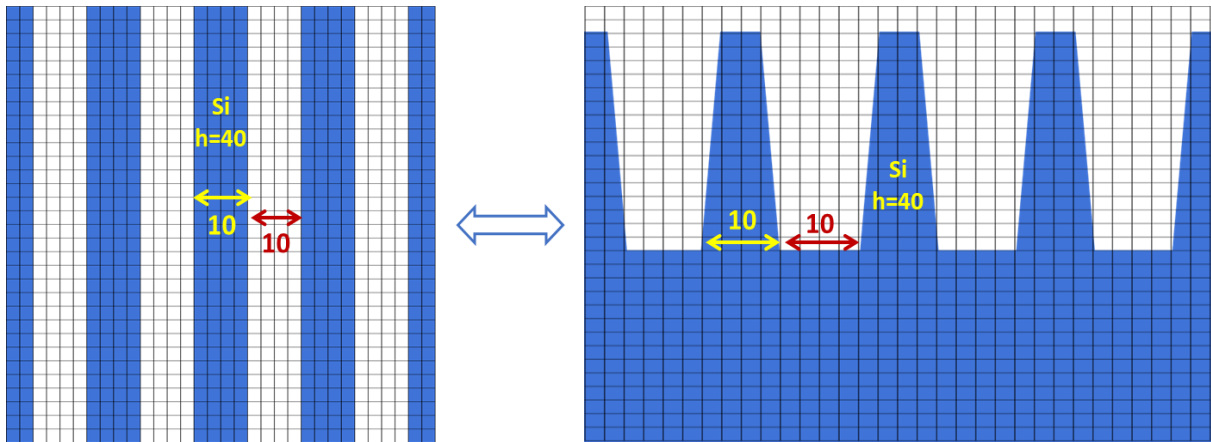


Figure 3.19 Schematic illustration of a test case of basic fin structures from top-down and cross-sectional views.

From schematic information a 3D model is created. Figure 3.20 shows the movement of this task from concept design, schematic illustration, and the 3D rendering of the structures.

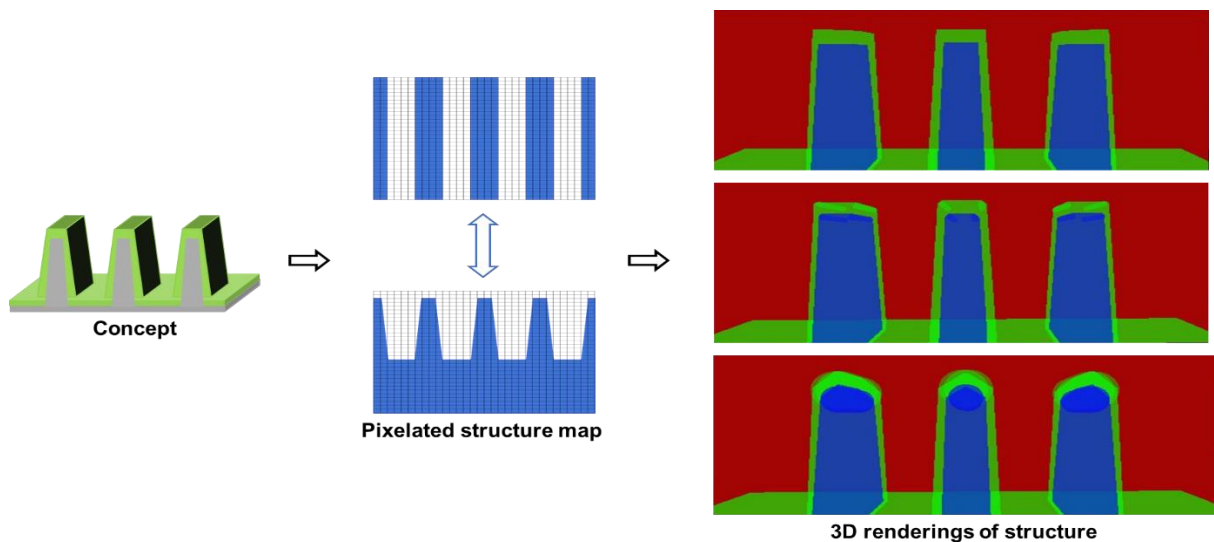


Figure 3.20 Process flow of structure generation from concept design, schematic illustration, and the 3D renderings of the structures process varied critical dimensions.

In this work a number of CD process variations were simulated, examples of which are feature height and sidewall angle (SWA). To comprise a complete DOE, a given class of CD variation, e.g., fin height, was replicated with the feature appearing in a sequence of sizes and/or aspect ratios. The set of variations aim to imitate those most commonly present in real fabricated devices. Pixel sizes were typically held equal to the FWHM of the spot size. The outputs are modular cell images with and without variations, which give the expected CD changes for further analysis.

3.2.2.2 APPLYING IMAGING CONDITIONS & CALCULATION OF ELECTRON EMISSION

BEHAVIOR

The SE baseline emissivity calculations for these structures were performed using JMONSEL. JMONSEL uses finite-element analysis to track primary electrons as they enter a material, scatter, lose energy, and generate SEs and backscattered electrons (BSEs) [120]–[126]. Here, SEs are defined as those reaching the detector with ≤ 50 eV energy, while BSEs are those with > 50 eV. Through observing the electrons that depart the material and are then identified by a detector, the electron yields can be found at any desired pixel. These pixel intensities can then be graphed as 1D linescans or 2D matrices (either SEs or BSEs). The physical models in JMONSEL are the best known models in the literature in the energy ranges that were used here, providing complete transparency in their documentation, definition, and execution. Unlike the simulations performed for defect inspection use case, for the measurement of CDs, the emissivity calculations were taken for beams impinging the material at tilt angles in the ranges that would be used by real tools given the proximity of neighboring features (Figure 3.21).

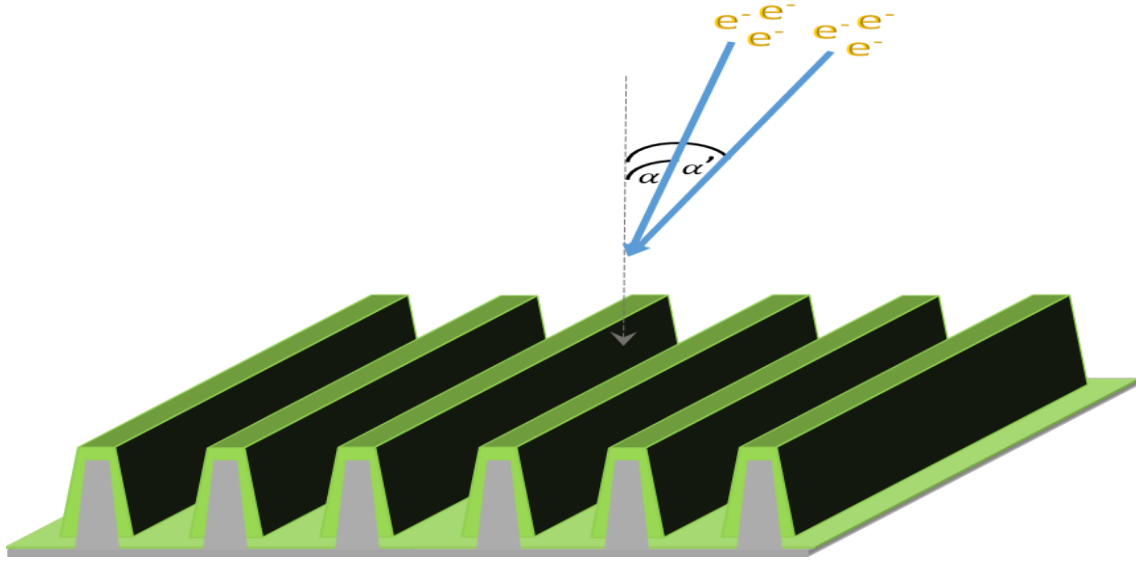


Figure 3.21 Illustration of a Fin array being struck by electron beams at successively different angles.

3.2.2.3 CREATION OF VIRTUAL IMAGES

Subsequent to the calculation of the pixel electron yields for SEs via JMONSEL, simulated yields are multiplied by a nominal dose to replicate an SE intensity map for a given combination of probe current and pixel dwell time. These intensities are then converted to 8-bit format grayscale images which then allow the treatment of these images as any other digital image and provide a great analog to the way digital images are created in actual electron beam tools. Figure 3.22 shows a sample output of such a conversion.

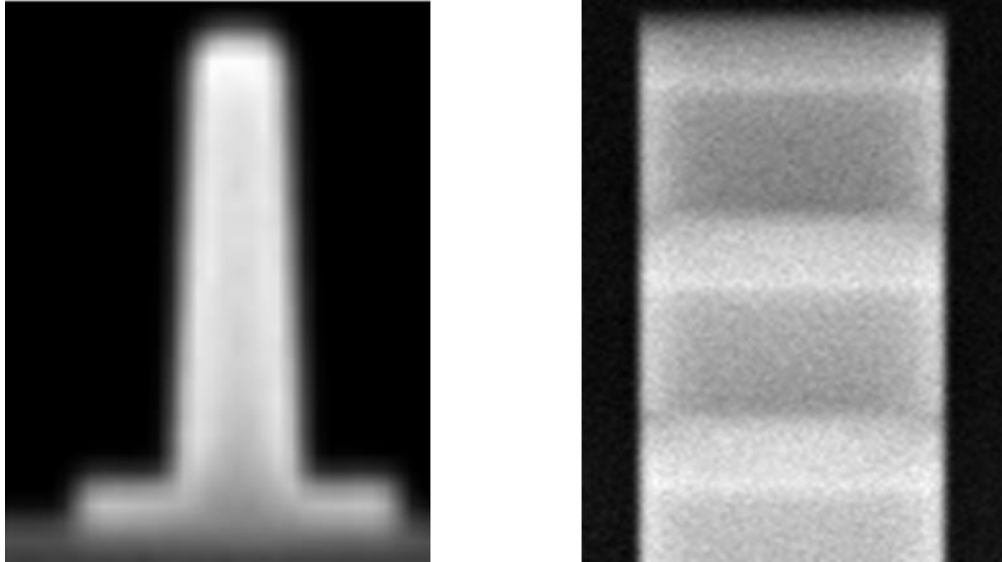


Figure 3.22 Sample grayscale images generated by converting the JMONSEL SE yields of a test structure into a grayscale image. Left – cross section view of Fin (10,000 electron trajectories per pixel). Right – 45 degree angle view of Fins (1,000 electron trajectories per pixel).

As with the previously mentioned, the two main sources of noise integrated into this study are the dose-dependent shot noise of electron emission from the sample, which modulates the SE emission profile, and the uncorrelated electronic noise (e.g., dark current) existent in the detection system. Each factor has an impact on the sensitivity of measurement of a critical dimension.

3.2.2.4 ANALYSIS OF CRITICAL DIMENSION DEVIATION SIGNATURES

The calculation of critical dimensions such as height from tilt images of a feature has a well-known form [99], [101]. First, the target feature is scanned by the electron beam at two different angles from normal, creating a pair of images of the same feature. If the feature is a simple shape (Figure 3.23) the height can be determined using straightforward trigonometry.

Though, the features in question here are more complex i.e., the fin of a FinFET. Therefore, a trapezoidal form is assumed for the fin structure investigated in this study, as in Figure 3.24.

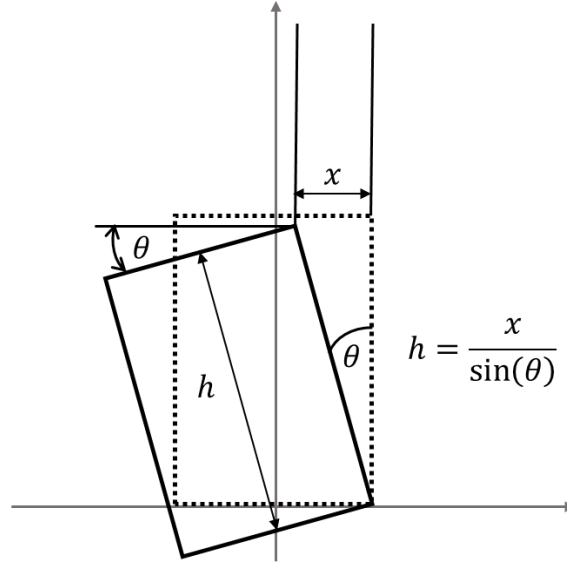


Figure 3.23 Simple rectangular shape is tilted counterclockwise by an angle θ generating an edgewidth x that can be measured. The height of the shape is then calculated by simple trigonometry.

The geometrical relationship between the height (h), sidewall angle (θ), and edgewidths (EW_1 & EW_2) measured at two different tilting angles (α_1 & α_2) where $\alpha_2 > \alpha_1$ and $EW_2 > EW_1$ are given by the following:

$$\tan(\alpha_2) = \frac{EW_2 - EW_1 + h * \tan(\alpha_1)}{h} \rightarrow \tan(\alpha_2) - \tan(\alpha_1) = \frac{EW_2 - EW_1}{h},$$

Equation 16 Geometric relationship of height to edgewidths and tilt angles for a trapezoidal structure.

$$\tan(\theta) = \frac{h}{EW_1 - h * \tan(\alpha_1)}.$$

Equation 17 Geometric relationship of sidewall angle to height and edgewidth & tilt angle from one image.

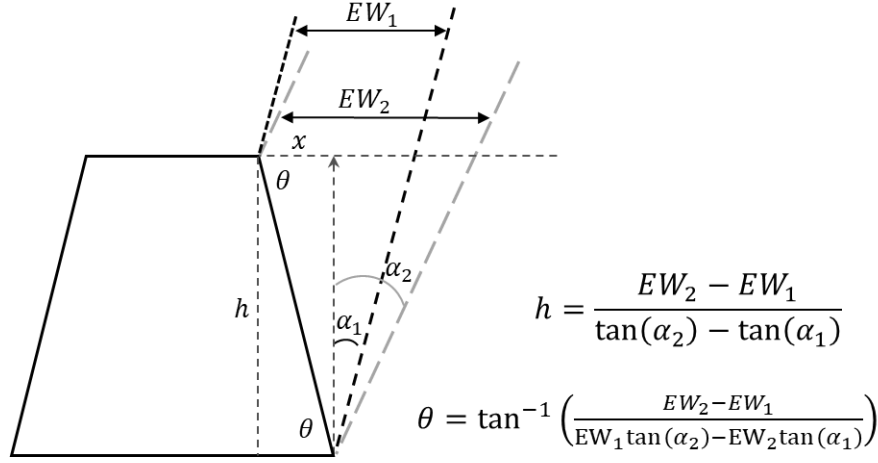


Figure 3.24 Operating method of the tilt-beam imaging: target feature is scanned twice with an electron beam at two different incident angles (α_1 & α_2) creating two images with different edgewidths (EW_1 & EW_2). Calculation of feature height and SWA can be accomplished by the relation of the tilt angles and edgewidths.

Solving the above equations for height and sidewall angle produces the following:

$$h = \frac{EW_2 - EW_1}{\tan(\alpha_2) - \tan(\alpha_1)},$$

Equation 18 Equation for determining height of a trapezoidal structure from edgewidths and tilt angles.

$$\theta = \tan^{-1} \left(\frac{EW_2 - EW_1}{EW_1 \tan(\alpha_2) - EW_2 \tan(\alpha_1)} \right).$$

Equation 19 Equation for determining sidewall angle of a trapezoidal structure from edgewidths and tilt angles.

By employing the small angle approximation (i.e., $\tan \theta \approx \theta$), the equation for height can be reduced to

$$h \approx \frac{EW_2 - EW_1}{\alpha_2 - \alpha_1}.$$

Equation 20 Equation for determining height of a trapezoidal structure from edgewidths and tilt angles.

Supposing that $\alpha_2 - \alpha_1$ stays constant during measurement (as it does in these simulations), height measurement error emanates from the measurement error of EW_2 and EW_1 . This allows for a straightforward calculation of the propagation of errors [132] and the 3σ precision of the height ($3\sigma_h$) can then be estimated to give

$$\begin{aligned}
 F &= aA \pm bB \rightarrow \sigma_F^2 = a^2\sigma_A^2 + b^2\sigma_B^2 \pm 2ab\sigma_{AB} \rightarrow \sigma_F = \sqrt{a^2\sigma_A^2 + b^2\sigma_B^2 \pm 2ab\sigma_{AB}} \\
 h &= \frac{1}{\alpha_2 - \alpha_1}(EW_2 - EW_1) = X(EW_2 - EW_1) \rightarrow \sigma_h = \sqrt{X^2\sigma_{EW_2}^2 + X^2\sigma_{EW_1}^2 \pm 2XX\sigma_{EW_2EW_1}} \\
 \rightarrow 3\sigma_h &= \frac{\sqrt{(3\sigma_{EW_2})^2 + (3\sigma_{EW_1})^2 - 2\sigma_{EW_2EW_1}}}{\alpha_2 - \alpha_1} \rightarrow 3\sigma_h = \frac{\sqrt{(3\sigma_{EW_2})^2 + (3\sigma_{EW_1})^2}}{\alpha_2 - \alpha_1}.
 \end{aligned}$$

Equation 21 Three sigma precision ($3\sigma_h$) of the height measurement. In the formula, $3\sigma_{EW_2}$ and $3\sigma_{EW_1}$ are the 3σ edgewidth measurements for the tilt angle images. Note that the covariance term is included only if there is a reliable estimate otherwise it can be removed as in the final form of the equation.

From Equation 21, the precision of the height is positively related to the precision of the edgewidth measurement of each tilt and it is inversely proportional to the difference between the two angles of imaging. The greater the angle difference, the better the precision but as shown earlier there is a hard limit on the angle difference achievable based on the aspect ratios of the target structures.

For certain processes the height may be assumed to be constant and the only important parameter that need extraction may be the sidewall angle. In this situation, it is more common to use the re-entrant trapezoid (top surface larger than the bottom) technique presented in Figure 3.25. Here, a trapezoidal profile is assumed for the structure, angles are beam tilt (α)

and the desired SWA ($= 90^\circ + \theta$). Height is h and measured edgewidth is EW . With this, the geometrical relationship can be deduced to give

$$\frac{\sin(\alpha - \theta)}{\cos(\alpha) * \cos(\theta)} = \frac{EW}{h} \rightarrow \theta = \tan^{-1} \left(\tan(\alpha) - \frac{EW}{h} \right).$$

Equation 22 Derived solution for SWA measurement from a re-entrant trapezoidal model.

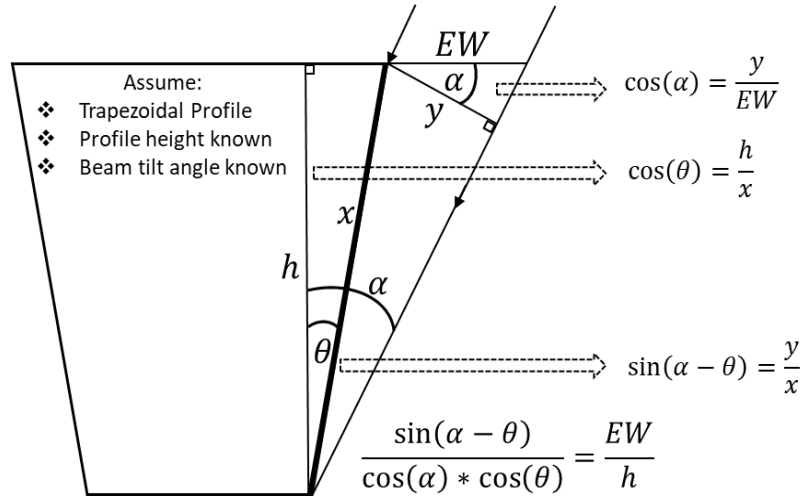


Figure 3.25 Diagram of re-entrant trapezoidal model with corresponding geometrical relationships.

Plotting out Equation 22 in Figure 3.26 shows the linear response of the SWA to the ratio of edgewidth to height for a range of tilt angles (0° to 15°). Also, from the gradients of these lines a sensitivity determination of the SWA measurement can be made. Figure 3.27 shows the number of nanometers of edgewidth per 1° of SWA for a series of heights and tilt angles.

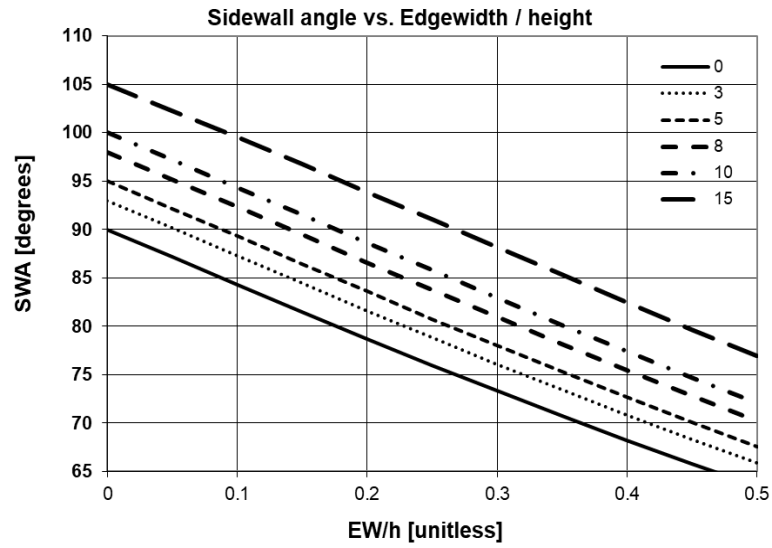


Figure 3.26 Dependence of sidewall angle on the edgewidth to height ratio.

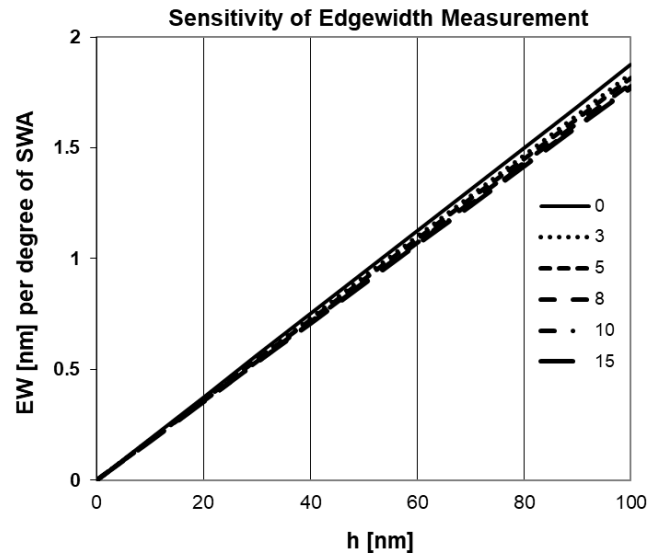


Figure 3.27 Sensitivity of the edgewidth to feature height and tilt angle.

4 RESULTS

4.1 DEFECT INSPECTION

The generation of virtual images is validated by replication of some of the features present in the SUNY Poly SEMATECH IDA wafer and comparing the output with experimental images. The IDA wafer consisted of 30-nm pitch grating of fins (width = 15 nm; height = 35 nm; link length = 300 nm) printed using direct-write electron beam lithography with hydrogen silsesquioxane (HSQ) electron beam resist, which effectively leaves SiO_2 for the pattern. A variety of programmed defects are present at different fractions of the design rule, but bridge defects were chosen for validation study. Real imaging data was collected with the 61-parallel beam Zeiss MultiSEM-505. A landing energy of 1.5 keV was chosen with a combination of dwell time and probe current that yielded a dose of 1,000 electrons per pixel. The FWHM probe diameter was matched to a pixel size of 2 nm. The virtual images, which were constructed following the protocols described above, along with the experimental image are shown below (Figure 4.1 and Figure 4.2). The normalized intensity profiles across the bridge

defect and across several of the fins are shown on the right of the figures for comparison.

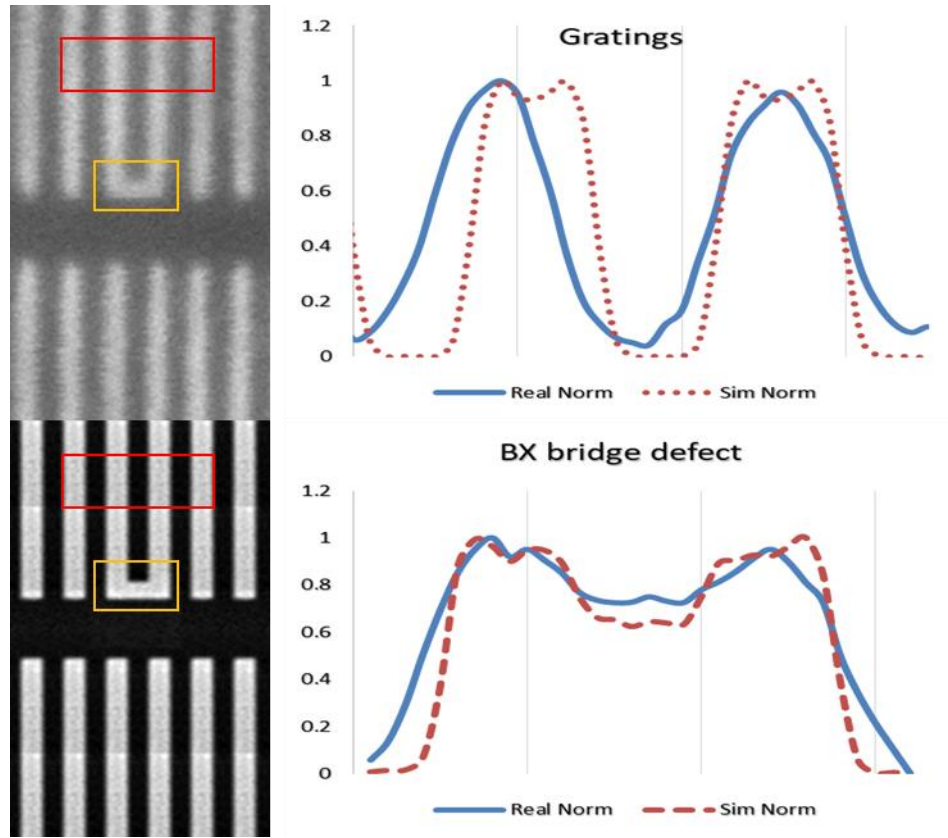


Figure 4.1 Left – Comparison between experimental (top) and simulated (bottom) images of X-bridge defect on 30-nm pitch IDA pattern. Both images represent approximately the same dose as well as pixel size and probe diameter (FWHM). Right – Comparison of line intensity profiles across the grating (top) and bridge defect (bottom).

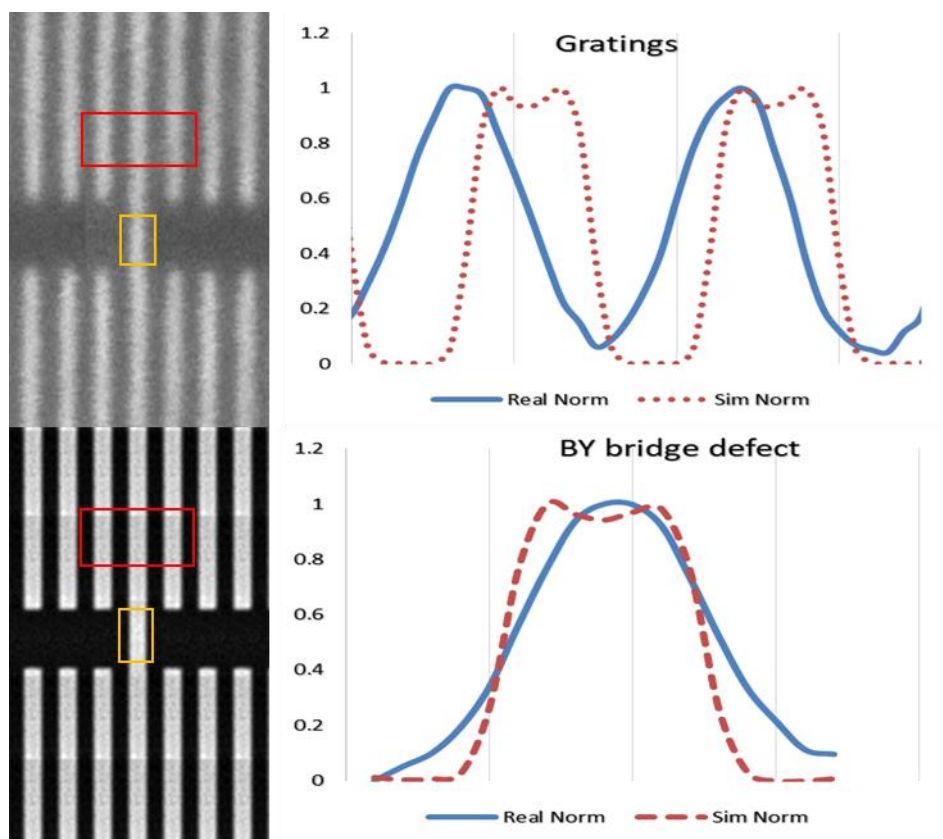


Figure 4.2 Left – Comparison between experimental (top) and simulated (bottom) images of Y-bridge defect on 30-nm pitch IDA pattern. Both images represent approximately the same dose as well as pixel size and probe diameter (FWHM). Right – Comparison of line intensity profiles across the grating (top) and bridge defect (bottom).

Overall, the simulated images are a good match for the experimental images. The remaining differences (e.g., profile plot offset of gratings) can be attributed to process variations: line edge roughness, sidewall angle, resist height loss, printing artifacts, and imaging noise considerations. Thus, we conclude that the simulation capabilities and strategies are sound, and the chief limitation is accurate reproduction of realistic specimen topography.

4.1.1 Virtual Experiment: Effect of Detector Dark Current on Defect Sensitivity

In specifying the performance of a defect inspection technology, a key managing principle is that throughput must be maximized while maintaining a minimum acceptable defect sensitivity. Consequently, the system would be operated as close to the minimum viable dose, shot-noise limit as possible. The presence of any additional noise in the system, such as dark current in the detector electronics, will compromise the defect sensitivity. Conversely, the contrast transfer function of the system will determine the sensitivity of the system to those variations. Simulation and prediction capabilities are powerful tools in addressing this issue. At this point, we simulate the effect of detector dark current on defect sensitivity by adding excess Gaussian noise to the shot-noise modulated virtual defect image and measuring the defect signature SNR. The process flow for the image generation and analysis is illustrated in Figure 4.3. Images were simulated where a variety of defects were added to an aggressively scaled 12 nm fin array (base pattern is 12 nm wide Si fins with vertical profiles of 20 nm height and 36 nm link length, and with 12 nm space to neighboring links in both x and y directions, on a Si substrate), using 500 eV for the landing energy and 3 nm pixels with probe diameter of 3 nm FWHM. The virtual image was generated and then modified with a transfer function mask in line with the performance of an SEM. The pixel intensities were then modulated with shot noise to emulate a dose of 210 electrons per pixel. Next, Gaussian noise was added with increasing rms amplitude. The defect signature SNR was then calculated for each image as a function of the noise rms.

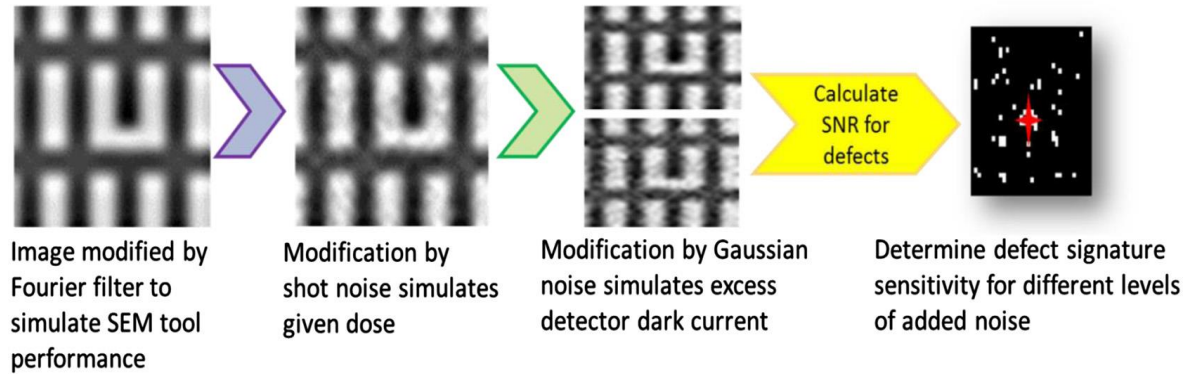


Figure 4.3 Virtual imaging processing flow for the defect sensitivity versus dark current virtual experiment.

Figure 4.4 below displays the comparison of 2, 4, and 8 nm X-bridge defects, along with small thumbnail images of the defective regions of interest with the various levels of added noise for each sized defect. Pattern and defect features are both oxide on silicon. At low levels of noise, the defect SNR is relatively insensitive to noise, but then the SNR drops rapidly beyond some critical dark current level. The final data point in a curve indicates the point where the defect is last detected, i.e., giving a nonzero defect SNR. As higher noise levels are added, the noise masks the defect sufficiently enough that it does not pass the threshold parameters set previously, therefore giving no defect SNR since the defect will be undetectable.

All together, the results take the expected form; larger defects will remain detectable against larger amounts of noise. Even 2 nm X-bridge defects should be able to be detected with an SNR of ~ 6 with up to 7 rms grayscale of added Gaussian noise.

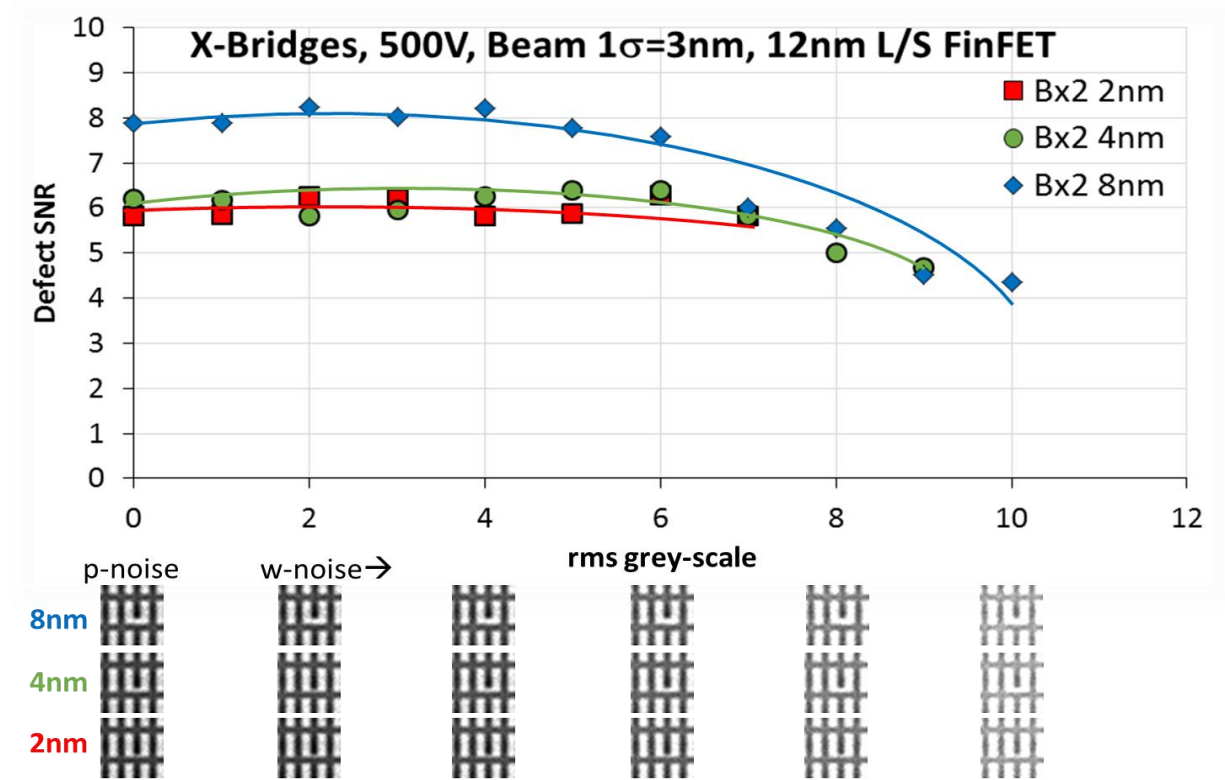


Figure 4.4 Defect SNR for X-bridge defects versus added white noise, resulting from X-bridge defect images of three different sizes being continually drenched with excess white noise. The small images underneath the graph show the defect region with only shot noise added on the left, and then with white noise added in steps of 2σ going to the right, for all three defect sizes as marked. Final data points in a curves show the point where the defect is last detected, i.e., giving a defect SNR. After that, the noise drowns out the defect enough that it does not pass the threshold parameters set previously therefore giving no defect SNR.

