

Defect Detection Using 3D Computed Tomography Images and Application on Nuclear Power Plants

by

Md. Jamiul Alam Khan

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Faculty of Engineering and Applied Science
University of Ontario Institute of Technology (Ontario Tech University)
Oshawa, Ontario, Canada

December 2022

© Md. Jamiul Alam Khan, 2022

THESIS EXAMINATION INFORMATION

Submitted by: **Md. Jamiul Alam Khan**

Master of Applied Science in Electrical and Computer Engineering

Thesis title: Defect Detection Using 3D Computed Tomography Images and Application on Nuclear Power Plants
--

An oral defense of this thesis took place on November 29, 2022 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee:	Dr. Sanaa Alwidian
Research Supervisor:	Dr. Hossam Gaber
Co-Research Supervisor:	Dr. Jing Ren
Examining Committee Member:	Dr. Akramul Azim
Thesis Examiner:	Dr. Faisal Qureshi, Ontario Tech University

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

Tools used in nuclear power plant (NPP) inspection are required to be inspected before and after use on a reactor to check their integrity. To address the long duration required for manual inspection, non-destructive testing (NDT) can be implemented. In this thesis, a novel NDT framework is developed using key image processing functions to localize and identify the missing and misplaced components of NPP inspection tools using 3D CT data. Analyzing the limitations in image registration, a new algorithm is proposed to improve existing image registration techniques to handle significant rotational differences. Additionally, to address the annotated data-related issues in deep learning-based approaches, a semi-automated technique for 3D CT data annotation is introduced using the modified image registration process and Computer-aided design (CAD) design files. The annotated data is then used to train a deep learning-based semantic segmentation model to segment each component of the tool in the CT data.

Keywords: Computed Tomography, Defect Detection, Image Registration, Data Annotation

Author's Declaration

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Md. Jamiul Alam Khan

Statement of Contribution

I hereby certify that I am the sole author of this thesis, and I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others.

Some parts of this research have led to the following publication and conference presentations:

- H. A. Gabbar, A. Chahid, M. J. A. Khan, O. G. Adegboro, and M. I. Samson, “CTIMS: Automated Defect Detection Framework Using Computed Tomography”, *Applied Sciences*, vol. 12, no. 4, pp. 2175, February 2022, doi: 10.3390/app12042175.
- M. J. A. Khan, A. Chahid, O. G. Adegboro, H. A. Gabbar, J. Ren, “Semi-Automated Annotation of the CT data”, *International Conference on Disruptive, Innovative and Emerging Technology in the Nuclear Industry (DIET)*, Canada, 1-3 November 2022.

Acknowledgements

This work would not have been possible without the years of guidance and encouragement of my research supervisor Dr. Hossam Gaber and co-supervisor, Dr. Jing Ren, both from the Ontario Tech University (University of Ontario Institute of Technology - UOIT). They have been an inspirational mentor, and I am very fortunate to have had the pleasure of working with them. A special thanks goes to Dr. Akramul Azim for his support and guidance as a member of the supervisory committee. Furthermore, I am thankful to the thesis examiner and the thesis committee for helping me to improve and elaborate the explanation of the thesis work. I would also like to thank Dr. Abderrazak Chahid, Oluwabukola G. Adegboro, and members of the Smart Energy Systems Lab (SESL) of Ontario Tech University for their immense support throughout the research.

I am also thankful to New Vision Systems Canada Inc. (NVS) and Mitacs for both technical and financial support throughout the research. I also thank the collaborators from Diondo and Fraunhofer, Germany.

Dedication

This is dedicated to my parents namely Md Jalal Uddin Khan and Fahmida Khatun Akanda and to my dear wife Niama Nahid Rahman.

Table of Contents

Thesis Examination Information	ii
Abstract	iii
Author's Declaration	iv
Statement of Contribution	v
Acknowledgements	vi
Dedication	vii
Table of Contents	viii
List of Figures	xii
List of Tables	xvi
List of Abbreviations	xvii

1	Introduction	1
1.1	Background	1
1.2	Research Motivation	5
1.3	Problem Definition	6
1.4	Research Objective	8
1.5	Research Contributions	9
1.6	Thesis Outline	10
2	Literature Review	12
2.1	Defect Detection Using CT Data	13
2.2	Image Registration	20
2.3	Data Annotation	23
2.4	Limitations Observed	25
3	Research Methodology	27
3.1	3D CT Data Processing	31
3.1.1	Background Subtraction and Volume Registration	31
3.1.2	CAD Volume File	44

3.2	Image Processing-Based Defect Detection	48
3.3	Semi-Automated CT Data Annotation	52
3.4	Deep Learning-Based Segmentation Model	55
4	Experimental Evaluation	58
4.1	Description of the Case Study Dataset	59
4.1.1	Real CT Data	59
4.1.2	Annotated CAD Volume	61
4.1.3	Simulated CT Data	61
4.2	CT Data Processing	65
4.3	Image Processing-Based Defect Detection	67
4.4	CT Volume Data Annotation	69
4.5	Segmentation Model Training Using Semi-Automated Annotated Datasets	72
5	Results and Analysis	74
5.1	Updates to Registration Algorithms: Rotational Angle Estimation . .	75
5.2	Image Processing-Based Defect Detection	78
5.3	Semi-Automated CT Volume Annotation	80

6 Conclusion	84
6.1 Limitations	85
6.2 Future Work	86
References	89

List of Figures

1.1	Example of an NPP inspection tool (left) and corresponding CAD image of the tool showing internal components (right).	3
1.2	Current inspection workflow.	3
1.3	Proposed inspection workflow.	5
2.1	Few iterations of region growing from seed voxels [1].	14
2.2	Conventional algorithm for intensity-based registration.	21
2.3	Conventional algorithm for feature-based registration.	22
3.1	Summarized workflow of the proposed method.	29
3.2	CT data processing workflow.	32
3.3	CT scanning process of the tool.	33
3.4	Background subtraction example.	34

3.5	Different possibilities of placement of an object in CT scanner and the transformations required to register it according to the reference volume.	38
3.6	Cross-sectional view of reference (green) and input (red) volume before and after registration.	39
3.7	The registration algorithm proposed by mattes et al. [2]. \mathbf{x} is any geometric location in the reference image. P , P_T , P_R , and ∇_P are the joint, marginal test, marginal reference, and derivative of the joint distribution, respectively. L-BFGS-B is an optimization package that searches the parameter space of μ	42
3.8	Examples of volume registration using our DIPY+ method.	44
3.9	Examples of Individual components/parts of the tool.	47
3.10	General workflow of the proposed defect detection approach.	49
3.11	Workflow of the CT data annotation process.	53
3.12	Examples of generated mask data overlapping over the input CT volume.	55
3.13	The construction of the 3D U-Net model and its different components [3].	56

4.1	Few components of the nuclear reactor tool that are assembled to construct the NPP tool.	59
4.2	3D view and cross-sectional images from 3 different directions of a CT scans.	60
4.3	Defects introduced to different scans.	63
4.4	aRTist software interface.	64
4.5	Generated labelled CAD volume file.	65
4.6	Reference volume processing.	66
4.7	Example output from defect detection algorithm.	68
4.8	Example output from defect detection algorithm applied on simulated CT data.	69
4.9	Example of CT data (left) and annotated mask (right).	70
4.10	Example of generated class information file.	70
4.11	Example of segmentation prediction using trained model.	73
5.1	Comparison between registration algorithms before and after applying our pre-registration updates.	77

5.2	Training and validation loss and Training and validation IOU curve of the trained model using real CT data.	81
5.3	Training and validation loss and Training and validation IOU curve of the trained model using simulated data, annotated by the proposed annotation method.	82
5.4	Training and validation loss and Training and validation IOU curve of the trained model using simulated data that did not require annotation.	83

List of Tables

2.1	Summary of existing defect detection methods.	18
2.1	Summary of existing defect detection methods (continued).	19
5.1	Summary result of the defective and non-defective classification of input scan.	79

List of Abbreviations

2D:	Two-Dimensional
3D:	Three-Dimensional
4D:	Four-Dimensional
ANN:	Artificial Neural Network
aRTist:	Analytical RT Inspection Simulation Tool
CAD:	Computer-aided design
CNN:	Convolutional Neural Network
CPU:	Central Processing Unit
CT:	Computed Tomography
DNN:	Deep Neural Networks
FCN:	Fully Convolutional Network
GAN:	Generative Adversarial Networks
GPU:	Graphics Processing Unit
IOU:	Intersection Over Union

LIDAR:	Light Detection and Ranging
MRI:	Magnetic Resonance Imaging
NDT:	Non-Destructive Testing
NPP:	Nuclear Power Plant
PSNR:	Peak signal-to-noise ratio
RCNN:	Region-Based Convolutional Neural Network
RF:	Random Forest
SSIM:	Structural Similarity Index
STL:	STereoLithography
SVM:	Support Vector Machine

Chapter 1

Introduction

This chapter introduces the background and motivation of this research explaining the importance and the problems related to tool inspection in nuclear power plants (NPP). Furthermore, the research problem and objectives of the thesis are discussed. Finally, a summary of the contribution of the thesis and the outline of the whole thesis are included in this chapter.

1.1 Background

About 10% of the electricity that the world needs is provided by approximately 440 operable nuclear power reactors [\[4\]](#). In addition, about 55 more reactors are in the

construction phase. In order to minimize risks to workers, the public, and the environment, NPP undergoes regular inspection and maintenance activities [5]. These inspection activities require several tools to be used in the reactor and a well-planned maintenance activity includes the inspection of the tools used for maintenance as well. There are many inspection tools which are used in reactors, they should be inspected before and after being used in the reactor. Not only during inspection operation but also tools are required to be inspected after purchase or fabrication to check their integrity. The tools are inspected after purchase or fabrication, and before entering the vaults to ensure their readiness to be used in the reactor, the use of any damaged tool could lead to accidents and additional risks to the reactor. After usage for reactor inspection, the tools are inspected again to check if there are any missing or broken components of the tool left inside the reactor. Any foreign object, such as broken components of the inspection tools left inside the reactor, could lead to accidents. Hence, the inspection of tools in NPP is very crucial for operational safety. An example of a tool used in NPP inspection along with the corresponding CAD image is shown in Figure 1.1.

The existing inspection process for tools in nuclear power plants is mainly manual. However, tools used in NPP consist of several smaller internal components, making them difficult for visual inspection without disassembling. Hence, it requires both

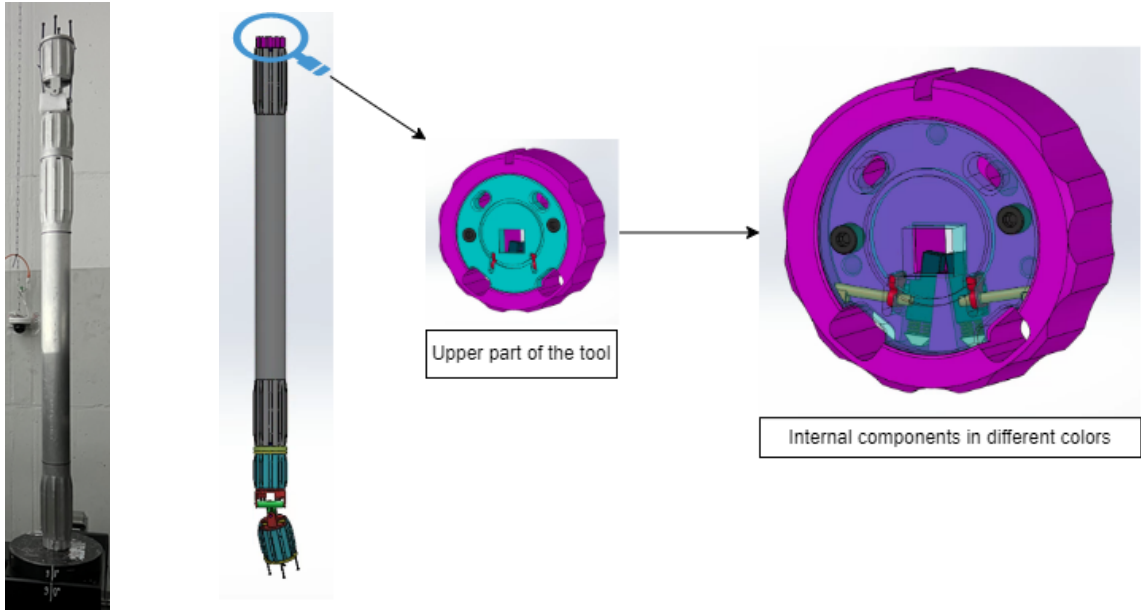


Figure 1.1: Example of an NPP inspection tool (left) and corresponding CAD image of the tool showing internal components (right).

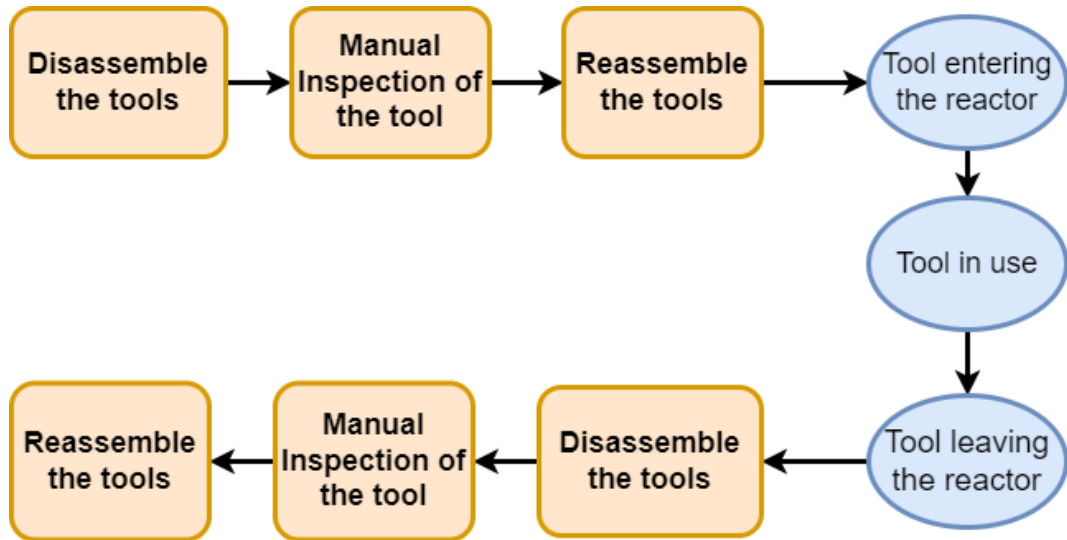


Figure 1.2: Current inspection workflow.

time and labour. Depending on the complexity of the tool, the manual inspection process lasts from several minutes to months. The current inspection workflow is shown in Figure 1.2, which includes disassembling the tool, visual inspection, and then reassembling the tool. Wait time before disassembling due to radiation from the tools used adds further delay to the process. As any remaining objects inside reactor vaults are dangerous, the reactor has to stay shut down until every object that has entered the vault can be clearly identified outside of the vault. The reactor remaining shut down means there is no production, the longer the outage time is the greater the loss in production. Therefore NPP continuously strives to reduce the outage time due to inspection and maintenance activities. Reduction of just 12 hours of downtime in a reactor can generate a saving of around \$450,000 [6]. Moreover, historically human failures occurred in nuclear power plants [7]. The manual inspection process has a risk of human-related errors. The human labour involved in the process additionally has the risk of health issues due to radiation from the tools. For these reasons, manual inspection of tools can lead to loss of time, production, money, health, and resources.

The losses occurring due to the manual inspection process can be addressed with an automated and fast solution. An automated system shown in Figure 1.3 is proposed to check the integrity of tools during the maintenance and inspection process, which will reduce outage time and human labour involved. The proposed framework

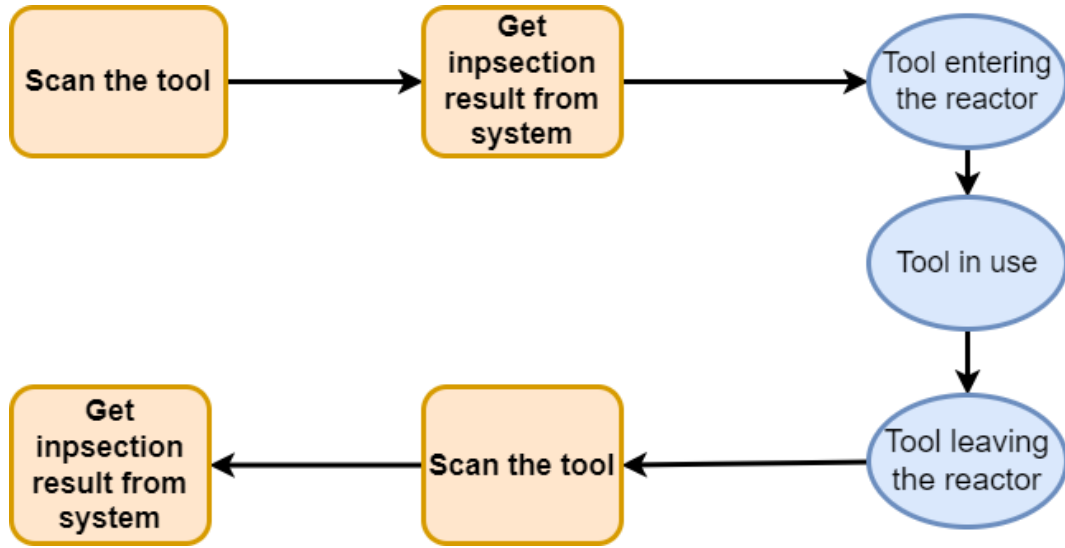


Figure 1.3: Proposed inspection workflow.

will collect the computed tomography (CT) volume images of tools before entering the vault and after leaving the vault to find out if there is any missing or broken component in the tool, or if there is any defect in the tool.

