Automated Void Detection in TSVs from 2D X-Ray Scans using Supervised Learning with 3D X-Ray Scans

Ramanpreet Singh Pahwa [1, 2], Saisubramaniam Gopalakrishnan [1], Huang Su [1], Ong Ee Ping [1],
Haiwen Dai [4], David Ho Soon Wee [3], Ren Qin [3], Vempati Srinivasa Rao [3]
[1] Institute for Infocomm Research (I^2R), A*STAR
[2] Artificial Intelligence, Analytics And Informatics (AI^3), A*STAR
[3] Institute of Microelectronics (IME), A*STAR
[4] Carl Zeiss SMT Inc.
{ramanpreet_pahwa, g_saisubramaniam, huangs, epong}@i2r.a-star.edu.sg,
haiwen.dai@zeiss.com, {hosw, renq, vempati}@ime.a-star.edu.sg

***Abstract*—Yield improvement is a critical component of semiconductor manufacturing. It is done by collecting, analyzing, identifying the causes of defects, and then coming up with a practical solution to resolve the root causes. Semiconductor components such as Through Silicon Vias (TSVs) and other package interconnects are getting smaller and smaller with the ongoing miniaturization progress in the industry. Detecting defects in these buried interconnects is becoming both more difficult and more important. We collect both 2D and 3D X-Ray scans of defective TSVs containing defects such as voids. We label the data in 3D and perform registration between 2D and 3D scans. We use this registration information to locate the TSVs and void defects in these 2D X-ray scans which would be difficult to label manually as these voids look very fuzzy in 2D scans. Thereafter we use a state-of-the-art deep-learning segmentation network to train models to identify foreground (TSV, void defects) from the background. We show that our model can accurately identify the TSVs and their voids in images where it is impossible to locate the defects manually. We report a dice score of 0.87 for TSV segmentation and a dice score of 0.67 for void detection. The dice score for voids demonstrates the capability of our models to detect these difficult buried defects in 2D directly.**

***Index Terms*—Deep-Learning, 2D Segmentation, X-Ray Analysis, Defect Detection**

I. Introduction

Through-Silicon Vias (TSVs) play a vital role in die-stacking configuration. Despite careful planning and principled approaches, non-systematic defects such as voids are still inevitable during fabrication. Failure analysis is a critical step in improving semiconductor manufacturing yields. Industry 4.0 tools have brought an increased focus on digital technology to improve yield across front-end and back-end manufacturers. With the increasing miniaturization of Through-Silicon Vias (TSVs) and other package interconnects, detecting defects in these buried interconnects becomes more challenging and is gaining prominence.

This research is supported by Economic Development Board (EDB), Singapore under its IAF-ICP Grant no. I1901E0048 and administered by the Agency for Science, Technology and Research (A*STAR).

Optical analysis using color or IR sensors can only provide information about surface defects, whereas X-Ray machines usually focus on image sub-surface defects. 2D X-Ray scans are the de-facto preference by industry for in-line inspection today due to their wider field of view and higher throughput than 3D X-ray microscopy (XRM). However, it is sometimes difficult to get complete picture of the via clearly as 2D X-Rays provide incomplete information about sub-surface defects. Thus, it is difficult to infer what the 3D surface looks like from a 2D image. Moreover, depending on the projection angle and resolution of these 2D scans, the problem gets exponentially harder. As such, discovering buried attributes becomes cumbersome from the viewpoint of the process engineer.

To overcome this challenge, recently, data is acquired non-destructively with a 3D X-ray microscope to obtain 3D voxelized data [1]. 3D XRMs are more granular and powerful enough to visualize these defects, but also come with the expense of limited coverage and longer scanning time. This data is manually analyzed to identify defects such as voids that may be present in different structures. Recently machine learning techniques are being used to automatically identify the structures and segment out different components such as solder, Cu-Pillar (CuP) and Cu-Pads. Artificial Intelligence (AI) has had significant impact on several technologies such as visual surveillance, predictive maintenance, object detection, and now, the semiconductor industry is seeing its influence. The unique combination of copious amount of data from 2D/3D XRM and data-hungry deep-learning has the potential to revolutionize automated failure analysis for certain structures such as TSVs.

The focus of this paper is to leverage the rich information of 3D XRM along with the speed and wider field of view of 2D X-ray imaging to perform in-line inspection on TSVs to detect buried voids that may be difficult to identify manually. We will use the 3D XRM scans to identify buried structures and defects and use its coordinate projection to aid in developing defect

