Imperial College London
Department of Computing

3D Shape Instantiation for Intra-operative Navigation from a Single 2D Projection

Xiao-Yun Zhou

October 2019

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London
and the Diploma of Imperial College London

Abstract

Unlike traditional open surgery where surgeons can see the operation area clearly, in robot-assisted Minimally Invasive Surgery (MIS), a surgeon's view of the region of interest is usually limited. Currently, 2D images from fluoroscopy, Magnetic Resonance Imaging (MRI), endoscopy or ultrasound are used for intra-operative guidance as real-time 3D volumetric acquisition is not always possible due to the acquisition speed or exposure constraints. 3D reconstruction, however, is key to navigation in complex *in vivo* geometries and can help resolve this issue. Novel 3D shape instantiation schemes are developed in this thesis, which can reconstruct the high-resolution 3D shape of a target from limited 2D views, especially a single 2D projection or slice. To achieve a complete and automatic 3D shape instantiation pipeline, segmentation schemes based on deep learning are also investigated. These include normalization schemes for training U-Nets and network architecture design of Atrous Convolutional Neural Networks (ACNNs).

For U-Net normalization, four popular normalization methods are reviewed, then Instance-Layer Normalization (ILN) is proposed. It uses a sigmoid function to linearly weight the feature map after instance normalization and layer normalization, and cascades group normalization after the weighted feature map. Detailed validation results potentially demonstrate the practical advantages of the proposed ILN for effective and robust segmentation of different anatomies.

For network architecture design in training Deep Convolutional Neural Networks (DCNNs), the newly proposed ACNN is compared to traditional U-Net where max-pooling and deconvolutional layers are essential. Only convolutional layers are used in the proposed ACNN with different atrous rates and it has been shown that the method is able to provide a fully-covered receptive field with a minimum number of atrous convolutional layers. ACNN enhances the robustness and generalizability of the analysis scheme by cascading multiple atrous blocks. Validation results have shown the proposed

method achieves comparable results to the U-Net in terms of medical image segmentation, whilst reducing the trainable parameters, thus improving the convergence and real-time instantiation speed.

For 3D shape instantiation of soft and deforming organs during MIS, Sparse Principle Component Analysis (SPCA) has been used to analyse a 3D Statistical Shape Model (SSM) and to determine the most informative scan plane. Synchronized 2D images are then scanned at the most informative scan plane and are expressed in a 2D SSM. Kernel Partial Least Square Regression (KPLSR) has been applied to learn the relationship between the 2D and 3D SSM. It has been shown that the KPLSR-learned model developed in this thesis is able to predict the intra-operative 3D target shape from a single 2D projection or slice, thus permitting real-time 3D navigation. Validation results have shown the intrinsic accuracy achieved and the potential clinical value of the technique.

The proposed 3D shape instantiation scheme is further applied to intra-operative stent graft deployment for the robot-assisted treatment of aortic aneurysms. Mathematical modelling is first used to simulate the stent graft characteristics. This is then followed by the Robust Perspective-n-Point (RPnP) method to instantiate the 3D pose of fiducial markers of the graft. Here, Equally-weighted Focal U-Net is proposed with a cross-entropy and an additional focal loss function. Detailed validation has been performed on patient-specific stent grafts with an accuracy between $1 - 3mm$. Finally, the relative merits and potential pitfalls of all the methods developed in this thesis are discussed, followed by potential future research directions and additional challenges that need to be tackled.

Acknowledgements

I would like to gratefully thank my supervisor Professor Guang-Zhong Yang. He is not only my research supervisor, where he led my research in medical robotics, but is also my lifelong mentor, giving me many useful suggestions on non-technical choices and circumstances. It is my great honor and luck to have such a supervisor, hence I was able to not only achieve academic merits, but was also able to enjoy a very happy four-year PhD life.

Thanks to my supervisor Dr Su-Lin Lee very much for her supervision during my PhD. In the first two years of my PhD, she supervised me with great patience and time. We fought for each deadline side by side. It is her great effort that gradually led me to adjust to my PhD life.

Great thanks to Lady Helen Hamlyn for funding the Hamlyn Centre for Robotic Surgery. It is her kind support that brings so many researchers together to work on medical robotics.

Thank my co-supervised students Jian-Qing Zheng, Zhao-Yang Wang and Peichao Li for their kind support of the collaboration on research. It is their great effort that makes my PhD work more complete and influential. Especially, I would like to thank my junior colleague Qing-Biao Li for his great support during my PhD. No matter as a collaborator or a friend, his gentle, kind, nice and supportive character lights my PhD life.

Thank you to my father, mother, brother, brother's wife and my cute niece for their selfless love and support during my PhD time. I love you guys 100%.

Thanks to Dr Anzhu Gao, Dr Haojie Zhang and Dr Haijie Chen for their kind help with the GPU. Thanks to my Hamlyn colleagues: Ning Liu, Dr Bidan Huang, Dr Liang Zhao, Dr Yun Gu, Dr Yao Guo, Daniel Freer, Mohamed E. M. K. Abdelaziz, Xu Chen for their gentle help and collaboration. Especially thank my roommates Dr Mali Shen and Dandan Zhang for their kind taking care in my daily life. Thanks to my best friends Dr Qi Ye and Dr Juan Wu for their kind advice and suggestions for many things.

I would like to thank Dr Karim Lekadir and Dr Robert Merrifield for supplying the cardiac data, Dr Maria Hawkins and Dr Diana Tait for the liver patient data, and Dr Mirna Lerotic for her assistance with the finite

element simulations. Also thanks to Northwick Park hospital for supplying the usage of CT.

Some of the research was supported by the Engineering and Physical Sciences Research Council UK (EP/L020688/1). I gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for some of the research.

Declaration of Originality

This is to certify that this thesis embodies the results of my own course of study and research. All else from other published or unpublished work is appropriately referenced.

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives (CC BY-NC) licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Contents

List of Tables

List of Figures

Abbreviations

AAA Abdominal Aortic Aneurysm

ACNN Atrous Convolutional Neural Network

AP Anteroposterior

API Application Programming Interface

ASPP Atrous Spatial Pyramid Pooling

BIN Batch-Instance Normalization

BIRNet Brain Image Registration Network

BN Batch Normalization

CAE Convolutional AutoEncoder

CFCN Cascade Fully Convolutional Network

CNN Convolutional Neural Network

CT Computed Tomography

DCNN Deep Convolutional Neural Network

DLIR Deep Learning Image Registration

DRL Deep Reinforcement Learning

DSC Dice Similarity Coefficient

DVF Displacement Vector Field

EVAR Endovascular Aortic Repair

FCN Fully Convolutional Network

FEM Finite Element Modeling

FEVAR Fenestrated Endovascular Aortic Repair

FFD Free Form Deformation

FGN Fine Group Normalization

GCN Graph Convolutional Network

G-CNN Group Convolutional Neural Network

GN Group Normalization

HCM Hypertrophic Cardiomyopathy

ILN Instance-Layer Normalization

IN Instance Normalization

IoU Intersection over Union

KPLSR Kernel Partial Least Square Regression

LARS-EN Least Angle Regression Elastic Net

LN Layer Normalization

LV Left Ventricle

Mask R-CNN Mask Region-CNN

MIS Minimally Invasive Surgery

mIoU mean Intersection over Union

MRI Magnetic Resonance Imaging

NIPALS Non-linear Iterative partial Least Squares

PCA Principal Component Analysis

PCR Principal Component Regression

PET Positron-Emission Tomography

PLSR Partial Least Squares Regression

R-CNN Regional Convolutional Neural Network

RFCA Radio-Frequency Cardiac Ablation

RNN Recurrent Neural Network

RoI Region of Interest

RPnP Robust Perspective-n-Point

RR Ridge Regression

RV Right Ventricle

SCoTLASS Simplified Component Technique Least Absolute Shrinkage and Selection operator

SGD Stochastic Gradient Descent

SPCA Sparse Principal Component Analysis

SSM Statistical Shape Model

SVM Support Vector Machine

TEE Transesophageal Echocardiography

1 Introduction

1.1 Motivation

With continuing technological advances, surgery is moving from traditional open surgery to Minimally Invasive Surgery (MIS), and more recently to robot-assisted MIS. In traditional open surgery, an operation is performed through a large incision, usually around $10cm$, causing large scars, long recovery time and much pain [7]. In MIS, operation tools are inserted through a much smaller incision, usually around $2cm$, with smaller scars, shorter recovery time and less pain [7]. In robot-assisted MIS, instruments are enhanced by articulated wrists, vision is enhanced for bimanual operation, and both pre-operative and intra-operative images can be combined for effective surgical navigation. An illustration of a current robot-assisted MIS system is shown in Fig. 1.1. Thanks to these developments, surgery is now focused on the systematic level impact on patients, avoiding isolated surgical treatment or anatomical alteration, with careful consideration of metabolic, haemodynamic and neurohormonal consequences that can influence the quality of life. These advances are underpinned by continuing technological developments in diagnosis and imaging.

Unlike in traditional open surgery, where surgeons can see the operation area with naked eyes, In MIS, due to the small incision, the operation area is usually not visible. Common 3D imaging techniques including Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and ultrasound are not applicable for supplying 3D navigation as well due to the radiation, time in-efficiency or low-resolution. For MIS that can be accessed by cameras or catheters, 2D RGB images or 3D point clouds are usually used for navigation. For example, for bronchoscopic biopsy, 2D RGB images are used to find the catheter position in airways with the help of electra-magnetic tracking [8, 9]. For Radio-Frequency Cardiac Ablation (RFCA), catheters are inserted intra-operatively first to collect a 3D point cloud and then the reconstructed mesh

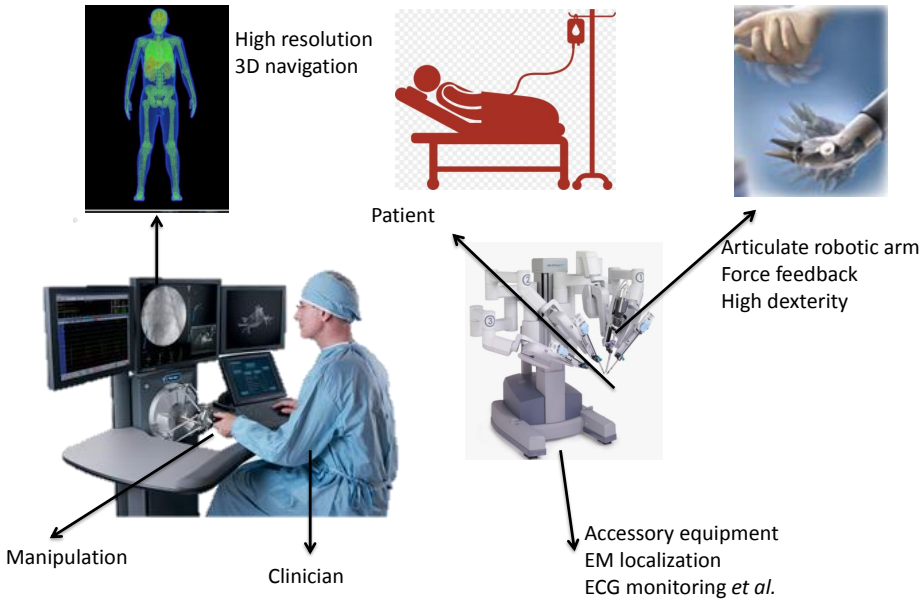


Figure 1.1: A brief illustration of robot-assisted MIS system, with robot, manipulation clinician, navigation system, and patient.

based on the collected point cloud is used for navigation [10]. For surgeries without the access of cameras or catheters, 2D views including projections from fluoroscopy, 2D images from ultrasound and also 2D slices from CT or MRI are used [6, 11–15]. This 2D navigation is not sufficient for MIS which is a 3D task. Therefore, there is a pressing need of developing 3D shape reconstruction techniques directly from 2D views. In this thesis, I am working on 3D shape instantiation which instantiates the 3D shape of a target from limited 2D views, especially a single 2D projection or slice. With limiting the required input to be a single 2D view, real-time 3D navigation could potentially possible. Fig. 1.2 illustrates the 3D shape instantiation concept developed in this thesis, where the 3D shape of metastatic liver is reconstructed from their corresponding single 2D MRI slice.

During MIS, the deformation of soft tissue is complex. Complex surgical navigation and planning are made possible through the use of both pre- and intra-operative imaging techniques such as ultrasound, CT, and MRI [16]. In this thesis, a learning-based general shape instantiation framework is proposed to reconstruct the 3D shape of a soft organ from its a single 2D view. The method learns the target deformation from pre-operatively collected training

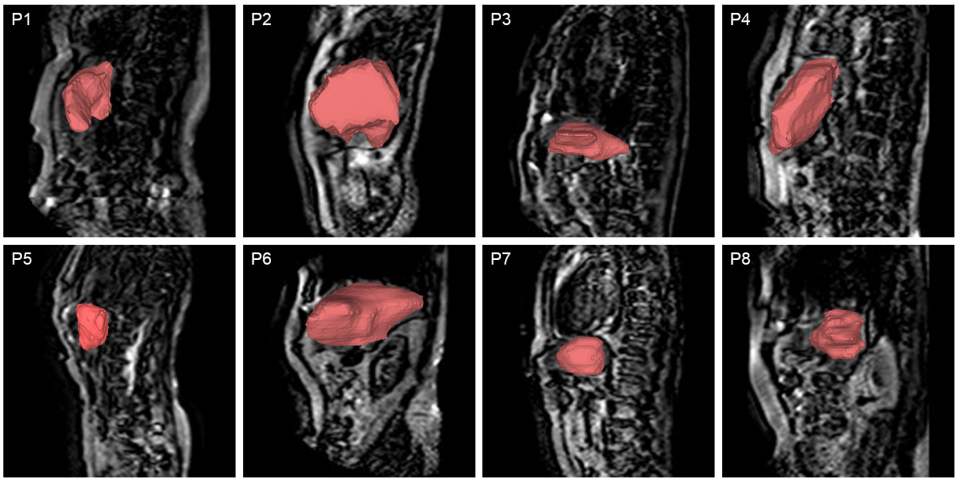


Figure 1.2: An illustration of the concept of 3D shape instantiation, where the 3D shapes of eight metastatic livers are reconstructed from their a single 2D MRI slice respectively. These liver shapes look different from normal ones, as these patients experienced the liver resection operation before.

data, including 2D projections or slices and the corresponding 3D shapes at different time positions. The learned model is then used to reconstruct intra-operatively in real-time the target 3D shape from a single new 2D projection or slice.

The method is also generalized to stent graft deployment during endovascular intervention of aortic aneurysms. During such procedures, a stent graft is compressed into a delivery device, advanced to the target aneurysm and then deployed. In this thesis, an effective 3D instantiation scheme is proposed for the interactive placement of stent graft with the alignment of fenestrations to side branch blood vessels and scallops to anchoring sites. With this method, the deformation of a stent graft is decomposed into multiple rigid deformation components. Each rigid deformation component is reconstructed by the Robust Perspective-n-Point (RPnP) method [17] and these rigid deformation components are then combined to reconstruct the 3D shape of the entire stent graft.

For achieving a complete and automatic 3D shape instantiation pipeline, image segmentation based on Deep Convolutional Neural Network (DCNN) is also explored, including the normalization methods in training DCNN and neural architecture design of Atrous Convolutional Neural Network (ACNN)

for medical image segmentation.

1.2 Thesis Outline

The outline of this thesis is as follows:

In Chapter 1, the background and basic concept of 3D shape instantiation are introduced.

In Chapter 2, recent techniques of machine learning in surgery are reviewed, particularly for those methods related to surgical planning and medical image analysis. The emerging trends and major challenges are also highlighted.

In Chapter 3, Batch Normalization (BN), Instance Normalization (IN), Layer Normalization (LN) and Group Normalization (GN) are reviewed in terms of relative merits and potential problems when used for medical image segmentation. Three datasets covering the Right Ventricle (RV), aorta, and Left Ventricle (LV) are used for the validation. Although most DCNNs adopt BN as the normalization method by default without a careful consideration of its performance, the results show that a detailed subdivision of the feature map, i.e., GN with a large group number or IN, achieves a higher accuracy. Considering the fact that in most of existing methods, normalization for each layer is fixed. Batch-Instance Normalization (BIN) is one of the first proposed methods that combines two different normalization methods to achieve diverse normalization for different layers. However, two potential issues exist in BIN, first, the clip function is not differentiable everywhere. Second, the combined feature map does not follow a normalized distribution, which may be detrimental for signal propagation in DCNN. Hence, Instance-Layer Normalization (ILN) is proposed by using the sigmoid function for combining feature maps and cascading group normalization afterwards. The performance of ILN is validated on the RV and LV segmentation, and the results show that the proposed ILN outperforms existing normalization methods with accuracy improvements.

In Chapter 4, the issue of down-sampling DCNN is investigated. The current DCNNs usually use down-sampling layers for increasing the receptive field and gaining abstract semantic information. These down-sampling layers decrease the spatial dimension of feature maps, which can be detrimental to medical image segmentation. Atrous convolution is an alternative to the down-sampling layer. It increases the receptive field whilst maintaining the

spatial dimension of feature maps. In this chapter, a method for effective atrous rate setting is proposed to achieve fully-covered receptive field with a minimum number of atrous convolutional layers. Furthermore, different atrous blocks, shortcut connections and normalization methods are explored to select the optimal network architecture settings. These lead to a new and full-scale DCNN - Atrous Convolutional Neural Network (ACNN), which incorporates cascaded atrous II-blocks, residual learning and Fine Group Normalization (FGN). Application results of the proposed ACNN demonstrate that the proposed ACNN can achieve comparable segmentation Dice Similarity Coefficients (DSCs) to that of the U-Net, optimized U-Net with IN and hybrid networks, but with significantly reduced trainable parameters and therefore is computationally efficient for both training and inference.

In Chapter 5, a real-time and registration-free framework for dynamic 3D shape instantiation is proposed. With this method, an approximate optimal scan plane is first determined by analysing the pre-operative 3D Statistical Shape Model (SSM) of the anatomy with Sparse Principal Component Analysis (SPCA) and considering practical constraints. Kernel Partial Least Square Regression (KPLSR) is then used to learn the relationship between the pre-operative 3D SSM and a synchronized 2D SSM constructed from 2D projections or slices obtained at the approximate optimal scan plane. Finally, the derived relationship is applied to a new intra-operative 2D projection or slice obtained at the same scan plane to predict the high-resolution 3D shape intra-operatively. A major feature of the proposed framework is that no extra registration between the pre-operative 3D SSM and the synchronized 2D SSM is required. Detailed validation is performed and the results (mean accuracy of $2.19mm$ on patients with a real-time computation speed of $1ms$) demonstrate its potential for clinical use for real-time, high-resolution, dynamic 3D intervention and guidance.

In Chapter 6, a real-time framework is proposed to reconstruct the 3D shape of a fenestrated stent graft utilising only a single low-dose 2D fluoroscopic projection. First, markers are placed on the fenestrated stent graft. Second, the 3D pose of each stent segment is reconstructed by the RPnP method. Third, the 3D shape of whole stent graft is reconstructed via graft gap interpolation. In addition, Equally Weighted Focal U-Net is proposed to segment the fluoroscopic projections of customized markers into multiple classes and to determine the centres of markers. The proposed Equally

Weighted Focal U-Net utilises U-Net as the network architecture, equally weighted loss function for initial marker segmentation, and then equally weighted focal loss function for improving the initial marker segmentation. The method is validated on patient-specific datasets, achieving an average distance error of $1 - 3mm$.

Finally, Chapter 7 summarises the technical achievements, relative merits and potential pitfalls of the methods proposed in this thesis, as well as potential future research directions.

1.3 Publications

Throughout this PhD, a number of conference and academic journal publications have been achieved. These include:

Publications included in this thesis:

1 . **Xiao-Yun Zhou**, Yao Guo, Mali Shen, Guang-Zhong Yang. "Artificial intelligence in surgery [J]". *Invited review paper by Frontier of Medicine*, accepted. [18]

2 . **Xiao-Yun Zhou**, Guang-Zhong Yang, Su-Lin Lee. "A real-time and registration-free framework for dynamic shape instantiation [J]", *Medical Image Analysis (**MedIA**)*, 44: 86-97, 2018. [19]

3 . **Xiao-Yun Zhou**, Jianyu Lin, Celia Riga, Guang-Zhong Yang, Su-Lin Lee. "Real-time 3D shape instantiation from single fluoroscopy projection for fenestrated stent graft deployment [J+C]". *IEEE Robotics and Automation Letters (**RAL**) + International Conference on Robotics and Automation (**ICRA**)*, 3(2): 1314-1321, 2018. [20]

4 . **Xiao-Yun Zhou**, Celia Riga, Su-Lin Lee, Guang-Zhong Yang. "Towards automatic 3D shape instantiation for deployed stent grafts: 2D multiple-class and class-imbalance marker segmentation with equally-weighted focal U-Net [C]", *IEEE/RSJ International Conference on Intelligent Robots and Systems (**IROS**)*, 1261-1267, 2018. [21]

5 . **Xiao-Yun Zhou**, Guang-Zhong Yang. "Normalization in training U-Net for 2D biomedical semantic segmentation [J]", *IEEE Robotics and Automation Letters (**RAL**)*, 4(2): 1792-1799, 2019. [4]

6 . **Xiao-Yun Zhou**, Qing-Biao Li, Mali Shen, Guang-Zhong Yang. "U-Net training with Instance-Layer Normalization [C]", *Multiscale Multimodal Medical Imaging workshop in conjunction with International Conference on Medical Image Computing and Computer Assisted Intervention (**MICCAI MMMI workshop**)* 101-108, 2019. [22]

Publications not included in this thesis, but related to my PhD work:

7 . **Xiao-Yun Zhou***, Zhao-Yang Wang*, Peichao Li, Jian-Qing Zheng, Guang-Zhong Yang. "One-stage shape instantiation from a single 2D image to 3D point cloud [C]", *International Conference on Medical Image Computing and Computer Assisted Intervention (**MICCAI**)* 30-38, 2019. [23]

8 . **Xiao-Yun Zhou**, Celia Riga, Guang-Zhong Yang, Su-Lin Lee. "Stent graft shape instantiation for fenestrated endovascular aortic repair [C]". *The Hamlyn Symposium on Medical Robotics*, 2017. [11]

9 . Jian-Qing Zheng*, **Xiao-Yun Zhou***, Celia Riga, Guang-Zhong Yang. "Real-time 3D shape instantiation of partially-deployed stent segment from a single 2D fluoroscopic image for fenestrated endovascular aortic repair [J]". *IEEE Robotics and Automation Letters (**RAL**)*, 4(4): 3703-3710, 2019. [13]

10 . **Xiao-Yun Zhou**, Sabine Ernst, Su-Lin Lee. "Path planning for robot-enhanced cardiac radiofrequency catheter ablation [C]", *IEEE International Conference on Robotics and Automation (**ICRA**)*, 4172-4177, 2016. [10]

11 . Jian-Qing Zheng, **Xiao-Yun Zhou**, Celia Riga, Guang-Zhong Yang. "Towards 3D path planning from a single 2D fluoroscopic image for robot assisted fenestrated endovascular aortic repair [C]", *IEEE International Conference on Robotics and Automation (**ICRA**)*, 8747-8753, 2019. [14]

12 . Yingjing Feng, Ziyan Guo, Ziyang Dong, **Xiao-Yun Zhou**, Ka-Wai Kwok, Sabine Ernst, Su-Lin Lee. "An efficient cardiac mapping strategy for radiofrequency catheter ablation with active learning [J]",

*International journal of computer assisted radiology and surgery (**IJCARS**)*, 12(7): 1199-1207, 2017. [15]

2 Machine Learning in Medical Imaging

[1]

Machine learning is a useful tool in medical imaging. Before detailing the technical approaches developed during this PhD, the applications of machine learning, especially deep learning, in medical classification, detection, segmentation, registration and shape instantiation are briefly discussed. These application domains cover both pre-operative diagnosis and intra-operative navigation.

2.1 Introduction

The use of machine learning for medicine can be dated back to the early years of developing the MYCIN system [24]. Machine learning is able to "see" patterns from data, extract meaningful features and combine features for computer-aided diagnosis and decision support system. It is now increasingly used for risk stratification [25], genomics [26], imaging and diagnosis [27, 28], precision medicine [6], and drug discovery [29]. The introduction of machine learning in surgery is more recent and it has a strong root in imaging and navigation, with early techniques focused on feature detection and computer assisted intervention.

Traditional supervised learning methods include Support Vector Machine (SVM) [30], decision tree [31] and naive Bayes [32] while traditional unsupervised learning methods include K-means [33], Gaussian mixture model [34] and Markov random fields [35]. With recent successes of AlexNet [36], deep learning methods, especially Deep Convolutional Neural Network (DCNN) where multiple convolutional layers are cascaded, have enabled automatically

[1] Part of the content in this chapter are based on [Xiao-Yun Zhou, Yao Guo, Mali Shen, Guang-Zhong Yang. "Artificial Intelligence in Surgery" Frontier of Medicine, accepted.]

learned data-driven descriptors, rather than ad hoc hand-crafted features, to be used for image understanding with improved robustness and generalizability. VGGNet extended the AlexNet with deeper layers and smaller convolutional kernels [37] while ResNet extended the VGGNet to be much deeper with using residual learning [38]. Faster Regional Convolutional Neural Network (R-CNN) extended the application of deep learning from image classification to object detection [39], where the bounding boxes of objects were also regressed. Fully Convolutional Network (FCN) extended the application of deep learning from object detection to image segmentation [40]. After that, machine learning especially deep learning has been widely applied in medical image analysis including classification, detection, segmentation, registration and shape instantiation. Common network architectures used in medical imaging are summarized in Fig. 2.1.

2.2 Applications of Machine Learning in Medical Imaging

2.2.1 Classification

Classification outputs the diagnostic value of the input which is a single or a set of medical images or volumes of organs or lesions. In addition to traditional machine learning and image analysis techniques, deep learning based methods for pre-operative planning are on the rise in the research community [41]. For the latter, the network architecture for classification is composed of convolutional layers for extracting information from the input images or volumes and fully connected layers for regressing the diagnostic value.

For example, a classification pipeline with a Convolutional Neural Network (CNN) architecture of Google's Inception, with Inception and ResNet algorithm and with different training strategies has been proposed to classify the lung, bladder and breast cancer types [42]. Chilamkurthy *et al.* demonstrate that deep learning can recognize intracranial haemorrhage, calvarial fracture, midline shift and mass effect through testing a set of deep learning algorithms on head Computed Tomography (CT) scans [27]. The mortality, renal failure and post-operative bleeding in patients after cardiosurgical care can be predicted by Recurrent Neural Network (RNN) in real time with improved

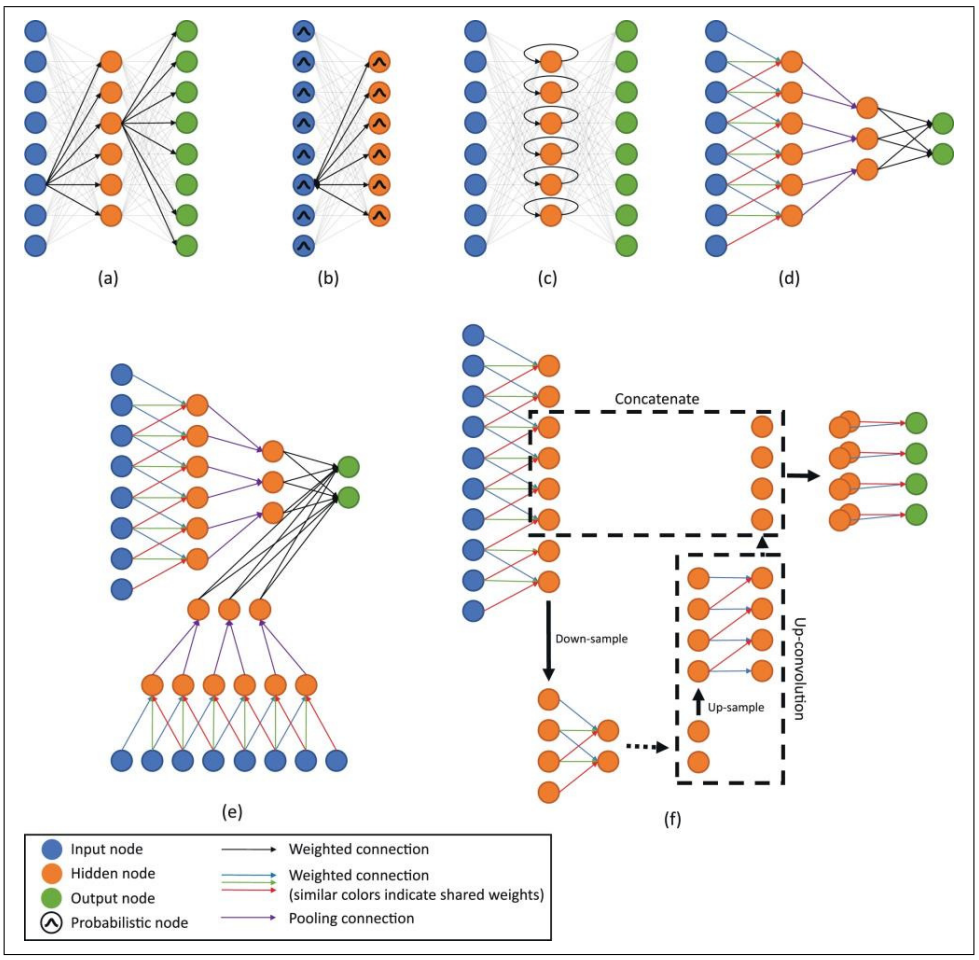


Figure 2.1: Node graphs of 1D representations of architectures commonly used in medical imaging. (a) Auto-encoder, (b) restricted Boltzmann machine, (c) recurrent neural network, (d) convolutional neural network, (e) multi-stream convolutional neural network, (f) U-net (with a single downsampling stage). *"Reprinted from Publication A survey on deep learning in medical image analysis, 42, Litjens, Geert and Kooi, Thijs and Bejnordi, Babak Ehteshami and Setio, Arnaud Arindra Adiyoso and Ciompi, Francesco and Ghafoorian, Mohsen and Van Der Laak, Jeroen Awm and Van Ginneken, Bram and Sánchez, Clara I, 2.2. Neural networks, Pages 63, Copyright (2019), with permission from Elsevier."*

accuracy compared to standard-of-care clinical tools [25]. ResNet-50 and Darknet-19 have been used to classify benign or malignant lesions in ultrasound images, showing similar sensitivity and improved specificity [43]. These studies show promising human-level accuracy with good reproducibility, but explainability of these approaches remains a potential hurdle for regulatory considerations.

2.2.2 Detection

Detection provides the spatial localization of regions of interest, often in the form of bounding boxes or landmarks, additionally to image- or region-level classification. Similarly, deep learning based approaches have shown promising results. Compared to traditional algorithms which are task-specific due to hand-crafted feature extractors, DCNNs for detection usually consist of convolutional layers for feature extraction and regression layers for regressing the bounding box properties.

For detecting prostate cancer from 4D Positron-Emission Tomography (PET) images, a deeply stacked convolutional autoencoder was trained to extract the statistical and kinetic biological features [44]. For pulmonary nodule detection, 3D Group Convolutional Neural Networks (G-CNNs) were proposed with good accuracy, sensitivity and convergence speed [45]. CNNs were frequently used in orthopaedics for cartilage lesion detection [46]. For breast lesion detection, Deep Reinforcement Learning (DRL) based on an extension of the deep Q-network was used to learn a search policy from dynamic contrast-enhanced Magnetic Resonance Imaging (MRI) [47]. To detect acute intracranial haemorrhage from CT scans and to improve network interpretability, Lee *et al.* [48] used an attention map and an iterative process to mimic the workflow of radiologists.

2.2.3 Segmentation

Segmentation can be treated as a pixel- or voxel-level image classification problem. Early works on deep learning for segmentation often adopted a sliding window based system. Specifically, each image or volume was divided into small windows, CNNs were trained to predict the target label at the central location of the window. Image- or voxel-wise segmentation can be achieved by running the CNN classifier over densely sampled image

windows. One of the well-known networks that falls into this category is Deepmedic, which had shown good performances for multi-modal brain tumour segmentation from MRI [49]. However, the sliding window based system is inefficient as the network activations of overlapping regions were computed repeatedly. More recently, it was replaced by FCNs [40]. The key idea was to replace the fully connected layers in a classification network with convolutional layers and up-sampling layers, which significantly improved the segmentation efficiency. For medical image segmentation, U-Net [3] [50], or more generally, encoder-decoder network is a representative FCN that has shown promising performances. The encoder has multiple convolutional and down-sampling layers that extract image features at different scales. The decoder has convolutional and up-sampling layers that recover the spatial resolution of feature maps and finally achieves pixel- or voxel-wise dense segmentation. A review of different normalization methods in training U-Net for medical image segmentation could be found in [4] and Instance-Layer Normalization (ILN) was proposed for training U-Net for medical image segmentation in [22].

For navigating the endoscopic pancreatic and biliary procedures, Gibson *et al.* [51] used dilated convolutions and fused image features at multiple scales for segmenting abdominal organs from CT scans. For interactive segmentation of placenta and fetal brains from MRI, FCN and user defined bounding boxes and scribbles were combined, where the last few layers of FCN were fine-tuned based on the user input [52]. For aortic MRI, Bai *et al.* [53] combined FCN with RNN to incorporate spatial and temporal information. The segmentation and localization of surgical instrument landmarks were modelled as heatmap regression and FCN was used to track the instruments in near real-time [54]. For the segmentation and labelling of vertebrae from CT and MRI, Lessmann *et al.* proposed an iterative instance segmentation approach with FCN, where the network concurrently performed vertebra segmentation, regressed the anatomical landmark and predicted the vertebrae visibility [55]. For pulmonary nodule segmentation, Feng *et al.* addressed the issue of requiring accurate manual annotations when training FCNs by learning discriminative regions from weakly-labelled lung CT with a candidate screening method [56].

2.2.4 Registration

Registration is the spatial alignment between two medical images, volumes or modalities, which is particularly important for both pre- and intra-operative planning. Traditional algorithms usually iteratively calculate a parametric transformation, i.e., elastic, fluid or B-spline model to minimize a given metric, i.e., mean square error, normalized cross correlation, or mutual information, between the two medical images, volumes or modalities. Recently, deep regression models have been used to replace the traditional time consuming and optimization based registration algorithm.

Example deep learning based approaches include VoxelMorph based on CNN structures for maximizing the standard image matching objective functions by leveraging auxiliary segmentation to map an input image pair to a deformation field [57]. An end-to-end deep learning framework was proposed with three stages: affine transform prediction, momentum calculation and non-parametric refinement to combine affine registration and vector momentum-parameterized stationary velocity field for 3D medical image registration [58]. Pulmonary CT images were registered by training a 3D CNN with synthetic random transformation [59]. A weakly supervised framework was proposed for multi-modal image registration, with training on images with higher-level correspondence, i.e., anatomical labels, rather than voxel-level transformation for predicting the displacement field [60]. Markov decision process with each agent trained with dilated FCN was applied to align a 3D volume to 2D X-ray images [61]. Brain Image Registration Network (BIRNet) was proposed to predict deformation from image appearance for image registration, with training an FCN with both the ground truth and image dissimilarity measures, where the FCN was improved with hierarchical loss, gap filling and multi-source strategies [62]. A Deep Learning Image Registration (DLIR) framework was proposed to train CNN on image similarity between fixed and moving image pairs, hence affine and deformable image registration can be achieved in an unsupervised manner [63]. RegNet had been proposed by considering multi-scale contexts and was trained on artificially generated Displacement Vector Field (DVF) to achieve a non-rigid registration [64]. 3D image registration can also be formulated as a strategy learning process with 3D raw image as the input, the next optimal action, i.e., up and down, as the output, CNN as the agent [65].

2.2.5 3D Shape Instantiation

For intra-operative 3D reconstruction, 3D volumes can be scanned with MRI, CT or ultrasound. In practice, this process (3D/4D) can be time-consuming or with a low resolution and is not applicable for intra-operative navigation. Real-time 3D shape instantiation which reconstructs the intra-operative 3D shape from a single or limited 2D images is an emerging area of research in intra-operative guidance.

For example, a 3D prostate shape was reconstructed from multiple non-parallel 2D ultrasound images with a radial basis function [66]. The 3D shape of Abdominal Aortic Aneurysm (AAA) was reconstructed from two 2D fluoroscopic images [67]. The 3D shapes of fully-compressed, fully-deployed and also partially-deployed stent grafts were reconstructed from a single projection of 2D fluoroscopy with mathematical modelling, combined with the Robust Perspective-n-Point (RPnP) method [17], graft gap interpolation and Graph Convolutional Network (GCN) [6, 11, 68]. Furthermore, Equally Weighted Focal U-Net [6] was proposed to automatically segment the makers on stent grafts to improve the efficiency of the intra-operative stent graft shape instantiation framework [21]. Moreover, the 3D AAA skeleton was reconstructed from a single projection of 2D fluoroscopy with skeleton deformation and graph matching [69]. The 3D liver shape was reconstructed from a single 2D projection or slice with Principal Component Analysis (PCA), Statistical Shape Model (SSM) and Partial Least Squares Regression (PLSR) [70]. This work was further generalized to a registration-free shape instantiation framework for any dynamic organ with sparse PCA, SSM and kernel PLSR [19]. Recently, an advanced deep and one-stage learning strategy that estimates 3D point cloud from a single 2D projection was proposed for 3D shape instantiation [23].

2.3 Conclusion and Future Outlook

Machine learning has been widely adopted in medical imaging for tasks ranging from anatomical classification, detection, segmentation, registration to instantiation. The results seem to suggest that the deep learning based methods can outperform those rely on conventional approaches. However, data-driven approaches often suffer from inherited limitations, making the

deep learning based approaches less generalizable for a different test data, less explainable in terms of the results and more data-demanding.

To overcome these issues, close collaborations between multidisciplinary teams, particularly the surgeons and machine learning researchers should be encouraged to generate large scale annotated data, providing more training data for machine learning algorithms. An alternative solution is to develop machine learning techniques such as meta-learning [71], or learning to learn [72], that enable generalizable systems to perform diagnosis with limited dataset yet improved explainability.

Although many state-of-the-art machine learning and deep learning algorithms have made breakthroughs in the field of general computer vision, the differences between medical and natural images may impede their direct clinical applicability. In addition, the underlying models and the derived results may not be easily interpretable by humans, therefore it raises issues such as potential risks and uncertainty in surgery. Potential solutions to these problems would be to explore different transfer learning techniques to mitigate the differences between natural and medical image modalities and to develop more explainable machine learning algorithms to enhance its decision-making performance. Furthermore, utilizing personalized multimodal patient information, including omics-data and life style information, in the development of machine learning can be useful in early detection and diagnosis, leading to personalized treatment. These also allow early treatment options featured with minimal trauma, smaller surgical risks and shorter recovery time.

In addition to these common challenges in applying machine learning in medicine, another two key and specific challenges for intra-operative navigation are: 1) the deformation of organs/tissues forcing the pre-operative planning to work with a dynamic and uncertain environment during surgery; 2) during a surgery, one important requirement is to assist surgeons in real-time. In this thesis, I am working on these two challenges and proposing 3D shape instantiation which is a real-time and instantaneous high-resolution 3D reconstruction technique based on an input of a single or limited 2D views. Specifically, in my work, the only input is usually a single 2D projection or slice and the algorithm running time is faster than the image update time, hence real-time 3D reconstruction could be achieved and updated along the dynamic environment.

3 Normalization in Training U-Net for Medical Image Segmentation

[1]]

Medical image segmentation, which labels the class, anatomy, or medical device of each pixel in an image, is important and fundamental for many medical tasks. In my PhD work, segmenting the Right Ventricle (RV) from 2D Magnetic Resonance Imaging (MRI) images is essential to reconstruct 3D RV shapes for intra-operative navigation in robotic cardiac interventions [6]. In 3D robotic path planning for Fenestrated Endovascular Aortic Repair (FEVAR), segmenting markers on fenestrated stent grafts is useful for reconstructing 3D stent graft shapes [73] and segmenting the aorta is useful for reconstructing the intra-operative 3D Abdominal Aortic Aneurysm (AAA) skeleton [14, 74]. For a complete 3D shape instantiation pipeline, during my PhD, I worked on proposing new and general segmentation methods. To connect with my 3D shape instantiation work in chapter 5 and 6, the RV, Left Ventricle (LV) and aortic segmentation are used as the main validation. In this chapter, I will introduce my work on the normalization in training U-Net for medical image segmentation.

3.1 Introduction

Conventional segmentation methods for both natural and medical problems are usually based on features (edge, region, angle, etc.) which need an expert-designed feature extractor and trained classifier, while recent segmentation methods based on Deep Convolutional Neural Network (DCNN) extract and

[1]The content of this chapter is based on [Xiao-Yun Zhou, and Guang-Zhong Yang. "Normalization in training U-Net for 2D biomedical semantic segmentation." IEEE Robotics and Automation Letters 4.2 (2019): 1792-1799.] and [Xiao-Yun Zhou, Peichao Li, Zhao-Yang Wang, Guang-Zhong Yang. "U-Net Training with Instance-Layer Normalization." MICCAI-MMMI workshop, 2019: 101-108.

classify the features automatically with multiple non-linear modules [75]. Fully Convolutional Network (FCN) is the first proposed DCNN which realized pixel-level classification and hence semantic segmentation by using convolutional and deconvolutional layers, as well as skip architectures [40]. Ronneberger *et al.* introduced FCN into 2D biomedical semantic segmentation, proposed U-Net with dense skip architectures, and achieved reasonable results on neuronal structure and cell segmentation [3]. A systematical review has been carried out by Litgens *et al.* on the application of DCNN in medical image analysis including segmentation, classification, detection, registration and other tasks [41]. DCNNs for medical semantic segmentation could be divided into 3D [76] and 2D [77] based on the dimension of convolution. As the main purpose of the work in this chapter is to increase the automation of 3D shape instantiation in chapter 5 and 6, where the input image is usually 2D, hence this chapter mainly focuses on 2D DCNN.

