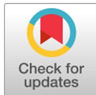

Research Article | Vol. 27, No. 13 | 24 Jun 2019 | OPTICS EXPRESS 17743

Optics EXPRESS

Optical inspection of nanoscale structures using a novel machine learning based synthetic image generation algorithm

SANYOGITA PURANDARE,[1,*] JINLONG ZHU,[1] RENJIE ZHOU, [2] GABRIEL POPESCU, [3] ALEXANDER SCHWING,[4] AND LYNFORD L. GODDARD [1]

[1] *Photonic Systems Laboratory, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[2] *Laser Metrology and Biomedicine Laboratory, Department of Biomedical Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China*
[3] *Quantitative Light Imaging Laboratory, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[4] *Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[*] *sanyogita.purandare@gmail.com*

Abstract: In this paper, we present a novel interpretable machine learning technique that uses unique physical insights about noisy optical images and a few training samples to classify nanoscale defects in noisy optical images of a semiconductor wafer. Using this technique, we not only detected both parallel bridge defects and previously undetectable perpendicular bridge defects in a 9-nm node wafer using visible light microscopy [Proc. SPIE **9424**, 942416 (2015)], but we also accurately classified their shapes and estimated their sizes. Detection and classification of nanoscale defects in optical images is a challenging task. The quality of images is affected by diffraction and noise. Machine learning techniques can reduce noise and recognize patterns using a large training set. However, for detecting a rare "killer" defect, acquisition of a sufficient training set of high quality experimental images can be prohibitively expensive. In addition, there are technical challenges involved in using electromagnetic simulations and optimization of the machine learning algorithm. This paper proposes solutions to address each of the aforementioned challenges.

© 2019 Optical Society of America under the terms of the OSA Open Access Publishing Agreement

1. Introduction

According to Gartner's September 2016 article [2], a cutting-edge chip manufacturing plant can cost ~\$5-10 billion. A 20% decrease in wafer yield can result in a ~ \$1 billion per year decrease in the gross margin for fabrication plants. Scanning electron microscopy (SEM), atomic force microscopy (AFM), and other electron beam (e-beam) defect inspection tools are widely used for nanoscale defect inspection because of their excellent resolution [3–5]. However, these tools have low areal inspection throughput and can be destructive. Thus, optical microscopy plays a crucial role in defect detection because of its ability to non-destructively image large areas at high speeds. Mass production of the 9-nm node started in late 2016. Because the light scattering from a deep subwavelength defect is extremely weak, optical defect detection is quite challenging in small node size wafers that have background device patterns. The signal to noise ratio (SNR) for the defect signature compared to the noise from the line edge roughness (LER) of the background patterns is poor. Researchers at NIST used a 193 nm microscope and through-focus scanning to enhance the sensitivity of 9-nm node defect detection [6]. Previously, our group applied diffraction phase microscopy (DPM) and noise reducing methods to detect

#362885 https://doi.org/10.1364/OE.27.017743
Journal © 2019 Received 20 Mar 2019; revised 27 Apr 2019; accepted 9 May 2019; published 12 Jun 2019

defects in a 22-nm node wafer [7]. In that paper, we presented the second order differential image stitching and convolution (2DISC) image post-processing technique and showed how it reduces the effects of specific sources of noise and system imperfections. Later, we applied DPM and 2DISC to successfully detect a 9-nm parallel bridge defect using a visible light source (405-nm laser) [1, 8]. Although we added through-focus scanning [8] and broadband illumination [1], we were unsuccessful at detecting a 9-nm perpendicular bridge because of its much weaker scattering signature compared to the background and noise.

Computational electromagnetic (EM) modeling is used in many optical metrology methods, e.g., scatterometry and through-focus scanning [6, 9, 10], to generate best-fit simulated images based on the experimental configuration and *a priori* sample info. Inferences can then be made about wafer features from the measurements. However, challenges arise, for example, when: (1) the *a priori* sample info is inaccurate, (2) the simulation domain is truncated because of finite computational resources, (3) the devices have intrinsic fabrication noise such as LER, or (4) the measurements are done in a noisy or non-ideal environment.

Machine learning and pattern recognition techniques are investigated widely for facial recognition, flaw detection, and microscopy [11–21]. However, the effectiveness of machine learning in microscopy and sensing depends critically on the quality of the training sets. In nanotechnology applications, it is challenging to obtain large amounts of experimental data for training, testing, and validation because creating appropriate test wafers and making the measurements are costly and time consuming. The data may also be inaccessible because of non-disclosure restrictions. An alternative strategy is to use simulated data for the training set; however, the exact dimensions of the features in the experimental sample may not be known and thus the simulation may not accurately reflect the actual sample. In addition, generating a large library of simulated data for machine learning may be computationally expensive.

To solve the aforementioned limitations of machine learning and challenges of EM modeling, we propose a novel machine learning technique. Principal component analysis (PCA) is a pattern recognition technique that is suitable for image compression and denoising [22]. We transform a few approximate electromagnetic simulation defect images to generate synthetic noisy defect images with trainable parameters such that its principal components can sufficiently capture variance related to the defect features. There are two challenges in implementing this approach. Firstly, the principal components must capture variance related to the defect features. Secondly, the captured variance should contain sufficient feature-related information to distinguish defect and no-defect images. We overcome these challenges by formulating a hypothesis about frame-to-frame varying noise in optical images and use it to generate synthetic images with defect feature dependent noise. In addition, we provide a physical explanation of our technique.

We use the models obtained during the learning stage to successfully detect, classify, and size various flaws and irregularities in the sample from the measured experimental microscope wafer images. It is extremely difficult even for experts to visually distinguish the defect signals from the other intensity signals and to classify defect shapes or sizes from the optical intensity images. In particular, we studied parallel and perpendicular bridge defects in a 9-nm node wafer. We found the previously undetectable perpendicular defects even in non-optimal experimental settings without specifically training the model to this defect type and even estimated its size. This size and shape classification is a significant new contribution of this paper compared to our prior work.

2. Microscope images

This section provides information about the experimental optical images used for training, validating, and testing the machine learning technique. A 9-nm Intentional Defect Array (IDA) wafer made by SEMATECH was used for the experiments. This wafer has numerous square

dies of size 100 $\mu m \times 100\ \mu m$ with a single nanoscale defect in the middle of the underlying pattern. Each die has a different nominal size and shape for the central defect. Examples of the parallel bridge (BX67 type, i.e., the designed linewidth for the defect is 67% the linewidth of the pattern) and perpendicular bridge (BY67 type) defects are shown in the SEM images in Figs. 1(a) and 1(b), respectively. Besides the central intentional defect, the wafer has other patterning irregularities such as LER, e-beam stitching errors, and contamination. The LER is quite visible in Figs. 1(a) and 1(b). Figures 1(c) and 1(d) show an e-beam stitching error and nanoscale dust, respectively. We collected experimental intensity images for the parallel bridge defect die using an epi-illumination bright-field microscope that included an xy in-plane translation stage. We also collected experimental intensity images for two different types of defects imaged using the same microscope and translation stage in the interferometric DPM configuration [1, 8, 23]. The data set contains some data previously presented in [1] along with additional data that has not been previously published. The DPM images are used as a common benchmark dataset for quantifying the performance improvement of the new machine learning algorithm relative to the results in [1]. The purpose of using intensity images from both interferometric DPM and non-interferometric measurements is to obtain images with different types and levels of noise to evaluate the robustness of the proposed machine learning technique. These microscope images are post-processed as described in Section 3.4. The post-processed interferometric microscope images of different defect types are displayed in Figs. 1(e) and 1(f). The 2DIS post-processing scheme (i.e., 2DISC without the convolution step) is applied to the microscope images [7]. 2DIS consists of second order difference (2DI) and image stitching. 2DI removes additive time-invariant spatial noise such as laser speckle, camera shot noise, or system imperfections. It also acts as a high pass filter that enhances the defect related signal. Image stitching removes temporal noise due to non-uniform illumination or camera responsivity. Throughout this paper, each pixel in an optical image corresponds to 31.3 nm $\times$ 31.3 nm. Experimental images were captured using a Hamamatsu C4742-80-12AG camera in a custom microscope that has a 40×0.9 numerical aperture objective lens with an additional back-end 4-f magnification system consisting of a $f = 75$ mm and a $f = 400$ mm lens to provide a total magnification of $\sim 213\times$. Additional details of the optical imaging system used to obtain the interferometric and non-interferometric images are described in [1, 8, 23].

It is time-consuming to obtain SEM images shown in Figs. 1(a) - 1(d) and this process can also leave the dark charging stains on the wafer. On the other hand, finding defects and classifying their shapes and sizes directly from the optical intensity images of Figs. 1(e) and 1(f) is not possible because of the background noise.