1.2 Research Motivation

As discussed earlier, the manual inspection process is time-consuming for tools consisting of different internal components. Therefore, the proposed research targets detecting defects in industrial tools using a Non-destructive testing (NDT) technique that can reduce the time required for internal components inspection. CT scans provide an NDT approach to inspect the internal structure of the tool, without dis-

mantling the tool. 3D CT is being used in various NDT-related applications including quality control [8, 9], medical diagnosis [10, 11], and material analysis [12, 13]. However, using CT technology in the defect detection of nuclear reactor inspection tools is not much investigated yet. A complete framework including the process of scanning reactor tools, processing the scanned data, and generating inspection results is required to be developed for NPP, which will reduce the time required for inspection by several folds. Therefore, the framework will also reduce outage time, labour, and human errors. This will eventually lead to more profit and increased safety in NPP. However, this thesis demonstrates the techniques of processing scanned CT data and analyzing them to detect defects. The CT scanning system to be combined with the proposed defect detection framework is developed by other research collaborators.

1.3 Problem Definition

Different techniques of defect detection using CT images are developed using image processing, conventional machine learning, and deep learning-based functions. The image processing-based approaches are commonly based on comparing input CT volume with reference CT volume [14]. These techniques require the specimen to be placed in the scanner using the same position and orientation and is to be compared with the reference. However, the placement of objects in the scanner is done manually

[15], making the precise placement of tools in the scanner very difficult. To handle this issue image registration is used to align the CT data [14] when they are not scanned in the exact alignment. However, in the case of significant orientational differences (in the case study of the thesis, this refers to the rotational difference when placing the tool into the scanner), registration algorithms struggle to align the CT volumes. This rotational difference must be handled properly to get accurate inspection results. Although suggestions for using robots for specimen placement were proposed [16, 17], this also requires research to make the system less complex and affordable.

Deep learning-based approaches [18, 19] are proposed, however, lack of sufficient training data is causing CT-based inspections to lag behind the success of deep learning in other fields, such as CT-based applications in medical field [20]. Manually annotating 3D datasets is expensive and sometimes requires expert presence. Hence, simulated data is used in model training, one such model is trained by Fuchs et al. [20]. However, using a large amount of simulated data on models can lead to overfitting when actual CT data is introduced. For these reasons, the process of automated CT data annotation is required to be researched for more applications based on CT data.

The defect detection function proposed by Szabo et al. [14] detects defects such as microscopic cracks, porosity, and voids in the additive manufacturing industry.

Other developed functions such as the one proposed by Ghorai et al. [21] look for defects in flat surfaces, and the function proposed by Sarigul et al. [22] looks for knot, decay, split, and bark in clear woods. Therefore, we see defect detection functions are generally developed according to the type of defects and object being inspected. In this research, the integrity of an NPP inspection tool will be inspected to check if there are any missing or misplaced internal components. Along with detecting defective areas in the NPP inspection tool, the defective components are identified by names. This would be the first approach to identify the defective components by name.

1.4 Research Objective

The main goal is to develop an integrated framework for defect detection using 3D CT scans using novel algorithms. The developed framework will analyze 3D CT scan data of nuclear reactor inspection tools and detect, characterize, and localize the defect of the tool. The following are the objectives to be completed to achieve the main goal.

1. **Develop an integrated framework for nuclear reactor tool inspection:**

An integrated framework will be developed using image processing-based functions that will be able to identify defects from 3D CT scans of the tools.

2. **Develop an intelligent algorithm for accurate registration with large rotational angle difference between 3D volumes:** Placement of all objects in a CT scanner for scanning is challenging. Scans might have the objects rotated in a different orientation in different scans. While existing algorithms fail to register the volumes with a large rotational difference, an improvement function to the existing algorithms will be developed.
3. **Develop an integrated algorithm for semi-automated 3D data annotation:** Using the nuclear reactor inspection tool's CAD design files and 3D CT scans, a data annotation system will be developed to generate data to train a semantic segmentation model. The trained model should be able to segment different components of the tools.

1.5 Research Contributions

The main contributions of this research work can be summarized into the following main points:

- An integrated image processing-based defect detection system for the integrity check of a nuclear reactor inspection tool is developed. The system classifies the volume data into defective and non-defective classes and localizes as well

as visualizes the defect area. Additionally, the names of missing or defective components are identified upon the availability of the tool's CAD design files.

- An additional function for improving the existing registration algorithms is developed to be able to register volume data with large rotational differences. The developed function determines the rotational difference between the reference and the input volume. After determining the difference angle, the input volume is rotated based on the angle before running registration functions.
- A data annotation framework that can annotate a large amount of CT volume data for semantic segmentation model training is developed. The framework generates the ground truth mask data using the CAD design file of the tool. Then, the generated data are used to train a segmentation model to validate the data annotation.

1.6 Thesis Outline

Chapter 1 introduces the background and motivation, problem definition, research objectives, and contributions. Chapter 2 of the thesis discusses a few related works on defect detection using CT data. It further discusses techniques used to overcome challenges in defect detection using CT data. Chapter 3 explains the proposed

methodology in detail. The case study data and implementation of the proposed methods are explained in chapter 4. Chapter 5 demonstrates the results from the proposed approach and discussions on the achieved results. Finally, chapter 6 is the conclusion chapter explaining the limitations, and future works related to the thesis.

Chapter 2

Literature Review

The approach of using CT data for defect detection has been implemented in various fields, including manufacturing, assembly, food, and medical diagnosis. These defect detection techniques are developed using image processing, conventional machine learning, and deep learning-based approaches. The thesis's proposed methodology includes image processing and deep learning techniques. Hence, the literature review includes research works from both approaches. 2D image or 3D volume registration is integral to image processing-based defect detection. Furthermore, data annotation is an essential part of deep learning-based techniques. Therefore, the literature review includes discussions on works related to image registration and data annotation as well.

2.1 Defect Detection Using CT Data

Computed Tomography (CT) is one of the most used and trusted methods for non-destructive testing (NDT). Image processing-based algorithms are based on techniques such as image subtraction and region growing. In deep learning-based approaches, images are fed into models, and deep segmentation or classification algorithms are applied.

A common approach in image processing-based algorithms is to use reference defect-free images. In such cases, the input CT images are compared with the reference images to identify the defects. The steps in the proposed method [14] include CT image enhancement, image registration, subtraction, segmentation (background removal), morphological operations, and defect characterization. CT image enhancement or restoration is mainly the task of de-noising and image contrast enhancement. They used Wiener filter [23] to de-noise the images. Then, before subtracting the input images from the reference image, they are registered using an intensity-based registration technique [24]. The resulting image from subtraction is segmented using the thresholding method. Morphological processing is applied [25] to remove the minor residual artifacts from the segmented image, then the clean image is analyzed for defect characterization. The proposed method resulted in results in both 2D images

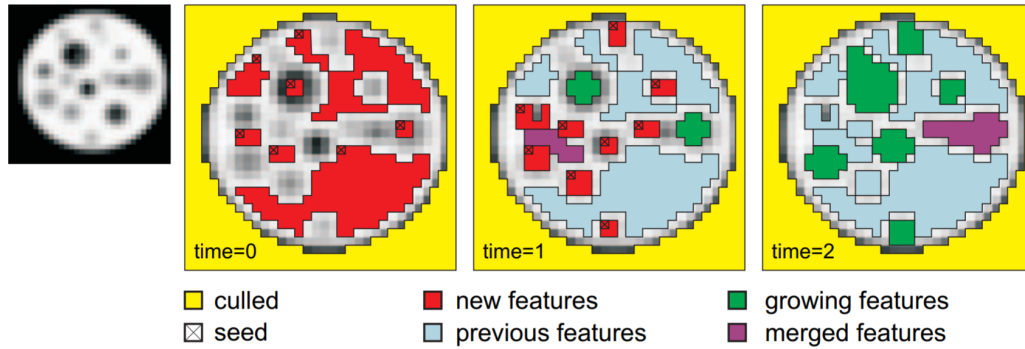


Figure 2.1: Few iterations of region growing from seed voxels [1].

and 3D reconstructed volumes. Although they seem to have produced good results, this approach has some limitations. The objects to be inspected must be placed in the same orientation and preferably in the same position and angle as the reference object was placed. This is challenging to place objects in exact positions every time. If they are not aligned properly, the results are inaccurate. Region growing segmentation is used by Hadwiger et al. to detect defects in industrial CT volumes [1]. They manually select the seed voxels, and the algorithms check all neighboring voxels for similar or grayscale values within the range. The process is repeated several times until the features merge as shown in Figure 2.1. In each iteration, new voxels are identified by similar features and they may grow and merge with existing features. Then the segmented regions are analyzed for size and density to classify them as defects. Although they could produce results, the method required manual tasks like

seed selection and parameter selection. Another approach using segmentation, feature extractions, and classification was proposed for defect detection in aluminum CT images [26]. Other image processing-based approaches include the use of the Kriging model with statistical models to compute the shape deviation errors [27, 28].

Random forest (RF) is used to identify defects in aluminum casting as well [29]. RF classifies the discontinuities into defects based on intensity and shape. Although the proposed method achieved a 94% detection rate in their dataset, there are some limitations. The process is time-consuming, and they need to address how to handle the defects with varying intensity and shapes. Support vector machines (SVM) is used for defect detection on flat steel products by Ghorai et al. [21].

A deep learning-based segmentation technique was used in detecting defects from cast aluminum parts [20]. A model was trained with a modified version of U-net [30] model architecture to segment the defects from CT images. Due to a lack of available CT data, they opted to use simulated data for model training. The model was then tested with CT data and achieved an accuracy of 65%.

There are applications based on CT technology for defect detection in hardwood logs [22]. They researched automatic detection and identification of defects in hardwood logs. They used CT images to identify internal defects. Previously, some researchers used an Artificial neural network (ANN) approach to classify the defects

from CT images [31]. In this paper [22], an ANN was designed to classify the log image pixels into five different groups (“knot,” “decay,” “split,” “bark,” or “clear wood”). Although this approach produced some excellent results, the variability of the woods made it difficult to reach complete success. So the researchers developed a post-processing module that uses domain knowledge to improve the classification result. ANN only works pixel-by-pixel; they ignore the defect’s shape, size, or position on the logs, which sometimes creates false results. The paper [22] mainly focuses on the post-processing module, so they selected an ANN with relatively poor performance for the classification. Their implementation of ANN has an accuracy of 85.1%. After the ANN provides tentative labelling to the pixels, the post-processing starts. The post-processing steps are mainly based on the tools of Mathematical Morphology. Mathematical morphology is popularly used to modify image properties such as shapes, reduce noise and detect features from images. Every region identified by the ANN has individual post-processing methods. The post-processing module for bark regions generates a filtered bark image, removing the regions that are not on the log’s border. The post-processing module for the knot region removes the isolated knot pixels to generate an output image. From the split images, the post-processing module generates more apparent and more accurate split marks in the image. An image of clear wood is generated with the openings filled using the identified clear wood pixels.

After performing functions on all layers, the results are combined into a composite result. When there are overlaps between the pixels, the precedence rules determine which label to apply. Although the developed post-processing module improved the defect labelling accuracy of ANN by 6%, there are more areas of improvement, such as defining the overlapping labelling or even defining more defect types, etc.

Using an object detection algorithm based on RetinaNet with ResNet-101 and novel data augmentation methods, Cheng et al. [32] proposed a defect detection system. The algorithm was tested to detect internal defects in the wheel hub and generated good results. A weakly-supervised Convolutional Neural Network (CNN) model for classification was again proposed to detect defects in the casting industry [19]. First, images were divided into defects and non-defect classes; then, the model was trained to classify these images.

A summary of the methods discussed in this literature review is presented in Table 2.1. It shows the approach taken, the type of defects targeted, and some remarks about the method.

Table 2.1: Summary of existing defect detection methods.

Paper	Detection approach	Type of defects	Remarks
Szabo et al. [14]	Image processing	Defect detection in additive manufacturing of near Net-Shape parts.	They detected defects such as microscopic cracks, porosity, and voids in additive manufacturing of near Net-Shape parts. The approach was to compare the input CT image with the reference image to detect the defects. Although it produced good results with some data, it struggled when the 3D scans had object placement at a different orientation.
Hadwiger et al. [1]	Image processing	Defect detection in cast metal parts.	The region-growing segmentation process is used to detect the defects. The process requires manual tuning of various parameters, such as target ranges of density and size, alongside the selection of seed points for each volume, which increases the processing time.
He et al. [26]	Image processing	Defect detection in industrial parts.	Defects are detected based on segmentation and feature extraction. However, further investigation is required to be able to detect more forms of defects.

Table 2.1: Summary of existing defect detection methods (continued).

Paper	Detection approach	Type of defects	Remarks
Zhao et al. [29]	Machine learning	Defect detection in aluminum casting.	Random forest (RF) was used to classify the discontinuities into defects based on intensity and shape. The process is time-consuming, and the intensity and shapes are not always consistent among defects, which were not appropriately addressed.
Ghorai et al. [21]	Machine learning	Defect detection on hot-rolled flat steel products.	Surface-level defect detection struggles with uneven illumination or vibrations while acquiring the images. Furthermore, there is a challenge in generating training data.
Fuchs et al. [20]	Deep learning	Defect detection in cast aluminum part.	A segmentation model was trained to segment defects from CT images. The model was trained with simulated CT data. When tested with real CT data, an accuracy of 65% was achieved.
Sarigul et al. [22]	Deep learning and image processing	Defect detection in hardwood logs.	Artificial neural network (ANN) was used to classify the image pixels: “knot,” “decay,” “split,” “bark,” or “clear wood.” The pixels were further processed to identify defects. Although this approach generated good results, there are more areas of improvement, such as defining the overlapping labelling or even defining more defect types, etc.

2.2 Image Registration

Image registration means aligning one image or volume according to a reference volume. Registration is required to analyze and compare images or 3D Volumes when they are not aligned. Based on the transformation type, image registration can be of several types: rigid, affine, and deformable. Rigid and affine transforms are used when the object in the image should not be warped or deformed. Rigid transformation does the work of translation and rotation of the input image. Affine transformation improves rigid transformation by translating, scaling, rotating, and shearing the input image. Deformable image registration is used in cases where the object's structure in the image can change dynamically, mainly in the case of medical images. In the thesis, tools that have specific shapes are inspected, so rigid and affine transformation techniques were explored for review.

Several registration methods have been presented with different criteria over the years. The approaches change according to the data dimensionality (2D, 3D, 4D), data source (MRI, CT, LIDAR), region of interest, and application. Registration methodologies based on the voxel or pixel intensity are known as intensity-based registration. Those based on geometric features, such as structures from images, are called feature-based or geometrical-based. Figure [2.2](#) and [2.3](#) show the typical

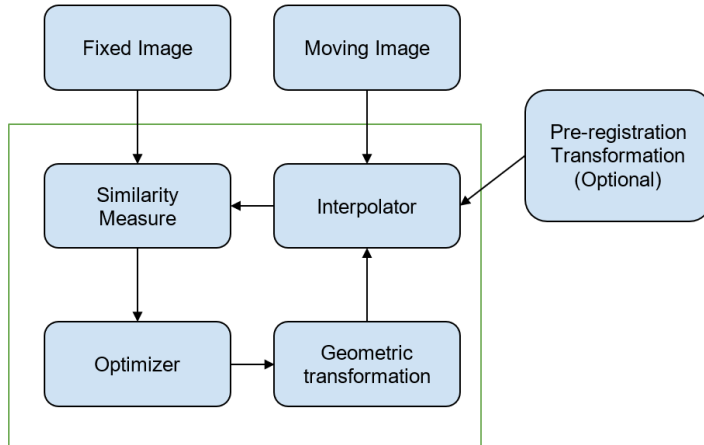


Figure 2.2: Conventional algorithm for intensity-based registration.

workflow of intensity-based and feature-based registration approaches [33].

Deep learning is used for image registration algorithms as well. The first deep learning-based rigid registration approach was proposed by Miao et al. [34, 35]. They used CNN to predict the transformation matrix for registration. The model was trained with data generated by transforming aligned volumes. Their experiment showed improved registration; however, the dataset used had slight rotational differences (less than 30 degrees) [36]. Sun et al. [37] also used synthesized training data to train a model for registration. The network was designed and trained to predict affine transformation matrices for CT-Ultrasound images. Although the process worked for synthesized data, it did not work fine for actual CT-Ultrasound images due to the appearance difference between the simulated and real data [38].

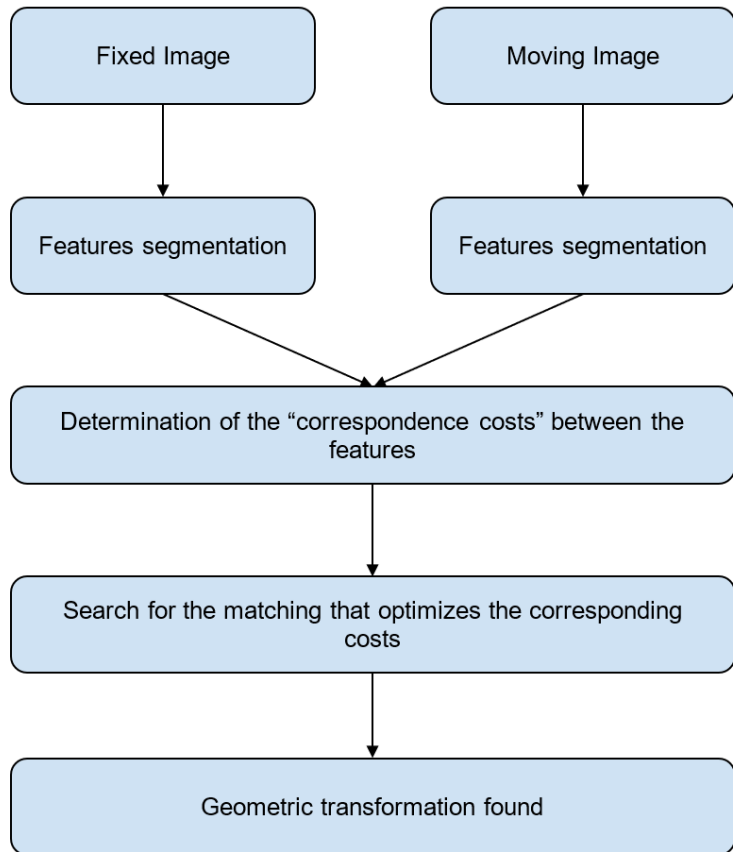


Figure 2.3: Conventional algorithm for feature-based registration.