Lastly, the relative noise sensitivity for different types of defects was explored at a fixed defect size. Figure 4.5 shows the defect SNR as a function of rms grayscale noise for silicon and copper island defects, mid and end of fin X-bridges, and a mousebite (the mousebite used was somewhat smaller than the others, but the 2D nature of the defect produces a defect signature that is comparable in magnitude to the other defects). The copper defect exhibited the highest SNR but was also the most susceptible to degradation by detector noise. This follows along from unpatterned defect contrast results arising from a particulate defect (e.g., glassy carbon

(glC), poly-(methylmethacrylate) (PMMA), silicon, silicon dioxide, copper, and tungsten) on a silicon substrate [133].

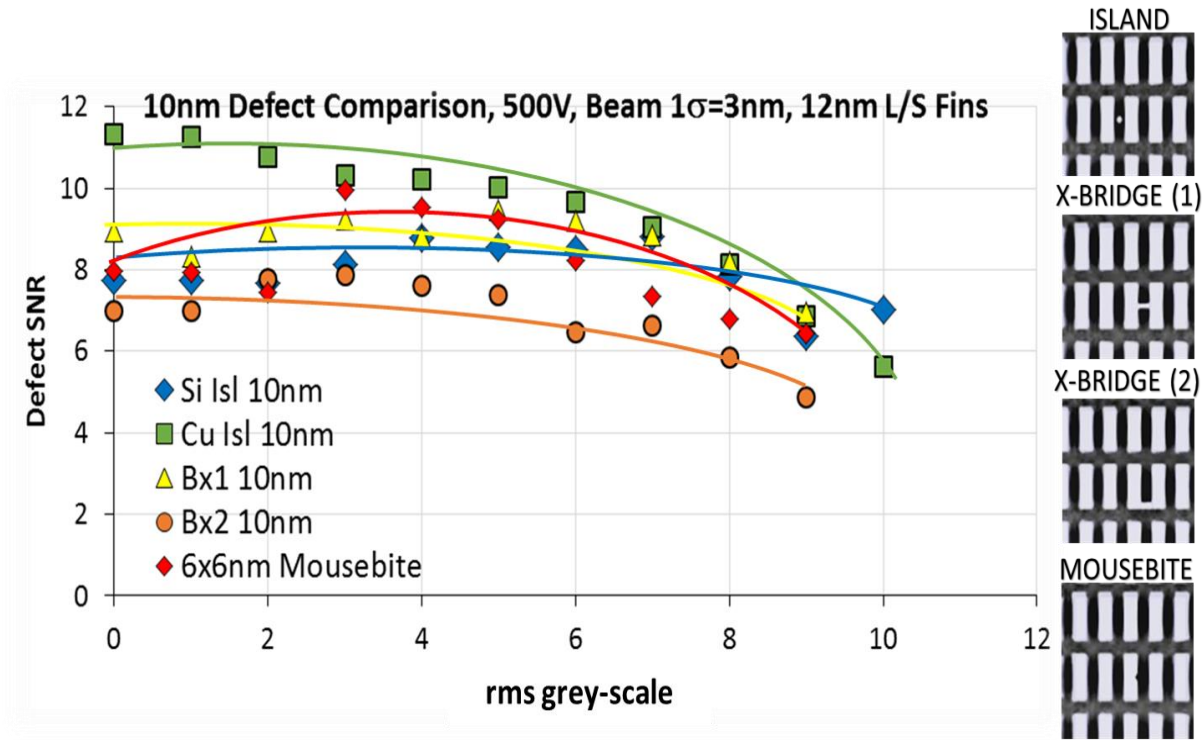


Figure 4.5 Defect SNR for various defect types of 10 nm size versus added white noise, resulting from the defect images being continually drenched with excess white noise. Defects included are silicon and copper islands, X-bridges with the bridge being aligned to either the center or end of a fin link, and also the case of a 6 nm square mousebite out of a Si fin link. Final data points in a curves show the point where the defect is last detected, i.e., giving a defect SNR. After that, the noise drowns out the defect enough that it does not pass the threshold parameters set previously therefore giving no defect SNR.

Overall, most defects appear to be relatively insensitive to noise but exhibit a fairly abrupt cutoff. Defect sensitivity analyses was also carried out using the 61-beam demonstration platform with two types of programmed defect samples [81], [85]. Below, the results of an indicative sample of this study are recounted.

4.1.1.1 SENSITIVITY ANALYSIS OF REAL SAMPLES (61-BEAM SYSTEM PERFORMANCE)

Initial imaging performance evaluations were conducted on multiple sample types including etched and resist patterned wafers and an extreme ultraviolet (EUV) mask. Both IDA and non-IDA patterns were imaged. Imaging conditions included a range of landing energies (0.5 – ~3keV), pixel sizes (~4nm, 10nm), and dwell times (50 ns – 400 ns).

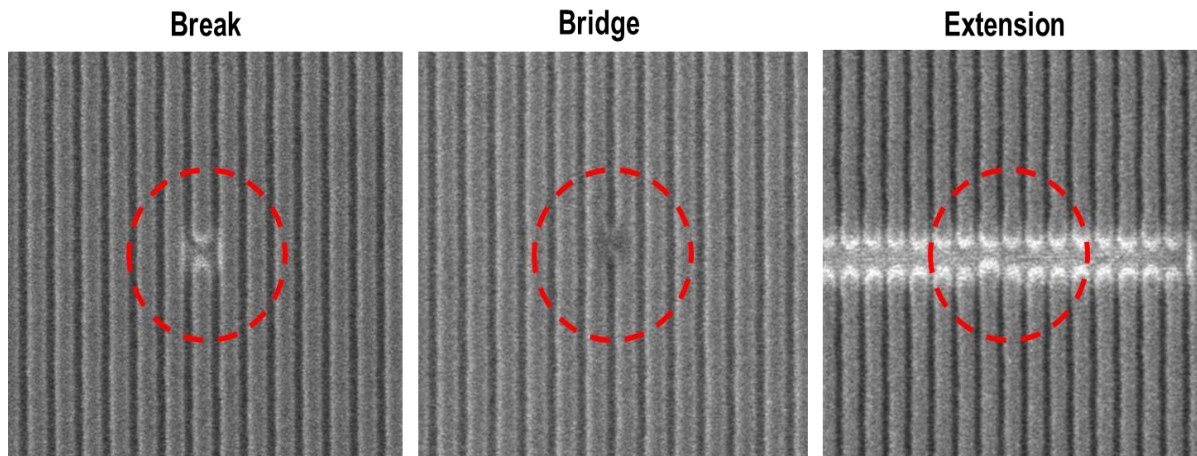


Figure 4.6 Images of intentional defects from a 28nm half-pitch wafer sample. The sample was imaged using a 3.76 nm pixel size, 50 ns dwell time and 1.5 keV landing energy.

Sensitivity analyses began by using the 61-beam demonstration platform with two types of programmed defect samples. The first sample was the 15 nm half-pitch line/space SUNY Poly SEMATECH IDA wafer containing patterns in HSQ resist on silicon. This was an early development sample from SEMATECH's next generation test vehicle development and fabrication project. The second sample contained 28 nm half-pitch line/space patterns etched in silicon and was part of the 28 nm design rule programmed defect array (PDA) provided by the International Business Machines Corporation (IBM). Each sample contained a wide variety of programmed defect types and sizes, including multiple sub-types. Initial results for three

general types of defects are shown below. Both were imaged using 50 ns dwell time and a 1.5 keV landing energy but the 15 nm wafer was imaged using 2 nm pixels whereas the 28 nm wafer was imaged using 3.76 nm pixel size.

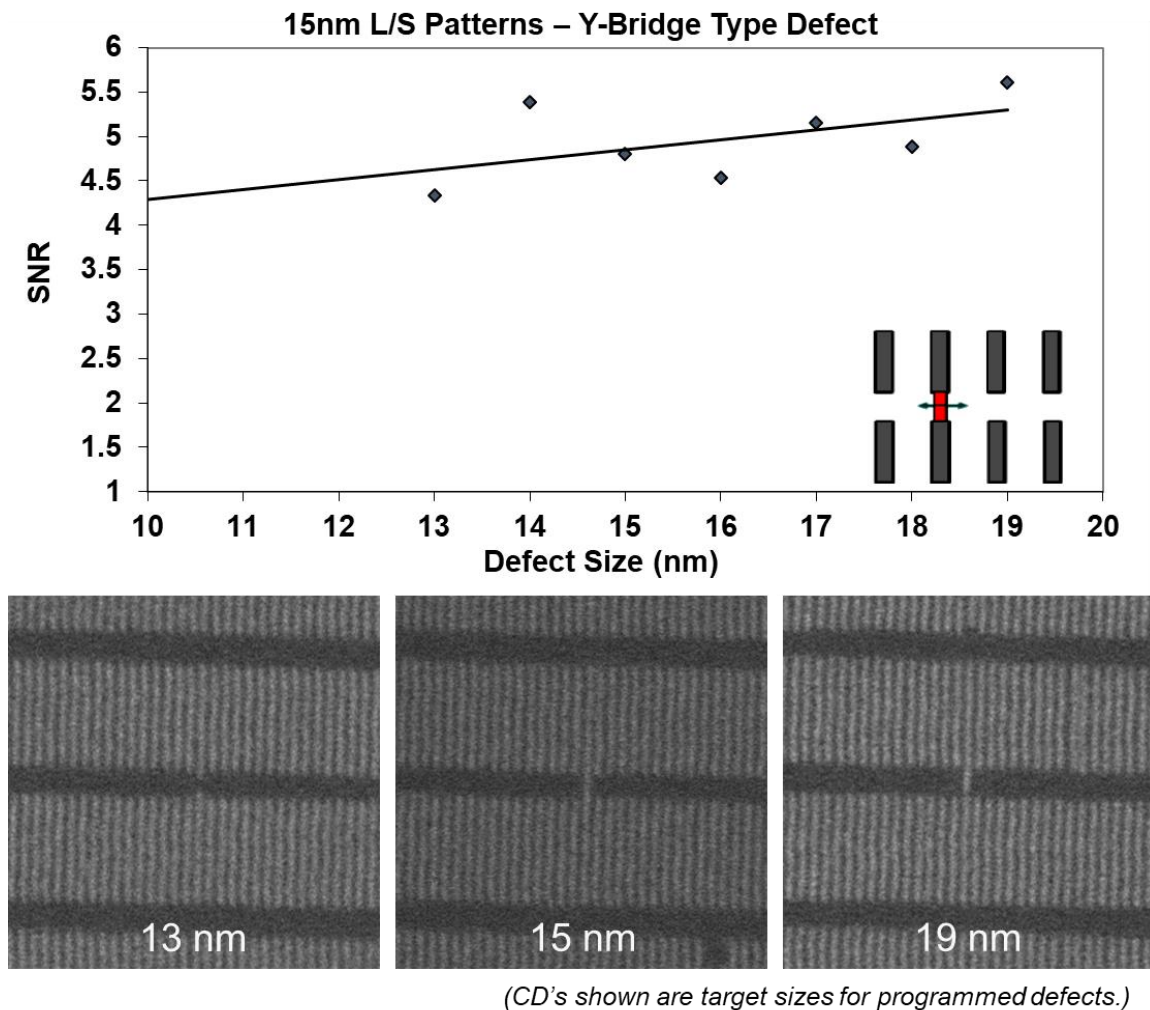
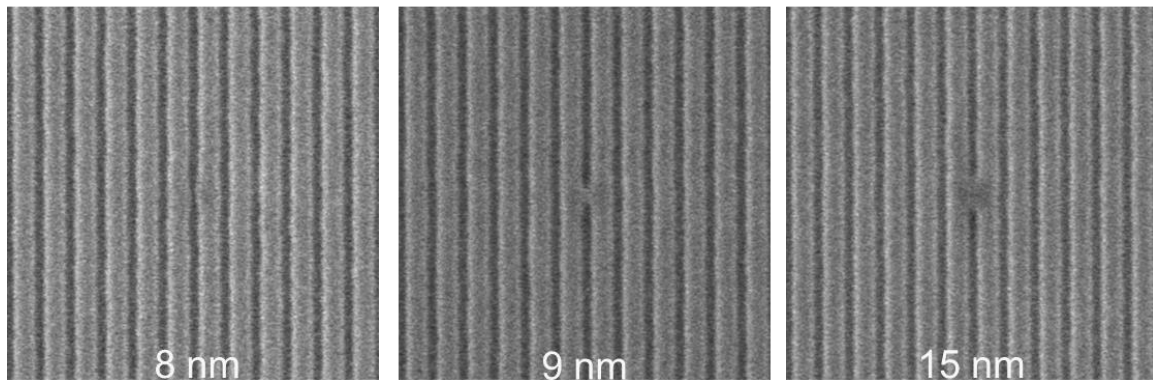
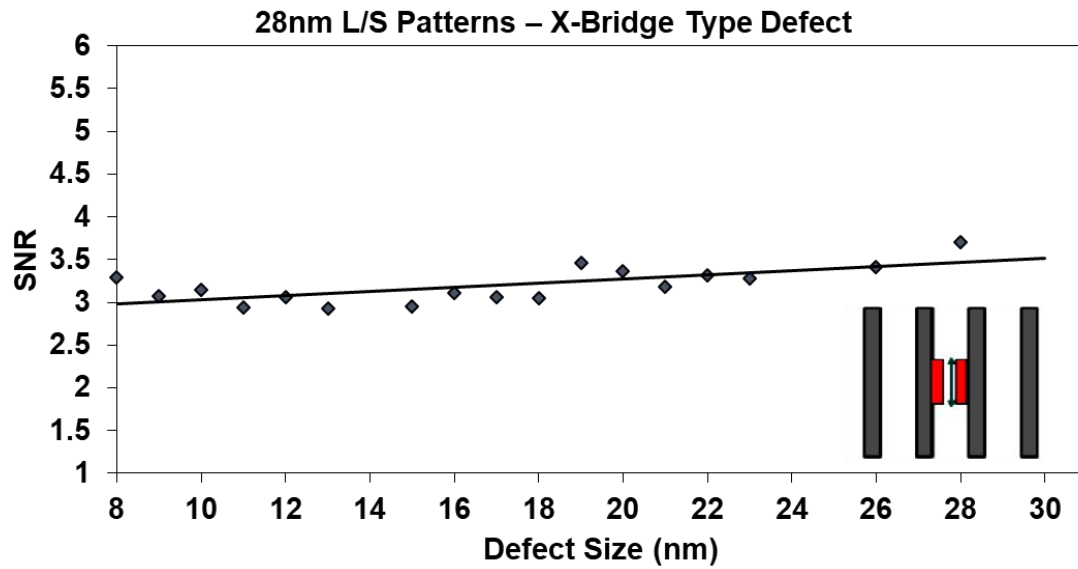


Figure 4.7 Top – SNR through programmed defect size for a bridge between two lines ending in the y-direction, with increasing x dimension in 15nm line/space HSQ patterns on silicon. Bottom – programmed defect example images.



(CD's shown are target sizes for programmed defects.)

Figure 4.8 Top – SNR through programmed defect size for a bridge between two line edges in the x-direction, with increasing y dimension in 28nm line/space etched silicon patterns. Bottom – programmed defect example images.

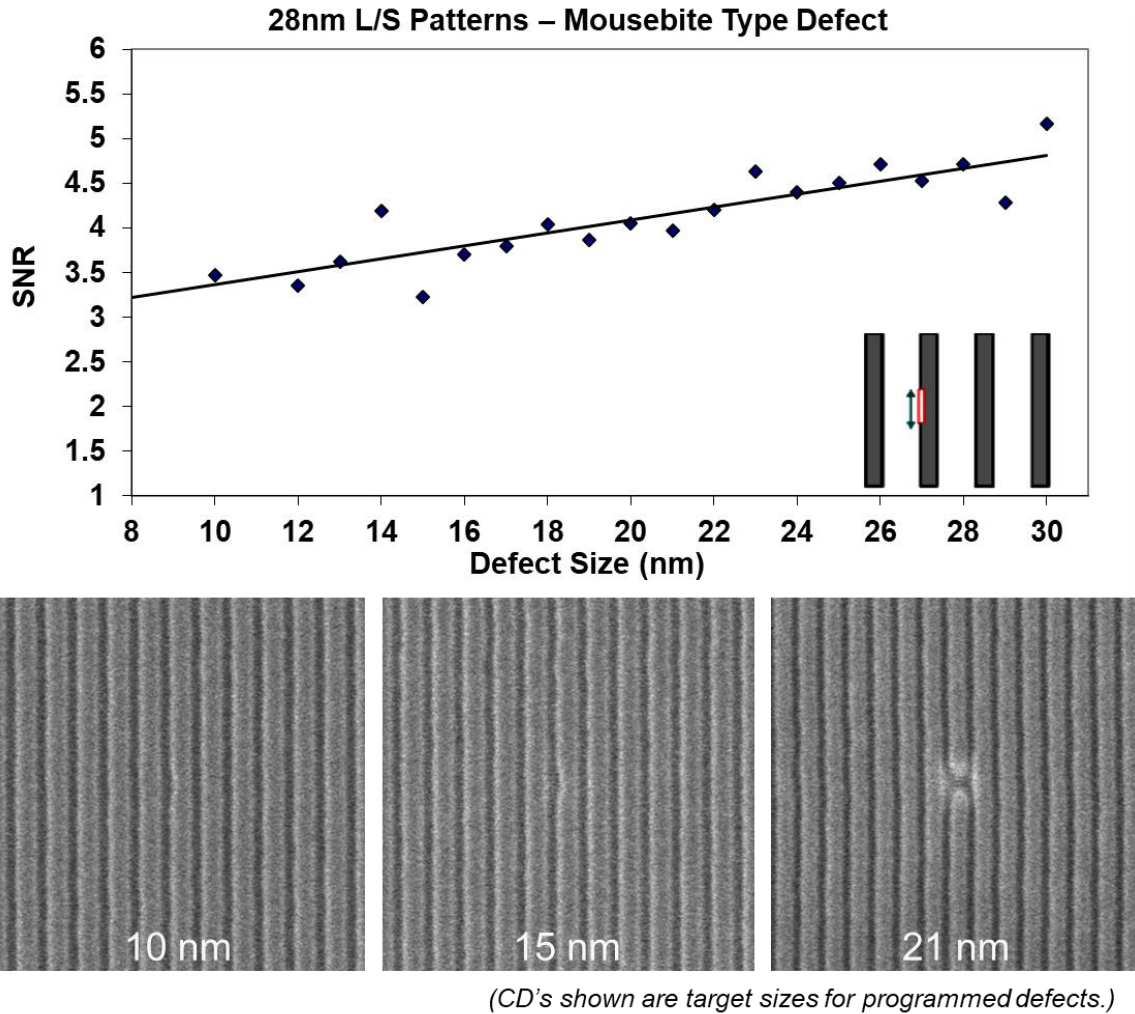


Figure 4.9 Top – SNR through programmed defect size for a “mousebite” in the side of a line, with increasing y dimension in 28nm line/space etched silicon patterns. Bottom – programmed defect example images.

The above figures show SNR versus defect size for the three different types of defects. As expected the SNR increases with defect size. These results demonstrate that the algorithms developed with NIST provide an adequate means for performance benchmarking in the early phases of the development program until a commercial defect inspection algorithm becomes available. The results also exhibit the impact of sample quality on SNR results. For example,

SNR outcomes from the lower quality 15nm line/space patterns result in considerably more variation than the higher quality 28nm line/space patterns.

Development of subsequent higher quality 15 & 18 nm IDA wafers in partnership with Cornell University displayed significantly better results. Below is a demonstrative sample of those results.

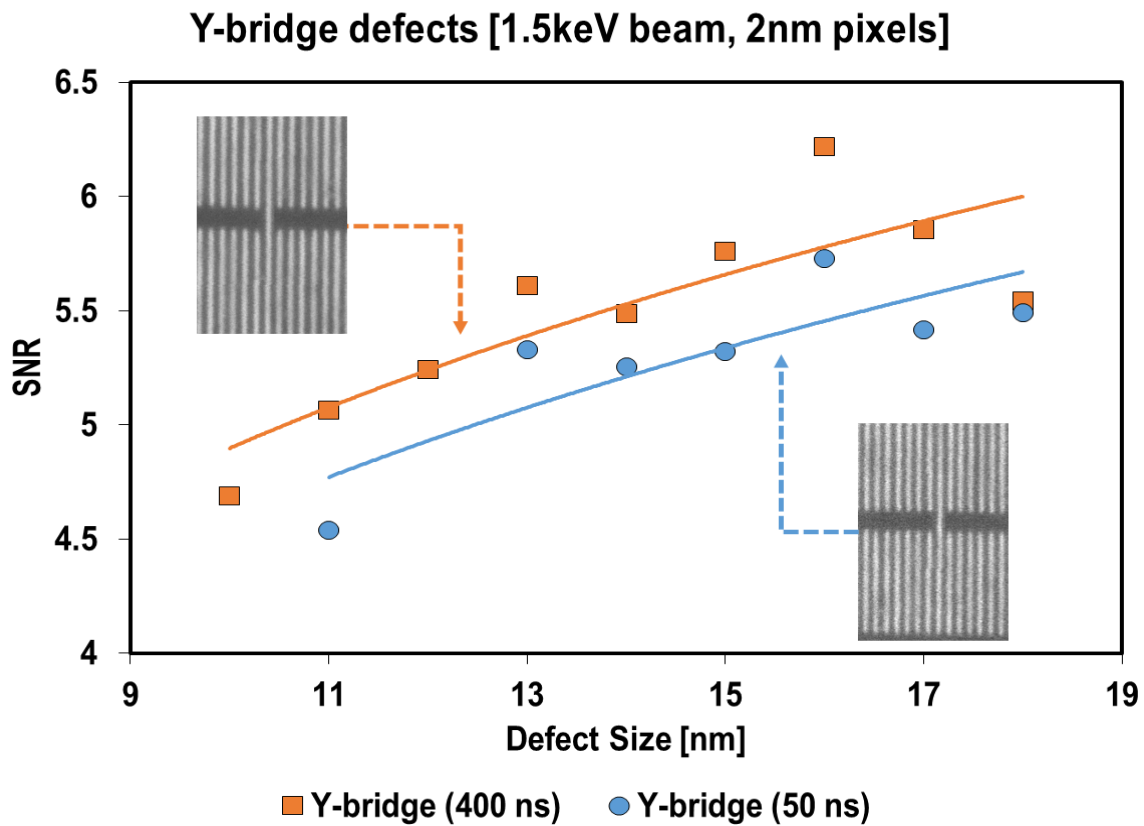


Figure 4.10 SNR through programmed defect size for a bridge between two lines ending in the y-direction, with increasing x dimension in 15nm line/space HSQ patterns on silicon. The top regression line represents the SNR trend for the sample imaged at 400 ns dwell time whereas the bottom line represents the sample at a 50 ns dwell time.

Again, as expected, the SNR increases with defect size in Figure 4.10. Also, results from the higher quality 15nm line/space patterns result in considerably higher SNR values.

Comparison of Y-bridges with different pixel dwell times does show that higher dose does lead to an expected higher SNR. In Figure 4.11, there is a comparison of different defect types varied through size.

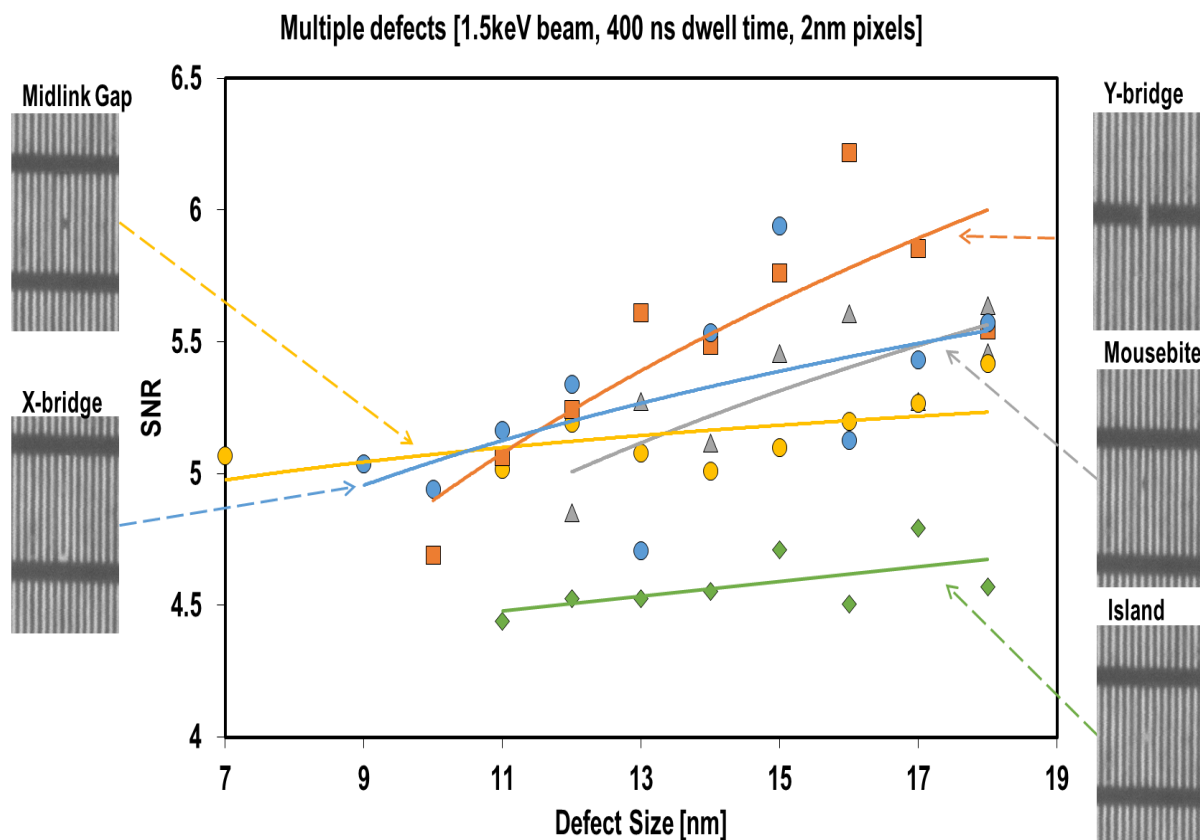


Figure 4.11 Sensitivity analysis of defects from real image data from the 61-beam tool with 2 nm pixels at 1.5 keV beam, imaging the 15 nm line/space IDA from SUNY Poly SEMATECH. Y-bridges result in the highest SNR when larger, whereas X-bridges show the greatest variance; mid-link gaps have consistent SNR (i.e., signature is about constant); mousebites have lowest SNR when small, and island defects are on the lower end of SNR overall.

As previously mentioned, in formulating multi-beam system requirements, it is vital to

understand the myriad of ways alterations to tool design and/or operating conditions influence throughput and defect sensitivity for a particular target application. In order to enable this objective, development of a vigorous suite of simulation protocols intended to model tool performance and to forecast system sensitivity to defects in various applications was developed. The SNR analysis used here offers a reliable method for assessing the impact of defect size, material contrast, image dose, tool performance, detector noise, etc. from samples imaged on the demonstration or other platforms.

4.2 CRITICAL DIMENSION MEASUREMENT

Validation of virtual images is carried out by the replication of features present in a GlobalFoundries patterned wafer and comparing the output with experimental images. The target structures are fins that are revealed post an etch process wherein there is oxide between the fin structures. The GlobalFoundries wafer consisted of several different patterned grating regions. The feature CDs of these fins are 15 nm with differing link lengths and pitches based on the particular region, but a gratings from the Malter region were chosen for validation study. The design in this region was 15 nm line/space of gratings 32 μm in the y-direction and $> 1\text{mm}$ in the x-direction. Height measurement of the gratings are confirmed by transmission electron microscope (TEM) imaging. Real imaging data was collected with the Zeiss LEO-1550, a high-performance Schottky field-emission SEM capable of resolution in 2-5 nm size range. For simulation, a landing energy was chosen with a combination of dwell time and probe current that yielded a dose consistent with that of real imaging i.e., 800 eV and 5,000 electrons per pixel. The FWHM probe diameter was matched to a pixel size of $\sim 0.8\text{ nm}$. The virtual image,

which was constructed following the protocols described above, along with the experimental images are shown below (Figure 4.12). The normalized intensity profiles across several of the fins are shown underneath the figures for comparison.

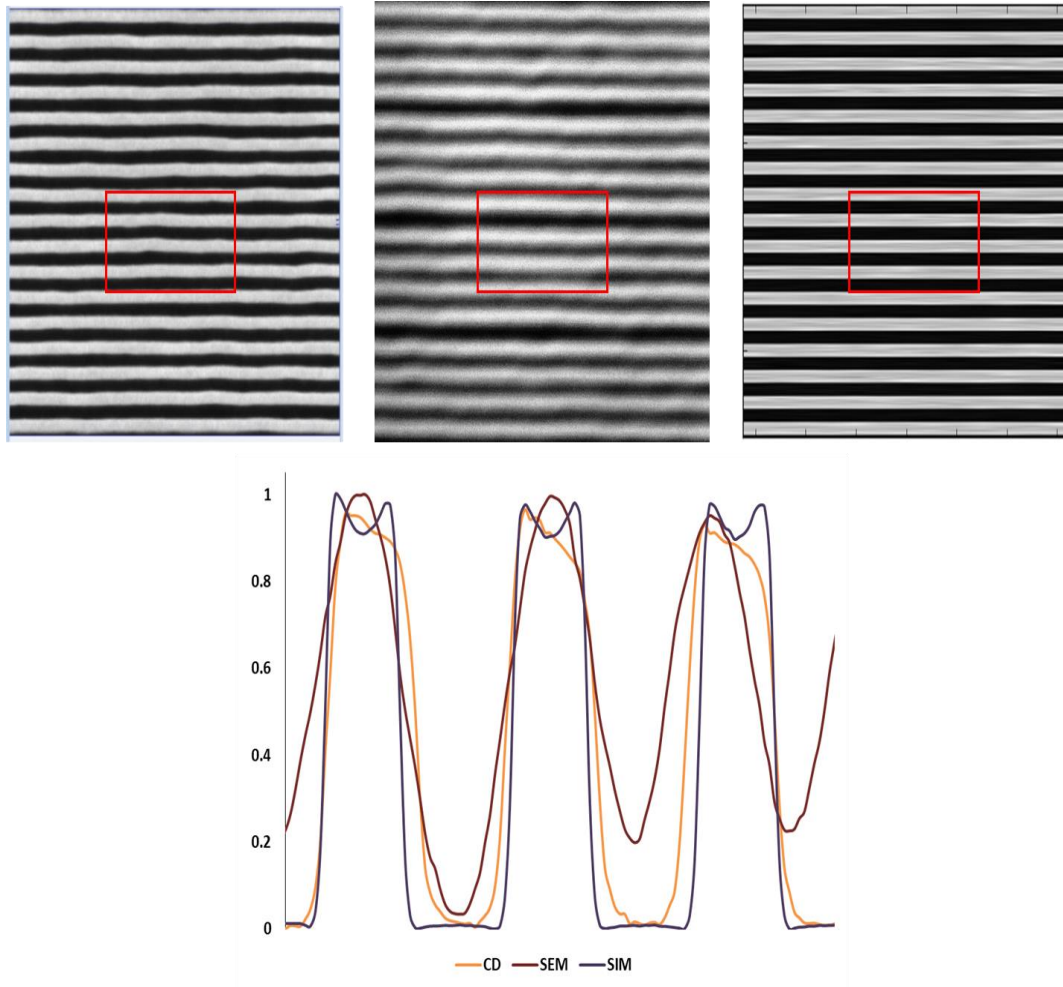


Figure 4.12 Top – Real (CD-SEM and SEM) and simulated images of a $0.5\ \mu\text{m} \times 0.5\ \mu\text{m}$ FOV of the Malter region, respectively. Bottom – normalized plot profiles of several gratings using the abovementioned methods.

A representative sampling of the Malter region was cleaved by focused ion beam (FIB) and imaged by TEM for comparison of height and sidewall angle measurement. The measurements obtained from TEM imaging are compared to those of tilted imaging using a conventional SEM (Zeiss LEO-1550) and simulations (Figure 4.14). A description of how edgewidths were

extracted from plot profiles is discussed in the following section.

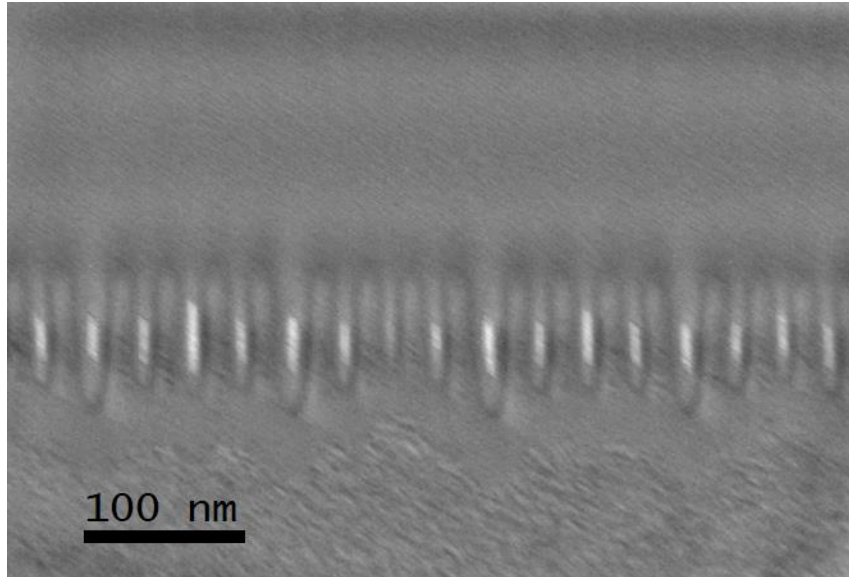


Figure 4.13 TEM image of a section of the Malter region.

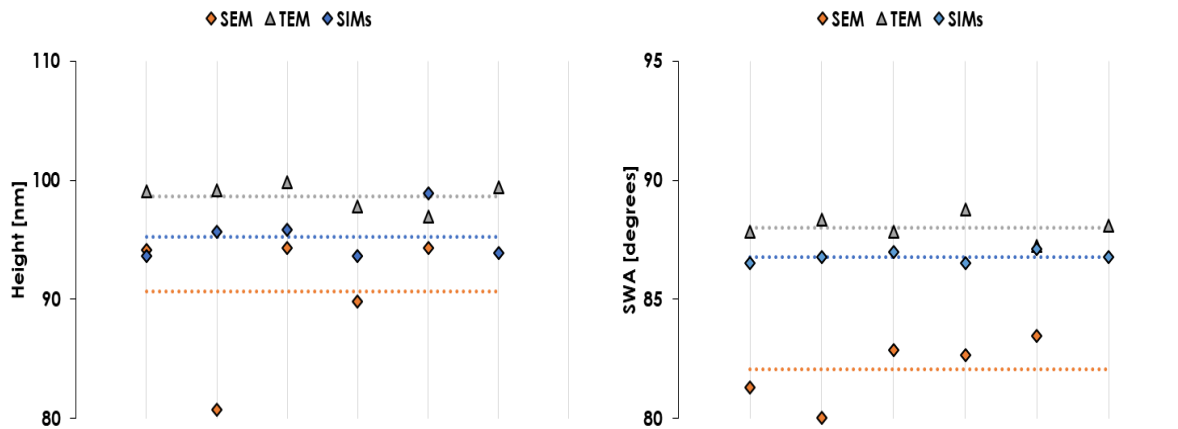


Figure 4.14 Left – sample height measurements of fin gratings with various metrology methods. Right – sample SWA measurement for those same gratings. Simulation and SEM tilt images were taken at 3 and 6 degree tilt angles.

Overall, the simulation images are consistent with the experimental measurements. The remaining differences (e.g., profile plot offset of gratings) can be attributed to process variations: line edge roughness, printing artifacts, and imaging noise considerations. Therefore,

we conclude that the simulation capabilities and strategies are sound, and the chief limitation is accurate reproduction of realistic specimen topography.

4.2.1 Virtual Experiment: Effect of Detector Dark Current on Measurement Sensitivity

At this point, we simulate the effect of detector dark current on measurement sensitivity by adding excess Gaussian noise to the shot-noise modulated virtual fin image and measuring the change in CD measurement. The process flow for the image generation and analysis is similar to the illustration in Figure 4.3 save the SNR calculation which is replaced with measurement of edgewidths and trigonometric calculations.