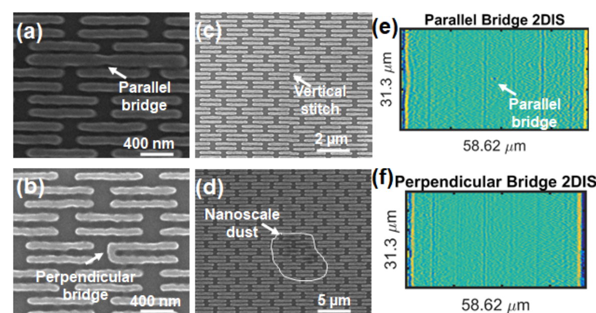

Fig. 1. SEM images of a typical (a) parallel bridge defect, (b) perpendicular bridge defect, (c) e-beam stitching error, and (d) region with nanoscale dust. Microscope image after 2DIS post-processing of the (e) parallel and (f) perpendicular bridge defect dies.

3. Framework and inference

Figure 2 is a summary block diagram of the machine learning technique to classify defect signals in the intensity images. We use a post-processed microscope image and information from simulated defects for training and optimizing the process for generating the synthetic defect image library, $\mathscr{L}_{syn}$. Optimization enables the model to generate $\mathscr{L}_{syn}$ for classifying defects in previously unexposed microscope images. We will use a top-down approach to define key algorithm parameters first, then overview the inference process that uses the optimized $\mathscr{L}_{syn}$ obtained in the learning process. Next, we will explain the learning process and then explain the post-processing of optical images along with our unique hypothesis about noisy optical images. Finally, we proceed to discuss data transformation for generating the synthetic set.

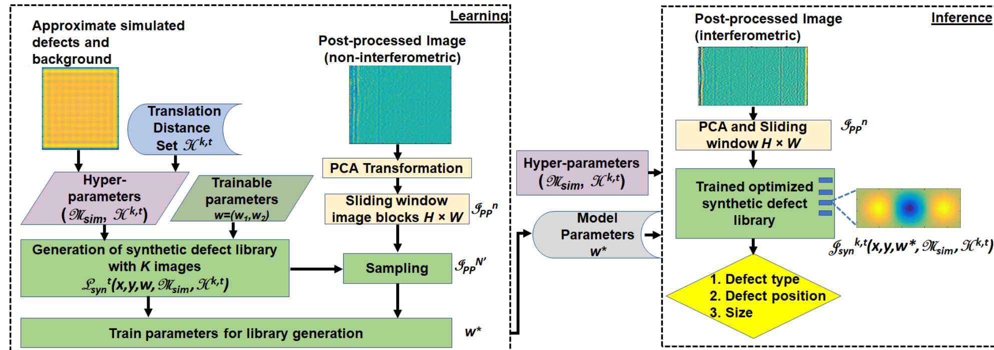

Fig. 2. Overview of the proposed approach. Given approximate simulated defect images, the set of random numbers, and a microscope image, we learn model parameters, w^*, which can be used to generate an optimized synthetic defect library, $\mathscr{L}_{syn}$, which can classify an experimental image as showing a defect or no defect. Section 3.6 provide image generation algorithm details.

3.1. Algorithm parameters

As shown in Fig. 2, two hyper-parameters sets $\mathscr{M}_{sim}$ and $\mathscr{H}^{k,t}$, whose values are fixed before the learning process begins and the trained model parameters $w^* = (w_1, w_2) \in \mathbb{R}^2$ are the inputs to the inference. $\mathscr{M}_{sim}$ is a noise-free set consisting of one simulated image of a defect of a given shape and size plus the peak intensity values from this image and from four other simulated images of defects of the same shape but slightly different sizes. We use Monte Carlo to mimic random experimental errors with $\mathscr{H}^{k,t} = (h_1^{k,t}, h_2^{k,t}) : h_1^{k,t}, h_2^{k,t} \in \mathbb{R}$. We generate $k \in \{1, \dots, K = 125\}$ Gaussian random number pairs for a single trial t and perform $t \in \{1, \dots, T = 20\}$ trials.

3.2. Inference

The goal of inference is to classify the given post-processed and PCA-transformed noisy microscope image block $\mathscr{I}_{PP}^n$ as a defect or no-defect. Here, $\mathscr{I}_{PP}^n(x, y) \in \mathbb{R}^{H\times W}$ represents $n \in \{1, \dots, 10^6\}$ post-processed and transformed intensity images in the Cartesian co-ordinate system with the height H and width W. We introduce labels $z_n \in \{0, 1\}$ to denote the ground truth in a given $\mathscr{I}_{PP}^n$. Here, $(z = 0)$ denotes absence of a defect and $(z = 1)$ denotes presence of a defect. Additional inference goals include classifying the defect by shape and size. In a trial t, we use the synthetic defect images $\mathscr{I}_{syn}^{k,t}(x, y, w^*, \mathscr{M}_{sim}, \mathscr{H}^{k,t}) \in \mathscr{L}_{syn}^t$; $\mathscr{I}_{syn}^{k,t} \in \mathbb{R}^{H\times W}$ obtained after optimization in the learning process. We apply PCA to ${\mathscr{L}_{syn}}^t(x, y, w^*, \mathscr{M}_{sim}, \mathscr{H}^{k,t})$ during the process described in Section 3.6 to obtain the eigenvectors $\mathbf{v}$ and weight vectors ${\Gamma_{syn}}^{k,t}(\mathbf{v}, w^*, \mathscr{M}_{sim}, \mathscr{H}^{k,t})$. We similarly transform $\mathscr{I}_{PP}^n(x, y)$ to obtain ${\Gamma_{syn}}^{k,t}(\mathbf{v})$. For each

$\mathscr{I}_{PP}^{n}(x,y)$, we select a single $\mathscr{J}_{syn}{}^{k^*,t}(x,y,w^*,\mathscr{M}_{sim},\mathscr{H}^{k^*,t})$ as its closest match, i.e., the image with minimum Euclidean distance (ED) from $\mathscr{I}_{PP}^{n}(x,y)$ in the eigen space:

$$k^* = \arg\min_k \left\{ \lambda_s{}^{-2} \left\| \Gamma_{PP}{}^{n}(\mathbf{v}) - \Gamma_{syn}{}^{k,t}(\mathbf{v}, w^*, \mathscr{M}_{sim}, \mathscr{H}^{k,t}) \right\|^2 \right\}, \tag{1}$$

$$ED^{n,t}(\mathscr{I}_{PP}^{n}, \mathbf{v}, w^*) = \lambda_s{}^{-2} \left\| \Gamma_{PP}{}^{n}(\mathbf{v}) - \Gamma_{syn}{}^{k^*,t}(\mathbf{v}, w^*, \mathscr{M}_{sim}, \mathscr{H}^{k^*,t}) \right\|^2. \tag{2}$$

Here, λ_s is the eigenvalue corresponding to a synthetic defect eigenvector. The eigen vectors are selected such that 99% variance is retained in PCA. By dividing the distance by λ_s^2, we reduce the effect of noise present in the synthetic images. $ED^{n,t}$ measures the similarity of the generic features between the experimental and synthetic defect intensity images. Because of the eigen space transformation, $ED^{n,t}$ depends not only on $\mathscr{I}_{PP}^{n}$ but also on the entire synthetic defect image set, $\mathscr{L}_{syn}{}^{t}$. The inference for $\mathscr{I}_{PP}^{n}(x,y)$ is performed using:

$$\hat{z}_n = \varphi \left[\left\langle \varphi \left(ED^{n,t} \leq c \times \min_{n' \in \{1,\ldots,10^6\}} \{ED^{n',t}\} \right) \right\rangle_t \geq \frac{1}{2} \right]. \tag{3}$$

where φ is the indicator function on the truth of a proposition. The value of c is explained in Section 4. Intuitively, $\hat{z}_n$ is computed by calculating generic similarities between the experimental and synthetic defect images using the principal components of the optimized synthetic images over T repetitions of varying noise.

3.3. Learning

As described in Fig. 2, we need a dataset $\mathscr{G} = \left\{ (\mathscr{I}_{PP}{}^{n}, z_n)_{n=1}{}^{N'}, w, \mathscr{M}_{sim}, \mathscr{H}^{k,t} \right\}$. Here, N' are sampled blocks obtained in Section 4 such that one image is of the intentional defect, denoted with an index n_0, and all the other blocks are no-defect images. The goal of the learning stage is to obtain $\mathscr{L}_{syn}{}^{t}(x,y,w,\mathscr{M}_{sim},\mathscr{H}^{k,t})$ such that for a $w = w^*$, the principal components of this synthetic set sufficiently capture the variance related to the experimental defect features. Based on distinguishing features, we want to be able to identify defect block $(\mathscr{I}_{PP}{}^{n_0}, z_{n_0} = 1)$ as a defect in comparison to the defect-free blocks $\{(\mathscr{I}_{PP}{}^{n}, z_n = 0) : \forall n \neq n_0\}$. The approach is to transform images such that those of the same class are more alike in the projected space. We optimize the synthetic image generation process to obtain w^* using a distance function f that measures the fraction of trials, i.e, the probability, for which the intentional defect is detected:

$$f_w(\mathscr{G}) = \left\langle \varphi \left[ED^{n_0,t}(\mathscr{I}_{PP}^{n_0}, \mathbf{v}, w) \leq c \times \min_{n' \in \{1,\ldots,N'\}} \{ED^{n',t}(\mathscr{I}_{PP}^{n'}, \mathbf{v}, w)\} \right] \right\rangle_t. \tag{4}$$

