University of London
Imperial College of Science, Technology and Medicine
Department of Computing

Fast Catheter segmentation and tracking based on X-ray fluoroscopic and echocardiographic modalities for catheter-based cardiac minimally invasive interventions

Xianliang Wu

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the University of London and
the Diploma of Imperial College, October 2014

Abstract

X-ray fluoroscopy and echocardiography imaging (ultrasound, US) are two imaging modalities that are widely used in cardiac catheterization. For these modalities, a fast, accurate and stable algorithm for the detection and tracking of catheters is required to allow clinicians to observe the catheter location in real-time. Currently X-ray fluoroscopy is routinely used as the standard modality in catheter ablation interventions. However, it lacks the ability to visualize soft tissue and uses harmful radiation. US does not have these limitations but often contains acoustic artifacts and has a small field of view. These make the detection and tracking of the catheter in US very challenging.

The first contribution in this thesis is a framework which combines Kalman filter and discrete optimization for multiple catheter segmentation and tracking in X-ray images. Kalman filter is used to identify the whole catheter from a single point detected on the catheter in the first frame of a sequence of x-ray images. An energy-based formulation is developed that can be used to track the catheters in the following frames. We also propose a discrete optimization for minimizing the energy function in each frame of the X-ray image sequence. Our approach is robust to tangential motion of the catheter and combines the tubular and salient feature measurements into a single robust and efficient framework.

The second contribution is an algorithm for catheter extraction in 3D ultrasound images based on (a) the registration between the X-ray and ultrasound images and (b) the segmentation of the catheter in X-ray images. The search space for the catheter extraction in the ultrasound images is constrained to lie on or close to a curved surface in the ultrasound volume. The curved surface corresponds to the back-projection of the extracted catheter from the X-ray image to the ultrasound volume. Blob-like features are detected in the US images and organized in a graphical model. The extracted catheter is modelled as the optimal path in this graphical model.

Both contributions allow the use of ultrasound imaging for the improved visualization of soft tissue. However, X-ray imaging is still required for each ultrasound frame and the amount of X-ray exposure has not been reduced. The final contribution in this thesis is a system that can track the catheter in ultrasound volumes automatically without the need for X-ray imaging during the tracking. Instead X-ray imaging is only required for the system initialization and for recovery from tracking failures. This allows a significant reduction in the amount of X-ray exposure for patient and clinicians.

Acknowledgements

This thesis is based on my research work as a PhD candidate at Imperial College London. During this period, I have received lots of support from other people.

There is no doubt that the most important support to complete this thesis comes from the academic circle. Firstly, I would like to gratefully thank my supervisor, Prof. Daniel Rueckert, for his help through all my PhD period. The knowledge and experience, which he has shared with me, has greatly changed my vision and furthermore changed my life. I cannot forget every Friday's morning at sharp nine, when an one-to-one meeting was held at Daniel's office. Secondly, I also would like to thank Prof. Kawal Rhode and Dr. James Houston from King's College, for their support for my experiments, as well as improving my English. Thirdly, I would like to thank Prof. Wayne Luk and Prof. Alejandro Frangi, who accompanied me through the last recall of my research and have witnessed one of the biggest moment in my life.

In addition to the academic circle, the support from several administrative organizations has been highly appreciated. I am grateful to my funding body, the Chinese Scholarship Council (CSC), who has provided my tuition fees and living expenses for me. I promise the money you have spent on me will be a sound investment. Also, the support staff in the Department of Computing and at Imperial College helped me a lot during my PhD period. Thanks to them, for without their support, I would have not been able to complete my research so smoothly.

At last, many people may not understand what I have written down in this thesis, but their emotional support does count and their contributions to this thesis should not be ignored. I would like to thank all my friends in London and at home. With their sharing, I enjoyed my PhD life very much. The exhausting daily work did not depress me because the courage you have given to me. I give my greatest thanks to my parents, who always be with me anytime. I do miss you all when I am in London. Also, By the way, thanks to my future wife and my future children, though I have never seen you and have no idea where you are, certainly you will come into my life one day. Thanks for not stepping into my life so early so I can focus on research.