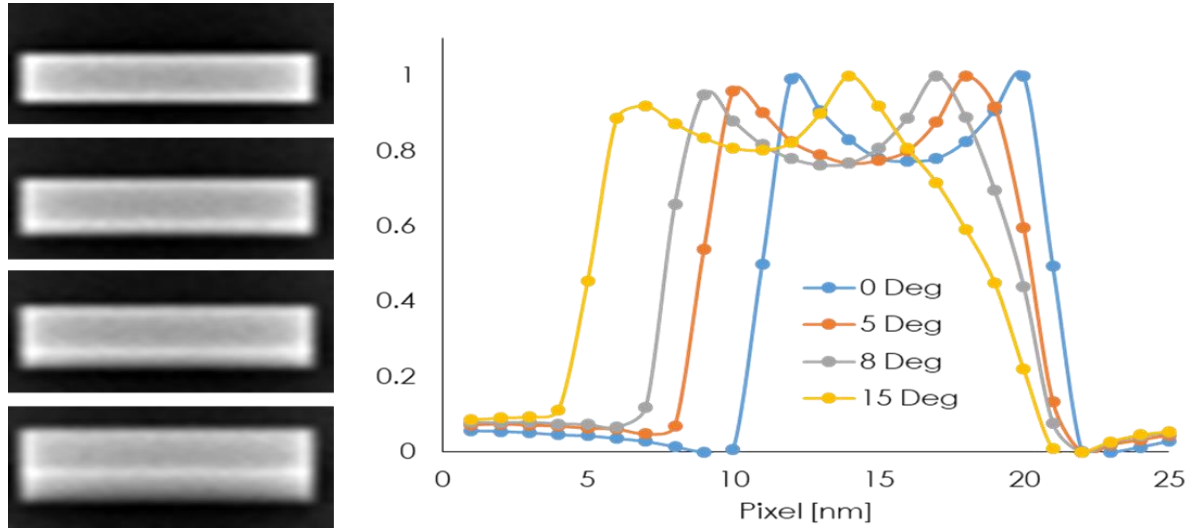


Figure 4.15 Normalized grayscale average profiles for a Si fin of height 25nm at four different angles i.e. 0, 5, 8, and 15 degrees.

In order to determine edgewidths from plot profiles (e.g., as in Figure 4.15) a standard method for determining base and top edges for the structures is implemented. Here, a basic thresholding scheme is established wherein the base of the figure is determined to be the intersection of the

minimum point and the regression of the flat profile of the substrate further away from the fin structure. The top is edge is determined to be the magnitude of the regression of the substrate taken from the top of the profile plot. This is illustrated in Figure 4.16.

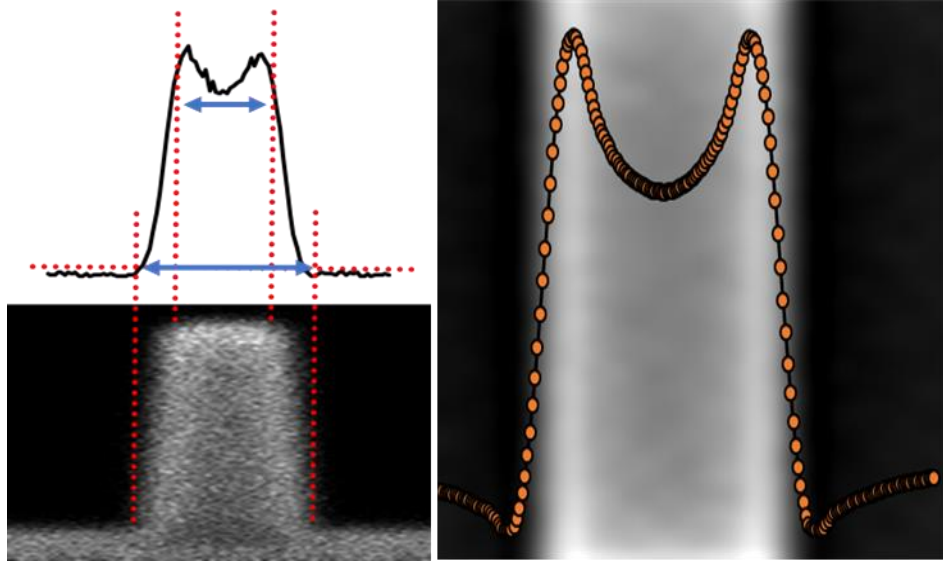


Figure 4.16 Secondary electron profile plot as related to a cross-sectional and top down view of simulated fin gratings, respectively.

Below, the expected edgewidth lengths is plotted along with the measurements taken using the abovementioned thresholding method for a variety of tilt angles for several differing heights at several different imaging pixel sizes. The final graph (bottom right of Figure 4.17) shows more clearly the effect of pixel size to measuring the edgewidth by plotting edgewidth measurements alongside expected edgewidths at differing heights for one tilt angle i.e. five degrees of tilt.

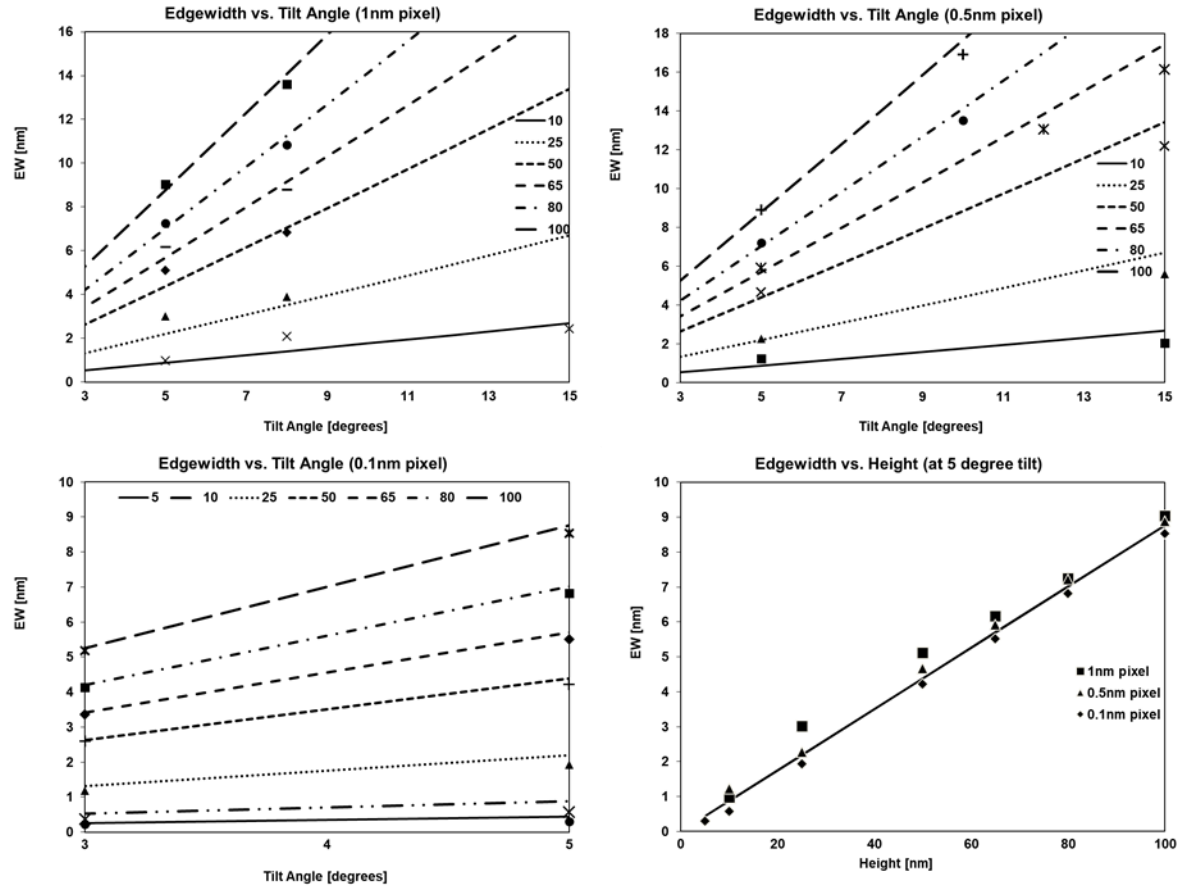


Figure 4.17 Width of projected edge versus tilt angle (in degrees) for various heights (in nanometers). Final plot (bottom right) shows the effect of pixel size on measurement of edgewidth versus height. Lines represent expected widths while the various points are measured edgewidths.

Subsequent to extracting the edgewidth from a plot profile the determination of the height of the structure can be made a few different ways. Taking the edgewidth measure of a structure from a single tilt angle e.g., 5 or 8 degrees and assuming that the sidewall angle of the feature to be the desired angle e.g., 90 degrees the measurement of height can be made. As well, using the measurements from each tilted angle along with the geometric relationships derived beforehand a height measurement can be calculated i.e., Equation 18.

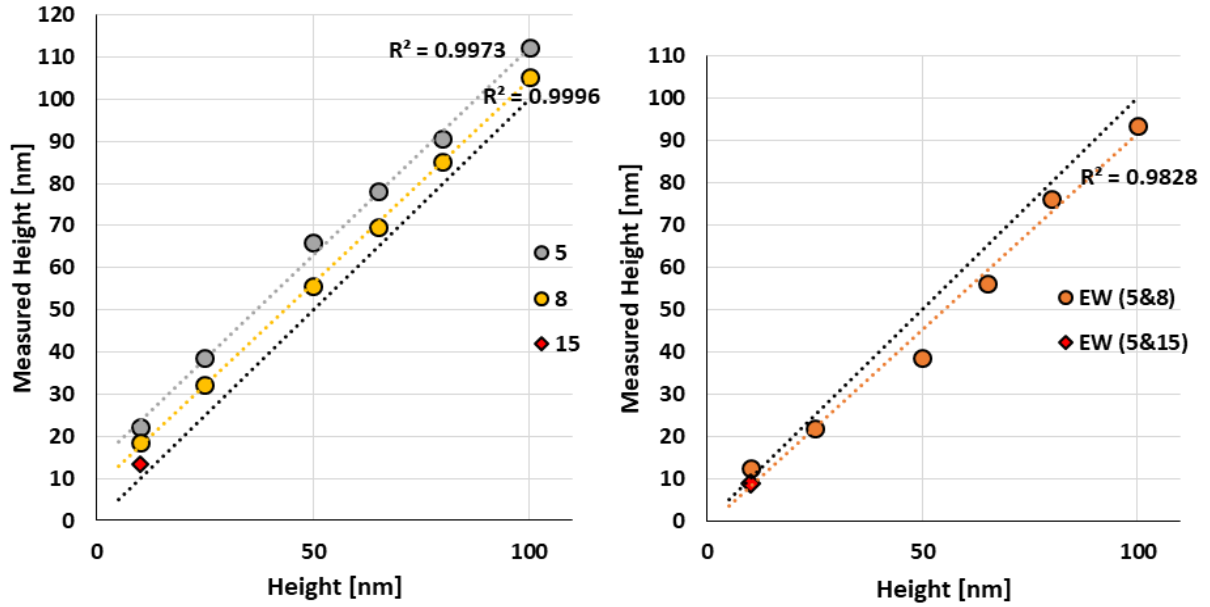


Figure 4.18 Left – measurement of height from single edgewidth from a single tilt angle i.e., 5, 8, or 15 degrees. Right – measurement of height using both the tilt angle measurements i.e., 5 & 8 and 5 & 15 degrees. Simulations implemented 10,000 electrons per pixel at 1 nm pixels for 10 nm width structures.

Both graphs above show linearity in the measurement of the height. Measurement of height from a single tilt image shows an over estimate of height whereas measurement from two angles shows a general underestimate. The extra point (i.e., edgewidth from 15 degrees and edgewidth from 5 & 15 degrees) on the graphs above are to emphasize that for the smaller sized structures extra tilting would be necessary to achieve an adequate measure of the edgewidth given the pixel size used.

Next, images were simulated where a variety of defects were added to an aggressively scaled 10 nm fin array (base pattern is 10 nm wide Si fins with differing vertical profiles from 100 nm height down to 5 nm and 36 nm link length, and with 20 nm space to neighboring links in

both x and y directions, on a Si substrate), using 500 eV for the landing energy and 0.1 nm pixels with probe diameter of ~.2355 nm FWHM. The virtual images were generated as earlier. The pixel intensities were then modulated with shot noise to emulate a dose of 1,000 electrons per pixel. Next, Gaussian noise was added with increasing rms amplitude. The height measurement was then calculated for each image as a function of the noise rms.

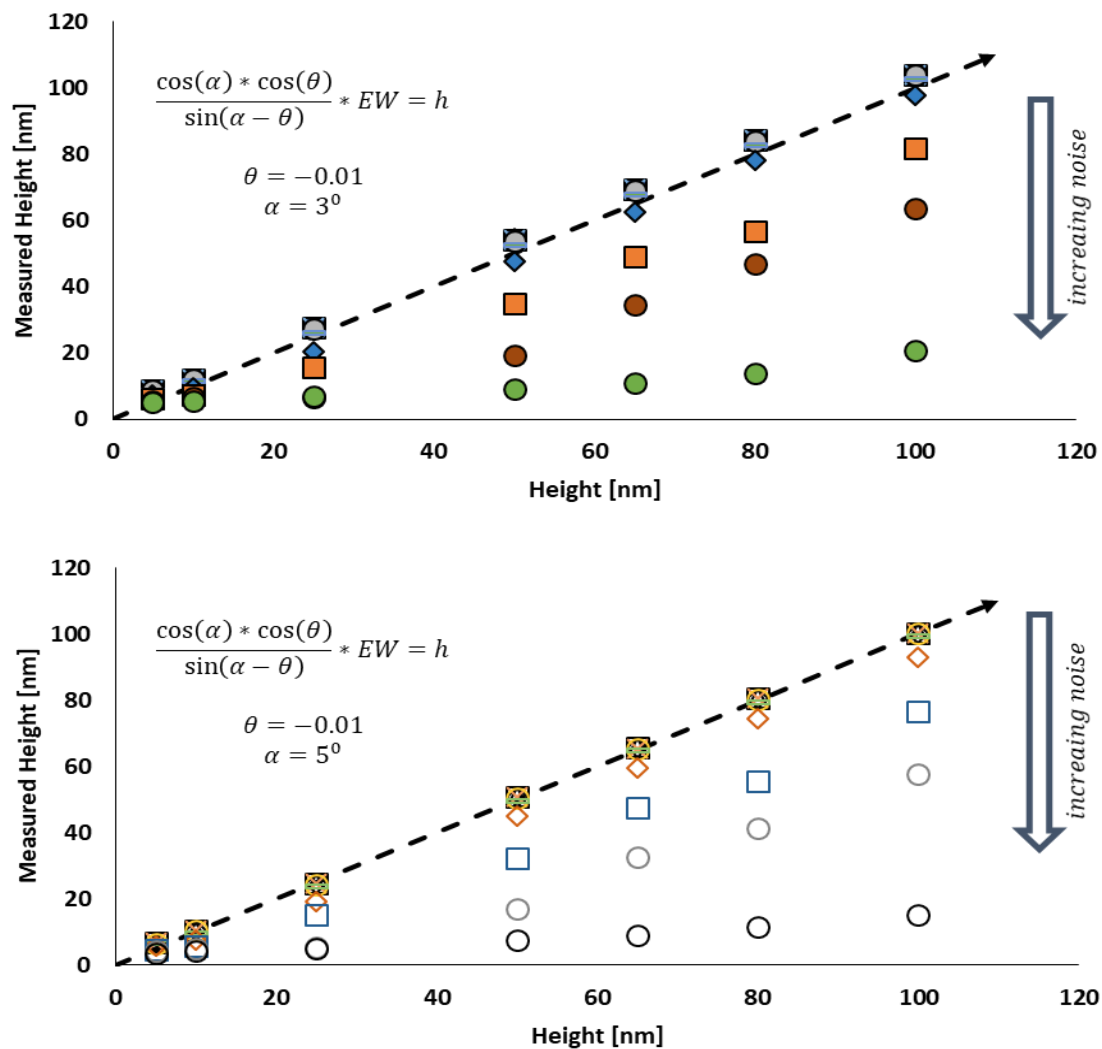


Figure 4.19 Measured height versus input model height for a range of noise additives for 3 and 5 degree tilts, respectively.

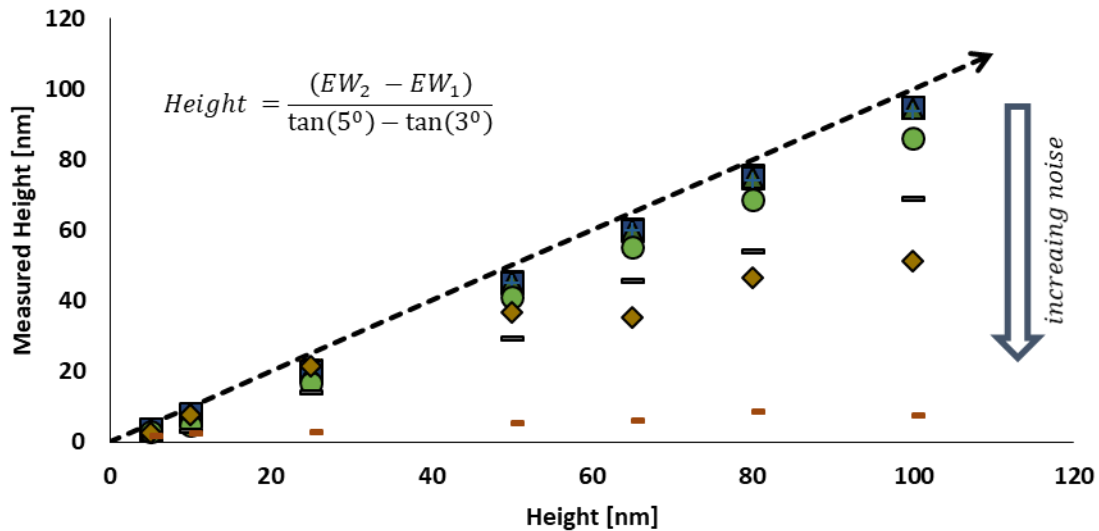


Figure 4.20 Measured height versus input model height for a range of noise additives using both 3 and 5 degree tilts.

Isolating one of these trends for a single height allows for a clearer viewing of the effect of noise on the ability of extracting feature height from the profile plot. Figure 4.21 shows this for the case of fin height equals 100 nm. As can be seen from the plot the height measurements gradually decline until they reach a critical point (around rms of >8) where measurement of the height is drastically effected by the additive noise. Figure 4.22 shows the plots for the rest of the heights, each isolated. Generally, most height features appear to be relatively insensitive to noise but exhibit a fairly sudden cutoff.

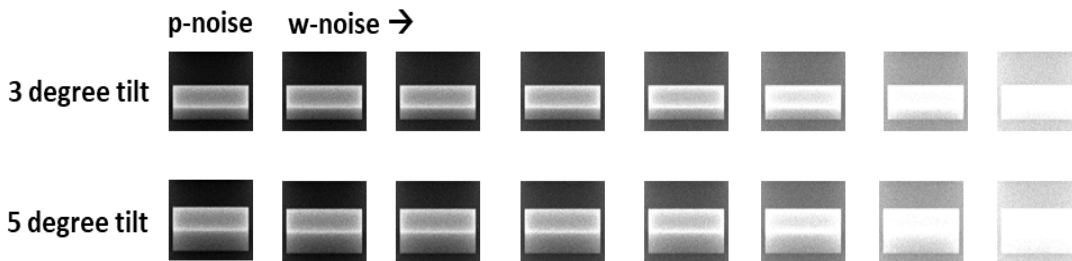
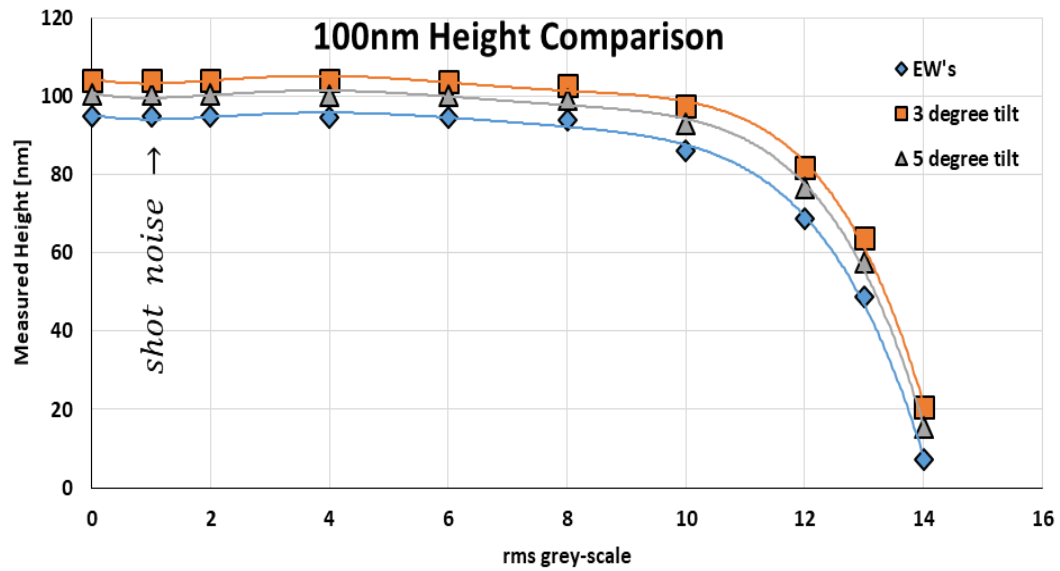


Figure 4.21 Height measurement for fins versus added white noise, resulting from fin images of two different tilts being continually drenched with excess white noise. The small images underneath the graph show the fin region with only shot noise added on the left, and then with white noise added in steps of 2σ going to the right.

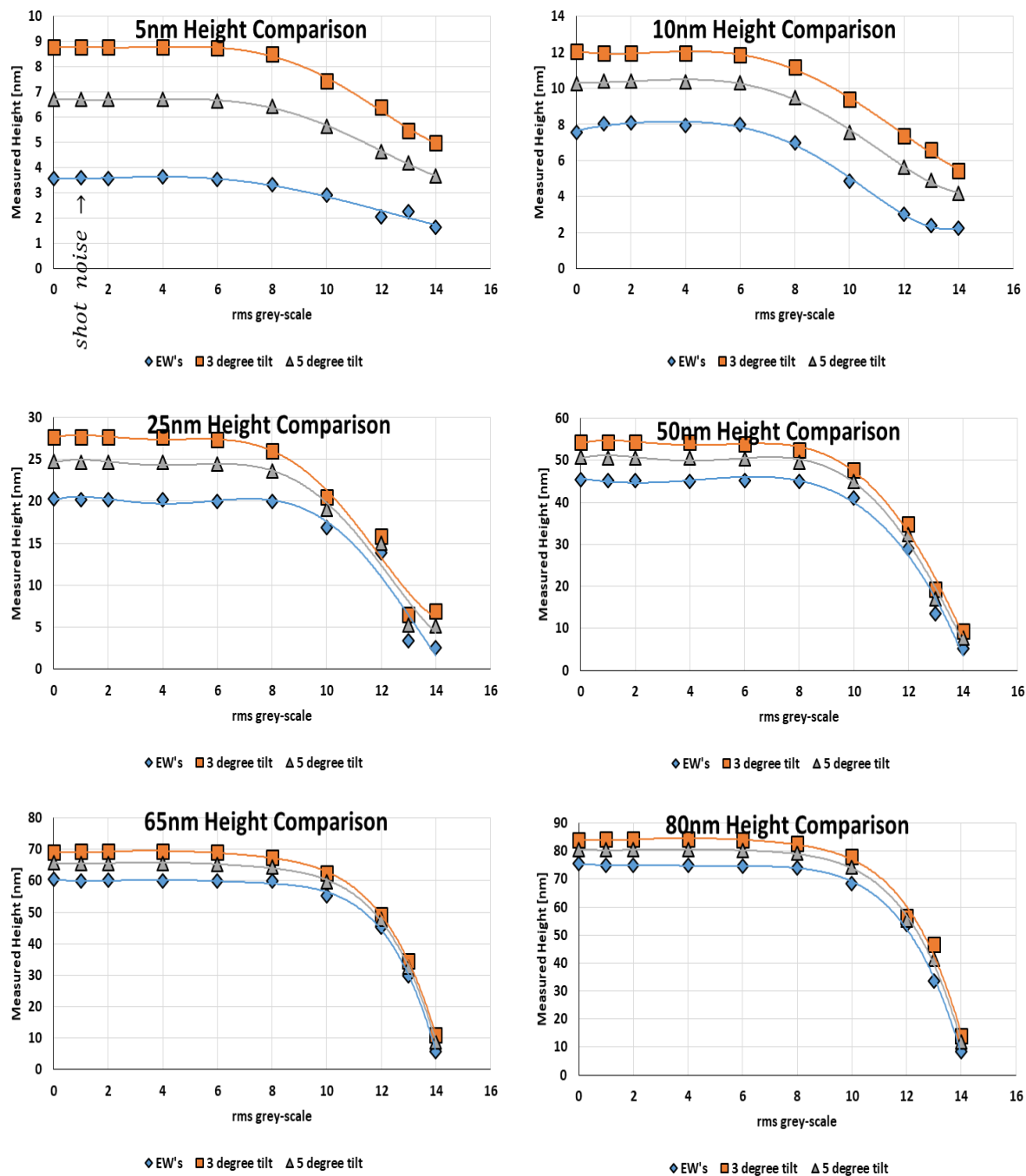


Figure 4.22 Height measurement for fins versus added white noise, resulting from fin images of two different tilts being continually drenched with excess white noise. The plots follow similar trends until the edgewidth reach sizes approximate to the interaction volume of the impinging electron beam.

Overall, the results take the same form as that of the SNR results of the simulated defects after noise considerations were implemented. For both cases, results (i.e., height measurement and SNR) were for the most part relatively stable up to 8 rms grayscale of added Gaussian noise after shot noise modulation, but an rms grayscale noise of 6 is more practical for operation. These results show that both the application of defect inspection and CD height measurement can be productively carried under very similar if not the same operating conditions.

In practice, the sidewall angle of fin structures are not direct ninety degree lines. Depending on the maturity of the fabrication process the sidewall angles can vary in major or minor ways from the desired design. Next, we explore the effect of a changing sidewall angle on the measurement of height of features. Simulation conditions are the same as those above i.e., 500 eV for the landing energy and 0.1 nm pixels with probe diameter of ~ 0.2355 nm FWHM.

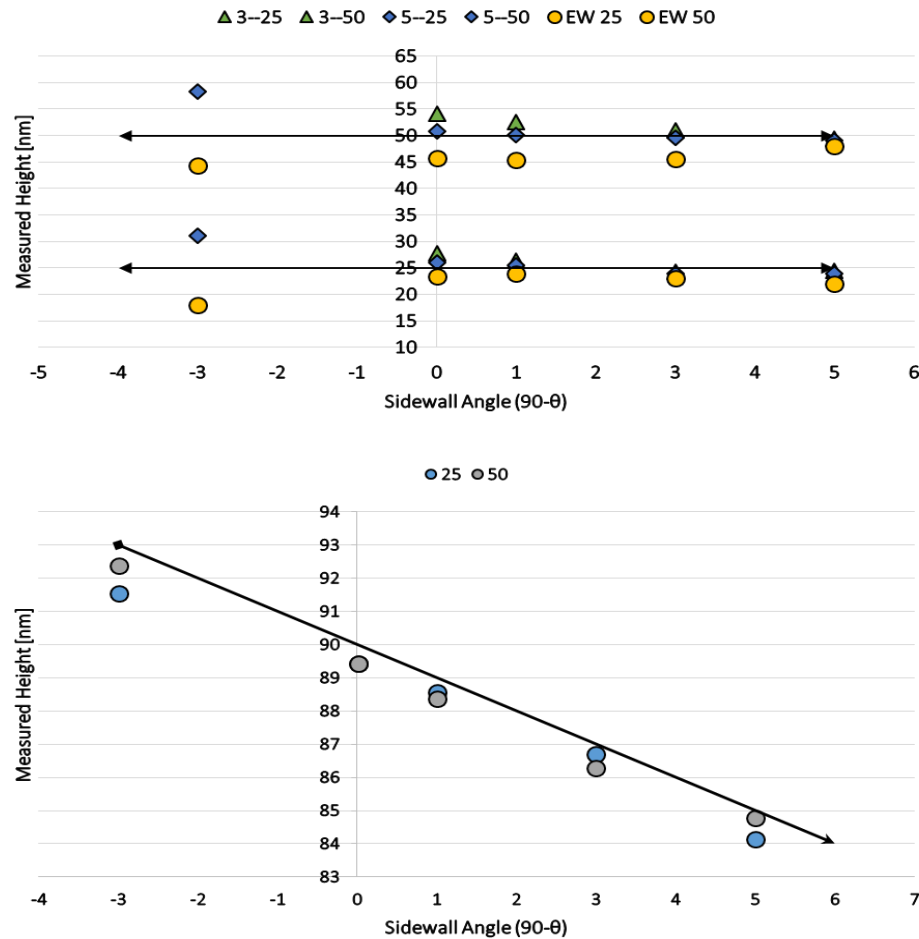


Figure 4.23 Top – measurement of height versus change in sidewall angle. Calculations were made using edgewidths from tilt angles of 3 & 5 degrees as well as for each individually (for the individual case assuming that the sidewall angle is the input value). Bottom – measured sidewall angle versus designed sidewall angle. Calculations were made using edgewidths from tilt angles of 3 & 5 degrees.

The changing sidewall angles do effect the measurement of height but not in any drastic fashion meaning that the height measurements can be reliable in a particular window of sidewall angle. Measurement of the sidewall angle itself is shown to be consistent until reaching an extrusion (SWA = 93). Here, as before, an even greater degree of tilting should resolve the issue dependent on the aspect ratios of the features in question.

This section continued the use of the aforementioned instituted infrastructure for generation of realistic virtual images of test structures for predicting the performance of electron beam defect inspection tools to further explore measurement of critical dimensions. Prospective next steps for this analysis would include continued refinement and advancement of measurement of non-planar structure height (mid, bottom, etc.) and sidewall angle along with exploration into Top Corner Rounding and Cross Correlation Effects to deduce a suitable metric for the deviation signatures of the varied critical dimension deviations. Thus, allowing for formulation of recommendations on viability of multi-beam critical dimension tools. The use of this methodology may also be supplemented and/or partially supplanted by the usage of proposed physics-based analytical linescan models as edge detectors and/or to extract CD measurements more generally. After calibration against a rigorous Monte Carlo simulator such as JMONSEL, for various materials involved in a sample, the linescan model can, with sufficient accuracy, forecast the linescans of those materials for a variety of geometric shapes [134]–[136]. Such models may provide much in terms of minimizing time and/or computational hardware restraints for model based prediction of critical dimension values.

5 CONCLUSIONS

5.1 SUMMARY

At this juncture, the necessary framework for generating realistic virtual images of test structures for forecasting the performance of electron beam defect inspection tools has been ascertained. This procedure for generation of virtual images is able to encompass a wide assortment of device structures, materials, design rules, and allows for incorporation of a variety of defect types. Furthermore, the effects of imaging conditions and system performance can also be incorporated, permitting the construction of especially realistic virtual images. In so having exhibited the capability to reproduce experimental images, this toolkit can therefore be used to forecast the effect of causing an assortment of variations to the operating conditions and/or system performance.

The same set of tools applied for the virtual exploration of defect inspection were consequently applied further to resolve the current critical dimension measurement issues with respect to the

non-planar device architectures that dominate production goals in industry. This methodology allows for the robust forecast of critical dimensions measurement with the incorporation of vital imaging condition and system performance effects. Overall, these resources provide valuable tools for predicting the system performance in these advanced application spaces.

The results produced for expected multi-electron beam operating parameters in this study have shown that multi-beam system approaches are promising for the use cases reviewed above. Although these use cases are sufficient for further exploration of multi-electron beam approaches to process metrology, the technology setups have much to improve upon. As of yet, these setups lack the capability of providing spectroscopic and other information that have become accustomed provision from standard scanning electron microscopy systems. Not to mention, these systems (i.e., MEMS based multicolumn and beam splitting) have yet to actualize the exact number of beams that would be required for high volume fabrication environments. As an example, for the multicolumn based systems, these engineering advancements would require further exploration into preserving the very low pressures necessitated by the field-emission sources, managing the heat generated by the filaments of the field-emission sources, properly directing large numbers of high voltage connections, etc. [33].

5.2 FUTURE DIRECTIONS

Having demonstrated the ability to reproduce experimental images, these tools can now be used to forecast the influence of rendering various changes to the operating conditions and/or system performance. For prospective use cases, because the virtual data generated in this study

was produced in a stepwise approach based on assumed benchmarks at every step, practical applications of this methodology would entail a more detailed description of the fundamental statistics involved in the formation of images by the utilization of impinging electron beams. This would include rigorous understanding of the statistical distribution of the point spread function (PSF) of the impinging electron beam, the generated secondary/backscattered electrons by the primary electrons, how this would then relay to the modulation of signal by the detector and signal detection more generally, further detailed understanding of the noise considerations emanating from the electronics of the tool, etc. To reiterate, the methodology here was executed so that any step along the process can and should be further delved into given the particular necessities of the objective at hand.

Also for future consideration, the implementation of electron emission calculation software that is more adept for high performance parallel computing than the one employed in this study would allow considerable progress in terms of minimizing time for creation of robust large scale physics-based virtual images and model based prediction more generally. These images may then be exported either to further developed in-house software for the analysis of defect signatures or extraction of critical dimension parameters. The images may also be exported to readily available commercial defect inspection or critical dimension measurement algorithms. The methodology employed here can further be advanced to investigate overlay measurements of device patterns (Appendix: B) and other use cases not yet considered. Continuing development of these resources will deliver effective tools for forecasting the system performance in advanced application spaces, where production of test structures may be difficult or impossible.

REFERENCES

- [1] M. Mukhtar and U. Pillai, “Nanomanufacturing: Application of nanotechnology in manufacturing industries,” *Nanotechnol. Law Bus.*, vol. 12, no. 1, pp. 5–18, 2015.
- [2] P.-E. Hansen and G. Roebben, Eds., *Introductory Guide to Nanometrology*. Seventh Framework Programme, 2010.
- [3] M. Keefer, R. Pinto, C. Dennison, and J. Turlo, “The Role Of Metrology And Inspection In Semiconductor Processing,” in *Handbook of Thin Film Deposition Processes and Techniques*, Elsevier, 2001, pp. 241–286.
- [4] P.-F. Huang, Y. Uritsky S., and C. R. Brundle, *Handbook of Silicon Semiconductor Metrology*, 1st ed. New York & Basel, 2001.
- [5] B. Thiel, M. Lercel, B. Bunday, and M. Malloy, “Assessing the viability of multi-electron beam wafer inspection for sub-20nm defects,” in *Proc. SPIE 9236, Scanning Microscopies*, 2014, vol. 9236, p. 92360E.
- [6] B. S. Doyle *et al.*, “High performance fully-depleted tri-gate CMOS transistors,” *IEEE Electron Device Lett.*, vol. 24, no. 4, pp. 263–265, Apr. 2003.
- [7] B. Bunday, “HVM metrology challenges towards the 5nm node,” in *Proc. SPIE 9778, Metrology, Inspection, and Process Control for Microlithography XXX*, 2016, vol. 9778, p. 97780E.
- [8] B. Bunday, E. Solecky, A. Vaid, A. F. Bello, and X. Dai, “Metrology capabilities and needs for 7nm and 5nm logic nodes,” in *Proc. SPIE 10145, Metrology, Inspection, and Process Control for Microlithography XXXI*, 2017, vol. 10145, p. 101450G.
- [9] A. Brand, “Precision Materials to Meet FinFET Scaling Challenges Beyond 14nm,” in *Leveraging Nonplanar Transistor Architectures and New Materials to Power Mobility Apps Beyond 20nm (Semicon West)*, 2013.
- [10] B. Colombeau *et al.*, “Advanced CMOS devices: Challenges and implant solutions,” *Phys. status solidi*, vol. 211, no. 1, pp. 101–108, Jan. 2014.