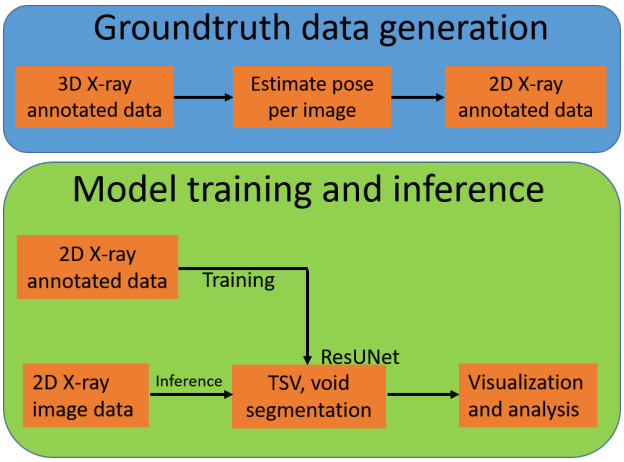
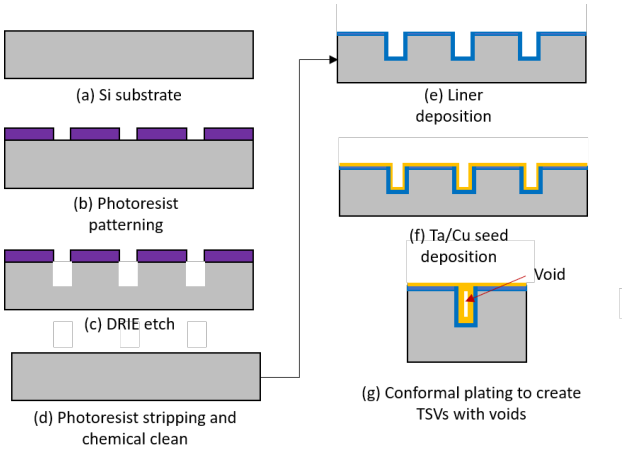
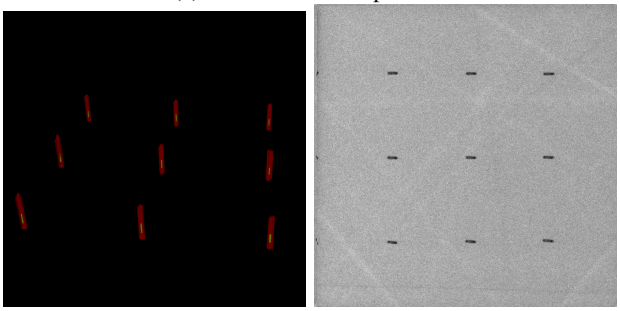


Fig. 1: Our approach for generating groundtruth labels for TSVs and voids is to use 3D annotations and project them onto 2D. We use the 2D labels along with 2D raw scans to train segmentation models for automated defect detection.



(a) TSV fabrication process.



(b) 3D XRM scan of fabricated TSV containing voids.

(c) Raw 2D XRM scan of fabricated TSV containing voids.

Fig. 2: Our approach for fabricating and scanning defective TSVs to generate 2D and 3D data. Voids are colored in yellow.

detection system in the relatively obscure 2D raw scans. 3D XRM datasets are typically generated from raw 2D cone beam X-ray projections using algorithms such as FDK reconstruction. The projections are spread over the selected angle range of -3° to 183°, with the additional 6° serving as fan angles to cover the field of view. The voids will be identified in 3D XRM scans and we will use the projection-angle table along with the projection parameters containing information such as focal length and optical center for registering the locations of TSVs and voids in corresponding 2D scans. These projections of 3D groundtruth attribute coordinates onto the 2D projections will provide the required supervision to train deep-learning models to detect voids in the raw 2D scans directly. We show our results on defective and non-defective TSVs and compare our void detections with state-of-the-art techniques currently used in the industry.

Our approach attempts to fuse the strengths of 2D mature detection techniques with accurate 3D information. As shown in Fig. 1, we locate and annotate the TSVs and voids in 3D scans and computationally register these 3D scans with raw 2D x-ray images. Once the registration is performed, the 3D information is projected onto 2D to create binary masks for TSVs and voids. Thereafter, we train a 2D segmentation model for automatically segmenting the TSVs and voids in each 2D X-ray scan. We show that our deep-learning based models are even capable of identifying voids accurately that may not be possible to locate by an experienced engineer.

II. Related Work

One of the most important tasks in AI based defect detection is of automated image segmentation. This can be either multi-class semantic segmentation or binary segmentation. In this work we are primarily interested in binary segmentation that consists of foreground-background segmentation. While 2D segmentation techniques have matured covering accuracy and inference speed over the past few years [2]–[6], 3D techniques are still being improved significantly [1], [7], [8]. Historically, an image is usually analyzed using various features such as edges, colors, size, and histograms at various scales. Such images are processed to obtain a foreground-background (binary) region. With the popularity of machine learning, various techniques such as U-Net [9], SegNet [10], and Mask-RCNN [11] have improved binary and multi-class segmentation over such traditional hand-crafted techniques. U-Net based segmentation has become the industry gold-standard for binary segmentation related tasks both in 2D and 3D. It contains a encoder that analyzes the entire image. The encoder contracts over 4 similar sequences of convolution, activation, and pooling layers. A bridge joins the encoder to a decoder. This is followed by a decoder that produces an accurate segmentation of the object of interest. The decoder, like a mirror image of encoder, expands over 4 similar sequences of deconvolution, convolution, and activation layers. The standard U-Net architecture usually expects a smaller image resolution such as 256×256 to perform image segmentation. Thus, images are first downsampled to perform image segmentation. Thereafter, the segmented masks are interpolated to the original resolution to produce final segmentation.