Most of previous research on DCNNs for medical image segmentation focused on architecture design, loss function, and network cascade for specific tasks. For example, atriaNet composed of multi-scaled and dual-pathed convolutional architectures was proposed for left atrial segmentation from late gadolinium enhanced MRI [78]. A hierarchical DCNN was designed with a two-stage FCN and dice-sensitivity-like loss function to segment breast tumours from dynamic contrast-enhanced MRI [79]. The thrombus was segmented from CT images with detectnet, FCN and holistically-nested edge detection [80]. Equally-weighted Focal U-Net combined with focal loss and U-Net was proposed to segment the small metal markers from fluoroscopic images of fenestrated stent grafts [21].

One fundamental component in DCNN is the normalization layer. Initially, one of the main motivations for normalization was to alleviate the internal covariate shift where layers' input distribution changes [81]. The main step in DCNN is to apply convolutional kernels with trainable parameters on feature maps to extract new features iteratively. During the training of a DCNN, the input of a layer depends on all the parameters/values in its previous layers/feature maps. Small changes in shallow input feature maps or image batches accumulate and amplify along the depth of network, causing deep layers to be trained to fit these distribution changes rather than the real and useful content. This phenomenon is called internal covariate shift [81]. However, recent work considers the use of normalization layer is

also beneficial, because it increases the robustness of networks to fluctuation associated with random weight initialization [82], or it achieves smoother optimization landscape [83]. In this chapter, we keep this motivation question open and focus on identifying efficient normalization strategies.

For a feature map with dimension of $(\mathrm{N}, \mathrm{H}, \mathrm{W}, \mathrm{C})$, where N is the batch size, H is the feature height, W is the feature width, C is the feature channel, Batch Normalization (BN) [81] [84] was the first proposed normalization method which calculated the mean and variance of a feature map along the $(\mathrm{N}, \mathrm{H}, \mathrm{W})$ dimension, then re-scaled and re-translated the normalized feature map with additional trainable parameters to preserve the DCNN representation ability. Instance Normalization (IN) [85] which calculated the mean and variance along the (H, W) dimension was proposed for fast stylization. Layer Normalization (LN) [86] which calculated the mean and variance along the $(\mathrm{H}, \mathrm{W}, \mathrm{C})$ dimension was proposed for recurrent networks. Group Normalization (GN) [87] calculated the mean and variance along the (H, W) and multiple-channels dimension (C) and was evaluated on image classification and instance segmentation. Weight normalization [88] [89] based on re-parameterization on weights was used in recurrent models and reinforcement learning. Batch Kalman normalization estimated the mean and variance considering all preceding layers [90]. There are also researches proceeding to other aspects. L^1 and L^∞ BN [91] was proposed for half-precision (16bit) implementation. Normalization propagation [92] estimated the mean and standard deviation data-independently. Spectral normalization [93] and virtual batch normalization [94] were proposed specifically for training generative adversarial networks. Cosine normalization [95] applied cosine similarity instead of dot multiplication in DCNN.

These normalization methods are proposed for different tasks and there are thus far no specific review comparisons regarding their performance in medical image segmentation. The comparisons in [87] between BN, IN, LN and GN/BN and GN are for image classification/instance segmentation. In natural semantic segmentation (3.1a), parameters are usually sharable between tasks and fine-tuning or extracting features from pre-trained feature maps are popular. A BN is often used by default without comparing its performance with other normalization methods. In medical semantic segmentation (3.1b), the target is a specific anatomy, medical device, tumor or functional region etc. A network trained from scratch is common, allowing

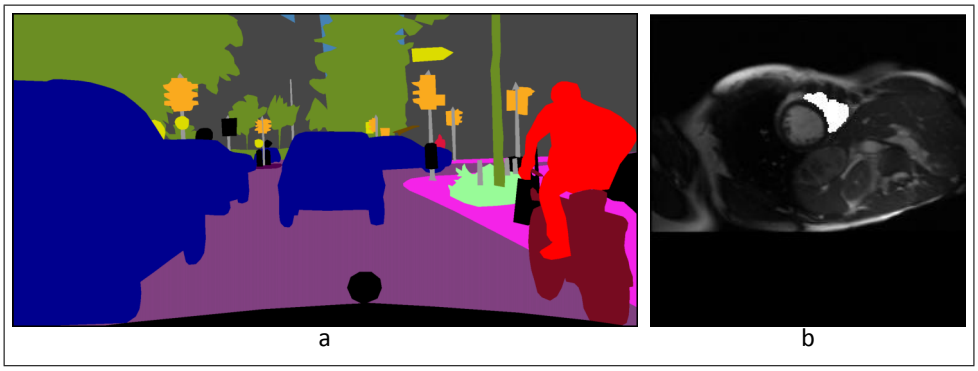


Figure 3.1: (a) semantic segmentation of cars, people, trees, etc. from a natural image [2], (b) semantic segmentation of RV from a MRI image.

exploring different normalization methods.

In this chapter, first, the most widely applied four normalization methods - BN, IN, LN, and GN are reviewed and compared specifically for medical image segmentation. U-Net is selected as the network architecture due to its wide application. Second, a new normalization method is proposed. Recently, Nam *et al.* proposed Batch-Instance Normalization (BIN) [96], which combined BN and IN with a trainable parameter. However, two risks potentially exist: 1) the trainable parameter was restricted in the range of [0, 1] with clip function which is not differentiable at input values of 0 and 1; 2) the combined feature map was no longer with a normal distribution, which is harmful for signal propagation in DCNN. In this chapter, Instance-Layer Normalization (ILN) is proposed to combine IN and LN: 1) sigmoid is used to solve the non-differentiable characteristic of clip function at input values of 0 and 1; 2) an additional GN16 - GN with a group number of 16 is added after the combined feature map to ensure a normal distribution of the combined feature map. A widely-applied and popular network architecture - U-Net [3] is used as the network to validate the proposed ILN on the RV and LV image segmentation. The U-Net details, four traditional normalization methods, the proposed ILN, data collection for the RV, aorta and LV, and the implementation details are introduced in Sec. 3.2. Detailed experiments and comparisons are provided in Sec. 3.3. It is shown that detailed subdivision of the feature map, i.e. GN with a large group number or IN, out-performed other normalization methods in terms of accuracy, despite the fact that BN is

currently widely used. No obvious improvements regarding the convergence speed and lowest converged loss are observed. The proposed ILN outperforms existing normalization methods with noticeable accuracy improvements in most validations in terms of the Dice Similarity Coefficient (DSC). Discussion and conclusion are in Sec. 3.4.

3.2 Methodology

Systematic details about DCNN can be found in [97], while this chapter only focuses on explaining the concepts of data propagation, network architecture and loss function in Sec. 3.2.1. The algorithms of BN, IN, LN and GN are explained in Sec. 3.2.2, 3.2.3, 3.2.4 and 3.2.5 respectively. The proposed ILN is stated in Sec. 3.2.6. The data collection and implementation details are given in Sec. 3.2.7.

3.2.1 Network Details

With an input feature map $F_{\mathrm{N}\times\mathrm{H}\times\mathrm{W}\times\mathrm{C}}$ (the first feature map is the image batch input), N is the batch size, H the height, W the width, C the channel, a trainable convolutional kernel $T_{\mathrm{C}\times\mathrm{K}\times\mathrm{K}}$ moves along the height and width of $F_{\mathrm{N}\times\mathrm{H}\times\mathrm{W}\times\mathrm{C}}$, indicating an output feature map:

$$\hat{F}_{\mathrm{N}\times\mathrm{H}'\times\mathrm{W}'\times 1} = F_{\mathrm{N}\times\mathrm{H}\times\mathrm{W}\times\mathrm{C}} \cdot T_{\mathrm{C}\times\mathrm{K}\times\mathrm{K}} \tag{3.1}$$

where K is the convolutional kernel size, $\mathrm{H}' = \mathrm{H}//\mathrm{S}$, $\mathrm{W}' = \mathrm{W}//\mathrm{S}$, where $//$ is floor division and S is the convolutional stride. When $\mathrm{S} > 1$, the feature spatial dimension decreases after the convolution. When $0 < S < 1$, the feature spatial dimension increases after the convolution. For extracting richer features, multiple $T_{\mathrm{C}\times\mathrm{K}\times\mathrm{K}}$ are trained, resulting in $\hat{F}_{\mathrm{N}\times\mathrm{H}'\times\mathrm{W}'\times\mathrm{C}'}$.

U-Net, which is a widely applied DCNN structure for medical semantic segmentation is used as the network architecture in this chapter, its architecture is shown in Fig. 3.2. It gradually increases the receptive field (the pixels it sees) with max-pooling layers, resulting in decreased spatial dimensions. Then U-Net recovers and increases the spatial dimension with deconvolutional layers.

An increased receptive field is useful for extracting the semantic information. However, the consequent decreased spatial dimension is disadvantageous for

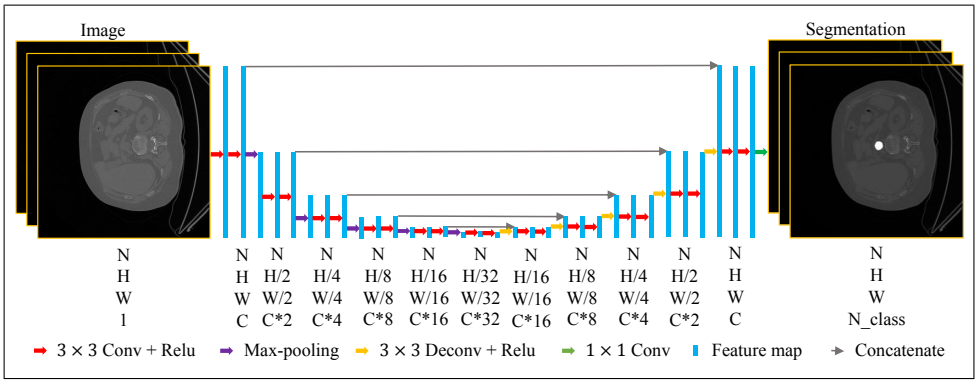


Figure 3.2: The structure of U-Net used in this chapter, Conv - convolution, Deconv - deconvolution.

spatial information. Skip connections are used to concatenate the feature maps from shallow layers and deep layers to combine the semantic and spatial information.

All the convolutional and deconvolutional layers are followed with a ReLU activation, except the 1×1 convolutional layer at the last which predicts the class probability. Softmax is used to transform the final feature map into probabilities and cross-entropy is used as the loss function:

$$loss(p, y) = \begin{cases} -log(p) & \text{if } y = 1.0 \\ -log(1.0 - p) & \text{if } y = 0.0 \end{cases} \tag{3.2}$$

where y is the ground truth, p is the prediction probability. Stochastic Gradient Descent (SGD) is adopted to train the $T_{\mathrm{C\times K\times K\times C'}}$ to obtain a minimum loss. When the distribution of $F_{\mathrm{N\times H\times W\times C}}$ changes, $T_{\mathrm{C\times K\times K\times C'}}$ is influenced and trained to fit this distribution change, resulting in interval covariate shift which decreases both the training speed and accuracy.

3.2.2 Batch Normalization (BN)

BN is the first proposed algorithm for solving the interval covariate shift. It normalizes $F_{\mathrm{N\times H\times W\times C}}$ to achieve a mean of 0.0 and a variance of 1.0 while maintains the representation capability of a DCNN with two more trainable parameters - γ, β.

In BN [81], the mean and variance are calculated along each channel:

$$\mu_c = \frac{1}{\mathrm{N} \times \mathrm{H} \times \mathrm{W}} \sum_{n=1}^{\mathrm{N}} \sum_{h=1}^{\mathrm{H}} \sum_{w=1}^{\mathrm{W}} f_{n,h,w} \tag{3.3}$$

$$\delta_c^2 = \frac{1}{\mathrm{N} \times \mathrm{H} \times \mathrm{W}} \sum_{n=1}^{\mathrm{N}} \sum_{h=1}^{\mathrm{H}} \sum_{w=1}^{\mathrm{W}} (f_{n,h,w} - \mu_c)^2 \tag{3.4}$$

The feature map is normalized by:

$$\hat{f}_{n,h,w} = \frac{f_{n,h,w} - \mu_c}{\sqrt{\delta_c^2 + \epsilon}} \tag{3.5}$$

where ϵ is a small value used to increase the division stability. After this normalization, $\hat{f}_{n,h,w}$ is always with the mean of 0.0 and the variance of 1.0, which limits the DCNN representation capacity [81]. Additional trainable parameters γ_c and β_c are added to each channel to recover the representation power:

$$f'_{n,h,w} = \gamma_c \hat{f}_{n,h,w} + \beta_c \tag{3.6}$$

BN is applied after the convolution and before the activation. There are two ways of applying BN during the inference: 1) use the moving average mean and variance in the training stage to normalize the test feature map, as recommended in [81]; 2) use the mean and variance in the test stage to normalize the test feature map, as recommended in [98]. In this chapter, both ways are explored and the optimal one is used for the comparison with other normalization methods.

3.2.3 Instance Normalization (IN)

In IN [98], the mean and variance are calculated for each channel and each instance of the batch:

$$\mu_{n,c} = \frac{1}{\mathrm{H} \times \mathrm{W}} \sum_{h=1}^{\mathrm{H}} \sum_{w=1}^{\mathrm{W}} f_{h,w} \tag{3.7}$$

$$\delta_{n,c}^2 = \frac{1}{\mathrm{H} \times \mathrm{W}} \sum_{h=1}^{\mathrm{H}} \sum_{w=1}^{\mathrm{W}} (f_{h,w} - \mu_{n,c})^2 \tag{3.8}$$

The feature map is normalized by:

$$\hat{f}_{h,w} = \frac{f_{h,w} - \mu_{n,c}}{\sqrt{\delta_{n,c}^2 + \epsilon}} \tag{3.9}$$

3.2.4 Layer Normalization (LN)

In LN [86], the mean and variance are calculated along each instance of the batch:

$$\mu_n = \frac{1}{\mathrm{H} \times \mathrm{W} \times \mathrm{C}} \sum_{h=1}^{\mathrm{H}} \sum_{w=1}^{\mathrm{W}} \sum_{c=1}^{\mathrm{C}} f_{h,w,c} \tag{3.10}$$

$$\delta_n^2 = \frac{1}{\mathrm{H} \times \mathrm{W} \times \mathrm{C}} \sum_{h=1}^{\mathrm{H}} \sum_{w=1}^{\mathrm{W}} \sum_{c=1}^{\mathrm{C}} (f_{h,w,c} - \mu_n)^2 \tag{3.11}$$

The feature map is normalized by:

$$\hat{f}_{h,w,c} = \frac{f_{h,w,c} - \mu_n}{\sqrt{\delta_n^2 + \epsilon}} \tag{3.12}$$

3.2.5 Group Normalization (GN)

In GN [87], the mean and variance are calculated along each instance of the batch and multiple instances of the channel. The difference between GN and IN/LN is that a group of channels $\mathrm{M} = C//G$ are grouped together for the normalization, G is the number of group, M is the number of channel per group:

$$\mu_{n,g} = \frac{1}{\mathrm{H} \times \mathrm{W} \times \mathrm{M}} \sum_{h=1}^{\mathrm{H}} \sum_{w=1}^{\mathrm{W}} \sum_{m=(g-1)\cdot \mathrm{M}+1}^{\mathrm{g}\cdot \mathrm{M}} f_{h,w,m} \tag{3.13}$$

$$\delta_{n,g}^2 = \frac{1}{\mathrm{H} \times \mathrm{W} \times \mathrm{M}} \sum_{h=1}^{\mathrm{H}} \sum_{w=1}^{\mathrm{W}} \sum_{m=(g-1)\cdot \mathrm{M}+1}^{\mathrm{g}\cdot \mathrm{M}} (f_{h,w,m} - \mu_{n,g})^2 \tag{3.14}$$

The feature map is normalized by:

$$\hat{f}_{h,w,m} = \frac{f_{h,w,m} - \mu_{n,g}}{\sqrt{\delta_{n,g}^2 + \epsilon}} \tag{3.15}$$

In this chapter, GN with different numbers of group is seen as different normalization methods and are compared.

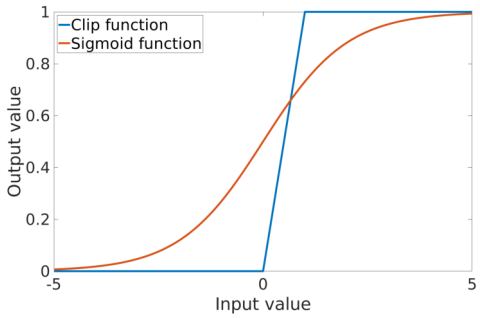


Figure 3.3: The curves of clip and sigmoid function.

In IN, LN, and GN, additional parameters are also used for recovering the DCNN representation ability. Multiple ways of adding γ and β may exist. In this chapter, we follow [87] and add parameters for each feature channel, which results in $2C$ parameters for each feature map.

3.2.6 Instance-Layer Normalization

A trainable parameter ρ is added to linearly weight the normalized feature map of IN $\hat{\mathbf{F}}^{\mathrm{I}}$ and the normalized feature map of LN $\hat{\mathbf{F}}^{\mathrm{L}}$. In the original BIN [96], ρ was clipped to be in the range of $[0, 1]$ with a clip function, as shown in Fig. 3.3.
However, clip function is not differentiable at input values of 0 and 1. In this chapter, sigmoid function $Sigmoid(x) = 1/(e^{-x} + 1)$ which is differentiable everywhere is applied to solve this potential issue:

$$\hat{\mathbf{F}}^{\mathrm{IL}} = Sigmoid(\rho) \cdot \hat{\mathbf{F}}^{\mathrm{I}} + (1 - Sigmoid(\rho)) \cdot \hat{\mathbf{F}}^{\mathrm{L}} \tag{3.16}$$

An additional potential issue in the original BIN is that the combined $\hat{\mathbf{F}}^{\mathrm{IL}}$ is no longer with a mean of 0.0 and a variance of 1.0, this non-normal distribution may be harmful for signal propagation in DCNN. In this chapter, we solve this issue with applying an additional GN16 on the weighted $\hat{\mathbf{F}}^{\mathrm{IL}}$:

$$\mu_{n,g} = \frac{1}{\mathrm{H} \times \mathrm{W} \times \mathrm{M}} \sum_{h=1}^{\mathrm{H}} \sum_{w=1}^{\mathrm{W}} \sum_{m=(g-1)\cdot \mathrm{M}+1}^{\mathrm{g}\cdot \mathrm{M}} \hat{f}^{\mathrm{IL}}_{n,h,w,m}, \mathrm{M} = C//16 \tag{3.17}$$

$$\delta_{n,g}^2 = \frac{1}{\mathrm{H} \times \mathrm{W} \times \mathrm{M}} \sum_{h=1}^{\mathrm{H}} \sum_{w=1}^{\mathrm{W}} \sum_{m=(g-1)\cdot \mathrm{M}+1}^{\mathrm{g}\cdot \mathrm{M}} (\hat{f}_{n,h,w,m}^{\mathrm{IL}} - \mu_{n,g})^2, \mathrm{M} = C//16 \tag{3.18}$$

where M is the number of channel in each feature group, // is floor division, $g \in [1, 16]$. Following BN [81], additional parameters γ and β are added to preserve the DCNN representation ability $f'^{\mathrm{ILN}}_{n,h,w,c} = \gamma_c \hat{f}^{\mathrm{ILN}}_{n,h,w,c} + \beta_c$.

3.2.7 Data Collection and Implementation Details

Three datasets: RV scanned with MRI, with 256×256 image size, aorta scanned with CT, with 512×512 image size, and LV scanned with MRI, with 256×256 image size are used for the validation. As the main purpose of this chapter is to improve the automation of 3D shape instantiation, where the input image is usually 2D, we mainly focus on 2D medical image segmentation. The 3D CT image is sliced into multiple 2D images for the validation.

37 RV scans [99] were acquired from a 1.5T MRI scanner (Sonata, Siemens, Erlangen, Germany), from both the asymptomatic and Hypertrophic Cardiomyopathy (HCM) subjects, from the atrioventricular ring to the apex, with a $10mm$ slice gap, a $1.5 - 2mm$ pixel spacing, and $19 - 25$ time frames for the cardiac cycle. 6082 images were collected in total. All images were labelled by one expert with Analyze (AnalyzeDirect, Inc, Overland Park, KS, USA) and were augmented by rotation from $-30°$ to $30°$ with $10°$ as the interval. The 37 subjects were split randomly into three groups for three-fold cross validation, with 12, 12, and 13 subjects for each group respectively.

20 aortic CT scans were acquired from the VISCERAL data set [100]. 4631 images were collected in total and were augmented by rotation from $-40°$ to $40°$ with $10°$ as the interval. The 20 subjects were split randomly into three groups for three-fold cross validation, with 7, 7, and 6 subjects for each group respectively.

45 LV MRI scans were acquired from the SunnyBrook data set [101]. 805 images were collected in total and were augmented by rotation from $-60°$ to $60°$ with $2°$ as the interval. The 45 subjects were split randomly into three groups for three-fold cross validation, with 15 subjects for each group.

The maximum and minimum intensity value of all subjects are used to

re-scale the image intensity to a maximum value of 1.0 and a minimum value of 0.0. In the cross validation, one group was used as the testing data while the other two groups were used for the training data. Due to the limitation of available images, no validation dataset was split or used.

The kernel size of convolutional and deconvolutional layers is 3, except the last convolutional layer whose kernel size is 1. The pool size for max-pooling is 2. The stride of deconvolutional layers is 2. The number of channel in the first feature map - C is 16. The momentum is 0.9. Step-wise learning rate schedule was used, as it allows careful and manual adjustment of the learning rate. Training with two epochs usually achieved the lowest converged loss and was used in our experiments. Three to five initial learning rates were tested for each experiment and the one with the best performance in terms of the accuracy is reported in this chapter. Several step-wise methods were explored, i.e. dividing the learning rate by 5 or 10 every half or one epoch. Dividing the learning rate by 5 at the second epoch showed optimal performance and was used. This is also consistent with the learning rate schedule in [102]. Momentum SGD was used as the optimizer with the momentum set as 0.9. Weights were initialized with a truncated normal distribution with the *stddev* as $2/(3^2 \times \mathrm{C})$, where C is the channel number. Biases were initialized as 0.1. ρ was initialized as 0.5. This training strategy was determined by testing multiple training strategies on the vanilla U-Net [3], and was applied to all other CNNs.

The largest explored batch size for the RV, aorta, and LV are 32, 16, and 32 respectively in this chapter. This is determined by the GPU memory. As the RV and LV are with smaller image size and consume less GPU memories, the largest batch size the GPU can hold is larger.

DSC was calculated as the evaluation metric:

$$DSC = 2 \cdot \frac{|\mathrm{Y} \cap \mathrm{P}|}{|\mathrm{Y} + \mathrm{P}|} \tag{3.19}$$

where Y is the ground truth and P is the prediction. As only two classes exist, only the foreground DSC is shown in this chapter.

3.3 Results

As stated in Sec. 3.2.2, there are two ways of applying BN during the inference. Both of them are evaluated in Sec. 3.3.1. The optimal results are selected to represent the BN performance and are used for later comparisons between different normalization methods in Sec. 3.3.2. It is known that training the same model multiple times would result in slightly different results [97]. In this chapter, the same phenomenon exists and the corresponding validations are in Sec. 3.3.3. Segmentation examples are illustrated in Sec. 3.3.4.

To evaluate the advantage of using the sigmoid function over the clip function (in original BIN [96]), three comparison experiments were set up: 1) using clip function with one trainable parameter $Clip(\rho)_0^1$ for IN feature map while the parameter for LN feature map is $1 - Clip(\rho)_0^1$; 2) using sigmoid function with one trainable parameter $Sigmoid(\rho)$ for IN feature map while the parameter for LN feature map is $1 - Sigmoid(\rho)$; 3) using softmax function with two trainable parameters $Softmax(\rho_1, \rho_2)$ for IN and LN feature map respectively. Comparison results are shown in Sec. 3.3.5.

To evaluate the advantage of adding GN16 after the combined feature map, two comparison experiments with or without GN16 are conducted. Results are shown in Sec. 3.3.6. As GN16 performed similarly to IN [4], no normalization, IN, LN, GN4 are chosen as the baseline to validate the performance of the proposed ILN, as presented in details in Sec. 3.3.7. The training curves of ρ at eight randomly-selected layers are shown in Sec. 3.3.8.

In the following paragraphs, RV-1 refers to the first fold of cross validation (using the first group as the testing while using the second and third group as the training) for the RV, this name also applies to RV-2, RV-3, Aorta-1, Aorta-2, Aorta-3, LV-1, LV-2, LV-3. BS refers to batch size. S/M/L refers to the small/medium/large batch size, indicating batch size of (1, 16, 32), (1, 8, 16), (1, 16, 32) for the RV, aorta, LV respectively. LR refers to the learning rate. GN4, GN8, GN16 refers to the group normalization with group number of 4, 8, 16.

3.3.1 Using BN during Inference

TestI (using the mean and variance of the testing feature map to normalize the testing feature map) and TrainI (using the moving average mean and variance of the training feature map to normalize the testing feature map)

are validated on the three datasets with small, medium and large batch size. The mean±std DSCs are shown in Tab. 3.1. It is evident that TestI outperformed TrainI in most experiments, except those for the CT images (aorta) with medium and large batch sizes. However, this conclusion needs more systematic experiments to validate its generalization. In this chapter, we use the optimal DSC achieved by either TestI or TrainI to represent the BN performance.

3.3.2 Comparison between Normalization Methods

To compare different normalization methods, U-Nets are trained with three datasets (RV, Aorta, LV), three cross validations, seven normalization methods (None, BN, GN4, GN8, GN16, IN, LN) and three batch sizes (small, medium large). The mean±std DSCs achieved are shown in Tab. 3.2, 3.3, and 3.4 respectively. It can be seen that for most experiments, GN16 or IN achieves the highest accuracy. For most exceptions, GN16 or IN could achieve similar accuracy to the highest value. As the number of channel in the first feature map - C in this chapter is 16, GN16 is similar to IN which divides the feature map into very small groups. It could be concluded that detailed subdivision of the feature map during normalization potentially leads to higher accuracy.

Adding normalization increases the running time of each iteration. In general, BN is faster than IN, LN and GN. There is no obvious trend regarding the LR. It is worth noting that training with small batch size outperformed that with large batch size. In the following paragraphs, we select experiments with small batch size for showing the convergence and patient errors.

Three experiments - RV-2, Aorta-1, LV-1 are selected randomly to show the loss convergence during the training in Fig. 3.4. Unlike the report in [81] where the DCNN was trained 14 times faster, no obvious improvements on the convergence speed is observed. The optimal normalization methods for RV-2, Aorta-1 and LV-1 are GN16, GN16, and IN respectively. Obvious lower loss is achieved by GN16 for the RV-2 test while this phenomenon is not obvious for the Aorta-1 and LV-1 test. We think the validations are not enough to make a general conclusion.

Table 3.1: Mean±std DSCs of segmenting the RV, aorta, and LV with BN; TestI and TrainI are used during inference; "-" means an optimal LR could not be found for that case; highest DSC in bold and blue.

BS	Test	Mean±std DSCs		Optimal LR	
		TestI	TrainI	TestI	TrainI
S	RV-1	**0.7133±0.2693**	0.6895±0.2760	0.5	0.05
	RV-2	**0.7139±0.2859**	0.6579±0.3306	1.0	0.1
	RV-3	**0.6745±0.3029**	0.6070±0.3390	1.0	0.05
	aorta-1	**0.8368±0.1405**	0.8249±0.1773	0.5	0.5
	aorta-2	0.7689±0.2178	**0.7832±0.2072**	0.5	0.1
	aorta-3	**0.8060±0.2294**	0.7707±0.2707	1.5	0.1
	LV-1	**0.9240±0.0808**	0.9020±0.1110	0.5	0.1
	LV-2	**0.8864±0.1391**	0.8686±0.1999	1.0	0.1
	LV-3	**0.8479±0.1643**	0.8063±0.2300	1.0	0.1
M	RV-1	**0.7025±0.2796**	0.6533±0.2930	1.0	0.1
	RV-2	**0.6833±0.3091**	0.6131±0.3316	1.0	0.1
	RV-3	**0.6415±0.3275**	0.5529±0.3678	1.0	0.05
	aorta-1	0.7804±0.2061	**0.8036±0.1714**	1.5	0.1
	aorta-2	0.7276±0.2525	**0.7726±0.2009**	1.0	0.5
	aorta-3	0.7408±0.2798	**0.7787±0.2453**	1.0	0.5
	LV-1	**0.9054±0.0864**	0.8384±0.2111	0.5	0.1
	LV-2	0.8431±0.1769	**0.8567±0.1815**	0.5	0.5
	LV-3	**0.7899±0.2186**	0.7085±0.2701	1.0	0.05
L	RV-1	**0.6794±0.2847**	0.6556±0.2873	1.0	0.5
	RV-2	**0.6670±0.3066**	0.6283±0.3108	1.0	0.5
	RV-3	**0.6380±0.3267**	0.5838±0.3377	0.5	0.1
	aorta-1	0.7668±0.2070	**0.7782±0.1842**	1.0	0.1
	aorta-2	0.7200±0.2423	**0.7458±0.2243**	1.5	0.1
	aorta-3	0.6200±0.3557	**0.7449±0.2594**	1.5	0.1
	LV-1	**0.8868±0.1432**	-	0.5	-
	LV-2	**0.7892±0.2325**	0.7085±0.1887	1.0	0.1
	LV-3	**0.7677±0.2151**	0.7076±0.2919	0.5	0.5

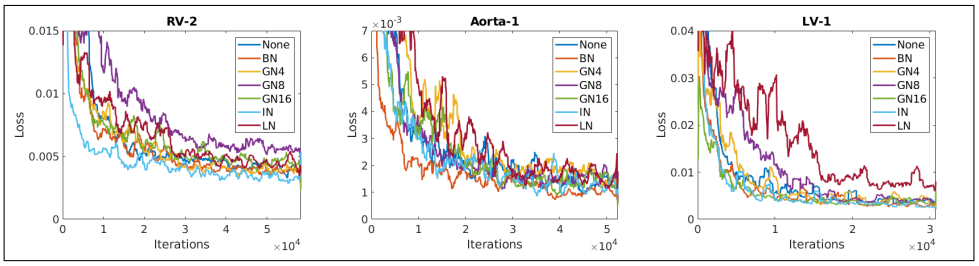


Figure 3.4: The training loss of U-Net with no normalization, BN, GN4, GN8, GN16, IN, and LN for RV-2 (left), aorta-1(middle), and LV-1(right) segmentation, the losses were recorded every 20 iterations, smoothed by a moving average window of 31, and truncated for clear plot.

Table 3.2: Mean±std DSCs of the RV segmentation with different normalization methods (highest DSC in bold and blue).

Test	Method	Mean±std DSCs			Optimal LR			Time (Seconds/20 Iterations)		
		S-BS	M-BS	L-BS	S-BS	M-BS	L-BS	S-BS	M-BS	L-BS
RV-1	None	0.6944±0.2428	0.6654±0.2838	0.6306±0.2647	0.5	0.1	1.0	0.5	2.6	4.8
	BN	0.7133±0.2693	**0.7025±0.2796**	**0.6794±0.2847**	0.5	1.0	1.0	0.88	3.6	6.6
	GN4	0.7023±0.3078	0.6887±0.2734	0.6791±0.2879	0.5	0.5	1.0	1.1	5.3	9.4
	GN8	0.6952±0.2932	0.6744±0.3033	0.6616±0.3098	1.5	0.5	1.5	1.1	5.2	9.3
	GN16	0.6989±0.2964	0.6755±0.2966	0.6732±0.2991	1.0	0.5	1.0	1.1	5.3	9.2
	IN	**0.7346±0.2352**	0.6856±0.2927	0.6662±0.3121	1.5	1.5	1.0	1.0	4.0	7.3
	LN	0.6906±0.2954	0.6928±0.2701	0.6606±0.2876	0.1	0.5	0.5	1.0	4.1	7.5
RV-2	None	0.6452±0.3297	0.6150±0.3130	0.5951±0.3339	0.1	0.5	0.1	0.5	2.6	4.8
	BN	0.7139±0.2859	0.6833±0.3091	0.6670±0.3066	1.0	1.0	1.0	0.88	3.6	6.6
	GN4	0.6795±0.3088	0.6238±0.3424	0.6439±0.3136	0.1	1.0	0.1	1.1	5.3	9.4
	GN8	0.7155±0.2752	0.6258±0.3503	0.6386±0.3253	1.0	0.1	1.0	1.1	5.2	9.3
	GN16	**0.7291±0.2720**	**0.6835±0.3091**	**0.6785±0.3027**	0.5	0.5	1.5	1.1	5.3	9.2
	IN	0.7022±0.3002	0.6565±0.3303	0.6382±0.3448	0.1	1.0	1.0	1.0	4.0	7.3
	LN	0.6789±0.3123	0.6376±0.3175	0.6418±0.3151	0.5	1.0	0.1	1.0	4.1	7.5
RV-3	None	0.6117±0.3455	0.5715±0.3490	0.5629±0.3358	0.05	0.1	0.05	0.5	2.6	4.8
	BN	0.6745±0.3029	0.6415±0.3275	0.6380±0.3267	1.0	1.0	0.5	0.88	3.6	6.6
	GN4	0.6548±0.2963	0.5931±0.3520	0.5850±0.3474	0.5	0.5	0.5	1.1	5.3	9.4
	GN8	0.6366±0.3354	0.6092±0.3459	0.5867±0.3495	1.0	1.0	0.5	1.1	5.2	9.3
	GN16	0.6735±0.3080	**0.7153±0.2683**	**0.6532±0.3161**	1.0	1.5	0.5	1.1	5.3	9.2
	IN	**0.7145±0.2732**	0.6317±0.3282	0.6158±0.3391	1.0	0.5	1.0	1.0	4.0	7.3
	LN	0.6118±0.3366	0.6292±0.3348	0.6025±0.3316	0.1	1.0	0.05	1.0	4.1	7.5

Table 3.3: Mean±std DSCs of the aorta segmentation with different normalization methods (highest DSC in bold and blue).

Test	Method	Mean±std DSCs			Optimal LR			Time (Seconds/20 Iterations)		
		S-BS	M-BS	L-BS	S-BS	M-BS	L-BS	S-BS	M-BS	L-BS
Aorta-1	None	0.8165±0.1843	0.7965±0.1689	0.7932±0.2030	0.05	0.05	0.5	1.0	7.9	15.0
	BN	0.8368±0.1405	0.8036±0.1714	0.7782±0.1842	0.5	1.5	1.0	1.5	6.5	12.5
	GN4	0.8310±0.1556	0.7928±0.1783	0.7615±0.2122	1.5	0.1	1.0	2.2	10.5	18.5
	GN8	0.8314±0.1620	0.8223±0.1745	0.8065±0.1496	1.0	1.0	1.0	2.0	10.7	19.0
	GN16	**0.8412±0.1483**	0.8207±0.1613	0.8155±0.1431	0.5	1.5	1.0	1.7	11.5	19.2
	IN	0.8320±0.1518	**0.8273±0.1313**	**0.8174±0.1478**	1.5	1.0	1.0	1.7	7.1	13.7
	LN	0.8193±0.1913	0.7292±0.2391	0.7038±0.2930	1.0	1.0	0.5	1.6	7.4	14.3
Aorta-2	None	0.7938±0.2081	0.7692±0.2175	0.7532±0.2480	0.05	0.1	0.5	1.0	7.9	15.0
	BN	0.7832±0.2072	0.7726±0.2009	0.7458±0.2243	0.5	1.0	1.5	1.5	6.5	12.5
	GN4	0.7863±0.2090	0.7681±0.2201	0.6923±0.2935	1.0	1.0	0.1	2.2	10.5	18.5
	GN8	**0.8099±0.1724**	0.7588±0.2425	**0.7623±0.2275**	0.5	0.5	0.5	2.0	10.7	19.0
	GN16	0.7916±0.1974	**0.7891±0.1934**	0.7084±0.2587	0.5	1.5	0.5	1.7	11.5	19.2
	IN	0.7734±0.2166	0.7529±0.2278	0.7249±0.2663	1.0	1.0	1.5	1.7	7.1	13.7
	LN	0.7793±0.2213	0.7270±0.2623	0.7053±0.2339	1.0	1.0	1.5	1.6	7.4	14.3
Aorta-3	None	0.7718±0.2712	0.7611±0.2821	0.7281±0.2798	0.05	0.5	0.5	1.0	7.9	15.0
	BN	0.8060±0.2294	0.7787±0.2453	0.7449±0.2594	1.5	1.0	1.5	1.5	6.5	12.5
	GN4	0.7917±0.2654	0.7423±0.2810	0.7217±0.3010	0.5	0.5	0.1	2.2	10.5	18.5
	GN8	0.8059±0.2490	0.7715±0.2632	**0.7774±0.2420**	0.5	1.0	1.0	2.0	10.7	19.0
	GN16	**0.8221±0.2055**	0.7721±0.2523	0.7664±0.2724	0.5	1.5	1.5	1.7	11.5	19.2
	IN	0.7942±0.2284	**0.7895±0.2355**	0.7600±0.2443	0.5	1.5	0.5	1.7	7.1	13.7
	LN	0.7586±0.2741	0.7122±0.3068	0.6992±0.3068	1.0	0.5	1.0	1.6	7.4	14.3

Table 3.4: Mean±std DSCs of the LV segmentation with different normalization methods (highest DSC in bold and blue).

Test	Method	Mean±std DSCs			Optimal LR			Time (Seconds/20 Iterations)		
		S-BS	M-BS	L-BS	S-BS	M-BS	L-BS	S-BS	M-BS	L-BS
LV-1	None	0.9240±0.0678	0.8344±0.2036	0.5277±0.3078	0.1	0.1	0.5	0.5	2.6	4.8
	BN	0.9240±0.0808	0.9054±0.0864	0.8868±0.1432	0.5	0.5	0.5	0.7	3.0	5.5
	GN4	0.9229±0.0979	0.8684±0.1552	0.8113±0.2102	0.5	1.5	0.1	1.1	5.3	9.4
	GN8	0.9233±0.0864	0.8876±0.1346	0.7582±0.2196	1.0	1.0	0.5	1.1	5.1	9.5
	GN16	0.9306±0.0560	**0.9233±0.0672**	**0.9006±0.1058**	0.1	1.0	1.0	1.0	5.1	9.1
	IN	**0.9313±0.0657**	0.9099±0.1064	0.8982±0.1230	0.5	1.0	1.0	1.0	3.9	7.3
	LN	0.8426±0.1775	0.8552±0.1719	0.8531±0.1592	1.0	1.0	0.5	1.0	4.0	7.5
LV-2	None	0.8874±0.1592	0.7287±0.2496	0.5219±0.2830	0.1	0.05	0.1	0.5	2.6	4.8
	BN	0.8864±0.1391	**0.8567±0.1815**	0.7892±0.2325	1.0	0.5	0.5	0.7	3.0	5.5
	GN4	**0.8931±0.1352**	0.8050±0.2018	0.7279±0.2128	0.5	0.5	1.0	1.1	5.3	9.4
	GN8	0.8844±0.1161	0.8430±0.1439	0.7815±0.2229	1.0	0.1	0.1	1.1	5.1	9.5
	GN16	0.8915±0.1288	0.8479±0.1736	**0.8188±0.1608**	1.0	0.5	1.0	1.0	5.1	9.1
	IN	0.8894±0.1459	0.8013±0.2428	0.7973±0.2138	0.5	0.5	1.0	1.0	3.9	7.3
	LN	0.7806±0.2038	0.8389±0.1925	0.7059±0.1642	0.5	0.5	1.0	1.0	4.0	7.5
LV-3	None	0.8081±0.2345	0.6956±0.2828	0.6526±0.2932	0.1	0.05	0.5	0.5	2.6	4.8
	BN	**0.8479±0.1643**	0.7899±0.2186	0.7677±0.2151	1.0	1.0	0.5	0.7	3.0	5.5
	GN4	0.8123±0.2355	0.7288±0.2526	0.7034±0.2515	0.5	1.5	0.1	1.1	5.3	9.4
	GN8	0.8116±0.2297	0.7620±0.2508	0.7756±0.2422	0.1	1.5	1.0	1.1	5.1	9.5
	GN16	0.8447±0.1882	**0.8255±0.1982**	**0.8013±0.2097**	0.5	0.5	0.5	1.0	5.1	9.1
	IN	0.8401±0.1856	0.8044±0.2107	0.7660±0.2404	1.0	1.5	1.5	1.0	3.9	7.3
	LN	0.7979±0.2437	0.7674±0.2555	0.6973±0.2699	0.1	0.1	1.0	1.0	4.0	7.5

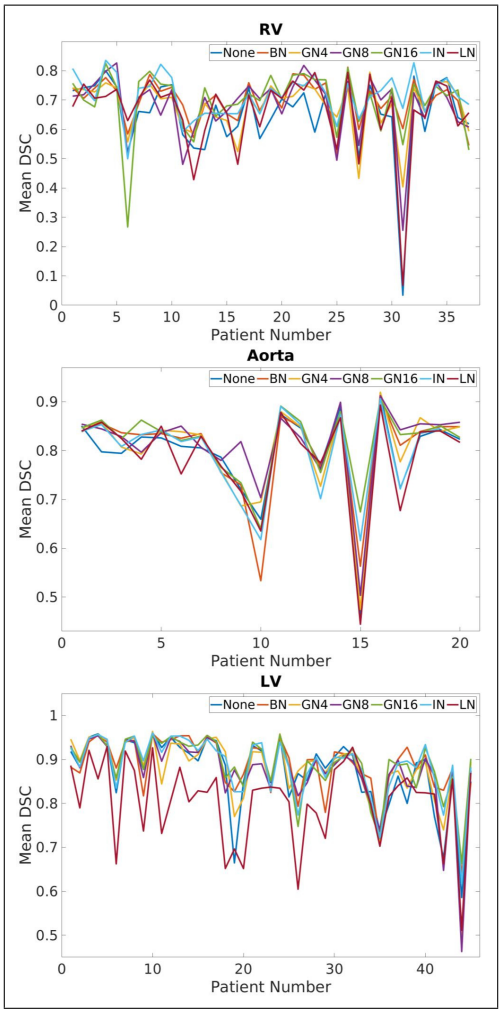


Figure 3.5: mean DSC for each patient for the RV (top), Aorta (middle), LV (bottom) segmented by U-Net with None, BN, GN4, GN8, GN16, IN, LN normalization methods.