- [11] A. Vaid *et al.*, “A Holistic Metrology Approach: Hybrid Metrology Utilizing Scatterometry, CD-AFM and CD-SEM,” in *Proc. SPIE 7971, Metrology, Inspection, and Process Control for Microlithography XXV*, 2011, vol. 7971, pp. 1–20.
- [12] M. LaPedus, “Measuring FinFETs Will Get Harder,” *Semiconductor Engineering*, 2015. [Online]. Available: <http://semiengineering.com/measuring-finfets-will-get-harder/>. [Accessed: 14-Feb-2017].
- [13] J. Goldstein *et al.*, *Scanning Electron Microscopy and X-ray Microanalysis*, 3rd ed. New York: Kluwer Academic Publishers, 2003.
- [14] T. W. O’Keeffe and R. M. Handy, “Resistless fabrication of integrated circuits,” in *1967 International Electron Devices Meeting*, 1967, pp. 148–148.
- [15] T. W. O’Keeffe, J. Vine, and R. M. Handy, “An electron imaging system for the fabrication of integrated circuits,” *Solid. State. Electron.*, vol. 12, no. 11, pp. 841–848, Nov. 1969.
- [16] Y. Zhang, “A 100-electron-beam source from a high brightness Schottky emitter for fast patterning applications,” Delft University of Technology, 2008.
- [17] T. H. P. Chang, “Arrayed miniature electron beam columns for high throughput sub-100 nm lithography,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 10, no. 6, p. 2743, Nov. 1992.
- [18] T. R. Groves, “Distributed, multiple variable shaped electron beam column for high throughput maskless lithography,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 16, no. 6, p. 3168, Nov. 1998.
- [19] L. P. Muray *et al.*, “Advances in arrayed microcolumn lithography,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 18, no. 6, p. 3099, 2000.
- [20] D. S. Pickard *et al.*, “Distributed axis electron-beam system for lithography and inspection—preliminary experimental results,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 20, no. 6, p. 2662, 2002.
- [21] T. Haraguchi, T. Sakazaki, S. Hamaguchi, and H. Yasuda, “Development of electromagnetic lenses for multielectron beam lithography system,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 20, no. 6, p. 2726, 2002.
- [22] L. R. Baylor *et al.*, “Digital electrostatic electron-beam array lithography,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 20, no. 6, p. 2646, 2002.
- [23] D. S. Pickard, T. R. Groves, W. D. Meisburger, T. Crane, and R. Fabian Pease, “Distributed axis electron beam technology for maskless lithography and defect inspection,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 21, no. 6, p. 2834, 2003.
- [24] T. Haraguchi *et al.*, “Multicolumn cell: Evaluation of the proof of concept system,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 22, no. 3, p. 985, 2004.
- [25] L. R. Baylor *et al.*, “Initial lithography results from the digital electrostatic e-beam

- array lithography concept,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 22, no. 6, p. 3021, 2004.
- [26] R. F. Pease, “Maskless lithography,” *Microelectron. Eng.*, vol. 78–79, pp. 381–392, Mar. 2005.
 - [27] D. S. Pickard, C. Kenney, S. Tanimoto, T. Crane, T. Groves, and R. F. W. Pease, “Monolithic multichannel secondary electron detector for distributed axis electron beam lithography and inspection,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 25, no. 6, p. 2277, 2007.
 - [28] C. S. Silver, J. P. Spallas, and L. P. Muray, “Multiple beam sub-80-nm lithography with miniature electron beam column arrays,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 25, no. 6, p. 2258, 2007.
 - [29] J. Spallas, D. Meisburger, K. Werder, and L. Muray, “High-brightness miniature column for high-speed multicolumn wafer inspection,” *J. Vac. Sci. Technol. B, Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.*, vol. 33, no. 6, p. 06FN03, Nov. 2015.
 - [30] J. Spallas, D. Meisburger, and L. Muray, “Ultralow voltage imaging using a miniature electron beam column,” *J. Vac. Sci. Technol. B, Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.*, vol. 34, no. 6, p. 06KB02, Nov. 2016.
 - [31] J. P. Spallas, C. S. Silver, L. P. Muray, T. Wells, and M. El-Gomati, “A manufacturable miniature electron beam column,” *Microelectron. Eng.*, vol. 83, no. 4–9, pp. 984–989, Apr. 2006.
 - [32] L. Muray, J. Spallas, and D. Meisburger, “On the limits of miniature electron column technology,” in *Proc. of SPIE 9236, Scanning Microscopies*, 2014, no. September 2014, p. 92360C.
 - [33] D. Meisburger, J. Spallas, K. Werder, and L. Muray, “Proposed architecture of a multicolumn electron-beam wafer inspection system for high-volume manufacturing,” *J. Vac. Sci. Technol. B, Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.*, vol. 33, no. 6, p. 06FN01, Nov. 2015.
 - [34] J. E. Schneider, “Semiconductor on glass photocathodes as high-performance sources for parallel electron beam lithography,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 14, no. 6, p. 3782, Nov. 1996.
 - [35] A. W. Baum, “Semiconductor on glass photocathodes for high throughput maskless electron beam lithography,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 15, no. 6, p. 2707, Nov. 1997.
 - [36] W. Barth *et al.*, “Field emission cathode array with self-aligned gate electrode fabricated by silicon micromachining,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 18, no. 6, p. 3544, 2000.
 - [37] M. Mankos *et al.*, “Multisource optimization of a column for electron lithography,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 18, no. 6, p. 3010, 2000.

- [38] E. Yin, A. D. Brodie, F. C. Tsai, G. X. Guo, and N. W. Parker, "Electron optical column for a multicolumn, multibeam direct-write electron beam lithography system," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 18, no. 6, p. 3126, 2000.
- [39] P. Arcuni, S. Presley, V. Aebi, and W. E. Spicer, "Recent tests of negative electron affinity photocathodes as source for electron lithography and microscopy," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 19, no. 6, p. 2585, 2001.
- [40] M. J. Wieland, B. J. Kampherbeek, P. Addessi, and P. Kruit, "Field emission photocathode array for multibeam electron lithography," *Microelectron. Eng.*, vol. 57–58, pp. 155–161, Sep. 2001.
- [41] M. Mankos *et al.*, "Basic constraints for a multibeam lithography column," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 19, no. 2, p. 467, 2001.
- [42] M. Mankos, A. Sagle, S. T. Coyle, and A. Fernandez, "Electron–electron interactions in multibeam lithography columns," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 19, no. 6, p. 2566, 2001.
- [43] F. Machuca, Z. Liu, J. R. Maldonado, S. T. Coyle, P. Pianetta, and R. F. W. Pease, "Negative electron affinity group III-nitride photocathode demonstrated as a high performance electron source," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 22, no. 6, p. 3565, 2004.
- [44] A. van Zuuk *et al.*, "Fabrication and characterization of silicon carbide field-emitter array," *Microelectron. Eng.*, vol. 73–74, pp. 106–110, Jun. 2004.
- [45] Z. Liu, Y. Sun, P. Pianetta, and R. F. W. Pease, "Narrow cone emission from negative electron affinity photocathodes," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 23, no. 6, p. 2758, 2005.
- [46] T. F. Teepen *et al.*, "Fabrication and characterization of p-type silicon field-emitter arrays for lithography," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 23, no. 2, p. 359, 2005.
- [47] M. L. Yu, S. T. Coyle, W. DeVore, and B. Shamoun, "Electron-electron interaction induced beam displacement in a multiple electron beam system," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 23, no. 6, p. 2589, 2005.
- [48] M. W. Geis, S. Deneault, K. E. Krohn, M. Marchant, T. M. Lyszczarz, and D. L. Cooke, "Field emission at 10Vcm^{−1} with surface emission cathodes on negative-electron-affinity insulators," *Appl. Phys. Lett.*, vol. 87, no. 19, p. 192115, Nov. 2005.
- [49] T. H. Newman, "Dot matrix electron beam lithography," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 1, no. 4, p. 999, 1983.
- [50] G. W. Jones, S. K. Jones, M. D. Walters, and B. W. Dudley, "Microstructures for control of multiple ion or electron beams," *IEEE Trans. Electron Devices*, vol. 36, no. 11, pp. 2686–2692, 1989.
- [51] H. Yasuda *et al.*, "Fast Electron Beam Lithography System with 1024 Beams

Individually Controlled by Blanking Aperture Array,” *Jpn. J. Appl. Phys.*, vol. 32, no. Part 1, No. 12B, pp. 6012–6017, Dec. 1993.

- [52] H. Yasuda, “Multielectron beam blanking aperture array system SYNAPSE-2000,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 14, no. 6, p. 3813, Nov. 1996.
- [53] I. L. Berry, “Programmable aperture plate for maskless high-throughput nanolithography,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 15, no. 6, p. 2382, Nov. 1997.
- [54] G. Winograd, V. Krishnamurthi, R. Garcia, L. H. Veneklasen, M. Mankos, and F. Pease, “Demonstration of multiblanker electron-beam technology,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 18, no. 6, p. 3052, 2000.
- [55] M. Muraki and S. Gotoh, “New concept for high-throughput multielectron beam direct write system,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 18, no. 6, p. 3061, 2000.
- [56] M. J. van Bruggen, B. van Someren, and P. Kruit, “Development of a multi-electron-beam source for sub-10 nm electron beam induced deposition,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 23, no. 6, p. 2833, 2005.
- [57] M. Nakasuji, S. Yoshikawa, T. Satake, and N. Noji, “Feasibility Study of Multiple-Beam Scanning Electron Microscopy for Defect Inspection,” *Jpn. J. Appl. Phys.*, vol. 44, no. 7B, pp. 5570–5574, Jul. 2005.
- [58] S. Eder-Kapl *et al.*, “Projection mask-less lithography (PML2): First results from the multi beam blanking demonstrator,” *Microelectron. Eng.*, vol. 83, no. 4–9, pp. 968–971, Apr. 2006.
- [59] S. Tanimoto *et al.*, “Inspection of all beams in multielectron beam system,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 25, no. 2, p. 380, 2007.
- [60] O. Kamimura *et al.*, “Optical properties of a multibeam column with a single-electron source,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 25, no. 1, p. 140, 2007.
- [61] P. Kruit, S. Steenbrink, R. Jager, and M. Wieland, “Optimum dose for shot noise limited CD uniformity in electron-beam lithography,” *J. Vac. Sci. Technol. B*, vol. 22, no. 6, pp. 2948–2955, 2004.
- [62] P. Kruit and S. Steenbrink, “Local critical dimension variation from shot-noise related line edge roughness,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 23, no. 6, p. 3033, 2005.
- [63] A. J. van den Brom, A. H. V. van Veen, W. M. Weeda, G. Z. M. Berglund, M. Wieland, and P. Kruit, “Cathode ray tube type electron gun as a source for multibeam electron lithography,” *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 25, no. 6, p. 2245, 2007.

- [64] P. Kruit, "The role of MEMS in maskless lithography," *Microelectron. Eng.*, vol. 84, no. 5–8, pp. 1027–1032, May 2007.
- [65] H. M. P. Van Himbergen, M. D. Nijkerk, P. W. H. De Jager, T. C. Hosman, and P. Kruit, "High throughput defect detection with multiple parallel electron beams," *J. Vac. Sci. Technol. B*, vol. 25, no. 6, pp. 2521–2525, 2007.
- [66] W. F. Van Dorp, "Sub-10 nm focused electron beam induced deposition," Delft University of Technology, 2008.
- [67] M. J. Van Bruggen, "Multi-electron beam system for high resolution electron beam induced deposition," Delft University of Technology, 2008.
- [68] E. Slot *et al.*, "MAPPER: high throughput maskless lithography," *SPIE Lithogr.*, vol. 6921, p. 69211P–69211P–9, 2008.
- [69] P. Kruit and A. M. Gheidari, "Multi-electron beam systems," in *2009 22nd International Vacuum Nanoelectronics Conference*, 2009, pp. 89–90.
- [70] A. K. Dokania, "Feasibility Study of Schottky Emitter Arrays," Delft University of Technology, 2010.
- [71] A. Mohammadi-Gheidari, C. W. Hagen, and P. Kruit, "Multibeam scanning electron microscope: Experimental results," *J. Vac. Sci. Technol. B, Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.*, vol. 28, no. 6, p. C6G5–C6G10, Nov. 2010.
- [72] A. Mohammadi-Gheidari and P. Kruit, "Electron optics of multi-beam scanning electron microscope," *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 645, no. 1, pp. 60–67, Jul. 2011.
- [73] T. Ichimura, Y. Ren, and P. Kruit, "A large current scanning electron microscope with MEMS-based multi-beam optics," *Microelectron. Eng.*, vol. 113, pp. 109–113, 2014.
- [74] T. Doi, M. Yamazaki, T. Ichimura, Y. Ren, and P. Kruit, "A high-current scanning electron microscope with multi-beam optics," *Microelectron. Eng.*, vol. 159, pp. 132–138, 2016.
- [75] P. Kruit and Y. Ren, "Multi-Beam Scanning Electron Microscope Design," *Microsc. Microanal.*, vol. 22, no. S3, pp. 574–575, 2016.
- [76] E. Platzgummer, C. Klein, and H. Loeschner, "Electron multibeam technology for mask and wafer writing at 0.1 nm address grid," *J. Micro/Nanolithography, MEMS, MOEMS*, vol. 12, no. 3, p. 31108, Aug. 2013.
- [77] M. Malloy *et al.*, "Massively parallel E-beam inspection: enabling next-generation patterned defect inspection for wafer and mask manufacturing," in *Proc. SPIE 9423, Alternative Lithographic Technologies VII*, 2015, vol. 9423, p. 942319.
- [78] V. Marx, "Neurobiology: Brain mapping in high resolution," *Nature*, vol. 503, no. 7474, pp. 147–152, Nov. 2013.

- [79] K. J. Hayworth, J. L. Morgan, R. Schalek, D. R. Berger, D. G. C. Hildebrand, and J. W. Lichtman, “Imaging ATUM ultrathin section libraries with WaferMapper: a multi-scale approach to EM reconstruction of neural circuits,” *Front. Neural Circuits*, vol. 8, no. 68, pp. 1–18, Jun. 2014.
- [80] J. W. Lichtman, H. Pfister, and N. Shavit, “The big data challenges of connectomics,” *Nat. Neurosci.*, vol. 17, no. 11, pp. 1448–1454, Oct. 2014.
- [81] K. D. Cummings *et al.*, “Patterning of defect arrays with e-beam lithography used to develop a high throughput e-beam defect inspection tool,” *J. Vac. Sci. Technol. B, Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.*, vol. 33, no. 6, p. 06FD03, Nov. 2015.
- [82] A. L. Eberle, S. Mikula, R. Schalek, J. Lichtman, M. L. Knothe Tate, and D. Zeidler, “High-resolution, high-throughput imaging with a multibeam scanning electron microscope,” *J. Microsc.*, vol. 259, no. 2, pp. 114–20, 2015.
- [83] A. L. Eberle, R. Schalek, J. W. Lichtman, M. Malloy, B. Thiel, and D. Zeidler, “Multiple-Beam Scanning Electron Microscopy,” *Micros. Today*, vol. 23, no. 2, pp. 12–18, 2015.
- [84] T. Kemen *et al.*, “Further advancing the throughput of a multi-beam SEM,” in *Proc. SPIE 9424, Metrology, Inspection, and Process Control for Microlithography XXIX*, 2015, vol. 9424, p. 94241U.
- [85] M. Malloy *et al.*, “Enabling future generation high-speed inspection through a massively parallel e-beam approach,” in *2015 26th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 2015, pp. 266–271.
- [86] M. Malloy *et al.*, “Enabling inspection solutions for future mask technologies through the development of massively parallel E-Beam inspection,” in *Proc. SPIE 9661, 31st European Mask and Lithography Conference*, 2015, vol. 9661, p. 96610O.
- [87] B. Bunday, M. Mukhtar, M. Malloy, and B. Thiel, “Enabling Future Generation High-Speed E-beam Defect Inspection,” in *Metrology and Inspection Forum (Semicon Korea)*, 2016.
- [88] A. L. Eberle and D. Zeidler, “Multi-beam Electron Microscopy: Principles and Applications,” *Microsc. Microanal.*, vol. 22, no. S3, pp. 584–585, 2016.
- [89] B. D. Bunday, M. Mukhtar, K. Quoi, B. Thiel, and M. Malloy, “Simulating massively parallel electron beam inspection for sub-20 nm defects,” in *Proc. SPIE 9424, Metrology, Inspection, and Process Control for Microlithography XXIX*, 2015, vol. 9424, p. 94240J.
- [90] “The International Technology Roadmap for Semiconductors 2.0,” 2014.
- [91] T. F. Crimmins, “Defect metrology challenges at the 11-nm node and beyond,” in *Proc. SPIE 7638, Metrology, Inspection, and Process Control for Microlithography XXIV*, 2010, p. 76380H.

- [92] B. Thiel, "Feasibility Study and Performance Specifications of a Multi-column Electron Beam Wafer Defect Inspection Tool [White Paper]," Albany, 2012.
- [93] B. Thiel, M. Mukhtar, K. Quoi, B. D. Bunday, and M. Malloy, "Patterned Wafer Inspection with Multi-beam SEM Technology," *Microsc. Microanal.*, vol. 22, no. S3, pp. 586–587, Jul. 2016.
- [94] A. C. Diebold and D. C. Joy, "Current State of Defect Review by Electron Beam Tools," Houston, 2000.
- [95] D. C. Joy, C. S. Joy, and R. D. Bunn, "Measuring the performance of scanning electron microscope detectors," *Scanning*, vol. 18, no. 8, pp. 533–538, Dec. 2006.
- [96] A. Vaid *et al.*, "Hybrid metrology universal engine: co-optimization," in *Proc. SPIE 9048, Metrology, Inspection, and Process Control for Microlithography XXVIII*, 2014, vol. 9048, p. 905009.
- [97] J. Hazart *et al.*, "Data fusion for CD metrology: heterogeneous hybridization of scatterometry, CDSEM, and AFM data," 2014, no. April 2014, p. 90502L.
- [98] M.-A. Henn *et al.*, "Optimizing hybrid metrology: rigorous implementation of Bayesian and combined regression," *J. Micro/Nanolithography, MEMS, MOEMS*, vol. 14, no. 4, p. 44001, Nov. 2015.
- [99] X. Zhang *et al.*, "Addressing FinFET metrology challenges in 1× node using tilt-beam critical dimension scanning electron microscope," *J. Micro/Nanolithography, MEMS, MOEMS*, vol. 13, no. 4, p. 41407, Oct. 2014.
- [100] B. Su, R. Oshana, M. Menaker, Y. Barak, and X. Shi, "Shape control using sidewall imaging," in *Proc. SPIE 3998, Metrology, Inspection, and Process Control for Microlithography XIV*, 2000, pp. 232–238.
- [101] B. Bunday *et al.*, "The Coming of Age of Tilt CD-SEM," *Proc. SPIE 6518, Metrol. Insp. Process Control Microlithogr. XXI*, vol. 6518, pp. 1–16, 2007.
- [102] T. Marschner, G. Eytan, and O. Dror, "Determination of best focus and exposure dose using CD-SEM sidewall imaging," in *Proc. SPIE 4344, Metrology, Inspection, and Process Control for Microlithography XV*, 2001, p. 355.
- [103] B. D. Bunday, "Quantitative profile-shape measurement study on a CD-SEM with application to etch-bias control and several different CMOS features," in *Proc. SPIE 5038, Metrology, Inspection, and Process Control for Microlithography XVII*, 2002, vol. 5038, pp. 383–395.
- [104] B. D. Bunday, M. Bishop, J. R. Swyers, and K. R. Lensing, "Quantitative profile-shape measurement study on a CD-SEM with application to etch-bias control," in *Proc. SPIE 4689, Metrology, Inspection, and Process Control for Microlithography XVI*, 2003, vol. 5038, p. 383.
- [105] M. LaPedus, "Can We Measure Next-Gen FinFETs?," *Semiconductor Engineering*, 2016. [Online]. Available: <http://semiengineering.com/can-we-measure-next-gen->

finfets/. [Accessed: 06-Jan-2017].

- [106] C. Shishido, M. Tanaka, and M. Osaki, "CD-bias reduction in CD-SEM line-width measurement for the 32-nm node and beyond using the model-based library method," *Proc. SPIE*, vol. 7272, no. mode I, p. 72722C–72722C–10, 2009.
- [107] C. Shishido, M. Tanaka, and M. Osaki, "Accurate measurement of very small line patterns in critical dimension scanning electron microscopy using model-based library matching technique," *J. Micro/Nanolithography, MEMS MOEMS*, vol. 10, no. 1, p. 13010, 2011.
- [108] B. Bunday, A. Cepler, A. Cordes, and A. Arceo, "CD-SEM metrology for sub-10nm width features," in *Proc. of SPIE*, 2014, vol. 9050, p. 90500T.
- [109] A. E. Vladár *et al.*, "10nm three-dimensional CD-SEM metrology," in *Proc. of SPIE Vol. 9050, Metrology, Inspection, and Process Control for Microlithography XXVIII*, 2014, p. 90500A.
- [110] V. N. Tondare, J. S. Villarrubia, and A. E. Vladár, "Three-Dimensional (3D) Nanometrology Based on Scanning Electron Microscope (SEM) Stereophotogrammetry," *Microsc. Microanal.*, vol. 23, no. 5, pp. 967–977, Oct. 2017.
- [111] J. Xie, "Stereomicroscopy: 3D Imaging and the Third Dimension Measurement," Palo Alto, CA, 2011.
- [112] M. Eulitz and G. Reiss, "3D reconstruction of SEM images by use of optical photogrammetry software," *J. Struct. Biol.*, vol. 191, no. 2, pp. 190–196, Aug. 2015.
- [113] A. P. Tafti, A. B. Kirkpatrick, Z. Alavi, H. A. Owen, and Z. Yu, "Recent advances in 3D SEM surface reconstruction," *Micron*, vol. 78, pp. 54–66, Nov. 2015.
- [114] L. C. Gontard, R. Schierholz, S. Yu, J. Cintas, and R. E. Dunin-Borkowski, "Photogrammetry of the three-dimensional shape and texture of a nanoscale particle using scanning electron microscopy and free software," *Ultramicroscopy*, vol. 169, pp. 80–88, Oct. 2016.
- [115] L. C. Gontard *et al.*, "Assessment of engineered surfaces roughness by high-resolution 3D SEM photogrammetry," *Ultramicroscopy*, vol. 177, pp. 106–114, 2017.
- [116] G. Piazzesi, "Photogrammetry with the scanning electron microscope," *J. Phys. E.*, no. 6, pp. 392–396, 1973.
- [117] W. Drzazga, J. Paluszynski, and W. Slowko, "Three-dimensional characterization of microstructures in a SEM," *Meas. Sci. Technol.*, vol. 17, no. 1, pp. 28–31, Jan. 2006.
- [118] W. Ito *et al.*, "Novel three dimensional (3D) CD-SEM profile measurements," 2014, p. 90500D.
- [119] M. Mukhtar, K. Quoi, B. D. Bunday, M. Malloy, and B. Thiel, "Image Simulation and Analysis to Predict the Sensitivity Performance of a Multi-Electron Beam Wafer Defect Inspection Tool," *Microsc. Microanal.*, vol. 22, no. S3, pp. 620–621, Jul. 2016.

- [120] J. S. Villarrubia, N. W. M. Ritchie, and J. R. Lowney, "Monte Carlo modeling of secondary electron imaging in three dimensions," in *Proc. SPIE 6518, Metrology, Inspection, and Process Control for Microlithography XXI*, 2007, p. 65180K.
- [121] J. R. Lowney, A. E. Vladar, and M. T. Postek, Jr., "High-accuracy critical-dimension metrology using a scanning electron microscope," in *Proc. SPIE 2725, Metrology, Inspection, and Process Control for Microlithography X*, 1996, pp. 515–526.
- [122] J. R. Lowney, "Monte Carlo simulation of scanning electron microscope signals for lithographic metrology," *Scanning*, vol. 18, no. 4, pp. 301–306, Dec. 2006.
- [123] J. S. Villarrubia and Z. J. Ding, "Sensitivity of SEM width measurements to model assumptions," in *Proc. SPIE 7272, Metrology, Inspection, and Process Control for Microlithography XXIII*, 2009, p. 72720R.
- [124] J. S. Villarrubia and Z. J. Ding, "Sensitivity of scanning electron microscope width measurements to model assumptions," *J. Micro/Nanolithography, MEMS, MOEMS*, vol. 8, no. 3, p. 33003, Jul. 2009.
- [125] A. J. Cepler, B. Bunday, B. L. Thiel, and J. S. Villarrubia, "Scanning electron microscopy imaging of ultra-high aspect ratio hole features," in *Proc. SPIE 8324, Metrology, Inspection, and Process Control for Microlithography XXVI*, 2012, p. 83241N.
- [126] J. S. Villarrubia *et al.*, "Scanning electron microscope measurement of width and shape of 10nm patterned lines using a JMONSEL-modeled library," *Ultramicroscopy*, vol. 154, no. July, pp. 15–28, Jul. 2015.
- [127] A. J. Cepler, "Charged Particle Imaging Methods for CD Metrology of Sub 22nm 3D Device Structures," State University of New York at Albany, 2013.
- [128] A. J. Cepler and B. L. Thiel, "Improving the performance of the critical dimension-scanning electron microscope with the contrast transfer function," in *Proc. of SPIE 8378, Scanning Microscopies 2012: Advanced Microscopy Technologies for Defense, Homeland Security, Forensic, Life, Environmental, and Industrial Sciences*, 2012, vol. 8378–30, p. 83780T.
- [129] D. C. Joy, J. Michael, and B. Griffin, "Evaluating SEM performance from the contrast transfer function," in *Proc. SPIE 7638, Metrology, Inspection, and Process Control for Microlithography XXIV*, 2010, vol. 7368, no. 2, p. 76383J.
- [130] G. D. Evangelidis and E. Z. Psarakis, "Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1858–1865, Oct. 2008.
- [131] Carl Zeiss GmbH, "MultiSEM 505/506." [Online]. Available: <https://www.zeiss.com/microscopy/us/products/scanning-electron-microscopes/multisem.html>. [Accessed: 06-Jan-2017].
- [132] H. H. Ku, "Notes on the use of propagation of error formulas," *J. Res. Natl. Bur. Stand. Sect. C Eng. Instrum.*, vol. 70C, no. 4, p. 263, Oct. 1966.

- [133] M. Mukhtar, B. D. Bunday, K. Quoi, M. Malloy, and B. Thiel, “Measuring multielectron beam imaging fidelity with a signal-to-noise ratio analysis,” *J. Micro/Nanolithography, MEMS, MOEMS*, vol. 15, no. 3, p. 34004, Aug. 2016.
- [134] C. A. Mack and B. D. Bunday, “Analytical linescan model for SEM metrology,” in *Proc. of SPIE 9424, Metrology, Inspection, and Process Control for Microlithography XXIX*, 2015, vol. 9424, p. 94240F.
- [135] C. A. Mack and B. D. Bunday, “Improvements to the analytical linescan model for SEM metrology,” in *Proc. of SPIE 9778, Metrology, Inspection, and Process Control for Microlithography XXX*, 2016, vol. 9778, p. 97780A.
- [136] C. A. Mack and B. D. Bunday, “Using the analytical linescan model for SEM metrology,” in *Proc. of SPIE 10145, Metrology, Inspection, and Process Control for Microlithography XXXI*, 2017, p. 101451R.

APPENDIX

A. THE JMONSEL SIMULATOR

The Java package JMONSEL expands upon the Electron Probe Quantification (EPQ) library [1] which was first developed to support x-ray microanalysis (capable of simulating x-ray generation, transmission, and detection & resulting x-ray spectra). In the EPQ library, electrons with energies insufficient to generate x-rays were functionally neglected. Though uninteresting to x-ray microanalysis, these electrons are significant for SEM imaging i.e. especially for SE images. Thus, JMONSEL was developed to address the relevant physics of this kind of imaging.

Model simulations are implemented by the execution of Jython (an application of the programming language Python intended to run on the Java platform) script whereby the script typically accesses the package that comprises utilities for carrying out Monte Carlo simulations

of electron trajectories through 3D samples (NISTMonte) and JMONSEL so as to delineate sample geometry and material sets, materials scattering models, electron gun parameters, initialize detectors, and set number of electron trajectories to be observed. After simulation, a resulting signal such as yield is calculated from detector observation. The previous steps are typically repeated in an iterative or loop fashion to produce the desired outcome such as yield vs. electron beam energy or position, etc.

The area of simulation is separated into regions of uniform composition. These regions are represented by constructive solid geometry (CSG) wherein 3D primitive structures (e.g., cylinders, spheres, polyhedrons, etc.) are combined to using simple set operations (union, difference, intersection, etc.) to construct more complex figures. Figure generation here is hierarchical with the root of the hierarchy being a spherical chamber region. Sample parts are added as subregions of the chamber and each subregion may have its own subregion(s). Sample shapes can also be transformed by any affine transformation or represented as height maps. Lastly, tetrahedral meshed regions may be imported from a file in the format used by Gmsh [2], a freely available (GNU General Public License) meshing software. These tetrahedra are converted to modified CSG shapes. For the effects of charging to be modeled JMONSEL requires the use of meshed regions.

Besides the shape of its boundary, a region is defined by the material it contains. A Material Scatter Model includes the material's elemental composition, stoichiometry, density, and basic electronic properties such as band gap, work function, Fermi energy, etc. As well, the scattering

properties of electrons in the material which include free path as a function of energy and a technique to compute the consequence of a scattering event. A technique to deal with boundary crossings of electrons exiting the material is also needed. Here the method decides the consequences of an electron reaching a crossing boundary i.e., whether transmission or reflection occurs at the boundary. Moreover, a continuous slowing down approximation (CSDA) model for the material is next. This stipulates the energy amount lost by the electron as a function of initial energy and distance traversed in the material. If, for example, a discrete inelastic scattering model is utilized where all energy losses are already included the continuous loss amount may be set to zero. Finally, a minimum electron energy is set where below which the electron is dropped from the simulation.

The electron trajectories themselves are divided into steps during simulation. The electron gun determines the position, energy, and direction of motion at the outset of the first step. In succeeding steps, these factors are equal to results from the previous. Primary electron scatter may occur prior to reaching material boundaries, where they may generate an SE or not – dependent on the type of scattering event taking place. On the other hand, if an electron reaches a boundary it will either be transmitted or reflect off.

Steps initiate with movement along the electron's initial direction of motion. The length of this movement is determined by the lesser of either the distance to the next boundary crossing or the electron's scattering free path (λ). The scattering free path is $\lambda = -\lambda_{\text{mfp}} \ln(R)$, where R symbolizes a random number uniformly generated between 0 and 1 and λ_{mfp} is the mean free

path ($\lambda_{\text{mfp}}^{-1} = \sum_i \lambda_i^{-1}$). Here, λ_i is the mean free path of the i^{th} scattering mechanism with summation of all scattering mechanisms given to the material in which the electron is located in. The random number logarithm $\ln(R)$ makes it so that the scattering free path is Poisson-distributed with mean value λ_{mfp} . If a CSDA component is contained in the transport model for the electron's position, the energy of the electron is discretely reduced. Though, if the scattering free path is greater than the distance to the nearest boundary, the energy and direction of the electron's motion at the culmination of the step are decided by the boundary crossing model. In this model, the potential energy is taken to be $U(x) = \Delta U / [1 + \exp(\frac{2x}{w})]$, where x is the perpendicular distance to the boundary and w is the boundary width. Potential energy change at material boundaries lead to refraction/reflection and a kinetic energy change. Thereafter, Schrödinger's equation can be solved analytically for the transmission probability. Apart from these, the scattering mechanism that ends the step is to be randomly chosen with probabilities weighted by their inverse mean free paths. The selected scattering mechanism is responsible for the final energy, direction and whether an SE is generated by the electron. To conclude, the final trajectory step is to dump the electron from additional simulation if the drop conditions have been reached. These are usually based on the electron's final energy not passing a certain threshold or whether a scattering event such as a trap specifies that the trajectory simulation must end.

Events are generated by the simulator at noteworthy times such as the start of the first of a set of multiple trajectories, the commencement and culmination of each individual trajectory, on each scattering occasion, once an SE is generated or its trajectory ends, as soon as an electron

crosses a regional boundary or hits the chamber wall. Detectors are notified of these events and record them for possible further analysis.

JMONSEL incorporates a variety of specimen-particle interaction physics models in executions of simulations. For elastic electron scattering it is common to use a hybrid of three algorithms: approximations of the solution of Mott's scattering equations for $50 \text{ eV} \leq E \leq 20 \text{ keV}$, screened Rutherford differential cross section for $E > 20 \text{ keV}$, and extrapolation below 50 eV according to Browning's formula. Secondary electron generation (inelastic scatter) can be calculated by a choice of two methods: dielectric function theory (DFT) or fitted inelastic scattering. While fitted inelastic is the simpler of the two models it is advantageous when the energy loss function (ELF) is not known as is the case for many materials imaged in semiconductor electronics and other applications. The fitted inelastic scattering model also takes a stopping power model as input while the DFT model does not. JMONSEL also includes a longitudinal optical phonon scattering model based on that of Llacer and Garwin as well as an electron trapping model based on that of Ganachaud and Mokrani. More detailed information on JMONSEL and the models it implements have been relayed previously [3]. Though JMONSEL provides calculations based on the most advanced specimen-particle interaction physics it has yet to be efficiently constructed for parallelization for high performance computing. Others have ventured to produce such programs capable of advanced parallelization with promising results [4], [5].

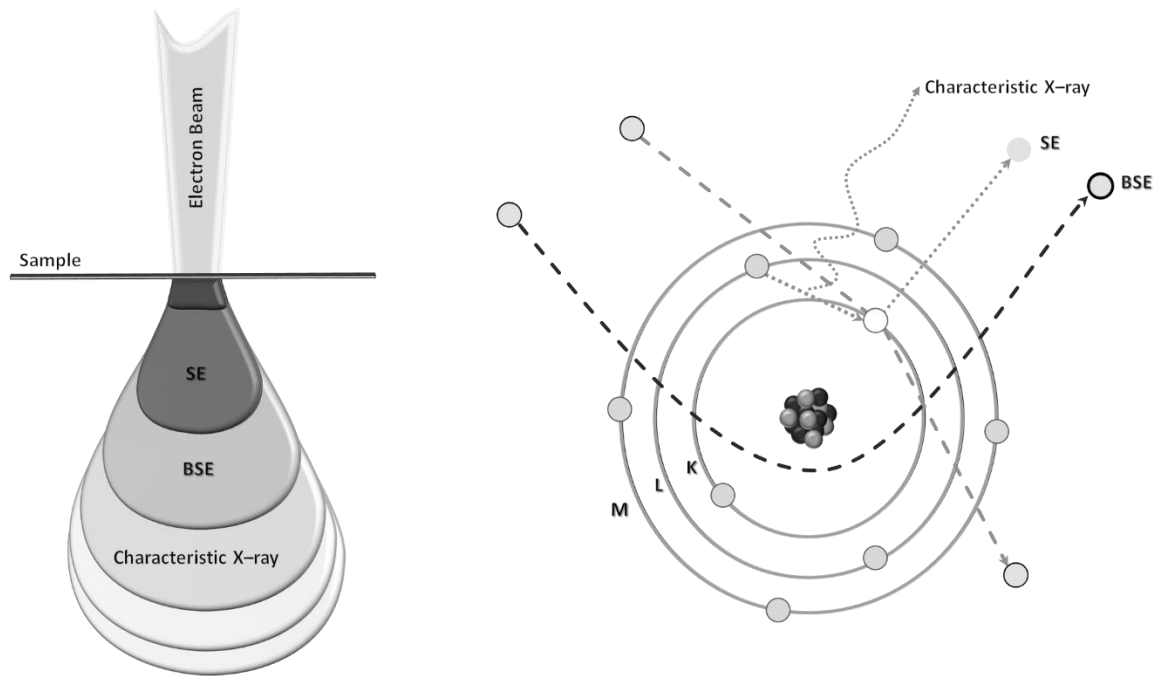


Figure A - 1 Left – illustration of the interaction volume produced when an electron beam impinges upon a sample and the relative volumes from which certain data is retrieved. The actual size and depth of the interaction volume is dependent on the sample material and the electron beam energy at play. Right – illustration of the source of secondary (SE) and backscattered electrons (BSE). Backscattered electrons are those primary electrons that have elastic interaction with the atom nucleus while secondary electrons are those that are forced from orbit by inelastic scatter by a primary electron.

References

- [1] N. W. M. Ritchie, “A new Monte Carlo application for complex sample geometries,” *Surf. Interface Anal.*, vol. 37, no. 11, pp. 1006–1011, Nov. 2005.
- [2] C. Geuzaine and J.-F. Remacle, “Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities,” *Int. J. Numer. Methods Eng.*, vol. 79, no. 11, pp. 1309–1331, Sep. 2009.
- [3] J. S. Villarrubia *et al.*, “Scanning electron microscope measurement of width and shape of 10nm patterned lines using a JMONSEL-modeled library,” *Ultramicroscopy*, vol. 154, no. July, pp. 15–28, Jul. 2015.
- [4] M. Ciappa, E. Ilgunsatiroglu, and A. Y. Illarionov, “Simulation of the measurement by scanning electron microscopy of edge and linewidth roughness parameters in nanostructures,” in *EUROSOI-ULIS 2015: 2015 Joint International EUROSOI Workshop and International Conference on Ultimate Integration on Silicon*, 2015, pp.