Contents

Abstract **i**

Acknowledgements **iii**

1 Introduction **1**

1.1 Motivation 1

1.1.1 Minimally Invasive Surgery 1

1.1.2 Minimally Invasive Cardiac Surgery 3

1.1.3 Cardiac Catheterization 3

1.1.4 Common cardiac diseases that can be treated using catheterization . . . 8

1.1.5 Imaging Modalities 11

1.2 Challenges 21

1.3 Contributions 23

1.4 Thesis Outline 25

2 Background **27**

2.1 Feature detection for segmentation and tracking 27

2.1.1 General Feature Detection 28

2.1.2 Feature Detection using Machine Learning 32

2.2 Optimization and Filtering 40

2.2.1 Optimization 40

2.2.2 Filtering 43

2.3 General Segmentation and Tracking Applications in Computer Vision 45

2.3.1 Segmentation 45

2.3.2 Tracking 49

2.4 Catheter Segmentation and Tracking for Cardiac Interventions 56

2.4.1 Catheter Segmentation and Tracking in the X-ray Images 56

2.4.2 Catheter Segmentation and Tracking in the Ultrasound Images 60

2.5 Hardware Acceleration 60

2.5.1 GPU Acceleration 61

2.5.2 FPGA Acceleration 62

2.5.3 CPU vs GPU vs FPGA 62

3 Fast Catheter Segmentation and Tracking in X-ray Fluoroscopic Images 64

3.1 Introduction 64

3.2 Catheter Extraction in X-ray Images 65

3.2.1 Blob Detection and Tubular Structure Detection 66

3.2.2 Patch Analysis 68

3.2.3 Kalman Filter-based Path Growing 73

3.2.4 Catheter Extraction Applied to Tracking in X-ray Sequences 74

3.3 Catheter Tracking in X-ray Images 78

3.3.1 Image Enhancement 78

3.3.2 Overview of the Discrete Optimization Framework 79

3.3.3 Objective Function 82

3.4 Experiments 87

3.4.1 Evaluation of the Catheter Extraction Algorithm 87

3.4.2 Evaluation of the Catheter Tracking Algorithm 92

3.5 Discussion 99

4 Catheter Segmentation in Ultrasound Images using Registered X-ray Images101

4.1 Introduction 101

4.2 Registration of TEE and X-ray Fluoroscopy 102

4.3 Catheter Extraction in Echocardiographic Images Using Shortest Path 103

4.3.1 Feature Detection in a Reduced Search Space 103

4.3.2 Catheter Extraction Based using a Graphical Model and the A⋆ Algorithm105

4.4 Catheter Extraction in Echocardiographic Images Based on Optimal Path 107

4.4.1 Search Space Straightening and SURF Feature Extraction 107

4.4.2 Two-Layer Hierarchical Graph Construction 111

4.4.3 Finding the Optimal Path 114

4.4.4 Post-processing 115

4.5 Experiments 116

4.5.1 Evaluation of the Single Layer Graphical Model 116

4.5.2 Evaluation of the Hierarchical Graphical Model 119

4.6 Discussion 124

5 Real-time Catheter Tracking using Ultrasound without Fluoroscopy 128

5.1 Introduction 128

5.2 Measurements in the Region of Interest 130

5.2.1 B-spline Curve Representation of the Catheter 130

5.2.2 Region of interest definition 130

5.2.3 Measurement of tubular structures 131

5.3 Catheter Tracking based on Two-Stage Fast-PD Optimization 133

5.3.1 Overview of the Fast-PD Framework 133

5.3.2 3D Catheter Tracking Framework 135

5.3.3 Transversal Catheter Tracking 136

5.3.4 Longitudinal Catheter Tracking 136

5.4 Kalman Filter Framework 140

5.5 Experiments 143

5.5.1 Datasets 145

5.5.2 Parameter Selection 147

5.5.3 Parameter Sensitivity 148

5.5.4 Overall Performance Comparison 151

5.6 Discussion 154

6 Conclusion 159

6.1 Summary 159

6.2 Applications 161

6.3 Future Work 162

Bibliography 166

List of Tables

1.1 Some specific procedures using MIS (Table adopted from [159]) 2

1.2 A comparison between X-ray and ultrasound modalities. 17

2.1 Geometric patterns and the corresponding eigenvalues $|\lambda_1| < |\lambda_2|(< |\lambda_3|)$ (Table from [69]) 32

3.1 Selected parameter values for the evaluation of the extraction algorithm 90

3.2 Results on X-ray image sequences for the extraction and tracking algorithm. . . 92

3.3 Parameter values for the evaluation of the tracking algorithm. 96

4.1 Dataset Information 120

5.1 Dataset Information 146

List of Figures

1.1 A comparison between the conventional thoracotomy (left) and MICS (right) for cardiac surgery (Figure from [165]). 4

1.2 A typical Cath Lab (Figure from [210]). 5

1.3 Coronary angiography (Figure from [191, 212, 134]). 6

1.4 Coronary angioplasty procedure (left) and an example of a balloon catheter with a stent for coronary angioplasty (right) (Figure from [133, 135]). 7

1.5 Balloon septostomy procedure (left) and some examples of the balloon catheter for septostomy (right) (Figure from [132, 148]). 7

1.6 Three types of catheters used for EP study and ablation. From the left to the right are an ablation catheter, a coronary sinus catheter and a mapping catheter. 9

1.7 Catheter ablation for atrial fibrillation (Figure from [96]). 10

1.8 An example of bi-plane X-ray intensifier (Allura Xper FD biplane systems developed by Philips) (Figure from [163]). 13

1.9 X-ray images of catheters for clinical (left) and porcine experimental (right) usages. 14

1.10 Artifacts in ultrasound images (Figure from [173]). 15

1.11 An comparison between TTE and TEE procedures (Figure from [136]). 16

1.12 Examples of orginial X-ray images (top) with the corresponding ultrasound volume visualized (bottom) (Figure from [208]). 18

1.13 An example of two types of IVUS catheters and one of example of the IVUS lumen image showing vessel plaque (Figure from [201]). 19

1.14 An example of an OCT catheter (right) and its corresponding image (left)(Figure from [201]). 19

1.15 An example of the EAM specialized lasso catheter (left) and its imaging under EAM mapping (right) (Figure from [23]). 20

1.16 The imaging principle of CBCT (left) and an example of 3D image (right) (Figure from [211, 184]). 21

1.17 A system integrating all contributions. 25

2.1 A example showing Harris corners on the left and DoG blobs. 30

2.2 An example of edge detection with Sobel (bottom left) and Canny (bottom right) operators, and tubular structure detection (top right), compared with the original image (top left). 33

2.3 An example of Hough line detection with two lines in the original coordinate (left) and the corresponding voting in the Hough space. 34

2.4 SIFT descriptor calculation with cells (left) and an example of one cell calculated based on local gradients (right). 35

2.5 HoG descriptor calculation with cells (left) and the histograms based on the gradient statistics (right). 35

2.6 A simple 2D linear SVM case. 37

2.7 An example of a two-layer stacked auto-encoder network with a softmax classifier. 39

2.8 Binary label classification based on max-flow/min-cuts optimization. 41

2.9 Kalman filter framework. 44

2.10 Particle filter framework. 45

2.11 An example of hidden Markov model with hidden states $\mathbf{x}$ in red circles, observable states $\mathbf{y}$ in green boxes and state transition probabilities $\mathbf{a}$ and $\mathbf{b}$. 47