We further show the mean DSC for each patient in Fig. 3.5. Due to the complex parameter setting when acquiring CT or MRI images, the image intensity distributions are always different between patients. Hence, internal covariate shift is produced and some patients are with low segmentation accuracy. In Fig. 3.5, the main accuracy improvements achieved by normalization appear at those patients with low initial accuracy, i.e. patient 31 for RV, patient 15 for Aorta, patient 44 for LV. This proves that the accuracy improvement with normalization methods come from its improved generalization ability.

Table 3.5: Mean and std of the mean DSC when training the same model in six times.

Normalization	Experiment	Mean DSC	Std DSC (6 runs)
None	LV-1	0.8916	0.0248
BN	LV-3	0.8274	0.0129
GN4	RV-3	0.6436	0.0222
GN8	Aorta-2	0.7924	0.0097
GN16	Aorta-3	0.8091	0.0148
IN	RV-1	0.7035	0.0226
LN	LV-2	0.8189	0.0562

3.3.3 Multiple Runs

It is known that training the same model multiple times indicate different results - 2% variance as stated in [97]. In medical semantic segmentation, this exists as well. We select randomly one experiment for each normalization method and train it additionally five times. The mean and std of the mean DSCs of different trainings are shown in Tab. 3.5. We can see that the std between multiple runs is very large, sometimes can be even larger than the accuracy improvement. In this chapter, all the results shown above are trained only once, this is fair for each method. However, running the experiments in multiple times may indicate different results.

3.3.4 Segmentation Results

The 3D aortic shape reconstructed from the aortic segmentation is shown in Fig. 3.6b, which could be registered to navigate the Magellan (Hansen Medical, CA, USA) robotic system. As the RV and LV are MRI images with 10mm slice gap, 3D reconstruction could not be extracted. The 2D RV and LV segmentation results are shown in Fig. 3.6a and 3.6c respectively, which could be used to instantiate 3D shapes and hence to navigate cardiac robotic interventions.

3.3.5 Sigmoid vs. Clip vs. Softmax Function

The mean±std segmentation DSCs of using clip, sigmoid and softmax function to combine the IN and LN feature map are shown in Tab. 3.6. We can see that sigmoid function achieves the highest DSC for most cross validations,

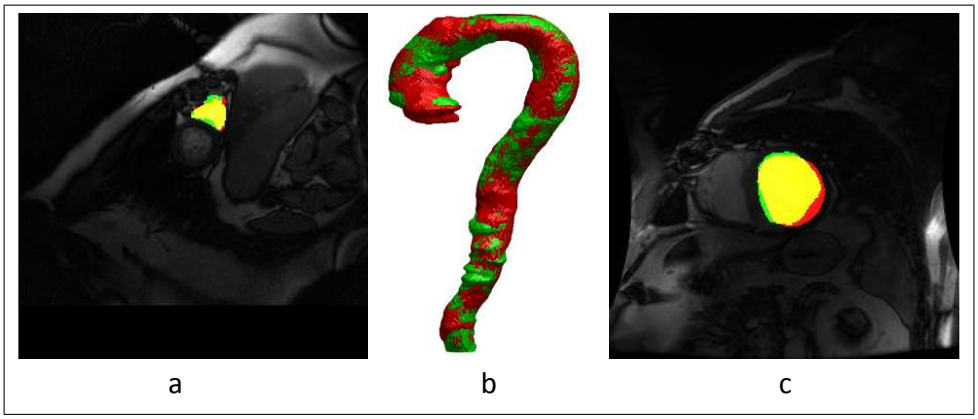


Figure 3.6: Segmentation examples of the RV (a), aorta (b) and LV (c). red - the ground truth, green - the segmentation results, yellow - the overlap between the ground truth and the segmentation results.

Table 3.6: Mean±std segmentation DSCs of using clip, sigmoid and softmax function to combine the feature map of IN and LN, highest DSCs are in blue and bold colour.

Method	RV-1	RV-2	RV-3
Clip	**0.702±0.295**	0.707±0.299	0.666±0.319
Sigmoid	0.692±0.304	**0.724±0.284**	**0.675±0.301**
Softmax	0.688±0.290	0.720±0.279	0.664±0.323
Method	LV-1	LV-2	LV-3
Clip	0.900±0.099	0.864±0.184	0.804±0.246
Sigmoid	**0.903±0.118**	**0.888±0.135**	**0.828±0.189**
Softmax	0.895±0.151	0.866±0.153	0.827±0.228

except RV-1 experiment, which proves the effectiveness of the proposed method in this chapter - replacing the clip function in original BIN [96] with sigmoid function.

3.3.6 With or Without GN16

The mean±std segmentation DSCs of adding or not adding GN16 after the combined feature map of IN and LN are shown in Tab. 3.7. We can see that, the method with adding GN16 achieves the highest DSC for most cross validations, except LV-3 experiment. This result proves the effectiveness of adding GN16 after the combined feature map and also proves the importance of maintaining the normal distribution of feature maps.

Table 3.7: Mean±std segmentation DSCs of adding or not adding GN16 after the combined feature map of IN and LN, highest DSCs are in blue and bold colour.

Method	RV-1	RV-2	RV-3
No	0.692±0.304	0.724±0.284	0.675±0.301
Yes	**0.714±0.290**	**0.737±0.267**	**0.680±0.305**
Method	LV-1	LV-2	LV-3
No	0.903±0.118	0.888±0.135	**0.828±0.189**
Yes	**0.919±0.098**	**0.893±0.127**	0.827±0.211

Table 3.8: Mean±std segmentation DSCs of using no normalization, IN, LN, GN4, and the proposed ILN with the U-Net framework, highest DSCs are in blue and bold colour.

Method	RV-1	RV-2	RV-3
None	0.688±0.296	0.678±0.318	0.661±0.323
IN	0.709±0.266	0.715±0.278	0.655±0.327
LN	0.702±0.287	0.718±0.270	0.662±0.309
GN4	0.679±0.303	0.701±0.291	0.671±0.309
ILN	**0.714±0.290**	**0.737±0.267**	**0.680±0.305**
Method	LV-1	LV-2	LV-3
None	0.899±0.134	0.872±0.167	0.784±0.280
IN	0.905±0.114	0.876±0.131	**0.836±0.207**
LN	0.898±0.120	0.858±0.187	0.793±0.262
GN4	0.908±0.113	0.841±0.196	0.800±0.255
ILN	**0.919±0.098**	**0.893±0.127**	0.827±0.211

3.3.7 Comparison of ILN to Other Methods

The mean±std segmentation DSCs of using no normalization, IN, LN, GN4, and the proposed ILN with the U-Net framework are shown in Tab. 3.8. We can see that, except the LV-3 experiment, the proposed ILN outperforms all other traditional methods with considerable accuracy improvements. This result proves the effectiveness of the proposed ILN in medical image segmentation.

3.3.8 Training Curves of ρ

In order to show that the proposed ILN does achieve a weighted normalization between IN and LN, the ρ training curves of eight layers were selected randomly from LV-1 experiment to be shown in Fig. 3.7. We can see that

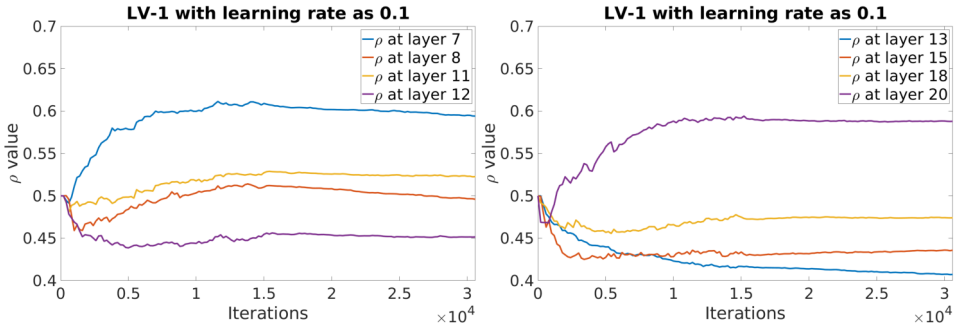


Figure 3.7: The training curves of eight ρ selected randomly from the 22 layers in U-Net.

ρ was trained to be different values and the proposed ILN achieved diverse normalization at different layers. As the ground truth of ρ is not known and it is impossible to judge the curve correctness, a comparison regarding the ρ training curves of ILN and BIN is not illustrated.

The CPU used is Intel Xeon(R) E5-1650 v4@3.60GHz×12. The GPU used is Nvidia Titan XP. Comparing ILN to IN, the parameter number increases 22, as one parameter is added to each layer. The training time for 200 iterations increases from 34.8s to 36.5s due to the additional GN16 calculation.

3.4 Discussion and Conclusion

Most DCNNs for semantic segmentation applied BN as the normalization method. For medical image segmentation which is usually trained from scratch, it is possible to substitute the BN with other normalization methods for better performance. In this chapter, we proved that detailed subdivision of the feature map, i.e. GN with a large group number or IN, facilitates the generalization of the trained model and hence improves the performance. Our experiments also indicate other conclusions: 1) small batch size out-performed large batch size; 2) TestI out-performed TrainI when applying BN during inference. However, we do not think our experiments are sufficient to fully prove these two conclusions. Hence, we would leave it open.

The proposed ILN strategy is generic and flexible. The three components - IN, LN and GN16 are selected as examples and it is worth to explore other combinations as well in the future. The proposed ILN framework is validated

on medical image segmentation with a U-Net framework. We believe that it could also be useful for other tasks, which needs further validation and exploration. The proposed ILN failed to achieve the highest DSC for the LV-3 experiment. It may due to the fact that the weighted normalization of IN, LN and GN16 is not suitable for this experiment. Due to the fact that weighting more normalization methods will increase the running time of each iteration in both the training and testing stage, in this chapter, only IN and LN are weighted. In the future, the proposed ILN framework would be extended to weight more normalization methods. The mean and *std* DSC are used to justify the achieved accuracy, other statistical values, i.e. p-value, can also be supplied to enhance the evaluation. To further evaluate the proposed trainable ILN, a baseline experiment with the ρ in ILN fixed rather than being trained can be added.

Although the focus of this chapter is a fundamental problem in training U-Net for medical image segmentation - normalization, this chapter connects and contributes to surgical robotic vision. The three segmented anatomies - RV, aorta, LV could be used for cardiac robotic navigation [10] and surgical robotic path planning, based on previous work of 3D shape instantiation [6] [67].

In conclusion, first, this chapter explores the medical image segmentation in surgical robotic vision and focuses on the normalization in training U-Nets. Four most popular normalization methods - BN, IN, LN and GN are reviewed and compared in details. Detailed subdivision of the feature map, i.e. GN with a large group number or IN, improves the accuracy of training U-Net for medical semantic segmentation. This accuracy improvement is mainly from improved generalization ability of the trained model. This work could help with indicating the future direction on proposing new normalization methods.

Second, to improve the accuracy of medical image segmentation based on U-Net, the ILN was proposed to combine the feature map of IN and LN with an additional trainable parameter and sigmoid function, then add GN16 after the combined feature map. Although, various normalization methods have been proposed, the accuracy improvements of the proposed ILN - almost 2% DSC shows the importance of carefully tuning the normalization strategy when training DCNNs for medical image segmentation.

4 Atrous Convolutional Neural Network (ACNN) with Full-scale Feature Maps

In chapter 3, I discussed my work on normalization methods in training Deep Convolutional Neural Network (DCNN) for medical image segmentation. Except normalization, network architecture design is another important research area that can improve the performance of training DCNN for medical image segmentation. In this chapter, I will introduce my work - Atrous Convolutional Neural Network (ACNN) for medical image segmentation. U-Net is the most popular traditional DCNN for medical image segmentation with using max-pooling layers to increase the DCNN receptive field and deconvolutional layers to recover the spatial dimension. In this chapter, I design a new network architecture where no max-pooling layers are used. The receptive field is increased by atrous convolution. As the main purpose of this chapter is to improve the automation of the 3D shape instantiation work in chapter 5 and 6, main validations are focused on the RV, LV and aortic 2D segmentation.

4.1 Introduction

In medical image segmentation, conventional methods are based on ad hoc, expert-designed feature extractors and trained classifiers. Recently, the use of DCNNs has shown promising results for many vision-based tasks including image classification [36], object detection [103], and semantic image segmentation [5]. In DCNN, features are extracted and classified automatically by training multiple non-linear modules [75]. Unlike traditional fully connected neural networks where each output node is linked to all input nodes, an output node of DCNN only links to regional input nodes, known as

the receptive field (the input nodes that influence an output node). Multiple convolutional layers, as shown in Fig. 4.1a, and down-sampling layers, i.e., pooling layers shown in Fig. 4.1b, are cascaded to achieve a large receptive field coverage. This large receptive field is essential for extracting and classifying underlying visual features and semantic details. The use of this kind of DCNN means that the feature map is also down-sampled, which can be detrimental to pixel-level tasks, i.e., segmentation. For medical images with focal lesions, local features with small sizes may be discarded due to down-sampling.

In order to compensate for decreased dimension of feature maps, various techniques have been proposed. For example, deconvolutional layers and non-linear up-sampling are used respectively in Fully Convolutional Network (FCN) [40] and SegNet [104] to recover the down-sampled feature map to the input image size. An alternative is to use atrous convolution [5], also known as dilated convolution [105], to replace the down-sampling layer in traditional DCNNs to increase the receptive field. Atrous convolution inserts zeros between non-zero filter taps to sample the feature map as shown in Fig. 4.1c. It increases the receptive field with the atrous rate but maintains the spatial dimension of feature maps without increasing the computational complexity. However, applying atrous convolution introduces a high demand on memory usage and the inserted zeros of atrous convolution cause input node or information missing. These challenges have limited the practical use of atrous convolution, particularly for medical image segmentation. For example, a high-resolution and compact CNN was designed with dilated convolution and residual learning for brain MR volume segmentation [106].

As mentioned above, memory shortage is the first challenge for applying atrous convolution, as high-resolution feature map propagation consumes a large amount of memory. In previous work, atrous convolution was usually applied jointly with down-sampling layers as a trade-off between the accuracy and memory. For example, in Deeplab [5], a feature map at $1/8$ spatial size of the input image was first extracted by multiple convolutional and down-sampling layers. Feature maps with a larger receptive field but with the same $1/8$ spatial size were then calculated by multiple atrous convolutional layers. Subsequently, bilinear interpolation was used to recover the spatial dimension of the down-sampled feature maps, while conditional random field was used to refine the predicted pixel-level probability. In multi-scale

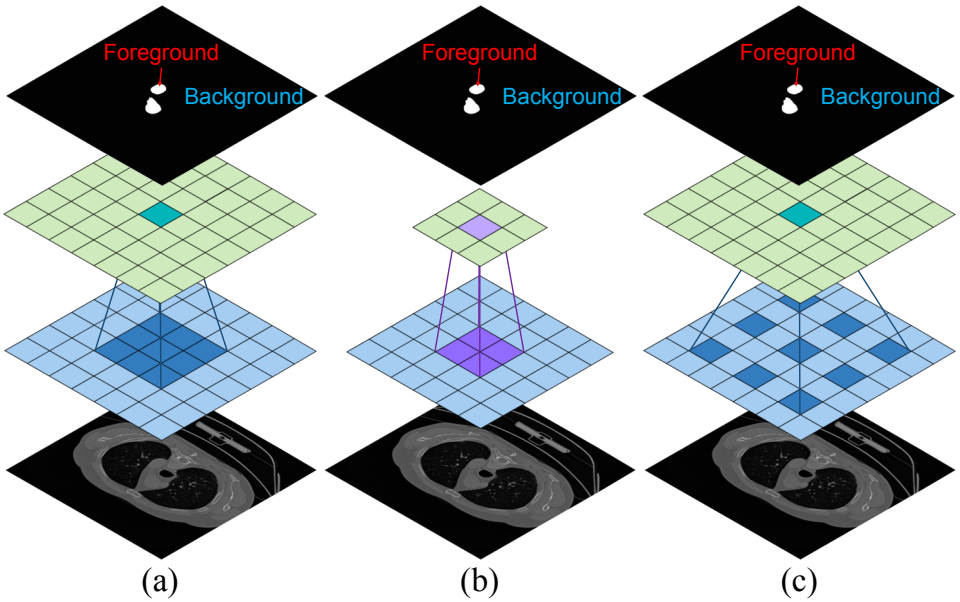


Figure 4.1: Illustrations of using DCNN with different receptive fields for medical image segmentation: (a) convolutional layer with a 3×3 receptive field; (b) pooling layer with a 2×2 receptive field; (c) atrous convolutional layer (atrous rate is 2) with a 5×5 receptive field.

context aggregation [105], a feature map with 64×64 dimension was firstly down-sampled from the input image, then a context module with seven atrous convolutional layers was applied to extract features with a larger receptive field at the same spatial dimension. Similar joint usage of atrous convolutional and down-sampling layers can also be found in [107].

In practice, setting the atrous rates is another challenge when applying atrous convolution. This is because the output node only links to input nodes which align with non-zero filter taps, as shown in Fig. 4.1c. The input nodes which align with zero filter taps are not considered. There are thus far no standard ways of setting the atrous rates. For example, an atrous rate setting of (1, 1, 2, 4, 8, 16, 1) was allocated for achieving a receptive field of 67×67 in [105] following the strides of max-pooling layers in FCN. Wang *et al.* found that an atrous rate setting of (2, 4, 8) would cause gridding effects (regular input nodes are missed) and proposed a hybrid atrous rate setting, i.e., (1, 2, 5, 9) to guarantee the coverage of all input nodes [107]. An atrous rate setting of (6, 12, 18) was used for each block and an atrous rate setting of (1, 2, 4) was set inside each block in [108] based on empirical knowledge.

In this chapter, we propose a full-scale DCNN where the spatial dimension of intermediate feature maps remains the same as that of the input image. This is different from the work of [109], for which the spatial dimension of intermediate feature maps at the residual stream is still smaller than that of the input image. For proposing a full-scale DCNN, the proposed network needs to: 1) maximize the receptive field with as few atrous convolutional layers as possible to save the memory usage; 2) fully cover the receptive field without missing any input node. In the following sections, we first prove a method that sets the atrous rate as $(k)^{n-1}$ at the n^{th} atrous convolutional layer, where k is the kernel size and n is the sequence number of atrous convolutional layer, can achieve the largest and fully-covered receptive field with a minimum number of atrous convolutional layers in Sec. 4.2.1. Then six atrous blocks, three shortcut connections and four different normalization methods are explored in Sec. 4.2.2, to select the optimal atrous block, shortcut connection and normalization method through experimental investigations. Finally, a full-scale DCNN - ACNN is proposed by using multiple cascaded atrous II-blocks, residual learning and Fine Group Normalization (FGN). Cardiovascular Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) image segmentation of the Right Ventricle (RV), Left

Ventricle (LV) and aorta are used to validate the proposed ACNN with data collection shown in Sec. 4.2.3 and with results shown in Sec. 4.3. U-Net [3], optimized U-Net with FGN [4] and a hybrid network similar to [5] are used as the comparison for performance assessment. It has been shown that the proposed ACNN can achieve comparable segmentation Dice Similarity Coefficients (DSCs) compared to other techniques with much less trainable parameters and model sizes, indicating the benefit of full-scale feature maps in DCNN. Discussions and conclusions are stated in Sec. 4.4.

4.2 Methodology

4.2.1 Atrous Rate Setting

In this section, we focus on optimizing the atrous rate setting which would achieve the largest and fully-covered receptive field with a minimum number of atrous convolutional layers. Before presenting the detailed mathematical derivation, three 1D receptive field examples with three different atrous rate settings are intuitively shown in Fig. 4.2. In this three-layer network, with an atrous rate setting of (1, 2, 4), a receptive field of 15 is achieved, while with an atrous rate setting of (1, 2, 9), a receptive field of 25 is achieved with a coverage ratio (the ratio of linked input nodes over all input nodes in the receptive field) of 0.84. With the proposed atrous rate setting of (1, 3, 9), the largest receptive field of 27 is achieved with a full coverage, i.e., the coverage ratio is 1.0. Detailed mathematical proofs are presented below. For simplification, batch size is considered as 1 here.

With an input feature map $\boldsymbol{F}^{n-1}$ of size $\mathrm{H} \times \mathrm{W} \times \mathrm{c}_{n-1}$, an output feature map $\boldsymbol{F}^{n}$ of size $\mathrm{H} \times \mathrm{W} \times \mathrm{c}_{n}$ is calculated by the n^{th} atrous convolutional layer with an atrous rate r_n and padding, where $\boldsymbol{F}^{\boldsymbol{0}} \in \mathbb{R}^{\mathrm{H} \times \mathrm{W} \times \mathrm{c}_0}$, $\boldsymbol{F}^{n} \in \mathbb{R}^{\mathrm{H} \times \mathrm{W} \times \mathrm{c}_n}$, $n \in [1, \mathrm{N}] \cap \mathbb{N}$, and $\boldsymbol{r} = (r_1 \cdots r_{\mathrm{N}})^\top \in \mathbb{N}^{\mathrm{N}}$, where $\mathrm{N} \in \mathbb{Z}_+$ is the total number of atrous convolutional layers. Here $\mathrm{H} \in \mathbb{N}$ is the feature height and $\mathrm{W} \in \mathbb{N}$ is the feature width, though these two values are usually equal for medical images. The channel number of feature maps is denoted as $\mathbf{c} = (\mathrm{c}_0 \cdots \mathrm{c}_{\mathrm{N}})^\top \in \mathbb{N}^{\mathrm{N}+1}$, and $\boldsymbol{F}^{\boldsymbol{0}}$ is the input image. By ignoring the non-linear modules, i.e., ReLU, and the biases, an equivalent 2D atrous convolution could achieve a backward propagation from $\boldsymbol{F}^{n}$ to $\boldsymbol{F}^{n-1}$, which can be decomposed into two 1D atrous convolutions [110], with kernel $\boldsymbol{v}^{n}$

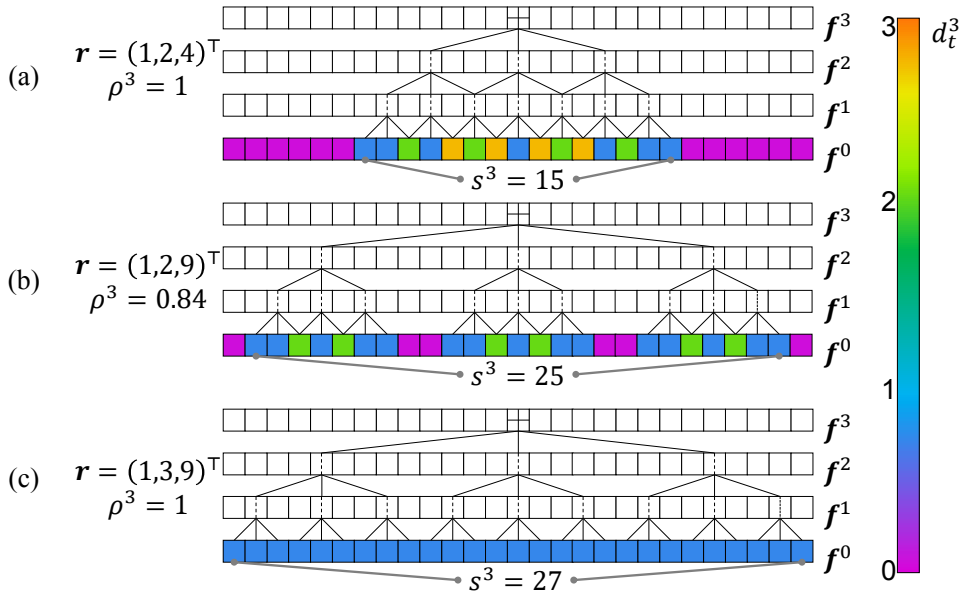


Figure 4.2: Three 1D receptive field examples with different atrous rate settings for a three-layer network: (a) an atrous rate setting of (1, 2, 4), (b) an atrous rate setting of (1, 2, 9), (c) an atrous rate setting of (1, 3, 9). The colour represents the link number from the bottom/input node to the top central/output node. ρ^3 is the coverage ratio defined by Equ. 4.7, $\boldsymbol{r}$ is the atrous rate array, s^3 is the receptive field size, $\boldsymbol{f}^{(1\sim3)}$ is the 1D feature map, $\boldsymbol{f}^0$ is the 1D input image, d_t^3 is the receptive field of f_0^3, these notations are explained and used in Sec. 4.2.1.

indexed by $t \in \mathbb{Z}$:

$$v_t^n(k, r_n) = \sum_{u=-\frac{k-1}{2}}^{\frac{k-1}{2}} w_u^n \cdot \delta(t - ur_n), \text{ where } \delta(t) := \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases} \tag{4.1}$$

Here, k is an odd number which represents the kernel size, i.e., 3, 5, or 7. t is the pixel index. w_u^n, each element of weight matrix $\boldsymbol{w}^n \in \mathbb{R}^k$, is a trainable variable, where $\mathbf{1}(t) : \mathbb{Z} \to \{0, 1\}$ is an indicator function defined as:

$$\mathbf{1}(t) := \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases} \tag{4.2}$$

Denote vectors $\boldsymbol{f}^0$, $\boldsymbol{f}^n$ as the 1D input image and the n^{th} 1D feature map, both indexed by t. $\boldsymbol{f}^0$ can be calculated from $\boldsymbol{f}^n$ by:

$$\boldsymbol{f}^0 = \boldsymbol{v}^1 * \cdots * \boldsymbol{v}^n * \boldsymbol{f}^n \tag{4.3}$$

Define $\boldsymbol{d}^n(k, \boldsymbol{r}) := \boldsymbol{f}^0(\boldsymbol{f}^n = \mathbf{1}(t))$, in which $\boldsymbol{f}^n = \mathbf{1}(t)$ indicates that only the central pixel of $\boldsymbol{f}^n$ is with a non-zero value (=1). It is calculated as:

$$\boldsymbol{d}^n(k, \boldsymbol{r}) := \boldsymbol{v}^1 * \cdots * \boldsymbol{v}^n * \delta(t) \tag{4.4}$$

By setting $\boldsymbol{w}^n = (1)_k, \forall n$, vectors consisting of 1, then $d_t^n \in \mathbb{N}$, the element indexed by $t \in \mathbb{Z}$, is the link number from f_0^n to the input image's pixel or node. Thus, $\boldsymbol{d}^n$ represents the receptive field of f_0^n, where its receptive field coverage could be represented by the non-zero element number in vector $\boldsymbol{d}^n$:

$$\|\boldsymbol{d}^n\|_0 := \sum_t \left(1 - \mathbf{1}(d_t^n)\right) \tag{4.5}$$

and its receptive field size $s^n \in \mathbb{N}$ is calculated as:

$$s^n(k, \boldsymbol{r}) = 1 + (k - 1) \sum_{m=1}^{n} r_m \tag{4.6}$$

The receptive field coverage ratio of f_0^n, denoted by $\rho^n \in \mathbb{R}_+$, is then defined as:

$$\rho^n(k, \boldsymbol{r}) := \frac{\|\boldsymbol{d}^n\|_0}{s^n} \tag{4.7}$$

In order to ensure a fully-covered receptive field from an output pixel or node, our target is to maximize the receptive field size with a constraint of receptive field coverage ratio:

$$\max_{\boldsymbol{r}\in\mathbb{N}^{\mathrm{N}}} \left\{ s^{\mathrm{N}} : \rho^{\mathrm{N}} = 1 \right\} \tag{4.8}$$

By substituting Equ. 4.6 and Equ. 4.7 into Equ. 4.8, the optimization problem can be converted as:

$$\max_{\boldsymbol{r}\in\mathbb{N}^{\mathrm{N}}} \left\{ \left\| \boldsymbol{d}^{\mathrm{N}} \right\|_0 : \left\| \boldsymbol{d}^{\mathrm{N}} \right\|_0 = 1 + (k-1) \sum_{n=1}^{\mathrm{N}} r_n \right\} \tag{4.9}$$

The total link number from f_0^n to $\boldsymbol{f}^0$ is represented by:

$$\left\| \boldsymbol{d}^n \right\|_1 = \sum_t d_t^n = (k)^n \tag{4.10}$$

where $(k)^{\mathrm{n}}$ represents an exponent calculation. It is the upper bound of $\left\| \boldsymbol{d} \right\|_0$ because:

$$\left\| \boldsymbol{d} \right\|_0 \leq \left\| \boldsymbol{d} \right\|_1, \forall d_t \in \mathbb{N}, \forall t \in \mathbb{Z} \tag{4.11}$$

where

$$\left\| \boldsymbol{d} \right\|_0 = \left\| \boldsymbol{d} \right\|_1 \Leftrightarrow d_t \in \{0, 1\}, \forall t \in \mathbb{Z} \tag{4.12}$$

We assume that Equ. 4.12 holds. By substituting this into the constraint of Equ. 4.9:

$$1 + (k-1) \sum_{n=1}^{\mathrm{N}} r_n = (k)^{\mathrm{N}} \tag{4.13}$$

This is a sum of geometric progression; one solution can be obtained as:

$$\boldsymbol{r}' = \begin{pmatrix} 1 & \cdots & (k)^{n-1} & \cdots & (k)^{\mathrm{N}-1} \end{pmatrix}^{\top} \tag{4.14}$$

It satisfies a uniformly covered receptive field: $d_t^{\mathrm{N}}(k, \boldsymbol{r}') = \begin{cases} 0 & t \notin \mathbb{S} \\ 1 & t \in \mathbb{S} \end{cases}$, where $\mathbb{S} := [-\frac{s^{\mathrm{N}}-1}{2}, \frac{s^{\mathrm{N}}-1}{2}] \cap \mathbb{Z}$ in 1D and the same in 2D, which satisfies the equivalent condition in Equ. 4.12 and thus is a solution to Equ. 4.9. Therefore, the atrous rate setting of $(k)^{n-1}$ at the n^{th} atrous convolutional layer could lead to the largest and fully-covered receptive field under the

condition that the same number of atrous convolutional layers is used.

Traditional DCNNs composed of convolutional layers and down-sampling layers are with Gaussian covered receptive field. The path number for nodes at $\boldsymbol{F}^0$ contribute to $F^{\mathrm{N}}_{0,0}$ shrinks quickly from the central area to the outer area, which is called Gaussian damage in [110]. The weights for the outer area nodes grow during the training, indicating that outer area nodes are also important. A weight initialization with higher weights at the outer area and lower weights at the central area was tried to compensate this Gaussian damage, however, the improvement is limited and unstable [110]. We propose uniformly covered receptive field which may be a solution for Gaussian damage, but with a quite different purpose - achieve fully-covered receptive field with minimum number of atrous convolutional layers.

4.2.2 Atrous Convolutional Neural Network

I first introduce the ACNN structure briefly in these two paragraphs and then explain each component of the proposed ACNN in details in below sections. With the proof in Sec. 4.2.1, a receptive field of $(k)^{\mathrm{N}}$ could be achieved by a block of N atrous convolutional layers. Each node in the receptive field is linked evenly. In this chapter, the kernel size of atrous convolutional layers is 3, following the settings used in [37]. A block of N atrous convolutional layers has a receptive field of $(3)^{\mathrm{N}}$. We call this block as atrous block and the one specific with N atrous convolutional layers as N-block, here N is expressed in the roman numeral.

The proposed ACNN is designed into multiple cascaded atrous blocks to increase the receptive field linearly by $(3)^{\mathrm{N}}$. For achieving a $\mathrm{H} \times \mathrm{W}$ (usually $\mathrm{W} = \mathrm{H}$) whole-image coverage, $\mathrm{H}/(3)^{\mathrm{N}}$ blocks are cascaded. For solving the gradient vanishing/exploding problems and facilitating back propagation, shortcut connections including residual learning, identity mapping and dense connection, and normalization methods including Batch Normalization (BN), Layer Normalization (LN), Instance Normalization (IN) and Group Normalization (GN), are explored and assessed.

Atrous Block

To determine the optimal ACNN structure, different atrous blocks are explored. As the test image size in this chapter is 512 or 256, six atrous blocks

(I-block, II-block, III-block, IV-block, V-block, VI-block) are assessed with the receptive field of 3, 9, 27, 81, 243, 729, respectively, as shown in Fig. 4.3. The feature channels of the input and output feature map are the same.

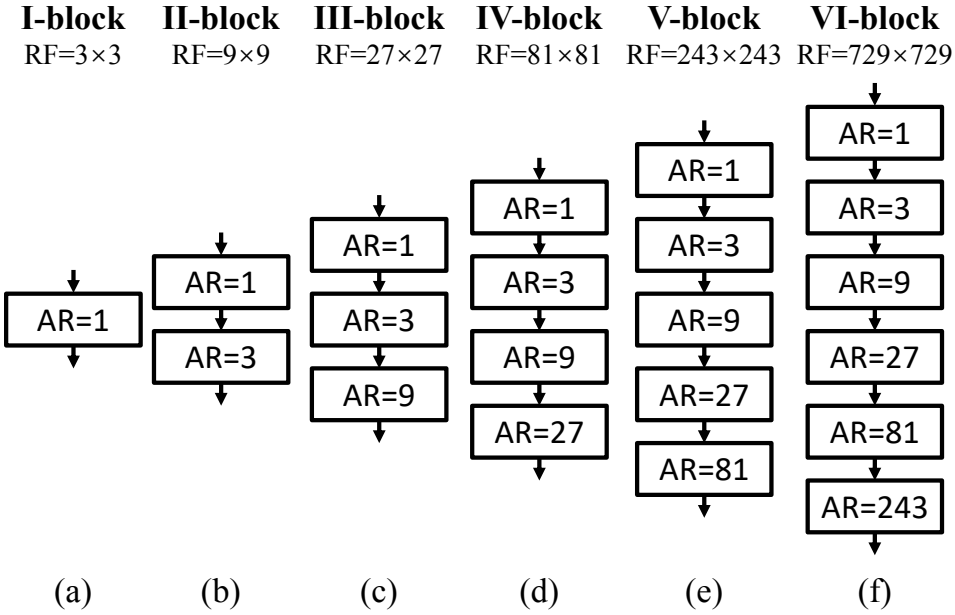


Figure 4.3: Six atrous blocks: I-block, II-block, III-block, IV-block, V-block, VI-block with 1, 2, 3, 4, 5, 6 atrous convolutional layers inside the block. The Atrous Rate (AR) is set as $(3)^{n-1}$ at the n^{th} layer, $n \in [1, \mathrm{N}] \cap \mathbb{N}$ is the sequence number of the atrous convolutional layer, $\mathrm{N} \in \{1, 2, 3, 4, 5, 6\}$ is the total number of atrous convolutional layers in each block.

The optimal atrous block is determined by experiments, as shown in Sec. 4.3.1. Here, we state and use the conclusion in advance - atrous II-block is the optimal atrous block and is used in the following context.

Shortcut Connection

Plain DCNN experiences the degradation and gradient vanishing/exploding problems [38]. Shortcut connection is a solution to these problems. Three popular shortcut connections are explored in this chapter: 1) residual learning [38], 2) identity mapping [111], 3) dense connection [112]. Dense connection is seen as a shortcut connection method, as it was shown to be re-exploring the feature maps while residual learning was shown to be re-using the feature maps in [113], hence it is classified as the same type of techniques as residual

learning - shortcut connection in this chapter. In residual learning, the normalization layer and ReLU are placed after the atrous convolutional layer and $\boldsymbol{F}^{n-2}$ is added to $\boldsymbol{F}^{n}$. For identity mapping, the normalization layer and ReLU are placed before the atrous convolutional layer and $\boldsymbol{F}^{n-2}$ is added to $\boldsymbol{F}^{n}$. In dense connection, the normalization layer and ReLU are placed before the atrous convolutional layer and $\boldsymbol{F}^{n-1}$ is concatenated to $\boldsymbol{F}^{n}$. As the feature map is in high-resolution and the layer number is large (64 layers for the RV and LV experiments while 128 layers for the aorta experiments) in this chapter, fully dense connection could not be achieved due to the extremely high memory usage. In this chapter, a dense connection is placed after 16 (for the RV and LV experiments) or 32 (for the aorta experiments) atrous II-blocks, resulting in four dense connections in total. This grouped dense connection is called dense4 connection, as the atrous convolutional layers are concatenated four times.

The optimal shortcut connection is determined by experiments, as shown in Sec. 4.3.2. Here we rely on the fact that residual learning is the optimal shortcut connection and is used in the following context.

Normalization Method

For DCNN, when the value distribution of shallow feature maps or parameters changes, the parameters of deep layers would be trained to fit this distribution change rather than to fit the real and useful content. This phenomenon was defined as interval covariate shift [81] and is detrimental to both the training speed and performance.

In this chapter, batch size of 1 is mainly explored, as it was shown that batch size of 1 out-performed larger batch sizes for medical image segmentation in chapter 3. For the proposed ACNN where the feature channel is the same for all intermediate feature maps, BN and IN are the same as Fine Group Normalization (FGN) (set the group number of GN as the feature channel in this chapter or the number of channel in the first feature map in [4]) when the batch size is 1. Hence, FGN which also represents BN and IN, GN4 which sets the group number of GN as 4 and LN are explored for the subdivision of feature maps.

During inference, one way to apply BN, IN, LN and GN is to use the mean and variance of the current testing feature maps to normalize the testing

feature maps. BN in this mode is called BN-train in this chapter. There is an additional way to apply BN, i.e., to use the moving average mean and variance of the training feature maps to normalize the testing feature maps. BN in this mode is called BN-infer, which is also explored in this chapter.

The optimal normalization method is selected based on experimental results, as shown in Sec. 4.3.3. Here, we assume that - FGN is the optimal normalization method. FGN was also shown to be the optimal normalization method when using a U-Net structure for medical image segmentation in chapter 3. For a U-Net structure, FGN is different from BN and IN, as the feature channel changes inside the DCNN. To be consistent with chapter 3 and for better generalizability, FGN is used as a representation of FGN, IN and BN in this chapter.

ACNN Architecture

The final proposed ACNN architecture is shown in Fig. 4.4. Multiple atrous II-blocks with residual learning and FGN are cascaded. The number of residual II-blocks - $^{(H-1)}/_{8}$ is determined by the input image size, i.e., 32 for a 256×256 image while 64 for a 512×512 image.

4.2.3 Experimental Setup and Validation

Three cardiovascular MRI and CT datasets for RV, LV and aorta segmentation were used for validation of the proposed ACNN.

Right Ventricle (RV) 37 patients, with different levels of Hypertrophic Cardiomyopathy (HCM) were scanned with a 1.5T MRI scanner (Sonata, Siemens, Erlangen, Germany) [6], involving 6082 images with 10mm slice gap, $1.5 \sim 2$mm pixel spacing, $19 \sim 25$ times frames, and 256×256 image size. Analyze (AnalyzeDirect, Inc, Overland Park, KS, USA) was used to label the ground truth. Rotation from -30° to 30° with 10° as the interval was used to augment the images. Three groups, with 12, 12, and 13 patients respectively, were split randomly from the 37 patients for three-fold cross validations.

Left Ventricle (LV) 45 patients, from the SunnyBrook MRI data set [101] were used, it has 805 images with 256×256 image size. Rotation from -60°

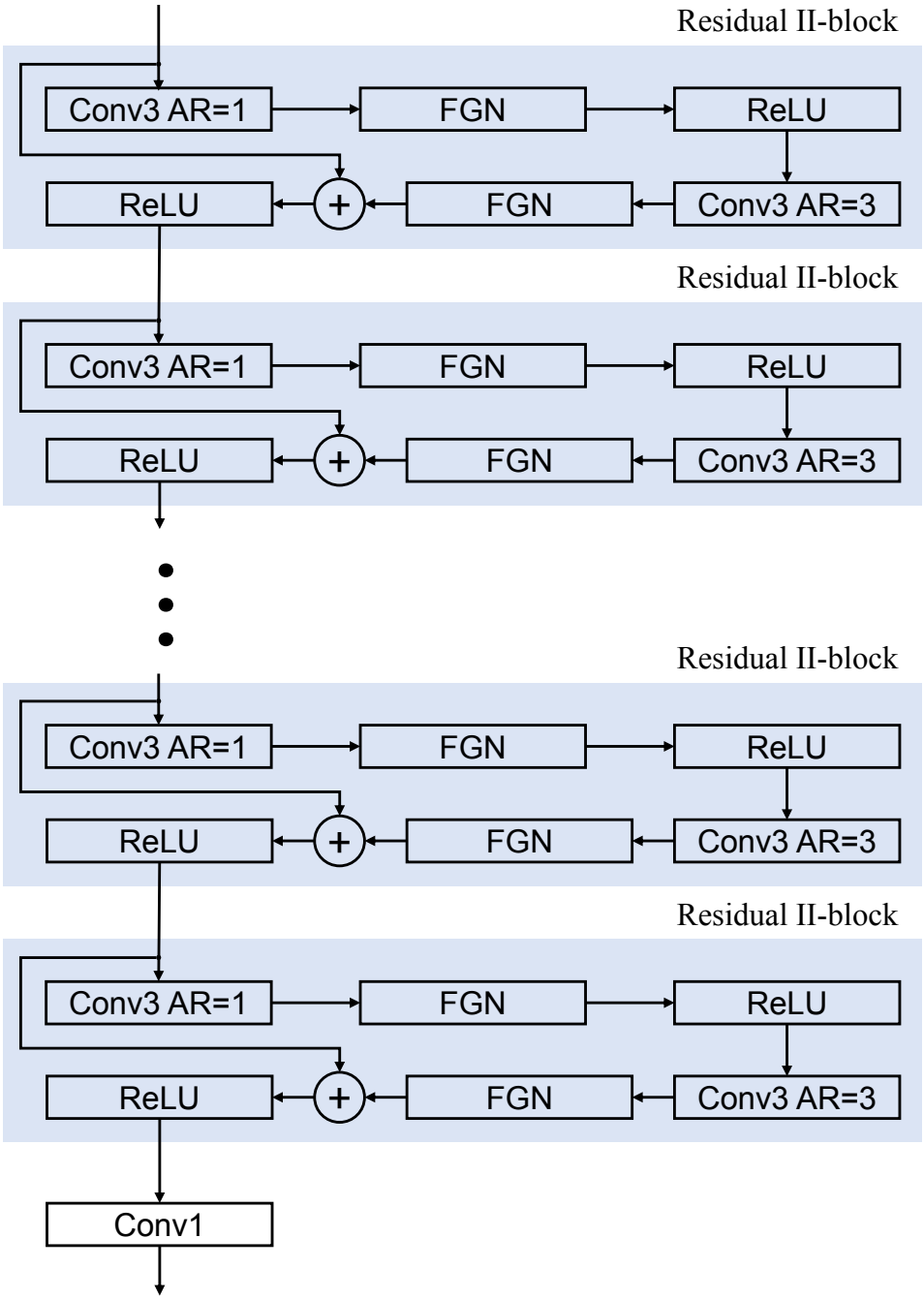


Figure 4.4: The network architecture of the proposed ACNN. The number of residual II-blocks is determined by $^{(H-1)}/_{8}$, H is the height or width of input image. AR - atrous rate, Conv3 - atrous convolution with kernel size of 3, Conv1 - atrous convolution with kernel size of 1.

to 60° with 2° as the interval was used to augment the images. Three groups, with 15 patients respectively, were split randomly from the 45 patients for three-fold cross validations.