We hypothesize that the approximate defect information from the synthetic defect set provided in $\mathbf{v}$ would be sufficient to reject other no-defects after the learning and we aid the learning process by utilizing the ED for the no-defect block that most closely resembles the synthetic defect set. The factor of c is the margin that accounts for the fact that the simulated data has only approximate defect feature information. We define the loss function for optimization in the following manner. We re-scale the probability using a sigmoid function:

$$P_w(z_{n_0} = 1|\mathscr{G}) = \frac{1}{(1 + \exp[-f_w(\mathscr{G})])}. \tag{5}$$

The first loss function for the learning stage is defined by the negative log-likelihood:

$$Loss_{1,w} = -\log \left[P_w(z_{n_0} = 1|\mathscr{G}) \right]. \tag{6}$$

Optics EXPRESS

The loss function is concise because of the use of principal components. It is expected that the optimized loss function would satisfy the condition in Eq. (4) for at least $\frac{T}{2}$ of the trials. Such an optimization would indicate that we can usually classify images correctly despite the measurement errors. The synthetic data set contains features for defect size estimation as well. We can select a second loss function for including explicit incorrect defect size based information:

$$Loss_{2,w} = -\log\left[P_w(z_{n_0} = 1|\mathscr{G})\right] - \log\left[1 - P_w(z_{n_0} = 1|\mathscr{G}')\right]. \tag{7}$$

Here, $P_w(z_{n_0} = 1|\mathscr{G}')$ is the re-scaled probability of identifying a defect using a synthetic library created with incorrect simulated defect sizes. We have compared the performance of three different loss functions in this paper: (1) Category 1: Eq. (6) with an added condition that we label any synthetic image with a relatively large intensity variation as an undesired defect (i.e., $z = 0$). We achieve this by incorporating a condition that the index of the synthetic image matching the experimental image must be below 78 because images with such indices represent smaller intensity variations. Hence, we have labeled synthetic image data, instead of including an explicit size-based constraint. (2) Category 2: We use Eq. (7). (3) Category 3: We use Eq. (7) with different start points and c value combinations than the previous two categories. The outcome of optimization for the loss functions from Table 1 is described in Table 2. The optimization algorithm is explained in Appendix A. The value of c is further tuned during the Validation Stage.

Table 1. Loss functions

Categories	Start point (w_2,w_1)	Value of c
Category 1	(2,3)	1.0
Category 1	(1,3)	1.2
Category 2	(2,3)	(z=1) 1.0, (22-nm , z=0) 1.1
Category 2	(1,3)	(z=1) 1.0, (22-nm , z=0) 1.2
Category 3	(4,5)	(z=1) 1.1, (22-nm , z=0) 1.1
Category 3	(2,4)	(z=1) 1.0, (22-nm , z=0) 1.1

3.4. Post-processing of microscope images

Experimental microscope images are extremely noisy. This section presents the post-processing scheme to denoise the experimental images and it also presents a crucial hypothesis for synthetic image generation. Figures 1(e) and 1(f) show the post-processed experimental optical images of the BX67 and BY67 defects. We observed that the central defect in the experimental images has different SNR in each camera frame because of the presence of noise and other systematic errors. We also observed variations in the SNR for different vertical stitching error regions even under the same experimental conditions. Hence, we concluded that different types of defects would display different scattering signatures for a given experimental setting not only because of differences in their specific sizes, tilts, and deviations, but also because of the added noise. During the experiment, spatially dependent noise from the scattered signal of the background pattern of lines is present in each frame. A single camera image $I_{image}(x, y)$ measures the interference between this dominant scattering field of the background, $\mathbf{E}_b(x, y)$, and the weak scattering field produced by the defect, $\mathbf{E}_d(x, y)$ in x, y coordinate system. We introduce a term M_{bd}, which will be used throughout this section:

$$I_{image}(x, y) = I_b(x, y) + 2Re\left[\mathbf{E}_b(x, y) \cdot \mathbf{E}_d(x, y)\right] + I_d(x, y), \tag{8}$$

$$I_{image}(x, y) \approx I_b(x, y) + M_{bd}(x, y). \tag{9}$$

Here, $I = |\mathbf{E}|^2$ denotes the intensity of the respective fields and $M_{bd} = 2Re\left[\mathbf{E}_b \cdot \mathbf{E}_d\right]$ is the mutual interference of the background and defect fields. The mutual interference, M_{bd}, varies

from frame to frame both because of small fluctuations in the intensity of the light source and because of the LER and other patterning irregularities. Further, the periodicity of the unit cell of the pattern is of comparable size to the diffraction limit and so the spatially varying interference is partially resolved in the collected images. We can model both of these effects by writing, $M_b(x, y) = M_{b0} + \Delta M_b(x, y)$, where M_{b0} is the average intensity of the interference, which is mostly affected by the fluctuations in the intensity of the light source, and $\Delta M_b(x, y)$ is the spatially varying remnant of the partially resolved interference image, which is affected by the LER and other patterning irregularities. These two effects result in defect signatures that appear to have distortions, i.e., the optical image of the defect signature looks different in each frame. The post-processing method was designed to minimize frame-to-frame variations in the background, $I_b(x, y)$ from Eq. (9), while preserving the defect signal [7]. Images are collected as the sample is translated a distance dx between frames by a mechanical stage. Then, the post-processed image that forms a tripole pattern for a defect signal is calculated in software as:

$$I_{PP}(x, y) \equiv I_{image}(x + dx, y) - 2I_{image}(x, y) + I_{image}(x - dx, y). \tag{10}$$

To remove the dominant background intensity $I_b(x, y)$, the translation distance dx is chosen to be an integer number of periods of the background pattern. Thus,

$$I_{PP}(x, y) \approx \delta M_{bo}(x, y) + M_{bd}(x + dx, y) - 2M_{bd}(x, y) + M_{bd}(x - dx, y). \tag{11}$$

Here, δM_{b0} is a constant offset in intensity that would be zero if the light source intensity and the interference did not vary from frame to frame. Although we have removed the dominant spatially varying background signal, there can still be noticeable variations in the interference between the defect signal and the background M_{bd} in different frames. Thus, most but not all of the frame-to-frame noise from the background has been removed. At the defect location, the 2DIS image has what we call a tripole pattern, which is a spot in the center and two spots with an opposite sign located at a distance of $\pm dx$ from the center. All the experimental images used in this paper are post-processed and form this tripole pattern. The image obtained after post-processing is $I_{PP}(x, y)$ that depends on the frame-to-frame varying mutual interference, M_{bd}. This frame-to-frame varying formulation of M_{bd} in Eq. (9) and Eq. (11) provides the physical explanation for noisy image formation, i.e., it is because of the interaction of physical features and the background noise. Some noise remains in the experimental images even after the post-processing. We apply PCA to further reduce noise. Experimental and simulation images have different sizes and normalizations. We crop the experimental image to $\sim 10^6$ sliding window image blocks of size B, the value of which is determined in the Section 4.

3.4.1. Physical insight and assumptions

We form critical assumptions on the basis of the above hypothesis and formulations. First, we assume that the noise in the background pattern affects the peak value of the scattered intensity of a defect differently in each experimental image. Second, we assume that the peak scattered intensity values of a defect contain the most important information about the defect size. This is reasonable because the strength of Rayleigh scattering decreases rapidly as the defect size decreases. The third assumption is how the scattering peak intensity, which depends on the defect size and shape, interacts with mutual interference $M_{bd}(x, y)$. Because M_{bd} varies from frame to frame, it appears that the peak intensities in different frames for a given defect correspond to defects of slightly varying sizes. When we take a 2nd order difference to form a tripole pattern, we typically obtain an asymmetric tripole pattern with different peak intensities at each pole. These assumptions about the outcome of the interaction between physical defect features and background noise place importance on the distinguishing physical features of the defects in the noisy images. We will utilize these assumptions for generating synthetic images.

3.5. Input for synthetic dataset generation

To obtain synthetic image data set, we first need to perform EM simulations of different types of defects to obtain $\mathscr{M}_{sim}$. See references [24–26] for details of the simulation method. Figure 3(a) shows a zoomed in view of the simulation geometry for the parallel bridge defect. Figure 3(b) shows the resulting simulated optical intensity image. In order to evaluate whether our machine learning technique can accurately detect, classify, and size the defects, simulations were performed for both types of defects with sizes ranging from 18 nm to 46 nm with a 2-nm step using the same experimental settings for both types of defect. The simulated data contains the approximate size and shape of the defect for the labeled training experimental image. We choose to simulate the 32-nm defect. In order to include the effect of M_{bd} on measurements, we use the peak intensity values from the four slightly varying defect sizes: 28, 30, 34, and 36 nm. To account for the defect dimensions in the training sample, we simulated across a size range for the defect width that contains the nominal design value and is wide enough to account for the typical fabrication error margins. The 32-nm image is called $M_{seed}(x, y)$ and $max(M_j); j \in \{1, 2, seed, 4, 5\}$ are the peak intensities for images $M_j(x, y)$. We will combine these to form $\mathscr{M}_{sim} = \{M_{seed}, \{max(M_j)_{j=1}{}^5\}\}$.