AI based 3D segmentation for defect detection has recently

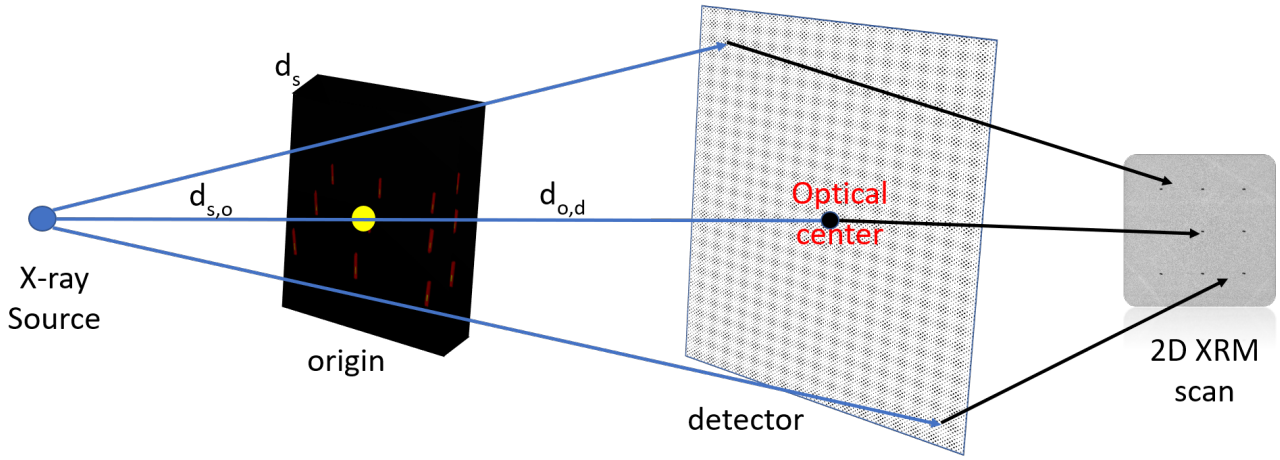


Fig. 3: The sample is rotated and 2D signals of projected X-rays are recorded at the detector. The are eventually captured as raw 2D XRM scans and saved for offline processing.

garnered a lot of interest both in research and industry [1], [7]. However, the results are still not up to industry standards and require lots of 3D data which is exponentially more difficult to obtain and annotate. The advantage of using 3D data for defect analysis is that buried defects such as voids are easier to identify and locate accurately. Engineers can use geometrical and temporal information to identify these buried defects accurately in 3D scans which may not be visible properly in 2D X-ray scans to humans.

III. Data Fabrication and Generation

In this section we will briefly describe our approach to fabricate and scan our custom 2.5D test vehicles. The test vehicles have been specifically designed to represent contemporary High-Performance Computing packages. These test vehicles utilize a silicon interposer with Through Silicon Vias (TSVs). One of the major requirements for an AI based defect detection solution is to having access to enough defective data so as to identify and measure these defects such as solder voids. Our TSVs are fabricated on $300mm$ silicon wafers using standard 2.5D manufacturing processes. As shown in Fig. 2, a conformal plating process was used to fill the TSVs. This is because using conformal plating allows us to create TSVs with voids that is desired to generate defective data for training our deep-learning based segmentation models. For more details on TSV fabrication, the readers can refer to [1].

3D X-ray microscopy is used to inspect and scan the TSVs. The benefit of 3D microscopy is that unlike SEM analysis we can inspect the test vehicles non-destructively. The test vehicles are mounted on sample holders, placed on the XRM autoloader and rotated incrementally from -3° through 183°. Raw 2D X-ray scans are imaged each time the sample is rotated about $0.22°$. These 2D scans along with geometrical information is used in a propriety algorithm to computationally obtain 3D X-Ray scans. This process, also known as computational tomography, allows the datasets to be visualized and processed in 3D where buried structures can be imaged at a high resolution. For more details on data generation, readers can refer to [1], [7]. The data output is approximately $1,000 \times 1,000 \times 1,000$ voxels for 3D scans and $1,015 \times 1,015$ for 2D scans consisting of about 800 images.

IV. 2D-3D Relationship

In this section, we describe our approach to finding the projection relation between the 3D XRM scans and raw 2D XRM scans. Internally in the XRM tool, the raw 2D scans are used to generate the 3D XRM scans. This is usually done using feature matching, 2D-3D registration, and bundle adjustment [12]. As this algorithm is proprietary and we do not have access to it, we approach this problem from another angle. Instead of generating a 3D scan from raw 2D images, we formulate the problem as finding the projection relationship - how the 3D scan can be projected onto each raw 2D image as accurately as possible.