2.12 Examples of GrabCut applications. (Figure from [172]). 48

2.13 An example of the optical flow of an traffic scene (left) and the magnitude of the flow (right) (Figure from [35]). 50

2.14 Tracking results based on online boosting (Figure from [76]). 52

2.15 The overview of the TLD framework. 53

2.16 Tracking of the heart in ultrasound images using ASMs (Figure from [48]). . . . 54

2.17 Pedestrian detection based on part-based model (Figure from [63]). 56

2.18 Catheter tip detection based on the method in [142]) (Figure from [142]). 58

2.19 Illustration of the curve tracking based on discrete optimization on multiple label assignment. 59

2.20 Catheter tip and its corresponding ultrasound image in [38] (Figure from [38]). . 61

3.1 Catheter extraction in an X-ray image (see text for details). 67

3.2 An example of multi-line fitting patch analysis with multiple line segments detected in yellow. 72

3.3 Catheter tracking in X-ray images 75

3.4 Examples for an original image (a), its corresponding tubular map (b), blob map (c), and tip map (d). 80

3.5 Comparison of experimental results on X-ray sequences. 91

3.6 Examples of the tracking results in porcine X-ray data. 93

3.7 Examples of the tracking results in patient X-ray data. 94

3.8 Comparison of curve tracking error and overlap percentage for different error thresholds ρ. 98

3.9 Example of the comparison for multiple catheter tracking with the original frame (a), the result from [86, 85] (b), and the results of our method without RDP (c) and with RDP (d). 99

4.1 Registration of the ultrasound probe model to its X-ray projection (a) and an example of the curved surface in US corresponding to the catheter curve in X-ray (b). 104

4.2 Search space straightening. 109

4.3 The sub-figures show (a) the original features that have been detected (yellow dots in the figure); (b) the top-level graph with each cluster of features as nodes (red); and (c) the bottom-level graph. The original features are organized in a two-layer graph corresponding to each cluster where the features inside the cluster are nodes related by edges (blue) based on distances. 112

4.4 Examples of the catheter extraction using the single-layer graphics model: (a) from phantom images and (b) from real patient. 118

4.5 Performance in 20 pairs from real patient data. 119

4.6 Experimental results for the ultrasound sequences. 122

4.7 Examples of the catheter tracking in ultrasound images from patients. 124

5.1 Search space definition and extraction of the ROI after unwarping. 131

5.2 Tracking of transversal (a) and longitudinal (b) motion. 137

5.3 The definition of the state vector and the measurement model for the Kalman filter. 142

5.4 Sensitivity analysis for λ_1. 150

5.5 Sensitivity analysis for λ_2. 151

5.6 Sensitivity analysis on Kalman filter parameters. 152

5.7 Comparison results only with Type A datasets. 153

5.8 Comparison results with both Type A and Type B datasets. 154

5.9 Examples of the tracking result on Type A (A1) sequence. 155

5.10 Examples of the tracking result on Type B (B7) sequence. 155

6.1 The overall system workflow. 163

Chapter 1

Introduction

1.1 Motivation

1.1.1 Minimally Invasive Surgery

Recent years have seen a rapid development in the area of Minimally Invasive Surgery (MIS), which has already evolved from laboratory experiments to practical applications, and replaced many traditional open surgical procedures.

In an open surgery, a large incision is usually made to allow the surgical procedure to be carried out under the direct vision of the surgeon. In most open surgical procedures, the structures and the tissues involved are exposed and can be observed and touched directly by the surgeon [131].

As defined in [186], a minimally invasive surgery (MIS) is a form of surgery that minimizes the size of surgical incisions in order to reduce trauma and postoperative pain, to avoid large blood loss, and to furthermore decrease the length of the hospital stay. Compared with traditional open surgery, the target structure is usually not directly visible and has to be visualized via imaging techniques such as laparoscopy, fluoroscopy, and ultrasound. The surgical instruments, which are usually specifically designed, are inserted to the target site through small incisions or body cavities and these instruments are manipulated remotely to carry out the surgery.

Table 1.1: Some specific procedures using MIS (Table adopted from [159])

Organ system	**Procedures**
Cardiac	Closing atrial septal defects; Coronary artery bypass graft ('off pump'); Repairing patents' foramen ovale; Valve surgery.
Gastrointestinal	Appendicectomy; Adrenalectomy; Cholecystectomy; Lymph node biopsy; Splenectomy; Hiatus hernia, umbilical and inguinal hernia repairs; Colonic cancer; Diverticular disease; Inflammatory bowel disease; Rectal prolapse; Dividing adhesions.
Gynaecological	Polypectomy; Sterilisation; Endometrial ablation; Fibroid removal.
Neurological	Removal of pituitary tumours; Treatment of intracranial aneurysms; Carotid angioplasty; Radiosurgery for brain tumours.
Orthopaedic	Arthroscopy of joints; Carpal tunnel release; Pelvic fracture repair; Rotator cuff repair.
Otorhinolaryngology	Removal of nasal/sinus tumours; Lymph node biopsy.
Respiratory/Thoracic	Lung surgery; Recurrent pleural effusions.
Urology	Biopsy; Removal of kidney and ureteric calculi; Nephrectomy.
Vascular	Stenting carotid and renal arteries; Repair of thoracic and abdominal aneurysms; Varicose veins.