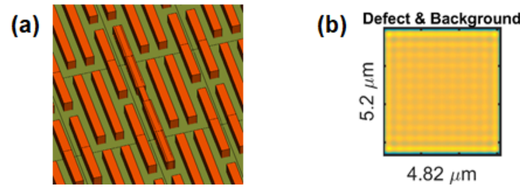

Fig. 3. (a) Zoomed in simulation geometry for a parallel bridge defect. The field of view is 900 nm × 900 nm. (b) Raw simulation intensity image.

Although we used powerful EM algorithms, our three-dimensional simulation volume was limited because of memory constraints. Thus, the raw simulated images have artifacts because of domain truncation. In addition, the simulations do not capture the random noise and systematic noise in the experimental data. As shown in the Figs. 1(a) and 1(b) SEM images, the LER is a significant fraction of the line width. Moreover, the exact 3D cross-sectional shape of the structure (e.g., rectangular or trapezoidal) is not known. Hence, it is extremely challenging to model the sample accurately.

3.6. Processing for synthetic dataset generation

The objective of this section is to describe the process of generating the trainable synthetic defect library shown in Fig. 2 using hyper-parameters $\mathscr{M}_{sim}$ from Section 3.5, trainable parameters $w = (w_1, w_2)$ (determined during the learning stage), and a microscope image from Section 3.4. We will introduce the details of the second hyper-parameter $\mathscr{H}^{k,t}$ subsequently. We perform five steps to generate a diverse library of synthetic images. Figure 4 shows key intermediate results.

Step 1 (Peak intensity transformations): To address noise and artifacts in the raw simulated data, we extract defect-specific info, crop and scale the image, and observe the variations in peak intensities. We subtract the background image (i.e., pattern with no defect) from each image of the pattern with a defect to obtain a set of $M_{bd}(x, y)$ for simulation images M_j from $\mathscr{M}_{sim}$. We crop the images to remove boundary artifacts from domain truncation. See Fig. 4(a). Next, we zero pad the images. We call the output image intensities M_j'. The following linear transformation is performed on images using trainable parameters w_1, w_2 using the assumptions from Section 3.4.1 to form a set of five numbers:

$$\mathscr{S} = \left\{w_1 \max(M_j')\right\}; j \in \{1, \ldots, 5\}. \tag{12}$$

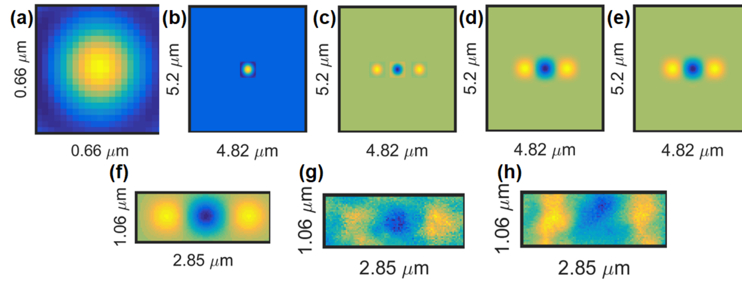

Fig. 4. Images at various steps of synthetic library generation process. Processing of a simulated image showing the (step 1) (a) extracted defect and (b) processed defect, (step 2) (c) 2nd order difference, (step 3) (d) Gaussian blur and (e) Gaussian noise, and (step 4) (f) Cropped final synthetic defect image that has a tripole pattern. Post-processed experimental image blocks after cropping and normalization, showing (g) a parallel defect image. (h) A typical edge-based signal that needs to be rejected as a parallel defect.

We multiply the seed image (e.g., 32-nm) with a different trainable constant, w_2, and denote this image as $w_2\ M'_{seed}$. We offset the seed image to match the peak intensities of $\mathscr{S}$. Next, we generate image library $\mathscr{A}_j$; $j \in \{1, 2, 3, 4, 5\}$ that is the outcome of Step 1:

$$\mathscr{A}_j(x, y) = w_2 \cdot M'_{seed}(x, y) + (\delta M_{bo})_j\,, \tag{13}$$

$$(\delta M_{bo})_j = w_1 \cdot \max\,(M'_j) - w_2 \cdot \max\,(M'_{seed}). \tag{14}$$

Thus, our set consists of the seed image normalized by w_2 with five different constant offset values $(\delta M_{b0})_j$. The differences in our transformed images are thus proportional to w_1:

$$\mathscr{A}_j - \mathscr{A}_k = w_1 \cdot \max(M'_j) - w_1 \cdot \max(M'_k). \tag{15}$$

Eq. (13) and Eq. (14) provides constraints for the optimization process of the learning stage. We restrict $w_1 > w_2 \geq 1$ to ensure that each intensity image in the data set has non-negative values at each pixel according to Eq. (13) and Eq. (14). The lower bound on w_2 also preserves adequate spatial variation in the intensity images. The images in Eq. (13) are physically meaningful if the offset $(\delta M_{b0})_j$ is small compared to $w_2 \cdot M'_{seed}(x, y)$. This puts an upper limit on w_1. Zero-padding makes the images in $\mathscr{A}_j$ large enough for performing 2nd order difference similar to Eq. (10). See Fig. 4(b). Next we used data conversion to create intensity variation in the images by converting the data type from double to unsigned 16-bit integers and back to double to mimic the processing that occurs with the experimental images.

Step 2 (2nd order difference and $\mathscr{H}^{k,t}$): To obtain the tripole defect pattern, the microscope stage is translated. However, the translation distance is not perfectly repeatable and varies between 24-32 pixels. To emulate this experimental error, we used a Monte Carlo approach where we generated $k \in \{1, \ldots, K = 125\}$ Gaussian random numbers, using $D_{mean} = 28$ pixels for the mean translation distance and 2 pixels for the standard deviation. We obtained hyper-parameter $\mathscr{H}^{k,t}$ of 125 Gaussian random number pairs in trial t generated for the microscope image translation distance. We conduct $T = 20$ trials for each classification. We shifted the simulated frames for the defect by $\mathscr{H}^{k,t}$ and calculated the 2nd order difference to obtain 5×5×5=125 different tripole images. There are 125 images because we include all combinations of the 5 intensity variations $\mathscr{A}_j$ for each of the 3 different image frames. We choose 5 images for two reasons. Firstly, we need to distribute the difference $w_1 M'_j - \mathscr{A}_j$ for the entire set symmetrically. Hence, we need odd number of images to select the middle image as a seed image. Secondly, we require some minimum number of images to apply PCA and extract meaningful features out of it.

Image stitching is not performed for the simulation images because they are already free of any nonuniform illumination errors. See Fig. 4(c).

Step 3 (Gaussian blur and Gaussian noise): There are unphysical edge effects because of the zero padding. Hence, we apply a Gaussian blur with the width comparable to the diffraction limit to not lose features. The blur improves robustness to noise. The peak intensity trends in the images are thereby disturbed. We scale and crop to focus on defect peak intensities. We apply a kernel that captures noise characteristics along with anti-aliasing and peak intensity-based linear transforms. See Figs. 4(d)-4(e), and Fig. 5. The steps are adapted from [27–30].

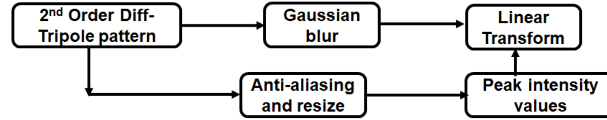

Fig. 5. Gaussian noise transformation.

Step 4 (Cropping, variations): We crop the simulation to match the block size of the experimental images. This blocks size is determined in the sampling stage. The cropping is done in such a way that it encourages more variance by comparing the left and the right tripole distance average with the D_{mean}. The cropping pixel is shifted to the left by one pixel when the right tripole distance average is more than the mean tripole distance. We perform data conversion between double to unsigned 16-bit integers to double as in Step 1. The outcome of the Step is a noise-added synthetic defect image set $\mathscr{L}^t_{syn}(x, y, w, \mathscr{M}_{sim}, \mathscr{H}^{k,t})$. See Fig. 4(f).

Step 5 (PCA transform): The resulting dataset has large dimensionality and features. We apply PCA to obtain a transformed representation of the dataset and to extract generic features of the synthetic image set and obtain eigen vectors $\mathbf{v}$. The weight vector of the transformed synthetic microscope defect image set is $\Gamma_{syn}{}^{k,t}(\mathbf{v}, w, \mathscr{M}_{sim}, \mathscr{H}^{k,t})$.

The post-processed and transformed microscope image blocks such as Figs. 4(g) and 4(h) are also transformed in the same eigen space $\mathbf{v}$ to obtain $\Gamma_{PP}{}^{k,t}(\mathbf{v})$ before the learning and inference stages. We use sampled blocks from Section 4 for the learning stage and all the sliding window image blocks for the inference stage. The trainable parameters generate diversity in the synthetic library because w_2 affects the mean intensity and (w_1, w_2) affect the variations in the intensity.

4. Implementation

Before the learning stage, we sampled the input image blocks to obtain N' representative image blocks using $\mathscr{I}_{PP}{}^n; n \in \{1, \ldots, N \approx 10^6\}$ of size B using the following method for the image in Fig. 6. Figure 6 shows a 2DI experimental image for the parallel bridge defect before applying PCA (SNR = 13.6 dB) and after applying PCA (SNR = 20.9dB). Stitching errors and noisy edges were less prominent than the central defect.