Lets assume sample frame of reference is *c* and X-ray frame of reference is *w*. We can transform the sample coordinate frame of reference into X-ray frame of reference using a Rotation matrix $\mathbf{R}_{c,w}$ and translation $\mathbf{t}_{c,w}$ vector:

$$\mathbf{X}_w = \mathbf{R}_{c,w}\mathbf{X}_c + \mathbf{t}_{c,w} \tag{1}$$

where $t_{c,w} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}_{c,w}$. As the sample is rotated with a known angle ϕ, we can compute the $\mathbf{R}_{c,w}$ as:

$$\mathbf{R}_{c,w} = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}_{c,w} \tag{2}$$

As can be seen in Fig. 3, we can use similar triangles to obtain the projection of each 3D point $\mathbf{X}_w$ onto detector plane as follows:

$$\mathbf{X}_p = \alpha \mathbf{X}_w; \qquad \alpha = \frac{d_{s,o} + d_s + d_{o,d}}{Z_w} \tag{3}$$

where Z_w refers to the Z co-ordinate of $\mathbf{X}_w$. Once we obtain the 3D coordinate of each point on detector plane, these can be transformed onto each 2D XRM raw image as follows:

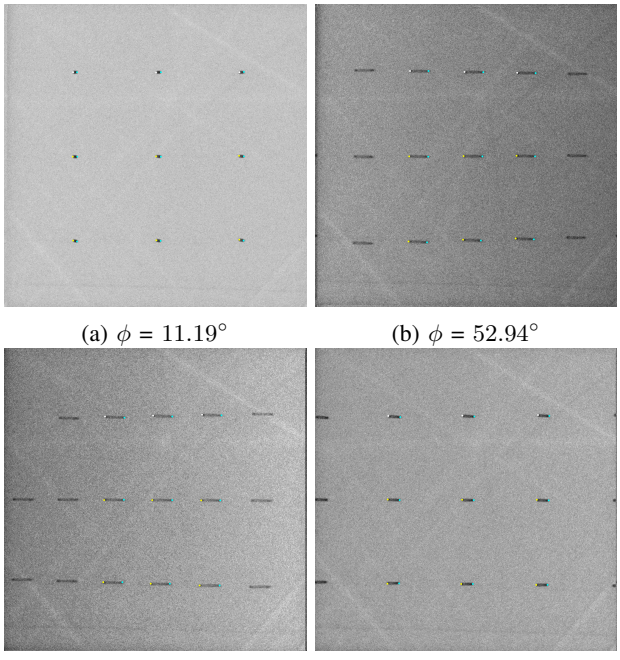
$$\mathbf{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{X}_p \tag{4}$$

where

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Here, parameters - $f, c_x, c_y, t_x, t_y, t_z$ are unknowns. This is based on the assumption that we have all the information regarding positions of TSVs known ($\mathbf{X}_c$). In the 3D scans obtained, we only have access to the layer number (voxelized planes) for the TSVs. This further introduces more unknown parameters that we need to estimate. One should note that all these parameters are static across all different 2D scans as the rotation $\mathbf{R}_{c,w}$ for each scan is known.

Each of our defective 3D TSV XRM scans consist of nine TSVs. We annotate the 3D scans by recording the top-most and bottom-most point of TSVs. We also annotate the 2D scans by recording the left most (corresponding to top part) and right most (corresponding to bottom part) of each TSV. This provides us with 18 3D-2D correspondences per annotated image. We only need to annotate the 3D scan once. We annotate 10 2D scans for varying ϕ from 0° to 169°. Some of the annotations in 2D are shown in Fig. 4



(a) ϕ = 11.19° (b) ϕ = 52.94°

(c) ϕ = 127.6° (d) ϕ = 155.43°

Fig. 4: Our manual annotations in 2D scans for each TSV at varying projection angles. Best seen in color.

V. 2D Segmentation Networks

Once the groundtruth pixels corresponding to TSVs and Voids are mapped onto the 2D scans using the annotated 3D scan as reference, we train a neural network to learn how to segment the two components, taking into consideration the different angles and slightly different resolution of the samples. Specifically, we use a modified version of the U-Net [9] architecture, having residual skip connections [13] and instance normalization [14] for better model convergence. Our architecture, shown in Fig. 5, consists of a shared encoder module with separate decoder modules for TSV and Void segmentation. The 2D scans are processed by repeated resnet blocks up to depth of 3, downsampled using convolution layers with stride of 2. The shared encoded representation is then branched into two decoders, each having upsampling and resnet blocks. These are concatenated by skip connections taken from the corresponding depth of the encoder. A single 1×1 convolution filter is then applied on top of the individual decoders for the network to predict the TSV and Void masks. All convolution filters in encoder are set to be 64, whereas in decoder, we set 8 filters for TSV branch and 64 filters for Void branch. The reduction in TSV filters is to prevent TSV segmentation from overfitting and dominating the Void segmentation. From hereon, we will call this modified U-Net as standard U-Net.