Aorta 20 patients, from the VISCERAL data set [100], were used, 4631 CT images with 512×512 image size. Rotation from $-40°$ to $40°$ with $10°$ as the interval was used to augment the images. Three groups, with 7, 7, and 6 patients respectively, were split randomly from the 20 patients for three-fold cross validations. As we mainly focus on 2D medical image segmentation in this chapter for increasing the automation of 3D shape instantiation in chapter 5 and 6, each 3D aortic CT scan is sliced into multiple 2D CT images for the validation.

The maximum and minimum image intensity of all patients were used to re-scale the image intensity to $0.0 \sim 1.0$. Due to the limitation of available image data, the validation dataset were not split or used. For cross validations, two groups were used in the training stage while the other group was used in the testing stage. The kernel size of the last atrous convolutional layer is 1 while the kernel size of all the other atrous convolutional layers is 3. The momentum was set as 0.9. Multiple epoch settings, i.e., 1, 2, or 3 and multiple learning rate schedules, i.e., dividing the learning rate by 5 or 10 at the second or third epoch, indicating an optimal learning schedule that: two epochs were trained and the learning rate was divided by 5 at the second epoch. Five initial learning rates: 1.5, 1.0, 0.5, 0.1, 0.05 were trained for each experiment and the highest accuracy achieved on the test dataset was recorded as the final accuracy to avoid non-optimal hyper-parameter settings. For all experiments conducted, Stochastic Gradient Descent (SGD) was utilized as the optimizer.

Pixel-level softmax was applied after the proposed ACNN to transfer the network outputs into probabilities:

$$p_{\mathrm{nc}} = \frac{e^{y_{\mathrm{nc}}}}{\sum_{i=1}^{\mathrm{N}C} e^{y_i}} \tag{4.15}$$

Here, y is the output of proposed ACNN, p is the predicted probability, NC is the number of predicted classes. Cross-entropy was used as the loss function:

$$Loss = -\sum_{i=1}^{\mathrm{W}}\sum_{j=1}^{\mathrm{H}}\sum_{nc=1}^{\mathrm{N}C} L_{(i,j,nc)} log(P_{(i,j,nc)}) \tag{4.16}$$

DSC was used as the evaluation matrix:

$$DSC = 2 \cdot \frac{|L \cap P|}{|L + P|} \tag{4.17}$$

The DSC of the foreground is selected to represent the segmentation accuracy. The workers used were *Titan Xp* (12G memory) and *1080Ti* (11G memory) with the CPUs of an Intel® Xeon(R) CPU E5-1650 v4 @ 3.60GHz × 12 and an Intel® Xeon(R) CPU E5-1620 v4 @ 3.50GHz × 8.

The method was implemented with the Tensorflow Estimator Application Programming Interface (API). The atrous convolution and BN were programmed with *tf.layers*. The IN, LN and GN were programmed with *tf.contrib.layers*. The data was shuffled globally when generating the *tfrecords* file and was shuffled again with shuffle size of 500 when feeding images with *tf.data*, which ensures a random image input. The Tensorflow version used is 1.8.0. The process status of the CPU and GPU both influence the training speed. Training all models under exactly the same computer process status is not possible. For a fair speed comparison, the time recorded in this chapter is for 100 iterations under the computer process status where all other processes are ended. The memory usage was recorded by using *watch nvidia-smi* command. The parameter amount is for the weights and biases in the atrous convolutional layers and was recorded based on *model.summary()* in Keras.

4.3 Results

Six atrous blocks were assessed and validated on the three datasets: RV, LV and aorta to select the optimal atrous block. For the RV and LV datasets with an image size of 256 × 256, 128 atrous I-blocks (Model 1), 32 atrous II-blocks (Model 2), 10 atrous III-blocks (Model 3), 3 atrous IV-blocks (Model 4), 1 atrous V-block (Model 5) were cascaded respectively for a whole-image receptive field. The feature channel was set as 12, 16, 24, 38, 64 to maintain a similar number of trainable parameters used in each model, this guaranteed a fair comparison between the five models. The parameter number in each

model was 1.66×10^5, 1.46×10^5, 1.51×10^5, 1.44×10^5, and 1.48×10^5 respectively. For the aorta dataset with an image size of 512×512, 256 atrous I-blocks (Model 1), 64 atrous II-blocks (Model 2), 20 atrous III-blocks (Model 3), 6 atrous IV-blocks (Model 4), 2 atrous V-blocks (Model 5), and 1 atrous VI-block (Model 6) were cascaded respectively for a whole-image receptive field. The feature channel was set as 12, 16, 24, 38, 64, 80 respectively to maintain a similar number of trainable parameters used in each model. The parameter number was 3.34×10^5, 2.95×10^5, 3.08×10^5, 3.00×10^5, 3.33×10^5, and 2.89×10^5 respectively. Before confirming the optimal shortcut connection and normalization method, identity mapping and FGN was used as the shortcut connection and normalization method respectively in this section of experiments. Detailed results are shown in Sec. 4.3.1.

Three shortcut connections: residual learning, identity mapping and dense4 connection were explored and validated on the three datasets to select the optimal shortcut connection. Details are illustrated in Sec. 4.3.2. Before confirming the optimal normalization method, FGN was used as the normalization method in this section of experiments. Four normalization methods: BN-infer, LN, FGN (the same as BN-train and IN in this chapter), GN4 were validated on the three datasets to select the optimal normalization method and details are presented in Sec. 4.3.3. Examples of the segmentation results are shown in Sec. 4.3.4.

Three popular DCNNs were used for the comparison. (1) U-Net proposed in [3] with five max-pooling layers; (2) Optimized U-Net with FGN proposed in [4] with seven max-pooling layers to achieve the largest receptive field; and (3) a hybrid DCNN similar to the Deeplab proposed in [5] but with much less trainable parameters to decrease the memory usage, the intermediate eight blocks with max-pooling and deconvolutional layers in the Optimized U-Net were replaced with four atrous convolutional blocks which were composed of two convolutional layers with atrous rates as 1 plus one atrous convolutional layer with atrous rate setting of (2, 4, 8, 16) respectively. The feature channel root was set as 16 for all methods. Details regarding the network structure are shown in Fig. 4.5. A detailed comparison regarding the accuracy, memory usage, and speed are given in Sec. 4.3.5. It was known that slight difference exists even by training exactly with the same model setting in multiple times [97]. In this chapter, this variance is given in Sec. 4.3.6.

In the following sections, RV-1 refers to the first cross validation (use the

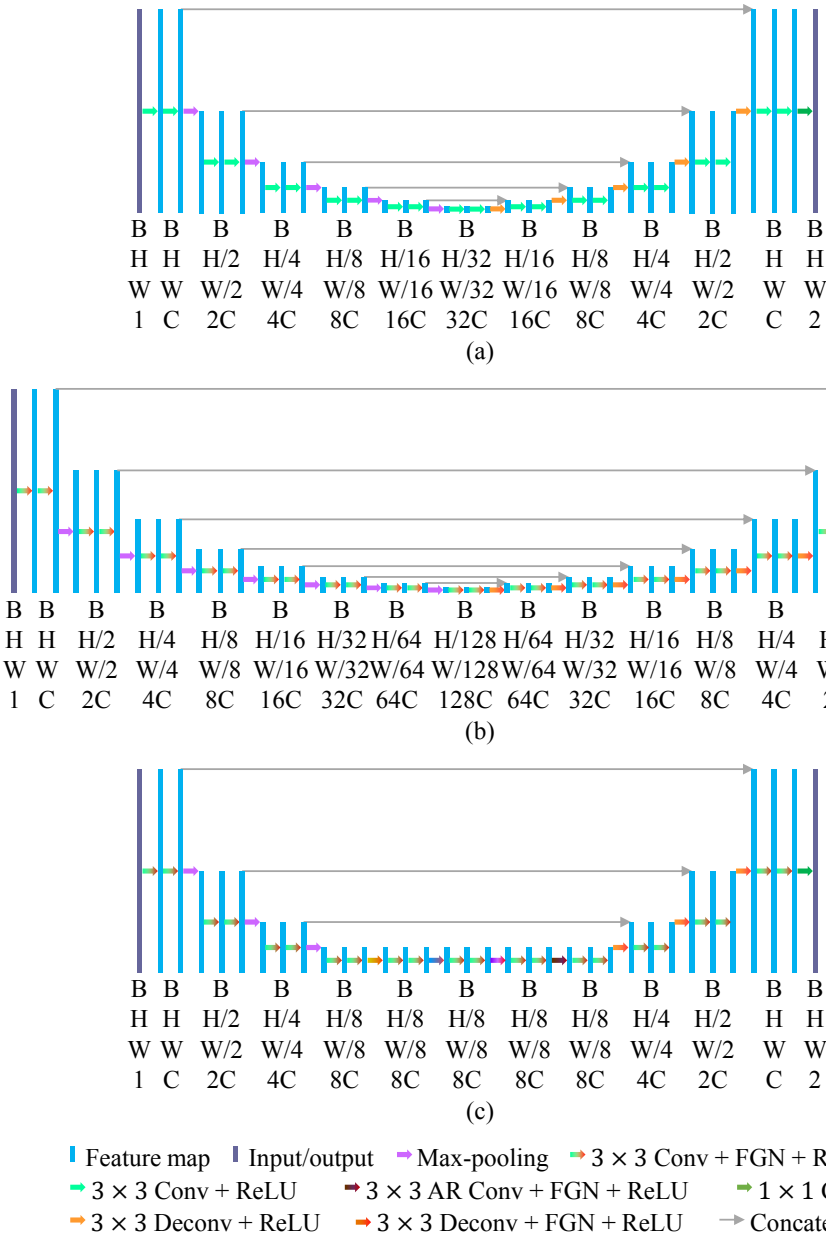


Figure 4.5: The network architectures of three comparison DCNNs: (a) U-Net [3]; (b) optimized U-Net [4]; (c) hybrid network [5]; Conv - convolutional layers, Deconv - deconvolutional layers, AR Conv - atrous convolutional layers with atrous rate setting of (2, 4, 8, 16) respectively, FGN - fine group normalization.

first group as the testing and use the second and third group as the training) of RV segmentation, this notation also applies to RV-2, RV-3, LV-1, LV-2, LV-3, Aorta-1, Aorta-2, and Aorta-3.

Table 4.1: The mean±std DSC, optimal learning rate (OLR), memory usage and training time for 100 iterations for the five or six ACNN models with different atrous blocks for the RV, LV and aorta segmentation, highest DSCs are labelled in bold and red.

Model	mean±std DSC	OLR	mean±std DSC	OLR	mean±std DSC	OLR	Memory	Time
	RV-1		RV-2		RV-3			
Model 1	0.6588±0.3333	0.5	**0.7253±0.2846**	1.5	0.6653±0.3261	0.5	2.55G	14.5s
Model 2	0.6425±0.3476	0.5	0.7169±0.2812	0.5	**0.6825±0.3265**	0.5	1.65G	9.4s
Model 3	**0.6688±0.3284**	0.05	0.7167±0.2831	0.05	0.6462±0.3344	0.5	1.67G	6.3s
Model 4	0.6470±0.3265	0.05	0.6685±0.3174	0.05	0.6208±0.3281	0.1	1.96G	4.6s
Model 5	0.6007±0.3278	0.1	0.6442±0.3205	0.05	0.5863±0.3556	1.0	8.59G	3.5s
	LV-1		LV-2		LV-3			
Model 1	0.8807±0.1831	1.5	0.8056±0.2467	0.1	0.7909±0.2451	1.5	2.55G	14.5s
Model 2	**0.9155±0.1107**	0.1	0.8590±0.1627	1.0	**0.8186±0.2310**	0.5	1.65G	9.4s
Model 3	0.9118±0.1172	0.5	**0.8721±0.1743**	0.05	0.8008±0.2493	0.5	1.67G	6.3s
Model 4	0.8857±0.1452	0.1	0.8580±0.1513	1.5	0.7921±0.2407	0.5	1.96G	4.6s
Model 5	0.8554±0.1501	0.1	0.7844±0.2204	0.05	0.7806±0.2175	0.1	8.59G	3.5s
	Aorta-1		Aorta-2		Aorta-3			
Model 1	0.8255±0.1833	0.1	0.7787±0.2019	0.05	0.7973±0.2185	1.5	11.72G	94.4s
Model 2	**0.8491±0.1543**	0.1	**0.7871±0.2095**	0.5	**0.8365±0.1726**	0.1	9.64G	62.5s
Model 3	0.8388±0.1731	0.05	0.7575±0.2528	0.5	0.8337±0.1915	0.1	9.72G	43.1s
Model 4	0.8006±0.1691	0.05	0.7235±0.2794	0.05	0.7828±0.2182	0.1	6.77G	31.8s
Model 5	0.7937±0.1580	0.1	0.6998±0.2619	0.5	0.7677±0.2568	0.1	8.82G	22.2s
Model 6	0.7026±0.2256	1.5	0.6564±0.2629	0.5	0.7427±0.2507	0.05	4.98G	24.1s

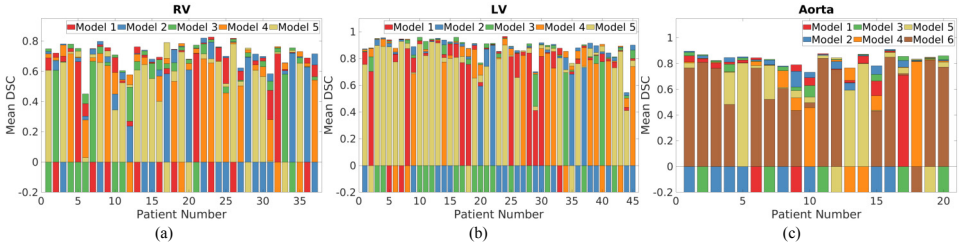


Figure 4.6: The patient mean DSC for the RV (a), LV (b) and aorta (c) dataset with using different ACNN models: Model 1, Model 2, Model 3, Model 4, Model 5 and Model 6, the bars at the negative Mean DSC axis indicate the model that achieves the highest mean DSC for that patient while the bars at the positive Mean DSC axis show the mean DSCs achieved by different models.

4.3.1 Atrous Block

The segmentation accuracy, optimal learning rate, memory usage and training time for 100 iterations of the five ACNN models for the RV segmentation, the five ACNN models for the LV segmentation, and the six ACNN models for the aorta segmentation are shown in Tab. 4.1. We can see that for most of the experiments including RV-3, LV-1, LV-3, Aorta-1, Aorta-2 and Aorta-3, Model 2 with atrous II-blocks achieves the highest accuracy. For those experiments that Model 2 under-performs (including RV-1, RV-2, and LV-2), it still achieves reasonable accuracy. For the RV and LV experiments, Model 2 with atrous II-blocks also consumes the minimum amount of memory. However, for the aorta experiments, this advantage no longer exists. This is because the aorta data is with a large image size of 512×512. This high resolution feature map propagation consumes a lot of memory and Model 2 contains many high resolution feature maps. The training time decreases along the number of atrous convolutional layers in each block - N for all the three datasets.

The mean DSC for each patient in the RV, LV and aorta dataset with using the five or six ACNN models as the segmentation methods are shown in Fig. 4.6. Model 2 with atrous II-blocks achieves the highest DSC for 12/37 RV patients, 18/45 LV patients and 8/20 aorta patients. For some patients, i.e., patient 31 in the RV dataset, patient 29 and 44 in the LV dataset, patient 10 and 15 in the aorta dataset show clearly that Model 2 with atrous II-blocks

out-performs other ACNN models. The atrous II-block is concluded as the optimal atrous block and is used in all experiments below.

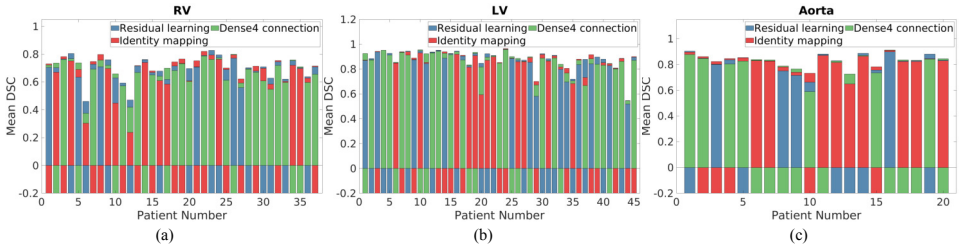


Figure 4.7: The patient mean DSC for the RV (a), LV (b) and aorta (c) dataset with different shortcut connections: residual learning, identity mapping and dense4 connection, the bars at the negative Mean DSC axis indicate the shortcut connection that achieves the highest mean DSC for that patient while the bars at the positive Mean DSC axis show the mean DSCs achieved by different shortcut connections.

4.3.2 Shortcut Connection

The segmentation accuracy, optimal learning rate, memory usage and training time for 100 iterations of the atrous II-block ACNN for segmenting the RV, LV and aorta with different shortcut connections: residual learning, identity mapping and dense4 connection are shown in Tab. 4.2. We can see that, even residual learning is not the shortcut connection which achieves the highest accuracy at most experiments. In fact, it achieves very similar accuracy to the highest value at those experiments where it under-performs, i.e., RV-3, LV-1, LV-2, LV-3, Aorta-1, and Aorta-2. Dense4 connection consumes the largest memory and takes the longest time to train. The consumed memory of the dense4 connection for the aorta experiment is an estimated value, as the real value is larger than 12G and the shown value is an optimized and approximate value. Residual learning takes almost a similar amount of memory and training time as identity mapping.

Table 4.2: The mean±std DSC, optimal learning rate (OLR), memory usage (Mem.) and training time for 100 iterations for atrous II-block ACNNs with different shortcut connections for the RV, LV and aorta segmentation, highest DSCs are labelled in bold and red.

	Residual learning			Identity mapping			Dense4 connection		
Test	mean±std DSC	OLR	Mem./Time	mean±std DSC	OLR	Mem./Time	mean±std DSC	OLR	Mem./Time
RV-1	**0.6755±0.3226**	0.5		0.6425±0.3476	0.5		0.6752±0.3256	0.05	
RV-2	**0.7267±0.2839**	1.5	1.66G/9.8s	0.7169±0.2812	0.5	1.65G/9.4s	0.7080±0.2940	0.1	2.68G/10.5s
RV-3	0.6823±0.3297	0.05		**0.6825±0.3265**	0.5		0.6559±0.3388	0.1	
LV-1	0.9133±0.1185	0.05		0.9155±0.1107	0.1		**0.9190±0.0668**	0.05	
LV-2	0.8712±0.1691	0.5	1.66G/9.8s	0.8590±0.1627	1.0	1.65G/9.4s	**0.8726±0.1627**	0.5	2.68G/10.5s
LV-3	0.8153±0.2346	0.5		**0.8186±0.2310**	0.5		0.8032±0.2392	0.5	
Aorta-1	0.8449±0.1457	0.5		**0.8491±0.1543**	0.1		0.8335±0.1743	0.01	
Aorta-2	0.7820±0.2318	0.1	9.65G/62.6s	**0.7871±0.2095**	0.5	9.64G/62.5s	0.7809±0.2094	0.05	≈11.89G/65.4s
Aorta-3	**0.8493±0.1554**	0.05		0.8365±0.1726	0.1		0.8344±0.1750	0.1	

The mean DSC for each patient in the RV, LV and aorta dataset with the three shortcut connections is shown in Fig. 4.7. For some patients, i.e. patient 6, 12, 31 in the RV dataset, patient 20 in the LV dataset, patient 19 in the aorta dataset, residual learning out-performs other shortcut connection methods. However, there are also some under-performed examples, i.e., patient 27 in the RV dataset, patient 29 in the LV dataset. Overall, residual learning is concluded as the optimal shortcut connection method and is used in later experiments.

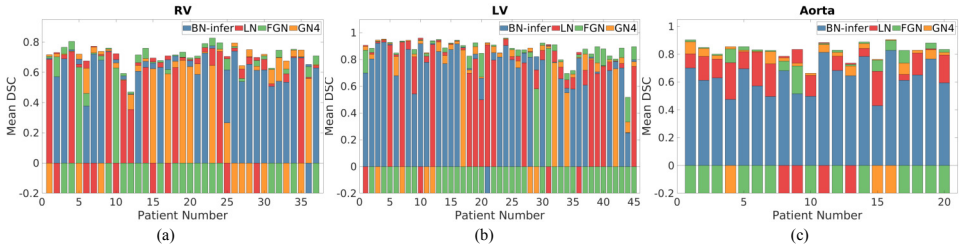


Figure 4.8: The patient mean DSC for the RV (a), LV (b) and aorta (c) dataset with different normalization methods: BN-infer, LN, FGN and GN4, the bars at the negative Mean DSC axis indicate the normalization that achieves the highest mean DSC for that patient while the bars at the positive Mean DSC axis show the mean DSCs achieved by different normalization methods.

4.3.3 Normalization Method

The mean DSC for each patient in the RV, LV and aorta dataset with the atrous II-block ACNN, residual learning and four normalization methods: BN-infer, LN, FGN, and GN4 is shown in Fig. 4.8. We can see that FGN (green color) achieves the highest accuracy at most patients. For some patients, i.e., patient 4, 14, 23 in the RV dataset, patient 39, 40 in the LV dataset, patient 17, 19 in the aorta dataset, FGN out-performs obviously. There isn't too much difference between the memory usage of the ACNNs with the four normalization methods (all around 1.65G for the RV and LV experiments and 9.64G for the aorta experiments). In terms of the training speed, FGN is similar to BN-infer and LN (around 10.0s for the RV and LV experiments while 60.0s for the aorta experiments), while GN4 is the slowest (around 14.5s for the RV and LV experiments while 95.5s for the aorta experiments). The optimal learning rates of BN-infer are usually very

small and around 0.05 while this trend does not exist for the LN, FGN and GN4 method. FGN is selected as the optimal normalization method and is used in the following experiments.

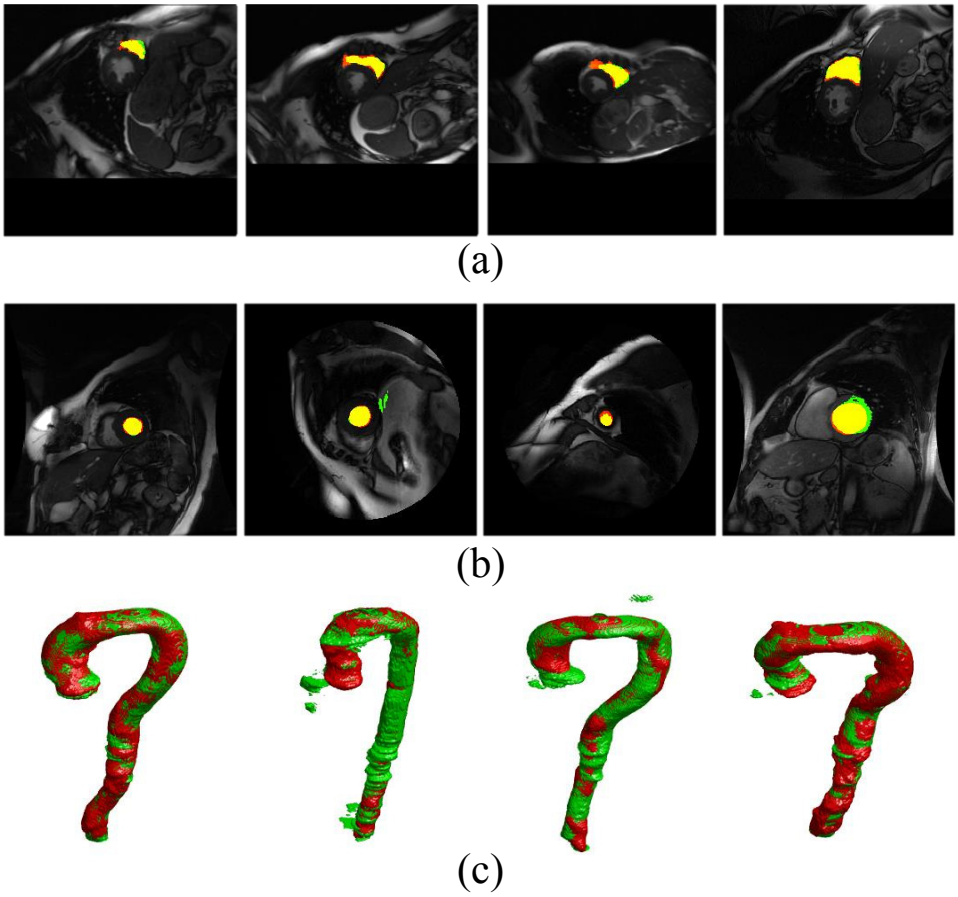


Figure 4.9: Four examples of the RV (a), LV (b) and aorta (c) segmentation results. The red color indicates the ground truth, the green color indicates the prediction, hence the yellow color indicates the overlapped pixels which are correctly segmented.

4.3.4 Segmentation Details

Four examples of the RV, LV and aorta segmentation results are shown in Fig. 4.9. As the RV and LV dataset are not volumetric MRI images, hence only 2D segmentation slices are shown.

Table 4.3: The mean±std DSC, optimal learning rate (OLR), and trainable parameters for the four different DCNNs for the RV, LV and aorta segmentation, highest DSCs are labelled in bold and red.

	The proposed ACNN		Hybrid network [5]		Optimized U-Net [4]		U-Net [3]	
Test	mean±std DSC	OLR	mean±std DSC	OLR	mean±std DSC	OLR	mean±std DSC	OLR
RV-1	0.6755±0.3226	0.5	0.7101±0.2875	1.0	**0.7204±0.2795**	0.5	0.6944±0.2428	0.5
RV-2	**0.7267±0.2839**	1.5	0.7175±0.2600	1.0	0.7002±0.2900	0.05	0.6452±0.3297	0.1
RV-3	0.6823±0.3297	0.05	**0.6907±0.2862**	0.5	0.6636±0.3047	0.1	0.6117±0.3455	0.05
LV-1	0.9133±0.1185	0.05	0.9205±0.0995	1.0	**0.9241±0.0965**	0.05	0.9240±0.0678	0.1
LV-2	0.8712±0.1691	0.5	0.8930±0.1300	0.5	**0.8932±0.1211**	0.05	0.8874±0.1592	0.1
LV-3	0.8153±0.2346	0.5	0.8306±0.2030	0.1	**0.8434±0.1912**	1.5	0.8081±0.2345	0.1
Aorta-1	**0.8449±0.1457**	0.5	0.8197±0.2018	1.0	0.8302±0.1652	0.5	0.8165±0.1843	0.05
Aorta-2	0.7820±0.2318	0.1	0.7846±0.2145	1.5	**0.8102±0.1764**	0.5	0.7938±0.2081	0.05
Aorta-3	**0.8493±0.1554**	0.05	0.7483±0.3072	1.0	0.8419±0.1737	1.0	0.7718±0.2712	0.05
Parameters	1.46M(RV, LV) / 2.95M(Aorta)		23.1M		1384.2M		86.5M	

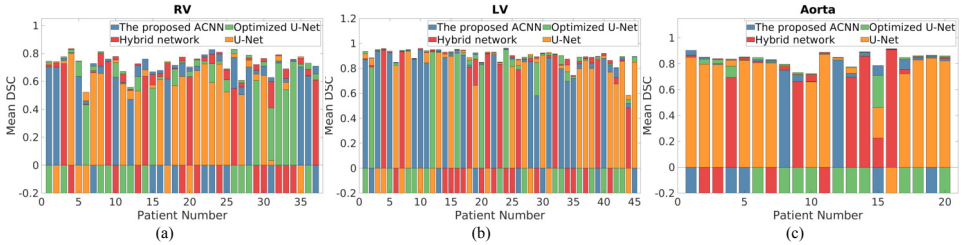


Figure 4.10: The patient mean DSC for the RV (a), LV (b) and aorta (c) dataset with different DCNNs: the proposed ACNN, hybrid network [5], optimized U-Net [4] and U-Net [3], the bars at the negative Mean DSC axis indicate the DCNN that achieves the highest mean DSC for that patient while the bars at the positive Mean DSC axis show the mean DSCs achieved by different DCNNs.

4.3.5 Comparison with Other Methods

The segmentation accuracy, optimal learning rate, and parameter number of the four different DCNNs: the proposed ACNN, hybrid network [5], optimized U-Net [4] with FGN and U-Net [3] on segmenting the RV, LV and aorta are shown in Tab. 4.3. The optimized U-Net achieves the highest DSCs for 5 cross validations including RV-1, LV-1, LV-2, LV-3, and Aorta-2, the proposed ACNN achieves the highest DSCs for 3 cross validations including RV-2, Aorta-1 and Aorta-3, the hybrid network achieves the highest DSC on 1 cross validation - RV-3. With achieving comparable DSCs to the other three methods, the proposed ACNN used less trainable parameters. This advantage is largely due to the efficiency of full-scale feature maps inside the proposed ACNN. Compared to the out-performed DCNN - optimized U-Net, the proposed ACNN also consumes less memory and training time for the RV and LV experiments (1.66G memory and 9.8s training time for 100 iterations for the proposed ACNN while 8.80G memory and 15.1s training time for 100 iterations for the optimized U-Net).

The mean DSC for each patient in the RV, LV and aorta dataset with the four different DCNNs as the segmentation methods is shown in Fig. 4.10. We can see that U-Net (yellow color) under-performs slightly than the other three methods, especially for the RV and aorta data.

Table 4.4: The mean and variance of the segmentation mean DSC of training the same model in six times, OLR - optimal learning rate, Var. - Variance, Aor - Aorta.

Test	RV-1	RV-2	RV-3	LV-1	LV-2	LV-3	Aor-1	Aor-2	Aor-3
Mean	0.651	0.705	0.654	0.912	0.861	0.807	0.834	0.783	0.845
Var.	0.019	0.015	0.017	0.004	0.019	0.009	0.016	0.016	0.008
OLR	0.5	1.5	0.05	0.05	0.5	0.5	0.5	0.1	0.05

4.3.6 Multiple Runs

The proposed ACNN was trained additionally five times (plus the one in Tab. 4.3, in total six times) for each cross validation. The mean and variance of six segmentation mean DSCs are shown in Tab. 4.4. We can see that the DSC variance is $< 2\%$, which is in the normal range - $1\% \sim 2\%$ stated in [97] and is comparable to the DSC variance - $0.97\% \sim 5.62\%$ when training U-Net in multiple times [4]. The average DSCs in Tab. 4.4 mostly are lower than the mean DSCs of the proposed ACNN in Tab. 4.3. This is normal, as the mean DSCs in Tab. 4.3 were optimized by training five different initial learning rates. This optimization would not cause unfairness, as it was applied to all other experiments as well.

4.4 Discussion and Conclusion

An atrous rate setting for determining the atrous rate at the n^{th} atrous convolutional layer as $(k)^{n-1}$ where k is the convolutional kernel size is proposed. It can achieve the largest and fully-covered receptive field with a minimum number of atrous convolutional layers. Comparison experiments with traditional atrous rate settings, i.e., (1, 2, 4, 8, ...) in [105], (1, 2, 5, 9, ...) in [107], or (1, 1, ..., 2, 2, ..., 4, 4, ...) in [106], are not conducted due to: 1) smaller receptive field resulted by traditional atrous rate settings would not definitely indicate lower segmentation accuracy, as a large receptive field may be redundant when the target is small; 2) in addition to the receptive field, complex factors, i.e., the link number of each input node and the trainable parameters influence the segmentation accuracy too. The complex reasons behind a good segmentation result make it difficult to judge the atrous rate setting from the segmentation accuracy. Hence, in this chapter, detailed mathematical proof and derivation are given.

Six atrous blocks: I-block, II-block, III-block, IV-block, V-block, VI-block with a receptive field of 3, 9, 27, 81, 243, 729 respectively are proposed and assessed. For an atrous block with a larger receptive field, i.e., VI-block, a fewer number of blocks and a fewer total number of atrous convolutional layers are needed to cover the whole input image. Under the network framework in this chapter, i.e., atrous block cascade, identity mapping, FGN, the experiments indicate that atrous II-block is optimal for medical image segmentation. However, if the network framework is changed or the task is changed, the optimal atrous block may be different. For transferring the atrous blocks proposed in this chapter to other works, additional exploration and assessment specific to the target task are essential to select the corresponding optimal atrous block.

Dense connection was shown to be efficient in [112]. In this chapter, it is not adopted due to its similar segmentation accuracy and high memory consuming. Identity mapping was shown to be an improvement of residual learning in [111]. In this chapter, it is not used due to its slightly lower robustness and stability. Finally, residual learning is used as the shortcut connection. BN, IN, LN and GN are the four most popular normalization methods used in medical image segmentation. It was shown in [4] that FGN is the optimal normalization method for U-Net structure. In this chapter, FGN also out-performs other normalization methods and is used.

The proposed ACNN achieves comparable segmentation accuracy with the hybrid network, optimized U-Net with FGN and U-Net, but with using less trainable parameters. We think this achievement comes from the efficient information contained in full-scale feature maps. This advantage is very useful when applying the trained model to mobile devices, as the trained model will require less memory. For data with a smaller image size, i.e., the RV and LV dataset with image size 256×256, the proposed ACNN also consumes less memory and training time. However, the consumed memory and training time increases significantly with the image size, i.e., the aorta dataset with image size 512×512. This would be further optimized with network architecture designs in our future work. Furthermore, target specific segmentation DCNNs are not compared in this chapter, i.e., Omega-Net proposed for cardiac MRI segmentation [114] and Equally Weighted Focal U-Net proposed for class-imbalance stent graft marker segmentation [21], as additional target-specific algorithms related to the target character is usually applied in these methods and hence these methods usually may not

be generalizable to other datasets.

Except Sec. 4.3.6, all the other accuracy shown was recorded from the first training only. For a fair comparison, five initial learning rates are explored for each experiment to avoid setting the learning rate less optimally. This process may indicate an optimized accuracy, as a sub-optimized training would not out-perform among the five trainings. However, this process would not cause unfairness, as it is the same for all experiments. In this chapter, only the mean and *std* of the DSC accuracy are presented, additional statistical values, i.e. p-value, would further enhance the validation in the future.

The shown training time is only for 100 iterations and under a clear computer process status. This time could be much longer when the computer and GPU are filled with other processes. In practice, the whole training time takes up to 16 hours to train one model. As five learning rates were tested for each experiment, it took up to 4 days to show one DSC in above tables and figures. This training speed is based on tf.layers and tf.contrib.layers programming and may be different if the implementation is programmed differently. Hyper-parameters, i.e. the momentum and optimizer are selected based on experience. Different results may exist if different hyper-parameter settings are utilized.

Based on the author's knowledge, all codes were optimized as much as possible. Further optimization may exist and may influence the recorded memory usage and training time. The applications of the proposed ACNN are not limited to medical image segmentation, but also could be expanded to natural image segmentation and other pixel-level tasks, which needs further detailed validations.

A new full-scale DCNN - ACNN is proposed with the use of cascaded atrous II-blocks, residual learning and FGN. A new atrous rate setting is proposed to achieve the largest and fully-covered receptive field with a minimum number of atrous convolutional layers. Six atrous blocks including I-block, II-block, III-block, IV-block, V-block, VI-block, three shortcut connections including residual learning, identity mapping, dense4 connection, and four normalization methods including BN, IN, LN, GN are assessed with detailed experiments to select the optimal method for the atrous block, shortcut connection and normalization layer. With less trainable parameters than that used in the hybrid network, optimized U-Net with FGN and U-Net, comparable accuracy is achieved by the proposed ACNN. The less parameters

needed in the proposed ACNN would contribute to the community, as transferring DCNN methods onto mobile devices and realizing real-time performances are the two common challenges faced by the current DCNN methods. This chapter contributes to some fundamental problems in DCNN with full-scale feature maps, i.e., the atrous rate setting, atrous block division, wider exploration and contribution from other researchers in the future would promote full-scale DCNNs.

5 A Real-time and Registration-free Framework for Dynamic 3D Shape Instantiation

[1]

Due to the respiration and cardiac beating, soft organs and tissues usually experience severe deformation and are dynamic during the surgery. Usually this deformation is complex and has no fixed pattern. It is less informative and insufficient to do an operation which is usually a 3D task under a 2D or static 3D navigation. Hence, in this chapter, I work on reconstructing the instantaneous 3D shape of an organ or tissue in real-time from its a single 2D view. This 2D view can be a 2D projection from fluoroscopy, 2D slice from MRI or 2D image from ultrasound. Based on my work in Chapter 3 and 4 - normalization and network architecture design for training Deep Convolutional Neural Network (DCNN) for medical image segmentation, whose segmentation results will be used as the input, in this chapter, the instantaneous 3D shape of an organ or tissue can be reconstructed with patient-specific learning.

5.1 Introduction

Current clinical systems for MISs, such as cardiac radio-frequency ablation, image-guided needle biopsies, and endovascular interventions, typically incorporate static 3D surfaces for guidance. Real-time dynamic tracking of 3D surfaces can help to optimize the interventional procedure, especially for complex anatomical structures undergoing gross tissue deformation, bulk organ motion, and potential topological changes during interventions.

[1]The contents of this chapter are published in [Xiao-Yun Zhou, Guang-Zhong Yang, and Su-Lin Lee. "A real-time and registration-free framework for dynamic shape instantiation." Medical image analysis 44 (2018): 86-97.]

A combination of multiple imaging modalities has been used for dynamic 3D navigation. For example, a real-time registration scheme based on both spatial registration and electrocardiography was proposed to overlay pre-operative 3D Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) volumes onto intra-operative 2D ultrasound images for dynamic 3D navigation [115]. 3D Transesophageal Echocardiography (TEE) was fused with 2D X-ray fluoroscopic images using image localization and calibration for dynamic cardiac navigation [116]. However, based on a combination of multiple imaging modalities, the dynamic 3D shapes were either interpolated from pre-operative 3D volumes or intra-operatively collected 3D volumes with low-resolution. A 3D shape recovery scheme based on intra-operative 2D images including X-ray, ultrasound, and MRI could take intra-operative information into account whilst achieving high-resolution at the same time. This kind of 3D shape recovery is termed dynamic shape instantiation. The scheme may or may not involve the use of template models [117]. Without template models used, more intra-operative information and longer image acquisition time are needed; for example, at least seven intra-operative 2D images were needed for reasonable 3D prostate reconstruction [66]. In this chapter, a single intra-operative 2D view is targeted and hence we focus on template-based 3D shape instantiation.

For template-based 3D shape instantiation methods, Statistical Shape Model (SSM) [118], Free Form Deformation (FFD) [119], and Laplacian surface deformation [120] can be used for the representation of templates. SSM [121] is a popular technique which represents a set of 3D meshes or 2D contours with the same number of vertices and connectivity. SSM-based 3D shape instantiation learns from shape variations rather than only applying smoothness and 2D/3D similarity as the constraints. It deforms an initial 3D SSM to match intra-operative sparse inputs such as ultrasound-derived surface points [122], digitized landmarks [123], or two or more calibrated X-ray images [124]. These methods usually learn a model from a training set of anatomies of multiple patients and deform the learned model for a new patient, which requires a high anatomical similarity between patients. This learning is not suitable for patients with anatomical anomalies, as these patients have different anatomical shapes. For example, patients who have undergone liver resection have a significantly different liver shape to other subjects. A possible solution for these specific cases has been proposed

in [70]. Here, limited optimal scan planes were determined by analyzing the pre-operative and patient-specific 3D SSM of the liver with Principal Component Analysis (PCA). The relationship between pre-operative 3D SSM and synchronized 2D SSM constructed from 2D images at the optimal scan planes was learned by Partial Least Squares Regression (PLSR). Finally, with new intra-operative 2D images obtained at the same scan planes, the 3D shape was instantiated intra-operatively by applying the PLSR-derived relationship. However, in [70], the optimal scan plane determination depended on the selected vertices that were deemed informative but were highly correlated and clustered. PLSR can only derive linear relationships while the deformations of most anatomies are non-linear. Based on [70], a framework which achieves more accurate, robust, generalizable and convenient shape instantiation from a single intra-operative 2D view is proposed in this chapter.

Subspace reprojection was proposed to determine an optimal scan plane for SSM-based 3D shape instantiation by fitting a plane to the most informative vertices [125]. This optimal scan plane was shown to have enhanced accuracy compared to other scan planes [125]. By applying PCA [126] on the pre-operative 3D SSM, the informative vertices which contribute most to the shape variations are determined by the loadings of principal components [70]. The downside of using PCA is that the derived principal components are linear combinations of multiple variables and therefore the selected informative variables are highly related and difficult to interpret. This phenomenon when reflected in our application is that the selected informative vertices are clustered area and are not the real and independent informative vertices. Many methods have been proposed to solve this issue, including rotation methods [127], limited set of integers [128], and Simplified Component Technique Least Absolute Shrinkage and Selection operator (SCoTLASS) [129]. Simple thresholding of the loadings is a common and informal method usually used in practice [70]; however, this method lacks theoretical support and usually causes problems [130]. Recently, Zou *et al.* proposed Sparse PCA (SPCA) which reformulated PCA into a regression-type optimization problem and then added a L1 constraint to achieve sparse loadings; they demonstrated improved performance of SPCA in selecting the real informative variables over previous methods [1]. A SPCA toolbox was later developed [131].