Fig. 6. Parallel defect image with 2nd order difference. (a) Before and (b) after PCA.

We generated $\mathscr{L}_{syn}{}^t(x, y, w, \mathscr{M}_{sim}, \mathscr{H}^{k,t})$ images using the 32-nm parallel defect seed image. In sampling stage 1 , we obtained 51 image blocks that are either the real defect or quite similar to it. As expected, the detection rate of the real defect was below 50% for our the intermediate

model. These 51 blocks are clearly the blocks that are most critical for the algorithm to learn how to differentiate. We test the effectiveness of this sampling using the Category 1 loss function in sampling stage 2. We have used modified gradient descent optimization from Appendix A.

Algorithm 1 Outline of sampling stage 1.

1: **Gradient descent optimization**
2: Input $(w_1, w_2) = (1.5, 1.0)$; Initial step size 0.05; $\alpha = 0.05$; Trials $T = 1$; $N \approx 10^6$
3: **repeat** ▹ minimization using gradient
4: $Loss_w = \langle \{ED^{n_0,t}(\mathscr{I}_{PP}{}^{n_0}, \mathbf{v}, w)\} \rangle_{t=1\, to\, T}$
5: Update $w_i{}^{it+1} \Leftarrow w_i{}^{it} - \alpha \hat{\nabla}_i Loss_w$ ▹ $\hat{\nabla}_i$ is the numerical partial gradient w.r.t. w_i
6:
7: **until** Iterations it exceed 100; w_i follows the constraints in Section 3.6
8: Outcome 1: $w^* = (7.39, 4.52)$ intermediate synthetic defect model that corresponds to the minimum ED^{n_0}
9: **Model testing**
10: Input: w^* model; trials $T = 20$; $N \approx 10^6$; $c = 2.1$ later $c = 1.2$ to reduce redundancy due to sliding window blocks :Execute Inference Eq.(3)
11: Outcome 2: $N' = 51$ image blocks that are either the real defect or quite similar to it
12: **Final outcome** $B^* = 34 \times 91, N' = 51$

Algorithm 2 Outline of sampling stage 2.

1: **Modified gradient descent optimization**
2: Input: $(w_1, w_2) = (3, 2)$ with an initial step size of 1.0; Trials $T = 20$; $N' = 51$ sampled blocks from Sampling Stage 1
3: **repeat** ▹ minimization using modified gradient
4: Category 1 Loss function from Section 3.3
5: Update w_i using the update rules from Appendix A
6:
7: **until** iterations it exceed 30; w_i follows the constraints in Section 3.6
8: Outcome 1: $w^* = (3.2529, 2.2342)$ intermediate synthetic defect model
9: **Model testing**
10: Input: w^* model; trials $T = 1$; $N \approx 10^6$; $c = 1.1$ to reduce redundancy due to sliding window blocks :Execute Inference Eq.(3)
11: Outcome 2: True defect $z = 1$ and 2 additional defect-like positions detected;
12: Add these two defect-like positions to original $N' = 51$ to obtain $N' = 53$
13: **Final outcome** $N' = 53$ sampled image blocks

4.1. Learning optimized models and validation

We used $N' = 53$ sampled blocks and applied the loss functions from Section 3.3 to obtain the optimized models. Table 2 shows the optimized results. We used 22-nm simulated seed defect to generate incorrect defect size data-set $\mathscr{G}'$ for Eq. (7). The simulated defect seed information is described in Appendix B.

It is important to note the large initial values for the cost function, i.e., the starting programs had low defect detection rates. By design, the algorithm is very sensitive to and exhibits non-linear behavior with variations in w_1 and w_2 . Changes in the third decimal digit of w_1 or w_2 are significant because the algorithm steps amplify certain effects in the synthetic library images and the resulting eigen vectors. After optimization, we analyzed the results

Table 2. Models obtained during training stage. The experimental sample used for training consisted of N' non-interferometric image blocks for the parallel defect die.

Categories for the loss function	Loss function at start point: Cost(w_2, w_1)	Loss function at local minimum: Cost(w_2, w_1)	Model label id
Category 1	Cost(2,3) = 0.513	Cost(1.9368,3.0164) = 0.387	1A
Category 1	Cost(1,3) = 0.474	Cost(1.035, 3.00) = 0.371	1B
Category 2	Cost(2,3) = 1.563	Cost(1.9991,3.0124) = 1.153	2A
Category 2	Cost(1,3) = 1.716	Cost(1.2947,3.2558) = 1.153	2B
Category 3	Cost(4,5) = 1.369	Cost(3.9684,4.9959) = 1.338	3A
Category 3	Cost(2,4) = 1.775	Cost(1.9862,3.9850) = 1.168	3B

using $c = 2.1$. Next, we reduced the number of redundant image blocks in the result to obtain $1.0 \leq c \leq 1.5$ for all the models. Model 3A works well for $1.5 \leq c \leq 2.1$. The allowed values of $c \in \{1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1\}$. Beyond this range, data redundancy due to sliding window images and false detection rate diminishes the performance.

For the validation stage, we used all $N \approx 10^6$ blocks instead of $N' = 53$ blocks. The outcome of the models is a list of image block positions that are determined as defects and their ED. Then, we use $c = 2.1$ on the outcome list to reduce data redundancy due to the sliding window method. Next, we used following criteria to lower the value of c further and reduce the data redundancy without lowering the maximum detection frequency: the same value of c from the allowed c values is selected for the models of the same category. Model 3A detects the central defect in the whole die image during the training-validation stage for $c = 1.5$ with higher false positive rates. We select $c = 1.1$; as a result all other models performed well and had 100% precision (i.e., no other defects, e.g., stitching error or nanoscale dust, were detected above the 50% detection rate threshold). Figure 7 shows the results for the precision and recall rates of the models.

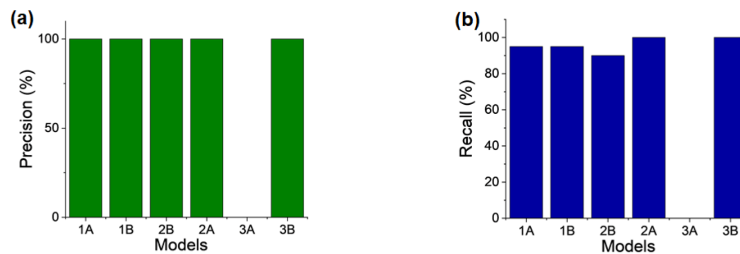

Fig. 7. (a) Precision, and (b) Recall for the models from the three different categories for the parallel bridge defect for the entire die image.

It is important to evaluate the predictive ability and robustness of each model for previously unused combinations of input simulation and experimental images. We used the perpendicular defect simulated data with another parallel defect experimental frame from the same die. This frame is shifted horizontally and has different noise than in the previous case. As expected, the parallel bridge defect is not identified in this case as a perpendicular bridge defect. The models are robust against shape misclassification. For model 2B, there is a defect detected with 55% detection rate in a block with corner pixel location (819, 211). This detected defect could be due to nano-scale dust and needs further investigation.

This program was executed on a desktop machine with an Intel Xeon 2-processor CPU with 28 cores and 128 GB memory. It was developed in MATLAB using the parallel computing toolkit. It takes a minute to generate and store the synthetic image library, which is a one-time activity for

Optics EXPRESS

a given defect shape and size classification. It takes 2 minutes to collect, process, and reconstruct through PCA all $N \approx 10^6$ sliding window experimental images on a disk. It takes an additional $2-3$ minutes to format the eigen distance information for these 10^6 images and perform the moving average of the reconstructed sliding window images. The reconstructed images are used for visualization purposes. In the future, batch processing and higher computing power resources can reduce the time significantly.

5. Inference results

5.1. Shape classification

We learned the models using sampled unstitched non-interferometric microscope image containing a parallel bridge defect. We validated the models using the unsampled original image. In addition, we validated the robustness of models against shape misclassification using another unstitched non-interferometric microscope image. We use the criteria developed in the validation stage for lowering the value of c for the outcome of the models and obtain the defect positions and frequency of detection for all the trials.

We evaluated the models for shape based prediction using the 2DIS parallel defect die image and a 32-nm 'H' shaped synthetic set for the defect. The results of initial PCA are displayed in Fig. 8. The SNR was 16.6 dB initially and 20.3 dB after PCA. We evaluated the confusion matrix of the selected classifiers. For testing with the 'H' shape image, we used $c = 1.2$ on the outcome defect list. For many models, the detection rate was above 50%, i.e., certain wafer regions were classified as containing a defect. Table 3 shows the results. The SEM image in Fig. 8(c) shows that there is a stained region at the left edge. Corner parameters of (453, 13), (516, 14), and (548, 71) in Table 3 correspond to different parts of the stain. The 'H' shape defect here means, unevenness in the width, separation, or tilt of the bars in the underlying pattern.

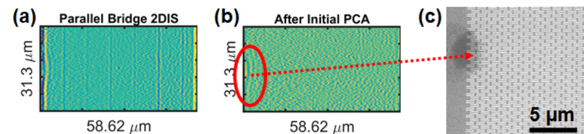

Fig. 8. Parallel defect image with 2DIS processing. (a) Before and (b) after applying PCA (c) SEM image of the region displaying stain.