A minimally invasive surgery has several advantages over traditional open surgery. It can reduce the incisions and the hospitalization time, enhance the manual dexterities of the surgeon and minimize the infection risk. As a result, patients undergoing MIS usually have less trauma, less post-operative pain, shorter hospital stays and smaller scarring. However, to achieve the same or even better performance than traditional open surgery, special imaging techniques, as well as special instruments for interventions, need to be developed. Specialists who are required to manipulate the imaging or surgery equipments need to be trained. In addition, the operative time is usually longer than in conventional open surgery [159].

Currently MIS techniques have been widely used in a range of procedures. Table 1.1 (Table from [159]) summarizes some typical procedures that can be carried out using MIS. We will not elaborate on every procedure since only cardiac procedures are relevant to this thesis.

1.1.2 Minimally Invasive Cardiac Surgery

Minimally Invasive Cardiac Surgery (MICS) is a type of heart surgery that is performed through several small incisions. Before MICS was widely used, a thoracotomy was commonly performed in conventional heart surgery. A thoracotomy is the process of making an incision (usually large) into the chest wall to gain the access to the pleural space of the chest [154]. The heart, the lungs, or the esophagus are the most common targets during a thoracotomy. A thoracotomy incurs many complications such as pneumothorax, infection, bleeding and respiratory failure. In addition, the intense postoperative pain can cause atelectasis or pneumonia [8, 33]. For many heart thoracotomies, the risk is extremely high since the heart has to be stopped during the surgery. In addition the breastbone and ribs must be sometimes separated for a better access to the target region.

Patients can benefit from MICS for the reasons that it can significantly reduce the incisions, trauma and enhance the manual dexterities of the surgeon. Besides, MICS can also be performed without stopping the heart (therefore it eliminates the need for a heart-lung machine) or separating the breastbone. All of the above advantages significantly reduce the risk of complications and decrease the duration of the hospitalization [94, 137, 126]. A comparison between the traditional open-chest thoracotomy with MICS is shown in Fig. 1.1.

Typical procedures in which MICS has been widely used include surgery for ventricular septal defect (VSD), atrial septal defect (ASD), heart valve surgery, multi or single vessel small thoracotomy, coronary artery bypass grafts (CABG) and the treatment of atrial fibrillation (AF) [192].

1.1.3 Cardiac Catheterization

Cardiac catheterization is a category of minimally invasive cardiac surgical procedures in which catheters are inserted into the heart through a small incision in a patient's neck, arm or groin, for either diagnostic or interventional purposes. Certain types of catheters are inserted using guidewires and steered towards the heart, monitored by imaging such as X-ray fluoroscopy or

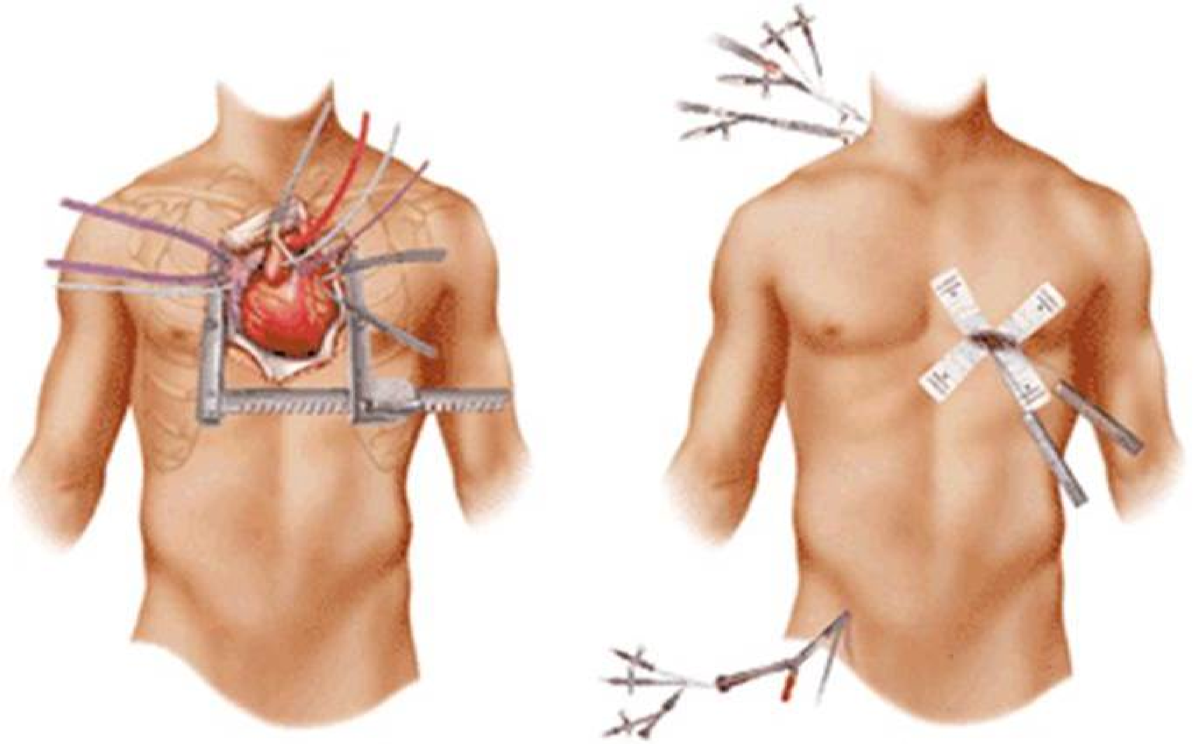


Figure 1.1: A comparison between the conventional thoracotomy (left) and MICS (right) for cardiac surgery (Figure from [165]).

ultrasound. Once in place, they can be used to carry out a number of procedures such as angioplasty, balloon septostomy, electrophysiology studies and catheter ablation [81]. Most of these procedures are carried out in a special examination room, i.e. a catheterization laboratory or Cath Lab. A typical Cath Lab consists of a patient couch, image intensifier, monitors, catheters, guidewires, angioplasty balloons, and stents [210]. Fig. 1.2 shows a typical Cath Lab.