2.3 Data Annotation

A survey by M'akinen et al. [39] found that the top problems of using deep learning in industrial applications are all related to datasets. The problems include lack of data, quality of data, and accessibility of data. For supervised machine learning applications, labelling the data is primarily manual work. The images or volume data for segmentation-related datasets are labelled using tools that require much manual work. Generating a dataset consisting of large amounts of data with high quality requires much time and is expensive. This factor is one of the main obstacles to using deep learning in more and more applications [40]. The data annotation challenge exists in applications on agricultural use-cases, medical use-cases, rock fractures, nanoparticles, etc [41].

The dataset for the segmentation task generally consists of input and corresponding mask images. Each pixel of the mask image is annotated according to the object they belong to. Generally using conventional labelling tools, human experts identify the bounding pixels of each object and do the annotation. Manual labelling of image data is more difficult as different images have different properties and pre-processing requirements. When the data is 3D, the work complexity increases more. Even though all these challenges are there, the use of deep learning is increasing day by day. So a

significant amount of research and developments are being carried out to make data labelling more automated and efficient. Due to the manual data annotations challenges discussed above, the use of deep learning-based solutions is still limited. Most systems are developed with a single objective or for a specific domain; when the data or item changes, the system does not work. An automated data annotation system can be used to generate a large amount of data for different domains. The developed DNN models can be implemented for different scenarios faster, as training data can be annotated automatically. For example, if a segmentation model is developed for one specific object, it can easily be implemented for different objects.

Many techniques are implemented and developed to handle data-related issues. The most widely used technique to increase the size of a dataset is the use of data augmentation [30]. Different transformations, such as rotation, color transformation, and flipping, are applied to existing data to generate new data. However, this approach cannot synthesize new properties to data. A more advanced approach to increasing the data size is using Generative Adversarial Networks to produce synthetic data [42]. This approach can increase the complexity of data augmentation. Usage of transfer learning and self-supervised learning techniques are similarly being explored to decrease the requirement of large datasets [43]. Even after all these, the importance and requirement of data annotation are still there.

Techniques are researched to automate and find more efficient processes of data labelling. Extreme clicking [44] was proposed to decrease the time required for data annotation. Instead of box dragging, this approach lets the user click the four most extreme points of the object for annotation. Human-augmenting AI-based labelling system [43] keeps a human in the loop for the process, but when the human labels data, another active learning algorithm learns from human labelling and suggests the following labels for human experts to approve. This overall decreases the workload for data annotation. The use of mask-RCNN was proposed to be used for data annotation for sand-like granular instance segmentation [40]. The proposed method allows the production of a large dataset for granular object segmentation without manual labelling.

2.4 Limitations Observed

Different approaches to defect detection using CT images come with different challenges due to data, defect variations, and data limitations. Image processing-based approaches struggle when the images cannot be aligned accurately (registration algorithms fail). This issue comes from the scanning stage of the object. For accurate defect detection, the object's placement in the scanner must be precise or close to how the reference object was placed. The discussed defect detection system can localize

the defect; however, in the case study of the thesis, the identification of the missing or defective component by name is investigated. Although deep learning-based techniques are being developed, the training and testing in most cases are being carried out with few publicly available data or simulated data. Those do not guarantee good performance when the algorithms are used with actual CT data.

Image registration can handle orientation differences such as rotation and tilt but only in small ranges. Image registration algorithms struggle to register 3D volumes when it has a large rotational difference (more details in chapter 3). Deep learning-based image registration algorithms are also proposed; they are primarily trained with simulated data, and their performance with actual CT data might not be satisfactory.

Chapter 3

Research Methodology

Developing the defect detection system using 3D CT data is a step-by-step process. The developed framework has four primary functions: CT data processing, image processing-based defect detection, CT data annotation, and deep learning-based segmentation model. First, the input CT data are processed and prepared for defect inspection. Then the image processing-based defect detection function looks for defects in the input data. If the CAD data of the tool being inspected is available, the defect detection function can identify the defective components by name. Moreover, upon the availability of CAD files, the deep learning-based technique for defect detection is explored. A CT data annotation function is developed to annotate 3D CT data. This data is then used as input to train a deep learning-based semantic

segmentation model in order to segment each component of the tool in the CT data.

A summarized workflow of the proposed method is shown in Figure 3.1.

Some terminologies that are used frequently are the following:

- **Volume data:** A 3D array that represents the scanned object data. It is reconstructed based on 2D X-ray projections at different angles.
- **2D Slice:** A CT volume can be deconstructed along a specific direction into a stack of 2D images called slices.
- **Voxel:** A voxel in 3D volume corresponds to pixels in 2D images. 3D volumes consist of these volumetric pixels called voxels.

The research is based on test data provided by the collaborators, New Vision Systems Canada Inc. (NVS). Diondo, an experienced company designing different CT systems, generated the CT data. A nuclear reactor inspection tool was scanned using a CT scanner, and the scanned data was provided for use in this research. The provided data are the 3D volumes after reconstructing the CT projection images. Hence, there was no need for the proposed methodology to further explore 3D CT reconstruction algorithms needed to generate 3D data from 2D projection images.

Although using 2D CT data is a faster and less resource-consuming option to work with, this thesis explores the usage of 3D CT volume data. Working with 3D volumes

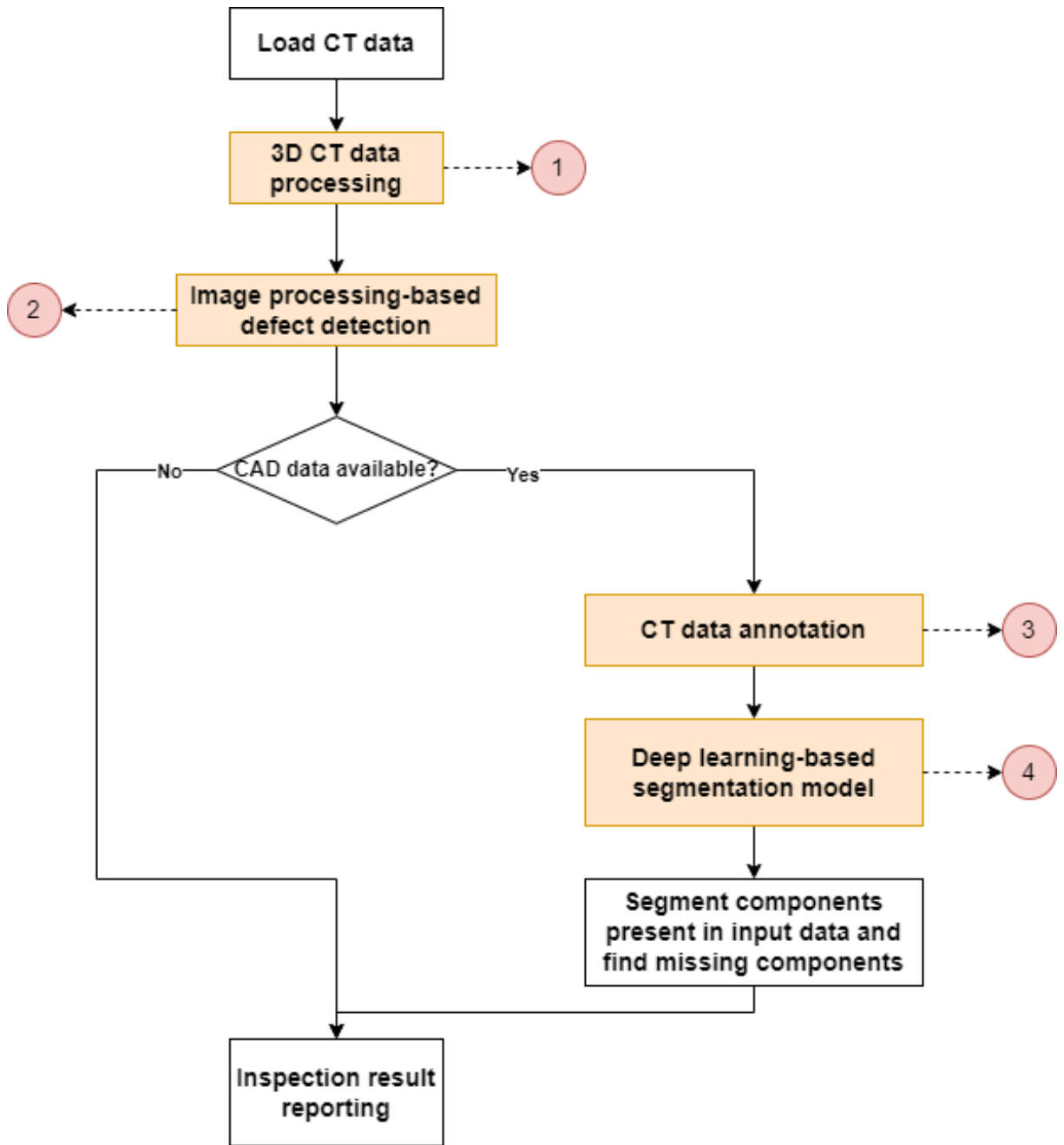


Figure 3.1: Summarized workflow of the proposed method.

would allow the system to display the defects clearly. Additionally, this thesis works on identifying the defects by name and displaying the defect component in 3D so the user can identify the defective component or area faster in the actual tool by looking at the display. In addition, defect detection using 2D data is investigated by our research colleagues and collaborators [45].

3D data can be represented in different formats. The most popular representation formats are voxel grid and point cloud. The input format is selected based on the developed system and the type of application. Point clouds are simply collections of points in 3D space where each point is specified by a coordinate (x,y,z) location. Voxels are like pixels in 3D. They are similar to point clouds except that they are quantized, fixed-sized point clouds. Unlike point clouds, voxels have fixed sizes and discrete coordinates. The thesis considers 3D volume data as voxels. Although voxels require higher memory and computing power to work with than point clouds, voxels are the closest representation of 3D data to images. Furthermore, the conversion of CT data to the point cloud is a complex process including several functions [46][47]. Also, having a fixed size and structure allows traditional convolutional layers to be used for feature learning in voxel data, whereas it is challenging to use traditional deep learning methods for point clouds as they are translation, rotation, and permutation invariant. Considering all these, voxels were selected as the 3D data representation

format for the thesis.

The CAD files of the tool used to generate the CT data are also available, so CT data annotation and deep learning-based techniques could be explored. A properly trained deep learning-based defect detection will allow the system to inspect 3D CT data faster as image processing-based 3D CT data processing and defect detection takes more time. In cases where the CAD files are not available, for example, if the designer company of the tool does not share their design files, then only the image processing-based defect detection function will be executed.

3.1 3D CT Data Processing

3.1.1 Background Subtraction and Volume Registration

The process starts with either volume data or a series of 2D slices of CT volume. If the input is a series of 2D slices, the module constructs the 3D volumes from 2D slices by stacking them along the sliced axis. CT volume data comes with some background noise data. Furthermore, the volumes are not always aligned; sometimes, they can get tilted or rotated for a bit. Before applying any defect detection algorithms, the data must be prepared by removing the background noise and aligning all the volumes according to one reference volume so they can be compared. Figure [3.2](#) shows general

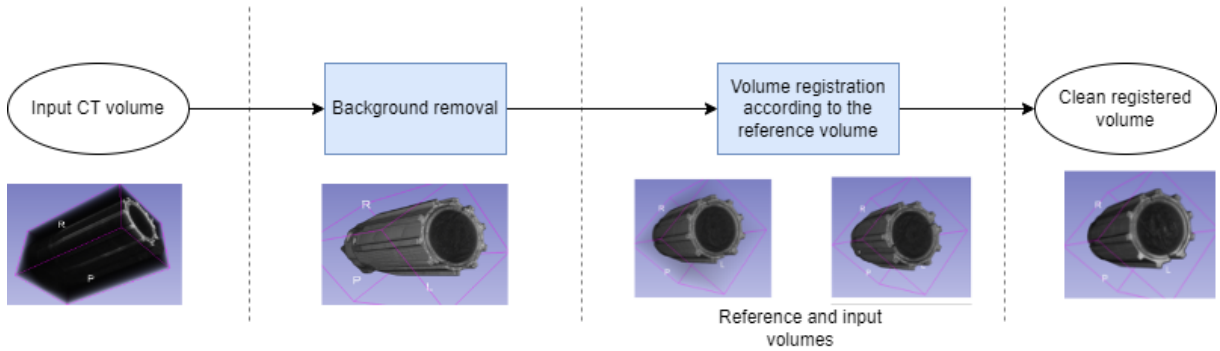


Figure 3.2: CT data processing workflow.

workflow of CT data processing.

The case study CT data for the thesis was generated by collecting a large number of 2D projection scans (813 projections) over 195 degrees of rotation. 2D projections are then converted to 3D tomography (3D volume) using the FDK (Feldkamp, Davis, and Kress) algorithm [48]. Figure 3.3 shows the CT machine and how the tool was placed in the machine for scanning.

A high level of x-ray dose was used in each CT data generation. Hence, unlike most CT volumes, the case study data has a lower chance of having artifacts to denoise and does not require extensive image enhancement. Median filter [49] is used for denoising the CT volume. The median filter can effectively remove tiny noises (salt and pepper) while preserving the edges. The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. If the value of the pixel is needed to be replaced,

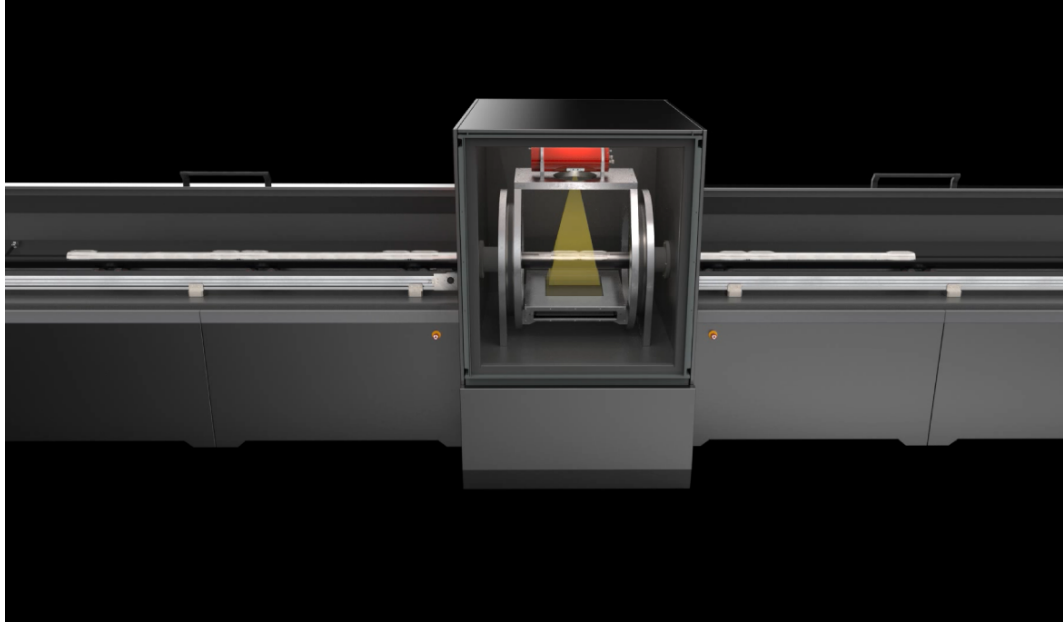


Figure 3.3: CT scanning process of the tool.

it is replaced by the median value of the neighboring pixel values. The equation 3.1 defines the filter for 2D input, where R is the defined neighboring region.

$$I'(u, v) \leftarrow \text{median}\{I(u + i, v + j) | (i, j) \in R\} \quad (3.1)$$

Following the exact principle, the median filter function developed for 3D input is used to denoise the input 3D CT volumes.

There are several thresholding algorithms where the threshold values are automatically determined. The otsu thresholding is one of such algorithm. The algorithm separates the volume voxels into two categories: foreground and background, then

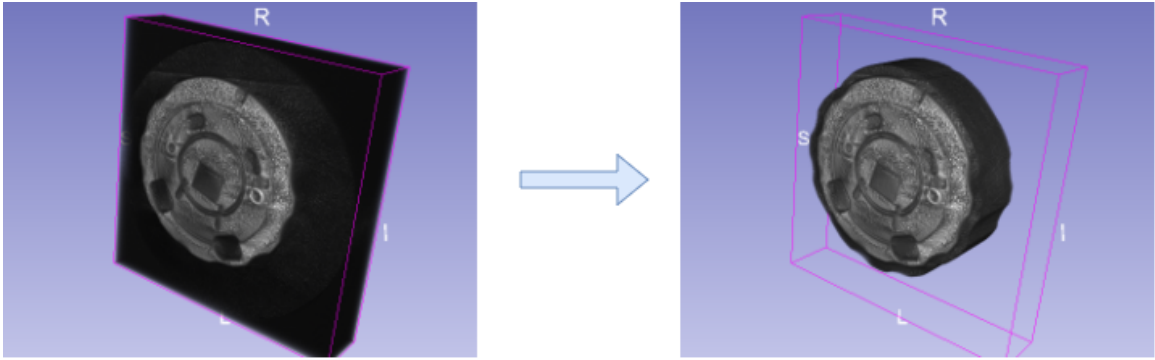


Figure 3.4: Background subtraction example.

it returns the foreground voxels' binary mask. The technique is to search for the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes. It divides all the pixels into two clusters. It minimizes the intra-cluster variation by maximizing the inter-cluster variance. Finally, it returns a single intensity value which is called a threshold value. The background noise of CT data is removed using otsu thresholding, and the binary mask volume is generated. By keeping the binary mask as the reference, the region consisting only of the tool in the CT volume is cropped out as the clean volume data. Scikit-image [50], an open-source image processing library, was used to implement the median filter and otsu thresholding functions. Figure 3.4 shows an example where the background data was removed from the CT volume.