Table 3. Defect detection with 32-nm seed.

Model	Detection rate	Defect corner id
1A	70%	Left edge (548, 71)
1B	100%	Left edge (459, 13) and Central defect
2A	85%	Left edge (516, 13)
2B	0%	No defect detected
3A	85%	Left edge (516, 14)
3B	70%	Left edge (453, 13)

The imaging system cannot directly resolve these irregularities, but the models are sensitive to them and have revealed to us the previously undetected stain region. To validate robustness against shape mis-classification, we used the parallel defect synthetic set on the perpendicular defect experimental data. For all six models, the central defect was never detected. Instead, the models find the stitching error region defects or edge defects because of their similarity to parallel bridge defects. See Appendix C for details.

Optics EXPRESS

5.2. Shape and size classification

The size of the nanoscale defect is related to the intensity of the defect signal in the experimental image. The models work by targeting the differences in physical features embedded in the noisy image. Thus, we will consider both precision and detection rate values for size-based detection in the noisy images. Figure 9 shows how the different models estimate defect shape and size in different types of experimental images.The performance for size detection is better for less noisy 2DIS interferometric parallel bridge defect even though the models were not trained with any interferometric images. Figures 9(j) and 9(m) shows a typical Euclidean distance map for model 2B for the parallel defect and for model 3B for the perpendicular defect, respectively. Compared to the parallel defect, the perpendicular defect has a weaker scattering signature as can be seen by comparing Figs. 9(k) and 9(n). The reconstructed images in Figs. 9(l) and 9(o) are obtained by projecting Figs. 9(k) and 9(n), respectively, into the eigen space of the synthetic defect images, and by adding the mean image of the respective synthetic defect library. The defect optical images are better reconstructed using the synthetic image eigen space in comparison to the rest of the optical images; thereby showing the effectiveness of the PCA transformed optimized synthetic images. The perpendicular defects were not visible even after 2DIS processing and initial denoising using PCA, which is why our machine learning technique is needed to provide the eigen distance map for detecting the perpendicular defect. As shown in Figs. 9(g) - 9(i), the models are able to successfully detect and classify the perpendicular bridge defect. The detection rate for the 32-nm synthetic perpendicular defect library for the perpendicular defect image is more than the 50% threshold and thus the algorithm correctly classifies the defect location, shape, and size. The other defect positions may have been identified as a 22-nm defect because those regions have similar characteristics to 22-nm wide defects. It is important to note that the accuracy in all the cases is close to 100% due to high true negative rate. See Appendix D for details.

Table 4 summarizes the capabilities of the six models. The postprocessed optical images used in Table 4 are obtained from both interferometric and non-interferometric measurements, and have varying SNR levels from 9.9 dB to 23.5 dB. Despite this variation in the SNR, each model correctly identifies the shape of the defect. Because size-based and shape-based inferences are combined, size rejection requires stronger targeting of size-based differences compared to shape-based similarity in a noisy image. All models generalize because they can target size-based and shape-based feature differences.

6. Summary

It is quite difficult even for image experts to detect a defect, classify its type, and estimate its size from the microscope images because the defect signal is indistinguishable from the background. We therefore developed a novel interpretable machine learning algorithm for automatic detection and classification of defects that uses only a few training samples. We interpret the algorithm as a multi-prong approach to mitigate simulation artifacts and target distinguishing defect features buried in the mutual interference of the background and defect fields. We have utilized the denoising abilities of PCA and overcome the challenges of using principal components for classification and of the limited availability of experimental defect images. We developed a modified gradient descent algorithm for efficient optimization of loss functions.

One downside of the method is the long overall computation time associated with the sliding window step. A second key limitation is that for generalizing this method to other types of images, e.g., phase images, the researcher must add physical insight appropriate to the image type, i.e., the function of the trainable parameters must be decided.

Nevertheless, we successfully validated and tested the models using images with different experimental configurations (interferometric versus non-interferometric) and with different defect types (parallel versus perpendicular bridge). These images had varying noise levels and different

Optics EXPRESS

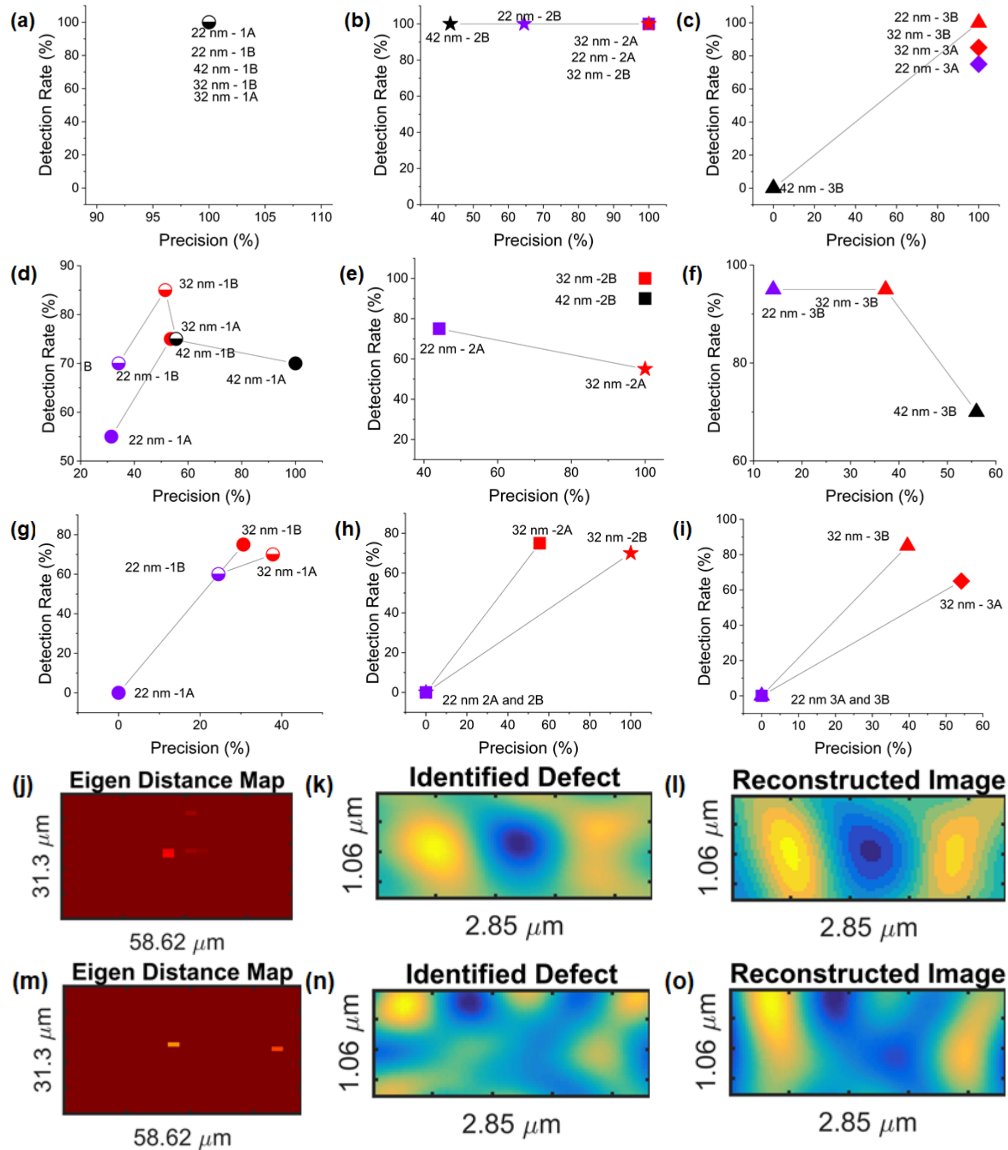

Fig. 9. Shape and size classification for 2DI non-interferometric image with parallel bridge defect for models (a) 1A, 1B (b) 2A, 2B (c) 3A, 3B. Classification for 2DIS interferometric image with parallel bridge defect for models (d) 1A, 1B (e) 2A, 2B (f) 3A, 3B. Classification for 2DIS interferometric noisy image with perpendicular bridge defect for models (g) 1A, 1B (h) 2A, 2B (i) 3A, 3B. All models successfully detect the perpendicular defect when using the 32-nm seed because their detection rate was above the 50% threshold that was set for deciding defect versus no-defect. (j) Eigen distance map identifies the parallel defect indicated with the light color box. It has the smallest Euclidean distance and thus is correctly identified. (k) Parallel defect identified. (l) Reconstructed defect. (m) Eigen distance map identifies the perpendicular defect indicated with the light color box. This region has the smallest Euclidean distance and thus is correctly identified. (n) Identified region for the perpendicular defect in the optical image is buried in the noisy background. (o) Reconstructed perpendicular defect image.

Table 4. Performance of the six models for different experimental and simulation inputs. All models successfully identify negative test cases and properly reject the incorrectly shaped simulation defects.