Angiography is a medical imaging method that can visualize either the lumen or the inside of blood vessels by injecting a radio-opaque contrast agent, which can be visualized under X-ray imaging, into the target blood vessels. As a result, the visualization of the blood vessels with the contrast agent under X-ray imaging can be used for diagnostics of vessel occlusion, stenosis, thrombosis or aneurisms. A common use of angiography is to visualize the coronary arteries via coronary catheterization. Catheters are inserted into the target vessel and eject the X-ray contrast agent into the vessels. The distribution of the contrast agent in the coronary arteries allows a visualization of the size of the artery openings, which is an important observation for disease diagnostics [191, 212]. Fig. 1.3 shows the procedure of coronary angiography. The

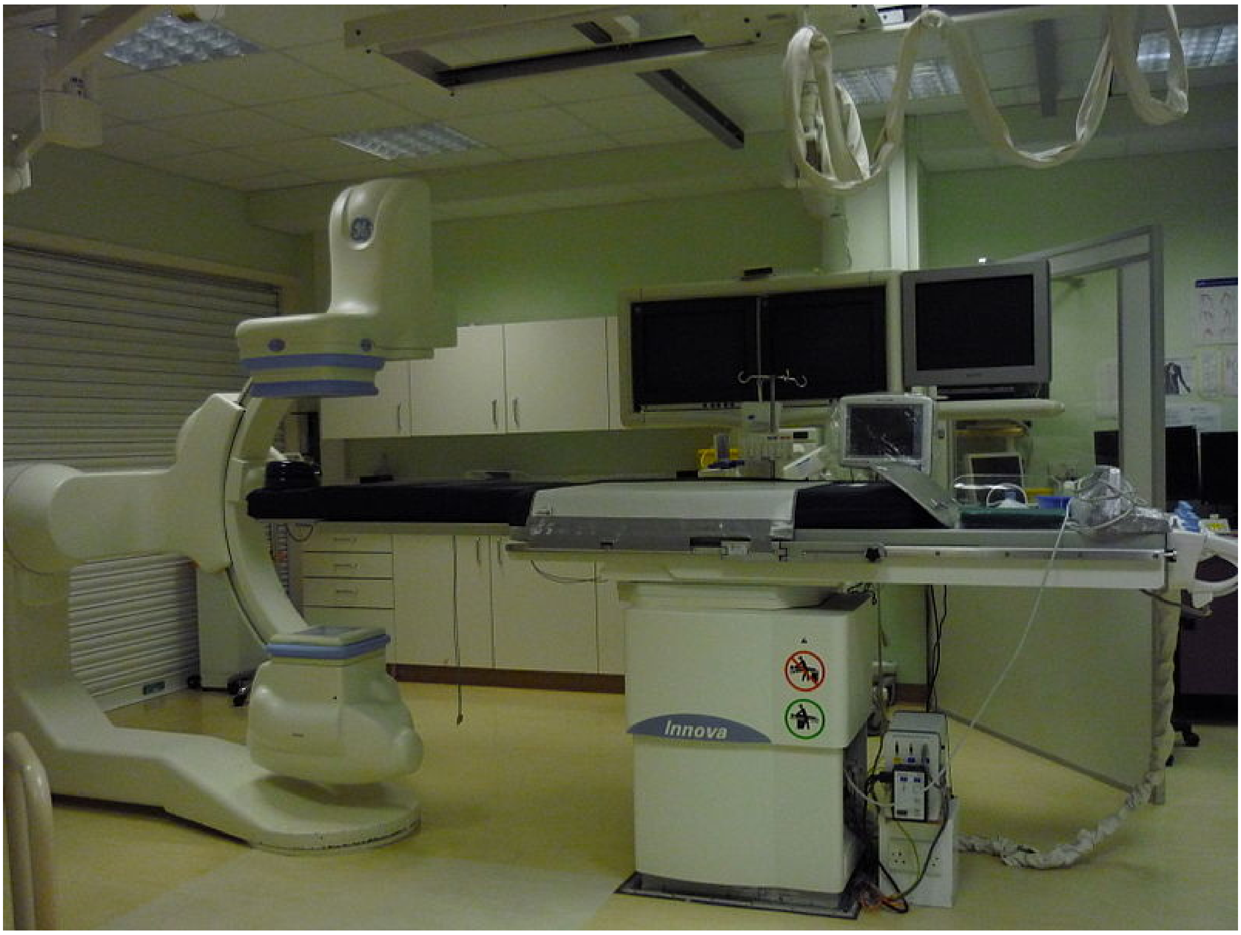


Figure 1.2: A typical Cath Lab (Figure from [210]).

figure also shows an example of the angiography X-ray image (bottom left) and some catheters commonly used in angiography (bottom right).

Angioplasty is an interventional procedure in which narrowed or obstructed arteries are widened mechanically. Obstructed arteries are typical caused by atherosclerosis [135]. During this intervention, a balloon catheter with an empty balloon is guided to the narrowed or obstructed location and then the balloon is inflated to open up the location of the obstruction or widen the narrowed vessel. As a result, the blood flow is improved. Then the balloon is deflated and the entire catheter is retracted. In some cases, a stent may be attached to the balloon catheter to reach the location of the narrowing which is then retained in the location after the catheter is withdrawn in order to keep the vessel open. Coronary angioplasty, also known as percutaneous coronary intervention (PCI), is used to widen the narrowed coronary arteries of the heart [135]. Figure 1.4 shows the procedure of coronary angioplasty as well as a balloon catheter with a stent on the bottom left.