PLSR is a linear regression method that finds the linear relationship between two matrices: predictors and responses. It finds the principal

components in predictors that can explain the principal components in responses maximally by projecting the predictors and responses into a new space. It has a similar prediction accuracy to Ridge Regression (RR) and Principal Component Regression (PCR) [132]. It is more widely used than RR and PCR in medical problems, such as cardiac motion prediction [133] and craniofacial reconstruction [134], as it is more suitable for problems with a larger number of variables and fewer number of observations [135]. However, its accuracy for non-linear motions is limited. Many non-linear PLSR variations have been proposed and they can be divided into two groups [136]: the first group reformulates the linear relationship into a non-linear one by polynomial functions, smoothing splines, artificial neural networks, and radial basis function networks while the second group maps the original variables into a higher dimensional space and regresses the mapped variables in the higher dimension, for example, kernel space. Kernel PLSR (KPLSR) [135] from the second group is adopted in this chapter for improved computation speed as its formulation is as time-efficient as PLSR and avoids the non-linear optimization in the first group.

In this chapter, the high-resolution 3D shape of a dynamic anatomy was instantiated from a single intra-operative 2D view in real-time. Firstly, the anatomy was scanned by MRI or CT pre-operatively for multiple 3D volumes along the dynamic cycle and a 3D SSM was constructed. SPCA was applied on the pre-operative 3D SSM to select the informative vertices which were used to fit an optical scan plane. Local adjustments of the scan plane parameters for better accessibility, visibility or satisfying other local constraints is possible without incurring major errors, as the later KPLSR-based 3D shape instantiation scheme is robust to optimal scan plane derivations. Secondly, 2D projections or slices synchronized with the pre-operative scanning were obtained at the approximate optimal scan plane and were sampled to generate a synchronized 2D SSM. KPLSR was applied to learn the relationship between the pre-operative 3D SSM and the synchronized 2D SSM. Finally, the high-resolution 3D shape was instantiated intra-operatively by applying the KPLSR-derived relationship onto a new intra-operative 2D projection or slice at the same scan plane. The overall framework of the proposed dynamic shape instantiation is illustrated in Fig. 5.1. Due to the learning of patient-specific models, the framework is applicable to any anatomy. No extra registration is needed for the pre-operative 3D

SSM and the synchronized 2D SSM. Validation was performed on the liver (two digital liver phantoms, one dynamic liver phantom, one in vivo porcine liver, eight metastatic livers) and the cardiac RV (18 asymptomatic RVs and 9 Hypertrophic Cardiomyopathy (HCM) RVs); we anticipate that potential applications of our work will include percutaneous liver biopsy, cardiac catheterization [137], and intra-myocardial therapy [138]. For example, in cardiac ablation, the instantiated 3D RV shape can be used to help navigate the catheter tip to the target ablation area.

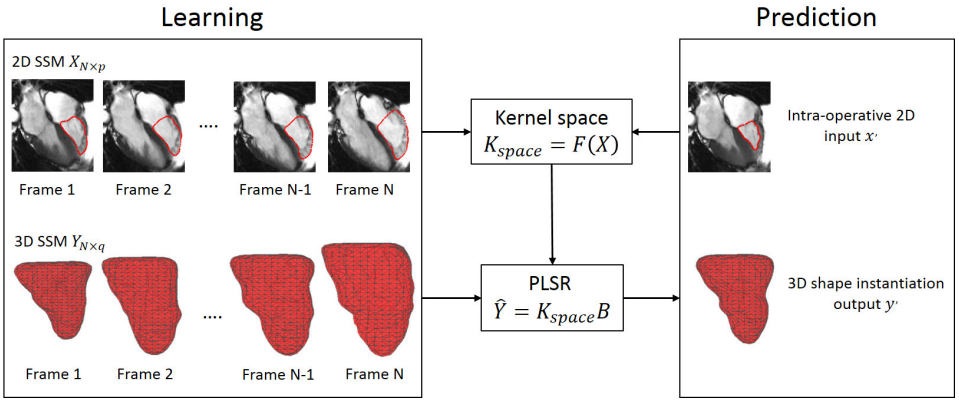


Figure 5.1: A schematic illustration of the overall framework of the proposed dynamic shape instantiation scheme: both the 2D projections or slices in the learning and prediction are taken at the approximate optimal scan plane; the learning 2D SSM and learning 3D SSM are not registered but synchronized.

5.2 Methodology

The proposed 3D shape instantiation framework mainly include three steps. First, SPCA is used to determine the optimal scan plane for the single 2D projection or slice by analyzing the 3D pre-operative SSM. This scan plane is usually informative and can capture the most information regarding the deformation of the target. Second, KPLSR is used to learn the relationship between the 2D SSM and 3D SSM. This learning process learns the relationship between two matrices with the dimension of $N \times p$ and $N \times p$ respectively. Third, this learned KPLSR model is used to instantiate intra-operatively a new 3D shape based on a new 2D projection or slice input. This new 2D projection or slice is also acquired at the optimal scan plane. The methods for

determining the optimal scan plane are described in Sec. 5.2.1. The learning and instantiation based on KPLSR are described in Sec. 5.2.2. Finally, the data collection and detailed validation experiments are in Sec. 5.2.3.

5.2.1 Optimal Scan Plane Determination

By pre-operatively scanning the target anatomy with CT or MRI, a 4D volume consisting of multiple 3D volumes at different time frames along the dynamic cycle of the anatomy was obtained. These 3D volumes were represented with 3D meshes using the same number of vertices and connectivity, which created a pre-operative 3D SSM (a point distribution model) with vertices $Y_{N\times numY\times 3}$, where N is the number of time frames and $numY$ is the number of vertices. By rearranging the (x, y, z) coordinates of the vertices as independent variables, $Y_{N\times q}$ was obtained, where $q = numY \times 3$ is the number of variables. Without loss of generality, $Y_{N\times q}$ was centered and normalized as Y_{norm} with the mean and norm of each column as 0 and 1.

For data Y_{norm}, its singular value decomposition is $Y_{norm} = UDV^T$, where $Z = UD$ are the principal components and V are the loadings of the principal components. The ith principal component $Z_i, i \in (1, N)$ represents the ith mode of variation in the anatomical deformation while the corresponding loadings V_i represent the contribution of each variable to this mode of variation [125]. The V_i calculated by PCA are usually all non-zero values and hence the selected informative vertices are highly related and clustered. The aim of SPCA is to achieve a sparse V_i. V_i can be recovered by:

$$\hat{\beta}_{ridge} = arg\ \ min_{\beta}\|Z_i - Y_{norm}\beta\|^2 + \lambda\|\beta\|^2 + \lambda_1\|\beta\|_1 \tag{5.1}$$

where, $\frac{\hat{\beta}_{ridge}}{\|\hat{\beta}_{ridge}\|} = V_i$, λ is a manually set positive parameter, $\|\beta\|_1 = \sum_{j=1}^{q}|\beta_j|$ and λ_1 is a manually set parameter which controls the sparsity or the number of non-zero values of $\hat{\beta}_{ridge}$. Equ. 5.1 can be solved with a fixed λ and any λ_1 by Least Angle Regression Elastic Net (LARS-EN) efficiently [139].

However, Equ. 5.1 is still based on PCA due to the inclusion of Z_i. To release this dependency, a two-stage exploratory analysis was formulated with PCA initialization and then optimization with sparse approximations.

With y_i - the ith row of Y_{norm}:

$$(\hat{\alpha}, \hat{\beta}) = arg \ \ min_{\alpha,\beta} \sum_{i=1}^{N} \|y_i - \alpha\beta^T y_i\|^2 + \lambda\|\beta\|^2 \tag{5.2}$$

When $\|\alpha\|^2 = 1$, then $\hat{\beta} \propto V_1$; the detailed proof can be found in [1]. If the first k principal components and the lasso penalty are included, Equ. 5.2 becomes

$$(\hat{A}, \hat{B}) = arg \ \ min_{A,B} \sum_{i=1}^{N} \|y_i - AB^T y_i\|^2 + \lambda \sum_{j=1}^{k} \|\beta_j\|^2 + \sum_{j=1}^{k} \lambda_{1,j}\|\beta_j\|_1 \tag{5.3}$$

Here, $A_{q\times k} = [\alpha_1, ..., \alpha_k]$ are the loadings of the first k principal components of PCA, with the restriction of $A^T A = I_{k\times k}$. Then $B_{q\times k} = [\beta_1, ..., \beta_k]$ are the approximated sparse loadings of $V_{1:k}$ [139].

The complete SPCA algorithm is listed in Table 5.1. The approximated sparse loadings $\hat{V}_j$ is a $q \times 1$ matrix with the loading or contribution of each variable to the jth principal component or mode of variation. The parameter $\lambda_{1,j}$ controls the sparsity or the number of non-zero values in $\hat{V}_j$. As suggested in [125], the contribution of three coordinates (x, y, z) at $\hat{V}_1$ was added together to represent the vertex contribution. The vertices at all time frames with non-zero contribution were selected as the informative vertices. A plane with the minimum sum of distances to all informative vertices was determined as the optimal scan plane. When calculating the sum, each distance was weighted by the vertex contribution. For multiple scan planes, $\hat{V}_j, j \in (2, N)$ can be used to determine the jth optimal scan plane; however, this is out of the scope of this chapter as we are targeting a single scan plane.

In [70], the real scan planes were registered to the optimal scan planes. In this chapter, as the proposed KPLSR-based 3D shape instantiation is robust to local scan plane deviations, which will be shown in Sec. 5.3.2, the final scan plane is an approximate one that is both accessible and convenient for imaging with parameters near the optimal scan plane. When the deformations or shapes of the anatomy are significantly different between patients and hence there are significantly different optimal scan planes between patients, such as

Table 5.1: SPCA [1]

SPCA
Initialize $A = V[:, 1:k]$:
the loadings of the first k principal components from PCA
Initialize $B_{q\times k} = [\beta_1, ..., \beta_k] = 0$
For j=1:k
If $\|\beta_j^{new} - \beta_j^{old}\| > criterion$, which has not converged
Solve the following minimum by LARS-EN:
$\beta_j = arg\ min_\beta(\alpha_j - \beta)^T Y_{norm}^T Y_{norm}(\alpha_j - \beta) + \lambda\|\beta\|^2 + \lambda_{1,j}\|\beta\|_1$
Update B with the normalized new β_j
Update A with the normalized new α_j:
$\alpha_j = (1 - A[:, 1:j-1]A[:, 1:j-1]^T)Y_{norm}^T Y_{norm}\beta_j$
End
End
Approximated sparse loadings $\hat{V}_j = \beta_j, j = 1, ..., k$.

the metastatic liver after oncological surgery, the optimal scan plane needs to be determined on a patient-specific basis. When the deformations or shapes of the anatomy are similar between patients and hence there are similar optimal scan planes between patients, such as the RV, the trend of the optimal scan planes for multiple patients is determined as a general optimal scan plane for the anatomy and will be used directly in subsequent interventional procedures, thus reducing the workload for clinicians. Detailed optimal scan plane determination and approximation in our practical experiments are given in Sec. 5.2.3.

The optimal scan plane is a suggestion. For MRI and ultrasound, it is possible to acquire an image slice at a scan plane. While for fluoroscopy, it is difficult to acquire a image at a scan plane, as only projections are available. Based on my experience, in this situation, I suggest to select a projection with a clear contour and maximum cross-section area as the replacement.

5.2.2 3D Shape Instantiation

With the pre-operative 3D SSM and the approximate optimal scan plane obtained, 2D projections or slices synchronized with the time frames for pre-operative scanning were obtained at the approximate optimal scan plane. The 2D anatomical contours were segmented and sampled to the same number of 2D vertices and connectivity, resulting in a 2D SSM with vertices $X_{N\times numX\times 2}$,

where N is the number of time frames and $numX$ is the number of vertices. By rearranging the (x, y) coordinates of vertices as independent variables, $X_{N\times p}$ was obtained, where $p = numX \times 2$ is the number of variables and typically $p \neq q$. The 3D volumes and 2D projections or slices do not need to be registered. KPLSR is then applied to learn the relationship between the 3D SSM which is the response in regression and the 2D SSM which is the predictor in regression. For 3D shape instantiation, the new intra-operative 2D projection or slice is obtained at the same scan plane and is sampled into the same number of vertices and connectivity as that for the original 2D SSM with vertices $x'_{numX\times 2}$. $x'_{1\times p}$ is obtained by rearranging the (x, y) coordinates as independent variables for applying the KPLSR-derived relationship to predict the intra-operative response $y'_{1\times q}$ whose (x, y, z) coordinates are then rearranged back to obtain the intra-operative 3D shape $y'_{numY\times 3}$.

In the following sections, we introduce PLSR and show its extension to KPLSR.

PLSR

PLSR extracts the relationship between two matrices which could have different dimensions. With predictors $X_{N\times p}$ and responses $Y_{N\times q}$, PLSR finds the relationship between them:

$$\hat{Y}_{N\times q} = X_{N\times p} B_{p\times q} \tag{5.4}$$

As PLSR is a regression and optimization process, approximation of Y rather than Y is achieved. Here, $\hat{Y}$ is the approximate prediction of Y. The difference between $\hat{Y}$ and Y depends on the criteria that user set. The latent variables in X are determined by $B_{p\times q}$ to explain the latent variables in Y maximally. $B_{p\times q}$ is later used to predict the intra-operative response $y'_{1\times q}$ from $x'_{1\times p}$. Non-linear Iterative partial Least Squares (NIPALS) is a widely applied PLSR algorithm [135]. In this chapter, an alternative algorithm - SIMPLS [140] - was used for increased time-efficiency.

Without loss of generality, both $X_{N\times p}$ and $Y_{N\times q}$ are centered with the mean of each column as 0, which are expressed by X_0 and Y_0 respectively below. The main problem for SIMPLS is to compute the weight factors r_i and d_i, where $i \in (1, M)$ and M is a manually set parameter denoting the number of components used. r_i and d_i maximize the covariance of $t_i = X_0 r_i$

Table 5.2: SIMPLS

SIMPLS
Initialize $S_0 = X_0^T Y_0$, X_0, Y_0 are the centered matrix of X, Y respectively
for $i = 1 : M$ (M is a manually set parameter - the number of components used)
if $i == 1$
r_i = first left singular vector of SVD of S_0, $(r_i - weights)$
else
r_i = first left singular vector of SVD of $S_0(I_p - C_{i-1}(C'_{i-1}C_{i-1})^{-1}C'_{i-1})$
end
$t_i = X_0 r_i$ $(t_i - scores)$
$c_i = X_0^T t_i/(t_i^t t_i)$ $(c_i - loadings)$
end
Coefficient: $B_{p\times q} = RT^{-1}Y_0$, where $R = [r_1, r_2 ... r_M]$, $T = [t_1, t_2 ... t_M]$

and $u_i = Y_0 d_i$ with the following four conditions:

1. maximized covariance: $u'_i t_i = d'_i (Y'_0 X_0) r_i = maximum$,
2. normalized r_i: $r'_i r_i = 1$,
3. normalized d_i: $d'_i d_i = 1$,
4. orthogonalized t: $t'_j t_i = 0, i > j$

To satisfy the fourth condition, $t'_j t_i = t'_j X_0 r_i = (t'_j t_j) c'_j r_i = 0$, where $c_j = X'_0 t_j/(t'_j t_j)$. When $i > 1$, any new r_i must be orthogonal to $C_{i-1} = [c_1, c_2 ... c_{i-1}]$. This orthogonal projector is $I_p - C_{i-1}(C'_{i-1}C_{i-1})^{-1}C'_{i-1}$, where I_p is an identity matrix. The SIMPLS algorithm is listed in Table 5.2. For more details of SIMPLS, please refer to [140].

KPLSR

PLSR is less suitable for regressing non-linear motions. KPLSR was used to compensate for this shortage. A kernel function maps the predictor $X_{N\times q}$ into a new feature space F non-linearly with $\Phi : x_i \in R^q \rightarrow \Phi(x_i) \in F, i \in (1, N)$. Φ satisfies the *kernel trick*: $\Phi(x_i)^T \Phi(x_j) = K(x_i, x_j)$. PLSR is then constructed in the feature space F to achieve a non-linear regression for X [135].

The kernel used in this chapter was a Gaussian kernel for its increased accuracy over a polynomial kernel:

$$K_{space} = exp(-K/W) \tag{5.5}$$

Here $K_{(i,j)} = K_{(j,i)} = (x_i - x_j)^2, i, j \in (1, N)$. W, the Gaussian width, was adjusted to $Ratio \times maximum(K_{N \times N})$ to facilitate parameter adjustment between different targets and subjects, $Ratio$ is a manually set ratio which we term the Gaussian ratio. After the Gaussian kernel, a new matrix K_{space} is acquired from X while contains more informative information than X due to the non-linear Gaussian kernel process. With the input as X, Tab. 5.2 is a normal PLSR process, while replacing the input X to be K_{space}, Tab. 5.2 allows PLSR to regress more informative and non-linear relationship, indicating a KPLSR process. For more details regarding the proof, please refer to [135] while for more details regarding the programming, please refer to Xiao-Yun Zhou's github.

5.2.3 Data Collection and Validation

The proposed framework was validated on both liver and cardiac RV studies. The experiments included two digital liver phantoms, one dynamic liver phantom, one in vivo porcine liver, eight livers from metastatic patients, 18 cardiac RVs from asymptomatic subjects, and 9 cardiac RVs from HCM patients.

The acquisition of 3D meshes and synchronized 2D contours at different time frames along the dynamic cycle for each data are given in Sec. 5.2.3 - 5.2.3. All data used the same methods to construct the 3D and 2D SSM. With known 3D shapes consisting of 3D vertices and connectivity at different time frames, the mid-state 3D mesh was first projected to meshes at other time frames by non-rigid registration [141]. Then the registered mid-state 3D mesh was mapped onto meshes at other time frames by projecting its vertices along the normal directions. Therefore a 3D SSM with point correspondences was constructed. With known 2D contours consisting of 2D vertices and connectivity at different time frames, the construction of a 2D SSM was in the same way as that for a 3D SSM but with a different registration method [142].

Digital Livers

XCAT is a digital whole body phantom with detailed, high-resolution and dynamic tissues [143] as shown in Fig. 5.2a and Fig. 5.2b. In this chapter, the isotropic resolution of the volume was set at $0.625mm$. 21 time frames were

collected between exhalation and inhalation. 3D meshes of two XCAT livers (one male and one female) were manually segmented and processed with Analyze (AnalyzeDirect, Inc, Overland Park, KS, USA) and MeshLab [144]. A 3D SSM was constructed for each digital liver.

The optimal scan plane for each liver was determined with approximately 200 informative vertices and was used to slice the meshes in the 3D SSM. The intersection contours were projected onto the slicing plane to simulate 2D contours. A 2D SSM was constructed for each liver.

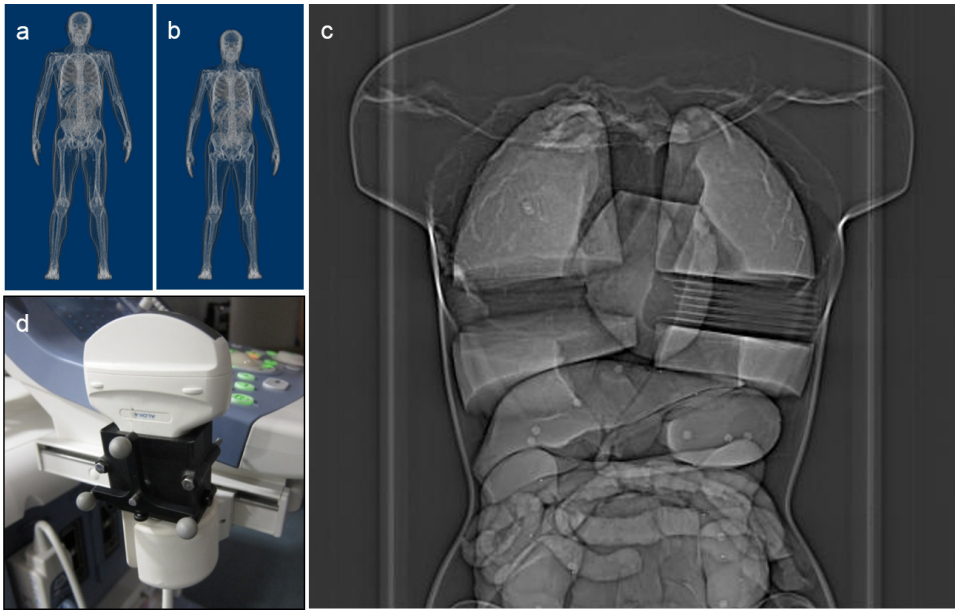


Figure 5.2: The digital livers and phantom experiment setup: (a) the male digital phantom, (b) the female digital phantom, (c) an X-ray image of the Regina phantom, whose lungs have been modified to simulate different respiratory positions, (d) the custom designed tracking frame based on a Polaris tracker mounted on the ultrasound transducer.

Dynamic Liver Phantom Experiment

A detailed female phantom modeled with silicone organs (the Regina model [145]) was used. The lungs were modified to simulate respiratory motion. In each lung, foam board inserts (each $5mm$ thick) were used, creating seven different liver deformation positions. Each respiratory position was scanned in a Siemens 64 slice SOMATOM™ Sensation CT Scanner with

images of $0.77mm \times 0.77mm$ in-plane resolution and $1mm$ slice separation. Segmentation and processing were performed with Analyze and MeshLab.

For real-time scanning, ultrasound imaging was used. A 2D imaging transducer used with the ALOKA prosound $\alpha 10$ system (Aloka Co. Ltd, Tokyo, Japan) was affixed with an NDI Polaris passive infrared tracker (Northern Digital, Inc, Waterloo, ON, Canada), enabling the recording of the spatial position and orientation of the scan plane. Calibration between systems was established by registering three known landmarks on the liver phantom in both frames of reference. The ultrasound images were captured from the S-video output feed of the scanner. The experiment setup is shown in Fig. 5.2c and Fig. 5.2d.

Freehand 3D ultrasound systems require calibration and a number of techniques and corresponding phantoms have been developed for this [146]. To calibrate the ultrasound images to the coordinate space of a tracking device, a three-point crossed wires phantom was built. The transforms from the coordinate space of the optical tracker to that of the CT imaging space were calculated by PRAXIS [147]. This defined a translation and a quaternion for the rotation between the ultrasound image points and the CT imaging space [148]. The mean distance between the registered ultrasound image points and the 3D meshes scanned by CT is less than $10^{-2}mm$.

Due to the constraints caused by the rib cage, the optimal scan plane fitted with 30 informative vertices was selected as the actual scan plane. The silicone phantom was filled with water. For each respiratory position, the optimal scan plane was acquired with the ultrasound probe. An experienced operator scanned the phantom using an in-house guidance system where the silicone liver was registered to a 3D guidance mesh by three manually chosen points. This guidance system provided the actual scan plane in real-time as well as the desired scan plane orientation.

A semi-automatic segmentation based on active contours [149] was used to delineate the liver contour from the 2D ultrasound images. It could determine a contour in an ultrasound images automatically when the two end points were selected manually. The contours were transformed to the CT coordinate frame to achieve registered contour coordinates from which a registered 2D SSM was constructed. The registration between 3D volumes and 2D images was only performed for the Regina phantom for later specific comparison and was not performed for all other data.

Porcine Liver

One contrast enhanced 3D CT scan was captured at full exhalation using a GE Innova 4100 interventional X-ray machine capable of fluoro-CT imaging. Due to the respirator design, the porcine liver could not be stopped at different respiratory positions for a 3D CT scan. Instead, fluoroscopic projections were obtained in an Anteroposterior (AP) direction over time to cover the animal's respiratory motion. As only one 3D volume at full exhalation was scanned with CT, 3D volumes at other respiration positions were simulated by image constrained Finite Element Modeling (FEM) [150] while the collected fluoroscopic projections were used as the image constraints. This created multiple liver 3D meshes at different time frames. In this chapter, the surface mesh at full exhalation was first turned into a tetrahedral mesh using Gmsh [151]. Then, the Open Source SOFA framework [152], chosen for its emphasis on real-time medical simulations, was used for the FEM. The material for the liver was set to be elastic and isotropic, with a Young's modulus of 640 Pa and Poisson's ratio of 0.3 [153]. A 3D SSM was constructed for the porcine liver.

The meshes in the 3D porcine liver SSM were sliced by the optimal scan plane determined with approximately 200 informative vertices. The sliced contours were projected onto the slicing plane with 2D coordinates to simulate 2D contours. A 2D SSM was constructed for the porcine liver as well.

Metastatic Livers

Clinical data from eight patients (6 male, 2 female, mean age 63) with metastatic liver tumors was collected. 4D volumes were scanned using a 1.5T MRI scanner (Intera, Philips, Amsterdam, Netherland) using a T1 weighted free-breathing sequence (TR = $7.83ms$, TE = $2.24ms$, $3.5mm \times 3.5mm$ in-plane resolution, $4.5mm$ slice thickness). Each volume consisted of 45 slices and was acquired in approximately $1.2s$. 60 time frames were collected to cover the liver motion during respiration. Due to motion artifacts caused by respiration, we could only confidently segment the livers at full inhalation and full exhalation. As before, the SOFA framework was used to generate the meshes at different respiratory positions but with the 3D volumes at full inhalation and full exhalation as the constraints. These meshes were used to construct a 3D SSM for each patient.

The eight metastatic patients have significantly different liver shapes and deformations and the optimal scan planes for each patient were very different. For this reason, for the metastatic liver, the optimal scan plane was determined patient-specifically with approximately 50 informative vertices for each patient and this was used to slice the meshes in the 3D SSM. The sliced contours were projected onto the slicing plane with 2D coordinates to simulate 2D contours. A 2D SSM was constructed for each patient.

Cardiac Data

27 subjects (18 asymptomatic subjects (Subjects 1-18) and 9 patients with Hypertrophic Cardiomyopathy (HCM) (Subjects 19-27)) were scanned with a 1.5T MRI scanner (Sonata, Siemens, Erlangen, Germany). HCM was selected as it is one of the diseases that influence both the shape and deformation of the heart significantly. Short-axis cine sequences from the atrioventricular ring to the apex were scanned with a $10mm$ slice gap and a $1.5-2mm$ pixel spacing. $19-25$ time frames were collected. To recognize the slice location of the atrioventricular ring and the apex, the $10mm$ slice gap was interpolated to $1mm$ in Analyze. 3D RV meshes were segmented and built with Analyze and MeshLab. A 3D SSM was constructed for each patient.

Even though HCM influences both the shape and deformation of the RV, the optimal scan planes for the 27 subjects, which were determined with approximate 150 informative vertices each, were mostly found to be along the long axis of the heart. Four examples are shown in Fig. 5.3. Even though the optimal scan planes in Fig. 5.3 are not exactly the same, they still share the same trend - lying along the long axis of the heart. This similarity of the optimal scan planes between patients is mainly due to the similarity in deformation and shape of the RVs between patients. As later KPLSR-based 3D shape instantiation is robust to optimal scan plane deviations, we made an adjustment to the optimal scan plane to ensure the accessibility of the scan plane and the visibility of the RV considering the following three issues: 1) The long-axis is accessible for 2D MRI, 2) The horizontal (four-chamber) long-axis has a clear view of the RV without overlap with other chambers, and 3) Clinicians are familiar with this plane as it features the apex and the atrioventricular ring. For these reasons, the horizontal (four-chamber) long-axis plane was selected as the actual scan plane for all RVs.

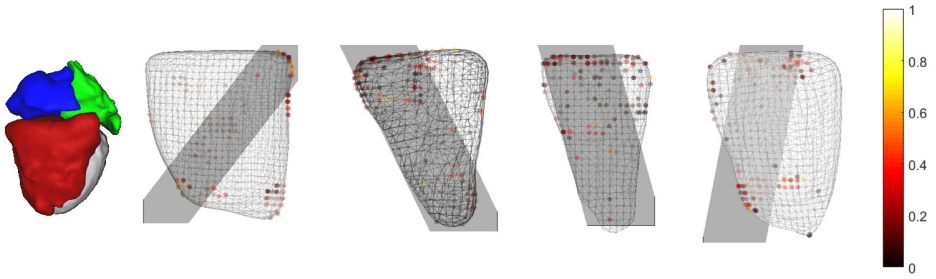


Figure 5.3: Four RVs, with optimal scan plane determination using the 150 most informative vertices: the vertices are colored by their normalized importance determined by SPCA and the grey plane is the optimal scan plane, with the overall view direction shown on the left hand side. The red/blue/green/grey chambers are the right ventricle/right atrium/left atrium/left ventricle, respectively.

2D MRI slices at the horizontal (four-chamber) long-axis plane with the synchronized time frames as that for the pre-operative 4D volume scanning were obtained for all 27 subjects. Analyze was used to segment the RV contours and a 2D SSM was constructed for each subject.

Validation

Leave-one-out cross validation was applied for all time frames for all data. The $ith, i \in (1, N)$ time frame in the 2D SSM was left out as a new predictor while the ith time frame in the 3D SSM was left out as the ground truth. All other time frames were used in the learning. The error was calculated as the Euclidean vertex-to-vertex distance between the 3D prediction and the ground truth. The shape variation was calculated as the mean vertex-to-vertex distance between the $(i-1)th$ and the $(i+1)th$ time frame in the 3D SSM.

It was shown that SPCA was able to better select the real and unrelated informative variables than PCA on a synthetic example [1]. For the synthetic example, the contribution of a variable and the relations between variables were known. However, for practical data, both this contribution and the relations were unknown; a comparison of the distribution of the informative vertices selected by PCA and SPCA is given in this chapter. In practice, adjusting the optimal scan plane is usually necessary for better scan plane accessibility and target visibility. To illustrate that this adjustment will

not incur major errors, multiple deviated optimal scan planes were used to instantiate the 3D shape.

Both PLSR and KPLSR regress the relationships between two matrices rather than two coordinate frames. Lee *et al.* applied PLSR with registration of pre-operative 3D SSM and synchronized 2D SSM [70]. In this chapter, this explicit registration between the 2D SSM and 3D SSM is not required. To demonstrate this, both the registered and non-registered 2D SSM of the dynamic phantom liver were used as the predictor for dynamic shape instantiation. The stability of an algorithm with respect to its parameters is important for judging its performance. PLSR has one parameter, the number of components used, while KPLSR has two parameters, the number of components used and the Gaussian ratio. To evaluate the stability of PLSR and KPLSR to the number of components used, the validation was applied on two HCM RVs with the number of components used set from $1-18$. In practice, the time frames at or near the boundaries, i.e. at maximal inhalation and exhalation or at diastole and systole, are the most difficult time frames to recover, as the learning is more weak for these time frames. We term these time frames *boundary time frames*. In this chapter, the performance of PLSR and KPLSR at boundary time frames were validated on two cardiac RVs (one asymptomatic RV and one HCM RV). The liver data was collected along half of the dynamic cycle - the first and last few time frames are the inhalation and exhalation respectively, i.e. the boundary time frames. The cardiac data was collected along the entire dynamic cycle, the first and last few time frames are at diastole while the middle few time frames are at systole, i.e. the boundary time frames.

Finally, the accuracy of the proposed dynamic shape instantiation was tested on two digital livers, one in vivo porcine liver, eight metastatic liver patients, and 27 RVs of asymptomatic subjects and HCM patients.

5.3 Results

The results from our experiments are shown in the following sections. The comparison between PCA and SPCA on selecting informative vertices is demonstrated in Sec. 5.3.1. The robustness of the KPLSR-based 3D shape instantiation to scan plane deviations is shown in Sec. 5.3.2. The validation on releasing the registration between pre-operative 3D SSM and synchronized

2D SSM is illustrated in Sec. 5.3.3. The stability of PLSR and KPLSR to the number of components used is compared in Sec. 5.3.4. Boundary time frames are tested in Sec. 5.3.5. Finally, the accuracy of the proposed dynamic shape instantiation is validated on the liver and the heart, which is shown in Sec. 5.3.6.

5.3.1 Comparison between PCA and SPCA

For most subjects, including the metastatic livers and cardiac RVs, it was found that the informative vertices selected by PCA were more clustered than the informative vertices selected by SPCA. Three examples are shown in Fig. 5.4. Clustered informative vertices were selected by PCA due to their related motion with the informative vertices considered to be in the same area. SPCA can remove this inter-relation and only select the true and sparse informative vertices.

5.3.2 Robustness to Scan Plane Deviations

To demonstrate the robustness of the proposed KPLSR-based 3D shape instantiation to scan plane deviations, example RV results from Subject 3 are illustrated below. 13 scan planes with some deviations from the optimal scan plane were used to slice the pre-operative 3D SSM for 3D shape instantiation. The distance error and the deviation for each scan plane is shown in Fig. 5.5a and Fig. 5.5b respectively. We can see that the achieved accuracy was scarcely influenced by local scan plane deviations, demonstrating the robustness of the proposed KPLSR-based 3D shape instantiation to scan plane deviations. This is important for practically implementing the proposed framework, as due to practical constraints (access window, or other local, physical constraints), it may be necessary to deviate slightly from the theoretical optimal scan plane. Such deviation should not induce large changes in instantiation errors.

5.3.3 Validation of Registration-Free Instantiation

The instantiation accuracy across all time frames with registered and non-registered predictors which were collected in the liver phantom experiment is shown in Fig. 5.6. It can be seen that PLSR is influenced by the registration while KPLSR shows little influence, demonstrating that explicit registration is not required in the proposed method.

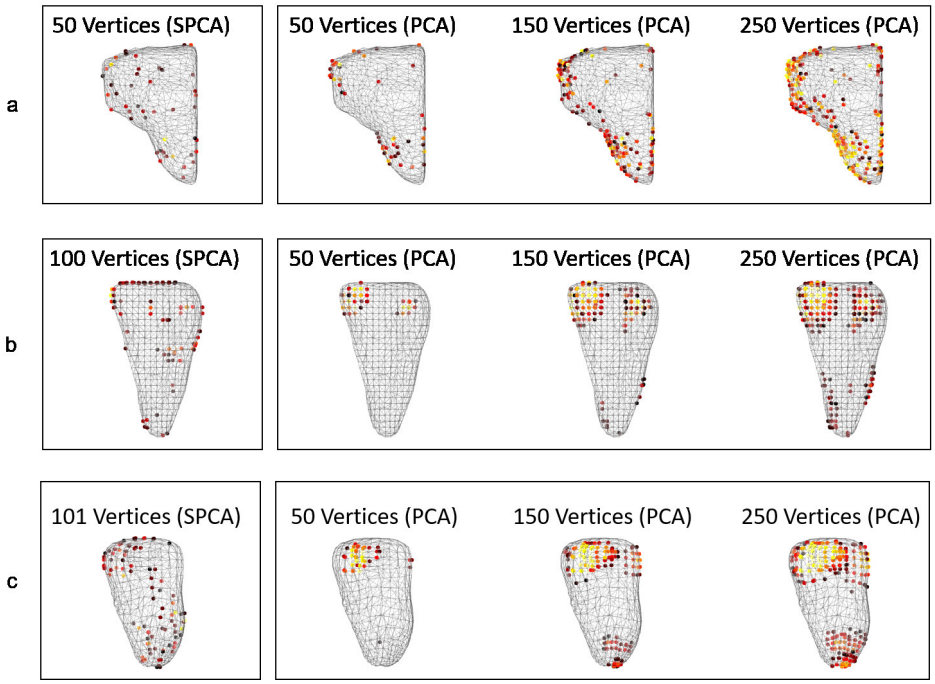
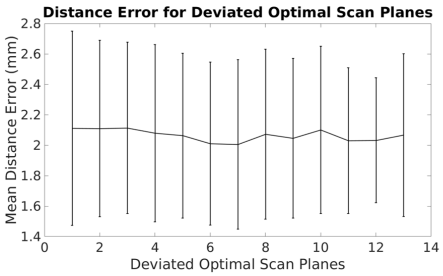


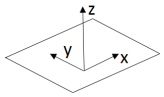
Figure 5.4: One liver and two RV examples showing the most informative vertices selected by SPCA and PCA: (a) a metastatic liver with 50 informative vertices determined by SPCA while 50, 150, 250 informative vertices determined by PCA, (b) an asymptomatic RV with 100 informative vertices determined by SPCA while 50, 150, 250 informative vertices determined by PCA, (c) a HCM RV with 101 informative vertices determined by SPCA while 50, 150, 250 informative vertices determined by PCA. The view directions for RVs and vertex coloring are in the same way as that in Fig. 5.3



a

Plane	Transformation	Plane	Transformation
1	(0,0,6)	8	$(-5^{\circ},0,0)$
2	$(0,10^{\circ},0)$	9	$(0,-5^{\circ},0)$
3	$(10^{\circ},0,0)$	10	$(0,0,-3)$
4	(0,0,3)	11	$(-10^{\circ},0,0)$
5	$(0,5^{\circ},0)$	12	$(0,-10^{\circ},0)$
6	$(5^{\circ},0,0)$	13	(0,0,−6)
7	(0,0,0) – No Deviation		

b



c

Figure 5.5: Testing the robustness of the proposed KPLSR-based 3D shape instantiation to scan plane deviations: (a) the mean distance error of the 3D shape instantiation with deviated optimal scan planes, with standard deviation calculated across 20 time frames, (b) the deviations of the scan planes. Even though a plane could have six transformations, three of them (rotation along the z axis, translation along the x axis and translation along the y axis do not influence the slicing results. The other three transformations were explored. For example, $(0, 0, 6)$ means rotating 0° along the x axis, rotating 0° along the y axis, and translating $6mm$ along the z axis, (c) illustration of the x, y, z axes of a plane.

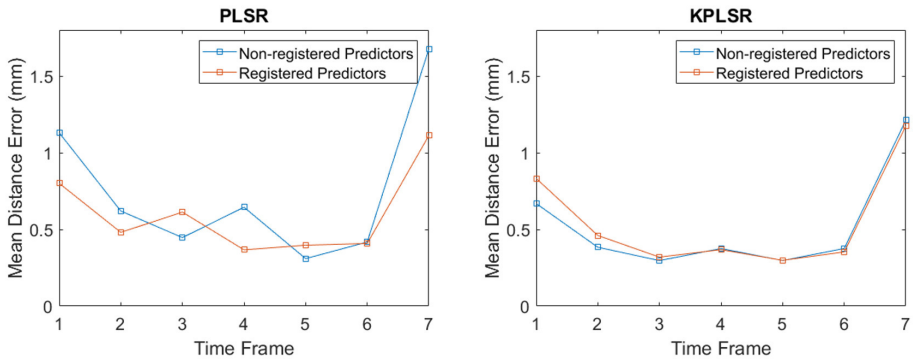


Figure 5.6: The instantiation accuracy for the liver phantom experiment: (left) the mean distance errors of PLSR with registered and non-registered predictors, (right) the mean distance errors of KPLSR with registered and non-registered predictors.

5.3.4 Stability to the Number of Components Used

Instantiation for two HCM patients (Subject 21 and Subject 27) was calculated along all time frames with a varying number of components used $(1-18)$, as shown in Fig. 5.7a and Fig. 5.7d. It can be seen that the accuracy of KPLSR is less sensitive to this parameter - the number of components used - than that of PLSR, as the standard deviations of KPLSR are less than that of PLSR at most time frames. Two time frames (time frame 5 of Patient 21, time frame 9 of Patient 27) are shown with the mean distance errors at different numbers of components used in Fig. 5.7b and Fig. 5.7e. Two instantiation examples colored by the distance errors are shown in Fig. 5.7c and Fig. 5.7f. The error is distributed evenly over the mesh.

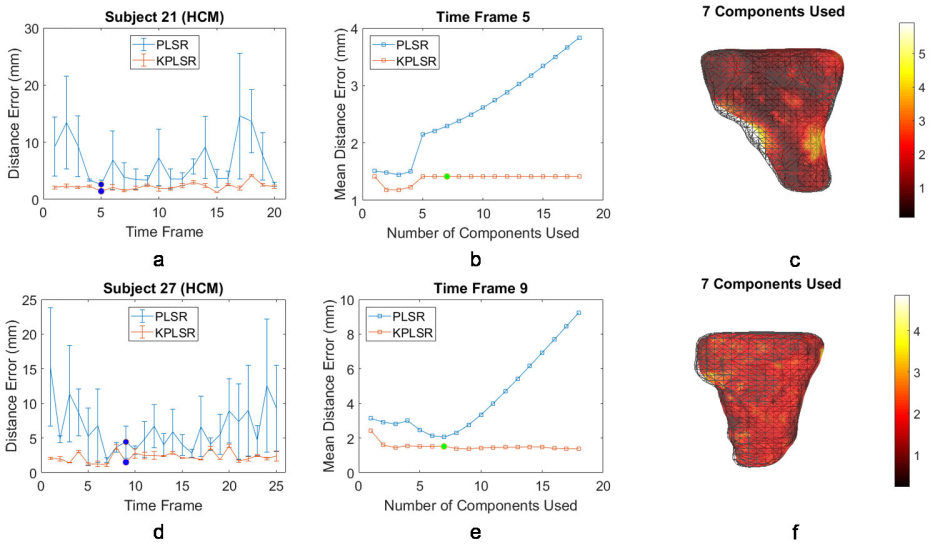


Figure 5.7: Testing the influence of the number of components used on PLSR and KPLSR: (a) the mean ± std errors for Subject 21, with the standard deviation calculated across $1-18$ components used, (b) mean distance errors with numbers of components used varying from $1-18$ for time frame 5 of Subject 21 (labeled with blue dots in a), (c) a shape instantiation example colored by the distance errors with 7 components used for time frame 5 of Subject 21 (labeled with green dot in b), with the same view direction in Fig.5.3, d,e,f are the same as a,b,c but for Patient 27, time frame 9, 7 components used respectively.

5.3.5 Performance at Boundary Time Frames

The mean distance errors for shape instantiation along all time frames are shown for two selected subjects (Subject 6 (asymptomatic) and Subject 19 (HCM)) in Fig. 5.8a. The PLSR errors show large peaks near systole (time frame 10 for Subject 6, time frame 11 for Subject 19) and diastole (time frame 1 and 25 for Subject 6, time frame 1 and 20 for Subject 19) while KPLSR errors show smaller increasing errors at these boundary time frames. It can be concluded that KPLSR has better performance at boundary time frames than PLSR.