Once the background noise from CT volumes is removed, the next step is volume registration. The input volume is registered according to the reference volume (back-

ground subtracted non-defect volume). Affine transformation for the volume registration is used. Affine transformation does translate, rotating, scaling, and shearing of the input volume. As the tool being inspected is placed in the CT scanner by hand, the placement of the tool for each scan is not the same. Based on the difference in how the tool is placed, several transformations are required for registration. Several transformations can be combined into a 4 by 4 affine transformation matrix, M . Where,

$$M = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

A voxel (x, y, z) is transformed by performing multiplication:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = M \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Translation of the input volume moves the voxels of the input volume along the three axes. The matrix for a simple translation can be represented as T :

$$T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation of the input volume rotates the input volume along one of the coordinate axes. The rotation matrix $R_x(\theta)$ along x-axis can be represented as:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Scaling transform changes the size of the object by expanding or contracting. The transformation matrix for scaling can be represented as:

$$S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Shearing transformation handles the tilting of objects. A transformation matrix

expressing shear along the x-axis can be represented as:

$$S_x(\theta) = \begin{pmatrix} 1 & \cot(\theta) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 3.5 shows how objects can be placed in different positions and which volume data transformation is required to register the input CT data. The gray colored box represents the area where the scanner collects the radiograph images.

Several existing affine registration techniques and software were used on case study data, but they failed to register the volume when there were significant rotational differences between them. Figure 3.6 shows some outputs from different approaches where input volume has an estimated rotational difference of around 40 degrees. Researchers mostly do not share their codes publicly, therefore the registration algorithms available publicly were used to register volumes. Pystackreg [51], a python extension for automatic image alignment, was used. It works on the slice-by-slice registration technique. Open-source software for image processing, Slicer [52], was also used to register the volumes. Finally, DIPY library [53] and SimpleITK [54] were applied to the case study data. In almost all cases, the translation of the volume was

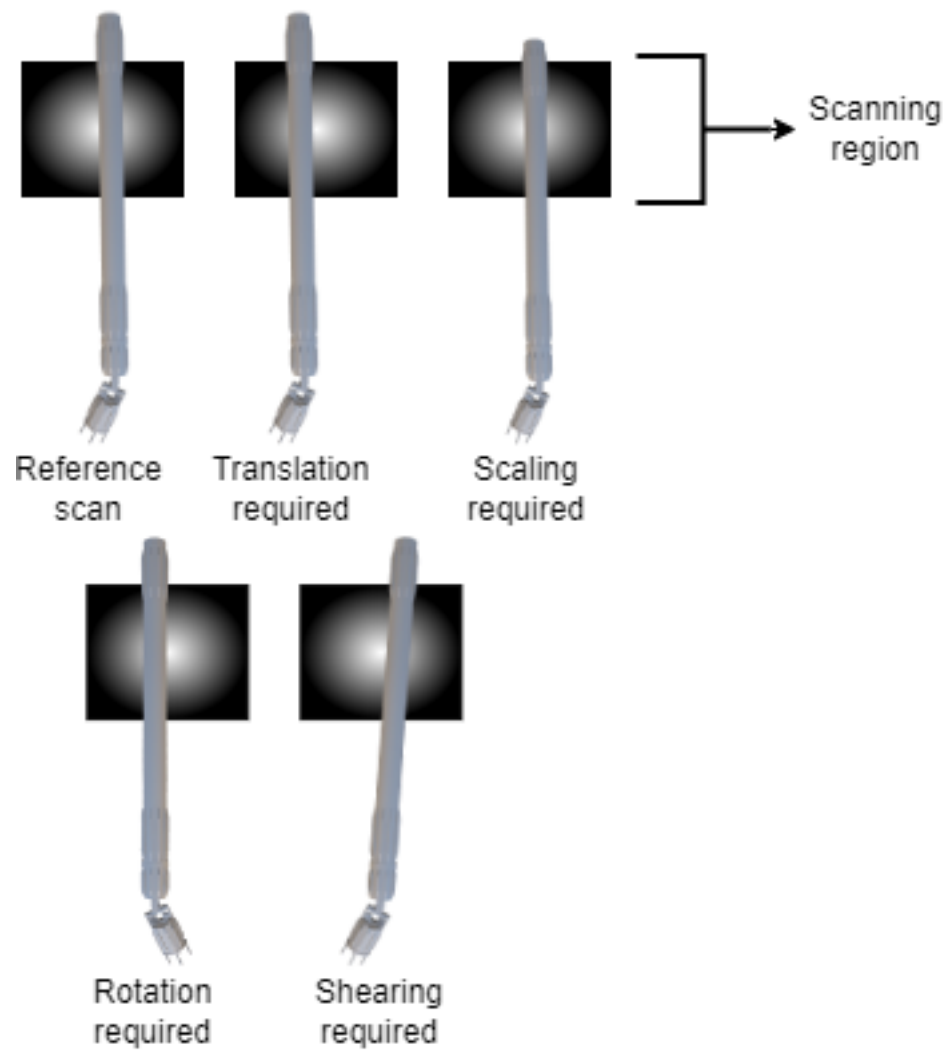


Figure 3.5: Different possibilities of placement of an object in CT scanner and the transformations required to register it according to the reference volume.

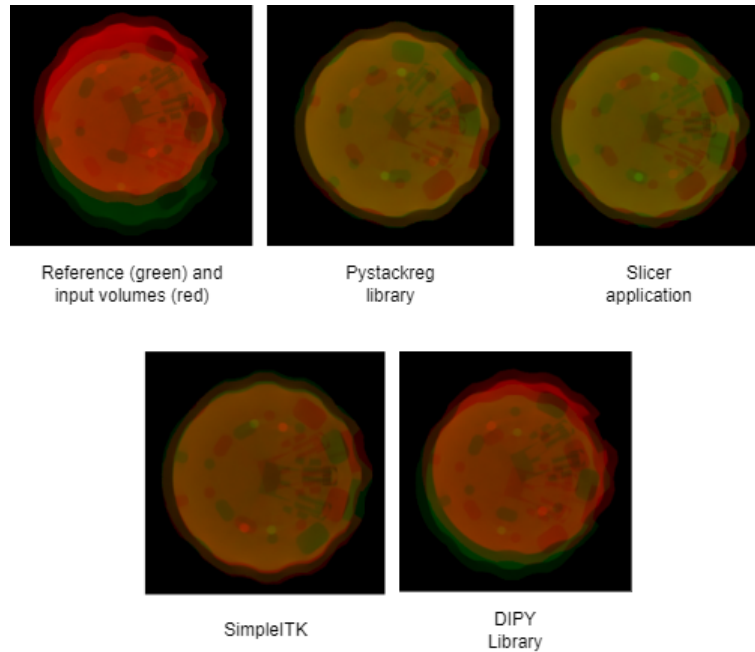


Figure 3.6: Cross-sectional view of reference (green) and input (red) volume before and after registration.

handled correctly. However, the rotation could not be handled for registration.

To register the volumes with existing volume registration algorithms, some initial processing of the volume data is required. An algorithm is developed to estimate the rotational difference between the input and reference volume.

For rotation angle estimation, five different slices of the volume from 5 different segments are selected from both the reference volume and the input volume. The slices from the input volume are rotated between -180 to 180 degrees and translated over the slices from the reference volume. The Structural Similarity Index (SSIM)

for each angle is calculated. SSIM is a method for calculating the similarity between two images. SSIM is calculated using the equation 3.2, where μ_x, μ_y are the mean intensities of the respective signals; σ_x, σ_y is the respective standard deviation and C_1, C_2 constants.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3.2)$$

The angle that produces the highest SSIM value is recorded in a list for each slice. The median angle from the list of angles is selected as the estimated angle of rotational difference. The input volume is rotated according to the derived angle. The volume is then resized to a smaller ratio (maximum 100 voxels in each dimension). The registration algorithm is applied and displayed (overlapping reference and registered volume) to the user for validation. If the user is satisfied with the registration, then the registration of the full-scale volume begins. Otherwise, the angle estimation algorithm runs again. This human verification step is important because, in some cases, angle estimation might fail if a substantial defect causes an overall structural change in the tool. In those cases, the tool is automatically understood to be defective, and further operation is unnecessary. The initial registration is done to a small volume to reduce the time required for registration. 3D volume registration is a computationally complex process; the larger the volume size, the more time it takes. The time required for angle estimation is also reduced by executing functions on smaller-sized volumes.

The registration algorithm used in the thesis is developed using DIPY library [53]. The library function does the affine translation operations to register 2D images or 3D volumes. The library computes the affine transformation of two 3D volumes by maximization of their mutual information as proposed by mattes et al. [2]. The target of this [2] methodology is to find a set of parameters $\bar{\mu}$ among μ that minimizes an image discrepancy function S . μ is a set of transformation parameters to be determined. $\bar{\mu}$ can be defined using the equation 3.3 where $f_T(x)$ is the input image over domain V_T , $f_R(x)$ is the reference image over domain V_R , and $g(x|\mu)$ is the deformation from V_T to V_R . Values for the transformation parameters μ are chosen iteratively to reduce the discrepancy between images.

$$\bar{\mu} = \mathit{arg\,min} S(f_R, f_T \circ g(\bullet|\mu)) \quad (3.3)$$

Figure 3.7 shows the flowchart of the algorithm used by the dipy library function to generate the affine transformation parameters for the input image f_R and reference image f_T . Once the transformation parameters are derived, they are applied to the input image to register the image.

Including the developed function of the initial transformation of the volume, the updated registration algorithm is named DIPY+. Algorithm 1 displays the updated

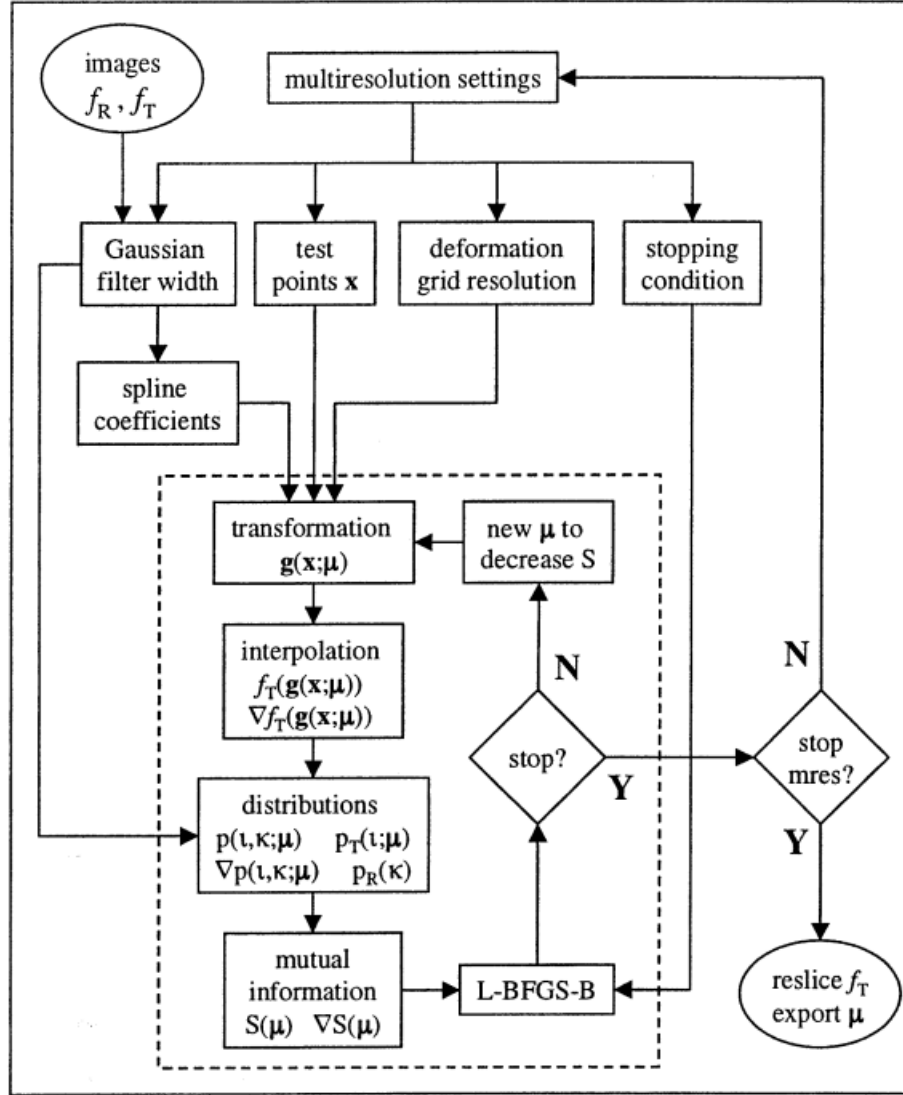


Figure 3.7: The registration algorithm proposed by Mattes et al. [2]. \mathbf{x} is any geometric location in the reference image. P , P_T , P_R , and ∇P are the joint, marginal test, marginal reference, and derivative of the joint distribution, respectively. L-BFGS-B is an optimization package that searches the parameter space of μ .

volume registration algorithm, including the functions of rotational difference estimation and user validation of volume registration. A few parameters must be set when using the library function to register image or volume data.

- **Reference_data:** The image or volume data, according to which the input data will be registered.
- **Input_data:** The image or volume to be registered.
- **Nbins:** A value indicating the number of bins to compute the intensity histogram. The default value of 32 is used in this case.
- **sampling_proportion:** A value between 0 and 1, indicating the sampling ratio. The 'None' value is used here to specify full sampling.
- **level_iters:** A sequence of numbers indicating the number of iterations per resolution. 3 resolution with [1000, 100, 100] iterations for 2D images and [10000, 1000, 100] iterations for 3D volumes are specified in this case.
- **Sigmas:** A sequence of floats indicating the custom smoothing parameters. The default value of [3, 1, 0] is used.
- **Factors:** A sequence of floats indicating the custom scale factors. The default value of [4, 2, 1] is used.

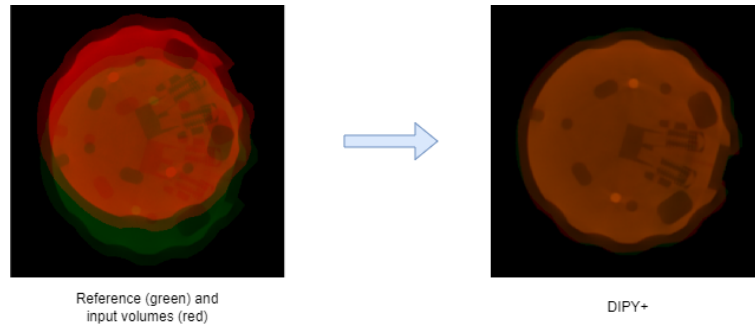


Figure 3.8: Examples of volume registration using our DIPY+ method.

An example of volume registration is shown in Figure 3.8, where the rotation angle was estimated beforehand, and an initial transformation is applied. The registered volumes overlapped perfectly, showing accurate registration.

3.1.2 CAD Volume File

Mechanical hardware tools can be designed on CAD software such as Solidworks, Autodesk, etc. Each tool component is designed separately and then assembled to form the complete tool CAD design. The designed tool can be exported as .STL files where each component is an individual STL file.

STL files are first converted to voxel data, with each voxel of the same component having the value of the corresponding class label. The size of the generated volume is set to the same value as the size of the actual CT volumes. Additionally, the module generates the information file that contains the class label and corresponding

component names.

The case study tool that was scanned for CT data has CAD files generated with Solidworks software. Figure 3.9 shows some examples of the components of the tool in both STL format and real images.

Although the CAD files of the experiment tool are available, it might not be the case always in reality. If the tool is designed by a company that does not want to

Algorithm 1: Rotation angle estimation and volume registration.

Input : P_1 : Reference volume path
 P_2 : Input volume path

Output: V_{ref} : Reference volume after background removal
 V_{input}^{reg} : Input volume data background removed and registered according to reference volume

◇ *Loading 3D volumes from P_1 and P_2 as 3D array*

$V_1, V_2 = \text{load_3D_volume}(P_1, P_2)$

◇ *Background subtraction*

$V_{ref} = \text{subtract_background}(V_1)$

$V_2^c = \text{subtract_background}(V_2)$

◇ *Register input volume*

• *Rotation angle estimation*

$V_{ref}^{resized} = \text{resize_into_smaller_dimension}(V_{ref})$

$V_2^{resized} = \text{resize_into_smaller_dimension}(V_2^c)$

$V_{ref}^{slices}, V_2^{slices} = \text{select_5_slices_from_random_indices}(V_{ref}^{resized}, V_2^{resized})$

$angles_{list} = []$ #list to store estimated angles for each slice

$ssim_{val} = 0$ #temporary variable to store calculated SSIM value

$angle_f = 0$ #temporary variable to store calculated angle value

```

for each corresponding slices  $V_{ref}^s, V_2^s$  do
  for angle between -180 to 180 do
     $V_2^r = \text{rotate\_image}(V_2^s, \text{angle})$ 
     $V_2^t = \text{translate\_to\_ref\_slice}(V_{ref}^s, V_2^r)$ 
     $ssim_{new} = \text{calculate\_ssim}(V_{ref}^s, V_2^t)$ 
    if  $ssim_{new} \geq ssim_{val}$  then
       $\text{angle}_f = \text{angle}$ 
       $ssim_{val} = ssim_{new}$ 
    end
     $\text{angles}_{list} \leftarrow \text{angle}_f$ 
     $ssim_{val} = 0$ 
  end
 $\text{angle}_{est} = \text{select\_optimum\_angle}(\text{angles}_{list})$ 
  • Register resized volume for user validation
   $V_2^{resized\_rot} = \text{rotate\_volume}(V_2^{resized}, \text{angle}_{est})$ 
   $V_2^{resized\_reg} = \text{register\_volumes\_DIPY}(V_{ref}^{resized}, V_2^{resized\_rot})$ 
  display_for_user_validation ( $V_2^{resized\_reg}, V_{ref}^{resized}$ )
  • Register full-scale volumes
if user approves the registration of resized volume then
  |  $V_2^{rot} = \text{rotate\_volume}(V_2^c, \text{angle}_{est})$ 
  |  $V_{input}^{reg} = \text{register\_volumes\_DIPY}(V_{ref}, V_2^{rot})$ 
else
  | re-estimate the rotation angle and do registration
end
return  $V_{ref}, V_{input}^{reg}$ 

```

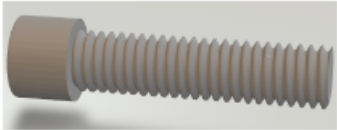



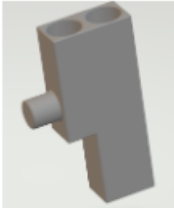



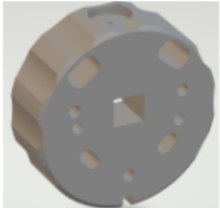

Component Name	STL file	Real tool
M6 X 30		
Inner Disk Clip		
Spring Holder		
Support Spring		
Support Part		

Figure 3.9: Examples of Individual components/parts of the tool.

share the CAD files, in such case the CAD volume file will not be generated, and any upcoming function depending on the CAD volume file will not be executed.