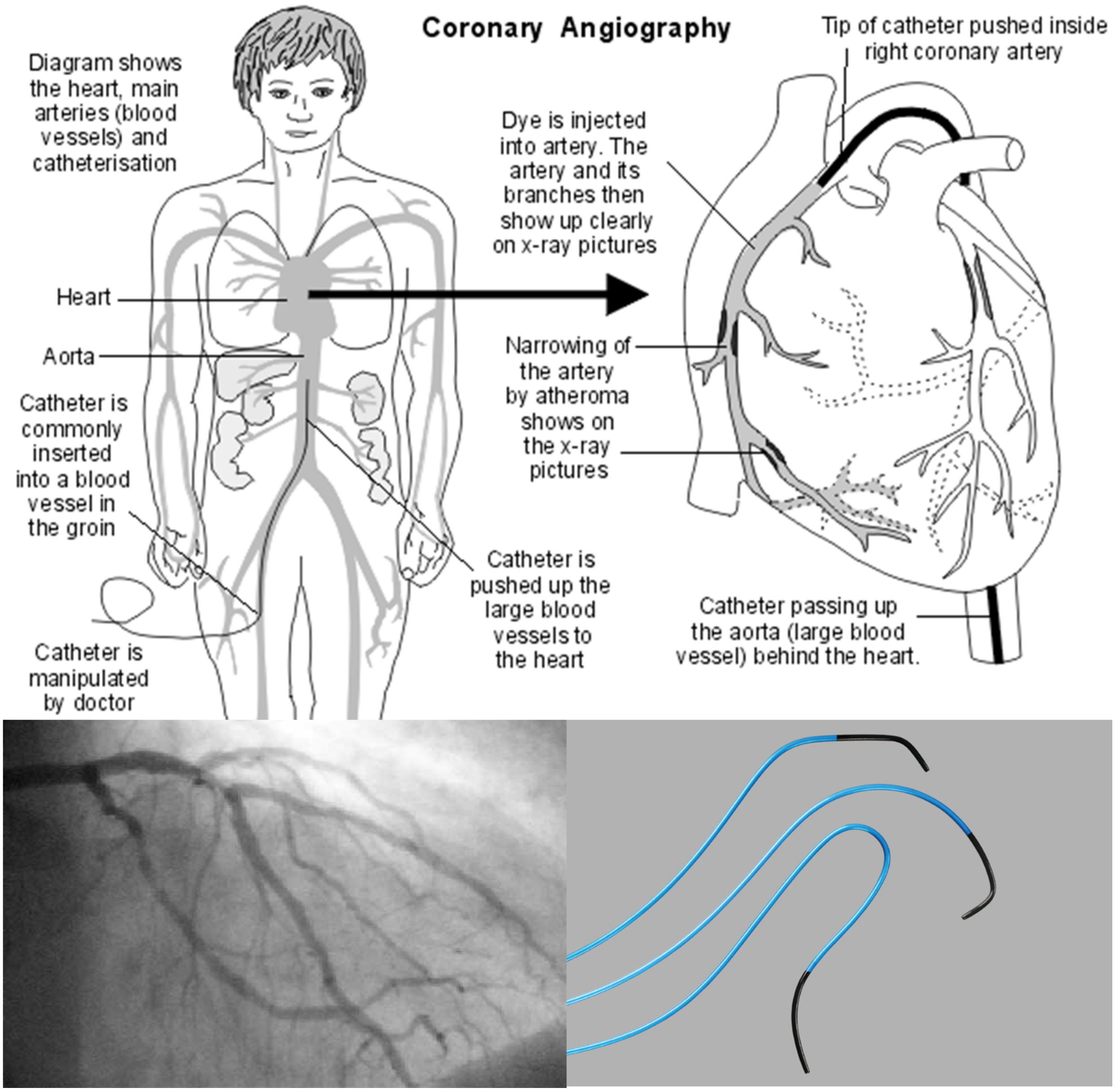

Coronary Angiography
Diagram shows the heart, main arteries (blood vessels) and catheterisation
Tip of catheter pushed inside right coronary artery
Dye is injected into artery. The artery and its branches then show up clearly on x-ray pictures
Heart
Aorta
Narrowing of the artery by atheroma shows on the x-ray pictures
Catheter is commonly inserted into a blood vessel in the groin
Catheter is pushed up the large blood vessels to the heart
Catheter passing up the aorta (large blood vessel) behind the heart.
Catheter is manipulated by doctor

Figure 1.3: Coronary angiography (Figure from [191, 212, 134]).

Balloon septostomy is another cardiac intervention in which an atrial septal defect (ASD) is widened by using a balloon catheter. As a result, a greater amount of oxygenated blood is allowed to enter the systemic circulation in cases of congenital heart defects (CHD). This procedure is similar to angioplasty: The balloon catheter is firstly passed through the ASD into the left atrium from the right atrium, with the balloon deflated. Then the balloon is inflated and the catheter is withdrawn. As result, the ASD is enlarged and a larger amount of blood can pass through. Normally a balloon septostomy is pre-procedure for a corrective surgery and the ASD is expected to be closed by either sutures or cardiac patches when the corrective surgery is ongoing [132]. Fig. 1.5 shows the procedure of balloon septostomy and some examples of the