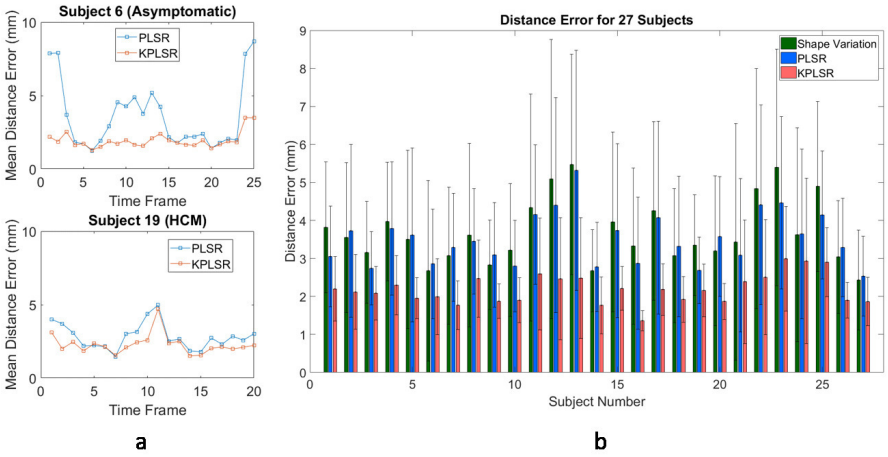


Figure 5.8: Results at the boundary time frames and for the RV experiments: (a) performance test for boundary time frames, (b) the instantiation errors for 27 subjects (Subjects 1-18 = asymptomatic subjects; Subjects 19-27 = HCM).

5.3.6 Accuracy of Dynamic Shape Instantiation

Mean distance errors of PLSR and KPLSR and the shape variation of two digital phantom livers and one porcine liver are shown along all time frames in Fig. 5.9. For the two digital livers, KPLSR achieved much lower errors at the time frames where PLSR showed high peaks. For the porcine data, the accuracy of KPLSR is higher than that of PLSR at most time frames. For both digital phantom and porcine liver studies, the mean distance error of KPLSR is much lower than the shape variation at most time frames. The peaks for KPLSR (time frames 18-19 in Fig. 5.9b and time frames 1-2 in

Fig. 5.9c) were caused by boundary time frames. The higher accuracy of KPLSR for the two digital livers is not as obvious as that for the porcine liver due to the design and linear deformation of the digital phantom.

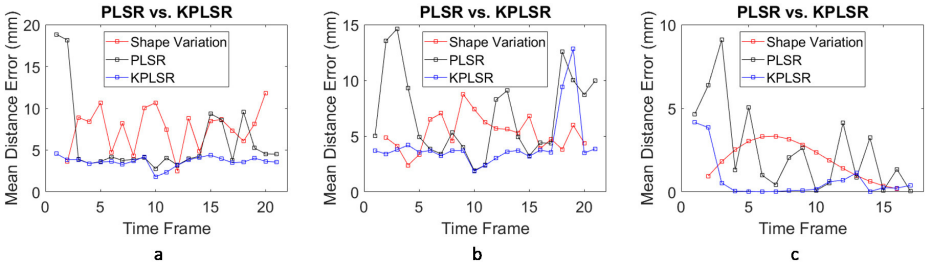


Figure 5.9: The mean distance errors and the shape variation of the two digital livers and the porcine liver: (a) the mean distance errors and the shape variation for the female digital liver, (b) the errors and shape variation for the male digital liver, (c) the errors and shape variation for the porcine liver.

Eight patients with metastatic tumors were used for instantiation validation with the mean distance errors of PLSR and KPLSR and the shape variation shown along all time frames in Fig. 5.10. For most of the time frames and patients, KPLSR achieved much more accurate instantiation results than those of PLSR. The mean distance errors of KPLSR were also much lower than the shape variation. The higher errors of KPLSR (time frames 29-30 for P1, time frames 1-3 and 29 for P4, time frame 22 for P5, time frame 14 for P7) were caused by boundary time frames.

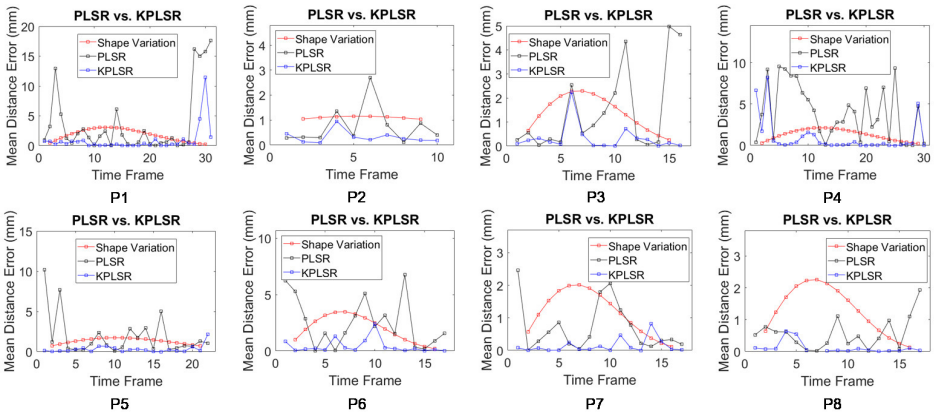


Figure 5.10: The mean distance errors and the shape variation for the eight metastatic livers.

Shape instantiation of 27 RVs was validated with the mean distance errors of PLSR and KPLSR and the shape variation shown in Fig. 5.8b; the standard deviation in the graphs was from the error variation along different time frames. Overall, KPLSR achieves both lower mean and standard deviation errors in the instantiation than PLSR for all subjects. The error achieved by KPLSR was also much lower than the shape variation for all subjects. The similar results between patients also demonstrate the availability of using one approximate optimal scan plane - the horizontal (four-chamber) long-axis plane for all RVs in this chapter.

For optimal scan plane determination, the number of informative vertices was determined as $5\% - 10\%$ of the total number of vertices in each test mesh, the parameter λ was fixed at 0.0001, k was set at 1 as we are targeting a single scan plane, and the parameter λ_1 was set as the number of informative vertices. For 3D shape instantiation, with the exception of the test for stability to the number of components used, all tests were validated with the number of components used for PLSR optimized between $1 - 8$ while that for KPLSR was empirically set between $1 - 18$. Overall, KPLSR achieved better accuracy at a higher number of components used than PLSR. The Gaussian ratio parameter of KPLSR was selected empirically.

Experiments were performed in MATLAB on an Intel(R) Core(TM) i7-4790 CPU @3.60Hz computer. The training took approximately $1s$ for one component deflation; the number of component deflations is the number of components used. The prediction or shape instantiation took approximately $1ms$.

5.4 Discussion and Conclusion

In this chapter, SPCA was applied instead of PCA to determine the informative vertices to find the optimal scan plane. We expect that nearby points on the surface of organs will tend to move dependently in a similar fashion. This is because the movement of one cell will cause the movement of its nearby cells due to the connectivity of tissues. The sparse informative vertices determined by SPCA and the clustered informative vertices determined by PCA could illustrate the ability of SPCA to derive principal components from unrelated original variables and hence select the true, unrelated informative vertices. However, from our experiments, the overall trend of the informative vertices

selected by PCA was shown to be similar to the trend determined by SPCA. It is more reasonable to conclude that SPCA facilitates the determination of the optimal scan plane more clearly and quickly than PCA rather than more accurately in this case. Setting a higher number of informative vertices when applying PCA could also achieve a good scan plane.

In practical applications, the calculated optimal scan plane is not always accessible. The robustness of the proposed KPLSR-based 3D shape instantiation to local scan plan deviations ensures the adjustment of the scan plane for better accessibility and visibility in practical clinical scenarios. The optimal scan plane for the RV, which will be used directly for future patients, was determined by analyzing the pattern of the optimal scan planes for 27 RVs. This method of determining the optimal scan plane could be adopted for other anatomies which share similar deformation and shape across patients. For anatomy such as the metastatic liver which has significantly different deformation and shape between patients, the optimal scan plane has to be determined for each patient individually.

The registration between the pre-operative 3D SSM and synchronized 2D SSM is no longer required in this chapter. The validation on a liver phantom experiment with both registered and non-registered predictors showed that the accuracy of KPLSR was not influenced by this. The removal of explicit registration will decrease the workload for clinicians significantly when applying the proposed method in practice. It was shown that KPLSR had much higher stability to the number of components used than PLSR. This is important for practical applications in case of the use of a suboptimal setting of this parameter. KPLSR also had better processing at boundary time frames than PLSR though the errors of KPLSR at boundary time frames are still higher than at other normal time frames. This boundary limitation corresponded to more time frames for the liver data than the cardiac data, as the SOFA framework generated meshes at the first few and last few time frames with very small shape variations which were usually less than $0.3mm$. In practical applications, always including the time frames at maximum inhalation and exhalation or at systole and diastole in the training data is highly recommended.

As pre-operative 4D volumes are not typically acquired for livers, FEM was applied to simulate the meshes between the inspiration and expiration. FEM or any other methods which could simulate the physical organ motion

can thus be used to generate pre-operative 4D volumes when transferring the proposed framework onto other target anatomies whose dynamic motion is difficult to gate.

In general, three kinds of data are needed to apply the proposed 3D shape instantiation: the 3D SSM for learning, the 2D SSM for learning, and the 2D intra-operative projections or slices for prediction. Synchronization is needed between the learning 3D SSM and the learning 2D SSM while registration is needed between the learning 2D SSM and the 2D intra-operative projections or slices for prediction. The 4D volume used for constructing the learning 3D SSM was scanned pre-operatively while the 2D projections or slices used for constructing the learning 2D SSM could be scanned pre-operatively or intra-operatively, as the learning only takes a few seconds. In practical applications, for organs whose motion could be gated easily, i.e. the RV, the synchronization between the learning 3D SSM and the learning 2D SSM could be achieved through dynamic motion gating, i.e. electrocardiogram (ECG) gating or respiratory gating. The registration between the learning 2D SSM and the intra-operative 2D projections or slices for prediction could be achieved by setting the scan machine at the same scan position. For organs whose motion is difficult to gate, i.e. the liver, FEM or other available methods which could simulate the 3D volumes at different time positions could be used to collect the learning 3D SSM and to slice for the learning 2D SSM. The registration between the learning 2D SSM and the intra-operative 2D projections or slices for prediction could be achieved by setting the scan machine to the same scan position as that used to slice the learning 2D SSM.

Two digital livers, one porcine liver, and eight metastatic livers were used to illustrate the applicability of the proposed method on livers. For patients after liver resection, monitoring is essential to see the growth of liver. This monitoring is usually achieved by frequent 3D CT scan, which causes radiation. With the 3D shape instantiation framework proposed in this chapter, this monitoring can be done with a single 2D MRI scan, leading both decreased radiation and time. As well, 27 RVs were used in our validation with real 2D MRI slices as the predictors, which demonstrates the potential value of the proposed method in practical operations. Even with only a single scan plane, a mean distance error of about $2.19mm$ was achieved for the RV. This error was comparable to the mean accuracy in [116] and [115] which were approximately $2.83mm$ and $3.55mm$ for patients and animals,

respectively. The computation time for prediction ($1ms$) demonstrates the real-time ability of the proposed method.

In conclusion, a real-time and registration-free framework for dynamic shape instantiation which is generalizable to multiple anatomies is proposed in this chapter. SPCA is applied to select the unrelated and real informative vertices from a pre-operative 3D SSM, which facilitates a more clear and quick determination for the optimal scan plane. KPLSR is used to improve the accuracy and robustness of the instantiation. For anatomies like the RV, the optimal scan plane only needs to be determined once and then can be used in subsequent interventions. The detailed experiments performed for the removal of explicit registration, the stability to the number of components used, and the performance at boundary time frames covers the issues which may occur during practical applications. FEM extends the application of the framework to anatomies like the liver, whose motion is difficult to gate. The patient-specific learning removes the restrictions on the applicable anatomy. This chapter sets the basis for applying the proposed framework to other interventional procedures involving dynamic anatomies. This chapter only considers the deformation caused by respiration and cardiac beating. In the future, a consideration of the instrument insertion during the surgery would be very helpful.

6 3D Shape Instantiation for Real-time Stent Graft Deployment

[1]

In chapter 5, I introduced my work on a general 3D shape instantiation framework. The training data collection and KPLSR-based learning are patient-specific, allowing its application to multiple soft organs with the required training data available. However, for medical devices, i.e. stent graft, the deployment or deformation has an easier and fixed pattern which can be decomposed into multiple rigid transformations. This kind of less complex shape deformation can be reconstructed more intuitively by combining multiple rigid reconstruction components, rather than using learning based methods. In this chapter, I will transfer my work on 3D shape instantiation specifically to deployed stent grafts.

6.1 Introduction

Endovascular Aortic Repair (EVAR), for the treatment of Abdominal Aortic Aneurysm (AAA), involves the insertion of compressed stent grafts via the femoral artery, advancement through the vasculature, subsequent device deployment, and exclusion of the aneurysmal wall. Blood flow is re-established through the deployed stent graft with reduced pressure on the diseased aneurysmal wall. The risk of rupture is abolished in the absence of endoleaks. For patients whose aneurysms involve or are adjacent to the renal and visceral

[1] The content of this chapter is based on [Xiao-Yun Zhou, Jianyu Lin, Celia Riga, Guang-Zhong Yang, and Su-Lin Lee. Real-time 3D shape instantiation from single fluoroscopy projection for fenestrated stent graft deployment. IEEE Robotics and Automation Letters 3.2 (2018): 1314-1321.] and [Xiao-Yun Zhou, Celia Riga, Su-Lin Lee and Guang-Zhong Yang. Towards automatic 3D shape instantiation for deployed stent grafts: 2D multiple-class and class-imbalance marker segmentation with equally-weighted focal U-Net. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018: 1261-1267.]

vessels, Fenestrated Endovascular Aortic Repair (FEVAR) is necessary; this includes the use of a fenestrated stent graft with fenestrations or scallops to allow perfusion of vital aortic branches and ensure optimum aneurysm exclusion [154]. A regular stent graft used in EVAR and a fenestrated stent graft used in FEVAR are shown in Fig. 6.1a and Fig. 6.1b, respectively. Each stent graft is composed of multiple stent segments, and the fabric between each two stent segments are graft gap. In addition to the location and size of fenestrations and scallops, the size and length of the stent graft are also customized according to patient-specific aortic geometries. An increasing number of stent graft manufacturers, such as Cook Medical (IN, USA) and Vascutek (Scotland, UK), are supplying fenestrated stent grafts today [155].

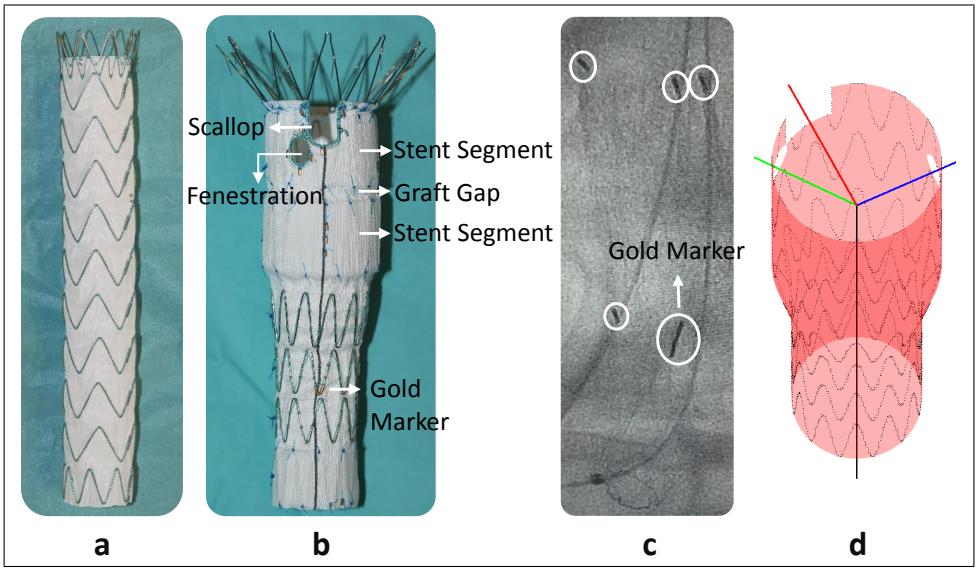


Figure 6.1: (a) a regular stent graft used in EVAR, (b) a fenestrated stent graft used in FEVAR with fenestrations, scallop and gold markers onside, (c) a fluoroscopic image example during FEVAR under normal radiation dose, (d) safe paths for robot-assisted vessel-fenestration cannulation. The black path is along the centrelline of the deployed main fenestrated stent graft while the green, blue and red path are from the black path end and aiming at the centers of the two fenestrations and the one scallop.

FEVAR is a challenging and complex procedure with multiple steps. The principal challenge is the alignment of the fenestrations or scallops with the target vessels. Selective cannulation of the target vessels through the

fenestrations, and subsequent branch stent graft delivery and deployment, are paramount to ensure successful aneurysm exclusion. This step can be challenging and time-consuming due to vessel tortuosity and angulation, leading to prolonged procedure and fluoroscopy time with a significant radiation burden to patients and operators [154]. Alternative cannulation strategies have therefore been explored such as robotic catheter systems aiming to improve navigational accuracy and stability. One commercially available system is the Magellan (Hansen Medical, CA, USA) which includes a master-slave catheter and guidewire driving system. Clinical experience with endovascular robotic systems is growing with potential advantages of increased accuracy, safety, and stability whilst minimizing the radiation exposure [156].

Despite advances in endovascular robotic technologies, navigation is still dependent on 2D fluoroscopy as shown in Fig. 6.1c. Both the stent and graft have poor visibility under fluoroscopy. High dosage fluoroscopy may improve the visualization, however, this will increase the radiation dose. To improve FEVAR navigation, markers are sewn onto the fenestrated stent grafts to indicate the position and orientation of the fenestrations and scallops (Fig. 6.1b). These markers are typically made of gold, have different shapes, and can be placed in various positions to aid in alignment of the device with the anatomy.

There has been previous research to improve stent graft deployment. Automatic detection and tracking of stent graft delivery devices from 2D fluoroscopic projections have been proposed [157], with Frangi filtering and robust principal component analysis. Optimized stent graft sizing and placement for pulmonary artery stenosis using cylindrical affine transformation and hill climbing have also been demonstrated [158]. A registration scheme combined with a semi-simultaneous optimization strategy that is to take the stent graft geometry into account was proposed to overlay 3D stent shapes onto 2D fluoroscopic images for navigation [159]. However, these methods have been demonstrated on regular off-the-shelf stent grafts for EVAR but have not taken into consideration fenestrations or scallops. Renal arteries and commercial markers have been highlighted on intra-operative fluoroscopic images to aid with stent graft deployment [160]; however, this is only in 2D and does not provide the 3D stent graft shape.

It is necessary to know exactly where fenestrations or scallops are to enable

a complete vessel-fenestration cannulation during FEVAR. A possible 3D navigation or robotic path is shown in Fig. 6.1d. The path travels along the centreline of the deployed main fenestrated stent graft (black path in Fig. 6.1d) and then is aimed at the center of corresponding fenestrations or scallops (green, blue, red path in Fig. 6.1d). In order to keep a minimum radiation dose during FEVAR, we aim to use a single fluoroscopic projection of several well-placed markers for 3D shape instantiation of the deployed main stent graft body. 3D shape instantiation in this chapter refers to 3D shape recovery but with only a single 2D fluoroscopy projection as the input.

After being deployed into an aneurysm, the stent graft may experience twisting, bending, rotation and translation with respect to its initial straight state, making 3D instantiation of its entire shape, orientation and deformation challenging. Most of these non-rigid deformations are caused by what we term the graft gap, shown in Fig. 6.1b, which is only made up of graft fabric. For the stent segments which include the metal stent and the graft attached on them, as shown in Fig. 6.1b, they tend towards their initial states closely due to their relative stiffness. Thus the deformation of the whole stent graft could be split into the rigid transformations of stent segments and the non-rigid deformations of graft gaps.

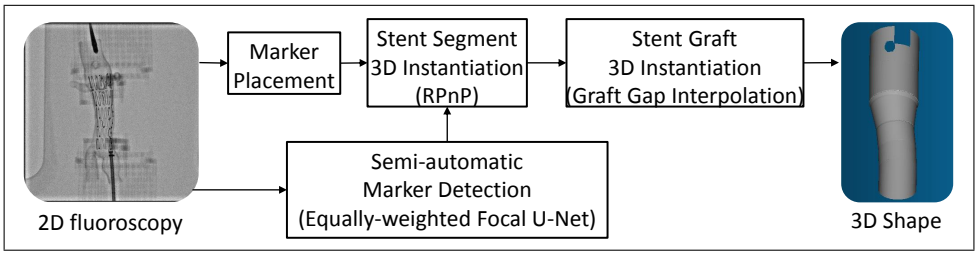


Figure 6.2: The proposed framework for real-time 3D shape instantiation of deployed fenestrated stent grafts.

We proposed a framework, as shown in Fig. 6.2, which reconstructs the 3D shape of a fenestrated stent graft from a single 2D fluoroscopic projection in real-time. First, five customized markers were placed on each stent segment of a fenestrated stent graft at different positions. Then, the rigid transformations of individual stent segments were calculated by the Robust Perspective-n-Point (RPnP) method [17] while the non-rigid deformation of the entire stent graft was reconstructed by graft gap interpolations. The proposed method was validated on five 3D printed AAA phantoms and three

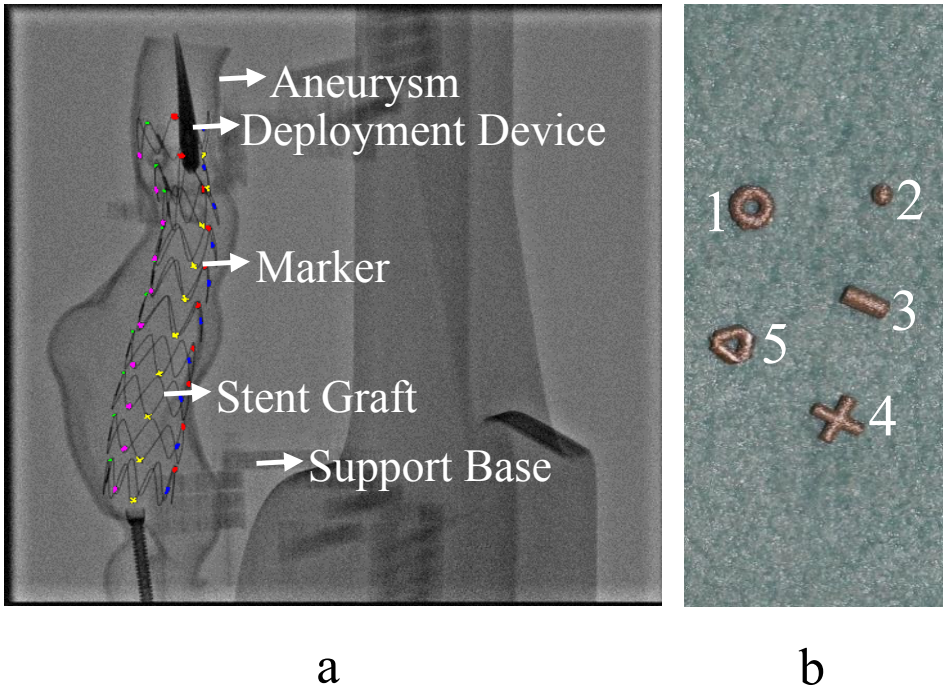


Figure 6.3: (a) an experimental fluoroscopic projection example with five markers - the red, green, blue, yellow, and purple color indicate marker 1, marker 2, marker 3, marker 4, and marker 5 respectively, this marker sequencing is valid across the whole chapter; (b) 3D printed customized markers.

stent grafts with newly placed markers, resulting in 78 images overall.

In order to improve the automation of the proposed 3D shape instantiation method, I further worked on marker detection. One experimental fluoroscopic projection with the customized markers labeled in different colors is shown in Fig. 6.3a. We use marker segmentation rather than marker detection to determine the marker center position, as segmentation is a pixel-level classification, offers more details and hence is more precise. There are two challenges in segmenting these customized markers into multiple-classes: 1) the markers are very small (the reason will be explained in Sec. 6.2.2), causing class-imbalance problems; 2) the markers are with similar appearances (the reason will be explained in Sec. 6.2.5).

Compared to conventional segmentation methods, deep convolutional neural network which extracts and classifies the features automatically with the

using of multiple non-linear modules has outperformed traditional methods in semantic segmentation. Fully Convolutional Network (FCN) was the very first proposed network which improved the image-level classification with CNN to a pixel-level classification with the using of fully convolutional layers, deconvolutional layers and skip architectures [40]. Ronneberger *et al.* firstly introduced FCN into biomedical segmentation and proposed U-Net on neuronal structure segmentation and cell segmentation [3]. The Deeplab series including Deeplabv1 [161], Deeplabv2 [5], Deeplabv3 [108], and Deeplabv3+ [162] with Atrous convolution, Atrous Spatial Pyramid Pooling (ASPP), and encoder-decoder modules were also popular networks in semantic segmentation.

Class-imbalance, where the background pixel number is much larger than the foreground pixel number, is a common challenging problem in semantic segmentation. Allocating large weights for the foreground pixels while allocating small weights for the background pixels were usually used to concentrate the training more on foreground pixels [3]. Three shortages exist when applying weighted loss in our application (will be proved in Sec. 6.3.1): 1) the weight needs to be manually set; 2) when the weight is too small, weighted loss could not distinguish between different foreground classes, while if the weight is too large, the background would be mis-classified as a foreground; 3) its performance is insufficient.

Two-stage networks were also widely explored in both medical and natural community to improve the network performance on small object or class-imbalance segmentation. Cascade Fully Convolutional Network (CFCN) was proposed to segment the liver first as a Region of Interest (RoI), and then another FCN was trained to segment the small lesion inside the liver RoI [163]. In Zhou *et al.*'s work, the pancreas was segmented firstly, and then the cyst inside the pancreas was segmented to improve the accuracy of the small cyst segmentation [164]. In general computer vision community, Mask Region-CNN (Mask R-CNN) was developed, where an object bounding box was regressed and classified firstly and then FCN was applied inside this bounding box [165].

Apart from improving the network structure and using two-stage networks, various researches have also been carried out on the loss function. Topology aware FCN was proposed with considering multi-region topological relationships and smooth boundaries into the loss function for histology gland

segmentation [166]. Convolutional AutoEncoder (CAE) was added to the loss function to consider the shape prior for semantic segmentation, which shown improved results in the kidney ultrasound image segmentation [167]. Recently, focal loss was introduced in the object detection domain, which added different scaling factors automatically to focus on training hard examples [103]. However, directly applying the focal loss in [103] into our application has three challenges: 1) the performance is insufficient (will be proved in Sec. 6.3.2); 2) it needs careful parameter initialization; 3) the weight used in [103] would introduce the same problems as stated before for the weighted loss.

In this chapter, Equally-weighted Focal U-Net was proposed. "Equally-weighted" means equal weight of 1 was applied to the foreground and the background. "Focal" means focal loss was used. The proposed method is a one-stage network but with two-step training, as shown in Fig. 6.4. First, U-Net with equally-weighted cross-entropy loss function was applied to segment a preliminary result. Second, U-Net with equally-weighted focal loss was used to improve the preliminary segmentation. It outperformed the focal loss in [103] and Weighted U-Net in [3] in: 1) the model trained by equally-weighted loss is used as the initialization for later equally-weighted focal loss, avoiding careful manual parameter initialization; 2) equally-weighted loss avoids the possible problems caused by weighted loss and also reduces one hyper-parameter - the weight; 3) even though equally-weighted loss underperforms weighted loss, the later equally-weighted focal loss will improve the preliminary segmentation result and outperform weighted loss. U-Net was selected as the network structure, as it is easy to be trained from scratch with limited training data (80 images in this chapter). The proposed Equally-weighted Focal U-Net was validated on 78 testing images, showing comparable results to the 3D shape instantiation based on manual detection.

6.2 Methodology

Stent graft modelling, 3D stent graft shape instantiation including marker placements, rigid transformation calculations of stent segments and non-rigid deformation instantiation of the whole stent graft, Equally-weighted Focal U-Net for semi-automatic marker detection, experimental setup, and data collection are introduced in this section.

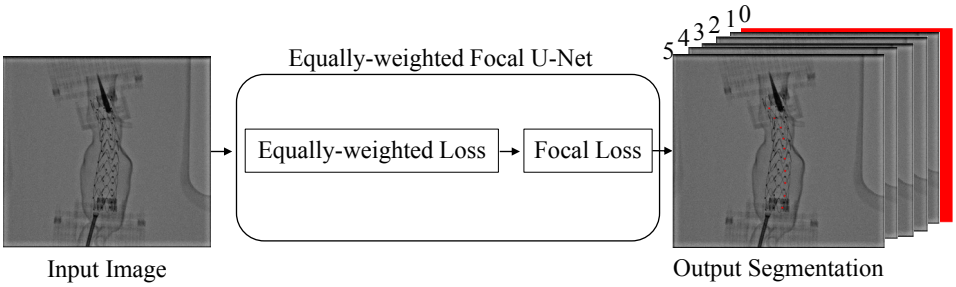


Figure 6.4: The framework of the proposed Equally-weighted Focal U-Net: the output map is consisted of six classes: class 0 represents the background, class $1-5$ represent the marker 1, marker 2, marker 3, marker 4 and marker 5. Red color indicates the pixels with probability of 1 in each output class.

6.2.1 Stent Graft Modelling

Previous work, i.e. [159], usually only focused on modelling the stents for EVAR. In FEVAR, the grafts are of equal or greater importance as fenestrations and scallops are on these grafts. CT could be used to acquire 3D stent shapes but not for grafts, due to the poor visibility of the fabric under CT. For fenestrated stent grafts, all parameters including the height, radius, gap, etc. are known via the original stent graft design and hence enable a mathematical modelling.

A stent graft is modelled with circles of different radii positioned at different heights. A circle vertex was defined by $\left[\begin{array}{ccc} r*cos\theta, & r*sin\theta, & h \end{array}\right]$, as shown in Fig. 6.5a. Neighboring vertices were connected by triangles regularly to generate a surface mesh. The resolution in the height was set as $1mm$ while that in the radial direction was set as 1° in this chapter. The accumulation of these circles made up the graft modelling. To model fenestrations and scallops, vertices within the fenestration or scallop were removed (Fig. 6.5b). $\left[\begin{array}{ccc} rcos(2\pi i/N_v), & rsin(2\pi i/N_v), & h'sin(2\pi iN_s/N_v)/2 \end{array}\right]$ was used to model the stent vertices [168], where $r = r_n + (r_x - r_n) * (h'sin(2\pi iN_s/N_v)/2 + h'/2)/h', i \in (1, N_v)$, N_v is the vertex number on a stent, N_s is the number of sine wave cycles describing the stent, h' is the height of each stent segment (Fig. 6.5c). For the example in Fig. 6.5c, $r_n = 11.5mm$, $r_x = 15mm$ and h' for the six stent segments from the bottom to top are $17mm$, $13mm$, $13mm$, $16mm$, $21mm$, $25mm$ respectively. In manufacturing, stents cannot lie across fenestrations or scallops and are forced onto fenestration or scallop edges; we

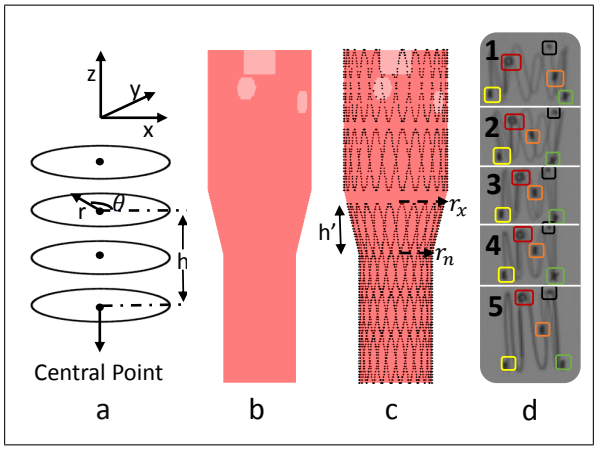


Figure 6.5: (a) modelling of circles, (b) modelling of graft, fenestrations and scallop, (c) modelling of a whole fenestrated stent graft, (d) marker placement and classification: markers are firstly classified into five types and then markers in each type are divided for each stent segment (five stent segments in this case).

modelled these crossed stents onto the nearest fenestration or scallop edges too.

6.2.2 3D Stent Segment Instantiation

The non-rigid deformation of the whole stent graft was split into multiple rigid transformations of stent segments in this chapter. The 3D pose of each stent segment was reconstructed based on the 2D fluoroscopic marker projections. By using the Robust Perspective-n-Point (RPnP) method, which estimates the pose of a calibrated camera given a set of n 3D points in the world coordinate system and their corresponding 2D projections in the image, the 3D pose of a stent segment could be reconstructed by the 3D pose of its n markers. Compared to the traditional 2D/3D registration, the RPnP method has the following advantages: 1) RPnP is fast and less ambiguous as it solves the 3D pose mathematically and non-interactively based on similar triangles; 2) RPnP only needs 4 points to reconstruct a reasonable 3D pose. The correspondences between 3D points and their 2D projections are supplied by marker placement and detection in this chapter.

Table 6.1: Marker Parameters

Marker Type	Circle	Sphere	Tube	Cross	Triangle
Marker Sequencing	1	2	3	4	5
Hole Radius (mm)	0.5	0.2	0.2	-	0.63
Thickness (mm)	0.8	0.8	0.8	0.8	0.8
Length (mm)	2.6	-	2.5	3	2.5

Marker Placement and Design

RPnP could achieve 3D pose recovery with a minimum $n = 4$ [17]. We adopted $n = 5$ for higher robustness. Five markers were sewn at five non-planar positions on each stent segment, as shown in Fig. 6.5d. The marker position pattern for each stent segment is similar. Stent graft markers were designed and inspired from commercially-used gold markers into five different shapes. The marker parameters are shown in Tab. 6.1. The lengths were designed to be similar to that of commercial markers which are around $1 - 3mm$. The thicknesses were empirically-determined for both minimized thickness and good imaging quality under lowest-radiation fluoroscopy. The shapes were designed with maximum differentiation and to be easily sewn onto the stents. Due to the high price of gold, these markers were printed on a Mlab Cusing R machine (ConceptLaser, Lichtenfels, Germany) with SS316L stainless steel powder for the experiment. The printed markers are shown in Fig. 6.3b. The small marker size caused class-imbalance. The five marker classes occupied 0.03%, 0.01%, 0.02%, 0.03%, 0.03% of the total pixels of the 512×512 fluoroscopic projection.

3D Pose Recovery for Stent Segment

For n markers on a stent segment with known reference 3D marker positions (via the original stent graft design): $\{P_1, ..., P_n\}$, after the compression and deployment, these 3D positions are transformed to target 3D marker positions: $\{P'_1, ..., P'_n\}$. With known corresponding 2D marker projections (via fluoroscopy projection): $\{p_1, ..., p_n\}$, the transformation matrix $\{P'_1, ..., P'_n\} = Tran \cdot \{P_1, ..., P_n\}$ could be recovered by solving a RPnP problem [17].

Firstly, a rotation axis was selected to reduce the number of unknown variables - here the Z axis was chosen. Secondly, the PnP problem was

divided into $(n-2)$ P3P(P3P is a special case of PnP where n is 3) problems with an equation system [169]:

$$f_i(x) = a_i x^4 + b_i x^3 + c_i x^2 + d_i x + e_i = 0, i \in (1, n-2) \tag{6.1}$$

where x was solved by the local minimum of $\sum_{i=1}^{n-2} f_i(x)^2$. Thirdly, the depth of each marker was determined by perspective similar triangles. Fourthly, the rotation along the Z axis with $c = cos\alpha, s = sin\alpha$ and translation $\begin{bmatrix} t_x, & t_y, & t_z \end{bmatrix}$ of the markers were solved by [17]:

$$\begin{bmatrix} A_{2n\times 2} & B_{2n\times 4} \end{bmatrix} \begin{bmatrix} c & s & t_x & t_y & t_z & 1 \end{bmatrix}^T = 0 \tag{6.2}$$

The derivation of $A_{2n\times 2}$ and $B_{2n\times 4}$ was explained in [17]. Finally, the solved transformation matrix was normalized by a standard 3D alignment based on least-squares estimation [170]. This normalized matrix is the 3D pose of the $n = 5$ markers and the corresponding stent segment. More details of the derivation, proof, and calculation can be found in [17, 169] and [170].

6.2.3 3D Stent Graft Instantiation

Continuous Constraints for Stent Segments

In theory, the RPnP method recovers both the position and pose accurately. In our experiments, the drifted markers, unsuitably-small delivery device and repeated stent graft compression and deployment (details explained in Sec. 6.3.4) caused non-rigid deformation between the reference and the target 3D marker positions. When the transformation between the reference and target 3D marker positions is non-rigid, errors will be introduced to the recovered position and pose. The position shift of stent segments influenced the continuity of the entire stent graft and was corrected by applying continuous constraints on the circle central points. The central points of all instantiated stent segments are aligned to that of the top stent segment, as briefly illustrated in Fig. 6.6.

Graft Gap Interpolation

After recovering the pose and correcting the position drift for each stent segment, the normal vectors and positions of graft gap circles were interpolated

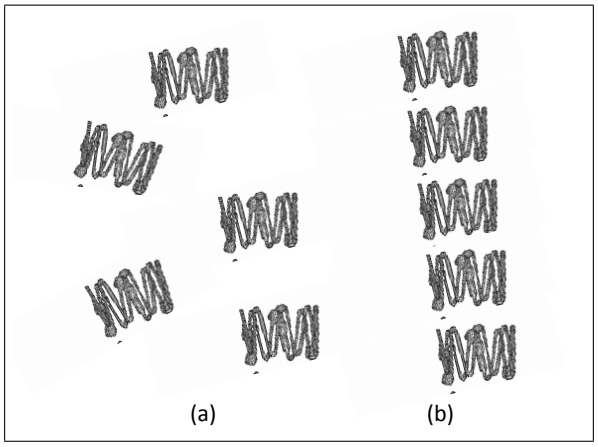


Figure 6.6: An illustration of the continuous constraint: (a) initially instantiated stent segments; (b) aligned stent segments after continuous constraint.

linearly by the normal vectors and positions of neighboring stent segment circles. With graft gap vertices $[r_i c_{\theta+T}, r_i s_{\theta+T}, 0]$, here $T \in (1°, 360°)$ controls the twisting and rotating of a circle, $\theta \in (1°, 360°)$ is the angle of a vertex, r_i is the radius, the interpolated graft gap vertices were calculated by:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \begin{bmatrix} r_i c_{\theta+T}, r_i s_{\theta+T}, 0 \end{bmatrix} \cdot \tag{6.3}$$

$$\begin{bmatrix} c_\Omega + \alpha^2 c_{\Omega p} & \alpha\beta c_{\Omega p} - \delta s_\Omega & \alpha\delta c_{\Omega p} + \beta s_\Omega \\ \alpha\beta c_{\Omega p} + \delta s_\Omega & c_\Omega + \beta^2 c_{\Omega p} & \beta\delta c_{\Omega p} - \alpha s_\Omega \\ \alpha\delta c_{\Omega p} - \beta s_\Omega & \beta\delta c_{\Omega p} + \alpha s_\Omega & c_\Omega + \delta^2 c_{\Omega p} \end{bmatrix}$$

where

$$c_{\Omega p} = 1 - c_\Omega \tag{6.4}$$

The rotation matrix rotates the normal vector of initial graft gap plane to be parallel to the interpolated one and was derived according to [171]. Here, $c_{\theta+T}$ represents $cos(\theta + T)$ and c_Ω represents $cos(\Omega)$. $s_{\theta+T}$ represents $sin(\theta+T)$ and s_Ω represents $sin(\Omega)$. Ω is the angle between the circle normal vector and the xy plane (the xy plane is shown in Fig. 6.5a). $\begin{bmatrix} \alpha, & \beta, & \delta \end{bmatrix}$ controls the bending and is the cross product of the circle normal and $\begin{bmatrix} 0, & 0, & -1 \end{bmatrix}$. $\begin{bmatrix} x_i, & y_i, & z_i \end{bmatrix}$ translates the rotated graft gap vertices to

the interpolated position.

6.2.4 Equally-weighted Focal U-Net

Given a training or testing data set $\{\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_k, ..., \mathbf{I}_\mathrm{K}\}, k \in [1, \mathrm{K}]$, where $\mathbf{I}_k$ is one image example with width W and height H, W = H = 512 in this chapter, K is the total number of images in the training or testing data set. The intensity of each pixel in $\mathbf{I}_k$ is normalized into $[0, 1]$ by: $\mathbf{I}_{norm_k} = \frac{\mathbf{I}_k - min(\mathbf{I}_k)}{max(\mathbf{I}_k)}$, where $min(\mathbf{I}_k)$ and $max(\mathbf{I}_k)$ are calculated from all images. The segmentation ground truth of $\mathbf{I}_k$ in the training data set is labelled as a labelling cube: $\mathbf{L}_k = \{\mathbf{L}_{k0}, \mathbf{L}_{k1}, ..., \mathbf{L}_{kn}, ..., \mathbf{L}_{k\mathrm{N}}\}, n \in [0, \mathrm{N}]$, where N is the number of marker classes, N = 5 in this chapter (Fig. 6.4), $\mathbf{L}_{kn}$ has the same width W and height H, $\mathbf{L}_{k0}$ is the background labelling layer with background pixels labelled as 1 and other pixels labelled as 0, $\mathbf{L}_{kn}$ is the n^{th} class foreground or marker labelling layer with the n^{th} class marker pixels labelled as 1 and other pixels labelled as 0. Since the markers are very small, those markers do not fully overlap each other frequently during the varying fluoroscopy view angle. Hence, it is reasonable to consider the multiple-class marker segmentation as a no-overlap problem, where one pixel only belongs to one class.