3.2 Image Processing-Based Defect Detection

Both image processing and deep learning-based defect detection system are proposed to be used. Deep learning-based defect detection would require a large amount of training data to train the model and then get prediction results. However initially, the system might not have enough data to train a deep-learning model. Image processing-based techniques do not require a large amount of initial CT data to start with.

The proposed defect detection method works on processed CT volumes using the CT volume processing module. The proposed detection approach is a reference image-based approach. Once a non-defect tool scan is saved to the system, the defect inspection can be carried out. Furthermore, missing components' names are estimated by referencing the annotated CAD volume file. In case the CAD volume file is not available, the proposed method would only be able to localize the defects and determine if the tool is defective or not. Figure 3.10 shows the general workflow of the algorithm.

The first step is the selection of the reference volume. The reference volume is

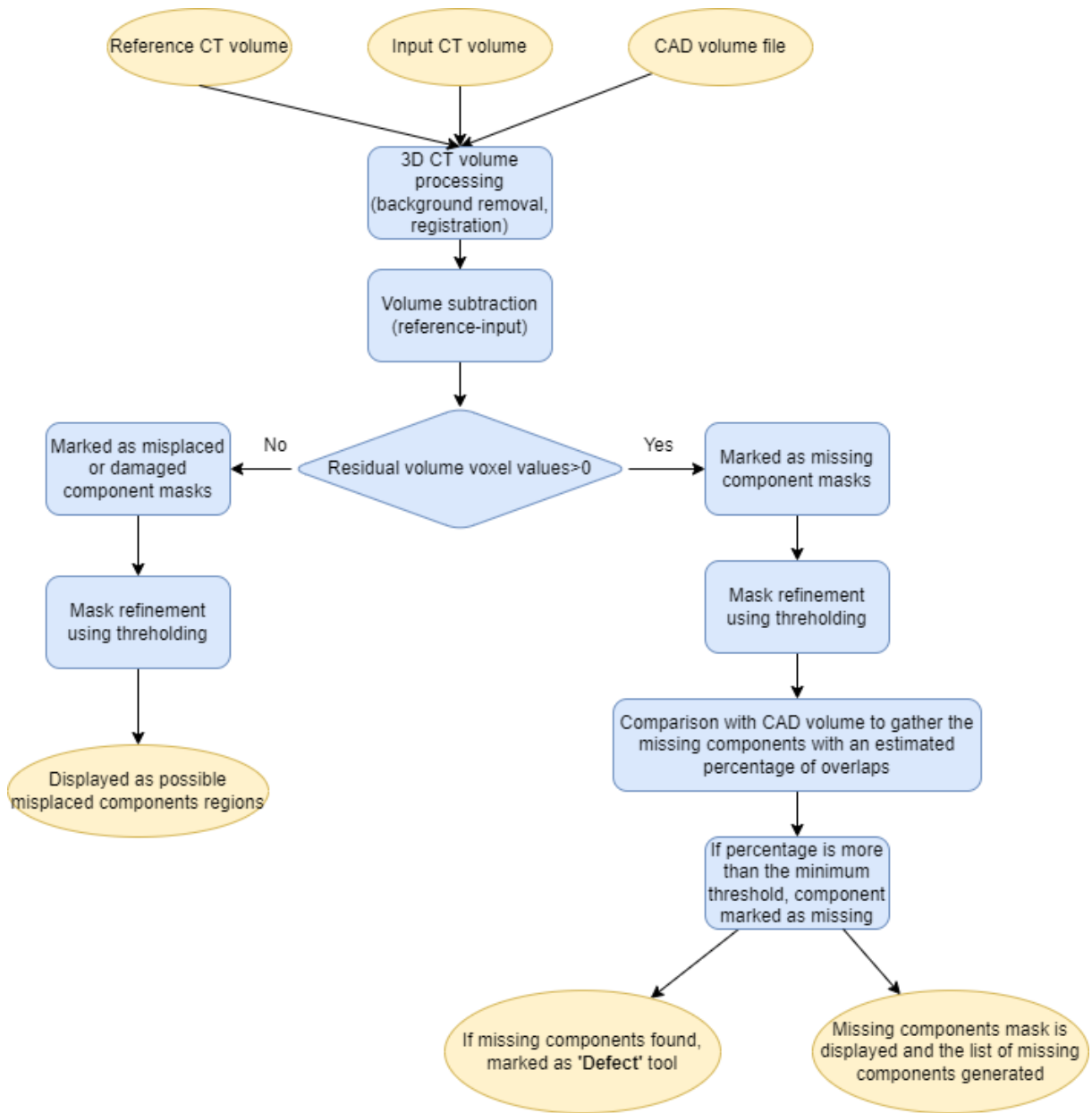


Figure 3.10: General workflow of the proposed defect detection approach.

the CT scan of the non-defect tool. The reference volume is registered according to the CAD volume file if annotated CAD volume data is available. Otherwise, only the background is subtracted from the reference image.

The input volume is first registered according to the reference volume. Then the input volume is subtracted from the reference volume. The characteristic of CT volume is that the hollow space in the CT volume has lower voxels values than the voxels that absorbed some X-ray due to the presence of obstruction between the emitter and detector. In residual volume generated by subtraction, voxel values higher than 0 mean that there was something present in the source volume whereas missing in the input volume. Moreover, values lower than 0 mean some new items were added to the input volume; this could be a misplacement of a component or a broken component. They can be localized as defective areas though the name of the broken or misplaced components can not be determined. The residual image is not a very clean image that shows the defects; there are background noise and surface level errors possibly generated due to cleaning and registration operation. Additionally, the voxel intensities vary from scan to scan, which adds more noisy voxels to the residual image. To remove these noisy data, thresholding was applied once again. Entropic thresholding [55] was applied to remove the background noises. This thresholding approach selects the thresholding value that creates maximum entropy.

After the thresholding, the binary masks of both the missing component volume and misplaced components volume are derived. Mask for the missing component is multiplied with the CAD volume file to only get the voxels present in both the missing components mask and the CAD volume file. Now, the annotated missing components mask is generated. Classes in the missing component mask are listed along with the corresponding pixel counts. Next, voxel counts for the same classes from annotated CAD volume are collected. The voxel counts from both sources are compared to find the percentage of voxel count in missing components mask. A threshold percentage value is set; if the calculated percentage exceeds the threshold value, the component corresponding to the class is identified as missing. If there are no missing components, the tool is marked as defect-free. The threshold value needs to be adjusted if there are new types of data or if the tool to be inspected is changed. This value is set manually based on the tool being inspected. In this thesis, the threshold was set to be 20% for experiments with actual CT data.

In the end, the estimated class (defect or defect-free), list of missing components, and the overlapping (input volume, missing components, possible misplacement) volumes are displayed.

Throughout the thesis, any 3D display operation is performed using napari [56]. Napari is a python-based multimodal interactive image viewer that can display 3D

volumes in different colorspace, and additionally lets users interact with the 3D volumes.

3.3 Semi-Automated CT Data Annotation

Although the proposed image processing-based defect detection technique would be able to identify defects in CT scans, this process would require time to process the input 3D CT data. However, if a trained deep-learning model is available for defect detection, the prediction using the model would require much less time. As already mentioned, initially there might not be enough data available for model training, so each scan given as input to the system must be annotated and saved to train the model. Until acceptable model performance is reached, the model should be trained with more and more data.

Deep learning-based defect detection or missing component identification system requires a segmentation model to be trained for defect detection or missing component identification. As this is a supervised training approach, mask images of the input CT volumes must be generated to train the model. There are a lot of different tools in an industrial setup, and they all have different components. Therefore, once the proper data are gathered, it will be helpful to have a semi-automated process of labelling the

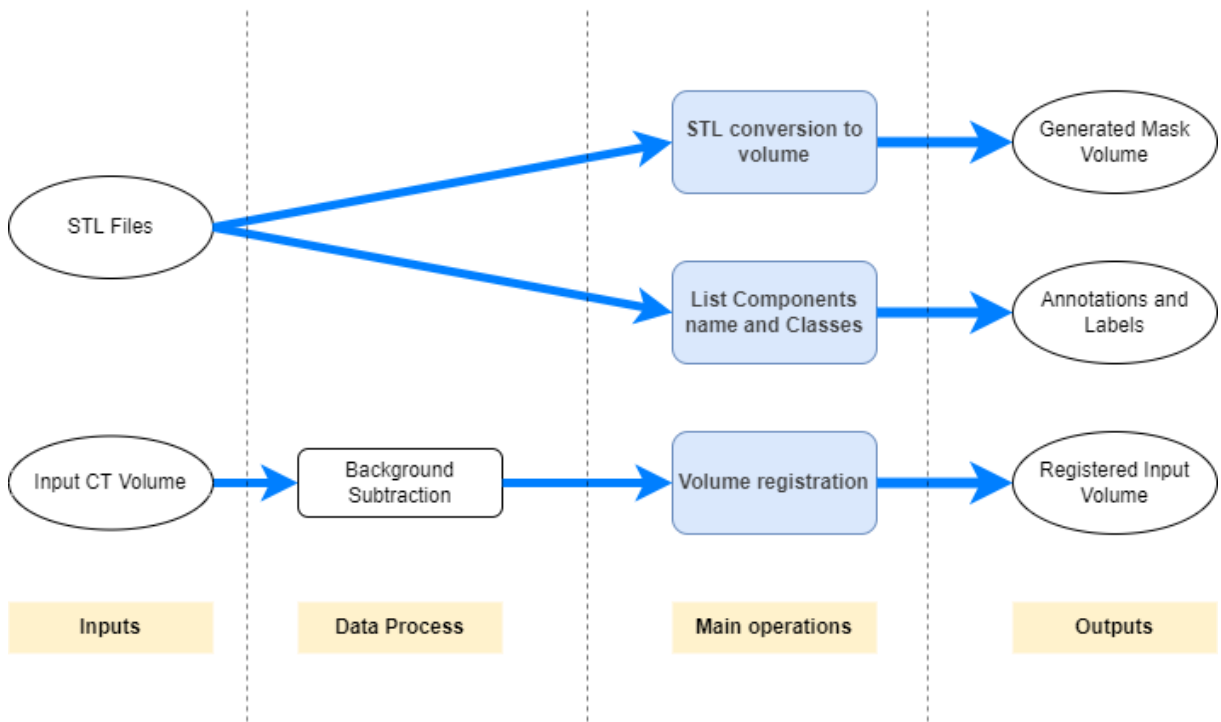


Figure 3.11: Workflow of the CT data annotation process.

data and generating masks so that the model can be trained for any new tools. Also, the tools contain many components; labelling them manually in CT volume would be time-consuming and challenging.

This module's inputs are the object's CT volume data and its CAD design files. The outputs include; processed CT data, corresponding annotated mask data, and the information file containing the mask's labelling information. Figure 3.11 shows the general workflow of the system. The proposed toolbox is a combination of three different modules. First, the reference labelled mask data from the CAD files are

generated. Then the CT data are processed (noise removal), and the mask data and CT data are aligned to get the labelled dataset. This is a 3-step process:

- **Step 1:** Generation of reference labelled mask data is the first step. The CAD volume file generated from the tool design files (STL files) is considered as the annotated mask data. Additionally, a file containing class information (text file) is generated during the generation of the CAD volume file.
- **Step 2:** As mentioned earlier, CT data comes with background noise data that is required to be removed before further operations. The background subtraction method explained previously is used to clean the input CT volume data.
- **Step 3:** The clean CT data is aligned to the generated mask volume in step 1. The function of aligning an image or volume according to a reference volume is called image or volume registration. The registration method explained previously is used to align the CT volume data to the mask data to get the training volume and mask data. The CT volume is registered according to the CAD volume. Then the CAD volume becomes the labelled mask of the CT volume, and they are saved as a pair of data for model training.

An example of generated mask data overlapping over an input CT volume is shown

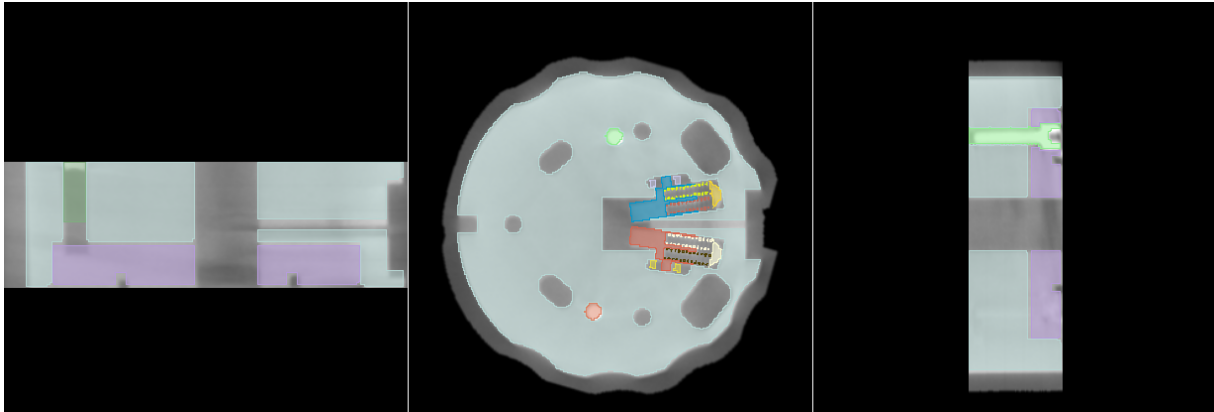


Figure 3.12: Examples of generated mask data overlapping over the input CT volume.

in Figure 3.12; the mask data has different voxel values for different components.

Hence, they are displayed in different colors.

3.4 Deep Learning-Based Segmentation Model

A semantic segmentation model is trained to evaluate the performance of the proposed semi-automated data annotation. The importance of the deep learning-based segmentation model is that if the model can segment the components of the tools, the missing components can be found easily. A properly trained model would perform faster defect detection than conventional image-based approaches. Some algorithms, such as volume registration, are computationally expensive and time-consuming.

U-net [53] was proposed for medical image segmentation. It is a Fully Convo-

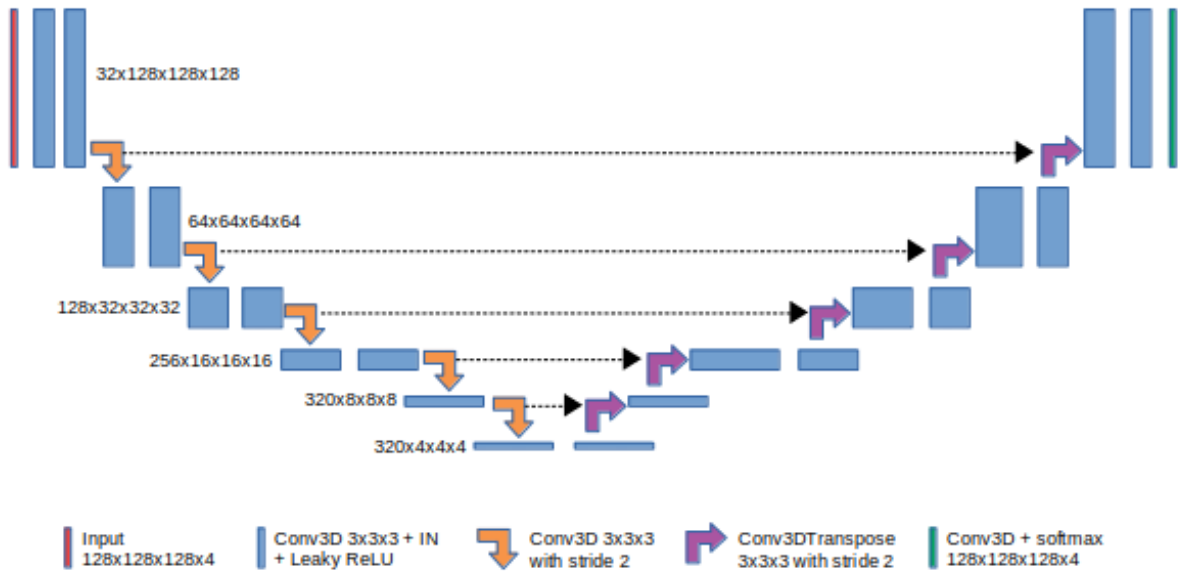


Figure 3.13: The construction of the 3D U-Net model and its different components [3].

lutional Network (FCN) based architecture. Since its publication, it has gained a reputation for generating excellent results while being trained with fewer images. Figure 3.13 shows the U-like structure of the U-net architecture; hence, the name U-net is given.

The algorithm can take in raw images as input and generate the segmented masks. The architecture consists of two parts; the first part is called the encoder. The encoder is used to capture the features from inputs. A Series of convolutional and max-pooling operations are carried out in the encoder part. The second part is called the decoder; it has the same structure as the decoder but is in reverse order. The decoder generates

the output image from the feature vectors derived in the encoder.

The training data was generated using the methodology explained in section 3.3. In addition, Resnet34 [57] was used as the backbone of the model, and sigmoid was used as the activation function.

After the model is trained, the input CT volumes are segmented using the model. The model generates an output of mask volume with voxel values corresponding to the component it belongs to. The segmented regions are further compared with annotated CAD volume. The number of voxels for large segmented regions is compared with the actual voxel count from CAD volumes to determine whether the segmented region is correct or false. If any classes are found missing in the segmented output mask, the tool can be declared defective due to missing components.