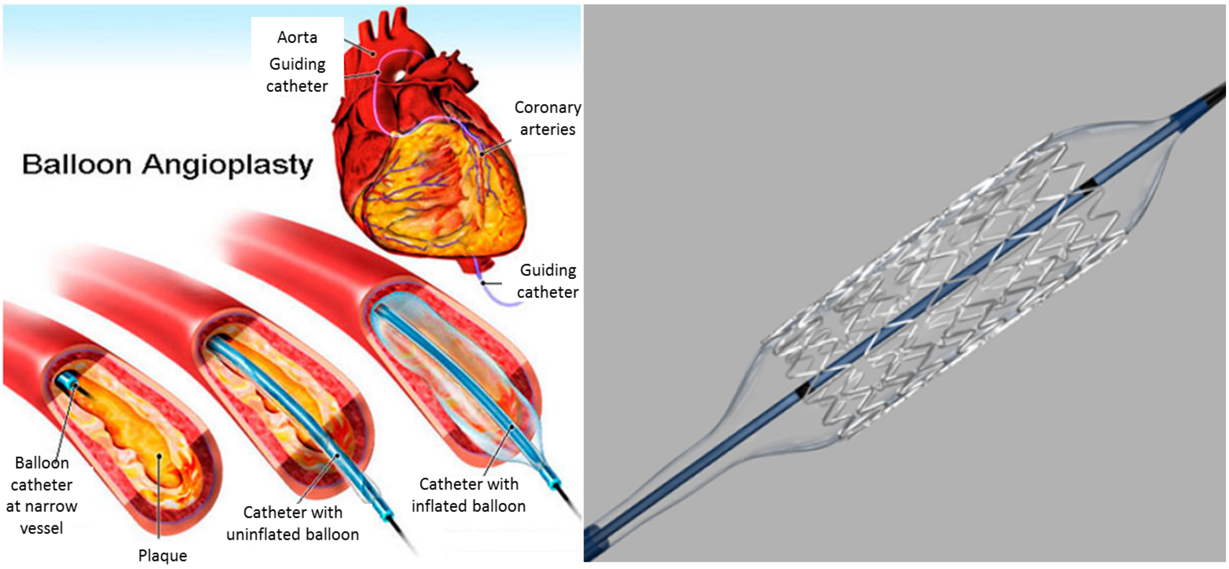
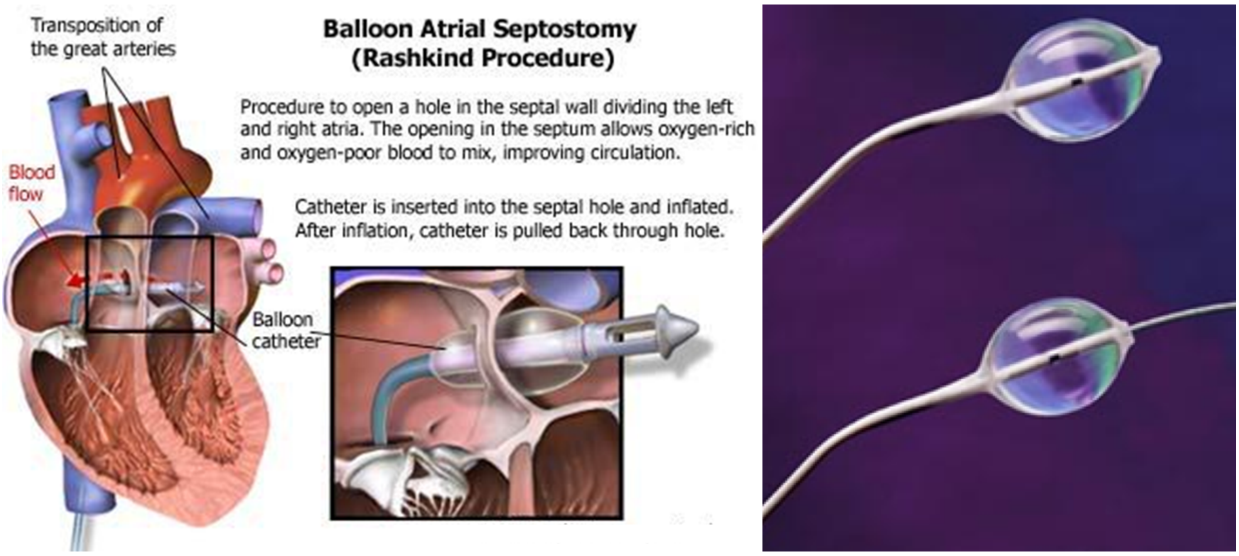

Aorta
Guiding catheter
Coronary arteries
Balloon Angioplasty
Guiding catheter
Balloon catheter at narrow vessel
Plaque
Catheter with uninflated balloon
Catheter with inflated balloon

Figure 1.4: Coronary angioplasty procedure (left) and an example of a balloon catheter with a stent for coronary angioplasty (right) (Figure from [133, 135]).


Transposition of the great arteries
Balloon Atrial Septostomy (Rashkind Procedure)
Blood flow
Balloon catheter

Procedure to open a hole in the septal wall dividing the left and right atria. The opening in the septum allows oxygen-rich and oxygen-poor blood to mix, improving circulation.

Catheter is inserted into the septal hole and inflated. After inflation, catheter is pulled back through hole.

Figure 1.5: Balloon septostomy procedure (left) and some examples of the balloon catheter for septostomy (right) (Figure from [132, 148]).

balloon catheter on the bottom right.

An electrophysiology study (EP test or EP study) is a MICS procedure in which the electrical activity of the heart is evaluated. The purpose of EP studies is to assess abnormal heart rhythms and to locate the position of pathways leading to arrhythmias. During this procedure, a single or multiple catheters are placed within the heart through a vein or artery. X-ray imaging is used to visualize the catheter positions. The electrodes are used to measure the electrical activity along the conduction pathways of the inner walls of the heart [227].