U-Net structure

According to the U-net structure [3], a normalized image $\mathbf{I}_{norm_k}$ is passed into the proposed network as an input, then a probability map cube $\mathbf{P}_k = \{\mathbf{P}_{k0}, \mathbf{P}_{k1}, ..., \mathbf{P}_{kn}, ..., \mathbf{P}_{k\mathrm{N}}\}, n \in [0, \mathrm{N}]$ is calculated, where $\mathbf{P}_{kn}$ is with the same width W and height H. The value of each pixel in $\mathbf{P}_{kn}$ is the probability of that pixel belongs to the n^{th} class and is between $[0, 1]$. The network structure used in this chapter is consisted of convolutional layers, max-pooling layers and deconvolutional layers, as illustrated in Fig. 6.7. It has two paths: a contracting path (left) and an expansive path (right). For convenience, we term the layers that manipulate on images with the same size as a block. In the contracting path, each block is consisted of two convolutional layers following by a max-pooling layer. In the expansive path, each block is consisted of two convolutional layers following by a deconvolutional layer. The last block is consisted of two convolutional layers, a 1×1 convolutional layer, a pixel-wise softmax layer. The network in Fig. 6.7 is defined as a

3-block U-Net, as three max-pooling/deconvolutional layers are used in total. In this chapter, the stride for the convolutional layer is always 1 while that for the max-pooling layer is always 2.

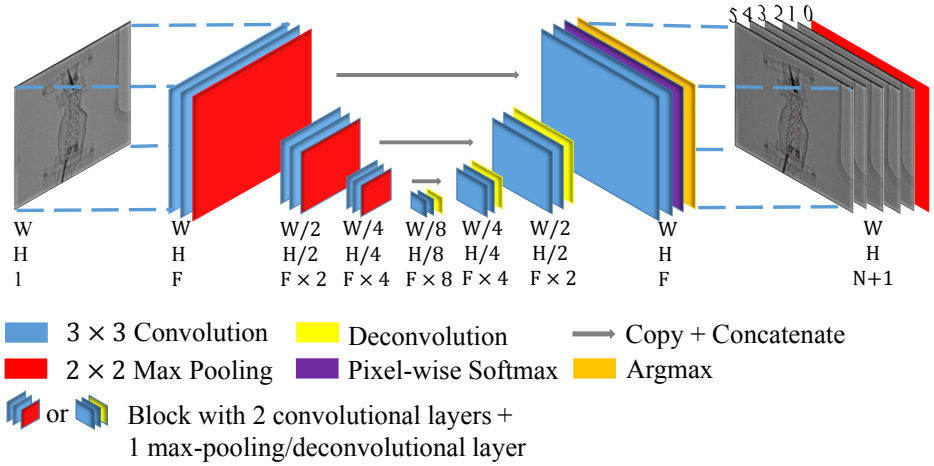


Figure 6.7: An illustration of a 3-block U-Net: three max-pooling or deconvolutional layers are used in total, two convolutional layers are used in each block, the width W and height H of the image are half/twice while the number of feature channel (F) is twice/half after each max-pooling/deconvolutional layer, $N = 5$ in this chapter.

Loss function

After passing $\mathbf{I}_{norm_k}$ through the U-Net, each pixel will have a U-Net-predicted value for the N+1 classes: $y_0, y_1, ..., y_n, ..., y_{\mathrm{N}}, n \in [0, \mathrm{N}]$. Pixel-wise softmax is used to transform y_n into the probability $p_n \in [0, 1]$ by:

$$p_n = \frac{e^{y_n}}{\sum_{i=0}^{\mathrm{N}} e^{y_i}} \tag{6.5}$$

Cross-entropy loss is calculated across the labelling and predicted probability cube to measure the difference between the predicted probability $\mathbf{P}$ and the ground truth $\mathbf{L}$:

$$CE_{loss} = -\sum_{i=1}^{\mathrm{W}}\sum_{j=1}^{\mathrm{H}}\sum_{n=0}^{\mathrm{N}} \mathbf{L}_{(i,j,n)} log(\mathbf{P}_{(i,j,n)}) \tag{6.6}$$

Usually, weighted loss was applied to solve the class-imbalance problem:

$$WCE_{loss} = -\sum_{i=1}^{\mathrm{W}}\sum_{j=1}^{\mathrm{H}}\sum_{n=0}^{\mathrm{N}} W_n \mathbf{L}_{(i,j,n)} log(\mathbf{P}_{(i,j,n)}) \tag{6.7}$$

Here, $W_0 = 1$ while $W_n > 1, n \in [1, \mathrm{N}]$. In this chapter, equally-weighted loss was applied for the first-step training. $W_n = 1, n \in [0, \mathrm{N}]$. When the loss converges to a minimum, equally-weighted focal loss was applied to improve the preliminary segmentation results:

$$Focal_{loss} = -\sum_{i=1}^{\mathrm{W}}\sum_{j=1}^{\mathrm{H}}\sum_{n=0}^{\mathrm{N}} (1 - \mathbf{P}_{(i,j,n)})^2 \mathbf{L}_{(i,j,n)} log(\mathbf{P}_{(i,j,n)}) \tag{6.8}$$

The scaling factor of $(1 - \mathbf{P}_{(i,j,n)})^2$ decreases heavily the loss contribution of correctly-segmented pixels (when $\mathbf{P}_{(i,j,n)} = 0.9, (1 - \mathbf{P}_{(i,j,n)})^2 = 0.01$). However, it decreases lightly the loss contribution of wrongly-segmented pixels (when $\mathbf{P}_{(i,j,n)} = 0.1, (1 - \mathbf{P}_{(i,j,n)})^2 = 0.81$). Thus the focal loss concentrates the training on wrongly-segmented pixels or hard pixels. In practice, fluoroscopic images are usually scanned in a coronal or oblique plane, which enables dividing the markers into the corresponding stent segment by their vertical positions manually, as shown by the white dividing lines in Fig. 6.5d.

6.2.5 Experimental Setup and Data Collection

Simulation of FEVAR

Five abdominal aneurysm phantoms, created from contrast-enhanced CT data of AAA patients, were printed on a Stratasys Objet (MN, USA) in VeroClear and TangoBlack. One example is shown in Fig. 6.8a. Three stent grafts: iliac ($6 - 10mm$ diameter, $90mm$ height, Cook Medical), fenestrated ($22 - 30mm$ diameter, $117mm$ height, Cook Medical) and thoracic ($30mm$ diameter, $179mm$ height, Medtronic, MN, USA) were used in the experiments. Each stent segment of the three stent grafts was sewn on five markers at non-planar positions. In a setup, a stent graft was compressed within a Captivia delivery system (Medtronic, $8mm$ diameter, shown in Fig. 6.8a), inserted into the 3D printed aneurysm, and deployed at the target position.

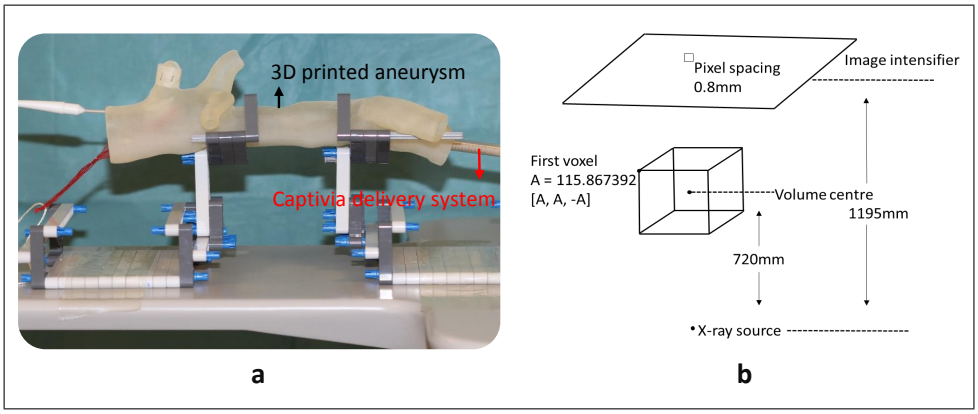


Figure 6.8: (a) experimental setup, (b) registration of the fluoroscopic image coordinate system to the CT coordinate system.

Data Collection

The three stent grafts with newly sewn markers were firstly scanned by a GE Innova 4100 (GE Healthcare, Bucks, UK) for the reference 3D marker positions before any experimental setup. For simulating FEVAR, the stent graft diameter needed to fit the artery diameter, resulting in 14 matching positions in total between the five phantoms and three stent grafts. Details of each setup are shown in Tab. 6.2. After deploying the stent graft in each setup, 13 2D fluoroscopic images from different view angles, varying from $-90°$ to $90°$ with $15°$ interval, were obtained by the same CT machine. This varying view angle is necessary for proving that the 3D shape instantiation works for any view angle. It caused the 2D marker shape appearances to be similar in the fluoroscopy images, even though these markers were designed to be differentiable in 3D.

There should be $14 \times 13 = 182$ images, however, 11 images were not stored by the operator. For the setups shown in Tab. 6.2, 7/14 setups expressed by $\odot$ were used for the training (80 images) of Equally-weighted Focal U-Net, 6/14 setups ($6 \times 13 = 78$ images) expressed by $\bigoplus$ were used for the testing; here the test image number corresponds to that in Sec. 6.3, 1/14 setup (13 images) expressed by $\bigotimes$ were abandoned due to one marker falling off. Due to the limited available images, no validation dataset was split. Sometimes, two experiments were set up with the same stent graft and the same phantom. These two setups were not the same due to the different positions inside the phantom. A CT scan was collected for each deployed

Table 6.2: Stent Graft - Phantom Matching ($\bigoplus$ - Test; $\bigodot$ - Train; $\bigotimes$ - Abandon.)

Phantom number	1	2	3	4	5
Iliac (S1)	$\bigoplus$	$\bigodot\ \bigoplus$	-	$\bigodot$	-
Test image number	$1-13$	$14-26$	-	-	-
Fenestrated (S2)	$\bigoplus$	$\bigoplus$	$\bigoplus\ \bigotimes$	$\bigodot\ \bigodot$	$\bigodot$
Test image number	$27-39$	$40-52$	$53-65$	-	-
Thoracic (S3)	-	$\bigodot$	$\bigodot$	$\bigoplus$	-
Test image number	-	-	-	$66-78$	-

stent graft. Usually, this CT scan was utilized as the ground truth of the markers and deployed stent graft, except for one comparison validation in Sec. 6.3.5 where the scanned 3D marker positions were used as the reference 3D marker positions too. The coordinates of marker projections on 2D fluoroscopic images were transformed into the CT coordinate system, as shown in Fig. 6.8b. 3D Slicer [172] was used to segment the stent 3D shape and marker 3D shape from the CT scan. The average unsigned distance between the reconstructed 3D shape and the ground truth was calculated in CloudCompare software [173].

Data augmentation

To evaluate the character of the proposed network to data augmentation, two different data augmentation methods were compared: 1) rotated the 80 training images from $-36°$ to $35°$ with $1°$ as the interval; 2) rotated the 80 training images from $-180°$ to $165°$ with $15°$ as the interval and flipped each rotated image along the horizontal and vertical direction respectively. Both data augmentation methods augmented the training images with 72 times, resulting 5760 training images.

Image enhancement

To evaluate the performance of the proposed network to image enhancement, image intensity adjustment and contrast-limited adaptive histogram equalization were applied with $MATLAB$ function:

$$\mathbf{I}'_k = adapthisteq(imadjust(\mathbf{I}_{norm_k})) \tag{6.9}$$

Ground truth labelling

The markers were labelled in Analyze (AnalyzeDirect Inc, Overland Park, KS, USA) with firstly magnifying the image from 512×512 to 4096×4096 for a clear labeling, and then shrinking the image from 4096×4096 back to 512×512.

Other parameters: the learning rate was set step-wisely and divided by two or five manually when the loss stopped decreasing. The dropout rate was set as 0.75. The weights in the neural network were initialized by truncated normal distribution with $mean = 0.0$ and $std = 0.1$ while the biases were initialized by constant 0.1. The optimizer was the momentum optimizer in Tensorflow with the momentum set as 0.95. The batch size was set as 1. The loss function was written by tf.nn.softmax, tf.log, and tf.reduce_mean. The mean Intersection over Union (mIoU), the overlap of the ground truth and the prediction over the union of the ground truth and the prediction, was calculated to evaluate the segmentation performance. Except Sec. 6.3.1, all training procedures were based on the data augmented with 30° image rotation and without image enhancement.

6.3 Results

Semi-automatic marker detection with automatic marker segmentation and manual correction for failure cases and 3D stent graft shape instantiation were validated with errors shown in this section. The characters of the proposed Equally-weighted Focal U-Net with respect to the number of U-Net block, image enhancement, data augmentation, and weight are illustrated in Sec. 6.3.1. The comparison between different methods is presented in Sec. 6.3.2. Detailed multiple-class marker segmentation results are shown in Sec. 6.3.3.

The 2D distance error of semi-automatic marker detection, the 3D distance and the angular error of marker instantiation, the 3D distance error of stent graft instantiation, reconstructed 3D shape details are given in Sec. 6.3.4, by using both manually and semi-automatically detected markers. The 3D distance error is the unsigned Euclidean distance between the reconstructed 3D markers or stent grafts and the ground truth with the position displacement (explained in Sec. 6.2.3) corrected by aligning the centers. The angular error

is the unsigned angle (θ in Fig. 6.5a) difference between the reconstructed marker and the ground truth. Angular errors were measured, as the facing and orientations of fenestrations or scallops are important for path planning (red, green, blue path in Fig. 6.1d) in robot-assisted FEVAR. A comparison with using pre-experimental and intra-experimental 3D marker positions as the reference 3D marker positions is provided in Sec. 6.3.5, showing potential accuracy improvements for *in-vivo* applications.

6.3.1 Network Characters

The mIoUs achieved with different setups are shown in Tab. 6.3, where the highest mIoU is emphasized in bold font.

Number of U-Net block

Equally-weighted Focal U-Net with block number from $1-6$ were trained to segment the multiple-class markers in Fig. 6.4, mIoUs are listed in Row $1-6$ in Tab. 6.3. It can be concluded that 1-block U-Net and 6-block U-Net under-performed slightly others. However, the training time increased from 36 hours for 1-block U-Net to 120 hours for 6-block U-Net. Based on this comparison result, 2-block U-Net was chosen as a trade-off between the efficiency and the performance in the following validations to test the network property.

Data augmentation

Equally-weighted Focal U-Net with 2 blocks was trained on the data augmented with 30° image rotation and with 180° image rotation respectively. The mIoUs for six classes on the 78 testing images are summarized in the Row 2 and the Row 7 in Tab. 6.3. The results showed that the mIoUs achieved with 30° image rotation are higher than that with 180° image rotation in most classes, except for Marker 3 and Marker 4. Hence, 30° image rotation was utilized as data augmentation in this chapter.

Image enhancement

Equally-weighted Focal U-Net with 2 blocks was trained on the training data with and without image enhancement respectively. The mIoUs of

the six classes achieved on the 78 testing images are summarized in the Row 2 and the Row 8 in Tab. 6.3. The results presented that the mIoUs decreased significantly when the training data was pre-processed with image enhancement. Therefore, the images in the training set will only be processed by normalization in the following training.

Table 6.3: U-Net with different setups (mIoU-mean Intersection over Union, B-Background, M-Marker, Num.-Number, Aug. - Augmentation)

Row	30° Aug.	180° Aug.	Image Enhancement	Block Num.	Weight	Focal Loss	B mIoU	M1 mIoU	M2 mIoU	M3 mIoU	M4 mIoU	M5 mIoU
1	√			1	1	√	**0.9996**	0.6392	0.5159	0.5929	**0.5692**	0.5998
2	√			2	1	√	**0.9996**	0.7030	0.5687	0.6778	0.5094	0.5765
3	√			3	1	√	**0.9996**	**0.7325**	**0.5828**	0.6952	0.5453	**0.6105**
4	√			4	1	√	**0.9996**	0.7280	0.5462	0.6831	0.5266	0.5883
5	√			5	1	√	**0.9996**	0.7254	0.5395	0.6843	0.5156	0.5841
6	√			6	1	√	**0.9996**	0.6179	0.5475	0.5596	0.4424	0.4986
7		√		2	1	√	**0.9996**	0.4793	0.5081	**0.6988**	0.5523	0.5001
8	√		√	2	1	√	0.9992	0.4092	0.0843	0.2779	0.0993	0.3133
9	√			2	1		0.9993	0.1900	0.0000	0.0000	0.0000	0.0000
10	√			2	20		0.9986	0.1508	0.0428	0.1151	0.1311	0.1387
11	√			2	50		0.9981	0.4168	0.2260	0.3639	0.3037	0.3630
12	√			2	100		0.9979	0.4222	0.2195	0.3439	0.2978	0.3415
13	√			2	500		0.9967	0.3020	0.1207	0.2782	0.2531	0.2868

Weight

2-block U-Net with the weight of 20, 50, 100, 500 were trained respectively. The mIoUs of the six classes on the 78 testing images are listed in the Row 9-13 in Tab. 6.3. The results illustrated that 2-block U-net with the weight of 50 presented optimal performance comparing with small weights (weight = 20) and the large weight (weight = 500). Thus, 2-block U-Net with the weight of 50 was applied in the following work.

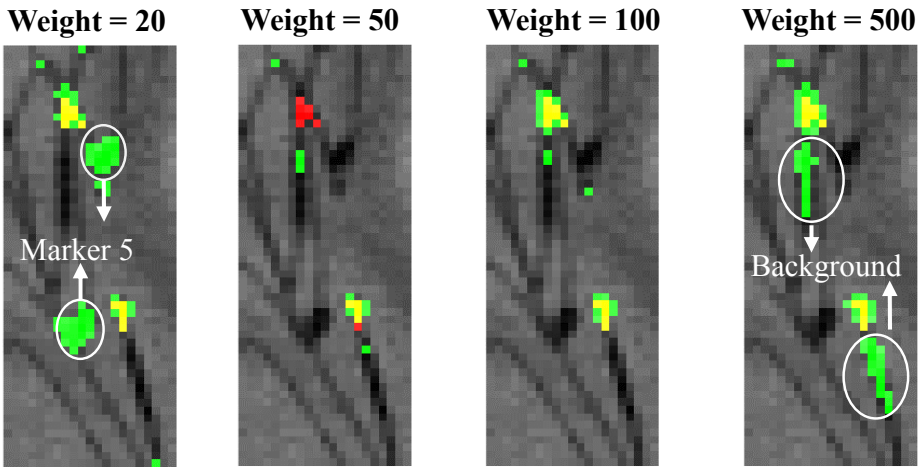


Figure 6.9: Cropped segmentation results for Marker 2 with the weight as 20, 50, 100, and 500, where red region - the ground truth, green region - the prediction, yellow region - the correctly-segmented pixels.

The segmentation results of the 2-block U-Net with different weights are illustrated in Fig. 6.9. It can be seen that the five foreground or marker classes could not be clearly distinguished between each other with a small weight, i.e. 20. However, if the weight of the network is too large, i.e. 500, the background was mis-classified as a foreground, as this wrong classification contributed too less to the total loss. For example, a wrongly-segmented background ($\mathbf{P}_{(i,j,n)} = 0.1$) contributed $(1 - \mathbf{P}_{(i,j,n)}) \times 1 = 0.9$ to the total loss while a wrongly-segmented foreground ($\mathbf{P}_{(i,j,n)} = 0.1$) contributed $(1 - \mathbf{P}_{(i,j,n)}) \times 500 = 450$ to the total loss. The mIoUs of the background decreased along the increased weight (Row 9-13 in Tab. 6.3), which also proves this trend.

6.3.2 Comparison between Different Methods

The performance of 2-block U-Net using five different methods were explored in Fig. 6.10: 1) Equally-weighted Focal U-Net (the proposed method); 2) Weighted U-Net with the weight as 50 for foreground and the weight as 1 for background; 3) U-Net with Equally-weighted Focal Loss which used an equally-weighted focal loss from the beginning of the training; 4) Equally-weighted U-Net with the weight set as 1 for both the foreground and the background; 5) Weighted Focal U-Net with the weight set as 50 for the first step training, and then focal loss with the weight of 50 for the second step training. The performance of these methods are shown by the mean and std IoUs. The Fig. 6.10 illustrated that the proposed method has slightly better performance on every marker class comparing with other methods.

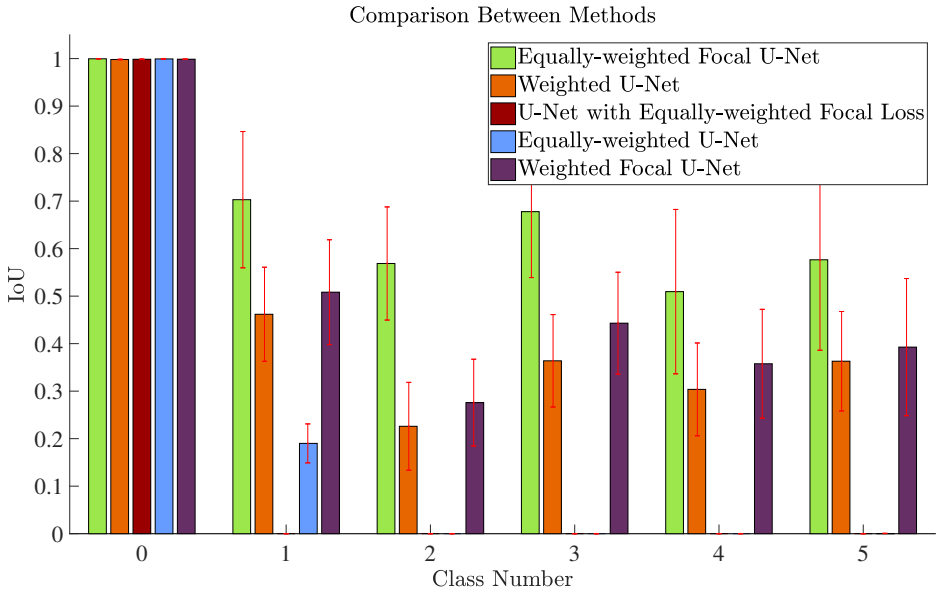


Figure 6.10: The *mean* ± *std* IoUs for the six classes segmented by five different methods

6.3.3 Multiple-class Marker Segmentation

In this section, I focus more on the accuracy and release the requirement of time-efficiency. Equally-weighted Focal U-Net with 3-block (Row 3 in Tab. 6.3) was applied to segment each testing image. The results are illustrated in Fig. 6.11. The Fig. 6.11 showed that the proposed network could segment

most of the images with outstanding performance, except from a few markers in the image No.10, No.13, No.59 and No.71. Besides, the Fig. 6.12 presents the segmentation details of image No.21 using the proposed method, where each marker class was segmented with a high overlap between the ground truth and the prediction.

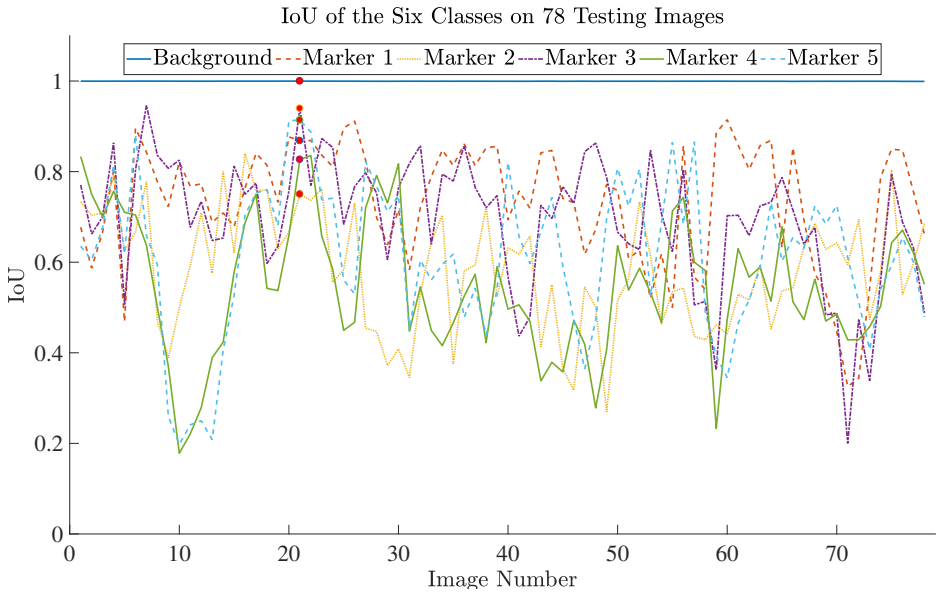


Figure 6.11: The IoU of the six classes on 78 testing images segmented with a 4-block Equally-weighted Focal U-Net.

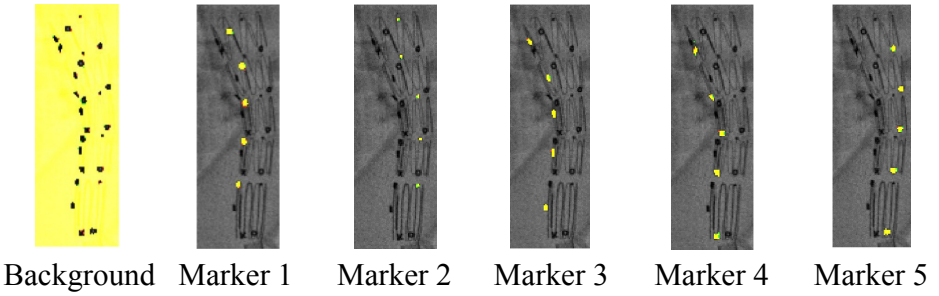


Figure 6.12: Cropped segmentation results for six classes on image NO.21: red - the ground truth, green - the prediction, yellow - the correctly-segmented pixels.

6.3.4 3D Shape Instantiation

The 78 images contain 2470 markers, 81.01% of them were segmented with a center position error $< 1.6mm$ which are 2 pixels on the fluoroscopic projection and half of the marker size. The marker center positions determined with $> 1.6mm$ error were corrected manually. With these marker center positions, the angular error and 3D distance error of 3D shape instantiation were illustrated in Fig. 6.13, showing that the proposed method presents comparable performance with 3D shape instantiation with both manual and semi-automatic marker center determination. More 3D shape instantiation results could be found in [6].

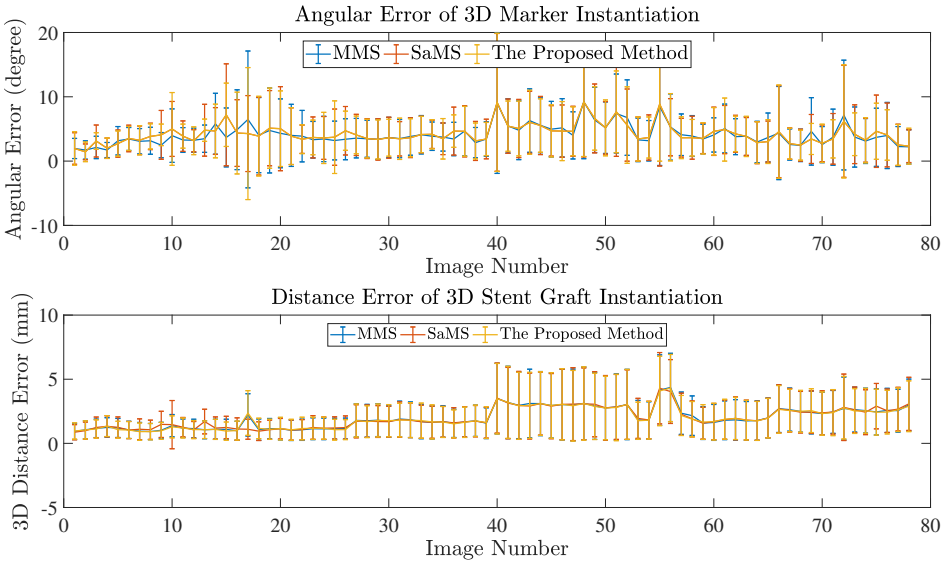


Figure 6.13: The angular error of 3D marker instantiation (top) and the 3D distance error of 3D stent graft instantiation (bottom) using three different marker center determination methods: MMS - Manual Marker Segmentation; SaMS - Semi-automatic Marker Segmentation (proposed in [6] where markers were segmented by the U-Net while were classified manually.); the proposed method (proposed in this chapter where markers were segmented and classified by the Equally-weighted Focal U-Net. For both the SaMS and the proposed method, manual correction was added when larger errors happen.

To purely validate the proposed 3D shape instantiation method, I further manually correct the distance error of semi-automatic marker center deter-

mination, as shown in Fig. 6.14 top. An average distance error of 0.42 mm (half a pixel) was achieved. Both the marker centers detected manually and semi-automatically were used to recover the 3D markers. The 3D distance errors of marker instantiation are shown in Fig. 6.14 bottom. An average distance error of $0.92mm$ for S1, $4.08mm$ for S2, and $6.52mm$ for S3 were achieved with semi-automatic marker detection, which were close to that achieved by manual marker detection ($0.86mm$ for S1, $4.08mm$ for S2, and $6.44mm$ for S3). This average distance error is comparable, as the marker size is almost $3mm$. The errors of S2 (image 27-65) and S3 (image 66-78) were higher than S1 (image 1-26) due to two reasons: 1) the diameters of S2 and S3 were larger than S1; 2) the deployment device was small for S2 and S3, causing more non-rigid stent segment deformations and hence more non-rigid deformations between the reference and target 3D marker positions. The errors of the latter two setups (image 40-65) are higher than that of (image 27-39), as the more times S2 was compressed and deployed, the more non-rigid stent segment deformations were introduced.

The angular errors of recovered markers and distance errors of reconstructed stent grafts are shown in Fig. 6.15. An average angular error of 4.24° was achieved with semi-automatic marker detection which is similar to that (4.12°) achieved with manual marker detection. An average distance error of $1.99mm$ was achieved with semi-automatic marker detection which is close to that ($1.97mm$) achieved with manual marker detection. The average angular and distance errors for the six setups are shown in Tab. 6.4.

Table 6.4: average errors(S1-iliac; S2-fenestrated; S3-thoracic; M-Manual; S-Semi-automatic; Angle-degree; Distance-mm)

Stent Graft	S1	S1	S2	S2	S2	S3
Image Number	1-13	14-26	27-39	40-52	53-65	66-78
Angle (S)	3.29	4.43	3.79	6.11	4.27	3.58
Angle (M)	2.83	4.25	3.66	6.18	4.24	3.57
Distance (S)	1.22	1.14	1.70	3.03	2.22	2.61
Distance (M)	1.10	1.18	1.72	3.04	2.23	2.57

Examples of 3D shape instantiation coloured by the distance error are shown in Fig. 6.16a – the light grey mesh is the proposed shape instantiation result while the coloured stents are the ground truth. It can be seen that

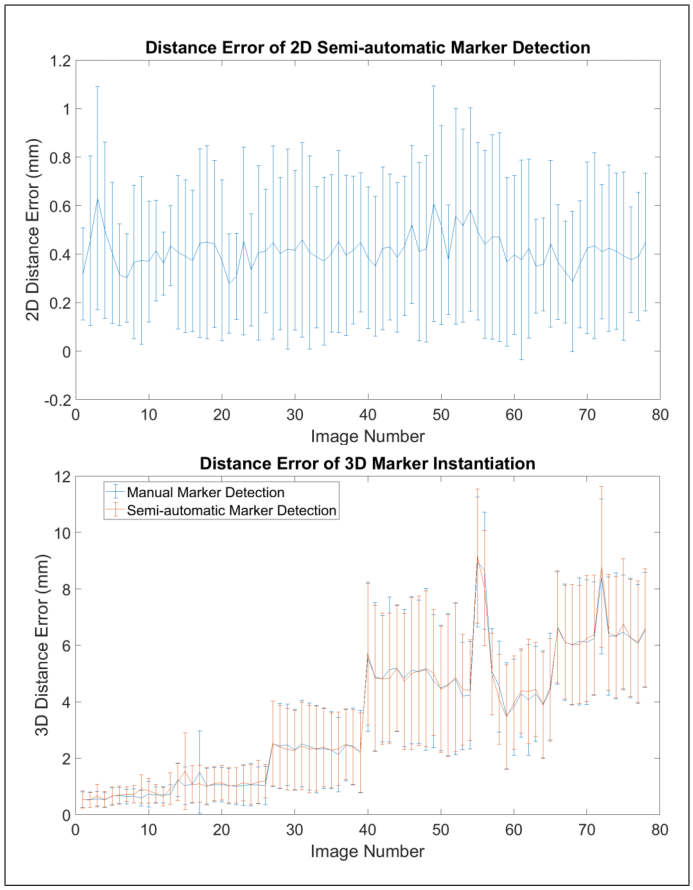


Figure 6.14: The (mean ± stdev) distance errors of semi-automatic marker detection (top) and 3D marker instantiation (bottom), the std errors were calculated across multiple markers on a stent graft.

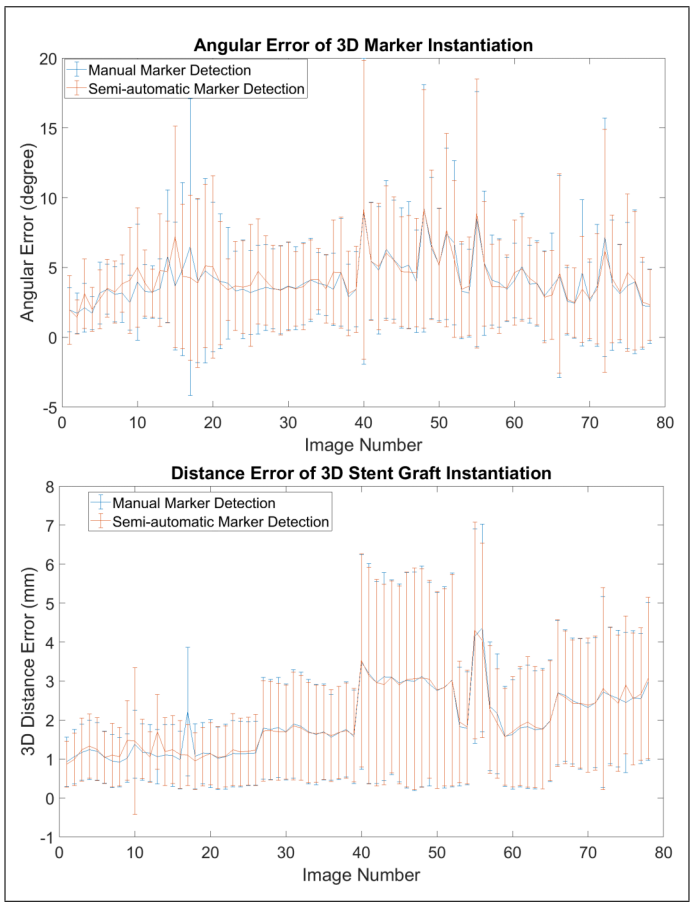


Figure 6.15: 3D Shape instantiation errors (mean ± stdev) of angular (top) and distance (bottom) for three stent grafts. The std of angular error was calculated across multiple markers on a stent graft while that of distance error was calculated across multiple vertices of a stent graft.

the bending, compressing, twisting, etc. of the stent graft, the scallops or fenestrations are reconstructed well. Examples of the reconstructed scallop and fenestration (Fig. 6.16b top) are compared with the real ones (Fig. 6.16b bottom). The dark grey stents in Fig. 6.16b top are the ground truth from CT with commercial gold markers indicating the scallop and fenestration.

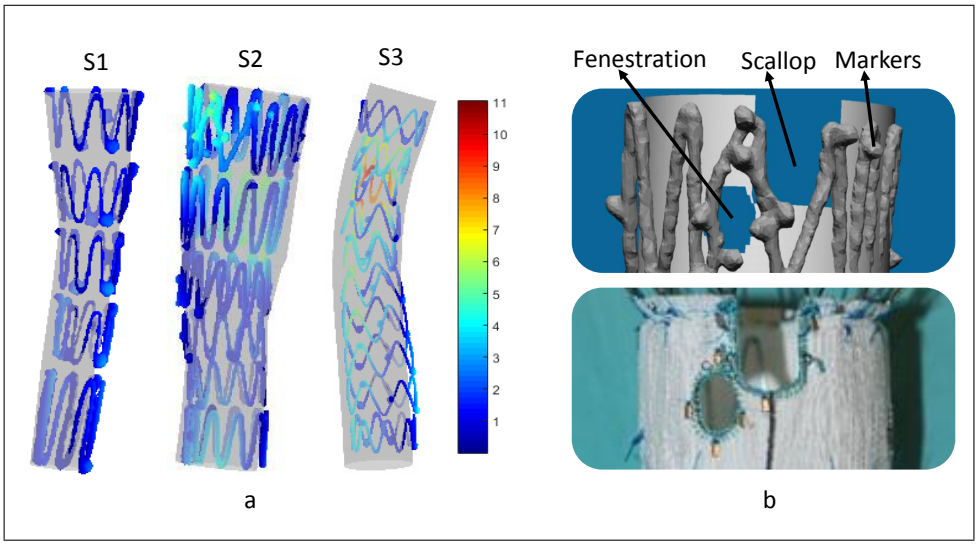


Figure 6.16: Examples of (a) 3D shape instantiation of the three stent grafts colored by the distance error (colorbar of errors in mm), (b) reconstructed scallop and fenestration (top) compared to the real ones (bottom).

6.3.5 Influence of Non-rigid Marker Set Deformation

3D marker instantiation errors with pre-experimental and intra-experimental 3D marker positions as the reference 3D marker positions are shown in Fig. 6.17. The errors with intra-experimental 3D references are much lower, proving that less non-rigid deformation between the reference and the target 3D marker positions could improve the instantiation accuracy. The higher errors in a few images (48, 55, 56, 72) are due to the mis-classification of the markers.

The computational time is less than $8ms$ in MATLAB for one stent segment instantiation on an Intel(R) Core(TM) i7-4790 CPU@3.60GHz computer. The marker segmentation takes less than $0.1s$ in Tensorflow on a NVIDIA TITAN Xp GPU. For training Focal U-Net, the first step takes about 30

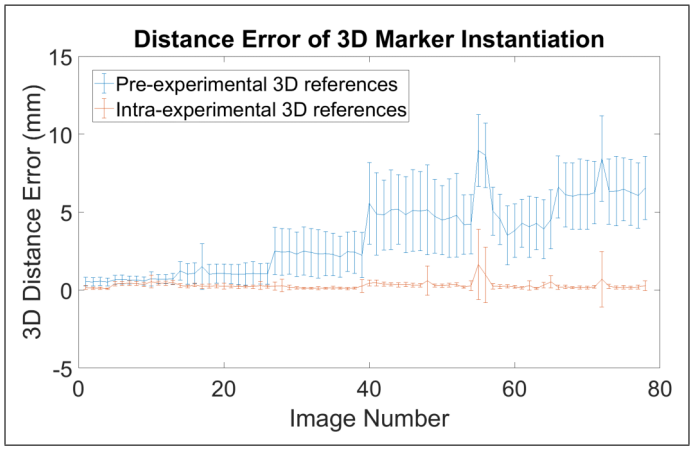


Figure 6.17: Distance errors of 3D marker instantiation with pre-experimental and intra-experimental 3D marker positions as the reference 3D marker positions.

minutes while the second step takes approximate 2 hours.

6.4 Discussion and Conclusion

In this chapter, the non-rigid deformation of the whole stent graft was split into piecewise stent segment rigid transformations and then was reconstructed by interpolating these reconstructed stent segments. The average distance error of reconstructed stent grafts at around $1 - 3mm$ and the average angular error of recovered markers at around 4° illustrates that this splitting is reasonable and could be used for future work on stent grafts. The average distance error of reconstructed stent grafts - $3mm$ is comparable, as the size of the markers is approximately $3mm$. Even with the limited experimental environment (the drifted markers, unsuitably-small delivery device in terms of the size, and repeated use of the stent graft), comparable average distance and angular errors were achieved. It is expected that the accuracy could be improved with more stable marker sewing, a suitable delivery device and a one-off use of the stent graft (the stent graft is only compressed and deployed once in *in-vivo* scenarios), and hence the accuracy is expected to be higher than the experiments in this chapter.

The only input for the proposed 3D shape instantiation is a single fluoroscopic projection of markers, which decreases the X-ray radiation to a

minimum, as markers are always visible, albeit not always clearly, even under lowest X-ray radiation. Marker imaging is also robust to respiratory and cardiac induced motions. The stents, 3D printed aneurysms, and the holders all show up in the 2D fluoroscopic projections in our experiments. In practice, the 2D fluoroscopic images, i.e. Fig. 6.1c, are much 'cleaner' than our experiments due to the block of tissue. The commercial markers made of gold also have higher visibility than the 3D printed markers made of steel used in this chapter. It is expected to be easier to segment and classify the markers in practical applications.

Equally-weighted Focal U-Net was proposed to segment the customized stent graft markers into multiple-classes. The segmented marker center positions would be used by the RPnP method and hence automatic 3D stent graft shape instantiation was possibly achieved. In Sec. 6.3.1,the performance of U-Net with different block number was explored. The results showed that Equally-weighted Focal U-Net did not achieve higher mIoU along with an increasing block number. Possible reasons could be network degradation and insufficient training data. In the future, the network structure will be explored in details. In Sec. 6.3.1, different weights were explored. Usually, weighted loss outperforms equally-weighted loss for class-imbalance segmentation, as it treats the foreground more importantly by assigning a higher weight for it. However, in this chapter, we consider the background as equally important as the foreground, as a mis-classified background will also decrease the foreground IoU. So equally-weighted loss was applied.

Due to the limited experimental environment, i.e., the expensive price of stent graft and for printing AAA phantoms, each stent graft and phantom are used multiple times in order to collect more data. This causes that the same stent graft and phantom sometime appear in both the training and testing. Even though different positions inside the phantom were used, this may still weak the convincing of the validation. The proposed method is capable for multiple-class marker segmentation, obtained an overall mIoU of 0.6943, and detected 81.01% markers with center position error $< 1.6mm$. Comparable 3D shape instantiation error was achieved ($1.9605mm$) with the approximately-automatic marker center determination method in this chapter, with respect to 3D shape instantiation with semi-automatic marker center determination ($1.9746mm$) and with manual marker center determination ($1.9874mm$) in [6].