For optimizing the network, we employ Dice loss given by,

$$L_{dice} = 1 - 2\frac{\sum_{i=1}^{N_i}\sum_{j=1}^{N_j} p_{ij} r_{ij} + \psi}{\sum_{i=1}^{N_i}\sum_{j=1}^{N_j}(p_{ij} + r_{ij}) + \psi} \tag{6}$$

where $< i, j >$ represents the 2D pixel coordinate pair, p_{ij} is the predicted probability by the network at that pixel, r_{ij} is the groundtruth at that pixel. ψ, set to 1, provides numerical stability while also avoiding situations where denominator is zero.

VI. Experiments

We fabricated and scanned 3 sets of TSV datasets. Each set contains 9 TSVs spread out in a 3×3 grid pattern. Every TSV contains void zones in them. We first annotate the TSVs and voids in 3D data manually. Then we use our 2D-3D registration formulation to project these labels onto 2D scans and create binary masks for TSVs and voids as shown in Fig. 6. We created 280 binary masks per component for each TSV dataset resulting in a total of 840 binary masks per component for all 3 TSV datasets. All experiments were run using 64GB RAM and RTX Titan GPU, and results obtained were averaged over three runs.

A. 2D-3D registration

We annotated 10 2D scans along with 3D scan for each of the three TSV datasets. As each dataset consists of different

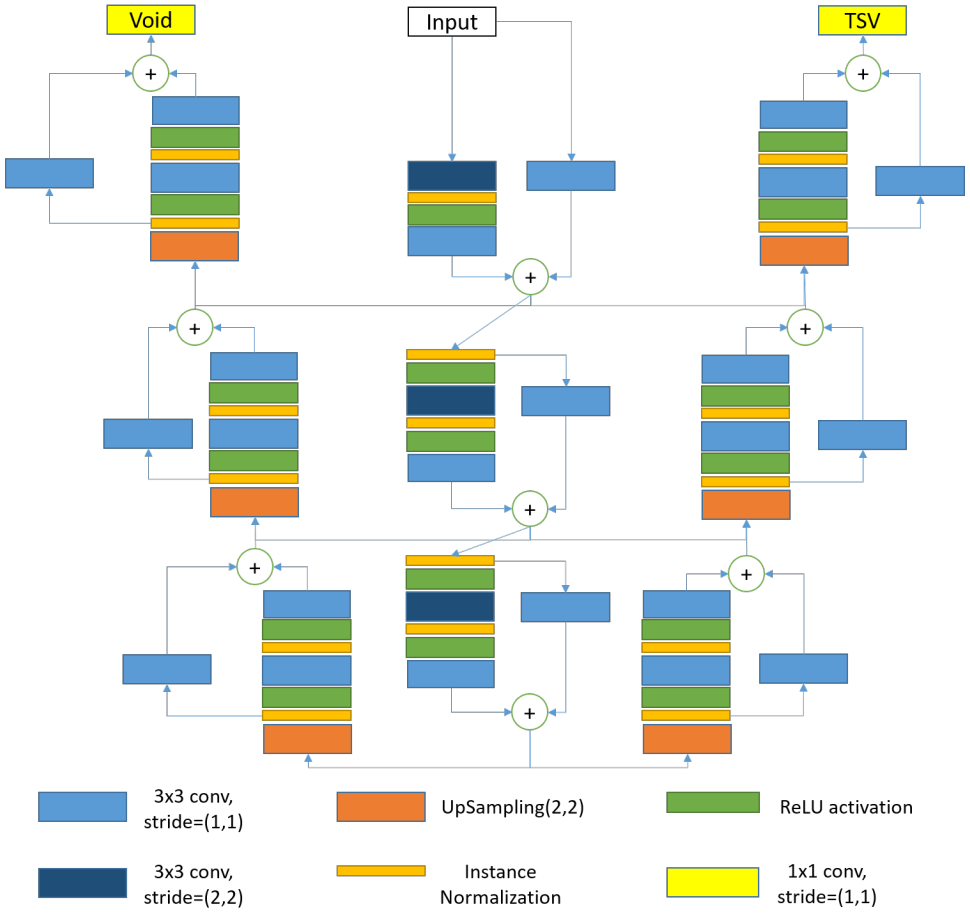


Fig. 5: U-Net architecture with shared encoder (center) and dual decoder branches for 2D (i) TSV (right), and (ii) Void (left) segmentation.

parameters for projections, we estimate three sets of projections parameters ψ^i, $i \in \{1, 2, 3\}$.

$$\psi = \{f, c_x, c_y, t_x, t_y, t_z, \phi, x_m, y_m, z_m\} \tag{7}$$

where, f, c_x, c_y refer to the focal length, and optical center of our 2D XRM image. t_x, t_y, t_z, ϕ refer to the translation vector and rotation of the sample coordinate frame with respect to world coordinate frame. x_m, y_m, z_m refer to the center of sample coordinate frame which is also unknown parameter for us.

$$\epsilon = \frac{1}{N * M} \sum_{i=1}^{N} \sum_{j=1}^{M} \sqrt{(u_p^{i,j} - u_g^{i,j})^2 + (v_p^{i,j} - v_g^{i,j})^2)} \tag{8}$$