Image Type	Performance	Model					
		1A	1B	2A	2B	3A	3B
2DI non-interferometric of the parallel defect	Rejects 22-nm parallel	✗	✗	✗	✓	✓	✗
	Classifies 32-nm parallel	✗	✓	✗	✓	✓	✗
	Rejects 42-nm parallel	✓	✗	✓	✓	✓	✓
2DIS DPM image of the parallel defect	Identifies ‘H’ defect stain	✓	✓	✓	✗	✓	✓
	Rejects 22-nm parallel	✓	✓	✓	✓	✗	✓
	Classifies 32-nm parallel	✓	✓	✓	✓	✗	✓
	Rejects 42-nm parallel	✓	✓	✓	✓	✗	✓
2DIS DPM image of the perpendicular defect	Rejects 22-nm perpendicular	✓	✓	✓	✓	✓	✓
	Classifies 32-nm perpendicular	✓	✓	✓	✓	✓	✓

frames had different defect positions. We used a unique sampling process to reduce the size of the training set without affecting the performance. As part of our algorithm, we used a modified Euclidean distance scheme to reduce the impact of synthetic image noise. We presented the six different optimized models to infer the presence of defects and classify their shapes and sizes. The SEM measured the defect size to be 30.5-31 nm. Thus, the classification system accurately sizes the defects even though they are an order of magnitude smaller than the diffraction limit. In all cases, our true negative rate was extremely high. Hence, our accuracy is close to 100%. The technique successfully makes generic inferences on untrained defect types and sizes. For semiconductor inspection applications, researchers can utilize LER estimation with design parameters to obtain initial simulation images for the learning stage.

Appendix A Modified gradient descent

There are many gradient descent algorithms that work well for optimization of smooth functions [31–33]. However, our loss function is not analytically differentiable because it is a sum of indicator functions. Even the numerical approximation for the gradient using forward difference approximation can be arbitrarily large and thus adversely affect these algorithms. It would be time-consuming to evaluate the numerical approximation of the gradient in all directions. To address these problems, we propose a modified gradient descent technique to evaluate numerical gradient descent for the loss functions of Eq. (6) and Eq. (7). The function is piecewise smooth and we clamp the derivatives at relatively non-smooth points. Equation (17) decides the relative smoothness of the point by comparing its numerical differentiation with the past weighted gradient. The loss function is defined in the domain for numerical differentiation. We have borrowed concepts from [32–34] to implement the modified gradient descent. Let $\hat{\nabla}$ denote the discrete numerical approximation that extends the concept of the gradient operator for the loss functions. On iteration i, we compute the step sizes, Δw_{1i} and Δw_{2i}, using:

$$m_{1i} = (1 - \beta_1)\left(\hat{\nabla} L_w \cdot \hat{w}_1\right); \quad m_{2i} = (1 - \beta_1)\left(\hat{\nabla} L_w \cdot \hat{w}_2\right), \tag{16}$$

$$u_i = \max(\beta_2 \times u_{i-1}, ||\hat{\nabla} L_w \cdot \hat{w}_i||), \tag{17}$$

$$\Delta w_{1i} = -\frac{\alpha}{1-\beta_1{}^2}\left(\frac{m_{1i}}{u_i}\right); \quad \Delta w_{2i} = -\frac{\alpha}{1-\beta_1{}^2}\left(\frac{m_{2i}}{u_i}\right). \tag{18}$$

The goal is to find the point of minimum loss in w_1, w_2 space. We do this by computing the directional derivatives of the loss with respect to w_1 and w_2. We measure the loss at three points: $L_w(w_1, w_2)$, $L_w(w_1+\Delta w_1, w_2)$, and $L_w(w_1, w_2+\Delta w_2)$ and calculate the forward difference approximations $[L_w(w_1+\Delta w_1, w_2)\text{-}L_w(w_1, w_2)]/\Delta w_1$ and $[L_w(w_1, w_2+\Delta w_2)\text{-}L_w(w_1, w_2)]/\Delta w_2$, respectively. Equation (17) provides information about the relative smoothness of the loss surface at a point. If the gradient in Eq. (17) is too steep compared to the previous gradient, we clamp the gradient by normalizing it. This prevents instability from using too large of a step. On the other hand, if the current point is relatively smooth, then it results in a gradient-dependent reasonable step size. Thus, the algorithm moves with either a constant or a variable but reasonable learning rate based on the update rules. This is useful for tackling the exploding gradient issues in optimization [31] for loss functions with large gradients. We use constraints on w derived from Step 1 of Section 3.6. Finally, we have the option to use a constant step size β_3 to pull the gradient from plateaus and to confine the function in the constrained domain. We can initialize the step size, Δw, to $\pm\ \beta_3$ and reset the gradient descent when the numerical gradient is less than a threshold, i.e., $||\hat{\nabla} L_w \cdot \hat{w}_i|| < \varepsilon$. Once $||\hat{\nabla} L_w \cdot \hat{w}_i|| \geq \varepsilon$, we can again apply the gradient update rules from Eq. (16)-(18). The optimization typically requires less than 30 iterations even for different starting points and different loss functions. We used the following constant values: β_1 = 0.9; β_2 = 0.999; $||\beta_3||$ = 0.25; α = 0.03. We kept the values of β_1 and β_2 the same as was done by Kingma et al. in [32]. We initialize the step size to 1.0 for both w_1 and w_2.

Appendix B Simulated seed data

The model training is done using the rectangular shaped simulated defects. We optimized the model for selecting classifier parameters for defect shape and size detection during the learning phase. The testing stage consists of a 22-nm, 32-nm, and 42-nm seed rectangular-shaped defect. In addition, we have used 'H' shape simulated defects in the testing stage for the shape classification. The width variation along the longer sides is: width 27.5 nm. See Fig. 10. Table 5 presents the peak scattering signatures for defects with different sizes and shapes.

Fig. 10. (a) Shape of rectangular-shaped simulated defect images. (b) Shape of defects for H shape image is formed by subtracting the background pattern of parallel bars from a pattern with defect.

Table 5. Peak intensity trends in the denoised simulated mutual interference images for various seeds.

	Parallel defect			'H' shaped defect		Perpendicular defect	
Seed	22-nm	32-nm	42-nm	32-nm	42-nm	22-nm	32-nm
max(M_1)	9.32	26.00	62.82	33.57	231.89	197.51	271.81
max(M_2)	11.74	32.28	72.07	66.49	278.30	224.63	278.80
max(M_3)	14.66	38.91	82.78	101.85	325.72	226.95	285.49
max(M_4)	17.86	46.22	93.77	141.09	372.95	238.06	302.52
max(M_5)	21.90	54.17	103.74	185.30	416.32	261.24	318.62

Appendix C Defect shape classification

Table 6 summarizes the cross-classification results. It is evident that only stitching or edge region errors are detected by using the parallel defect model to detect defects in the perpendicular bridge defect die. Stitching error regions are a tilted form of the parallel bridge defect and so they are detected with the parallel defect model. There is a high rate for true negatives for the entire die. All models correctly reject the central region as a defect.

Table 6. Defect detection 32-nm.

Model	Detection rate	Defect id	Precision = TP/(TP+FP)
1A	60%	Edge-based (122, 12)	100%
1B	60%	Stitching error region (716, 266)	52%
2A	0%	No defect detected	0%
2B	80%	Stitching error region (719, 265)	59%
3A	65%	Edge-based (754, 40)	100%
3B	65%	Stitching error region (717, 265)	100%

Appendix D Shape and size classification

The effect of noise reduction using PCA is displayed in Fig. 11 The SNR was 23.5 dB initially and 22.2 dB after PCA. Compared to the parallel defect, the perpendicular defect has a weaker scattering signature. The background pattern and the noise overwhelm the perpendicular defect signal. As a result, the defects were not visible even after 2DIS processing. We used PCA on the experimental data to improve its SNR from 9.9 dB to 18.2 dB. See Fig. 12.

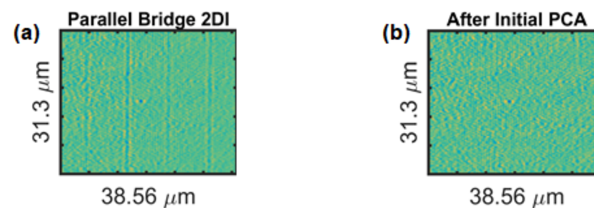

Fig. 11. Parallel defect non-interferometric image with 2nd order difference (2DI). (a) Before and (b) after applying PCA.

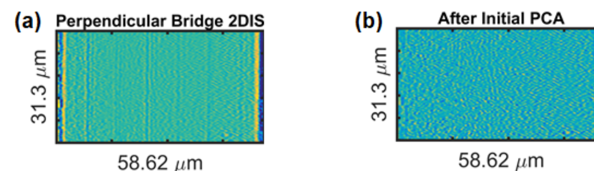

Fig. 12. Perpendicular defect interferometric image with 2DIS. (a) Before and (b) after applying PCA. The defect signal is buried in background and noise.