The experiments demonstrated the potential robustness of the proposed framework to fluoroscopic view angles - fluoroscopic images from 13 view angles were tested and shown with neglectable difference in accuracy. However, a clear view without marker overlapping and hence easier marker classification is still preferred to avoid the mis-classification of the markers (examples shown in Sec. 6.3.5). The computation time of $0.1s$ per image indicates that the proposed framework can work in real-time potentially, as typical fluoroscopy acquisitions used in clinic are approximately 2-5 frames per second.

In conclusion, a 3D shape instantiation framework for fenestrated stent grafts including marker placement, stent segment pose instantiation, stent graft shape instantiation and semi-automatic marker detection was proposed in this chapter. The proposed framework only needs a single fluoroscopic projection and is only based on markers, which decreases the X-ray radiation to a minimum. Compared with the state-of-art 2D fluoroscopy navigation used in robot-assisted FEVAR procedures, the proposed framework reconstructs not only the 3D shapes of the stents but also the grafts, fenestrations and scallops. Equally-weighted Focal U-Net was proposed for multiple-class marker segmentation and then automatic 3D stent graft shape instantiation could be achieved. The performance of the proposed network was explored and discussed with different characters, such as the number of blocks, method of data augmentation, image enhancement, and different weights. Based on these results, 3-block Equally-weighted Focal U-Net showed optimal accuracy in multiple-class marker segmentation. In this chapter, the proposed Equally-weighted Focal U-Net is only validated on fully-deployed stent graft. Markers on fully-compressed and partially-deployed stent segments are more difficult to segment, due to the cluster of markers and the trained model for fully-deployed state could not be transferred to fully-compressed and partially-deployed state. In the future, the proposed network will be further improved and extended to a general framework for wider applications. This work is a first step towards a complete 3D shape instantiation which predicts the 3D shape of a fenestrated stent graft after the deployment from a single 2D fluoroscopic image of its compressed state to improve robotic navigation for FEVAR.

7 Conclusions and Future Perspectives

2D medical image segmentation is the basis of both fundamental and advanced medical image analysis tasks. Currently, most image segmentation problems are solved by deep learning with training DCNNs. DCNN based methods have good portability, however, 1) the performance of DCNN is not good enough yet for practice, and post-processing and supplementary algorithms are usually needed; 2) the generalizability of trained networks when transferring to unseen patients is an unsolved problem; 3) DCNN is data hungry, and in medical problems it is difficult to collect enough training data to see all possible situations; 4) the explainability of deep learning is also a problem that needs to be further explored. In this thesis, to achieve a complete and automatic 3D shape instantiation scheme, I worked on the normalization and architecture design of training DCNNs for medical image segmentation.

For normalization while training a U-Net for medical image segmentation, I reviewed the most popular four normalization methods including BN, IN, LN and GN on the RV, LV and aortic segmentation with three-fold cross validation. By comparing the performance, convergence speed and converged lowest loss, it was found that calculating the mean and variance of a detailed feature map division usually performed better. This conclusion can help with the development of new normalization methods. I also proposed the ILN which linearly weighted the feature map after IN and LN to train a U-Net for medical image segmentation under two conditions: 1) using sigmoid function to weight the feature map after IN and LN; 2) cascading a GN after the weighted feature map. The validation shown that the proposed ILN could outperform other traditional normalization methods on small medical dataset.

I further worked on 3D normalization for medical volume segmentation. In 3D DCNNs, with an input volume $\boldsymbol{V}$ of dimensions D $\times$ H $\times$ W, where

D, H, W represent the depth, height and width of the volume respectively, intermediate feature maps $\boldsymbol{F}$ have a dimension of $\mathrm{N} \times \mathrm{D} \times \mathrm{H} \times \mathrm{W} \times \mathrm{C}$, where N is the batch size and C is the feature channel. C weights were added to the feature map with one weight for each channel. To reduce the computational complexity, the weights were multiplied to the mean of each feature channel rather than being multiplied to each feature channel. The variance was calculated for each feature channel independently. Hence adjustable weighted normalization was achieved, which potentially could achieve IN, LN, and GN with training the added weights.

For architecture design in training DCNNs for medical image segmentation, ACNN was proposed using only convolutional layers and setting the atrous rate as k^n at the n^{th} layer in the atrous block. Multiple atrous blocks were cascaded to form the proposed ACNN. No max-pooling layers or deconvolutional layers were used. All intermediate feature maps were with the same resolution. It can achieve comparable segmentation results to a traditional U-Net with fewer trainable parameters.

I further worked on 3D neural architecture design for medical volume segmentation. Currently, due to the restriction of GPU memory and the limited training data, medical volumes are usually cropped into patches with the size of $64 \times 64 \times 64$ or $64 \times 128 \times 128$. However, this patch cropping will cause class-imbalance problem, as some patches may contain only foreground or background. I and my colleagues proposed a new patch cropping method which cropped the medical volumes into a size of $512 \times 512 \times 8$. Hence, the class-imbalance problem was eliminated in the XY plane. The 3D convolutional layers in the traditional U-Net and V-Net were decomposed into XY and Z convolutions separately to compensate for the asymmetrical property of the new patches.

The main work of my PhD is 3D shape instantiation, which is potentially useful and important to the development and popularization of robot-assisted MIS. However, difficulties and bottlenecks exist: 1) time is limited during a surgery, so the input can only be a limited amount of image and the 3D shape instantiation algorithm has to be fast for real-time navigation; 2) reconstructing a 3D shape from a 2D image is a problem that crosses different modalities; 3) accuracy is important for a safe operation; 4) robustness to unseen situations, i.e., tissue obstruction, decreased resolution and so on, is vital; 5) high-resolution instantiated 3D mesh is essential. The methodology

for 3D shape instantiation is currently based on the combination of multiple algorithms usually. It is also task specific and one algorithm usually can not be transferred to other tasks.

For 3D shape instantiation of soft organs, a general, real-time and registration-free framework is proposed. It is applicable to multiple targets, once the required training data at multiple time-frames is available. SPCA was used to determine the optimal scan plane by analyzing the 3D SSM of the target while KPLSR was used to learn the relationship between the 2D SSM and the 3D SSM. Multiple deficiencies also exist in the proposed framework. First, the optimal scan plane needs to be determined patient-specifically. Second, the KPLSR model needs to be trained for each patient. Third, the 3D SSM needs to be collected and generated each time to determine the optimal scan plane and to train the model. Fourth, segmenting the RV contours from 2D MRI slice and generating the 2D SSM through non-rigid registration are essential in both the training and testing. These four points bring heavy workloads to clinicians, which limits the practicality of the proposed framework. A fifth deficiency is that even though the KPLSR-based instantiation is robust to the number of components used, it is sensitive to the Gaussian width parameter. This point decreases its robustness.

To compensate for the fifth point, I and my colleagues replaced the KPLSR learning with a deep learning method in [23], where convolutional layers were used to extract information from the 2D MRI slice while fully-connected layers were used to regress the 3D vertex coordinates of 3D shape. Rather than using the popular L1 and L2 loss, Chamfer loss is used. It releases the correspondence between the ground truth vertex and the prediction vertex during the loss calculation, hence allowing larger exploration space for the network. This newly proposed instantiation based on deep learning can regress 3D point cloud directly from a single 2D image with fully automatic training, which eliminates the third and fourth deficiency mentioned above. However, it loses the vertex correspondence and hence 3D mesh is not achievable. I and my colleagues are working on combining Graph Convolutional Network (GCN) and deep learning to instantiate directly a 3D mesh from a single 2D image.

For 3D shape instantiation of stent grafts, the 3D shape of a fully-deployed stent graft was instantiated with mathematical modelling, the RPnP method, and graft gap interpolation. There are three statuses for a stent graft

during a FEVAR or EVAR surgery: fully-compressed, partially-deployed and fully-deployed. In addition, I worked on predicting the deployed stent graft shape from a single 2D fluoroscopic projection of its compressed state in [11]. The 3D mesh of a compressed stent graft was instantiated with the same mathematical modelling and RPnP method in this thesis, but the radius r was expended to the deployed value at each height to predict the deployed stent graft shape. During the experiment, as the stent graft was at a compressed state, it was impossible to collect the instantaneous 3D deployed shape as the ground truth, hence we deployed the stent graft manually and then scanned it with CT as the ground truth. It was obvious that the state of the compressed stent graft was changed during the manual deployment process, indicating that the ground truth was not accurate. Hence, the instantiated angular error of 10° to 20° in [11] was larger than that in this thesis.

I and my colleagues also worked on instantiating the 3D shape of partially-deployed stent segments in [13], where the main bottleneck was the unknown reference 3D marker positions. For both the fully-compressed or fully-deployed state of stent segments, the reference 3D marker positions were acquired from stent graft design or CT scans. However, the reference 3D marker positions of partially-deployed stent segments were unknown. In [13], a GCN was trained to learn the deformation from fully-deployed reference 3D marker positions to partially-deployed ones and the training data was collected by partially deploying multiple stent segments multiple times. Deep learning was also tried, however the performance was worse than GCN, as shown in the validation in [13]. The trained model was used to predict the partially-deployed reference 3D marker positions intra-operatively, and then these predicted reference 3D marker positions were combined with the mathematical modelling, RPnP method and graft gap interpolation in this thesis for 3D shape instantiation for partially-deployed stent grafts.

We also worked on instantiating the 3D skeleton of AAA from a single intra-operative 2D fluoroscopic image [14]. Graph matching was used to match the pre-operative 3D AAA skeleton and the intra-operative 2D AAA skeleton projection, and these two skeletons were registered to instantiate the intra-operative 3D AAA skeleton. The instantiated 3D AAA skeleton is the central-line of the intra-operative AAA and is a safe path for FEVAR robot to follow to insert, deploy and rotate the fenestrated stent graft with the

delivery device. In this thesis, the 3D shape instantiation for fully-compressed, partially-deployed and fully-deployed stent grafts are validated separately. For a complete FEVAR navigation system in practical applications, these three components are essential to be integrated. The 3D shape of AAA, not only the 3D skeleton, is also very important for a complete navigation system for FEVAR.

Due to the fact that the segmented results from DCNN usually are not good enough and manual corrections are needed, in this thesis, the 2D segmentation and 3D shape instantiation are evaluated separately. An integrated evaluation is essential for showing the overall navigation property in the future. Only the deformation caused by respiration and cardiac beating is considered in this thesis, the deformation caused during the intervention, i.e. insertion of instruments, is not considered. For the experiments, usually phantom or off-line patient data are used, animal and patient test are essential in the future if clinicians wish to use the proposed technique in practical surgeries.

To summarize, 3D shape instantiation can be useful in intra-operative and dynamic 3D navigation. First, it only needs a single 2D projection slice as the input which potentially can achieve real-time update of the intra-operative 3D shape. Second, the instantiated 3D shape is not only an interpolation or rigid transformation of pre-operative 3D shapes, but also considers intra-operative deformation through taking the intra-operative 2D projection or slice into account. In this thesis, I proposed 3D shape instantiation frameworks for both soft organs and the stent graft. In addition, 2D medical image segmentation is used to supply the input of 3D shape instantiation and is also very important for a complete and automatic 3D shape instantiation pipeline. Hence, I also worked on the normalization and architecture design for training DCNNs for medical image segmentation.

Licensed Content Re-Use Permissions

Elsevier

Authors can include their articles in full or in part in a thesis or dissertation for non-commercial purposes.

IEEE

The IEEE does not require individuals working on a thesis to obtain a formal reuse license.

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Imperial College London's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to this link to learn how to obtain a License from RightsLink.

Springer

The permission of replicating the content published in [22] in Chapter 3 has been acquired from Springer on 8th Jan with a license number of 4744400989097.

Bibliography

[1] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of computational and graphical statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015, pp. 234–241.

[4] X.-Y. Zhou and G.-Z. Yang, "Normalization in training U-Net for 2D biomedical semantic segmentation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1792–1799, 2019.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[6] X.-Y. Zhou, G.-Z. Yang, and S.-L. Lee, "A real-time and registration-free framework for dynamic shape instantiation," *Medical image analysis*, vol. 44, pp. 86–97, 2018.

[7] R. H. Taylor, A. Menciassi, G. Fichtinger, P. Fiorini, and P. Dario, "Medical robotics and computer-integrated surgery," in *Springer handbook of robotics*. Springer, 2016, pp. 1657–1684.

[8] M. Shen, S. Giannarou, and G.-Z. Yang, "Robust camera localisation with depth reconstruction for bronchoscopic navigation," *International journal of computer assisted radiology and surgery*, vol. 10, no. 6, pp. 801–813, 2015.

[9] M. Shen, Y. Gu, N. Liu, and G.-Z. Yang, "Context-aware depth and pose estimation for bronchoscopic navigation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 732–739, 2019.

[10] X.-Y. Zhou, S. Ernst, and S.-L. Lee, "Path planning for robot-enhanced cardiac radiofrequency catheter ablation," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 4172–4177.

[11] X. Zhou, G. Yang, C. Riga, and S. Lee, "Stent graft shape instantiation for fenestrated endovascular aortic repair," in *The Hamlyn Symposium on Medical Robotics*. The Hamlyn Symposium on Medical Robotics, 2017.

[12] X.-Y. Zhou, J. Lin, C. Riga, G.-Z. Yang, and S.-L. Lee, "Real-time 3-d shape instantiation from single fluoroscopy projection for fenestrated stent graft deployment," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1314–1321, 2018.

[13] J.-Q. Zheng, X.-Y. Zhou, C. Riga, and G.-Z. Yang, "Real-time 3-D shape instantiation for partially deployed stent segments from a single 2-d fluoroscopic image in fenestrated endovascular aortic repair," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3703–3710, 2019.

[14] ——, "Towards 3d path planning from a single 2d fluoroscopic image for robot assisted fenestrated endovascular aortic repair," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8747–8753.

[15] Y. Feng, Z. Guo, Z. Dong, X.-Y. Zhou, K.-W. Kwok, S. Ernst, and S.-L. Lee, "An efficient cardiac mapping strategy for radiofrequency catheter ablation with active learning," *International journal of computer assisted radiology and surgery*, vol. 12, no. 7, pp. 1199–1207, 2017.

[16] V. Vitiello, S.-L. Lee, T. P. Cundy, and G.-Z. Yang, "Emerging robotic platforms for minimally invasive surgery," *IEEE Reviews in Biomedical Engineering*, vol. 6, pp. 111–126, 2012.

[17] S. Li, C. Xu, and M. Xie, "A robust O(n) solution to the perspective-n-point problem," *PAMI*, vol. 34, no. 7, pp. 1444–1450, 2012.

[18] X.-Y. Zhou, Y. Guo, M. Shen, and G.-Z. Yang, "Artificial intelligence in surgery," *arXiv preprint arXiv:2001.00627*, 2019.

[19] X.-Y. Zhou, G.-Z. Yang, and S.-L. Lee, "A real-time and registration-free framework for dynamic shape instantiation," *Medical Image Analysis*, vol. 44, pp. 86–97, 2018.

[20] X.-Y. Zhou, J. Lin, C. Riga, G.-Z. Yang, and S.-L. Lee, "Real-time 3-D shape instantiation from single fluoroscopy projection for fenestrated stent graft deployment," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1314–1321, 2018.

[21] X.-Y. Zhou, C. Riga, S.-L. Lee, and G.-Z. Yang, "Towards automatic 3D shape instantiation for deployed stent grafts: 2D multiple-class and class-imbalance marker segmentation with equally-weighted focal U-Net," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1261–1267.

[22] X.-Y. Zhou, P. Li, Z.-Y. Wang, and G.-Z. Yang, "U-net training with instance-layer normalization," in *International Workshop on Multiscale Multimodal Medical Imaging*. Springer, 2019, pp. 101–108.

[23] X.-Y. Zhou, Z.-Y. Wang, P. Li, J.-Q. Zheng, and G.-Z. Yang, "One-stage shape instantiation from a single 2D image to 3D point cloud," *arXiv preprint arXiv:1907.10763*, 2019.

[24] E. Shortliffe, *Computer-based medical consultations: MYCIN*. Elsevier, 2012, vol. 2.

[25] A. Meyer, D. Zverinski, B. Pfahringer, J. Kempfert, T. Kuehne, S. H. Sündermann, C. Stamm, T. Hofmann, V. Falk, and C. Eickhoff, "Machine learning for real-time prediction of complications in critical care: a retrospective study," *The Lancet Respiratory Medicine*, vol. 6, no. 12, pp. 905–914, 2018.

[26] T. Yue and H. Wang, "Deep learning for genomics: A concise overview," *arXiv preprint arXiv:1802.00810*, 2018.

[27] S. Chilamkurthy, R. Ghosh, S. Tanamala, M. Biviji, N. G. Campeau, V. K. Venugopal, V. Mahajan, P. Rao, and P. Warier, "Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study," *The Lancet*, vol. 392, no. 10162, pp. 2388–2396, 2018.

[28] Q. Li, J. Lin, N. T. Clancy, and D. S. Elson, "Estimation of tissue oxygen saturation from rgb images and sparse hyperspectral signals based on conditional generative adversarial network," *International journal of computer assisted radiology and surgery*, pp. 1–9, 2019.

[29] N. Stephenson, E. Shane, J. Chase, J. Rowland, D. Ries, N. Justice, J. Zhang, L. Chan, and R. Cao, "Survey of machine learning techniques in drug discovery," *Current drug metabolism*, vol. 20, no. 3, pp. 185–193, 2019.

[30] N. Cristianini, J. Shawe-Taylor *et al.*, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[31] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[32] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[33] K. Krishna and N. M. Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems Man And Cybernetics-Part B: Cybernetics*, vol. 29, no. 3, pp. 433–439, 1999.

[34] D. Birant and A. Kut, "St-dbscan: An algorithm for clustering spatial–temporal data," *Data & Knowledge Engineering*, vol. 60, no. 1, pp. 208–221, 2007.

[35] R. Chellappa and A. Jain, "Markov random fields. theory and application," *Boston: Academic Press, 1993, edited by Chellappa, Rama; Jain, Anil*, 1993.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[39] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[40] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[41] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[42] P. Khosravi, E. Kazemi, M. Imielinski, O. Elemento, and I. Hajirasouliha, "Deep convolutional neural networks enable discrimination of heterogeneous digital pathology images," *EBioMedicine*, vol. 27, pp. 317–328, 2018.

[43] X. Li, S. Zhang, Q. Zhang, X. Wei, Y. Pan, J. Zhao, X. Xin, C. Qin, X. Wang, J. Li *et al.*, "Diagnosis of thyroid cancer using deep convolutional neural network models applied to sonographic images: a retrospective, multicohort, diagnostic study," *The Lancet Oncology*, vol. 20, no. 2, pp. 193–201, 2019.

[44] E. Rubinstein, M. Salhov, M. Nidam-Leshem, V. White, S. Golan, J. Baniel, H. Bernstine, D. Groshar, and A. Averbuch, "Unsupervised tumor detection in dynamic PET/CT imaging of the prostate," *Medical Image Analysis*, vol. 55, pp. 27–40, 2019.

[45] M. Winkels and T. S. Cohen, "Pulmonary nodule detection in CT scans with equivariant CNNs," *Medical Image Analysis*, vol. 55, pp. 15–26, 2019.

[46] F. Liu, Z. Zhou, A. Samsonov, D. Blankenbaker, W. Larison, A. Kanarek, K. Lian, S. Kambhampati, and R. Kijowski, "Deep learning approach for evaluating knee MR images: achieving high diagnostic performance for cartilage lesion detection," *Radiology*, vol. 289, no. 1, pp. 160–169, 2018.

[47] G. Maicas, G. Carneiro, A. P. Bradley, J. C. Nascimento, and I. Reid, "Deep reinforcement learning for active breast lesion detection from DCE-MRI," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2017, pp. 665–673.

[48] H. Lee, S. Yune, M. Mansouri, M. Kim, S. H. Tajmir, C. E. Guerrier, S. A. Ebert, S. R. Pomerantz, J. M. Romero, S. Kamalian *et al.*, "An explainable deep-learning algorithm for the detection of acute intracranial haemorrhage from small datasets," *Nature Biomedical Engineering*, vol. 3, no. 3, p. 173, 2019.

[49] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Medical Image Analysis*, vol. 36, pp. 61–78, 2017.

[50] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2016, pp. 424–432.

[51] E. Gibson, F. Giganti, Y. Hu, E. Bonmati, S. Bandula, K. Gurusamy, B. Davidson, S. P. Pereira, M. J. Clarkson, and D. C. Barratt, "Automatic multi-organ segmentation on abdominal CT with dense v-networks," *IEEE Transactions on Medical Imaging*, vol. 37, no. 8, pp. 1822–1834, 2018.

[52] G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin *et al.*, "Interactive medical image segmentation using deep learning with image-specific fine tuning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 7, pp. 1562–1573, 2018.

[53] W. Bai, H. Suzuki, C. Qin, G. Tarroni, O. Oktay, P. M. Matthews, and D. Rueckert, "Recurrent neural networks for aortic image sequence segmentation with sparse annotations," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2018, pp. 586–594.

[54] I. Laina, N. Rieke, C. Rupprecht, J. P. Vizcaíno, A. Eslami, F. Tombari, and N. Navab, "Concurrent segmentation and localization for tracking of surgical instruments," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2017, pp. 664–672.

[55] N. Lessmann, B. van Ginneken, P. A. de Jong, and I. Išgum, "Iterative fully convolutional neural networks for automatic vertebra segmentation and identification," *Medical Image Analysis*, vol. 53, pp. 142–155, 2019.

[56] X. Feng, J. Yang, A. F. Laine, and E. D. Angelini, "Discriminative localization in CNNs for weakly-supervised segmentation of pulmonary nodules," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2017, pp. 568–576.

[57] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "VoxelMorph: a learning framework for deformable medical image registration," *IEEE Transactions on Medical Imaging*, 2019.

[58] Z. Shen, X. Han, Z. Xu, and M. Niethammer, "Networks for joint affine and non-parametric image registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4224–4233.

[59] K. A. Eppenhof and J. P. Pluim, "Pulmonary CT registration through supervised learning with convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 38, no. 5, pp. 1097–1105, 2018.

[60] Y. Hu, M. Modat, E. Gibson, W. Li, N. Ghavami, E. Bonmati, G. Wang, S. Bandula, C. M. Moore, M. Emberton *et al.*, "Weakly-supervised convolutional neural networks for multimodal image registration," *Medical Image Analysis*, vol. 49, pp. 1–13, 2018.

[61] S. Miao, S. Piat, P. Fischer, A. Tuysuzoglu, P. Mewes, T. Mansi, and R. Liao, "Dilated FCN for multi-agent 2D/3D medical image registration," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2018.

[62] J. Fan, X. Cao, P.-T. Yap, and D. Shen, "BIRNet: Brain image registration using dual-supervised fully convolutional networks," *Medical Image Analysis*, vol. 54, pp. 193–206, 2019.

[63] B. D. de Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring, and I. Išgum, "A deep learning framework for unsupervised affine and deformable image registration," *Medical Image Analysis*, vol. 52, pp. 128–143, 2019.

[64] H. Sokooti, B. de Vos, F. Berendsen, B. P. Lelieveldt, I. Išgum, and M. Staring, "Nonrigid image registration using multi-scale 3D convolutional neural networks," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2017, pp. 232–239.

[65] R. Liao, S. Miao, P. de Tournemire, S. Grbic, A. Kamen, T. Mansi, and D. Comaniciu, "An artificial agent for robust image registration," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2017.

[66] D. Cool, D. Downey, J. Izawa, J. Chin, and A. Fenster, "3D prostate model formation from non-parallel 2D ultrasound biopsy images," *Medical Image Analysis*, vol. 10, no. 6, pp. 875–887, 2006.

[67] D. Toth, M. Pfister, A. Maier, M. Kowarschik, and J. Hornegger, "Adaption of 3D models to 2D X-ray images during endovascular abdominal aneurysm repair," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015, pp. 339–346.

[68] J.-Q. Zheng, X.-Y. Zhou, C. Riga, and G.-Z. Yang, "Real-time 3D shape instantiation for partially deployed stent segments from a single 2D fluoroscopic image in fenestrated endovascular aortic repair," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3703–3710, 2019.

[69] ——, "3D path planning from a single 2D fluoroscopic image for robot assisted fenestrated endovascular aortic repair," *arXiv preprint arXiv:1809.05955*, 2018.

[70] S.-L. Lee, A. Chung, M. Lerotic, M. A. Hawkins, D. Tait, and G.-Z. Yang, "Dynamic shape instantiation for intra-operative guidance," in *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2010, pp. 69–76.

[71] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial intelligence review*, vol. 18, no. 2, pp. 77–95, 2002.

[72] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 2012.

[73] X.-Y. Zhou, J. Lin, C. Riga, G.-Z. Yang, and S.-L. Lee, "Real-time 3-d shape instantiation from single fluoroscopy projection for fenestrated stent graft deployment," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1314–1321, 2018.

[74] J.-Q. Zheng, X.-Y. Zhou, Q.-B. Li, C. Riga, and G.-Z. Yang, "Abdominal aortic aneurysm segmentation with a small number of training subjects," *arXiv preprint arXiv:1804.02943*, 2018.

[75] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[76] X. Li, Q. Dou, H. Chen, C.-W. Fu, X. Qi, D. L. Belavỳ, G. Armbrecht, D. Felsenberg, G. Zheng, and P.-A. Heng, "3d multi-scale fcn with random modality voxel dropout learning for intervertebral disc localization and segmentation from multi-modality mr images," *Medical image analysis*, vol. 45, pp. 41–54, 2018.

[77] O. Bernard, A. Lalande, C. Zotti, F. Cervenansky, X. Yang, P.-A. Heng, I. Cetin, K. Lekadir, O. Camara, M. A. G. Ballester *et al.*, "Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: Is the problem solved?" *IEEE Transactions on Medical Imaging*, 2018.

[78] Z. Xiong, V. V. Fedorov, X. Fu, E. Cheng, R. Macleod, and J. Zhao, "Fully automatic left atrium segmentation from late gadolinium enhanced magnetic resonance imaging using a dual fully convolutional neural network," *IEEE Transactions on Medical Imaging*, 2018.

[79] J. Zhang, A. Saha, Z. Zhu, and M. A. Mazurowski, "Hierarchical convolutional neural networks for segmentation of breast tumors in mri with application to radiogenomics," *IEEE transactions on medical imaging*, 2018.

[80] K. López-Linares, N. Aranjuelo, L. Kabongo, G. Maclair, N. Lete, M. Ceresa, A. García-Familiar, I. Macía, and M. A. G. Ballester, "Fully automatic detection and segmentation of abdominal aortic thrombus in post-operative cta images using deep convolutional neural networks," *Medical image analysis*, vol. 46, pp. 202–214, 2018.

[81] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[82] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization," in *NeurIPS*, 2018, pp. 7705–7716.

[83] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *NeurIPS*, 2018, pp. 2488–2498.

[84] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *NeurIPS*, 2017, pp. 1945–1953.

[85] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[86] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *stat*, vol. 1050, p. 21, 2016.

[87] Y. Wu and K. He, "Group normalization," *arXiv preprint arXiv:1803.08494*, 2018.

[88] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 901–909.

[89] Y. Xu and X. Wang, "Understanding weight normalized deep neural networks with rectified linear units," in *NeurIPS*, 2018, pp. 130–139.

[90] G. Wang, J. Peng, P. Luo, X. Wang, and L. Lin, "Batch kalman normalization: Towards training deep neural networks with micro-batches," *arXiv preprint arXiv:1802.03133*, 2018.

[91] E. Hoffer, R. Banner, I. Golan, and D. Soudry, "Norm matters: efficient and accurate normalization schemes in deep networks," in *NeurIPS*, 2018, pp. 2164–2174.

[92] D. Arpit, Y. Zhou, B. Kota, and V. Govindaraju, "Normalization propagation: A parametric technique for removing internal covariate shift in deep networks," in *ICML*, 2016, pp. 1168–1176.

[93] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *ICLR*, 2018. [Online]. Available: https://openreview.net/forum?id=B1QRgziT-

[94] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *NeurIPS*, 2016, pp. 2234–2242.

[95] C. Luo, J. Zhan, X. Xue, L. Wang, R. Ren, and Q. Yang, "Cosine normalization: Using cosine similarity instead of dot product in neural networks," in *International Conference on Artificial Neural Networks*. Springer, 2018, pp. 382–391.

[96] H. Nam and H.-E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," *arXiv preprint arXiv:1805.07925*, 2018.

[97] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[98] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization. corr (2016)," *arXiv preprint arXiv:1607.08022*, 2016.

[99] K. Lekadir, R. Merrifield, and G.-Z. Yang, "Outlier detection and handling for robust 3-d active shape models search," *IEEE Transactions on Medical Imaging*, vol. 26, no. 2, pp. 212–222, 2007.

[100] O. Jimenez-del Toro, H. Müller, M. Krenn, K. Gruenberg, A. A. Taha, M. Winterstein, I. Eggel, A. Foncubierta-Rodríguez, O. Goksel, A. Jakab *et al.*, "Cloud-based evaluation of anatomical structure segmentation and landmark detection algorithms: Visceral anatomy benchmarks," *IEEE transactions on medical imaging*, vol. 35, no. 11, pp. 2459–2475, 2016.

[101] P. Radau, Y. Lu, K. Connelly, G. Paul, A. Dick, and G. Wright, "Evaluation framework for algorithms segmenting short axis cardiac mri," *The MIDAS Journal-Cardiac MR Left Ventricle Segmentation Challenge*, vol. 49, 2009.

[102] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le, "Don't decay the learning rate, increase the batch size," *arXiv preprint arXiv:1711.00489*, 2017.

[103] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2999–3007.

[104] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 12, pp. 2481–2495, 2017.

[105] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[106] W. Li, G. Wang, L. Fidon, S. Ourselin, M. J. Cardoso, and T. Vercauteren, "On the compactness, efficiency, and representation of 3d convolutional networks: brain parcellation as a pretext task," in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 348–360.

[107] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," *arXiv preprint arXiv:1702.08502*, 2017.

[108] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[109] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3309–3318.

[110] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 4898–4906.

[111] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[112] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, vol. 1, no. 2, 2017, p. 3.

[113] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 4467–4475.

[114] D. M. Vigneault, W. Xie, C. Y. Ho, D. A. Bluemke, and J. A. Noble, "ω-net (omega-net): Fully automatic, multi-view cardiac mr detection, orientation, and segmentation with deep neural networks," *Medical image analysis*, vol. 48, pp. 95–106, 2018.

[115] X. Huang, J. Moore, G. Guiraudon, D. L. Jones, D. Bainbridge, J. Ren, and T. M. Peters, "Dynamic 2D ultrasound and 3D CT image registration of the beating heart," *IEEE transactions on medical imaging*, vol. 28, no. 8, pp. 1179–1189, 2009.

[116] G. Gao, G. Penney, Y. Ma, N. Gogin, P. Cathier, A. Arujuna, G. Morton, D. Caulfield, J. Gill, C. A. Rinaldi, J. Hancock, S. Redwood, M. Thomas, R. Razavi, G. Gijsbers, and K. Rhode, "Registration of 3D trans-esophageal echocardiography to X-ray fluoroscopy using image-based probe tracking," *Medical image analysis*, vol. 16, no. 1, pp. 38–49, 2012.

[117] S. Filippi, B. Motyl, and C. Bandera, "Analysis of existing methods for 3D modelling of femurs starting from two orthogonal images and development of a script for a commercial software package," *Computer methods and programs in biomedicine*, vol. 89, no. 1, pp. 76–82, 2008.

[118] A. F. Frangi, D. Rueckert, J. A. Schnabel, and W. J. Niessen, "Automatic construction of multiple-object three-dimensional statistical shape models: Application to cardiac modeling," *IEEE transactions on medical imaging*, vol. 21, no. 9, pp. 1151–1166, 2002.

[119] K. Koh, Y. H. Kim, K. Kim, and W. M. Park, "Reconstruction of patient-specific femurs using X-ray and sparse CT images," *Computers in biology and medicine*, vol. 41, no. 7, pp. 421–426, 2011.

[120] V. Karade and B. Ravi, "3D femur model reconstruction from biplane X-ray images: a novel method based on laplacian surface deformation," *International journal of computer assisted radiology and surgery*, vol. 10, no. 4, pp. 473–485, 2015.

[121] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[122] D. C. Barratt, C. S. Chan, P. J. Edwards, G. P. Penney, M. Slomczykowski, T. J. Carter, and D. J. Hawkes, "Instantiation and registration of statistical shape models of the femur and pelvis using 3D ultrasound imaging," *Medical image analysis*, vol. 12, no. 3, pp. 358–374, 2008.

[123] K. T. Rajamani, M. A. Styner, H. Talib, G. Zheng, L. P. Nolte, and M. A. G. Ballester, "Statistical deformable bone models for robust 3D surface extrapolation from sparse data," *Medical Image Analysis*, vol. 11, no. 2, pp. 99–109, 2007.

[124] N. Baka, B. Kaptein, M. de Bruijne, T. van Walsum, J. Giphart, W. J. Niessen, and B. P. Lelieveldt, "2D-3D shape reconstruction of the distal femur from stereo X-ray imaging using statistical shape models," *Medical image analysis*, vol. 15, no. 6, pp. 840–850, 2011.

[125] S.-L. Lee, P. Horkaew, W. Caspersz, A. Darzi, and G.-Z. Yang, "Assessment of shape variation of the levator ani with optimal scan planning and statistical shape modeling," *Journal of computer assisted tomography*, vol. 29, no. 2, pp. 154–162, 2005.

[126] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[127] I. T. Jolliffe, "Rotation of principal components: choice of normalization constraints," *Journal of Applied Statistics*, vol. 22, no. 1, pp. 29–35, 1995.

[128] S. Vines, "Simple principal components," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 49, no. 4, pp. 441–451, 2000.

[129] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin, "A modified principal component technique based on the lasso," *Journal of computational and Graphical Statistics*, vol. 12, no. 3, pp. 531–547, 2003.

[130] J. Cadima and I. T. Jolliffe, "Loading and correlations in the interpretation of principle compenents," *Journal of Applied Statistics*, vol. 22, no. 2, pp. 203–214, 1995.

[131] K. Sjöstrand, L. H. Clemmensen, R. Larsen, and B. Ersbøll, "SpaSM: A matlab toolbox for sparse statistical modeling," http://www2.imm.dtu.dk/projects/spasm/, 2012, accessed: 2017-11-25.

[132] L. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–135, 1993.

[133] N. A. Ablitt, J. Gao, J. Keegan, L. Stegger, D. N. Firmin, and G.-Z. Yang, "Predictive cardiac motion modeling and correction with partial least squares regression," *IEEE transactions on medical imaging*, vol. 23, no. 10, pp. 1315–1324, 2004.

[134] F. Duan, D. Huang, Y. Tian, K. Lu, Z. Wu, and M. Zhou, "3D face reconstruction from skull by regression modeling in shape parameter spaces," *Neurocomputing*, vol. 151, pp. 674–682, 2015.

[135] R. Rosipal and L. J. Trejo, "Kernel partial least squares regression in reproducing kernel hilbert space," *Journal of machine learning research*, vol. 2, no. Dec, pp. 97–123, 2001.

[136] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Subspace, latent structure and feature selection*. Springer, 2006, pp. 34–51.

[137] R. Razavi, D. L. Hill, S. F. Keevil, M. E. Miquel, V. Muthurangu, S. Hegde, K. Rhode, M. Barnett, J. van Vaals, D. J. Hawkes, and E. Baker, "Cardiac catheterisation guided by MRI in children and adults with congenital heart disease," *The Lancet*, vol. 362, no. 9399, pp. 1877–1882, 2003.

[138] M. Saeed, D. Saloner, O. Weber, A. Martin, C. Henk, and C. Higgins, "MRI in guiding and assessing intramyocardial therapy," *European radiology*, vol. 15, no. 5, pp. 851–863, 2005.

[139] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[140] S. De Jong, "SIMPLS: an alternative approach to partial least squares regression," *Chemometrics and intelligent laboratory systems*, vol. 18, no. 3, pp. 251–263, 1993.

[141] Manu, "nonrigidicp," https://uk.mathworks.com/matlabcentral/fileexchange/41396-nonrigidicp, 2016, accessed: 2016-02-20.

[142] D.-J. Kroon, "Shape context based corresponding point models," https://uk.mathworks.com/matlabcentral/fileexchange/

30845-shape-context-based-corresponding-point-models, 2016, accessed: 2016-02-20.

[143] W. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B. M. Tsui, "4D XCAT phantom for multimodality imaging research," *Medical physics*, vol. 37, no. 9, pp. 4902–4915, 2010.

[144] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: an open-source mesh processing tool," in *Eurographics Italian Chapter Conference*, V. Scarano, R. D. Chiara, and U. Erra, Eds. The Eurographics Association, 2008.

[145] M. Lerotic and S.-L. Lee, "A multimodal silicone phantom for robotic surgical training and simulation," in *The Hamlyn Symposium on Medical Robotics, London, UK*, 2010, pp. 65–66.

[146] L. Mercier, T. Langø, F. Lindseth, and L. D. Collins, "A review of calibration techniques for freehand 3D ultrasound systems," *Ultrasound in medicine & biology*, vol. 31, no. 2, pp. 143–165, 2005.

[147] K. R. Gegenfurtner, "Praxis: Brent's algorithm for function minimization," *Behavior Research Methods, Instruments, & Computers*, vol. 24, no. 4, pp. 560–564, 1992.

[148] R. W. Prager, R. N. Rohling, A. Gee, and L. Berman, "Rapid calibration for 3D freehand ultrasound," *Ultrasound in medicine & biology*, vol. 24, no. 6, pp. 855–869, 1998.

[149] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 294–302, 1995.

[150] M. Lerotic, S.-L. Lee, J. Keegan, and G.-Z. Yang, "Image constrained finite element modelling for real-time surgical simulation and guidance," in *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on.* IEEE, 2009, pp. 1063–1066.

[151] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3D finite element mesh generator with built-in pre-and post-processing facilities," *International*

Journal for Numerical Methods in Engineering, vol. 79, no. 11, pp. 1309–1331, 2009.

[152] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni, "Sofa - an open source framework for medical simulation," in *MMVR 15-Medicine Meets Virtual Reality*, vol. 125. IOP Press, 2007, pp. 13–18.

[153] W.-C. Yeh, P.-C. Li, Y.-M. Jeng, H.-C. Hsu, P.-L. Kuo, M.-L. Li, P.-M. Yang, and P. H. Lee, "Elastic modulus measurements of human liver and correlation with pathology," *Ultrasound in medicine & biology*, vol. 28, no. 4, pp. 467–474, 2002.

[154] J. Cross, K. Gurusamy, V. Gadhvi, D. Simring, P. Harris, K. Ivancev, and T. Richards, "Fenestrated endovascular aneurysm repair," *British Journal of Surgery*, vol. 99, no. 2, pp. 152–159, 2012.

[155] T. Resch, "Custom-made devices: Current state of the art," http://evtoday.com/2016/03/custom-made-devices-current-state-of-the-art/, 2016, accessed: 2016-02-20.

[156] C. V. Riga, C. D. Bicknell, A. Rolls, N. J. Cheshire, and M. S. Hamady, "Robot-assisted fenestrated endovascular aneurysm repair (FEVAR) using the magellan system," *Journal of Vascular and Interventional Radiology*, vol. 24, no. 2, pp. 191–196, 2013.

[157] D. Volpi, M. H. Sarhan, R. Ghotbi *et al.*, "Online tracking of interventional devices for endovascular aortic repair," *IJCARS*, vol. 10, no. 6, pp. 773–781, 2015.

[158] L. Gundelwein, J. Miró, and L. Duong, "Automatic stent placement and stent size selection for preoperative planning of pulmonary artery intervention," in *Joint MICCAI workshops on CVII-STENT(2015)*, 2015.

[159] S. Demirci, A. Bigdelou, L. Wang *et al.*, "3D stent recovery from one x-ray projection," in *MICCAI 2011*. Springer, 2011, pp. 178–185.

[160] S. Reiml, M. Pfister, D. Toth *et al.*, "Automatic detection of stent graft markers in 2D fluoroscopy images," in *Joint MICCAI workshops on CVII-STENT (2015)*, 2015.

[161] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[162] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *arXiv preprint arXiv:1802.02611*, 2018.

[163] P. F. Christ, M. E. A. Elshaer, F. Ettlinger, S. Tatavarty, M. Bickel, P. Bilic, M. Rempfler, M. Armbruster, F. Hofmann, M. D'Anastasi *et al.*, "Automatic liver and lesion segmentation in ct using cascaded fully convolutional neural networks and 3d conditional random fields," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 415–423.

[164] Y. Zhou, L. Xie, E. K. Fishman, and A. L. Yuille, "Deep supervision for pancreatic cyst segmentation in abdominal ct scans," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 222–230.

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

[166] A. BenTaieb and G. Hamarneh, "Topology aware fully convolutional networks for histology gland segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 460–468.

[167] H. Ravishankar, R. Venkataramani, S. Thiruvenkadam, P. Sudhakar, and V. Vaidya, "Learning and incorporating shape models for semantic segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 203–211.

[168] N. Demanget, S. Avril, P. Badel *et al.*, "Computational comparison

of the bending behavior of aortic stent-grafts," *J MECH BEHAV BIOMED*, vol. 5, no. 1, pp. 272–282, 2012.

[169] S. Li and C. Xu, "A stable direct solution of perspective-three-point problem," *IJPRAI*, vol. 25, no. 05, pp. 627–642, 2011.

[170] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *PAMI*, no. 4, pp. 376–380, 1991.

[171] Wikipedia, "Rotation matrix," https://en.wikipedia.org/wiki/Rotation_m\atrix#Axis_and_angle, 2017, accessed: 2017-11-17.

[172] S. Pieper, M. Halle, and R. Kikinis, "3D slicer," in *Biomedical imaging: nano to macro, IEEE international symposium on.* IEEE, 2004, pp. 632–635.

[173] D. Girardeau-montaut, "Cloudcompare [v2.6.2], 3D point cloud and mesh processing free software," EDF Research and Development, Telecom ParisTech. Available: http://cloudcompare.org/, Tech. Rep., 2015.