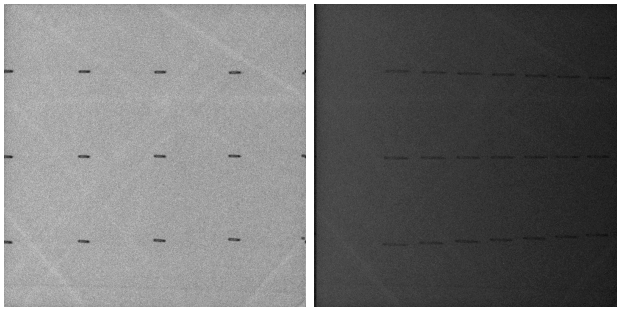
where ϵ refers to the average projection error in pixels, u_p, v_p refer to the x and y coordinate of projected point while u_g, v_g refer to the corresponding groundtruth. N refers to the number of images annotated and M refers to the points annotated per image. We iteratively optimize for these parameters by minimizing the euclidean distance between projected corners and actual corners using Levenberg–Marquardt algorithm [15]. We report the average projection error (in pixels) in Table I.

TABLE I: Our average projection error, in pixels, after optimization for 2D-3D registration.

Avg Error	TSV1	TSV2	TSV3
ϵ	1.4734	1.949	1.201

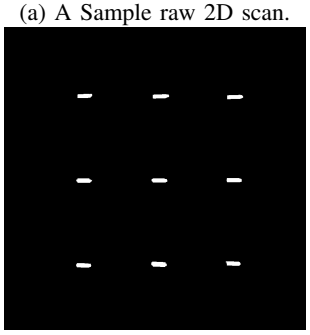
B. 2D segmentation

Original 2D scans of bad TSV of dimension 1015×1015 were resized to 256×256 due to limited training size and hardware capacity. The U-Net network was trained using $90\% - 10\%$ training and validation split, for 100 epochs with a batchsize of 16. Modelcheckpoint and early stopping were also included to save the best model based on lowest validation loss. For augmentation we employed random vertical or horizontal flip, scaling with a factor of 0.1, and rotation with factor of $30°$. Each sample had a 0.5 probability of being augmented every epoch. Introducing random brightness and contrast deteriorated performance of both training and validation, which is likely due to the inability of the network to distinguish void and background under different settings. Since there was already presence of significant noise in each

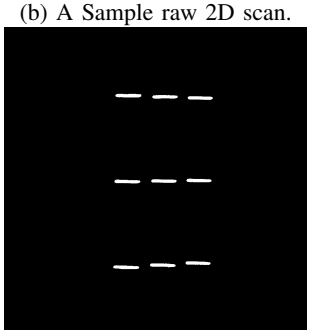


(a) A Sample raw 2D scan.

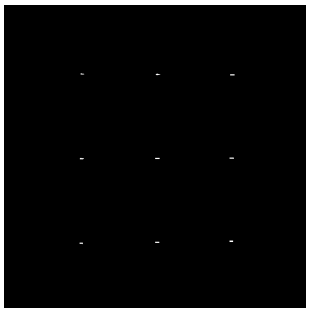
(b) A Sample raw 2D scan.



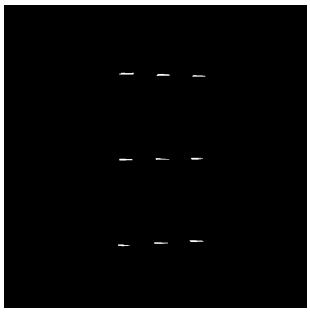
(c) Binary TSV "groundtruth" mask.



(d) Binary TSV "groundtruth" mask.



(e) Binary void "groundtruth" mask.

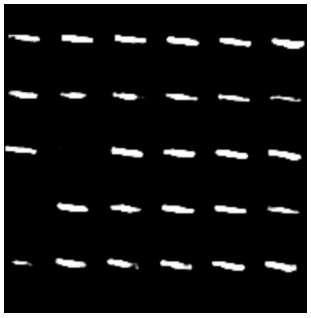


(f) Binary void "groundtruth" mask.

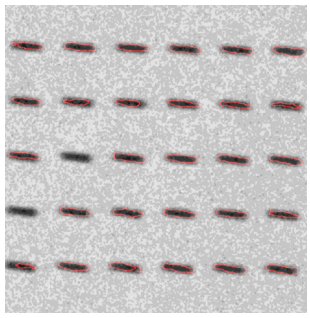
Fig. 6: Some samples of our binary groundtruth generated masks using 3D to 2D projection formulation and 3D annotations.

slice, further addition of Gaussian noise also did not improve our results.