Chapter 4

Experimental Evaluation

This chapter presents a case study to understand the methodology of the thesis. In addition, the study will help to understand the dataset being experimented with and how the developed functions (numbered 1-4 in Figure 3.1) are applied to the dataset.

The CT volume data of a tool used for nuclear power plant inspection are collected for the experiments. Getting a large number of actual CT scan data scanning the industrial tool is challenging as CT scanning machines are not always accessible and it requires both expertise and time to operate a CT machine to scan an object to generate 3D data. Therefore, additional simulated CT data mimicking the actual CT data is generated. Simulated data and actual CT data were used for experiments and methodology validation.



Figure 4.1: Few components of the nuclear reactor tool that are assembled to construct the NPP tool.

4.1 Description of the Case Study Dataset

4.1.1 Real CT Data

The test tool is constructed by assembling multiple various-sized components. And then, the scan of the non-defect tool is taken. The non-defect tool is scanned multiple times to increase the dataset's size and to understand the background or artifacts that could be generated during scanning. Then components of tools are removed randomly and scanned again to generate the data of defective tools. Figure 4.1 shows some components of the constructed nuclear reactor tool. The tool was scanned multiple

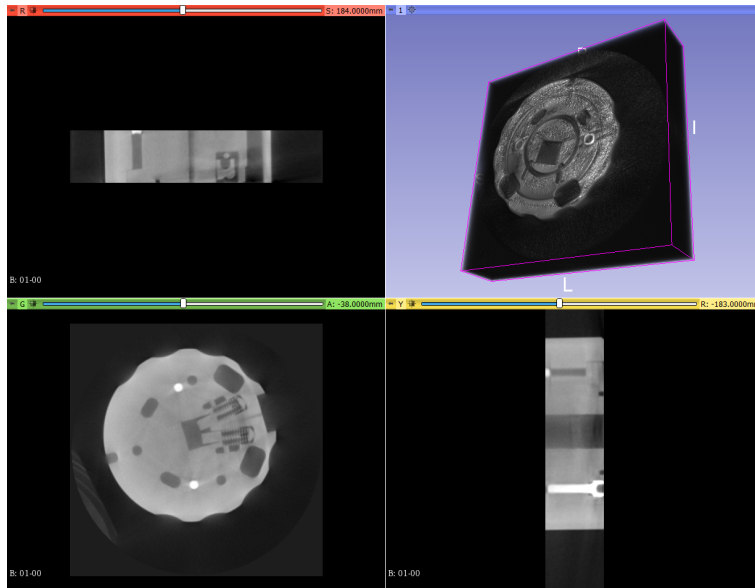


Figure 4.2: 3D view and cross-sectional images from 3 different directions of a CT scans.

times without any defects and with several missing parts. Figure 4.2 shows the 3D view and cross-sectional images from 3 different direction of a CT scans data received after reconstruction from 2D projections. The tool was scanned as segments, as the real tool is too large to be scanned at once. Our experiments were conducted on one segment of the tool. There is a total of 33 different scan data, of which 4 are scans of defect-free tools, and the rest have nine different kinds of faults.

As mentioned above, several components were removed manually during scanning to imitate the missing component scenario. Figure 4.3 shows how different defects were produced in each scan. Among 33 different scans, 4 are defect-free scans and

the other 29 scans have 9 different kinds of defects introduced to them. The introduced defects had variations among themselves with varying numbers of missing components, different-sized missing components, and the inclusion of new components. This allowed the system to be tested for different scenarios of defects.


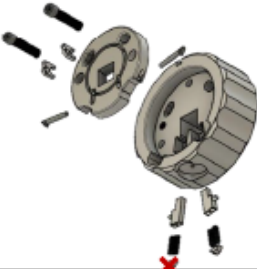
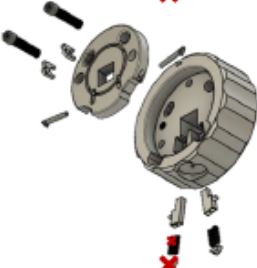

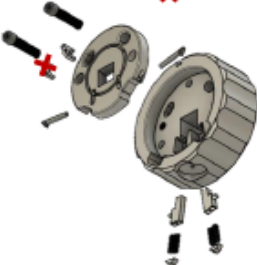
4.1.2 Annotated CAD Volume

To mimic the real CT data, the CAD design file of the tool is cropped out to select the corresponding segment of the tool that is scanned while generating real CT data. The cropped-out segment consists of 19 different components. When generating the volume data from CAD data (STL files), 20 different classes (19 components and background class) were labelled in the volume. The size or dimension of the volume is set to be the same as the real CT data.

4.1.3 Simulated CT Data

Analytical RT Inspection Simulation Tool (aRTist) [58] is a software tool that can mimic the CT scan environment and generate CT data from CAD volume data. The aRTist software is used to generate a set of 3D volume CT data of an inspection tool used in nuclear power plant. Figure 4.4 shows the interface of the aRTist software.

The simulated CT volume is replicated to mimic the real CT scenario, so the

Scan Name	Defects
<ul style="list-style-type: none"> • 01-00 • 01-01 • 01-02 • 01-03 	
<ul style="list-style-type: none"> • 02-00 • 02-01 • 02-02 • 02-03 	
<ul style="list-style-type: none"> • 03-00 • 03-01 • 03-02 • 03-03 	
<ul style="list-style-type: none"> • 04-00 • 04-01 • 04-02 • 04-03 	
<ul style="list-style-type: none"> • 05-00 • 05-01 • 05-02 • 05-03 	

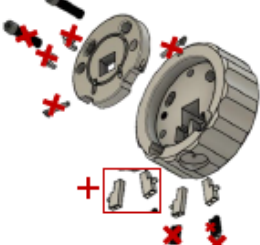
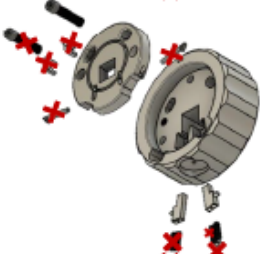
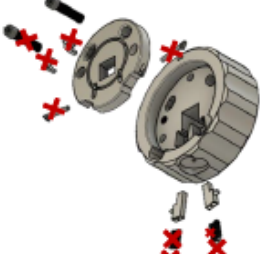
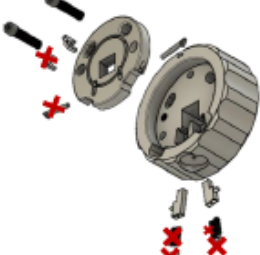
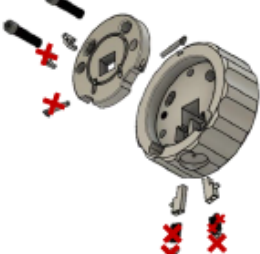
Scan Name	Defects
<ul style="list-style-type: none"> • 01 • 01-1 • 01-2 • 01-3 • 01-4 	
<ul style="list-style-type: none"> • 02 • 02-1 	
<ul style="list-style-type: none"> • 03 • 03-1 	
<ul style="list-style-type: none"> • 04 • 04-1 	
<ul style="list-style-type: none"> • 05 • 05-1 	

Figure 4.3: Defects introduced to different scans.

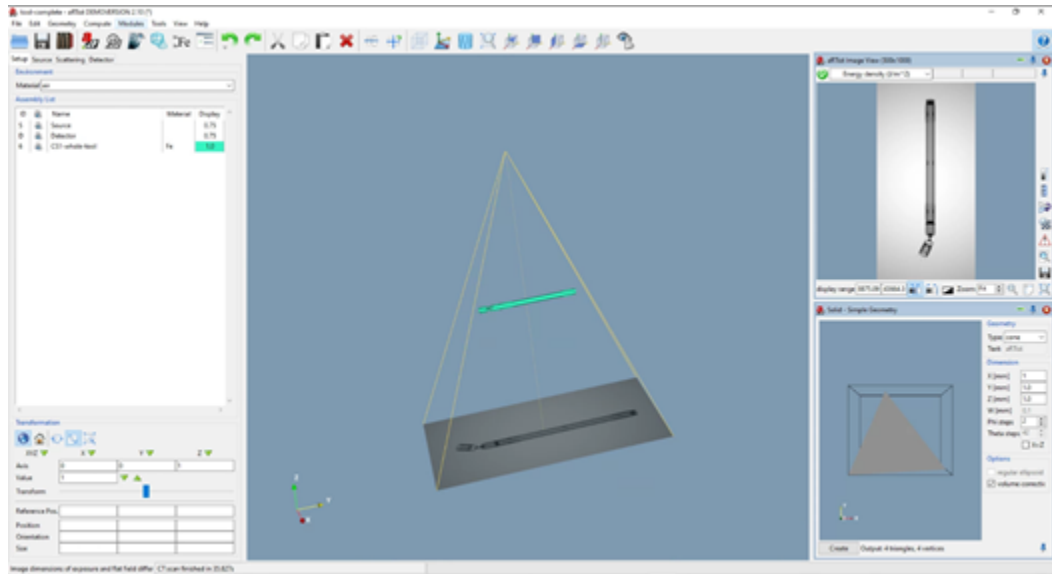


Figure 4.4: aRTist software interface.

volumes have background noises. Once the background noise is cleaned, the volume data is derived. The non-defect volume data is generated, and the missing components scenario is recreated according to Figure 4.3 to generate the defect volume data. Some different kinds of defects were also introduced to the simulated data. A total of 40 simulated CT volumes are generated, where 4 of them are non-defect, and the rest have 12 different kinds of defects.

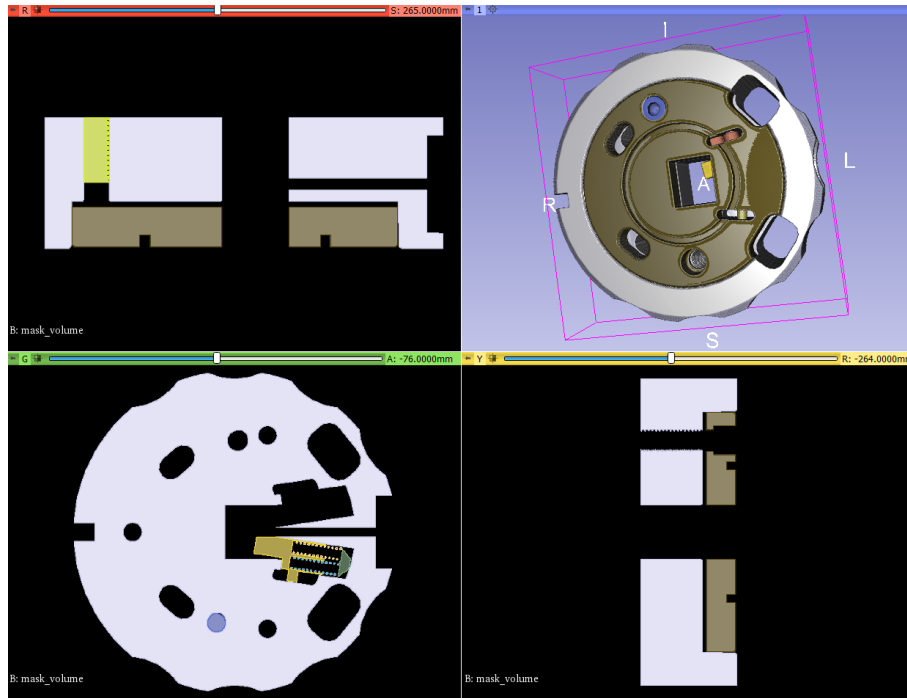


Figure 4.5: Generated labelled CAD volume file.

4.2 CT Data Processing

If the CAD design files of the tool being inspected are available, the labelled CAD volume file is generated using the CAD design files of the tool. In this case study, the CAD files are available, so the labelled CAD volume file is generated at the beginning. The process explained in section 3.1.2 is used to generate the labelled CAD volume file. Figure 4.5 shows the generated CAD volume file for the case study, each component is shown in different colors.

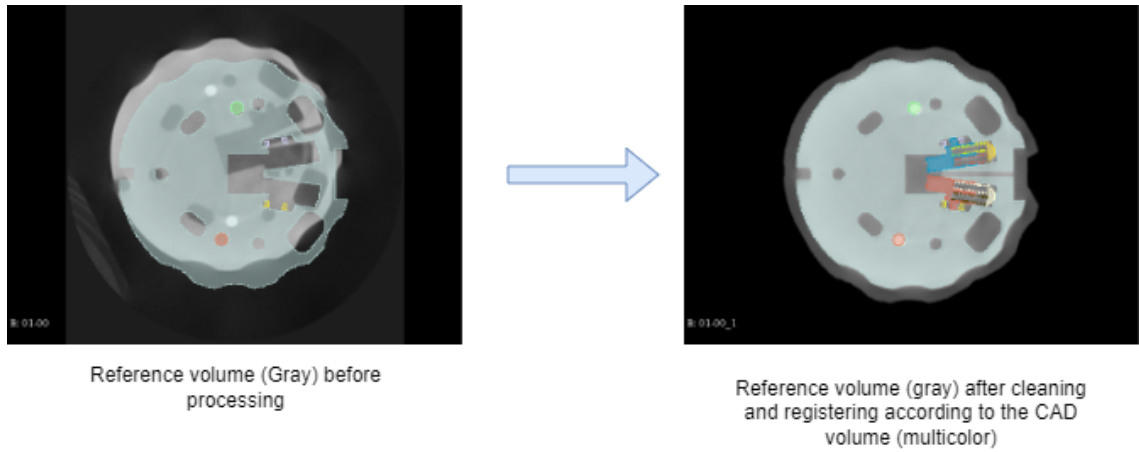


Figure 4.6: Reference volume processing.

The scanned CT data in this case study comes with background noise data around the whole tool image, as shown in Figure 4.2. First, the background noise is removed from the CT volumes. If the CT volume is known as the non-defect volume, it is considered the reference volume. As the annotated CAD volume data of the experiment tool is present, the reference volume is registered according to the CAD volume file. Figure 4.6 shows the reference image after background subtraction and volume registration. All other CT volumes are registered according to the reference volume data. A total of 33 real CT volumes data and 40 simulated CT volume data are processed using the process discussed in section 3.1.1.

4.3 Image Processing-Based Defect Detection

The developed image processing-based defect detection function (described in section 3.2) is applied to the processed 3D CT data to localize and identify the defects. Both sets of real and simulated processed CT data were used for the experiment.

Real CT data

One defect-free scan was selected as the reference volume. The background was subtracted and then registered according to the CAD volume. The rest of the volumes were considered experimental data. Then, one by one, each volume is inspected for defects.

One of the outputs generated by the defect detection module is shown in Figure 4.7. The first image there shows the actual defect that was introduced before scanning. One component was removed before scanning. The component was named ‘Support part (Spring stopper)’ during the tool’s design (in CAD design). The next image shows the defect (missing component) detected by the algorithm and localized. The input volume is green, and the missing component is red for easy visualization. The next image shows some possible misplacement or scratch-type marks in blue. Some marks around the screw are seen; this could be initialized from the assembly

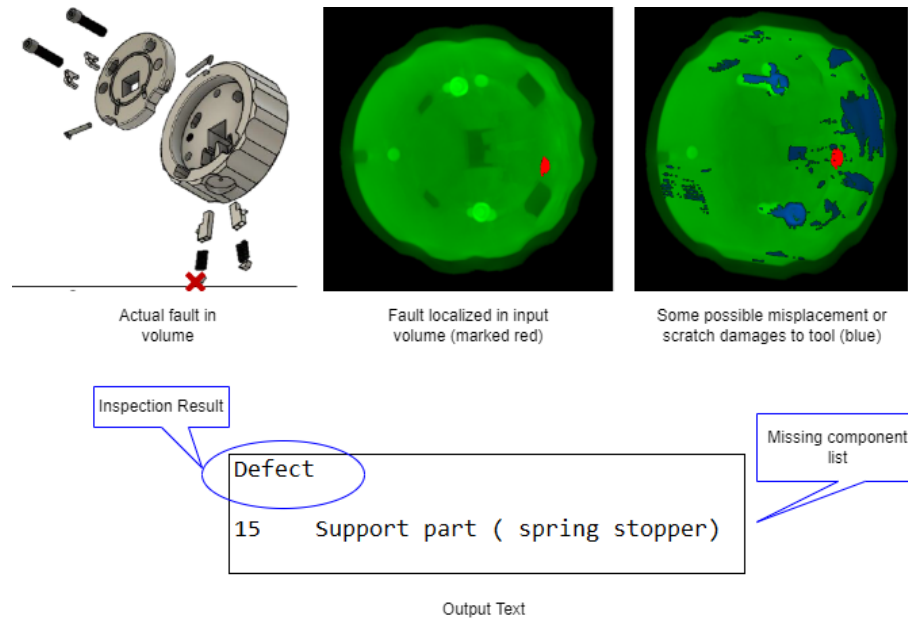


Figure 4.7: Example output from defect detection algorithm.

of the tool; the screw might be tightened differently or may be left a little tilted during insertion. These types of marks are not displayed as defects and do not affect the final inspection result. The final output text outputs the inspection result as ‘Defect’ or ‘Non-defect.’ The names of the missing components are listed with their corresponding class labels from the CAD volume file.

Simulated CT data

Similarly, for simulated CT data, a defect detection algorithm was applied. As simulation data are generated more controlled, the volume data have much less noise. So

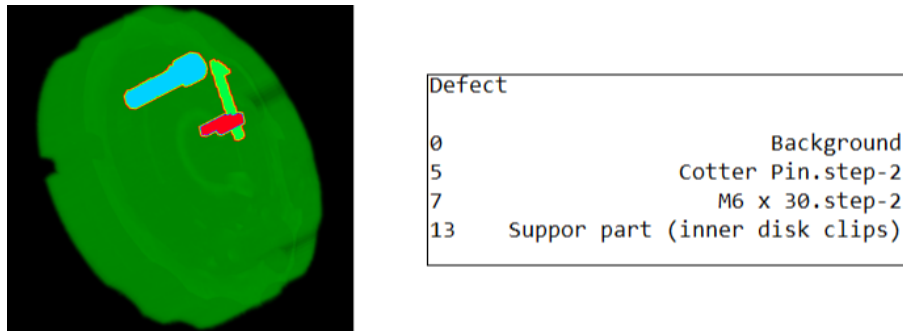


Figure 4.8: Example output from defect detection algorithm applied on simulated CT data.

the images are sharper than the real CT data. Figure 4.8 shows one output of defect detection using simulated CT volume. There were three missing components in this scan, and the algorithm successfully identified all three missing components. Components are displayed over the input volume (green), each showing different colours.