205–208.

- [5] E. Ilgünsatiroglu, “Synthesis of SEM images by means of High-Performance Computing,” ETH Zürich, 2015.

B. BACKSCATTERED ELECTRON SIMULATIONS TO EVALUATE SENSITIVITY
AGAINST ELECTRON DOSAGE OF BURIED FEATURES

Introduction

At present, efforts to advance semiconductor device performance utilize the reduction of feature size along with advanced non-planar architecture design. In the fabrication of these devices, the demand for greater precision measurement of the end dimensions of contact holes and deep trenches has risen. Furthermore, each layer's overlay alignment precision has similarly become very significant.

Overlay measurements of device patterns have conventionally been performed using optical methods. Beginning with image-based techniques using box-in-box targets that progressed to aerial image (AIM) and multi-level blossom targets to the more recently employed diffraction-based overlay (DBO). Another way to do this measurement is use of SEM overlay, which is now under discussion for use in in-device overlay, because overlay measurements from dedicated kerf structures frequently do not match performance in-circuit, use cases demanding enhanced resolution, as well as a reference metrology. Two main application spaces are measurement features from multiple mask levels on the same surface and buried features [1].

Contemporary CD-SEMs are proficient at measuring overlay for cases where all features are on the surface. So as to measure overlay of buried features, high voltage SEM (HV-SEM) is needed. Gate-to-fin and back end of line (BEOL) overlay are central use cases for this method. The generation of realistic virtual data for the case of buried features is a multi-step process whereby virtual samples are realistically designed and resultant simulations are analyzed for coherence.

In an effort to probe the backscattered electron behavior of buried features, the backscattered electron data profile is considered as an image stack to further explore sensitivity dependency versus energy loss. Here, a Java Monte Carlo Simulator of Secondary Electrons (JMONSEL) [2], [3] simulation investigation was performed using 10 nm line/space gratings of graduated increase in depth of burial. Results obtained from these simulations for backscattered energy loss were applied to compute the sensitivity measurements of buried features versus electron dosage for a selection of electron beam voltages. Results indicate that there may be optimal energy loss windows and imaging dosages for the detection of buried features.

Objectives

The making and analysis of realistic virtual samples for the buried features use case entails a number of steps. Concisely, these steps comprise (1) creating a pixelated virtual sample containing the feature shape(s) and material set(s), (2) simulating nominal electron emission behavior, and (3) analyzing the resultant imaging data stream to assess the sensitivity of buried features versus electron dosage applied emission behavior. This process is illustrated

graphically in Figure B - 1. This process is patterned after a more detailed simulated defect inspection process provided elsewhere [4]–[6].

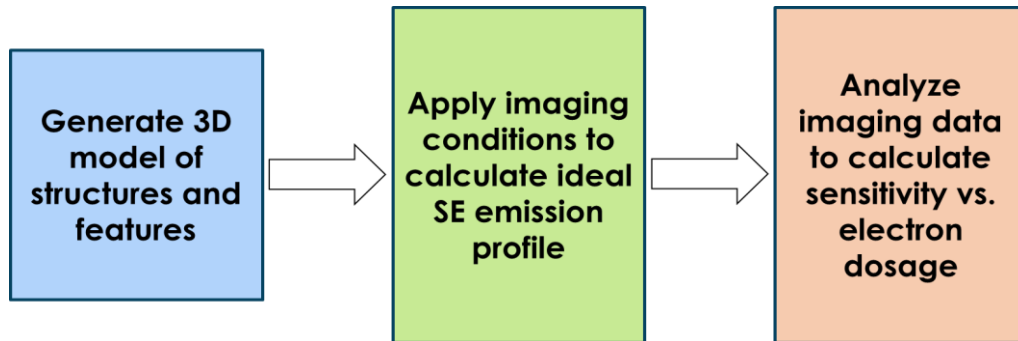


Figure B - 1 Flowchart describing the flow of constructing a simulated image.

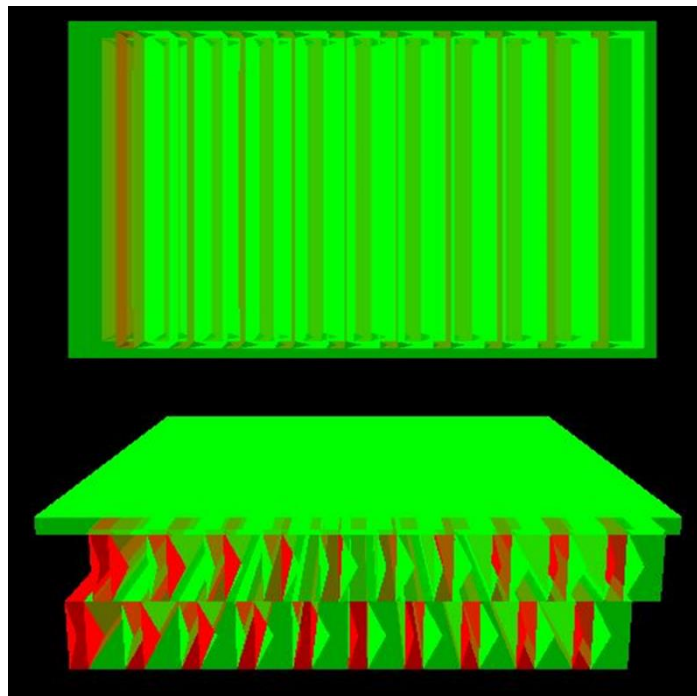


Figure B - 2 Top down and cross sectional view of an example 3D model of buried semiconductor features that used for simulating emission behavior.

Methods

Below in Figure B - 3 are backscattered electron simulated images and a grayscale plot of the 3D model from Figure B - 2. The gratings are 40 nm in height, 10 nm in width, and a pitch of 30 nm. They are filled and overlaid (10 nm) with SiO₂. For each pixel, N incident electrons are simulated, and the numbers of secondary (SEs) and backscattered electrons (BSEs) reaching the detector were recorded. Here, SEs are defined as those reaching the detector with ≤ 50 eV energy, while BSEs are those with > 50 eV. The yield is given by the ratio of electrons arriving at the detector to the number incident at the sample. By means of SE emission being a normally distributed stochastic process, the uncertainty in the predicted yield scales as the inverse square root of the number of incident electron trials. Thus, $N = 10,000$ yields an emission coefficient with 1% uncertainty in the predicted value (suitable for rapid screening of outputs), whereas the $N = 1,000,000$ produces an estimate with an uncertainty of 0.1%.

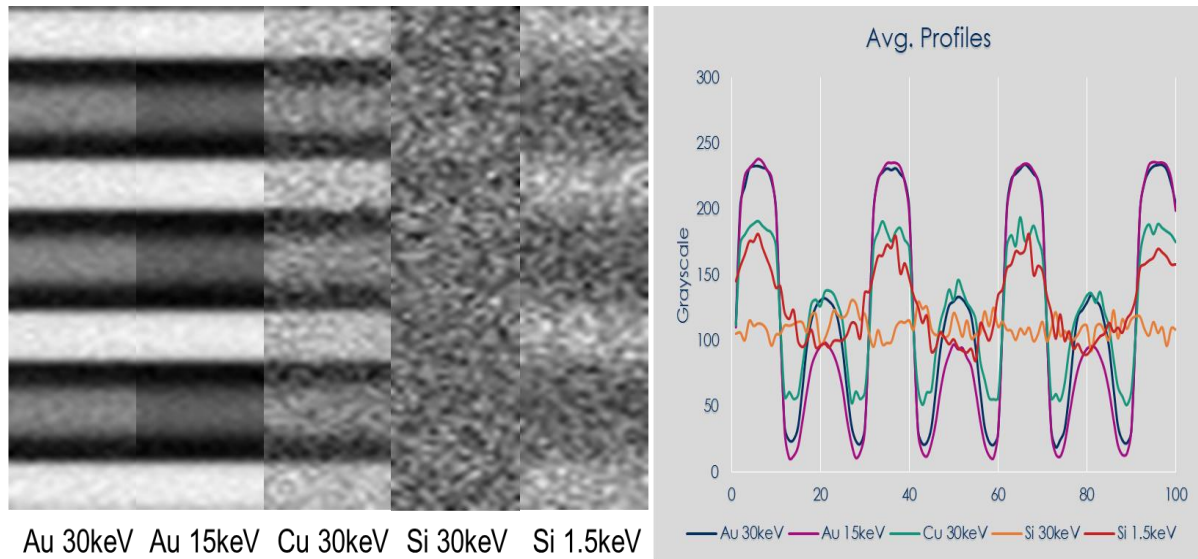


Figure B - 3 Virtual images of several cases of buried semiconductor features of varying material and voltage used for imaging ($N = 10,000$). Simulations were carried out using JMONSEL.

These and previous outcomes illustrate that SEM overlay does not seem a good candidate for fin-to-gate overlay because the contrast between Si and SiO₂ under SiO₂ is weak (can only be seen under up to 40 nm SiO₂). Though, contrast between Cu and SiO₂ is much better (Z and density are considerably different) so that individual Cu lines can be resolved under up to ~100 nm of SiO₂, and a centroid of the grating can be detected at much greater depths, which might also allow for some kind of overlay measurement. As such, prospective SEM overlay appears to be largely suited for BEOL applications¹.

Analysis development

To further investigate the backscattered electron behavior of buried features, the backscattered data profile is considered as an image stack to further explore sensitivity dependency versus energy loss. Example stack profiles are shown below for the cases of the Au and Cu gratings buried under SiO₂ imaged under 30 keV from Figure B - 3.

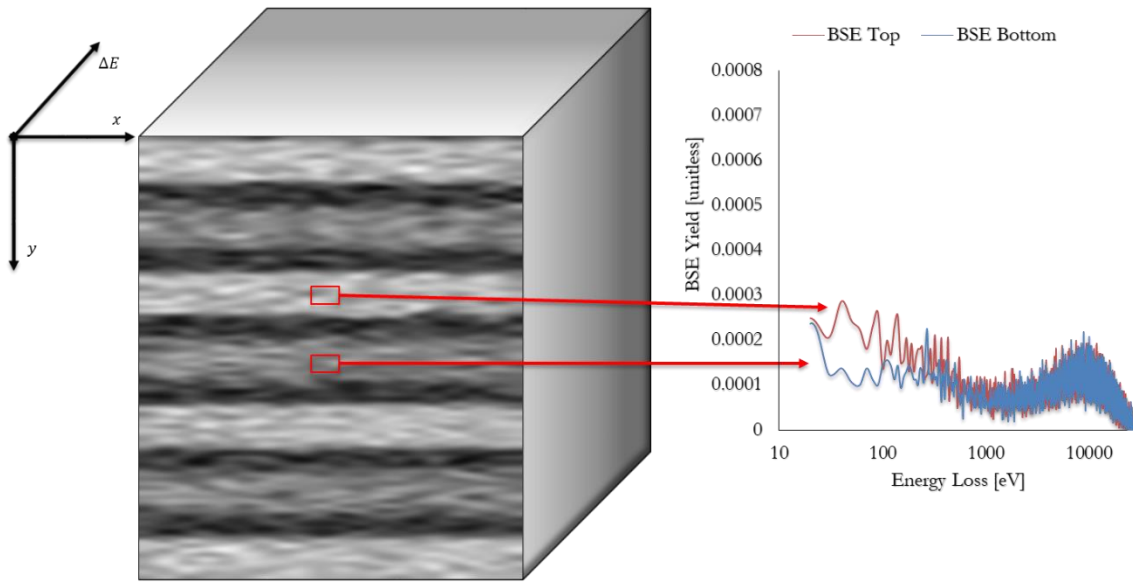


Figure B - 4 Example simulated energy loss stack profile for buried Cu gratings filled with SiO₂ under SiO₂ overlayer.

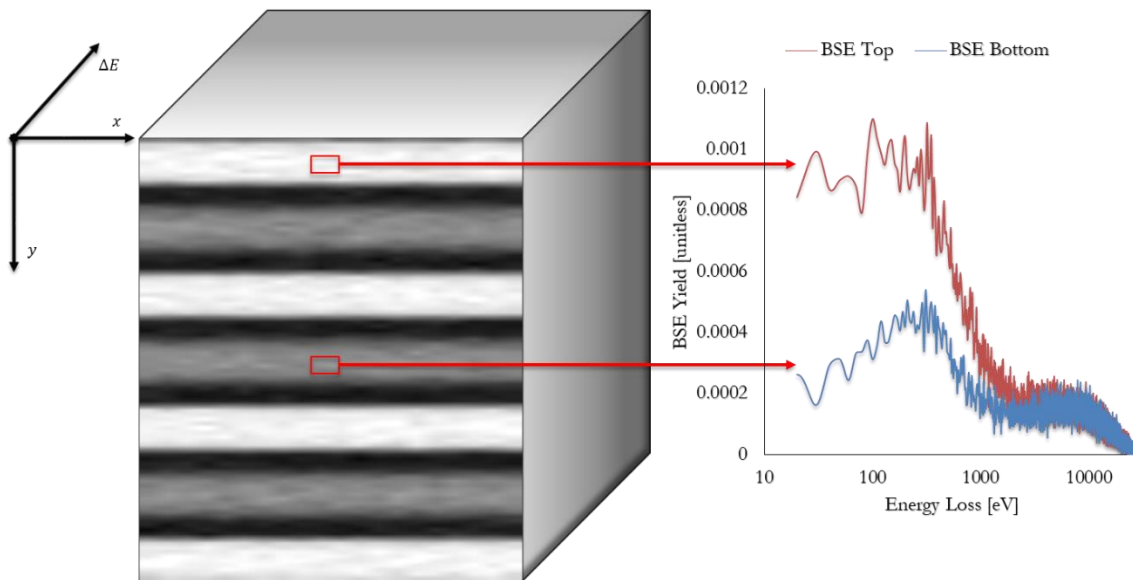


Figure B - 5 Example simulated energy loss stack profile for buried Au gratings filled with SiO₂ under SiO₂ overlayer.

Figure B - 6 shows a 3D model of a graduated increase in depth of burial for 10 nm/linespace gratings filled with SiO₂ under SiO₂ overlayer on the left. On the right it shows the plot of sensitivity vs. voltage for Au, Cu, and Si gratings under a dosage of 625 electrons per pixel for the two cases of 10 nm and 30 nm depth of burial (at $N = 1,000,000$). Sensitivity ($\Delta S/N$), here, is measured using the formula

$$\frac{C * \eta_{ref} \left(\frac{I_B * \tau}{e} \right)}{\sqrt{\eta_{ref} \left(\frac{I_B * \tau}{e} \right)}}.$$

In the formula, C is the contrast between the material in question and the background, η_{ref} is the backscatter yield of the background material in question, and $(I_B * \tau)/e$ is the electron dosage (where I_B is the beam current, τ is the dwell time, and e is the elementary charge). These plots show that for these materials the greatest sensitivity is seen at landing energies of around 5 keV. Also, that for the Si gratings case the sensitivity has a drastic fall off above this energy range. Taking this into account the final run of simulations were run for the landing energies of 5 and ~30 keV.

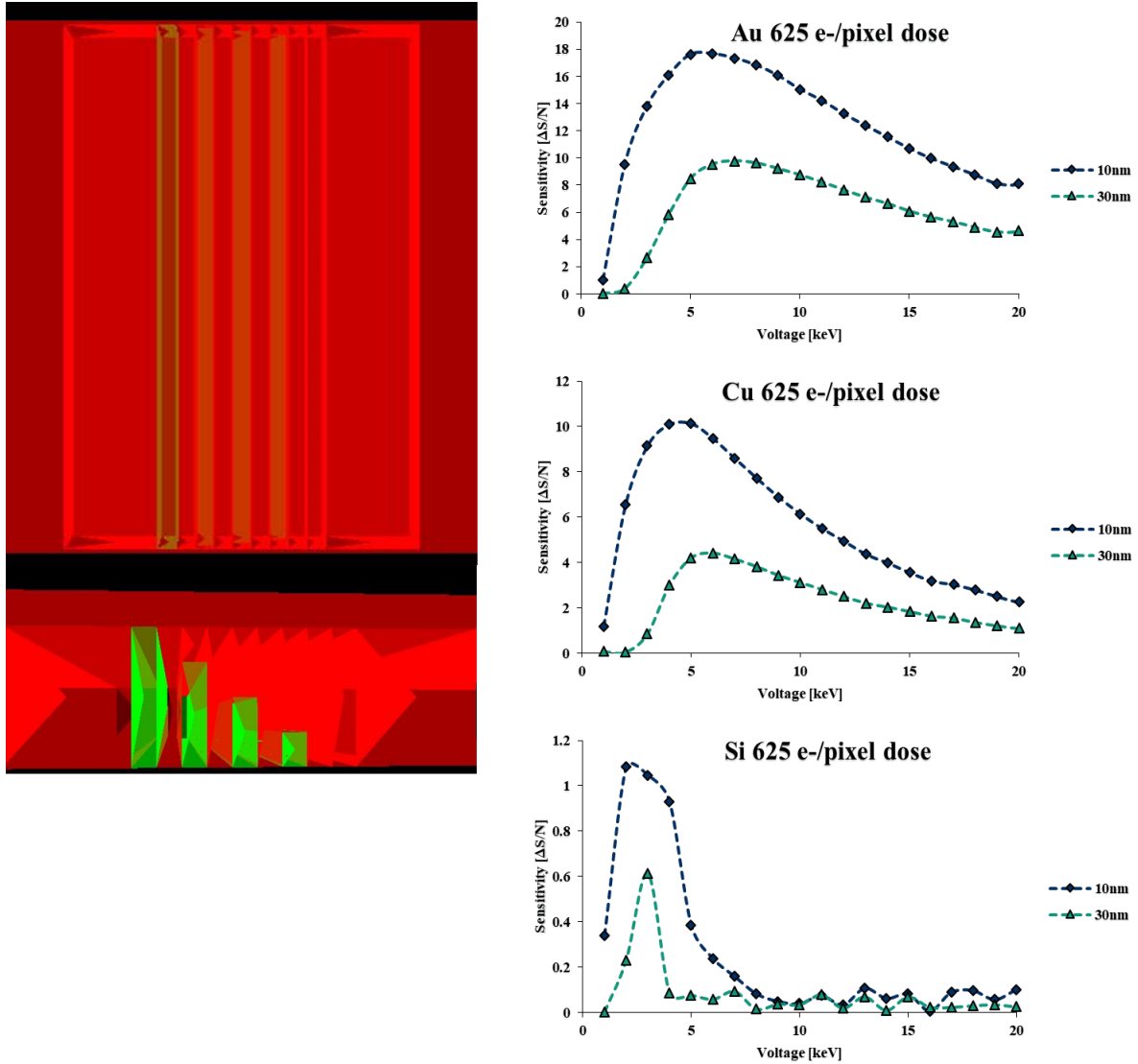


Figure B - 6 Left, 3D model of graduated increase in depth of burial for 10 nm/linespace gratings filled with SiO₂ under SiO₂ overlayer. Right, Sensitivity vs. Voltage for Au, Cu, and Si gratings under 625 e-/pixel dosage.

Results

Below (Figure B - 7 and Figure B - 8), the multi-step process for the calculation of sensitivity vs. dwell time (in nanoseconds) is shown. First, the BSE yield vs. energy loss is plotted ($N = 1,000,000$). The varying lines represent the heights of the gratings buried in 50 nm of SiO₂. Next, the plot of contrast vs. energy loss followed by the plot of sensitivity vs. energy loss

where lines represent the depth the gratings are buried under and finally the max contrast points are used to plot the sensitivity of the buried feature vs. dwell time of the beam of probing electrons. The backscattered electron image stacks were binned in energy loss intervals of 10 eV.

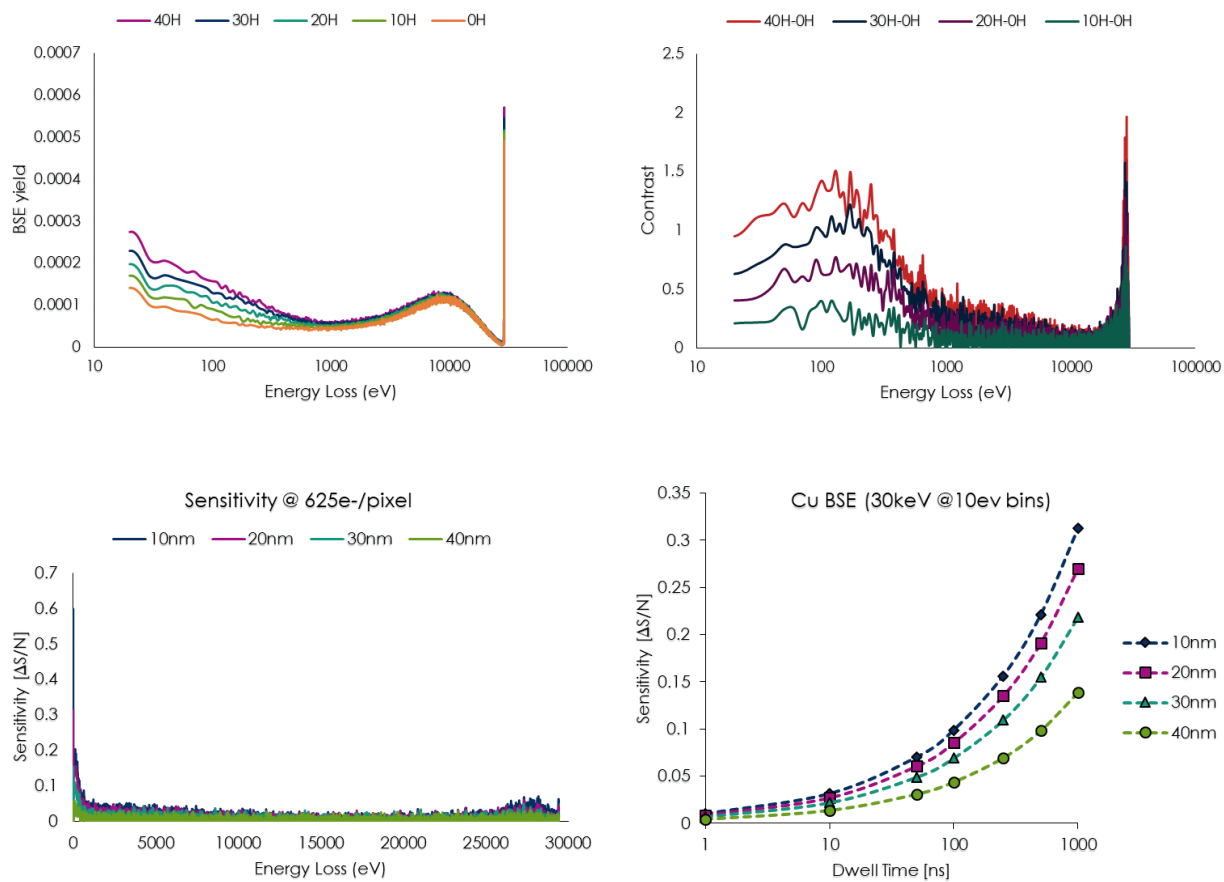


Figure B - 7 Left to Right: Graphs of backscattered yield vs. energy loss, contrast vs. energy loss, sensitivity vs. energy loss, and sensitivity vs. dwell time (electron dosage @ 1nA) for buried Cu/SiO₂ grating under SiO₂ overlayer under 30 keV landing energy.

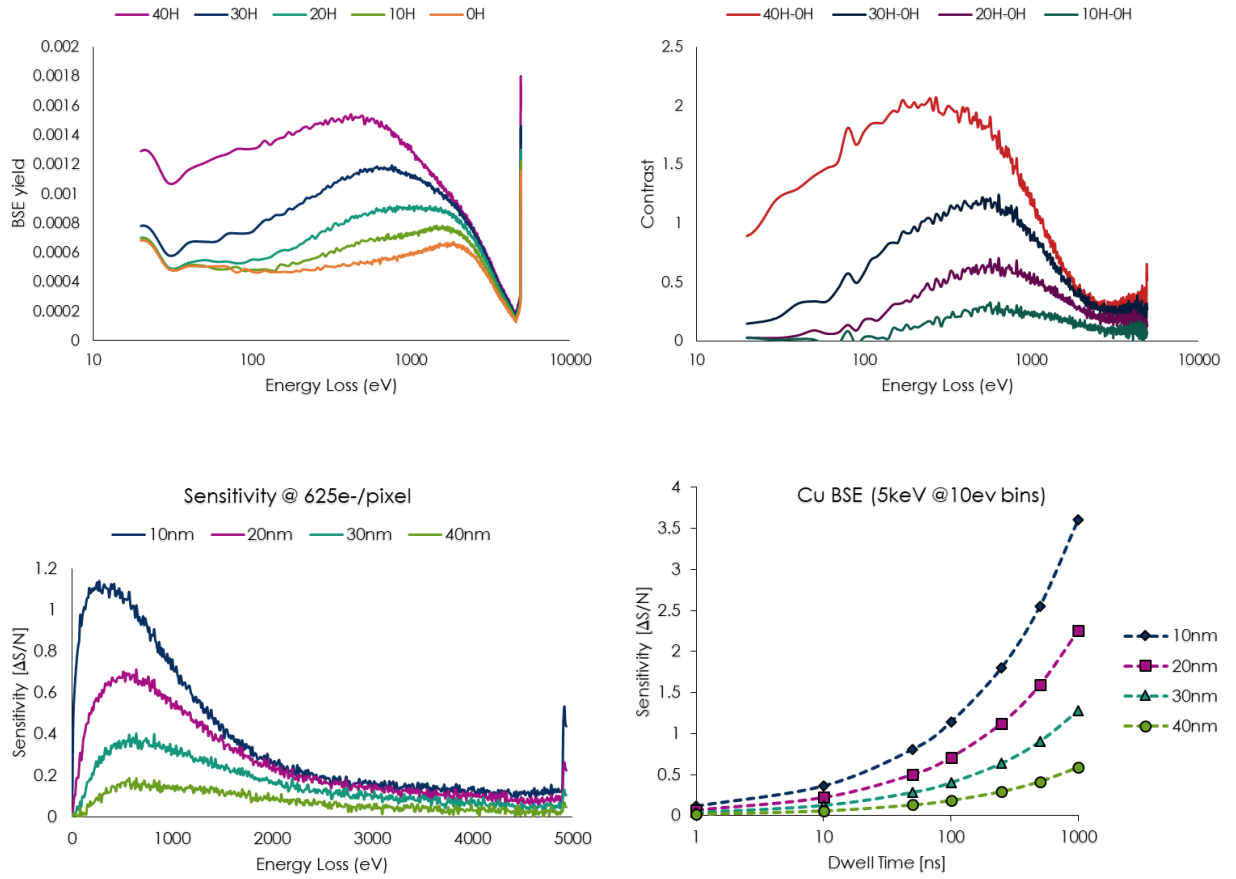


Figure B - 8 Left to Right: Graphs of backscattered yield vs. energy loss, contrast vs. energy loss, sensitivity vs. energy loss, and sensitivity vs. dwell time (electron dosage @ 1nA) for buried Cu/SiO₂ grating under SiO₂ overlayer under 5 keV landing energy.

As could have been inferred from the previous figures, the sensitivity of the buried Si gratings in Figure B - 9 is relatively weak and the differentiation based on dwell time follows accordingly. For the buried Au gratings, the results are clearer. The sensitivity results are better under 5 keV landing energy until the depth of burial increases to 30 nm where the 30 keV landing energy produces slightly better results.

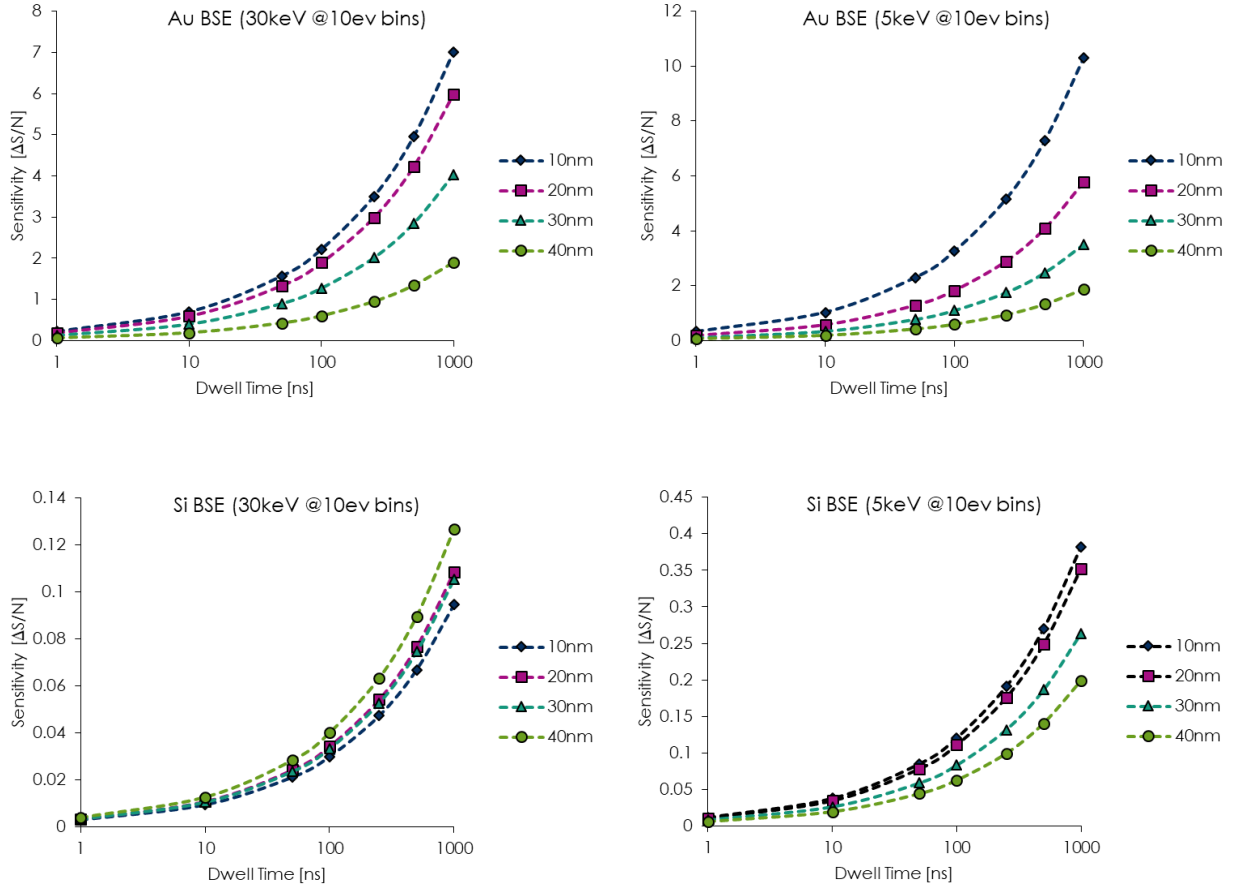


Figure B - 9 Sensitivity vs. dwell time (electron dosage @ 1nA) for buried Au/SiO₂ and Si/SiO₂ grating under SiO₂ overlayer using 10 eV bins.

Next, the binning size for the backscattered energy loss is increased from 10 eV to 20 eV so as to imitate an increase in window size for a backscattered electron detector. The effect of this improved sensitivity for both landing energies

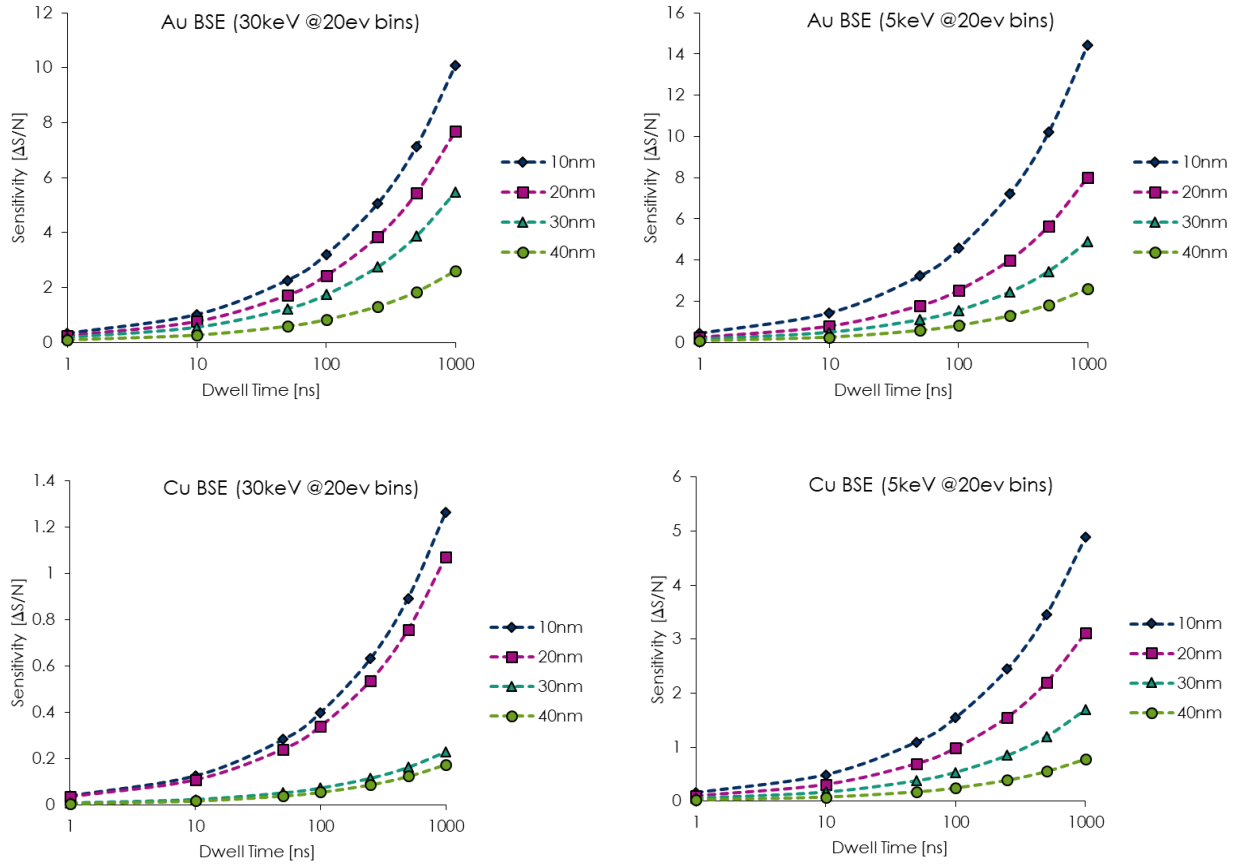


Figure B - 10 Sensitivity vs. dwell time (electron dosage @1nA) for buried Au/SiO₂ and Cu/SiO₂ grating under SiO₂ overlayer using 20 eV bins.

Finally, the binning size for the backscattered energy loss is increased from 10 eV to 50 eV and the process of calculating the sensitivity vs. dwell time is repeated. Depending on the material(s) in question, increasing the window size of the energy loss bins may be useful or they may be counterproductive. Considering Figure B - 12, where the BSE yield, contrast and sensitivity plots for the 50 eV binning case using 5 keV landing energy are shown, it is easily observed that the curves are being flattened out and further increase in the bin size may cause the original max contrast peak to be outside the window where the new max contrast lies. These

results show that there may be optimum energy loss windows along with imaging electron dosages for the detection of buried features for a given material set(s) use case.