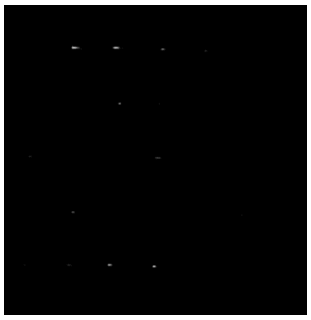
Results on training with original 2D scans are provided in Table II and with augmentation in Table III. As seen from the two tables, augmentations certainly improve the scores by $2-4\%$. Furthermore, we observed severe overfitting in the training and validation scores when training on original data. This was mitigated completely after augmentation. Given the 3 bad TSV samples, we employed hold-one-out strategy for test sample while including the other two. Due to subtle differences between the 3 samples, in terms of resolution, intensity and void size, we observe slight differences in the predicted dice scores. Additionally, we also combined together (denoted as Mix) and sampled randomly an equivalent amount of test slices. Dice score improved greatly for void, attributed to homogeneity of the combined dataset. Current results can



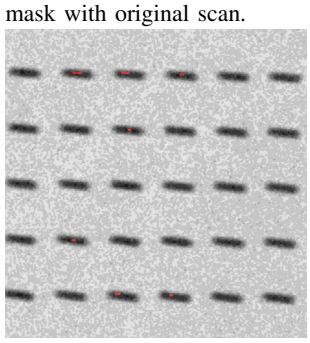
(a) Good Sample TSV predicted mask from U-Net.



(b) Good Sample TSV overlay mask with original scan.



(c) Good Sample Void predicted mask from U-Net.



(d) Good Sample Void overlay mask with original scan.

Fig. 7: TSV and void segmentation masks and overlay for the Good sample. Network prediction on unseen data with varying resolution and TSV placement, as observed, most of the TSVs are segmented correctly with minimal false void positives.

be further improved with more training data.

Figure 8 provides a visual demonstration of the TSV and void segmentation masks. Groundtruth and predicted masks are compared together for an enlarged region of interest. We also show the overlay of both groundtruth and prediction on the TSV and void components to highlight the boundary of detected components. Green denotes the groundtruth, and red denotes prediction. Their intersection is given by yellow. We can observe yellow overlay in most regions. This indicates the predicted masks are well aligned with the groundtruth.

We also tested our network performance on a new TSV sample, different in terms of resolution, number of TSVs in the image, and no voids (all are good). This was done to observe the number of false void positives predicted by the network. Fig. 7 shows the segmentation masks as well as the overlay. It is observed that our network can segment almost all the TSVs, while avoiding a lot of false void positives. Reducing the small presence of predicted void patches can be improved with more samples during training.

VII. Conclusion

In this paper, we have presented a novel framework to utilize 3D information along with 2D dataset for deep-learning based automated defect detection. We leverage on visible buried defects in 3D XRM scans and use geometrical information to project this information onto 2D scans. Thereafter, we train