4.4 CT Volume Data Annotation

The segment of the tool that was selected for scanning consisted of 19 different components. The design CAD files with 19 different STL files were first used to generate the mask data. Then, the class label '0' was given to the background pixels, and other components were labelled as different numbers ranging from 1-19. The generated information file contains the class label and corresponding component names. However, some components are identical, meaning components are used multiple times in tool

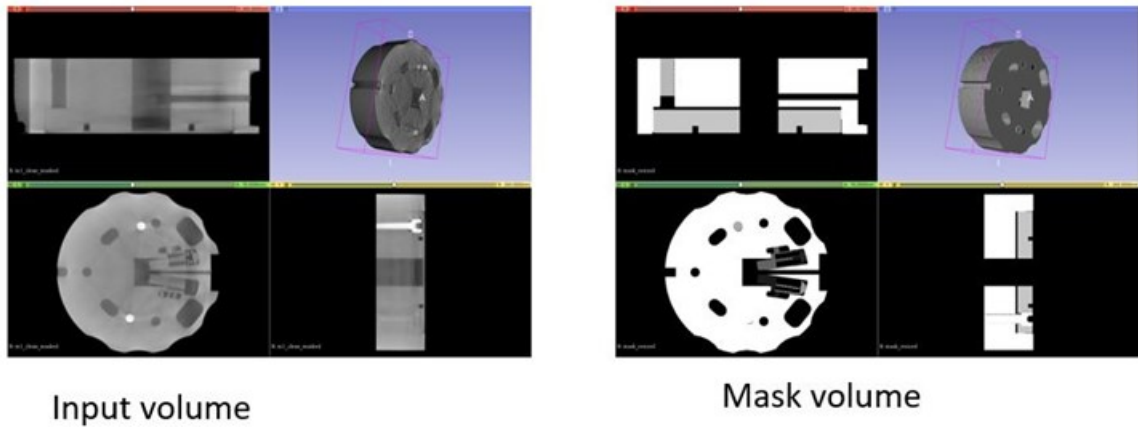


Figure 4.9: Example of CT data (left) and annotated mask (right).

```

class_num, component_name, voxel_count
0, Background, 7335789
1, M6 x 40.step-2, 5747
2, M6 x 40.step-5, 5824
3, M6 x 40.step-8, 5699
4, Cotter Pin.step-1, 3284
5, Cotter Pin.step-2, 3288
6, M6 x 30.step-1, 15379
7, M6 x 30.step-2, 15419
8, Spring for support part.step-1, 1120
9, Spring for support part.step-2, 1111
10, Spring for support part.step-3, 1143
11, Spring for support part.step-4, 1126
12, Suppor part (inner disk clips), 3432
13, Suppor part (inner disk clips), 3484
14, Support part ( spring stopper), 1592
15, Support part ( spring stopper), 1570
16, Support part (spring holder), 10893
17, Support part (spring holder), 10858
18, Support Part internal disk.step-1, 532374
19, Support part.step-1, 2468516

```

Figure 4.10: Example of generated class information file.

construction. Segmenting them as different classes with a trained model would be a problem of instance segmentation. The target here is to train a semantic segmentation model to segment the components present in the input scan and identify the missing scans. The tool scanned to generate the real CT data had some internal components already deformed, so those components were excluded from inclusion in segmentation classes. Finally, there were 7 different classes where some classes contained multiple numbers of components.

Using 33 real CT data, a dataset was created where the ground truth masks were generated using the process explained in section 3.3. The CT data were processed and passed for registration according to the mask volume to get the segmentation data pair. Figure 4.9 shows the registered CT data (input volume) and corresponding mask volume. Figure 4.10 shows the information file containing the details of each label.

Experiments were carried out using simulated CT data as the validation experiment of the semi-automated labelling algorithm. During simulation, the object is placed at different angles to mimic the real CT scenario where the objects are not always in the same plane. Input CT volume and corresponding ground truth masks for the 40 simulated data were also generated.

One more set of segmentation datasets was generated using simulated data, but

in this case, the simulated data were all generated without rotation. Placing them perfectly aligned with the annotated CAD volume file. So the data did not require any annotation processes. Some defective and defect-free scans were performed to generate 23 simulated data, which all are aligned with CAD volume files. So, ground truth masks for segmentation are generated by suppressing the missing components' voxels from the CAD volume file.

4.5 Segmentation Model Training Using Semi-Automated Annotated Datasets

As discussed in section 3.4, the U-net model architecture with ResNet34 backbone is used to develop the model for semantic segmentation. The model uses the Adam optimizer with a learning rate of 0.00001. As there is less data, the learning rate is kept smaller to keep the model training for longer. The batch size was selected as 2. The real CT scans were (256*96*256) voxels in dimension, and the simulated data were (192*96*192) voxels. Though they seem smaller, in 3D computation, using this dimension requires significant CPU and GPU memory. So the model had to be trained with a small batch size.

Volumes in each dataset were divided into train, test, and validation set in the

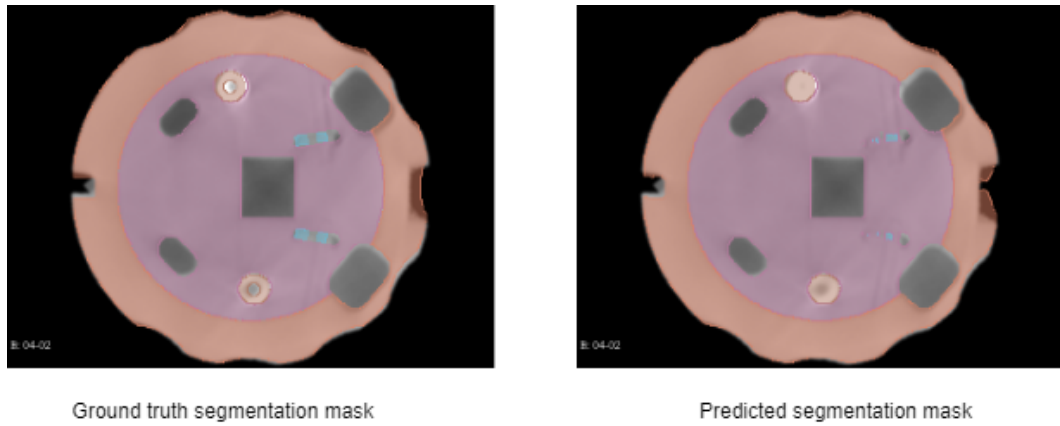


Figure 4.11: Example of segmentation prediction using trained model.

ratio of 80:10:10. During the training, both training and segmentation loss and IOU values were monitored. The training was kept running for 800 epochs. Figure 4.11 shows one example of segmentation prediction in one of the test volumes of the real CT dataset. Although a very small data size was used for training, the prediction shows promise in the future.

Chapter 5

Results and Analysis

The derived results are discussed in this chapter and compared with existing methods. In chapter 2, it was discussed that image processing-based defect detection algorithms struggle to handle volume registration when the volumes have large rotational or orientation differences, leading to lower defect detection accuracy. Existing registration algorithms generally struggles to handle large rotation difference in volume registration. It further discusses how difficulty in data annotation can limit the use of deep learning-based approaches in more practical use. An update to the existing image or volume registration algorithms was developed so that the case-study data could be registered accurately. The developed image-processing-based defect detection algorithm not only localizes but also identifies (by naming missing components) the defect

areas. And then finally, a semi-automation method for CT volume data annotation using CAD design files of the object is proposed.

5.1 Updates to Registration Algorithms: Rotational Angle Estimation

Figure 3.6 shows how few existing and publicly available image registration techniques failed to register the case study volume data according to the reference volume. The developed pre-registration function estimates the rotational difference angle and then rotates the volume data accordingly. The volumes are registered accurately using this angle estimation function and the DIPY Library function together. However, this update is not limited to DIPY; other algorithms that previously failed to register out CT volumes were able to register the volumes after applying the pre-registration function.

Figure 5.1 shows the improvements the developed rotation angle estimation function brought into existing registration algorithms. Visual observations already show the alignment differences between reference and input volumes. Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) are calculated for comparison. Both are used to perform image similarity checks. SSIM is calculated using the

equation 3.2. PSNR can be calculated using the following equation:

$$\text{PSNR} = 10 \log_{10} \frac{(2^d - 1)^2 WH}{\sum_{i=1}^W \sum_{j=1}^H (p[i, j] - p'[i, j])^2} \quad (5.1)$$

where d is the bit depth of pixel, W the image width, H the image height, and $p[i, j]$, $p'[i, j]$ is the i th-row j th-column pixel in the original and registered image respectively.

Figure 5.1 clearly shows SSIM and PSNR values increase once the volumes are registered after applying the angle estimation function. The improvements in the registration can also be seen in the overlapping volume images in Figure 5.1 for registration methods. On the other hand, applying the pre-registration function increases the time required for overall registration by almost 120 seconds. However, in defect detection most accurate result is the priority, and poor registration can lead to faulty results. Therefore, the increase in time required for volume registration due to the additional function can be ignored. Nevertheless, efficient parallelization of the registration operation for large CT volumes can be implemented to decrease the time required.

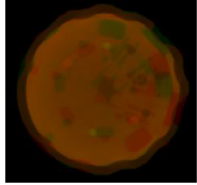
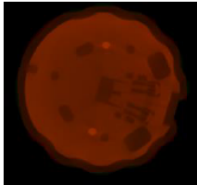
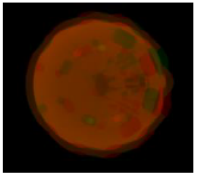
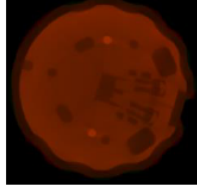
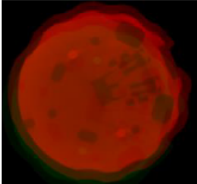
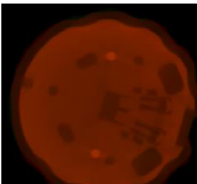
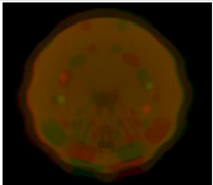
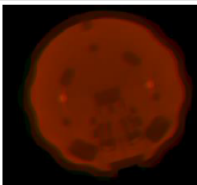
Registration Method	Without rotational angle estimation			With rotational angle estimation		
	overlapping volumes	SSIM	PSNR	overlapping volumes	SSIM	PSNR
Pystackreg		0.58	24.16		0.75	30.56
Slicer		0.83	22.21		0.94	29.38
DIPY		0.72	17.02		0.94	30.74
SimpleITK		0.83	52.88		0.84	52.89

Figure 5.1: Comparison between registration algorithms before and after applying our pre-registration updates.

5.2 Image Processing-Based Defect Detection

The developed defect detection algorithm was used to inspect the CT volumes. Although the main target is to determine if the tool scanned is defective or not, the system does both the localization of the defect and also identifies the missing components. The accuracy of the system in classifying defect and defect-free tools is calculated using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.2)$$

where True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) classifications are considered.

Among 33 real CT data, one defect-free scan was selected as the reference volume. Other 32 volumes were inspected. Considering the defect class as the positive class and the non-defect class as the negative class, there was only one false positive and one false negative prediction. This makes the accuracy of the proposed system to be 93.75%. Tests with simulated data generated an accuracy of 91%. Accuracy with simulated data is shown to be less than the real CT data. This is because in the simulated dataset several additional kinds of defects were introduced, and there is more variation in the data.

In industries like nuclear power plants, any false inspection result could lead to fatal accidents. Mainly when the tool is defective but identified as non-defect, false negative is the main concern. This means the recall must be higher. The recall value is calculated using the following formula which looks into the number of false negatives:

$$Recall = \frac{TP}{TP + FN} \quad (5.3)$$

The recall value for real CT data is 96.55% and with simulated data recall is 92.5%.

Defect detection results are summarized in Table 5.1.

Table 5.1: Summary result of the defective and non-defective classification of input scan.

Data type	Number of CT scans	Accuracy	Recall
Real CT data	33	93.75%	96.55%
Simulated CT data	40	91%	92.5%

The proposed method performs well in both simulated and actual CT data. Due to time constraints and limitations in generating CT data, the system is tested with a small amount of data. Once more CT data are acquired, the system would be tested more thoroughly. Unfortunately, the proposed method could not be tested with other datasets used by other researchers; also, due to the lack of publicly available codes

proposed in other papers, those algorithms could not be tested with this case study dataset. Some techniques [20] showed promising results with simulated data but they struggled when tested with real CT data.

The proposed method not only detects defects but also names the missing components using annotated CAD volume data which is a new approach to defect identification. Defects in the dataset were produced by removing random components from the tools before each scan. Identification of missing components by name makes this proposed method different from other approaches. Moreover, the defects are viewed in different colors for easier visual localization of the defect. Although the classification of defective or defect-free yields higher accuracy. Identification of missing components by name requires more research as it struggles when the components are misplaced or deviate from their original positions.

5.3 Semi-Automated CT Volume Annotation

The purpose of automating the annotation system is to annotate large datasets quickly and easily. In addition, the segmentation model can also be trained to identify the missing components in CT volumes. Two sets of semantic segmentation datasets were developed using the annotation process described in chapter 3. In addition, another

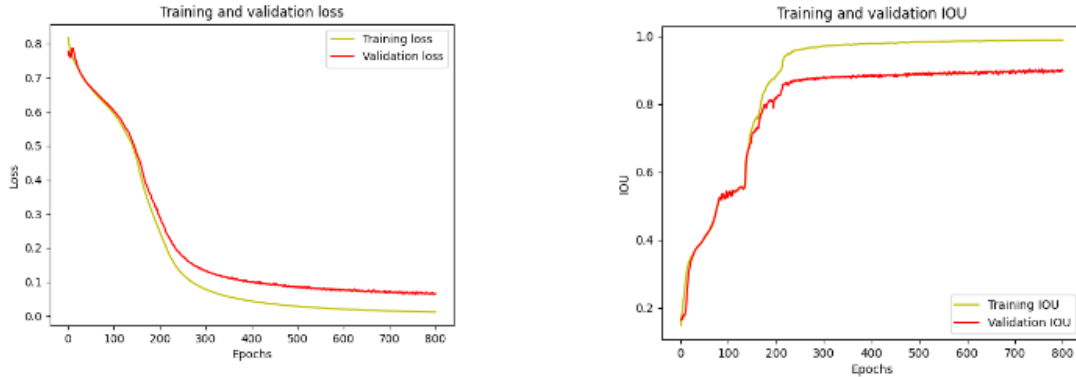


Figure 5.2: Training and validation loss and Training and validation IOU curve of the trained model using real CT data.

set of annotated CT data was simulated, those data did not require any manual annotation or annotation by the proposed method.

The annotated data were used to train a simple segmentation model developed using U-net model architecture just to show how the annotated data can be used and to validate if the annotation function could generate data that can be used for model training. The real CT data set consisted of 33 volumes in total. The model was trained for 800 epochs. Figure 5.2 shows the trained model achieved a final training loss of approximately 0.02 and a final validation loss of around 0.1. The model also had a final training IOU of around 0.90 and a validation IOU of more than 0.80.

The model was also tested using test data; test data were completely new data for the model. The predictions resulted in an average IOU value of 0.70. This clearly

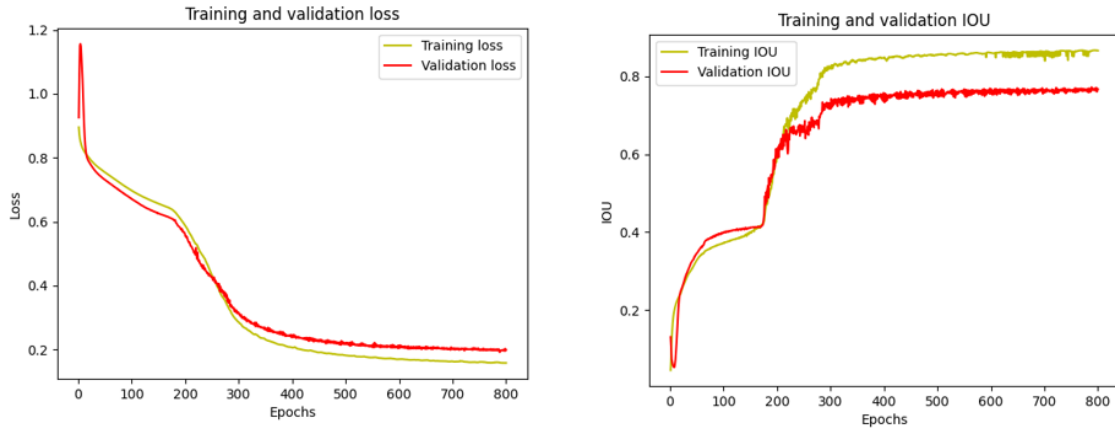


Figure 5.3: Training and validation loss and Training and validation IOU curve of the trained model using simulated data, annotated by the proposed annotation method.

indicates that the model is learning from the annotated data. Figure 4.11 shows one example of the predicted output.

The other simulated CT datasets were also used for model training and showed similar learning progress. Models trained with simulated CT data showed a similar learning curve. The simulated dataset annotated using the proposed method shows a validation loss of less than 0.2 and a validation IOU of almost 0.80. Also, the average IOU with test data is 0.64. Figure 5.3 shows the training curves of the model trained.

The simulated dataset that did not require labelling displays a validation loss of approximately 0.15 and a validation IOU of almost 0.80. Also, the average IOU with test data is 0.64. Figure 5.4 shows the training curves of the model trained.