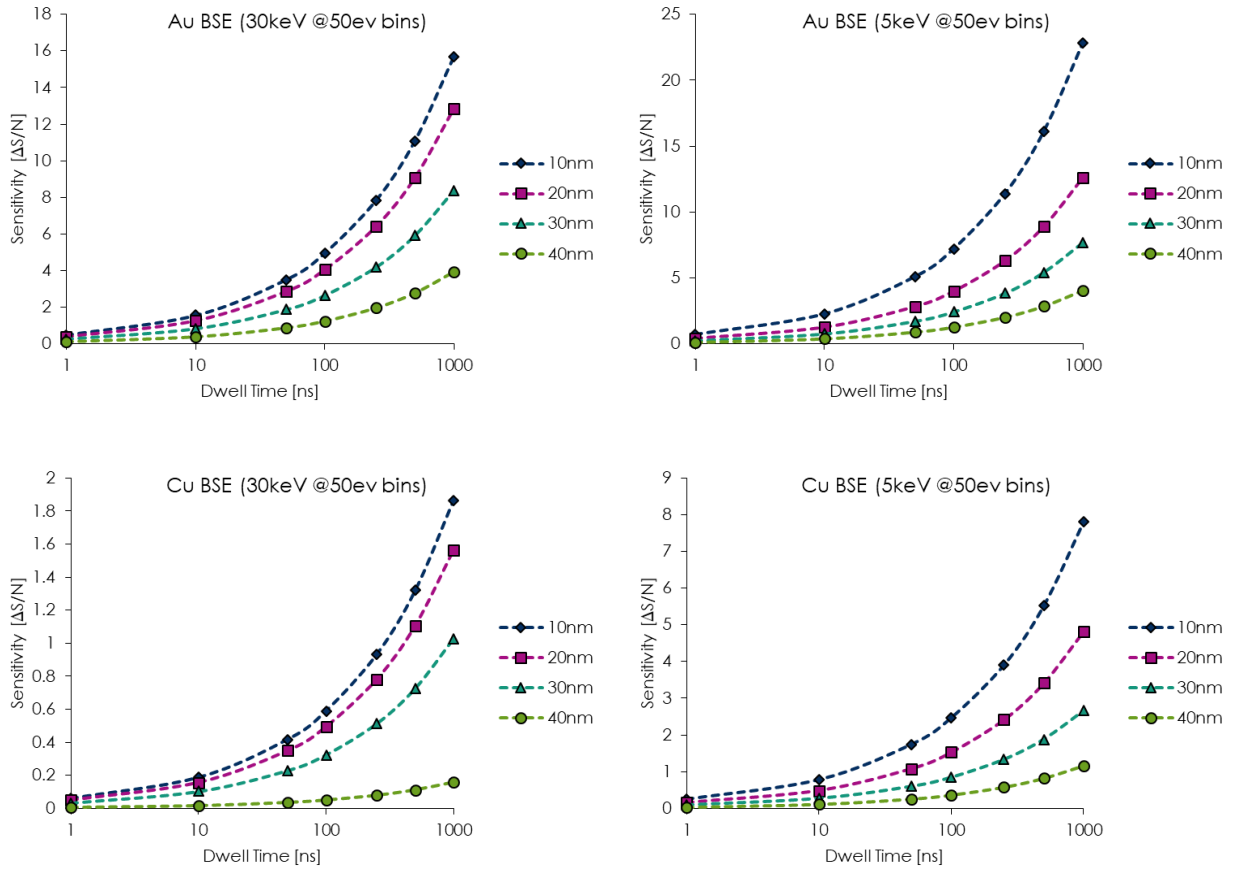


Figure B - 11 Sensitivity vs. dwell time (electron dosage @1nA) for buried Au/SiO₂ (top) and Cu/ SiO₂ (bottom) grating under SiO₂ overlayer using 50 eV bins.

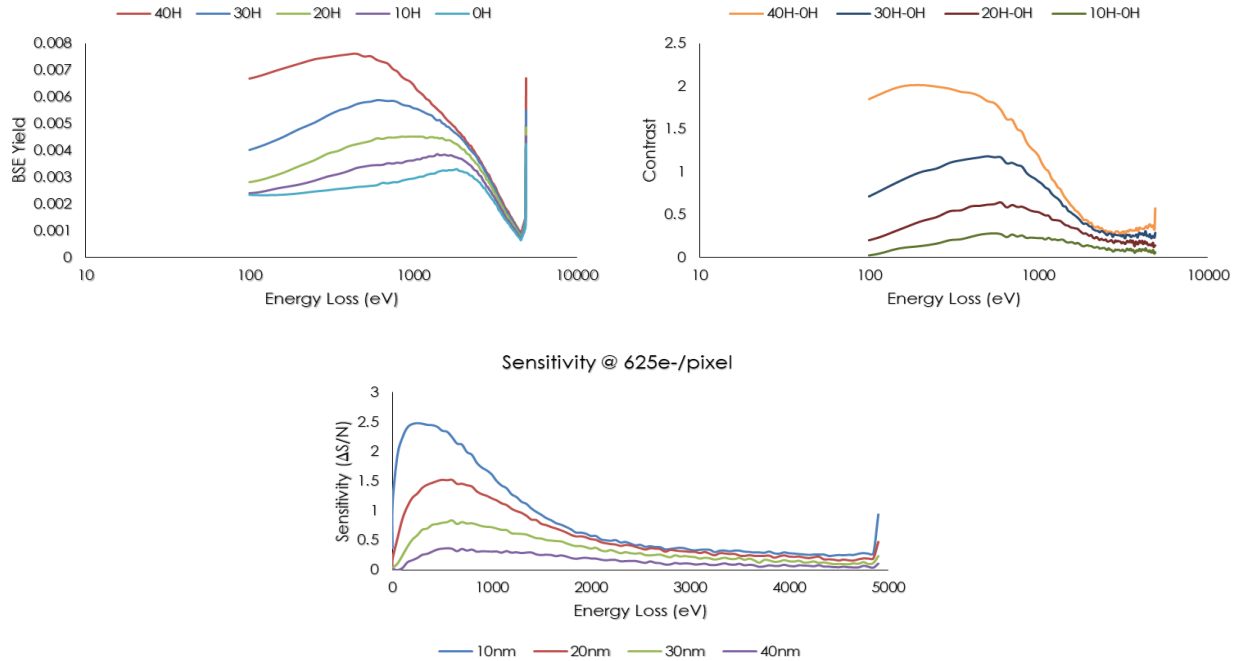


Figure B - 12 Graphs of backscattered energy loss, contrast, and sensitivity (electron dosage @ 1nA) for buried Cu/ SiO₂ grating under SiO₂ overlayer using 50 eV bin size and 5 keV landing energy.

Conclusions

Here, the use of formerly established infrastructure for the generation of realistic virtual images of test structures for prediction of performance of electron beam inspection tools was extended further to explore the measurement of overlays. Subsequent steps for this type of analysis would include simulation of image data by modulating the emission behaviors with dose-dependent shot noise and instrumental artifacts. As well, analyzing the resultant imaging data stream to assess the noise tolerance of the buried feature signature as a function of energy loss. Finally, formulation of recommendations on the viability of this technique.

References

- [1] Bunday B., Solecky E., Vaid A., Bello A.F., and Dai X., “Metrology capabilities and needs for 7nm and 5nm logic nodes,” in Proc. SPIE 10145, Metrology, Inspection, and Process Control for Microlithography XXXI, 2017, vol. 10145, p. 101450G.
- [2] Villarrubia, J. S. , Ritchie, N. W. M., and Lowney, J. R. “Monte Carlo modeling of secondary electron imaging in three dimensions,” Proc. SPIE 6518 (2007)
- [3] Villarrubia, J.S., Vladár, A.E., Ming, Bin, Kline, R.J., Sunday, D.F., Chawla, J.S., List, S. “Scanning electron microscope measurement of width and shape of 10nm patterned lines using a JMONSEL-modeled library,” Ultramicroscopy, vol. 154, no. July, pp. 15–28, Jul. 2015.
- [4] Mukhtar M. and Thiel B., “Image Simulation and Analysis to Predict the Sensitivity Performance of a Multi-Electron Beam Critical Dimension Metrology Tool”, Microsc. Microanal. 23 Suppl. 1 (2017)
- [5] Mukhtar M., Quoi, K., Bunday, B. D., Malloy, M., and Thiel, B., “Image Simulation and Analysis to Predict the Sensitivity Performance of a Multi-Electron Beam Wafer Defect Inspection Tool”, Microsc. Microanal. 22 Suppl. 3 (2016)
- [6] Mukhtar M., Bunday, B. D., Quoi, K., Malloy, M., and Thiel, B., "Measuring multielectron beam imaging fidelity with a signal-to-noise ratio analysis," J. Micro/Nanolith. MEMS MOEMS 15(3), 034004 (2016)

C. SIGNAL TO NOISE RATIO (SNR) PRODUCING SCRIPT

The following script was executed for determining the signal to noise. The script generates three different signal to noise ratios:

- Unfiltered lines vs. Background Noise
- Filtered lines vs. Background Noise
- Identified defect vs. Remaining Differential Image

The SNR value determined from taking the identified defect against the remaining differential image is what is reported in the body of this work.

```

1. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2. % Filename: defect_detection_w_comparison.m
3. %
4. % Main code by Francois Goasmat, (former) Guest Researcher at NIST
5. % edits by Bryan Barnes, NIST Staff
6. %      (301) 975-3947      bmbarnes@nist.gov
7. % subroutines for Fourier filtering by Martin Sohn, NIST staff
8. %
9. % Version 0.1 - code release to SEMATECH 7/22/2014
10. %      subsequent edits and enhancements by Maseeh Mukhtar, SUNY
    POLY
11. %      mmukhtar@sunypoly.edu
12.
13. % this code uses the ecc package from "MATLABcentral"
14. % Code available from
15. % http://www.mathworks.com/matlabcentral/fileexchange/27253-ecc-
    image-alignment-algorithm--image-registration-
16. % See README for that package for license information.
17. % That software implements the ECC image alignment algorithm as
    it is
18. %      presented in the paper
19. %      G.D.Evangelidis, E.Z.Psarakis, "Parametric Image
    Alignment
20. %      using Enhanced Correlation Coefficient", IEEE Trans. on
    PAMI,
21. %      vol.30, no.10, 2008"
22. %%%
23.
24. % The following script establishes the necessary environment for
    using these
25. % packages.
26.
27. % Change file paths to the location of the installation or add
    them
28. % permanently within MATLAB.
29.
30. install_dir = 'c:\Users\Admin\Desktop\SEMANalysis';
31.
32. addpath([install_dir '\ecc']); % or wherever installed
33. addpath([install_dir '\SEMANalysis']);
34. addpath([install_dir '\SEMANalysis\filtering_multi_dim']);
35. addpath([install_dir '\SEMANalysis\ecc']);
36. %%
37.
38. % Remember the starting folder
39. ccd = pwd;
40. % Suppress warnings about images being too large to display.
41. warning('off', 'Images:initSize:adjustingMag');
42. close all;
43. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%
44. % Input Dialog
45. prompt={'Folder','Print Out?','magnification','intensity defect
    thresh','Area min',...
46.      'side size(x)','side size(y)','high pass','low
    pass','nshift','x range',...

```



```

47.         'y range','No. of Iterations','Im2Bw','thresh'
48.     };
49.
50.     name = 'Input Parameters';
51.     numlines= 1;
52.
53.     defaultanswer={'C:\Users\Admin\Desktop\IBM_PDA_Hex_Defects\'',...
54.         'NO','13196','6','15','2','6','0.01','0.99','75','50','50','5','1','.3
55.         '};
56.
57.     options.Resize='on';
58.     options.WindowStyle='normal';
59.     options.Interpreter='tex';
60.     parameters =
        inputdlg(prompt,name,numlines,defaultanswer,options);
61.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%%
62.     % Define Image to be Processed Here
63.     % Also, Identify the characteristics of the defect to be sought.
64.
65.     %NOTE: % These parameters will change with defect type
66.     % Example 1:
67.     % cd('A_Defect');
68.     % image_name='A_100_2048_2_2.Tiff';
69.     % magnification=45722;
70.     % %defect isolation parameters (size filtering)
71.     % intensity_defect_thresh=6;
72.     % Area_min=15;
73.     % side_size=[2 6];
74.
75.     % Example 2:
76.     % cd('By_Defect');
77.     % image_name='BY_100_512_4_1.Tiff';
78.     % magnification=13196;
79.     % %defect isolation parameters (size filtering)
80.     % intensity_defect_thresh=6;
81.     % Area_min=15;
82.     % side_size=[2 6];
83.
84.     % Example 3:
85.     % cd('Bx_Defect');
86.     % image_name='BX_100_2048_2_2.Tiff';
87.     % magnification=22874;
88.     % %defect isolation parameters (size filtering)
89.     % intensity_defect_thresh=10;
90.     % Area_min=10;
91.     % side_size=[9 2];
92.
93.     % % Actual Inputs
94.     cd(parameters{1}); c2d=pwd;
95.     reference_mag = 13196; % Magnification of reference image (if all
        images are of same magnification simply keep same input numbers)
96.     magnification = str2double(parameters{3});
97.     intensity_defect_thresh = str2double(parameters{4});

```

```

98.     Area_min= str2double(parameters{5}); % Mostly using whole numbers
99.     side_size=[str2double(parameters{6}) str2double(parameters{7})];
100.     resize_scale = reference_mag/magnification;
101.
102.
103.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%
104.     % START CODE FOR IDENTIFYING DEFECT
105.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%
106.     %%% starting parameters not often changed
107.
108.     % Fourier filtering parameters
109.     high_pass=str2double(parameters{8}); %.01 or 0.1, etc.
110.     low_pass=str2double(parameters{9}); %.99 or .9, etc.
111.
112.     %convolution parameters
113.     im2bwlevel = str2double(parameters{14});
114.
115.     thresh= str2double(parameters{15}); % Grayscale image to binary
    image threshold
116.     nshift=str2double(parameters{10}); % Shifts have to be larger
    than image periodicity in x and in y
117.     x_range=str2double(parameters{11}); % at least equal to half the
    period in x
118.     y_range=str2double(parameters{12}); % at least equal to half the
    period in y
119.
120.     %ecc parameters
121.     NoI = str2num(parameters{13}); %No. of iterations
122.     NoL = 1; %No. of levels -- higher levels will flip images back to
    match original
123.
124.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%
125.     imageFolder = c2d; %Start of image collection from
    folder%
126.     if ~isdir(imageFolder)
127.         errorMessage = sprintf('Error: The following folder does not
    exist:\n%s', imageFolder);
128.         uiwait(warndlg(errorMessage));
129.         return;
130.     end
131.     filePattern = fullfile(imageFolder, '*.tif');
132.     imageFiles = dir(filePattern);
133.     N = length(imageFiles);
134.     Unfiltered = zeros(N,1);
135.     Filtered = zeros(N,1);
136.     Defect = zeros(N,1);
137.     ImgName = cell(N,1);
138.     Number_of_Defects=zeros(N,1);
139.
140.     for P = 1:N
141.         image_name = imageFiles(P).name;
142.         [A] = imread(image_name);
143.         baseFileName = imageFiles(P).name;

```

```

144.         fullFileName = fullfile(imageFolder, baseFileName);
145.         fprintf(1, 'Now reading %s\n', fullFileName);
146.
147.         figure(1);subplot(2,2,1); imshow(A);title(['Image as read in:',
            image_name]); axis square;
148.
149.         % Determine pixel where the legend begins -- if there is a legend
            (Comment out otherwise).
150.         % Images usually 512, 1024 or 2048 pixel wide images.
151.         % Alter to suit new images.
152.
153.         switch size(A,1)
154.             case 512
155.                 nolegend_index=512; %Simulated images contained no legend
156.             case 1024
157.                 nolegend_index=875;
158.             case 2048
159.                 nolegend_index=1900;
160.             otherwise
161.                 error('image size does not match legend assumptions');
162.         end
163.
164.         % % % Image w/o legend
165.         % Either use conversion of RGB conversion method (lines 166-167)
            or direct grayscale transform (line 168)
166.         % nolegend_hsv=rgb2hsv(A(1:nolegend_index,:,1:3));
167.         % nolegend_image_grayscale=nolegend_hsv(:,:,3);
168.         nolegend_image_grayscale=mat2gray(A); %Bypassing the RGB
            requirement if using single channel image without legend (Comment out
            lines 153-162)
169.         %
            nolegend_hsv=rgb2hsv(A);nolegend_image_grayscale=nolegend_hsv(:,:,3);
            %Bypass legend truncation
170.
171.         nolegend_image_grayscale=imresize(nolegend_image_grayscale,resize
            _scale);
172.         Size_of_image_RESIZED = size(nolegend_image_grayscale)
173.
174.         % % % Fourier filtering begins here
175.         ft_ref=fftn(nolegend_image_grayscale);
176.         ft_ref=fftshift(ft_ref);
177.         [low_filter]=FL_FilterPass_n(ft_ref, low_pass,'low'); %**From
            FL_FilterPass_n.m by Martin Sohn
178.         [high_filter]=FL_FilterPass_n(ft_ref, high_pass,'high');
179.         filter=low_filter.*high_filter;
180.         ft_ref_filtered=ft_ref.*filter;
181.         ft_ref_filtered=fftshift(ft_ref_filtered);
182.         filtered_ref=abs(ifftn(ft_ref_filtered));
183.
184.         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
185.         % Determine the center of the image
186.
187.         image_Center_bw=double(im2bw(filtered_ref(1+nshift:end-
            nshift,1+nshift:end-nshift),thresh)); % imshow(image_Center_bw);
188.         image_Center=filtered_ref(1+nshift:end-nshift,1+nshift:end-
            nshift); %imshow(image_Center);

```

```

189.     flip_image_Center_bw=flipud(fliplr(image_Center_bw));
    %imshow(image_Center_bw);
190.
191.
192.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
193.     % Our basic approach to defect detection is that one needs a
    "reference"
194.     % and a "target" containing the defect image. When only one
    image is
195.     % available, we need to make a "reference" from the "target".
    Here, we
196.     % make four images, representing a "North", "South," East"," and
    "West"
197.     % shift approach. First, the four image portions are extracted.
198.
199.     image_East=filtered_ref(1+2*nshift:end,1+nshift:end-nshift);
200.     image_East_bw=double(im2bw(image_East,thresh));
    %imshow(image_East_bw)
201.
202.     image_West=filtered_ref(1:end-(2*nshift),1+nshift:end-nshift);
203.     image_West_bw=double(im2bw(image_West,thresh));
    %imshow(image_West_bw)
204.
205.     image_North=filtered_ref(1+nshift:end-nshift,1+2*nshift:end);
206.     image_North_bw=double(im2bw(image_North,thresh));
    %imshow(image_North_bw)
207.
208.     image_South=filtered_ref(1+nshift:end-nshift,1:end-2*nshift);
209.     image_South_bw=double(im2bw(image_South,thresh));
    %imshow(image_South);
210.
211.     nx=1:size(image_Center,2);
212.     ny=1:size(image_Center,1);
213.
214.
215.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %
216.     % Next, the reference image is created by correlating in 2d the
    North,
217.     % South, East, and West images so that they can be averaged, thus
218.     % minimizing the effect in subtraction of the defect by a factor
    of four.
219.
220.     %% EAST
221.     %east alignment
222.     %convolution
223.     image_convolution=conv2(image_East_bw,flip_image_Center_bw);
    %imshow(image_convolution)
224.     convolution_center_index=floor((size(image_convolution)+1)/2);
225.     search_domain=image_convolution(convolution_center_index(1)-
    x_range: ...
226.     convolution_center_index(1)+x_range,convolution_center_index(2)- ...
    y_range:convolution_center_index(2)+y_range);
227.     optimum=find(search_domain==max(max(search_domain)));
228.     %if multiple optimum, take the one implying the least translation
229.

```

```

230.    optimum_positions=[floor((optimum-1)/size(search_domain,1))+1,
...
231.        rem(optimum-1,size(search_domain,1))+1];
232.    possible_shifts=optimum_positions- ...
233.        repmat(floor((size(search_domain)+1)/2),[size(optimum_positions,1)
1]);
234.    possible_shifts_norm=sum(possible_shifts'.^2);
235.    best_shift_index=
        possible_shifts_norm==min(possible_shifts_norm);
236.    shift=possible_shifts(best_shift_index,:);
237.
238.    translation_warp=eye(2,3);
239.    translation_warp(:,end)=shift;
240.
241.    %ecc using convolution results as a starting point
242.    figure;
243.    [results]=ecc(image_East, image_Center, NoL, NoI, 'affine',
        translation_warp);
244.    close;
245.    final_translation_warp=eye(2,3);
246.    final_translation_warp(:,end)=results(end).warp(:,end);
247.    shift_East=results(end).warp(:,end);
248.    final_translated_East = spatial_interp(double(image_East),
        final_translation_warp, 'linear', 'affine', nx, ny);
249.
250.    % figure(6);subplot
        (2,2,1);imshow(final_translated_East);title('East');axis square;
251.
252.    %% WEST
253.    %west alignment
254.    %convolution
255.    image_convolution=conv2(image_West_bw,flip_image_Center_bw);
256.    convolution_center_index=floor((size(image_convolution)+1)/2);
257.    search_domain=image_convolution(convolution_center_index(1)-
        x_range: ...
258.        convolution_center_index(1)+x_range,convolution_center_index(2)- ...
259.        y_range:convolution_center_index(2)+y_range);
260.    optimum=find(search_domain==max(max(search_domain)));
261.    %if multiple optimum, take the one implying the least translation
262.    optimum_positions=[floor((optimum-1)/size(search_domain,1))+1,
...
263.        rem(optimum-1,size(search_domain,1))+1];
264.    possible_shifts=optimum_positions- ...
265.        repmat(floor((size(search_domain)+1)/2),[size(optimum_positions,1)
1]);
266.    possible_shifts_norm=sum(possible_shifts'.^2);
267.    best_shift_index=find(possible_shifts_norm==min(possible_shifts_n
orm));
268.    shift=possible_shifts(best_shift_index,:);
269.
270.    translation_warp=eye(2,3);
271.    translation_warp(:,end)=shift;
272.

```

```

273. %ecc using convolution results as a starting point
274. figure;
275. [results]=ecc(image_West, image_Center, NoL, NoI, 'affine',
    translation_warp);
276. close;
277. final_translation_warp=eye(2,3);
278. final_translation_warp(:,end)=results(end).warp(:,end);
279. shift_West=results(end).warp(:,end);
280. final_translated_West = spatial_interp(double(image_West),
    final_translation_warp, 'linear', 'affine', nx, ny);
281.
282. % figure(6);subplot
    (2,2,2);imshow(final_translated_West);title('West');axis square;
283.
284. %% NORTH
285. %north alignment
286. %convolution
287. image_convolution=conv2(image_North_bw,flip_image_Center_bw);
288. convolution_center_index=floor((size(image_convolution)+1)/2);
289. search_domain=image_convolution(convolution_center_index(1)-
    x_range: ...
290.     convolution_center_index(1)+x_range,convolution_center_index(2)- ...
291.     y_range:convolution_center_index(2)+y_range);
292. optimum=find(search_domain==max(max(search_domain)));
293. %if multiple optimum, take the one implying the least translation
294. optimum_positions=[floor((optimum-1)/size(search_domain,1))+1,
    ...
295.     rem(optimum-1,size(search_domain,1))+1];
296. possible_shifts=optimum_positions- ...
297.     repmat(floor((size(search_domain)+1)/2),[size(optimum_positions,1)
    1]);
298. possible_shifts_norm=sum(possible_shifts'.^2);
299. best_shift_index=find(possible_shifts_norm==min(possible_shifts_n
    orm));
300. shift=possible_shifts(best_shift_index,:);
301.
302. translation_warp=eye(2,3);
303. translation_warp(:,end)=shift;
304.
305. %ecc using convolution results as a starting point
306. figure;
307. [results]=ecc(image_North, image_Center, NoL, NoI, 'affine',
    translation_warp);
308. close;
309. final_translation_warp=eye(2,3);
310. final_translation_warp(:,end)=results(end).warp(:,end);
311. shift_North=results(end).warp(:,end);
312. final_translated_North = spatial_interp(double(image_North),
    final_translation_warp, 'linear', 'affine', nx, ny);
313.
314. % figure(6);subplot
    (2,2,3);imshow(final_translated_East);title('North');axis square;
315.
316. %% SOUTH

```

```

317.     %south alignment
318.     %convolution
319.     image_convolution=conv2(image_South_bw,flip_image_Center_bw);
320.     convolution_center_index=floor((size(image_convolution)+1)/2);
321.     search_domain=image_convolution(convolution_center_index(1)-
        x_range:convolution_center_index(1)+x_range,convolution_center_index(2)
        )-y_range:convolution_center_index(2)+y_range);
322.     optimum=find(search_domain==max(max(search_domain)));
323.     % shift=[floor((optimum-1)/size(search_domain,1))+1,rem(optimum-
        1,size(search_domain,1))+1]-floor((size(search_domain)+1)/2);
324.     %if multiple optimum, take the one implying the least translation
325.     optimum_positions=[floor((optimum-
        1)/size(search_domain,1))+1,rem(optimum-1,size(search_domain,1))+1];
326.     possible_shifts=optimum_positions-
        repmat(floor((size(search_domain)+1)/2),[size(optimum_positions,1)
        1]);
327.     possible_shifts_norm=sum(possible_shifts'.^2);
328.     best_shift_index=find(possible_shifts_norm==min(possible_shifts_n
        orm));
329.     shift=possible_shifts(best_shift_index,:);
330.
331.     translation_warp=eye(2,3);
332.     translation_warp(:,end)=shift;
333.
334.     %ecc using convolution results as a starting point
335.     figure;
336.     [results]=ecc(image_South, image_Center, NoL, NoI, 'affine',
        translation_warp);
337.     close;
338.     final_translation_warp=eye(2,3);
339.     final_translation_warp(:,end)=results(end).warp(:,end);
340.     shift_South=results(end).warp(:,end);
341.     final_translated_South = spatial_interp(double(image_South),
        final_translation_warp, 'linear', 'affine', nx, ny);
342.
343.     % figure(6);subplot
        (2,2,4);imshow(final_translated_South);title('South');axis square;
344.
345.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
346.     % Now we isolate the final "Target Image" ...
347.     final_Center = double(image_Center);
348.
349.     % And the final differential image, which is the difference
        between the
350.     % "Target" and the "Reference" images
351.
352.     diff = abs((4*final_Center)...
353.         - final_translated_East...
354.         - final_translated_West...
355.         - final_translated_North...
356.         - final_translated_South);
357.
358.     % figure(10);
359.     % subplot(3,2,1);imshow(filtered_ref,[]); title('Image');
360.     % subplot(3,2,2);imshow(final_Center,[]); title('Center');
361.     % subplot(3,2,3);imshow(final_translated_East,[]); title('East');

```

```

362. % subplot(3,2,4);imshow(final_translated_West,[]); title('West');
363. % subplot(3,2,5);imshow(final_translated_North,[]);
    title('North');
364. % subplot(3,2,6);imshow(final_translated_South,[]);
    title('South');
365.
366.
367.
368. % Diff = abs((4*final_Center) - image_East - image_West -
    image_North - image_South); imshow(Diff); axis on;
369. % Diff = im2bw(Diff); imshow(Diff,im2bwlevel); axis on;
370.
371. % Figure(2) shows the differential image in absolute value
372. figure(1);subplot(2,2,2);imshow(final_Center,[]); title('Image
    Center in absolute value');axis on;
373. % figure(6);imshow(diff,[]);axis on; %final_Center
374.
375.
376. % For image subtraction, we allowed subpixel interpolation, but
    for further
377. % analysis we need to know how much the "Reference" was shifted.
378. ceil_shift_East=sign(shift_East).*(floor(abs(shift_East))+1);
379. ceil_shift_West=sign(shift_West).*(floor(abs(shift_West))+1);
380. ceil_shift_North=sign(shift_North).*(floor(abs(shift_North))+1);
381. ceil_shift_South=sign(shift_South).*(floor(abs(shift_South))+1);
382.
383. matrix_shift=[ceil_shift_East ceil_shift_West ceil_shift_North
    ceil_shift_South];
384. y_shift=[min(0, min(matrix_shift(1,:)) ) max(0,
    max(matrix_shift(1,:)) )];
385. x_shift=[min(0, min(matrix_shift(2,:)) ) max(0,
    max(matrix_shift(2,:)) )];
386.
387. % The amount of the shift has to be chopped off the edges of the
    image,
388. % else false positives would occur due to the nature of the
389. % "spatial_interp" function.
390. sizediff = size(diff);
391. cropped_diff=diff(max(1,1-
    x_shift(1)):min(sizediff(1),sizediff(1)-x_shift(2)),max(1,1-
    y_shift(1)):min(sizediff(2),sizediff(2)-y_shift(2)));
392. % Figure(3) - this is the active area ( compare w/ Figure (2)).
393. figure(1); subplot(2,2,3);imshow(cropped_diff,[]); title('Active
    area (compare w/ Fig 2)');axis on;
394.
395.
396. % mean(mean(cropped_diff)) or
    mean(mean(nolegend_image_grayscale))
397. intensity_normalization_factor = mean(mean(cropped_diff));
398.
399. %
    cropped_diff_bw=double(im2bw(cropped_diff,intensity_defect_thresh));
400. %trick to have a threshold greater than 1
401. cropped_diff_bw=double(im2bw(cropped_diff/(intensity_normalizatio
    n_factor*intensity_defect_thresh),im2bwlevel));
402.

```



```

403.    % Figure(4) - this is the binary mask that will be used to find
        interesting
404.    % areas in the figure.  im2bw and other functions below require
        the Image
405.    % Processing Toolbox.
406.    figure(4);imshow(cropped_diff_bw,[]);axis on;
407.    figure(1); subplot(2,2,4);
        imshow(cropped_diff_bw,[]);title('Binary Mask to find interesting
        areas');axis on;
408.
409.    % % % Now, we find defects based upon the parameters at the
        start.
410.    % The binary mask is inspected for its location (Centroid), its
        extents
411.    % (Bounding Box), its area, and the pixel # of the ID'd defect
412.    % (PixelIdxList).
413.
414.    [labeled,numObjects] = bwlabeln(cropped_diff_bw,8);
415.    l =
        regionprops(labeled,'Centroid','Area','PixelIdxList','BoundingBox');
416.    n=[];
417.    dl = length(l);
418.
419.    for q=1:dl
420.        n(q,1)=l(q).Area;
421.        n(q,2)=sum(l(q).PixelIdxList);
422.        n(q,3)=q;
423.
424.        n(q,4)=l(q).BoundingBox(3);
425.        n(q,5)=l(q).BoundingBox(4);
426.        n(q,6)=l(q).Centroid(1);
427.        n(q,7)=l(q).Centroid(2);
428.    end;
429.
430.    n= sortrows(n,1);
431.
432.
433.    % % % This is the test to discriminate defects
434.    % We need minimum extents in x and y as well as a minimum area.
435.    matching_region=find( n(:,4)>=side_size(1) & n(:,5)>=side_size(2)
        & n(:,1)>=Area_min );
436.    gm=l(matching_region);
437.    nn=n(matching_region,:);
438.
439.    % If there is one defect crossing the threshold, we will proceed
        to analyze
440.    % it.  If there are none, the user is informed and the program
        stops.
441.    % If there are multiple, only the defect with the largest area is
        used and
442.    % the user is informed of the multiple defects.
443.
444.    nnl = size(nn);
445.    switch nnl(1)
446.        case 0

```

```

447.         disp(' ');disp(['No defects found for ' image_name '
    using min_x: ' ...
448.         num2str(side_size(1)) ', min_y: '
    num2str(side_size(2)) ...
449.         ', and area: '      num2str(Area_min) '. Moving
    On.']);cd(c2d); %set back to pwd
450.     %      return; %Stops from breaking the continuation of loop
451.     Coordinates = cat(2,0,0);
452.         case 1
453.         disp(' ');disp(['Defect found for ' image_name ' using
    min_x: ' ...
454.         num2str(side_size(1)) ', min_y: '
    num2str(side_size(2)) ...
455.         ', and area: '      num2str(Area_min) '.']);
456.         disp(['Using defect at ' num2str(nn(1,6)) ', '
    num2str(nn(1,7)) '.']);
457.         disp(' (red point on Figure 4)');
458.         figure(4);axis on; hold
    on;plot(nn(1,6),nn(1,7),'r*','MarkerSize',10);title(['Single/No
    Defect: ',image_name]);%Adjust Marker
459.     Coordinates = cat(2,nn(1,6), nn(1,7));
460.         otherwise
461.         disp(' ');disp(['Multiple defects found for ' image_name
    ' using min_x: ' ...
462.         num2str(side_size(1)) ', min_y: '
    num2str(side_size(2)) ...
463.         ', and area: '      num2str(Area_min) '.']);
464.         disp(['All defects will be used. Adjust parameters to
    reduce sensitivity.']);
465.         disp([' (red points on Figure 4)']);
466.         figure(4); hold on; % Put all the multiple defects on one
    plot in fig4
467.         str = ' : Multiple Defect Locations - ';
468.         for i = 1:nnl(1)
469.             plot(nn(i,6),nn(i,7),'r*','MarkerSize',10);
    title([num2str(nnl(1)), str, image_name]); axis on;
470.             hold on;
471.         x(i) = nn(i,6); y(i) = nn(i,7);
472.         Coordinates = cat(2,x,y);
473.         end
474.
475.     % Coordinates = cat(2, image_name, num2str(Coordinates))
476.     end
477.
478.
479.
480.     cd(c2d);
481.
482.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
483.     % START IMAGE ANALYSIS
484.     %
485.     % The objective in the following section is to:
486.     %     Define signals for signal to noise ratios.
487.     %     Unfiltered lines vs. background noise
488.     %     Filtered lines vs. background noise

```

```

489.      %      Identified defect vs. remaining differential image
490.
491.      %
492.      % These are SNR_unfiltered, SNR_filtered, and SNR_defect.  These
      numbers
493.      % were shown in a previous report (Winter 2014).
494.
495.      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
496.      % % nolegend_image_grayscale
497.      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
498.
499.      % %reverse intensity for white and black images, instead of black
      and white
500.      % % nolegend_image_grayscale=1-nolegend_image_grayscale;
501.      %
502.      %
503.      %%% NOTE: While fairly comfortable with this analysis, there are
      major
504.      %%% questions about this one particular section, as some images
      need
505.      %%% filtering so badly that this simple approach does not work.
506.      %%% BECAUSE it will fail for BX_100_2048_2_2.Tiff, I will use a
      try-catch
507.      %%% approach.
508.
509.
510.      try
511.          %This effectively sets the threshold at 2* mean of the image.
512.
513.          nolegend_image_grayscale_intensity_thresh = 2;
514.
515.          nolegend_image_grayscale_Area_min = 5*resize_scale;
516.
517.          intensity_normalizatin_factor=mean(mean(nolegend_image_grayscale));
518.
519.          nolegend_image_grayscale_bw=double(im2bw(...
520.              nolegend_image_grayscale /...
521.              (intensity_normalizatin_factor*nolegend_image_grayscale_intensity_thre
      sh),im2bwlevel));
522.
523.
524.          figure(5);subplot(2,3,1);imshow(nolegend_image_grayscale_bw,[]);
525.          title('unfiltered image binary mask');axis on;
526.
527.          [labeled,numObjects] =
      bwlabeln(nolegend_image_grayscale_bw,8);
528.          l = regionprops(labeled,'area','PixelIdxList');
529.          n=[];
530.          dl = length(l);
531.          for q=1:dl
532.              n(q,1)=l(q).Area;

```

```

533.         end;
534.
535.         nolegend_image_grayscale_matching_region=find(n(:,1)>=nolegend_image_g
            rayscale_Area_min);
536.         nolegend_image_grayscale_gm=1(nolegend_image_grayscale_matching_region
            );
537.
538.         final_Pixel_List=[];
539.         for k=1:size(nolegend_image_grayscale_gm,1)
540.             final_Pixel_List=[final_Pixel_List
                nolegend_image_grayscale_gm(k).PixelIdxList'];
541.         end
542.
543.         signal_pixels_vector=nolegend_image_grayscale(final_Pixel_List);
544.         noise_pixels_vector=nolegend_image_grayscale;
545.         noise_pixels_vector(final_Pixel_List)=[];
546.
547.         nolegend_image_grayscale_signal_value=mean(signal_pixels_vector)-
            mean(noise_pixels_vector);
548.
549.         nolegend_image_grayscale_noise_value=std(noise_pixels_vector);
550.
551.         SNR_unfiltered=nolegend_image_grayscale_signal_to_noise_ratio;
552.         catch
553.             SNR_unfiltered=NaN;
554.         end
555.         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
556.         % % filtered_ref
557.         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
558.
559.         filtered_ref_intensity_thresh=2;
560.         filtered_ref_Area_min = 5 * resize_scale;
561.         intensity_normalizatin_factor=mean(mean(filtered_ref));
562.
563.         filtered_ref_bw=double(im2bw(...
564.             filtered_ref /
565.             (intensity_normalizatin_factor*filtered_ref_intensity_thresh),im2bwlev
566.             el));
567.
568.         figure(5);subplot(2,3,4);imshow(filtered_ref_bw,[]);
569.         title('filtered image binary mask');axis on;
570.
571.