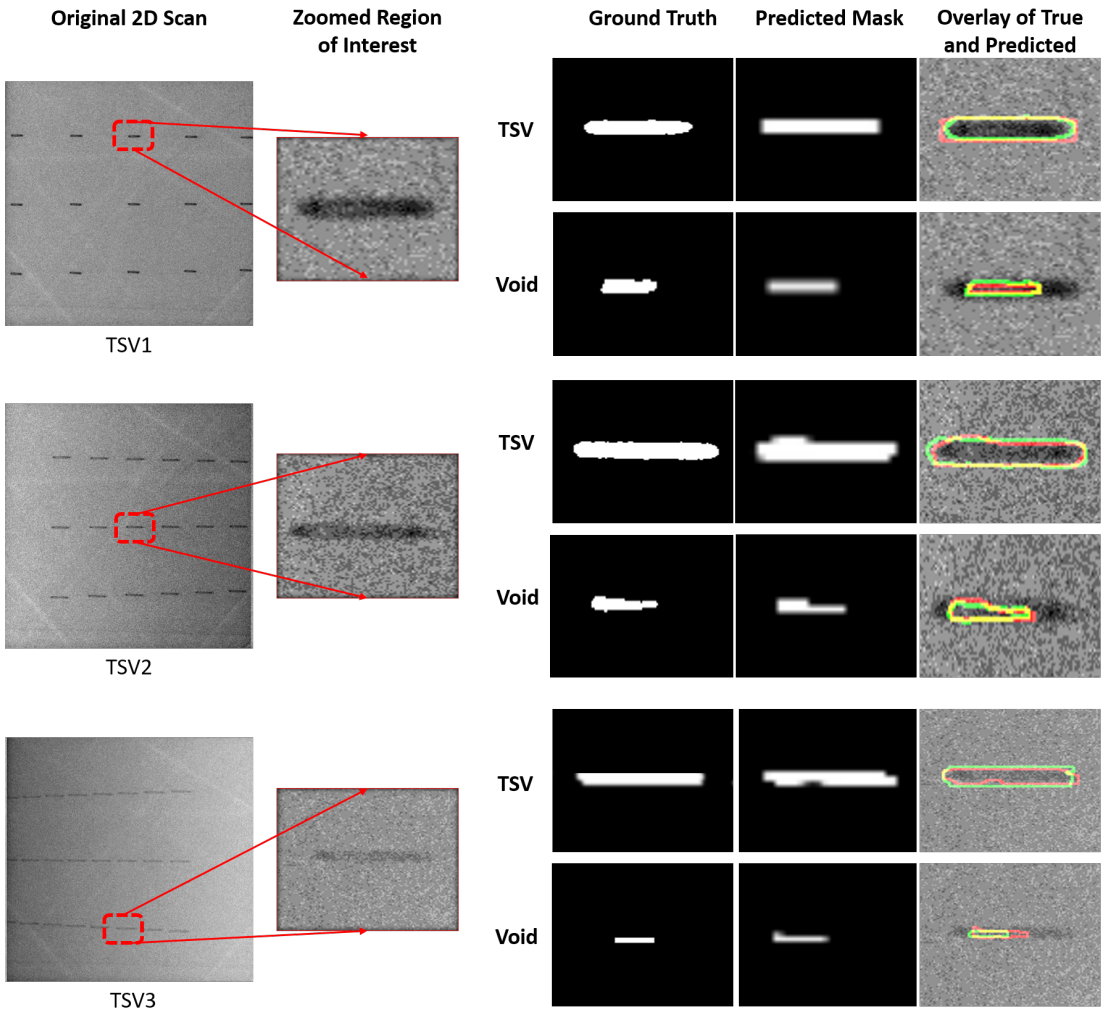


Fig. 8: Bad TSV and Void overlay for 3 Bad TSV samples. (Left) Region of interest is zoomed to a single TSV, and (right) the groundtruth and predicted mask are compared together for both TSV and Void. Overlay of the two masks are also presented to show their boundary intersection. Green denotes groundtruth, red denotes corresponding prediction and yellow denotes intersection of groundtruth and our prediction.

TABLE II: Our 2D dice scores for our 2D segmentation for defective Raw TSV scans without augmentations.

w/o Aug	TSV			Void		
Test Set	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
TSV1	0.9198	0.8942	0.7347	0.7853	0.6720	0.3921
TSV2	**0.9405**	0.9107	0.7873	**0.8870**	**0.7918**	0.4231
TSV3	0.8772	0.8578	0.8301	0.7426	0.6426	0.5247
Mix	0.8631	**0.9198**	**0.8737**	0.6773	0.6165	**0.6354**

TABLE III: Our 2D dice scores for our 2D segmentation for defective Raw TSV scans with augmentations.

w/ Aug	TSV			Void		
Test Set	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
TSV1	0.7677	0.7556	0.7840	0.3410	0.4034	0.4184
TSV2	0.7539	0.8169	0.7849	0.4030	0.3716	0.4445
TSV3	0.7309	0.7535	0.8632	0.4365	0.4181	0.4180
Mix	**0.8593**	**0.9016**	**0.8716**	**0.6525**	**0.6086**	**0.6658**

accurate 2D segmentation models to automatically segment out TSVs and voids that may be present in them. We hope that this work showcases the potential of using 2D and 3D information together where we utilize their individual strengths together. In future, we intend to collect more data as we see major overfitting in our segmentation models. We also hope to apply this approach to other buried structures such as memory and logic die.

ACKNOWLEDGMENT

We would like to thank IME, A*STAR and Carl Zeiss SMT Inc. for their invaluable support and expertise in designing the fabrication and producing the 3D scans. This work is supported by Economic Development Board (EDB), Singapore the under its IAF-ICP Grant no. I1901E0048 and administered by the Agency for Science, Technology and Research.

REFERENCES

[1] R. S. Pahwa, T. L. Nwe, R. Chang, W. Jie, O. Z. Min, S. W. Ho, R. Qin, V. S. Rao, Y. Yang, J. T. Neumann, R. Pichumani, and T. Gregorich, "Deep learning analysis of 3d x-ray images for automated object detection and attribute measurement of buried package features," in *IEEE 22nd Electronics Packaging Technology Conference (EPTC)*, 2020, pp. 221–227.

[2] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Departmental Papers (CIS)*, p. 107, 2000.

[3] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[4] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer vision, graphics, and image processing*, vol. 29, no. 1, pp. 100–132, 1985.

[5] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[6] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , no. 11, pp. 1768–1783, 2006.

[7] R. S. Pahwa, T. L. Nwe, R. Chang, W. Jie, O. Z. Min, S. W. Ho, V. S. Rao, Y. Yang, J. T. Neumann, R. Pichumani, and T. Gregorich, "Machine-based learning methodologies for 3d x-ray measurement, characterization and optimization for buried structures in advanced ic packages," in *17th International Wafer Level Packaging Conference (IWLPC)*, 2020.

[8] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, Eds., Cham, 2016, pp. 424–432, Springer International Publishing.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[10] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2980–2988.

[12] R. S. Pahwa, K. .Y. Chan, J. Bai, V. B. Saputra, M. N. Do, and S. Foong, "Dense 3D Reconstruction for Visual Tunnel Inspection using Unmanned Aerial vehicle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[15] R. S. Pahwa, M. N. Do, T. T. Ng, and B. Hua, "Calibration of depth cameras using denoised depth images," in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 3459–3463.