The performance details of the models in Fig. 9 are explained here. For Figs. 9(a)-9(c), Category 1 models use $c = 1.1$, Category 2 models use $c = 1.2$, and Category 3 models use $c = 1.0$ on the outcome list. Model 2B can display shape-based similarity for parallel defect and can classify the 32-nm size. The low precision of 3B indicates that other defects such as stitching error or nano-scale dust are detected above the 50% threshold. For model 3A, trial #17 provides $ED = 16.518$. It is the only defect detected in this trial. This trial corresponds to

$\mathscr{A}_j = \{26.00, 38.91, 46.22\}$. This provides evidence of our hypothesis that experimental data has frame-to-frame variations in the SNR and intensity due to noise and changes to the background.

We now discuss size inference in stitched DPM parallel bridge defect images in Figs. 9(d)-9(f). Category 1 models use $c = 1.1$, Category 2 models use $c = 1.1$, and Category 3 models use $c = 1.2$ on the outcome list. For model 2B, trial #15 shows the lowest distance ($ED = 13.958$) and highest precision. It detects only the central defect. The intensity pattern is $\mathscr{A}_j = \{46.22, 26.00, 46.22\}$. Size inference in stitched DPM perpendicular defect images is displayed in Figs. 9(g)-9(i). Unlike with the parallel bridge defect, we could not estimate the mean stage translation distance using the experimental optical images because the tripole pattern was not visible. Thus, we estimated $D_{mean} = 29$ pixels using the model parameters obtained in the learning stage. Category 1 models use $c = 2.1$, Category 2 models use $c = 2.0$, and Category 3 models use $c = 2.0$ on the outcome list due to lack of data redundancy. We expect BX67 and BY67 defects to have similar sizes. For model 3B, trial #5 provided the best results. It detects the central defect with $ED = 9.40 \times 10^{-4}$ and $\mathscr{A}_j = \{302.5, 318.6, 302.5\}$. This indicates that similar to our hypothesis, the noisy experimental data has different intensity variations.

Funding

Cisco Systems Inc (CG 587589, CG 624009, CG 1141107).

Acknowledgments

S. Purandare, L. Goddard, and A. Schwing conceptualized the machine learning algorithm. S. Purandare developed the machine learning procedure and codes and collected the SEM images. J. Zhu performed the EM modeling simulations. R. Zhou performed the optical measurements. L. Goddard supervised the research in each of these three areas. G. Popescu supervised the optical measurement research. We thank Dr. Edmond Chow for providing SEM training and helping with collecting some of the images.

References

1. R. Zhou, C. Edwards, C. A. Bryniarski, G. Popescu, and L. L. Goddard, “9nm node wafer defect inspection using three-dimensional scanning, a 405nm diode laser, and a broadband source,” Proc. SPIE **9424**, 942416 (2015).
2. B. Johnson, S. T. Wang, S. F. Jacobson, and J. Walker, “Learning from IoT/OT implementation in semiconductor manufacturing,” https://www.gartner.com/doc/3442917/learning-iotot-implementation-semiconductor-manufacturing (2016).
3. N. G. Shankar and Z. W. Zhong, “Defect detection on semiconductor wafer surfaces,” Microelectron. Eng **77**, 337–346 (2005).
4. D. Carrera, F. Manganini, G. Boracchi, and E. Lanzarone, “Defect detection in SEM Images of nanofibrous materials,” IEEE Trans. Ind. Inform **13**, 551–561 (2017).
5. P. Walle, E. Kramer, J. van der donck, W. Mulckhuyse, L. Nijsten, F. A. B. Arango, A. de Jong, E. Zeijl, H. Spruit, J. H. van den Berg, G. Nanda, A. K. van Langen-Suurling, P. F. A. Alkemade, S. Pereira, and D. Maas, “Deep sub-wavelength metrology for advanced defect classification,” Proc. SPIE **10329**, 103294N (2017).
6. B. M. Barnes, M. Y. Sohn, F. Goasmat, H. Zhou, A. E. Vladár, R. M. Silver, and A. Arceo, “Three-dimensional deep sub-wavelength defect detection using $\lambda = 193$ nm optical microscopy,” Opt. Express **21**, 26219–26226 (2013).
7. R. Zhou, C. Edwards, A. Arbabi, G. Popescu, and L. L. Goddard, “Detecting 20 nm defects in large area nanopatterns using optical interferometric microscopy,” Nano Lett. **13**, 3716–3721 (2013).
8. R. Zhou, C. Edwards, G. Popescu, and L. L. Goddard, “9 nm node wafer inspection using visible light,” Proc. SPIE **9050**, 905017 (2014).
9. P. Thony, D. Herisson, D. Henry, E. Severgnini, and M. Vasconi, “Review of CD measurement and scatterometry,” in *AIP Conf. Proc,* vol. 683 (2003), pp. 381–388.
10. A. J. Boef, “Optical wafer metrology sensors for process-robust CD and overlay control in semiconductor device manufacturing,” Surf. Topogr. Metrol. Prop. **4**, 023001 (2016).
11. H. Ng, “Automatic thresholding for defect detection,” Pattern Recogn. Lett **27**, 1644–1649 (2006).
12. A. Kumar, “Computer-vision-based fabric defect detection: a survey,” IEEE Trans. Ind. Electron **55**, 348–363 (2008).
13. L. Xie, R. Huang, and Z. Cao, “Detection and classification of defect patterns in optical inspection using support vector machines,” in *Intelligent Computing Theories. ICIC 2013. Lecture Notes in Computer Science,* vol. 7995 D. S. Huang, V. Bevilacqua, J. C. Figueroa, and P. Premaratne, eds. (Springer, Berlin, Heidelberg, 2013), pp. 376–384.

Optics EXPRESS

14. L. Waller and L. Tian, "Computational imaging: machine learning for 3D microscopy," Nature **523**, 416–417 (2015).
15. D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," CIRP Ann. Manuf. Techn **65**, 417–420 (2016).
16. C. L. Chen, A. Mahjoubfar, L. Tai, I. K. Blaby, A. Huang, K. R. Niazi, and B. Jalali, "Deep learning in label-free cell classification," Sci. Reports **6**, 21471 (2016).
17. N. H. Saad, A. E. Ahmad, and H. M. S. A. Hasan, "Automatic semiconductor wafer image segmentation for defect detection using multilevel thresholding," in *MATEC Web Conf,* vol. 78 (2016), p. 01103.
18. J. K. Park, B. K. Kwon, and J. H. Park, "Machine learning-based imaging system for surface defect inspection," Int. J. of Precis. Eng. and Manuf.-Green Tech **3**, 303–310 (2016).
19. A. Sinha, J. Lee, S. Li, and G. Barbastathis, "Lensless computational imaging through deep learning," Optica **4**, 1117–1125 (2017).
20. L. Xing, Y. Rivenson, N. T. Yardimci, M. Veli, M. Jarrahi, and A. Ozcan, "All-optical machine learning using diffractive deep neural networks," CoRR **361**, 1004–1008 (2018).
21. N. G. Orji, M. Badaroglu, B. M. Barnes, C. Beitia, B. D. Bunday, U. Celano, R. J. Kline, M. Neisser, Y. Obeng, and A. E. Vladar, "Metrology for the next generation of semiconductor devices," Nat. Electron. **1**, 532–547 (2018).
22. L. Zhang, W. Dong, D. Zhang, and G.Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," Pattern Recogn **43**, 1531–1549 (2010).
23. C. Edwards, R. Zhou, S. W. Hwang, S. J. McKeown, K. Wang, B. Bhaduri, R. Ganti, P. J. Yunker, A. G. Yodh, J. A. Rogers, L. L. Goddard, and G. Popescu, "Diffraction phase microscopy: monitoring nanoscale dynamics in materials science," Appl. Opt. **53**, G33–G43 (2014).
24. J. Zhu, Y. Shi, L. L. Goddard, and S. Liu, "Application of measurement configuration optimization for accurate metrology of sub-wavelength dimensions in multilayer gratings using optical scatterometry," Appl. Opt. **55**, 6844–6849 (2016).
25. J. Zhu, K. Zhang, N. Davoudzadeh, and X. Wang, "Electromagnetic field modeling for defect detection in 7 nm node patterned wafers," Proc. SPIE **9778**, 97780P (2016).
26. J. Zhu, Y. Shi, S. Liu, and L. L. Goddard, "Generalized measurement configuration optimization for accurate reconstruction of periodic nanostructures using optical scatterometry," Proc. SPIE **9778**, 977823 (2016).
27. S. Weisenburger and V. Sandoghdar, "Light microscopy: an ongoing contemporary revolution," Contemp. Phys. **52**, 123–143 (2015).
28. R. Szeliski, *Computer vision: algorithms and applications* (Springer Science & Business Media, 2010).
29. R. Gonzalez and R. E. Woods, *Digital image processing*, vol. 2 (Prentice-Hall, Inc., 2006). Chapter 2 and Chapter 4 .
30. A. K. Jain, *Fundamentals of digital image processing* (Prentice-Hall, Inc., 1989). Chapter 4 .
31. R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," Comput. Res. Repos. (CoRR) **abs/1211.5063** (2012).
32. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," Comput. Res. Repos. (CoRR) **abs/1412.6980** (2015).
33. S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," Comput. Res. Repos. (CoRR) **abs/1904.09237** (2019).
34. A. Y. Kruger, "On Fréchet subdifferentials," Journal of Mathematical Sciences **116**, 3325–3358 (2003).

128.174.216.206 {ts '2019-06-14 11:44:08'}