```

```

572.     [labeled,numObjects] = bwlabeln(filtered_ref_bw,8);
573.     l = regionprops(labeled,'area','PixelIdxList');
574.     n=[];
575.     dl = length(l);
576.
577.     for q=1:dl
578.         n(q,1)=l(q).Area;
579.     end;
580.
581.     filtered_ref_matching_region=find(n(:,1)>=filtered_ref_Area_min);
582.     filtered_ref_gm=l(filtered_ref_matching_region);
583.
584.     final_Pixel_List=[];
585.     for k=1:size(filtered_ref_gm,1)
586.         final_Pixel_List=[final_Pixel_List
587.             filtered_ref_gm(k).PixelIdxList'];
588.     end
589.
590.
591.     signal_pixels_vector=filtered_ref(final_Pixel_List);
592.     noise_pixels_vector=filtered_ref;noise_pixels_vector(final_Pixel_
593.         List)=[];
594.
595.     filtered_ref_signal_value=mean(signal_pixels_vector)-
596.         mean(noise_pixels_vector);
597.     filtered_ref_noise_value=std(noise_pixels_vector);
598.
599.     filtered_ref_signal_to_noise_ration=filtered_ref_signal_value/fil
600.         tered_ref_noise_value;
601.     SNR_filtered=filtered_ref_signal_to_noise_ration;
602.
603.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
604.     % % cropped_diff
605.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
606.
607.     %defect detection
608.     intensity_normalizatin_factor=mean2(cropped_diff);
609.
610.     cropped_diff_bw=double(im2bw(cropped_diff / ...
611.         (intensity_normalizatin_factor*intensity_defect_thresh),im2bwlevel));
612.
613.     figure(5);subplot(2,3,[2 3 5 6]);imshow(cropped_diff_bw,[])
614.     title('cropped difference image binary mask'); axis on;
615.
616.     [labeled,numObjects] = bwlabeln(cropped_diff_bw,8);
617.     l =
618.         regionprops(labeled,'Centroid','area','PixelIdxList','BoundingBox');
619.     n=[];
620.     dl = length(l);

```

```

620.     for q=1:dl
621.         n(q,1)=l(q).Area;
622.         n(q,2)=sum(l(q).PixelIdxList);
623.         n(q,3)=q;
624.
625.         n(q,4)=l(q).BoundingBox(3);
626.         n(q,5)=l(q).BoundingBox(4);
627.     end;
628.
629.     matching_region=find( n(:,4)>=side_size(1) & n(:,5)>=side_size(2)
        & n(:,1)>=Area_min );
630.     cropped_diff_gm=l(matching_region);
631.
632.
633.     % Multiple defects would contribute.
634.     final_Pixel_List=[];
635.     for k=1:size(cropped_diff_gm,1)
636.         final_Pixel_List=[final_Pixel_List
            cropped_diff_gm(k).PixelIdxList'];
637.     end
638.
639.
640.     signal_pixels_vector=cropped_diff(final_Pixel_List);
641.     noise_pixels_vector=cropped_diff;
642.     noise_pixels_vector(final_Pixel_List)=[];
643.
644.     cropped_diff_signal_value=mean(signal_pixels_vector)-
        mean(noise_pixels_vector);
645.     cropped_diff_noise_value=std(noise_pixels_vector);
646.
647.     cropped_diff_signal_to_noise_ratio=cropped_diff_signal_value/cro
        pped_diff_noise_value;
648.     SNR_defect=cropped_diff_signal_to_noise_ratio;
649.
650.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
651.     disp(' ');disp(['SNR_unfiltered: '
        num2str(SNR_unfiltered,'%02.1f')]);
652.     disp(['SNR_filtered: '      num2str(SNR_filtered,'%02.1f')]);
653.     disp(['SNR_defect: '       num2str(SNR_defect,'%02.1f')]);
654.
655.     % Collect Outputs
656.     Unfiltered(P,1) = SNR_unfiltered;
657.     Filtered(P,1) = SNR_filtered;
658.     Defect(P,1) = SNR_defect;
659.     ImgName{P} = image_name;
660.     Number_of_Defects(P,1) = nnl(1)
661.     RefMag(P,1) = reference_mag;
662.     Mag(P,1) = magnification;
663.     ImgLength_Y(P,1) =
        Size_of_image_RESIZED(1);ImgLength_X(P,1)=Size_of_image_RESIZED(2);
664.     Intensity_Defect_Threshold(P,1) = intensity_defect_thresh;
665.     Min_Area(P,1) = Area_min;
666.     Size_X(P,1) = side_size(1); Size_Y(P,1) = side_size(2);
667.     Rescale(P,1) =resize_scale;
668.     High(P,1) =high_pass;

```

```

669.     Low(P,1) =low_pass;
670.     Threshold(P,1) =thresh;
671.     Nshift(P,1) =nshift;
672.     Xrange(P,1) =x_range;
673.     Yrange(P,1) =y_range;
674.     Iterations(P,1) = NoI;
675.     Image2BW(P,1) = im2bwlevel;
676.
677.     end
678.
679.
680.     % Output dataset
681.     Output =
        dataset(ImgName,Unfiltered,Filtered,Defect,Number_of_Defects,...
682.         RefMag,Mag,ImgLength_X,ImgLength_Y,Intensity_Defect_Threshold,
        Min_Area,Size_X,Size_Y,Rescale, ...
683.         High,Low,Threshold,Image2BW,Nshift,Xrange,Yrange,Iterations);
684.
685.     % Print Out
686.     switch parameters{2}
687.     case 'NO'
688.         figure (6)
689.         Plot = bar(Defect);title('Defect SNRs');
690.         xlabel('Defect Types'); ylabel('Signal to Noise Ratio');
691.
        set(gca,'XTickLabel',ImgName,'FontWeight','bold','FontSize',5);
692.
693.     case 'YES'
694.         % export(DS,'XLSfile',filename)
695.         Output_filename = 'Defect SNRs_Gratings_ByDose';
696.         export(Output,'XLSfile',Output_filename);
697.
698.         figure (6)
699.         Plot = bar(Defect);title('Defect SNRs');
700.         xlabel('Defect Types'); ylabel('Signal to Noise Ratio');
701.
        set(gca,'XTickLabel',ImgName,'FontWeight','bold','FontSize',5);
702.         % saveas(Plot, Output_filename, image extension)
703.         saveas(Plot,Output_filename,'tif');
704.
705.     end

```

D. JMONSEL SIMULATION SCRIPT

The following example script, after execution, would produce a 32×32 pixel secondary and backscatter electron emission profile which can then be transformed into a 32×32 pixel image as below:

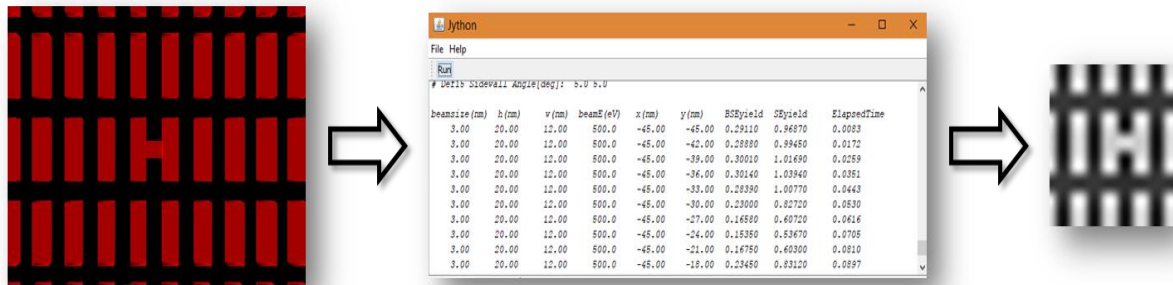


Figure D - 1 Left to Right: Illustrated flow of simulation from 3D image construction to secondary and backscatter electron emission calculation to generation of 32×32 pixel grayscale image of the simulated area.


```

1. # Copyright: Pursuant to title 17 Section 105 of the United States
   Code this software is not subject to copyright protection and is in
   the public domain.
2.
3. # Institution: National Institute of Standards and Technology
4.
5. # Foundational script authored by: John Villarrubia, July 23, 2010
6. # Subsequent additions and amendments by: Benjamin Bunday & Maseeh
   Mukhtar
7.
8.
9. # This script determines the electron yields incident on 1 or more
   trapezoidal lines on a 3-layer substrate with a bridge defect between
   the two central lines.
10.
11.
12.     WinOrLin = 1      #=1 work PC, =2 for LINUX, =3 for
   WINDOWS/Metrosim group sim computer, =4 for WINDOWS/Prolith2 computer
13.     if (WinOrLin == 2):
14.         import sys
15.         files = [file.strip() for file in open( "./jar.list", "r" )]
16.         for file in files:
17.             sys.path.append(file)
18.
19.         import gov.nist.microanalysis.EPQLibrary as epq
20.         import gov.nist.microanalysis.EPQTools as ept
21.         import gov.nist.microanalysis.NISTMonte as nm
22.         import gov.nist.nanoscalemetrology.JMONSEL as mon
23.         import gov.nist.microanalysis.Utility as nmu
24.         import java.io as jio
25.         import java.util as jutil
26.         import java.lang as jl
27.         import jarray
28.         import java.nio.charset as cs
29.
30.         # Location where results will be saved
31.         dest=DefaultOutput;
32.         jio.File(dest).mkdirs()
33.         filename = DefaultOutput+PathSep+"Results.txt"
34.         file = open(filename,'w')
35.         print "Output will be to file: ",filename
36.
37.         if (WinOrLin == 1):
38.             tablePathOuter = "C:\\NIST\\JMONSEL\\ScatteringTables"
   # for workPC
39.         if (WinOrLin == 2):
40.             tablePathOuter = "/home/X/NIST/JMONSEL/ScatteringTables"
   # for LINUX PC
41.         if (WinOrLin == 3):
42.             tablePathOuter = "D:\\Program
   Files\\NIST\\JMONSEL\\ScatteringTables"    # for group sim computer
43.         if (WinOrLin == 4):
44.             tablePathOuter = "C:\\Users\\prolith2\\Desktop\\NIST\\JMONSEL"
   # for Prolith2
45.
46.         # Model parameters
47.         nTrajectories = 100000

```

```

48.
49.     # Shape parameters.
50.     pitchnm = 24.    #Distance between line centers in nm
51.     nlines = 12      #Number of lines [for doe, 1 is iso, 7 is dense]
52.     RotAngDeg = 0.   #Rotation {tilt} Angle of Grating (in degrees)
53.     ScanOrigX = -45.
54.     ScanOrigY = -45.
55.     ScanStepX = 3.
56.     ScanStepY = 3.
57.     NumPixX = 32
58.     NumPixY = 32
59.     linklengthnm = 36.
60.     linkspacenm = 12.
61.     hnmvals = [20.]    #Resist line height in nm
62.     wnmvals = [12.]    #Resist line bottom width in nm
63.     linelengthnm = 4000 #Resist line length in nm
64.
65.     # Note that sidewall angles (SWA) are specified with respect to
        vertical i.e., 0.0 is vertical.
66.     # Positive angles make bottom wider [Reverse for Negative
        angles].
67.     SWAvals = [0.]     #Can loop thru SWA with multiple values if
        SWAdecision=1, in this case these values overwrite values in both
        thetardeg and thetaldeg and make them equal.
68.     # Set to only one value and have SWAdecision=0 if using set
        thetardeg and thetaldeg.
69.     SWAdecision = 0    #If =1, then both right and left vals will
        loop thru SWAvals. if =0 then thetardeg and thetaldeg carry thru.
70.     thetardeg = 0.0    #Resist line right sidewall angle in degrees
71.     thetaldeg = 0.0    #Resist line left sidewall angle in degrees
72.
73.     radrnm = 0.000001    #Resist line top right corner radius in
        nm
74.     radlnm = 0.000001    #Resist line top left corner radius in nm
75.
76.     layer1thicknessnm = 60. # Thickness in nm of the 1st layer
        (immediately below the lines)
77.     layer2thicknessnm = 150. # Thickness in nm of the 2nd layer
78.     # Substrate will be made infinitely thick.
79.
80.     beamEeVvals = [500.]    # Beam energies in eV
81.     beamsizenmvals = [3.0]  # Beam size in nm
82.
83.     DefectType = 3          # Choose defect type
84.     #0=No Defect, 1=type A defect, 2=type Bx defect, 3=type Bx
        defect, 4=type Bx2 defect at link end, 5=line extension, 6=sidewall
        bump, 7=roughened link, 8=misaligned link, 9=CD variation&misalign,
        10=missing link, 11=mid-link gap, 12=mouse-bite, 13=shortened link,
        14=top rounding, 15=sidewall angle
85.
86.     CylDefDiam = 6.         #DefectType=1
87.     CylDefHeight= 6.
88.     BridgeDefWidth= 10.     #DefectType=2/3/4
89.     BridgeDefHeight=20.     #DefectType=2/3/4/5/6/7/8/9/12/13
90.     LineExtDefLength=3.     #DefectType=5
91.     LinkShorten=3.          #DefectType=13

```

```

92.     SidewallBumpWidth=2.      #DefectType=6, in x
93.     SidewallBumpLength=4.    #DefectType=6, in y
94.
95.     LERTabAmplitude=0.        #DefectType=7
96.     LERTabLength=2.          #DefectType=7
97.     LERTabPitch=4.           #DefectType=7
98.
99.     LinkMisAlign=2.           #Center Link MisAlignment, goes with
    DefectType=8 and also DefectType=9
100.    CDvarW=5.                 #input width of center link if different
    from standard link width, goes with DefectType=9
101.    CDvarH=6.                 #input height of center link if different
    from standard link height, goes with DefectType=9
102.    CDvarL=6.                 #input length of center link if different
    from standard link length, goes with DefectType=9
103.
104.    MouseBiteWidth=2.          #in x, amplitude of mousebite,
    DefectType=12
105.    MouseBiteLength=5.         #in y, how long the mousebite is along
    link, DefectType=12
106.
107.    MidLinkGap=2.              #DefectType=11
108.
109.    TopRadRnm = 3.              #resist line top right corner radius in
    nm
110.    TopRadLnm = 3.              #resist line top left corner radius in nm
111.
112.    LinkThetaRdeg = 5.          #resist line right sidewall angle in
    degrees
113.    LinkThetaLdeg = 5.          #resist line left sidewall angle in
    degrees
114.
115.    # Make a record of the random seed we use so we can exactly
    repeat this calculation
116.    # (same random number sequence) if necessary.
117.    seed = nmu.Math2.rgen.nextLong() # Pick a random seed
118.
119.    # To exactly repeat a previous calculation (e.g., for bug fix)
    uncomment the next line
120.    # and replace the question marks with the seed that was recorded
    in the previous calculation's
121.    # output file.
122.    #seed = -2769499132481846261L
123.    print >>file,"Random number seed: ",seed
124.    print "Random number seed: ",seed
125.    nmu.Math2.initializeRandom(seed)
126.    for i in range(0,10):
127.        r = nmu.Math2.rgen.nextDouble()
128.        print >>file, r
129.        print r
130.
131.    # The following parameter needs a bit of explaining. In the model
    we'll build below, the infinitely deep
132.    # layer 3 will be artificially divided into two parts, a skin
    layer that is close to the surface and a

```

```

133.    # deep part that is farther from the surface. The definition of
    # "deep" is set by
134.    # the deepnm parameter on the next line. Both the skin region and
    # the deep region will contain the same
135.    # material (Si), but in the deep region we'll make a model in
    # which electrons with energies less than
136.    # 50 eV are dropped from the simulation. This can save lots of
    # time (particularly if beam energies are large)
137.    # because there are lots of secondary electrons with energies <
    # 50 eV, and lots of simulation time
138.    # must be devoted to tracking them. We can't afford to drop them
    # when they are generated near the surface,
139.    # because they might escape and be detected. I.e., they're
    # important there. However, low energy electrons
140.    # that are deep inside the sample can't escape, so there is no
    # harm done in not tracking them. Thus, the
141.    # parameter below should be set to several times the typical
    # escape depth (so there's little chance of
142.    # dropping an electron that would have escaped). This is only
    # important for high beam energies, because
143.    # only then will the electrons have sufficient range to reach the
    # deeper layer, but in such cases there
144.    # can be a significant time savings.
145.    deepnm = 15. # Depth below which to use the "deep model."
146.
147.    trajImg = 0
148.    trajImgMaxTraj = 500
149.    trajImgSize = 150.e-9
150.
151.    VRML = 1
152.    VRMLImgMaxTraj = 0 # Include no trajectories in VRML (Show sample
    # only.) Leaving trajectories
153.    # out makes a VRML that displays easily. It's good for checking
    # the sample. I find that adding
154.    # trajectories significantly slows down the display. If you want
    # to try it, keep the number of
155.    # displayed trajectories small (20 is a reasonable number) and
    # turn off collision detection in
156.    # your VRML viewer.
157.
158.
159.    # Make materials
160.
161.    # A Secondary Electron vacuum
162.    vacuum = mon.SEmaterial()
163.    vacuum.setName("SE vacuum")
164.    vacuumBarrier = mon.ExpQMBBarrierSM(vacuum)
165.    vacuumMSM = mon.MONSEL_MaterialScatterModel(vacuum)
166.    vacuumMSM.setBarrierSM(vacuumBarrier)
167.
168.    # PMMA
169.    # Scattering tables to use the DFT model with PMMA are not yet
    # available.
170.    # Instead the code below specifies a back-up model based on the
    FittedInelSM

```

```

171.     # and GanachaudMokraniPolaronTrapSM classes. Each of these
172.     # classes has two
173.     # free parameters. The parameters given below were chosen to
174.     # provide a good
175.     # fit to measured yield vs. energy on PMMA. There is no guarantee
176.     # that these
177.     # parameters will also provide a good fit to topographic yield
178.     # (yield vs. angle)
179.     # since the data employed for the fit were all acquired at normal
180.     # incidence.
181.     breakE = epq.ToSI.eV(45.)
182.     density = 1190.
183.     workfun = 5.5
184.     bandgap = 5. # width of band gap in eV, based on TPP. Svorcik,
185.     Lyutakov, Huttel get about 5.3
186.     EFermi = -bandgap # This puts the Fermi level at the top of the
187.     valence band.
188.     potU = -workfun-EFermi # Gives 0.5 for this material for now.
189.     This is based on Sayyah et al.,
190.     # Int. J. Polymeric Mat. 54, p 505 (2005). It's about mid-range
191.     # for the values they report for
192.     # different forms of PMMA.
193.     # Material defined in terms of its constituent elements and their
194.     # weight fractions
195.     # Elemental Constituents
196.     C = epq.Element.C
197.     Ox = epq.Element.O
198.     H = epq.Element.H
199.     PMMAcomp = epq.Composition()
200.     PMMAcomp.defineByMoleFraction([C,Ox,H],[5,2,8])
201.     PMMA = mon.SEmaterial(PMMAcomp,density)
202.     PMMA.setName("PMMA")
203.     PMMAWorkfunction=epq.ToSI.eV(workfun)
204.     PMMA.setWorkfunction(PMMAWorkfunction)
205.     PMMA.setBandgap(epq.ToSI.eV(bandgap))
206.     PMMA.setEnergyCBbottom(epq.ToSI.eV(potU))
207.     # Create scatter mechanisms
208.     PMMANISTMott =
209.     mon.SelectableElasticSM(PMMA,mon.NISTMottRS.Factory)
210.     PMMACSD = mon.JoyLuoNieminenCSD(PMMA,breakE)
211.     PMMAfittedInel = mon.FittedInelSM(PMMA,epq.ToSI.eV(65.4),PMMACSD)
212.     # Parameters in the next line are from my fits. (See
213.     PMMAOptimization.nb)
214.     PMMApolaron =
215.     mon.GanachaudMokraniPolaronTrapSM(2.e7,1./epq.ToSI.eV(4.))
216.     #PMMAClassicalBarrier = mon.ExpQMBBarrierSM(PMMA)
217.     # Make a material scatter model
218.     # to be used in thin layer
219.     PMMAMSM = mon.MONSEL_MaterialScatterModel(PMMA)
220.     PMMAMSM.addScatterMechanism(PMMANISTMott)
221.     PMMAMSM.addScatterMechanism(PMMAfittedInel)
222.     PMMAMSM.addScatterMechanism(PMMApolaron)
223.     PMMAMSM.setCSD(PMMACSD)

```

```

214. #PMMAMSM.setBarrierSM(PMMAClassicalBarrier)
215. # MSM to be used deep inside (drops electrons with E<50 eV)
216. PMMAMSMDeep = mon.MONSEL_MaterialScatterModel(PMMA)
217. PMMAMSMDeep.addScatterMechanism(PMMANISTMott)
218. PMMAMSMDeep.addScatterMechanism(PMMAfittedInel)
219. PMMAMSMDeep.addScatterMechanism(PMMApolaron)
220. PMMAMSMDeep.setCSD(PMMACSD)
221. #PMMAMSMDeep.setBarrierSM(PMMAClassicalBarrier)
222. PMMAMSMDeep.setMinEforTracking(epq.ToSI.eV(50.))
223.
224. # TODO: Generate an ARC model.
225.
226. # BEGIN TEMPORARY
227. # Replace the following lines with an ARC model when available.
228. # I'm replacing the ARC with PMMA during this test phase.
229. ARCMSM = PMMAMSM
230. # END TEMPORARY
231.
232.
233. # Si
234. phononE = 0.063 # I've seen the number reported as 510 cm^-1.
    this is conversion of that to eV.
235. phononStrength = 3. # Turner & Inkson dispersion curves appear to
    show 3 LO phonon modes converging to the same
236. # energy at the Gamma point.
237. density = 2330.
238. workfun = 4.85
239. bandgap = 1.1 # width of band gap in eV
240. EFermi = -bandgap # This puts the Fermi level at the top of the
    valence band.
241. potU = -workfun-EFermi
242. Si = mon.SEmaterial([epq.Element.Si],[1.],density,"Silicon")
243. SiWorkfunction=epq.ToSI.eV(workfun)
244. Si.setWorkfunction(SiWorkfunction)
245. Si.setEnergyCBbottom(epq.ToSI.eV(potU))
246. Si.setBandgap(epq.ToSI.eV(bandgap))
247. Si.setCoreEnergy([epq.ToSI.eV(99.2),epq.ToSI.eV(99.8),epq.ToSI.eV
    (149.7),epq.ToSI.eV(1839.)])
248. # Edit the string below so it is the path to the folder where you
    have stored the silicon scattering tables
249. # that I provide
250. if (WinOrLin<>2):
251.     tablePath = tablePathOuter+"\SiTables"+PathSep
252.     #If Windows:         tablePath =
    tablePathOuter+"\SiTables"+PathSep
253.     #If LINUX:          tablePath =
    tablePathOuter+PathSep+"SiTables"+PathSep #for
    LINUX, change all \ in all lines to "PathSep+"
254. if (WinOrLin==2):
255.     tablePath = tablePathOuter+PathSep+"SiTables"+PathSep
256. SiTables = [tablePath + "IIMFPPFullPennInterpSiSI.csv",
257. tablePath + "interpNUSimReducedDeltaEFullPennSiSI.csv",
258. tablePath + "interpNUThetaFullPennSiBGSI.csv",
259. tablePath + "interpSimESEONUSiBGSI.csv"]
260. # Create scatter mechanisms
261. SiNISTMott = mon.SelectableElasticSM(Si,mon.NISTMottRS.Factory)

```

```

262.     SiDS = mon.TabulatedInelasticSM(Si,3,SiTables,epq.ToSI.eV(13.54))
263.     # The eps0 value is n^2, where n=3.4155 is taken from Palik.
        epsInfinity is from Palik's 2000 eV value of
264.     # n = 0.9999048
265.     Siphonon =
        mon.GanachaudMokraniPhononInelasticSM(phononStrength,epq.ToSI.eV(phononE),300.,11.7,1.)
266.     #SiAbruptBarrier = mon.ExpQMBBarrierSM(Si,0.)
267.     # Make a material scatter model
268.     # MSM to be used in thin layer (includes SE generation)
269.     SiMSM = mon.MONSEL_MaterialScatterModel(Si)
270.     SiMSM.addScatterMechanism(SiNISTMott)
271.     SiMSM.addScatterMechanism(SiDS)
272.     SiMSM.addScatterMechanism(Siphonon)
273.     #SiMSM.setBarrierSM(SiAbruptBarrier) # Omitting this line causes
        the barrier to default to gradual/classical
274.     # MSM to be used deep inside (drops electrons with E<50 eV)
275.     SiMSMDeep = mon.MONSEL_MaterialScatterModel(Si)
276.     SiMSMDeep.addScatterMechanism(SiNISTMott)
277.     SiMSMDeep.addScatterMechanism(SiDS)
278.     SiMSMDeep.addScatterMechanism(Siphonon)
279.     #SiMSMDeep.setBarrierSM(SiAbruptBarrier) # Omitting this line
        causes the barrier to default to gradual/classical
280.     SiMSMDeep.setMinEforTracking(epq.ToSI.eV(50.)) #default
281.
282.
283.     #   glassy carbon, set up for TabulatedInelasticSM mode 3 with
        energy levels as follows:
284.     #   CB bottom at -25.4 eV relative to vacuum = 0 eV.
285.     #   Interface barrier is gradual.
286.     density = 1800.
287.     workfun = 5.0
288.     bandgap = 0. # width of band gap in eV
289.     EFermi = 20.4 #
290.     potU = -workfun-EFermi
291.     glC = mon.SEMaterial([epq.Element.C],[1.],density,"glassy
        Carbon")
292.     glCWorkfunction=epq.ToSI.eV(workfun)
293.     glC.setWorkfunction(glCWorkfunction)
294.     glC.setEnergyCBbottom(epq.ToSI.eV(potU))
295.     glC.setBandgap(epq.ToSI.eV(bandgap))
296.     glC.setCoreEnergy([epq.ToSI.eV(284.2)])
297.     # Edit the string below so it is the path to the folder where you
        have stored the glassy carbon
298.     # scattering tables that I provide
299.     if (WinOrLin<>2):
300.         tablePath = tablePathOuter + "\glassyCTables"+PathSep
301.         #If Windows:         tablePath = tablePathOuter +
            "\glassyCTables"+PathSep
302.         #If LINUX:         tablePath = tablePathOuter +
            PathSep+"glassyCTables"+PathSep
303.     if (WinOrLin==2):
304.         tablePath = tablePathOuter + PathSep+"glassyCTables"+PathSep
305.     glCTables = [tablePath +"IIMFPPennInterpglassyCSI.csv", tablePath
        +"interpNUSimReducedDeltaEglassyCSI.csv", tablePath

```

```

+"interpsimTableThetaNUglassyCSI.csv", tablePath
+"interpSimESE0NUglassyCSI.csv"]
306. # Create scatter mechanisms
307. glCNISTMott = mon.SelectableElasticSM(glC,mon.NISTMottRS.Factory)
308. glCDS = mon.TabulatedInelasticSM(glC,3,glCTables)
309. #glCAbruptBarrier = mon.ExpQMBBarrierSM(glC,0.)
310. # Make a material scatter model
311. # MSM to be used in thin layer (includes SE generation)
312. glCMSM = mon.MONSEL_MaterialScatterModel(glC)
313. glCMSM.addScatterMechanism(glCNISTMott)
314. glCMSM.addScatterMechanism(glCDS)
315. #glCMSM.setBarrierSM(glCAbruptBarrier)
316. # MSM to be used deep inside (drops electrons with E<50 eV)
317. glCMSMDeep = mon.MONSEL_MaterialScatterModel(glC)
318. glCMSMDeep.addScatterMechanism(glCNISTMott)
319. glCMSMDeep.addScatterMechanism(glCDS)
320. #glCMSMDeep.setBarrierSM(glCAbruptBarrier)
321. glCMSMDeep.setMinEforTracking(epq.ToSI.eV(50.))
322.
323. # Cu
324. #           Copper
325. density = 8933.
326. nve = 11
327. #plasmonE = 9.11
328. workfun = 4.65
329. EFermi = 8.7
330. potU = -workfun-EFermi # Assumes Cu Fermi energy is 8.7 eV
331. Cu = mon.SEmaterial([epq.Element.Cu],[1.],density,"Copper")
332. CuWorkfunction=epq.ToSI.eV(workfun)
333. Cu.setWorkfunction(CuWorkfunction)
334. Cu.setEnergyCBbottom(epq.ToSI.eV(potU))
335. Cu.setEpsr(50.0) # a large number, to mimic a metal by using
    large dielectric constant
336. if (WinOrLin<>2):
337.     tablePath = tablePathOuter + "\CuTables"+PathSep
338.     #If Windows:     tablePath = tablePathOuter +
    "\CuTables"+PathSep
339.     #If LINUX:     tablePath = tablePathOuter +
    PathSep+"CuTables"+PathSep
340. if (WinOrLin==2):
341.     tablePath = tablePathOuter + PathSep+"CuTables"+PathSep
342. CuTables =
    [tablePath+"IIMFPPennInterpCuSI.csv",tablePath+"interpNUSimReducedDelt
    aECuSI.csv",tablePath+"interpsimTableThetaNUCuSI.csv",tablePath+"inter
    pSimESE0NUCuSI.csv"]
343. Cu.setCoreEnergy([epq.ToSI.eV(75.1),epq.ToSI.eV(77.3),epq.ToSI.eV
    (122.5),epq.ToSI.eV(932.7),epq.ToSI.eV(1096.7),epq.ToSI.eV(8979.)])
344. #densitylelectron = Cu.getDensity()/epq.Element.Cu.getMass()
345. #Cu.addBindingEnergy(epq.ToSI.eV(0.)+CuWorkfunction,nve*densityle
    lectron)
346. # Create scatter mechanisms
347. CuNISTMott = mon.SelectableElasticSM(Cu,mon.NISTMottRS.Factory)
348. CuDS = mon.TabulatedInelasticSM(Cu,3,CuTables)
349. #CuMoller = mon.MollerInelasticSM(Cu)
350. #CuPlasmon = mon.KoteraPlasmonInelasticSM(Cu,1.)
351. CuBarrier = mon.ExpQMBBarrierSM(Cu)

```



```

352.     CuCSD = mon.ZeroCSD()
353.     # Make a material scatter model
354.     # MSM to be used in thin layer (includes SE generation)
355.     CuMSM = mon.MONSEL_MaterialScatterModel(Cu)
356.     CuMSM.addScatterMechanism(CuNISTMott)
357.     CuMSM.addScatterMechanism(CuDS)
358.     CuMSM.setCSD(CuCSD)
359.     CuMSM.setBarrierSM(CuBarrier)
360.     # MSM to be used deep inside (drops electrons with E<50 eV)
361.     CuMSMDeep = mon.MONSEL_MaterialScatterModel(Cu)
362.     CuMSMDeep.addScatterMechanism(CuNISTMott)
363.     CuMSMDeep.addScatterMechanism(CuDS)
364.     CuMSMDeep.setCSD(CuCSD)
365.     CuMSMDeep.setBarrierSM(CuBarrier)
366.     CuMSMDeep.setMinEforTracking(epq.ToSI.eV(50.))
367.
368.
369.     #SiO2
370.     # SiO2 with care taken for TabulatedInelasticSM mode 3 with
    energy levels as follows:
371.     #             CB bottom at -1.1 eV
372.     #             VB top at -10 eV (i.e., 8.9 eV bandgap)
373.     #             VB bottom at -30 eV (i.e., 28.9 eV offset between
    CB and VB bottoms)
374.     #             All above energies are relative to vacuum = 0 eV.
375.     # Two phonon modes near 0.145 eV and a gradual barrier are
    assumed.
376.     density = 2200.
377.     workfun = 10.
378.     phononStrength = 2. # Number of phonon modes
379.     phononE = 0.145 # Phonon mode energy in eV
380.     bandgap = 8.9 # width of band gap in eV
381.     EFermi = -bandgap # This puts the Fermi level at the top of the
    valence band.
382.     potU = -workfun-EFermi
383.     Si = epq.Element.Si
384.     Ox = epq.Element.O
385.     SiWeight = Si.getAtomicWeight()
386.     OxWeight = 2*Ox.getAtomicWeight()
387.     totalWeight = SiWeight+OxWeight
388.     SiO2 =
    mon.SEmaterial([Si,Ox],[SiWeight/totalWeight,OxWeight/totalWeight],den
    sity,"Silicon Dioxide")
389.     SiO2.setEpsr(3.9)
390.     SiO2Workfunction=epq.ToSI.eV(workfun)
391.     SiO2.setWorkfunction(SiO2Workfunction)
392.     SiO2.setBandgap(epq.ToSI.eV(bandgap))
393.     SiO2.setEnergyCBbottom(epq.ToSI.eV(potU))
394.     SiO2.setCoreEnergy([epq.ToSI.eV(41.6),epq.ToSI.eV(99.2),epq.ToSI.
    eV(99.8),epq.ToSI.eV(149.7),epq.ToSI.eV(543.1),epq.ToSI.eV(1839.)])
395.     if (WinOrLin<>2):
396.         tablePath = tablePathOuter + "\SiO2Tables"+PathSep
397.         #If Windows:         tablePath = tablePathOuter +
    "\SiO2Tables"+PathSep
398.         #If LINUX:         tablePath = tablePathOuter +
    PathSep+"SiO2Tables"+PathSep

```

```

399.     if (WinOrLin==2):
400.         tablePath = tablePathOuter + PathSep+"SiO2Tables"+PathSep
401.         SiO2Tables =
[tablePath+"IIMFPPennInterpSiO2SI.csv",tablePath+"interpNUSimReducedDe
ltaESiO2SI.csv",tablePath+"interpsimTableThetaNUSiO2SI.csv",tablePath+
"interpSimESE0NUSiO2SI.csv"]
402.         # Create scatter mechanisms
403.         SiO2NISTMott =
mon.SelectableElasticSM(SiO2,mon.NISTMottRS.Factory)
404.         SiO2DS =
mon.TabulatedInelasticSM(SiO2,3,SiO2Tables,epq.ToSI.eV(20.+bandgap))
405.         SiO2phonon =
mon.GanachaudMokraniPhononInelasticSM(phononStrength,epq.ToSI.eV(phononE),300.,3.82,1.)
406.         SiO2polaron =
mon.GanachaudMokraniPolaronTrapSM(1.0e9,1./epq.ToSI.eV(1.))
407.         SiO2Barrier = mon.ExpQMBBarrierSM(SiO2,1.e-9)
408.         #SiO2CSD = mon.ZeroCSD() #
The default. No need to actually execute this line.
409.         # Make a material scatter model
410.         # MSM to be used in thin layer (includes SE generation)
411.         SiO2MSM = mon.MONSEL_MaterialScatterModel(SiO2)
412.         SiO2MSM.addScatterMechanism(SiO2NISTMott)
413.         SiO2MSM.addScatterMechanism(SiO2DS)
414.         SiO2MSM.addScatterMechanism(SiO2phonon)
415.         # SiO2MSM.addScatterMechanism(SiO2polaron)
416.         #SiO2MSM.setCSD(SiO2CSD)
417.         SiO2MSM.setBarrierSM(SiO2Barrier)
418.         # MSM to be used deep inside (drops electrons with E<50 eV)
419.         SiO2MSMDeep = mon.MONSEL_MaterialScatterModel(SiO2)
420.         SiO2MSMDeep.addScatterMechanism(SiO2NISTMott)
421.         SiO2MSMDeep.addScatterMechanism(SiO2DS)
422.         SiO2MSMDeep.addScatterMechanism(SiO2phonon)
423.         #SiO2MSMDeep.setCSD(SiO2CSD)
424.         SiO2MSMDeep.setBarrierSM(SiO2Barrier)
425.         SiO2MSMDeep.setMinEforTracking(epq.ToSI.eV(50.))
426.
427.
428.         # Tungsten
429.         density = 19300.
430.         #plasmonE = 9.11
431.         workfun = 4.55
432.         EFermi = 10.1
433.         potU = -workfun-EFermi
434.         W = mon.SEmaterial([epq.Element.W],[1.],density,"Tungsten")
435.         WWorkfunction=epq.ToSI.eV(workfun)
436.         W.setWorkfunction(WWorkfunction)
437.         W.setEnergyCBbottom(epq.ToSI.eV(potU))
438.         tablePath = "C:\NIST\JMONSEL\ScatteringTables\WTables"+PathSep
439.         WTables =
[tablePath+"IIMFPPennInterpWSI.csv",tablePath+"interpNUSimReducedDelta
EWSI.csv",tablePath+"interpsimTableThetaNUWSI.csv",tablePath+"interpSi
mESE0NUWSI.csv"]
440.         coreEnergies = [31.4, 33.6, 36.8, 45.3, 75.6, 243.5, 255.9,
423.6, 490.4, 594.1,