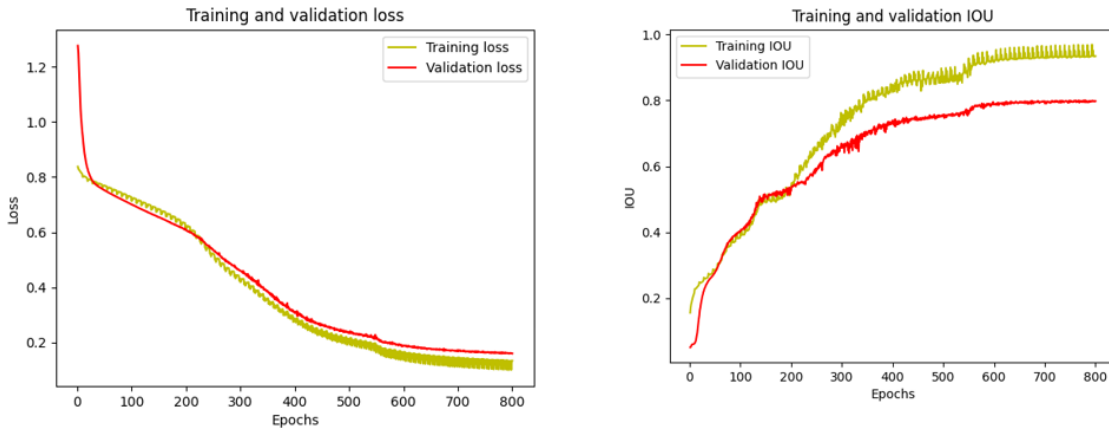


Figure 5.4: Training and validation loss and Training and validation IOU curve of the trained model using simulated data that did not require annotation.

The segmentation model trained with annotated data showed proof of learning even with a small number of CT volume data. When comparing the results of the different datasets (actual and simulated CT data), they all displayed similar learning trends. Smaller components were challenging to segment using the trained model, but this can be overcome by using a model trained with a large dataset.

The above discussions show that the proposed methodology resulted in the existing registration algorithms generating more accurate registration results. The defect detection system could identify defective and non-defective tools with high accuracy and recall in both actual and simulated volumes. Although the system is tested with a small number of data, it still shows promising results. Moreover, the proposed annotation system could generate data that can be used to train deep learning models.

Chapter 6

Conclusion

Compared to conventional destructive testing, the use of CT technology offers more reliable, quicker, and in some cases, more straightforward solutions. There are increasing tendencies to apply CT technology in NDT. The thesis proposes a novel NDT technique based on image-processing methods, a processing function for CT data that includes background noise removal, and an improved registration function that can register 3D volumes with a large rotational difference. The proposed defect detection system identifies defective and non-defective CT volumes and can identify the missing components by their names. Finally, a data annotation method is proposed to annotate CT data for segmentation model training. The proposed data annotation will encourage more proposals of deep learning-based techniques to be

used with CT data. The proposed annotator can also participate in building useful annotated data for further deep learning-based applications.

6.1 Limitations

Working with CT data is always challenging, and the challenge increases more with 3D CT data. 3D data operations are computationally very expensive, so a high-performance machine is required to work with 3D CT data. As the case study tool was a long tool, it had to be scanned segment by segment. Although the works of the thesis were based on a specific segment, handling multiple segments of CT data is still challenging.

In addition, though the modification to existing volume registration functions could handle rotational differences while registering the input 3D data, it still can not handle the difference when the input volume is rotated in a different plane or placed in the scanner with a completely different orientation. In this case, the input volume needs to be oriented manually so that the reference and input volume are in the same orientation and the corresponding slices from each volume match with each other. Although shearing transformation is applied to the input volume, this transformation can handle very little tilting of the scanned volume.

Image processing-based defect detection approaches struggle where there is permanent deformation in some objects. In addition, some hardware tools might have mobile components that can be oriented differently or misplaced in different scans. These factors were not addressed adequately in the proposed methodology. The proposed methodology is heavily dependent on CAD design files, which means the hardware must be assembled very carefully so that they do not differ from the CAD design.

6.2 Future Work

The research will be extended in future to include analysis of more datasets or real scans and case studies, which will provide more validation and tuning of the proposed approach. The analysis of more datasets will be used to fine-tune the proposed automated solution and framework for real industrial applications. The proposed framework can also be updated to classify the defects into different kinds such as missing, broken, misplaced, scratched, etc. Also, the defect detection algorithm needs some parameters to be manually set for now; in the future, work can be done to make the parameter selection process automatic.

Industries like NPP will not authorize the use of any software or hardware products

without proper evaluation and licensing. Once every evaluation step is passed, new products can be integrated into NPP. These evaluation criteria and licences required will be investigated in future. After studying the conditions for acceptance, if any modifications are required to the developed system, these modifications can be carried out.

To handle large objects scanned in smaller segments, further research can be expected to merge the segmented CT scans to generate the whole object's 3D CT volume. The challenging factor in this task is to identify the overlapping areas in each segmented scan and merge them accordingly. Functions such as 3D volume registration in large 3D CT volumes is time consuming, so more research on parallelizing the tasks and better utilization of GPU is required to ensure faster execution.

Finally, once enough CT data is acquired, deep learning-based defect detection approaches can be investigated further. Trained deep learning models for defect classification or component segmentation can be used for defect detection. Both image processing-based and deep learning-based approaches can be used to develop a hybrid system for more accurate defect detection and identification.

Once enough CT data is acquired, deep learning-based defect detection approaches can be investigated further. Classification, segmentation, and clustering-based deep learning and machine learning approaches can be investigated further for defect de-

tection function development. Both image processing-based and deep learning-based approaches can be used to develop a hybrid system for more accurate defect detection and identification.

References

- [1] Markus Hadwiger, Laura Fritz, Christof Rezk Salama, Thomas Höllt, Georg Geier, and Thomas Pabel. Interactive volume exploration for feature detection and quantification in industrial ct data. *IEEE transactions on visualization and computer graphics*, 14:1507–14, 11 2008.
- [2] David Mattes, David Haynor, Hubert Vesselle, Thomas Lewellen, and William Eubank. Pet-ct image registration in the chest using free-form deformations. *Medical Imaging, IEEE Transactions on*, 22:120 – 128, 02 2003.
- [3] Nvidia NGC. 3d-unet medical image segmentation for tensorflow. 2022. https://catalog.ngc.nvidia.com/orgs/nvidia/resources/unet3d_medical_for_tensorflow (accessed Sept 28, 2022).
- [4] World Nuclear Association. Nuclear power in the world today. 2022. <https://www.world-nuclear.org/information-library/current-and-future->

- generation/nuclear-power-in-the-world-today.aspx (accessed Sept 28, 2022).
- [5] International Atomic Energy Agency. Operation and maintenance of nuclear power plants. 2016. <https://www.iaea.org/topics/operation-and-maintenance> (accessed Sept 28, 2022).
- [6] Flyability. Inspection of a nuclear power plant. 2017. <https://www.flyability.com/casestudies/inspection-of-a-nuclear-power-plant> (accessed Sept 28, 2022).
- [7] Pekka Pyy. An analysis of maintenance failures at a nuclear power plant. *Reliability Engineering and System Safety*, 72(3):293–302, 2001.
- [8] Euclid Seeram. *Computed Tomography-E-Book: Physical Principles, Clinical Applications, and Quality Control*. Elsevier Health Sciences, 2015.
- [9] Herminso Villarraga-Gómez. Seeing is believing: X-ray computed tomography for quality control. *Quality Magazine*, 55(6):20–23, 2016.
- [10] Elluru Venkatesh and Snehal Venkatesh Elluru. Cone beam computed tomography: basics and applications in dentistry. *Journal of istanbul University faculty of Dentistry*, 51(3 Suppl 1):S102, 2017.

- [11] MEHAJ Korobkin, EA White, HY Kressel, AA Moss, and JP Montagne. Computed tomography in the diagnosis of adrenal disease. *American Journal of Roentgenology*, 132(2):231–238, 1979.
- [12] Felix H. Kim, Adam Pintar, Anne-Françoise Obaton, Jason Fox, Jared Tarr, and Alkan Donmez. Merging experiments and computer simulations in x-ray computed tomography probability of detection analysis of additive manufacturing flaws. *NDT E International*, 119:102416, 2021.
- [13] F. Nachtrab, S. Weis, P. Keßling, F. Sukowski, U. Haßler, T. Fuchs, N. Uhlmann, and R. Hanke. Quantitative material analysis by dual-energy computed tomography for industrial ndt applications. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 633:S159–S162, 2011. 11th International Workshop on Radiation Imaging Detectors (IWORID).
- [14] Istvan Szabo, Jiangtao Sun, Guojin Feng, Jamil Kanfoud, Tat-Hean Gan, and Cem Selcuk. Automated defect recognition as a critical element of a three dimensional x-ray computed tomography imaging-based smart non-destructive testing technique in additive manufacturing of near net-shape parts. *Applied Sciences*, 7(11), 2017.

- [15] Nicholas Hashem, Mitchell Pryor, Derek Haas, and James Hunter. Design of a computed tomography automation architecture. *Applied Sciences*, 11(6), 2021.
- [16] Michael Reiter, Christian Gusenbauer, Rainer Huemer, and Johann Kastner. At-line x-ray computed tomography of serial parts optimized by numerical simulations. In *Proceedings of Symposium on Digital Industrial Radiology and Computed Tomography (DIR2019), Fürth*, 2019. International Symposium on Digital Industrial Radiology and Computed Tomography (DIR2019) ; Conference date: 02-07-2019 Through 04-07-2019.
- [17] L. De Chiffre, S. Carmignato, J.-P. Kruth, R. Schmitt, and A. Weckenmann. Industrial applications of computed tomography. *CIRP Annals*, 63(2):655–677, 2014.
- [18] Samet Akcay and Toby P. Breckon. Towards automatic threat detection: A survey of advances of deep learning within x-ray security imaging. *CoRR*, abs/2001.01293, 2020.
- [19] Lili Jiang, Yongxiong Wang, Zhenhui Tang, Yinlong Miao, and Shuyi Chen. Casting defect detection in x-ray images using convolutional neural networks and attention-guided data augmentation. *Measurement*, page 108736, 2020.

- [20] Patrick Fuchs, Thorben Kröger, and Christoph S. Garbe. Defect detection in ct scans of cast aluminum parts: A machine vision perspective. *Neurocomputing (Amsterdam)*, 453:85–96, 2021.
- [21] Santanu Ghorai, Anirban Mukherjee, M. Gangadaran, and Pranab K. Dutta. Automatic defect detection on hot-rolled flat steel products. *IEEE Transactions on Instrumentation and Measurement*, 62(3):612–621, 2013.
- [22] Erol Sarigul, A.Lynn Abbott, and Daniel L. Schmoltdt. Rule-driven defect detection in ct images of hardwood logs. *Computers and electronics in agriculture*, 41(1):101–119, 2003.
- [23] Jingdong Chen, Jacob Benesty, Yiteng Huang, and Simon Doclo. New insights into the noise reduction wiener filter. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14:1218 – 1234, 08 2006.
- [24] Inc. The MathWorks. Intensity-based automatic image registration. 2022. <https://www.mathworks.com/help/images/intensity-based-automatic-image-registration.html> (accessed Sept 28, 2022).
- [25] Robert M. Haralick, Stanley R. Sternberg, and Xinhua Zhuang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):532–550, 1987.

- [26] Ning He, Lulu Zhang, and Ke Lu. Aluminum ct image defect detection based on segmentation and feature extraction. In *Design, User Experience, and Usability. User Experience Design for Diverse Interaction Platforms and Environments*, Lecture Notes in Computer Science, pages 446–454, Cham. Springer International Publishing.
- [27] Xiayun Zhao and David W Rosen. Simulation study on evolutionary cycle to cycle time control of exposure controlled projection lithography. *Rapid Prototyping Journal*, 2016.
- [28] Guoying Dong, Julien Marleau-Finley, and Yaoyao Fiona Zhao. Investigation of electrochemical post-processing procedure for ti-6al-4v lattice structure manufactured by direct metal laser sintering (dmls). *The International Journal of Advanced Manufacturing Technology*, 104(9):3401–3417, 2019.
- [29] Fei Zhao, Paulo R. S. Mendonça, Jie Yu, and Robert Kaucic. Learning-based automatic defect recognition with computed tomographic imaging. In *2013 IEEE International Conference on Image Processing*, pages 2762–2766, 2013.
- [30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing*

- and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.
- [31] Daniel L Schmoltdt, Jing He, and A Lynn Abbott. Automated labeling of log features in ct imagery of multiple hardwood species. *Wood and Fiber Science*, pages 287–300, 2000.
- [32] Long Cheng, Ping Gong, Guanghui Qiu, Jing Wang, and Ziyuan Liu. Small defect detection in industrial x-ray using convolutional neural network. In *Pattern Recognition and Computer Vision*, Lecture Notes in Computer Science, pages 366–377. Springer International Publishing, Cham, 2019.
- [33] Francisco P.M. Oliveira and João Manuel R.S. Tavares. Medical image registration: a review. *Computer Methods in Biomechanics and Biomedical Engineering*, 17(2):73–93, 2014. PMID: 22435355.
- [34] Shun Miao, Z. Jane Wang, and Rui Liao. A cnn regression approach for real-time 2d/3d registration. *IEEE transactions on medical imaging*, 35(5):1352–1363, 2016.
- [35] Shun Miao, Z. Jane Wang, Yefeng Zheng, and Rui Liao. Real-time 2d/3d registration via cnn regression. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 1430–1434. IEEE, 2016.

- [36] Seyed Sadegh Mohseni Salehi, Shadab Khan, Deniz Erdogmus, and Ali Gholipour. Real-time deep pose estimation with geodesic loss for image-to-template rigid registration. *IEEE transactions on medical imaging*, 38(2):470–481, 2019.
- [37] Yuanyuan Sun, Adriaan Moelker, Wiro J. Niessen, and Theo van Walsum. Towards robust ct-ultrasound registration using deep learning methods. In *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, Lecture Notes in Computer Science, pages 43–51. Springer International Publishing, Cham, 2018.
- [38] Yabo Fu, Yang Lei, Tonghe Wang, Walter J Curran, Tian Liu, and Xiaofeng Yang. Deep learning in medical image registration: a review. *Physics in medicine biology*, 65(20):20TR01–20TR01, 2020.
- [39] Sasu Mäkinen, Henrik Skogström, Eero Laaksonen, and Tommi Mikkonen. Who needs mlops: What data scientists seek to accomplish and how can mlops help? *arXiv.org*, 2021.
- [40] Zhiyong Zhang, Xiaolei Yin, and Zhiyuan Yan. Rapid data annotation for sand-like granular instance segmentation using mask-rcnn. *Automation in construction*, 133:103994–, 2022.

- [41] Christoffer Bøgelund Rasmussen, Kristian Kirk, and Thomas B Moeslund. The challenge of data annotation in deep learning—a case study on whole plant corn silage. *Sensors (Basel, Switzerland)*, 22(4):1596–, 2022.
- [42] Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific reports*, 9(1):16884–9, 2019.
- [43] Douwe van der Wal, Iny Jhun, Israa Lakloul, Jeff Nirschl, Lara Richer, Rebecca Rojansky, Talent Theparee, Joshua Wheeler, Jörg Sander, Felix Feng, Osama Mohamad, Silvio Savarese, Richard Socher, and Andre Esteva. Biological data annotation via a human-augmenting ai-based labeling system. *NPJ digital medicine*, 4(1):145–145, 2021.
- [44] Dim P Papadopoulos, Jasper R. R Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4940–4949. IEEE, 2017.
- [45] Hossam A. Gabbar, Abderrazak Chahid, Md. Jamiul Alam Khan, Oluwabukola Grace Adegboro, and Matthew Immanuel Samson. Ctims: Automated defect detection framework using computed tomography. *Applied Sciences*, 12(4), 2022.

- [46] Vikas Chougule, Arati Mulay, and B. Ahuja. Conversions of ct scan images into 3d point cloud data for the development of 3d solid model using b-rep scheme. 12 2013.
- [47] Vikas Chougule, Arati Mulay, and B. Ahuja. Three dimensional point cloud generations from ct scan images for bio-cad modeling. 10 2013.
- [48] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical cone-beam algorithm. *J. Opt. Soc. Am. A*, 1(6):612–619, Jun 1984.
- [49] Thomas S. Huang, G Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27:13–18, 1979.
- [50] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.
- [51] P. Thevenaz, U.E. Ruttimann, and M. Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Transactions on Image Processing*, 7(1):27–41, 1998.

- [52] Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, John Buatti, Stephen Aylward, James V Miller, Steve Pieper, and Ron Kikinis. 3D slicer as an image computing platform for the quantitative imaging network. *Magn. Reson. Imaging*, 30(9):1323–1341, November 2012.
- [53] Eleftherios Garyfallidis, Matthew Brett, Bagrat Amirbekian, Ariel Rokem, Stefan van der Walt, Maxime Descoteaux, and Ian Nimmo-Smith. Dipy, a library for the analysis of diffusion mri data. *Frontiers in Neuroinformatics*, 8, 02 2014.
- [54] Richard Beare, Bradley Lowekamp, and Ziv Yaniv. Image segmentation, registration and characterization in r with simpleitk. *Journal of Statistical Software*, 86(8):1–35, 2018.
- [55] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3):273–285, 1985.
- [56] Nicholas Sofroniew, Talley Lambert, Kira Evans, Juan Nunez-Iglesias, Grzegorz Bokota, Philip Winston, Gonzalo Peña-Castellanos, Kevin Yamauchi, Matthias Bussonnier, Draga Doncila Pop, Ahmet Can Solak, Ziyang Liu, Pam Wad-

hwa, Alister Burt, Genevieve Buckley, Andrew Sweet, Lukasz Migas, Volker Hilsenstein, Lorenzo Gaifas, Jordão Bragantini, Jaime Rodríguez-Guerra, Hector Muñoz, Jeremy Freeman, Peter Boone, Alan Lowe, Christoph Gohlke, Loic Royer, Andrea PIERRÉ, Hagai Har-Gil, and Abigail McGovern. napari: a multi-dimensional image viewer for Python, May 2022.

- [57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [58] Carsten Bellon and Gerd-Ruediger Jaenisch. artist-analytical rt inspection simulation tool. 01 2007.