```

```

441.     1809., 1949., 2281., 2575., 2820., 10207., 11544., 12100.,
        69525.]
442.     for i in range(len(coreEnergies)):
443.         coreEnergies[i] = epq.ToSI.eV(coreEnergies[i])
444.         W.setCoreEnergy(coreEnergies)

445.         #densitylelectron = W.getDensity()/epq.Element.W.getMass()
446.         #W.addBindingEnergy(epq.ToSI.eV(0.)+WWorkfunction,nve*densitylele
            ctron)
447.         # Create scatter mechanisms
448.         WNISTMott = mon.SelectableElasticSM(W,mon.NISTMottRS.Factory)
449.         WDS = mon.TabulatedInelasticSM(W,3,WTables)
450.         #WMoller = mon.MollerInelasticSM(W)
451.         #WPlasmon = mon.KoteraPlasmonInelasticSM(W,1.)
452.         #WBarrier = mon.ExpQMBBarrierSM(W)
453.         WCSD = mon.ZeroCSD()
454.         # Make a material scatter model
455.         # MSM to be used in thin layer (includes SE generation)
456.         WMSM = mon.MONSEL_MaterialScatterModel(W)
457.         WMSM.addScatterMechanism(WNISTMott)
458.         WMSM.addScatterMechanism(WDS)
459.         WMSM.setCSD(WCSD)
460.         #WMSM.setBarrierSM(WBarrier)
461.         # MSM to be used deep inside (drops electrons with E<50 eV)
462.         WMSMDeep = mon.MONSEL_MaterialScatterModel(W)
463.         WMSMDeep.addScatterMechanism(WNISTMott)
464.         WMSMDeep.addScatterMechanism(WDS)
465.         WMSMDeep.setCSD(WCSD)
466.         #WMSMDeep.setBarrierSM(WBarrier)
467.         WMSMDeep.setMinEforTracking(epq.ToSI.eV(50.))
468.
469.
470.         # Gold
471.         density = 19282.
472.         #plasmonE = 9.11
473.         workfun = 5.1
474.         EFermi = 9.0
475.         potU = -workfun-EFermi #
476.         Au = mon.SEmaterial([epq.Element.Au],[1.],density,"Gold")
477.         AuWorkfunction=epq.ToSI.eV(workfun)
478.         Au.setWorkfunction(AuWorkfunction)
479.         Au.setEnergyCBbottom(epq.ToSI.eV(potU))
480.         Au.setEpsr(50.0) # a large number, to mimic a metal by using
            large dielectric constant
481.         if (WinOrLin<>2):
482.             tablePath = tablePathOuter + "\AuTables"+PathSep
            #If Windows: tablePath = tablePathOuter + "\AuTables"+PathSep
            If LINUX: tablePath = tablePathOuter + PathSep+"AuTables"+PathSep
483.         if (WinOrLin==2):
484.             tablePath = tablePathOuter + PathSep+"AuTables"+PathSep
485.         AuTables =
            [tablePath+"IIMFPPennInterpAuSI.csv",tablePath+"interpNUSimReducedDelt
            aEAuSI.csv",tablePath+"interpsimTableThetaNUAuSI.csv",tablePath+"inter
            pSimESE0NUAuSI.csv"]
486.         AuCoreEnergieseV = [57.2, 74.2, 83.9, 87.6, 107.2, 335.1, 353.2,
            546.3, 642.7, \

```

```

487.     762.1, 2206, 2291, 2743, 3148, 3425, 11919, 13734, 14353, 80725]
488.     for en in AuCoreEnergies.eV:
489.         Au.addCoreEnergy(epq.ToSI.eV(en))
490.         #densitylelectron = Au.getDensity()/epq.Element.Au.getMass()
491.         #Au.addBindingEnergy(epq.ToSI.eV(0.)+AuWorkfunction,nve*densityle
            lection)
492.         # Create scatter mechanisms
493.         AuNISTMott = mon.SelectableElasticSM(Au,mon.NISTMottRS.Factory)
494.         AuDS = mon.TabulatedInelasticSM(Au,3,AuTables)
495.         #AuMoller = mon.MollerInelasticSM(Au)
496.         #AuPlasmon = mon.KoteraPlasmonInelasticSM(Au,1.)
497.         AuBarrier = mon.ExpQMBBarrierSM(Au,0.05e-9)
498.         AuCSD = mon.ZeroCSD()
499.         # Make a material scatter model
500.         # MSM to be used in thin layer (includes SE generation)
501.         AuMSM = mon.MONSEL_MaterialScatterModel(Au)
502.         AuMSM.addScatterMechanism(AuNISTMott)
503.         AuMSM.addScatterMechanism(AuDS)
504.         AuMSM.setCSD(AuCSD)
505.         AuMSM.setBarrierSM(AuBarrier)
506.         # MSM to be used deep inside (drops electrons with E<50 eV)
507.         AuMSMDeep = mon.MONSEL_MaterialScatterModel(Au)
508.         AuMSMDeep.addScatterMechanism(AuNISTMott)
509.         AuMSMDeep.addScatterMechanism(AuDS)
510.         AuMSMDeep.setCSD(AuCSD)
511.         AuMSMDeep.setBarrierSM(AuBarrier)
512.         AuMSMDeep.setMinEforTracking(epq.ToSI.eV(50.))
513.
514.
515.         # Conversions of shape parameters to SI units.
516.         # Shape parameters.
517.         meterspernm = 1.e-9 # conversion from nanometers to meters
518.         pitch = pitchnm*meterspernm
519.
520.         linelength = linelengthnm*meterspernm
521.         # Note that sidewall angles are specified with respect to
            vertical,
522.         # so 0. is vertical, positive angles have bottom wider than top,
            and
523.         # negative angles are the reverse (undercut).
524.         radperdeg = jl.Math.PI/180. # conversion from degrees to radians
525.         thetar = thetardeg*radperdeg
526.         thetal = thetaldeg*radperdeg
527.         LinkThetaR = LinkThetaRdeg*radperdeg
528.         LinkThetaL = LinkThetaLdeg*radperdeg
529.         RotAng = RotAngDeg*radperdeg
530.         radr = radrnm*meterspernm
531.         radl = radlnm*meterspernm
532.         layer1thickness = layer1thicknessnm*meterspernm
533.         layer2thickness = layer2thicknessnm*meterspernm
534.
535.         deep = deepnm*meterspernm
536.         RotPitch = pitch/(jl.Math.cos(RotAng))
537.
538.
539.

```

```

540.     # Print simulation parameters to window and output file.
541.     print >>file, "# Trajectories at each landing position:
        ",nTrajectories
542.     print "# Trajectories at each landing position: ",nTrajectories
543.     print >>file, "# Defect Type: ",DefectType
544.     print "# Defect Type: ",DefectType
545.     print >>file, "# 0=No Defect, 1=type A defect, 2=type By defect,
        3=type Bx defect, 4=type Bx2 defect at link end, 5=line extension,
        6=sidewall bump"
546.     print "# 0=No Defect, 1=type A defect, 2=type By defect, 3=type
        Bx defect, 4=type Bx2 defect at link end, 5=line extension, 6=sidewall
        bump"
547.     print >>file, "# 7=roughened link, 8=misaligned link, 9=CD
        variation&misalign, 10=missing link, 11=mid-link gap, 12=mouse-bite,
        13=shortened link, 14=top rounding, 15=sidewall angle"
548.     print "# 7=roughened link, 8=misaligned link, 9=CD
        variation&misalign, 10=missing link, 11=mid-link gap, 12=mouse-bite,
        13=shortened link, 14=top rounding, 15=sidewall angle"
549.     print >>file, "# lines: ",nlines
550.     print "# lines: ",nlines
551.     print >>file, "Pitch of lines (nm): ",pitchnm
552.     print "Pitch of lines (nm): ",pitchnm
553.     print >>file, "Rotated Pitch of lines (nm):
        ",pitchnm/(jl.Math.cos(RotAng))
554.     print "Rotated Pitch of lines (nm):
        ",pitchnm/(jl.Math.cos(RotAng))
555.     print >>file, "Grating Rotation Angle (deg): ",RotAngDeg
556.     print "Grating Rotation Angle (deg): ",RotAngDeg
557.     print >>file, "Line height (nm): ",hnmvals
558.     print "Line height: ",hnmvals
559.     print >>file, "Line bottom width (nm): ",wnmvals
560.     print "Line bottom width (nm): ",wnmvals
561.     print >>file, "# LinkLength y[nm]: ",linklengthnm
562.     print "# LinkLength y[nm]: ",linklengthnm
563.     print >>file, "# LinkSpace y[nm]: ",linkspacenm
564.     print "# LinkSpace y[nm]: ",linkspacenm
565.     print >>file, "Line length (nm): ",linelengthnm
566.     print "Line length (nm): ",linelengthnm
567.     print >>file, "SWA range (deg): ",SWAvals
568.     print "SWA range (deg): ",SWAvals
569.     print >>file, "Left and right SWA (deg): ",thetaldeg,thetardeg
570.     print "Left and right SWA (deg): ",thetaldeg,thetardeg
571.     print >>file, "Left & Right Link Center SWA (deg):
        ",LinkThetaLdeg,LinkThetaRdeg
572.     print "Left & Right Link Center SWA (deg):
        ",LinkThetaLdeg,LinkThetaRdeg
573.     print >>file, "Left & Right Center Link Top Rounding:
        ",TopRadLnm,TopRadRnm
574.     print "Left & Right Center Link Top Rounding:
        ",TopRadLnm,TopRadRnm
575.     print >>file, "Left and right top corner radii (nm):
        ",radlnm,radrnm
576.     print "Left and right top corner radii (nm): ",radlnm,radrnm
577.     print >>file, "Thicknesses of 1st and second layers (nm):
        ",layer1thicknessnm,layer2thicknessnm

```

```

578.     print "Thicknesses of 1st and second layers (nm):
        ",layer1thicknessnm,layer2thicknessnm
579.     print >>file, "Beam landing energies (eV): ",beamEeVvals
580.     print "Beam landing energies (eV): ",beamEeVvals
581.     print >>file, "Beam size (standard deviation, in nm):
        ",beamsizenmvals
582.     print "Beam size (standard deviation, in nm): ",beamsizenmvals
583.     print >>file, "Scan Origin (x,y, in nm): ",ScanOrigX,ScanOrigY
584.     print "Scan Origin (x,y, in nm): ",ScanOrigX,ScanOrigY
585.     print >>file, "Scan Step (x,y, in nm): ",ScanStepX, ScanStepY
586.     print "Scan Step (x,y, in nm): ",ScanStepX,ScanStepY
587.     print >>file, "# Pixels (x,y): ",NumPixX, NumPixY
588.     print "# Pixels (x,y): ",NumPixX, NumPixY
589.     print >>file, "# CylDefDiam[nm]: ",CylDefDiam
590.     print "# Def1 CylDefDiam[nm]: ",CylDefDiam
591.     print >>file, "# Def1 CylDefHeight[nm]: ",CylDefHeight
592.     print "# Def1 CylDefHeight[nm]: ",CylDefHeight
593.     print >>file, "# Def2/3/4 BridgeDefWidth[nm]: ",BridgeDefWidth
594.     print "# Def2/3/4 BridgeDefWidth[nm]: ",BridgeDefWidth
595.     print >>file, "# Def2/3/4/5/6/7/8/9/12/13 BridgeDefHeight[nm]:
        ",BridgeDefHeight
596.     print "# Def2/3/4/5/6/7/8/9/12/13 BridgeDefHeight[nm]:
        ",BridgeDefHeight
597.     print >>file, "# Def5 LineExtDefLength[nm]: ",LineExtDefLength
598.     print "# Def5 LineExtDefLength[nm]: ",LineExtDefLength
599.     print >>file, "# Def13 LinkShorten[nm]: ",LinkShorten
600.     print "# Def13 LinkShorten[nm]: ",LinkShorten
601.     print >>file, "# Def6 SidewallBumpWidth x[nm]:
        ",SidewallBumpWidth
602.     print "# Def6 SidewallBumpWidth x[nm]: ",SidewallBumpWidth
603.     print >>file, "# Def6 SidewallBumpLength y[nm]:
        ",SidewallBumpLength
604.     print "# Def6 SidewallBumpLength y[nm]: ",SidewallBumpLength
605.     print >>file, "# Def7 LERTabAmplitude[nm]: ",LERTabAmplitude
606.     print "# Def7 LERTabAmplitude[nm]: ",LERTabAmplitude
607.     print >>file, "# Def7 LERTabLength[nm]: ",LERTabLength
608.     print "# Def7 LERTabLength[nm]: ",LERTabLength
609.     print >>file, "# Def7 LERTabPitch[nm]: ",LERTabPitch
610.     print "# Def7 LERTabPitch[nm]: ",LERTabPitch
611.     print >>file, "# Def8/9 LinkMisAlign[nm]: ",LinkMisAlign
612.     print "# Def8/9 LinkMisAlign[nm]: ",LinkMisAlign
613.     print >>file, "# Def9 CDvar[nm] (H,W,L): ",CDvarH, CDvarW, CDvarL
614.     print "# Def9 CDvar[nm] (H,W,L): ",CDvarH, CDvarW, CDvarL
615.     print >>file, "# Def12 MouseBiteWidth x[nm]: ",MouseBiteWidth
616.     print "# Def12 MouseBiteWidth x[nm]: ",MouseBiteWidth
617.     print >>file, "# Def12 MouseBiteLength y[nm]: ",MouseBiteLength
618.     print "# Def12 MouseBiteLength y[nm]: ",MouseBiteLength
619.     print >>file, "# Def11 MidLinkGap[nm]: ",MidLinkGap
620.     print "# Def11 MidLinkGap[nm]: ",MidLinkGap
621.     print >>file, "# Def14 Top Rounding[nm]: ",TopRadRnm, TopRadLnm
622.     print "# Def14 Top Tounding[nm]: ",TopRadRnm, TopRadLnm
623.     print >>file, "# Def15 Sidewall Angle[deg]: ",LinkThetaRdeg,
        LinkThetaLdeg
624.     print "# Def15 Sidewall Angle[deg]: ",LinkThetaRdeg,
        LinkThetaLdeg
625.

```

```

626.     print >>file      # Blank line before start of calculation results
627.     print
628.
629.     print >>file, "beamsize(nm)  h(nm)      w(nm)  beamE(eV)    x(nm)
        y(nm)  BSEyield    SEyield    ElapsedTime"      #print >>file,
        "beamsize(nm) h(nm) w(nm) SWAl(deg) SWAr(deg) beamE (eV)  x(nm)  y
        (nm)  BSE yield    SE yield    Elapsed Time"
630.     print            "beamsize(nm)  h(nm)      w(nm)  beamE(eV)    x(nm)
        y(nm)  BSEyield    SEyield    ElapsedTime"      #print "beamsize(nm)
        h(nm) w(nm) SWAl(deg) SWAr(deg) beamE (eV)  x(nm)  y (nm)  BSE yield
        SE yield    Elapsed Time"
631.
632.     t0 = jl.System.currentTimeMillis()
633.
634.
635.
636.     #loop beamsizenmvals
637.     for beamsizenm in beamsizenmvals:
638.
639.         beamsize = beamsizenm*meterspernm
640.
641.         # create an instance of the model
642.         monte=nm.MonteCarloSS() #creates an instance of the model
        with all default characteristics
643.         eg = nm.GaussianBeam(beamsize) # makes electron gun, Gaussian
        with standard deviation = beamsize
644.         monte.setElectronGun(eg) # This gun is "attached" to the
        model.
645.
646.         # SAMPLE DESCRIPTION
647.
648.         # NISTMonte provides us with a "chamber" in the form of a 0.1
        m sphere inside of which we build
649.         # out sample. Replace the default vacuum in the chamber with
        SEvacuum. (SEmaterials define additional
650.         # properties, such as work function, that are needed by
        JMONSEL.)
651.         chamber = monte.getChamber()
652.         chamber.updateMaterial(chamber.getScatterModel(),vacuumMSM)
653.
654.         # Generate the sample. The Gaussian Beam electron gun has
        this peculiarity: it defines the +z axis to
655.         # be in the direction of travel of the electrons. When we
        describe the sample in this coordinate
656.         # system, it is inverted along the z direction.
657.
658.         # Make sample component shapes
659.         normalvector = [0.,0.,-1.]
660.
661.         # First we make the layers. The simplest way to do this is to
        define each as a MultiPlaneShape with a single
662.         # plane, each nested inside the previous one.
663.         layer1 = mon.NormalMultiPlaneShape()
664.         layer1.addPlane(normalvector,[0.,0.,0.]) #layer 1 is now the
        half space of everything above the x-y plane

```

```

665.      # This region has shape defined by layer1, scattering
        properties defined for Si, and is a subregion of (is
666.      # wholly contained within) the chamber.
667.      layer1Region = monte.addSubRegion(chamber, SiMSM, layer1)
668.
669.      layer2 = mon.NormalMultiPlaneShape()
670.      layer2.addPlane(normalvector, [0., 0., layer1thickness]) #layer
        2 starts layer1thickness farther up.
671.      # We give it the properties of Si, and make it a subregion of
        layer1Region. At this point, layer1Region
672.      # extends only for 0<=z<=layer1thickness.
673.      layer2Region =
        monte.addSubRegion(layer1Region, SiMSMDeep, layer2)
674.
675.      layer3 = mon.NormalMultiPlaneShape()
676.      layer3.addPlane(normalvector, [0., 0., layer1thickness+layer2thickness])
        #layer 3 starts
677.      # yet another layer2thickness farther up.
678.      # We give it the properties of Si, and make it a subregion of
        layer2Region. At this point, layer2Region
679.      # extends only for
        layer1thickness<=z<=layer1thickness+layer2thickness.
680.      layer3Region =
        monte.addSubRegion(layer2Region, SiMSMDeep, layer3)
681.
682.
683.      #loop hnm in hnmvals
684.      for hnm in hnmvals:
685.
686.          #loop wnm in wnmvals
687.          for wnm in wnmvals:
688.
689.              h = hnm*meterspernm
690.              w = wnm*meterspernm
691.              #loop SWAtemp in SWAvals
692.              for SWAtemp in SWAvals:
693.                  if SWAdecision:
694.                      thetardeg=SWAtemp
695.                      thetaldeg=SWAtemp
696.
697.                      thetar = thetardeg*radperdeg
698.                      thetal = thetaldeg*radperdeg
699.
700.                      # Make the array of lines. The integer divide in
        (nlines/2) truncates fractions.
701.                      # The result always has one line centered at
        (x,y)=(0,0). If nlines is odd
702.                      # the remaining lines are placed symmetrically
        left and right of this one.
703.                      # If nlines is even, there will be one more line
        on the right side than on the
704.                      # left side.
705.
706.
707.                      #IDA SRAM cell

```



```

708.         nlinks = 12
709.         linklength = linklengthnm*1.e-9
710.         linkspace = linkspacenm*1.e-9
711.         linkpitch = linklength + linkspace
712.         PitchWalk = 0 * 1.e-9
713.         leftmostLineCenterx = -pitch * (nlines/2)
714.         bottommostLinkCentery = -linkpitch * (nlinks/2)
715.
716.
717.         for i in range(nlines):
718.             if (i % 2 == 0):
719.                 xcenter = leftmostLineCenterx + i * pitch
720.             if (i % 2 == 1):
721.                 xcenter =
leftmostLineCenterx+i*pitch+PitchWalk
722.                 for k in range(nlinks):
723.                     ycenter = bottommostLinkCentery + k *
linkpitch
724.                     if (DefectType < 8):
#note this makes normal links everywhere for DefectType<8
725.                         link = mon.NShapes.createLine(-
h,w,linklength,thetal,thetar,radl,radr)
726.                         link.translate([xcenter,ycenter,0.])
727.                         linkRegion =
monte.addSubRegion(chamber,SiO2MSM,link) # **set material for the
lines here**
728.
729.
730.
731.                     if ((DefectType >= 8) and ((xcenter<>0)
or (ycenter<>0))): #note If DefectType>=8 and at center link,
will not place a link (leaves room for custom link for subtractor
defects)
732.
733.                         link = mon.NShapes.createLine(-
h,w,linklength,thetal,thetar,radl,radr)
734.                         link.translate([xcenter,ycenter,0.])
735.                         linkRegion =
monte.addSubRegion(chamber,SiO2MSM,link) # **set material for the
lines here**
736.
737.
738.
739.                     if (DefectType == 0): #No Defect
740.                         ntab=linklength % LERTabPitch
741.                         #No Defect, "Perfect Cell", no further action
necessary
742.
743.                         #Cylindrical or Spherical Defects
744.                         if (DefectType == 1): #Type A defect
745.                             CylDefect1 = mon.NormalCylindricalShape([-
w,0*1e-9,0.], [-w,0*1e-9,-CylDefHeight*1e-9],CylDefDiam*1e-9/2)
746.                             #CylDefect1 =
mon.NormalCylindricalShape([0.,0.,0.], [0.,0.,-h],w/2)
747.                             #SphDefect1 = mon.NormalSphereShape([0.,0.,-
w/2],w/2)

```

```

748.             CylDefectRegion =
monte.addSubRegion(chamber,CuMSM,CylDefect1)      # **set material for
the lines here**
749.             #SphDefect1Region =
monte.addSubRegion(chamber,SiMSM,SphDefect1)      # **set material for
the lines here**
750.
751.             #Line defects
752.             if (DefectType == 2):      #By defect
753.                 LineDefect1 = mon.NShapes.createLine(-
BridgeDefHeight*1e-9,BridgeDefWidth*1e-
9,linkspace,thetal,thetar,radl,radr)      #bridge-x      (w.r.t. grating)
754.                 LineDefect1.translate([0.,-
(linklength/2+linkspace/2),0.])
#LineDefect1.translate([0.,+(linklength/2+linkspace/2),0.])
755.                 LineDefectRegion =
monte.addSubRegion(chamber,SiO2MSM,LineDefect1)  # **set material for
the lines here**
756.
757.                 if (DefectType == 3):      #Bx defect @ link
center
758.                 LineDefect2 = mon.NShapes.createLine(-
BridgeDefHeight*1e-9,BridgeDefWidth*1e-9,pitch-
w,thetal,thetar,radl,radr) #bridge-y      (w.r.t. grating)
759.                 LineDefect2.rotate([0.,0.,0.],90.*radperdeg,0.,0.)
760.                 LineDefect2.translate([pitch/2,0.,0.])
761.                 LineDefectRegion =
monte.addSubRegion(chamber,SiO2MSM,LineDefect2)  # **set material for
the lines here**
762.
763.                 if (DefectType == 4):      #Bx2 defect @ link end
764.                 LineDefect3 = mon.NShapes.createLine(-
BridgeDefHeight*1e-9,BridgeDefWidth*1e-9,pitch-
w,thetal,thetar,radl,radr) #bridge-y      (w.r.t. grating)
765.                 LineDefect3.rotate([0.,0.,0.],90.*radperdeg,0.,0.)
766.                 LineDefect3.translate([pitch/2,linklength/2-
BridgeDefWidth/2*1e-9,0.])
767.                 LineDefectRegion =
monte.addSubRegion(chamber,SiO2MSM,LineDefect3)  # **set material for
the lines here**
768.
769.                 if (DefectType == 5):      #Line Extension
770.                 LineDefect4 = mon.NShapes.createLine(-
BridgeDefHeight*1.e-9,w,LineExtDefLength*1.e-
9,thetal,thetar,radl,radr)      #bridge-y      (w.r.t. grating)
771.                 LineDefect4.rotate([0.,0.,0.],0.,0.,0.)
772.                 LineDefect4.translate([0.,linklength/2+LineExtDefLength/2*1.e-9,0.])
773.                 LineDefectRegion =
monte.addSubRegion(chamber,SiO2MSM,LineDefect4)  # **set material for
the lines here**
774.
775.                 if (DefectType == 6):      #Sidewall Bump

```

```

776.             LineDefect5 = mon.NShapes.createLine(-
BridgeDefHeight*1e-9,SidewallBumpWidth*1.e-9,SidewallBumpLength*1.e-
9,thetal,thetar,radl,radr)      #bridge-y      (w.r.t. grating)
777.             LineDefect5.rotate([0.,0.,0.],0.,0.,0.)
778.
LineDefect5.translate([w/2+SidewallBumpWidth/2*1.e-9,0.,0.])
779.             LineDefectRegion =
monte.addSubRegion(chamber,SiO2MSM,LineDefect5)  # **set material for
the lines here**
780.
781.             if (DefectType == 7):      #Roughened Link
782.                 ntab = (linklengthnm / LERTabPitch - 1)
783.                 bottommostTabCenterY = (-1*LERTabPitch*1.e-
9)*(ntab/2) + (LERTabLength*1.e-9)
784.                 for iLER in range(ntab):
785.                     ycenterLER = bottommostTabCenterY +
iLER*LERTabPitch*1.e-9
786.                     linetab6 = mon.NShapes.createLine(-
h,w*1.3*1.2,(LERTabLength*1.e-9),thetal,thetar,0.,0.)
787.                     linetab6.translate([(-
LERTabAmplitude*1.e-9/2),ycenterLER,0.])
788.                     linetab6Region =
monte.addSubRegion(chamber,SiO2MSM,linetab6)  # **set material for
the lines here**
789.                     linetab7 = mon.NShapes.createLine(-
h,w*1.2,(LERTabLength*1.e-9),thetal,thetar,0.,0.)
790.                     linetab7.translate([(-
LERTabAmplitude*1.e-9/2),ycenterLER,0.])
791.                     linetab7Region =
monte.addSubRegion(chamber,SiMSM,linetab7)      # **set material for
the lines here**
792.                     # linetab7 = mon.NShapes.createLine(-
h,(LERTabAmplitude*1.e-9),(LERTabLength*1.e-9),0.,0.,0.,0.)
793.                     #
linetab7.translate([(w/2+LERTabAmplitude*1.e-9/2),ycenterLER,0.])
794.                     # linetab7Region =
monte.addSubRegion(chamber,SiO2MSM,linetab7) # **set material for the
lines here**
795.
796.             if (DefectType == 8):      #Mis-Aligned Link
#note if DefectType=8 it adds center link, shifted over a little. If
DefectType>8 and at center link, no link (leaves room for custom link
for subtractor defects)
797.                 link8 = mon.NShapes.createLine(-
BridgeDefHeight*1.e-9,w,linklength,thetal,thetar,radl,radr)
798.                 link8.translate([0.+LinkMisAlign*1.e-
9,0.,0.])
799.                 link8Region =
monte.addSubRegion(chamber,SiO2MSM,link8)      # **set material for
the lines here**
800.
801.             if (DefectType == 9):      #CD Variation &
MisAlign      #note this builds an alternative link at the
center link

```

```

802.                link9 = mon.NShapes.createLine(-(h-
CDvarH*1.e-9),w-CDvarW*1.e-9,linklength-CDvarL*1.e-
9,thetal,thetar,radl,radr)
803.                # link9 = mon.NShapes.createLine(-(h-
CDvarH*1.e-9),w,linklength,thetal,thetar,radl,radr)      #height change
804.                # link9 = mon.NShapes.createLine(-h,w-
CDvarW*1.e-9,linklength,thetal,thetar,radl,radr)      #width change
805.                # link9 = mon.NShapes.createLine(-
h,w,linklength-CDvarL*1.e-9,thetal,thetar,radl,radr)      #length
change
806.                link9.translate([0.+LinkMisAlign*1.e-
9,0.,0.])
807.                #link9Region =
monte.addSubRegion(chamber,SiO2MSM,link9)                # **set material for
the lines here**
808.                link9Region =
monte.addSubRegion(chamber,SiMSM,link9)                # **set material for
the lines here**
809.
810.                if (DefectType == 10):                #Missing Link
811.                    ntab=linklength % LERTabPitch
812.                    #Missing Link, Link at (0,0) skipped above in
link logic, no further action necessary
813.
814.                if (DefectType == 11):                #Mid-Link Gap
#note this builds an alternative link at the center link
815.                    LineDefect8 = mon.NShapes.createLine(-
h,w,(linklength/2-MidLinkGap*1e-9/2),thetal,thetar,radl,radr) #bridge-
x      (w.r.t. grating)
816.                    LineDefect8.translate([0.,(-linklength/4-
(MidLinkGap*1e-9/4)),0.])
817.                    LineDefect8Region =
monte.addSubRegion(chamber,SiO2MSM,LineDefect8) # **set material for
the lines here**
818.                    LineDefect9 = mon.NShapes.createLine(-
h,w,(linklength/2-MidLinkGap*1e-9/2),thetal,thetar,radl,radr) #bridge-
x      (w.r.t. grating)
819.                    LineDefect9.translate([0.,(+linklength/4+(MidLinkGap*1e-9/4)),0.])
820.                    LineDefect9Region =
monte.addSubRegion(chamber,SiO2MSM,LineDefect9) # **set material for
the lines here**
821.
822.                if (DefectType == 12):                #MouseBite
#note this builds an alternative link at the center link
823.                    LineDefect10 = mon.NShapes.createLine(-
h,w,(linklength/2-(MouseBiteLength*1e-9)/2),thetal,thetar,radl,radr)
#bridge-x      (w.r.t. grating)
824.                    LineDefect10.translate([0.,(-linklength/4-
(MouseBiteLength*1e-9/4)),0.])
825.                    LineDefect10Region =
monte.addSubRegion(chamber,SiO2MSM,LineDefect10) # **set material
for the lines here**
826.                    LineDefect11 = mon.NShapes.createLine(-
h,w,(linklength/2-(MouseBiteLength*1e-9)/2),thetal,thetar,radl,radr)
#bridge-x      (w.r.t. grating)

```

```

827.     LineDefect11.translate([0., (+linklength/4+(MouseBiteLength*1e-
828.         LineDefect11Region =
monte.addSubRegion(chamber, SiO2MSM, LineDefect11)    # **set material
for the lines here**
829.         LineDefect12 = mon.NShapes.createLine(-
BridgeDefHeight*1.e-9, (w-MouseBiteWidth*1e-9), MouseBiteLength*1e-
9, thetal, thetar, radl, radr)    #bridge-x    (w.r.t. grating)
830.         LineDefect12.translate([(-w/2+(w-
MouseBiteWidth*1e-9)/2), 0., 0.])
831.         LineDefect12Region =
monte.addSubRegion(chamber, SiO2MSM, LineDefect12)    # **set material
for the lines here**
832.
833.         if (DefectType == 13):    #Shortened Link
#note if DefectType=13 it adds center link, shifted up a little.  If
DefectType>8 and at center link, no link (leaves room for custom link
for subtractor defects)
834.             link13 = mon.NShapes.createLine(-
BridgeDefHeight*1e-9, w, linklength-LinkShorten*1.e-
9, thetal, thetar, radl, radr)
835.             link13.translate([0., LinkShorten/2*1.e-9, 0.])
836.             link13Region =
monte.addSubRegion(chamber, SiO2MSM, link13)    # **set
material for the lines here**
837.
838.             if (DefectType == 14):    #Top Rounding
#note this builds an alternative link at the center link
839.                 link = mon.NShapes.createLine(-
h, w, linklength, thetal, thetar, TopRadLnm*1.e-9, TopRadRnm*1.e-9)
840.                 link.translate([0., 0., 0.])
841.                 linkRegion =
monte.addSubRegion(chamber, SiO2MSM, link)    # **set
material for the lines here**
842.
843.             if (DefectType == 15):    #Sidewall Angle
#note this builds an alternative link at the center link
844.                 link = mon.NShapes.createLine(-
h, w, linklength, LinkThetaL, LinkThetaR, radl, radr)
845.                 link.translate([0., 0., 0.])
846.                 #linkRegion =
monte.addSubRegion(chamber, SiO2MSM, link)    # **set
material for the lines here**
847.
848.                 #Rotation Options
849.                 #chamber.rotate([0., 0., 0.], RotAng, 0., 0.)
#Entire sample rotated
850.                 chamber.rotate([0., 0., 0.], 0., RotAng, 0.)
#Tilt Right->Left
851.                 #chamber.rotate([0., 0., 0.], RotAng, RotAng, 0.)
852.
853.
854.                 # Scan parameters
855.
856.                 yvals = []

```

```

857.         yend = ScanOrigY + NumPixY * ScanStepY
858.         y = ScanOrigY
859.         while y < yend:
860.             yvals.append(y)
861.             y += ScanStepY
862.
863.         xvals = []
864.         xend = ScanOrigX + NumPixX * ScanStepX
865.         x = ScanOrigX
866.         while x < xend:
867.             xvals.append(x)
868.             x += ScanStepX
869.
870.         binSizeEV = 10. # Width (in eV) of bins in energy
    histogram
871.
872.         for beamEeV in beamEeVvals:
873.             beamE = epq.ToSI.eV(beamEeV)
874.             monte.setBeamEnergy(beamE) # Sets this
    model's beam energy
875.             for xnm in xvals:           # To switch x and
    y, do it here and in 4 lines below
876.                 x = xnm*meterspernm
877.                 for ynm in yvals:
878.                     y = ynm*meterspernm
879.                     eg.setCenter([x,y,-h-
    20.*meterspernm]) # Aims the gun at x,y.
880.
881.                     # Define our backscatter detector.
882.                     back=nm.BackscatterStats(monte)
883.                     nbins = int(beamEeV/binSizeEV)
884.                     monte.addActionListener(back)
885.                     back.setEnergyBinCount(nbins)
886.
887.                     # Add a trajectory image
888.                     if trajImg: # output the trajectory
    image
889.                         img=nm.TrajectoryImage(512,512,trajImgSize)
890.                         img.setMaxTrajectories(trajImgMaxTraj)
891.                         img.setYRange(-h-
    trajImgSize/10,.9*trajImgSize-h)
892.                         img.setXRange(x-
    trajImgSize/2.,x+trajImgSize/2.)
893.                         monte.addActionListener(img)
894.
895.                         # Add a vrml
896.                         if VRML: # output the trajectory
    image
897.                             #fos=jio.FileOutputStream("%s/angle - %lg deg.wrl" % (dest, phi))
898.                             fos=jio.FileOutputStream(
    "%s.wrl" % (dest))
899.                             tw=jio.OutputStreamWriter(fos,cs.Charset.forName("UTF-8"))

```

```

900.                                vrml=nm.TrajectoryVRML(monte,tw)
901.
    vrml.setMaxTrajectories(VRMLImgMaxTraj)
902.                                vrml.setTrajectoryWidth(0.25e-9)
903.                                vrml.setDisplayBackscatter(1)
904.                                vrml.addView("Gun",[0.0,0.0,-
    5.0e-7],[0.0,0.0,0.0])
905.                                vrml.addView("X-Axis",[1.0e-
    6,0.0,0.0,0.0],[0.0,0.0,0.0])
906.                                vrml.addView("Y-Axis",[0.0,1.0e-
    6,0.0,0.0],[0.0,0.0,0.0])
907.                                vrml.addView("Close
    perspective",[-110.e-8,100.e-8,-100.e-8],[-100.e-9,0.,0.])
908.                                vrml.renderSample()
909.                                monte.addActionListener(vrml)
910.
911.                                # Run the simulation
912.
    monte.runMultipleTrajectories(nTrajectories)
913.
914.                                hist =
    back.backscatterEnergyHistogram()
915.                                #fhist =
    back.forwardsatterEnergyHistogram()
916.                                energyperbineV =
    beamEeV/hist.binCount()
917.                                maxSEbin = 50./energyperbineV    # bin
    number of the one with 50 eV
918.                                totalSE = 0
919.                                for j in range(0,int(maxSEbin)):
920.                                    totalSE = totalSE+hist.counts(j)
921.                                    #print j,hist.counts(j)
922.                                    #totalSE =
    totalSE+hist.counts(j)+fhist.counts(j)
923.
924.                                SEf = float(totalSE)/nTrajectories
925.                                bsf = back.backscatterFraction()-
    SEf#+float(totalFwd)/nTrajectories
926.                                t = (jl.System.currentTimeMillis()-
    t0)/3600000.
927.                                print >>file, "%8.2f \t%8.2f \t%8.2f
    \t%8.1f \t%8.2f \t%8.2f \t%3.5f \t%3.5f \t%8.4f" %
    (beamsizenm,hnm,wnm,beamEeV,xnm,ynm,bsf,SEf,t)
928.                                print "%8.2f \t%8.2f \t%8.2f \t%8.1f
    \t%8.2f \t%8.2f \t%3.5f \t%3.5f \t%8.4f" %
    (beamsizenm,hnm,wnm,beamEeV,xnm,ynm,bsf,SEf,t)
929.                                #print "%8.2f \t%8.2f \t%8.2f \t%8.2f
    \t%8.2f \t%8.1f \t%8.2f \t%8.2f \t%3.4f \t%3.4f \t%8.4f" %
    (beamsizenm,hnm,wnm,thetaldeg,beamEeV,xnm,ynm,bsf,SEf,t)
930.                                rdirname =
    jl.Integer(int(beamEeV)).toString()+"eV_"+jl.Integer(int(xnm)).toS
    tring()+"y"+jl.Integer(int(ynm)).toString()
931.
932.                                #back.dump(jio.FileOutputStream(dest+PathSep+rdirname+"backscatter.prn
    ")) #If backscatter statistics is desired

```

```
933.                 monte.removeActionListener(back)
934.
935.                 if trajImg: # output the trajectory
    image
936.                 img.dumpToFile(dest+PathSep+rdirname)
937.                 monte.removeActionListener(img)
938.
939.                 if VRML:
940.                     tw.flush()
941.                     fos.close()
942.                     monte.removeActionListener(vrml)
943.
944.     file.close()
```