Catheter ablation is an interventional procedure in which an abnormal pathway of the heart is located and destroyed by using an ablation catheter. This procedure is used to treat cardiac arrhythmias such as atrial fibrillation (AF). Usually it is the post-procedure of an electrophysiology study and can be integrated with an EP study, for interventional purposes rather than only diagnostic purposes. In this procedure, an ablation catheter is used to advance to the heart through a vein. The ablation catheter tip can be used to destroy the abnormal tissue that is causing the arrhythmia by either local heating or freezing. Fig. 1.6 shows three types of catheters which are commonly used in EP studies and ablations [96].

Fig. 1.7 shows a standard procedure of catheter ablation for atrial fibrillation: (1) The relevant catheters are inserted through a small incision, which is made by a cardiologist, into the blood vessels and later to the heart, guided by live X-ray images; (2) once the catheters, normally the lasso catheter and the coronary sinus catheter, are in place, small electrodes are placed around the area of the heart to measure the electrical activities and to identify the abnormal tissue; (3) after the source of the problem is located, the ablation catheter is used to delivery energy to destroy the abnormal tissue [96].

1.1.4 Common cardiac diseases that can be treated using catheterization

Arteriosclerosis and Atherosclerosis

Arteriosclerosis is an artery disease in which the wall of the abnormal arteries becomes thick, hard and lacks elasticity [193], Atherosclerosis (arteriosclerotic vascular disease, ASVD) is one type of arteriosclerosis that caused by accumulation of fatty plaques, which consist of white blood cells (both dead and active), cholesterol, triglycerides, calcium or other crystallized materials in the artery [61]. Angiography can be used for the diagnosis of both arteriosclerosis and the later stage of atherosclerosis. However, atherosclerosis is a chronic disease and can go undetected without symptoms for decades. Thus early atherosclerosis normally shows no clear symptoms in angiography images. Possible surgery-based treatments for both arteriosclerosis

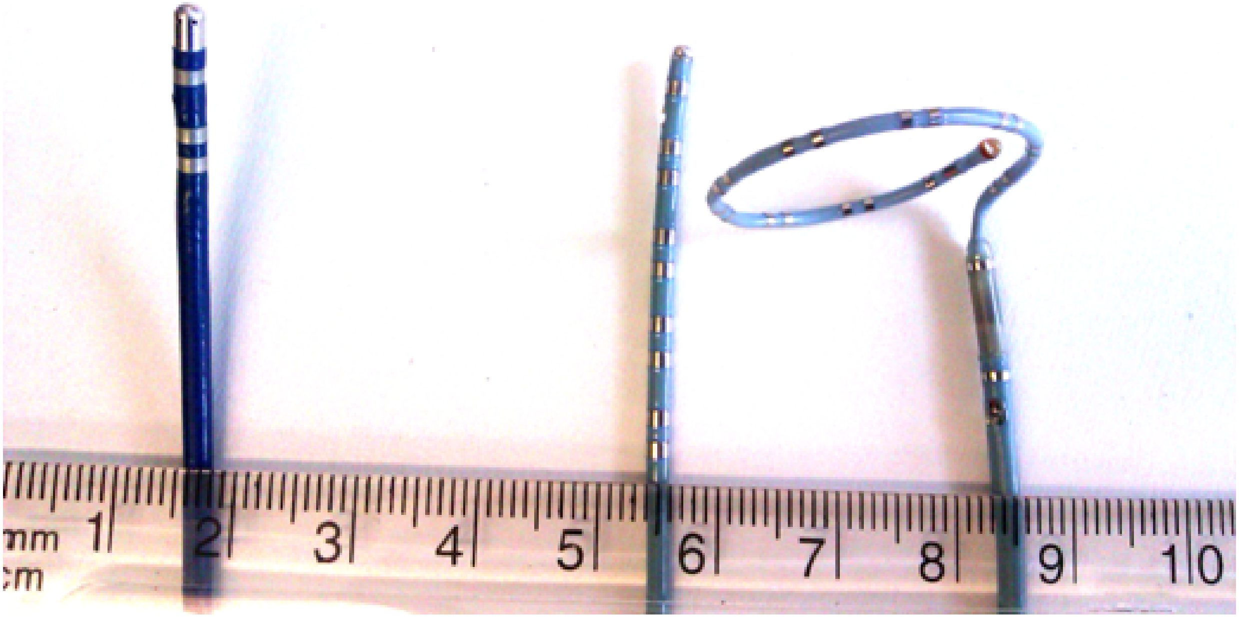


Figure 1.6: Three types of catheters used for EP study and ablation. From the left to the right are an ablation catheter, a coronary sinus catheter and a mapping catheter.

and atherosclerosis include angioplasty (see details in the last section), coronary artery bypass surgery and endarterectomy. Coronary artery bypass surgery (coronary artery bypass graft, CABG) is a cardiac intervention in which other arteries or veins are grafted to the coronary arteries to bypass atherosclerotic narrowings and increase the blood supply to the heart. This can be carried out without stopping the beating of the heart. Endarterectomy is another procedure in which the atheromatous plaque in an artery is removed from the arterial wall directly.

ASD and cyanotic CHD

An atrial septal defect (ASD) is a type of congenital heart defect (CHD) in which the left and the right atria of the heart are not completely isolated. The ASD enables blood flow between the left and right atria. As a result, the oxygen-rich blood from the left atria is mixed with the oxygen-poor blood in the right atria, or vice versa. This may lead to a low oxygen amount in the arteries when it is supplied to other organs and tissues of the body. However, the main shunt of

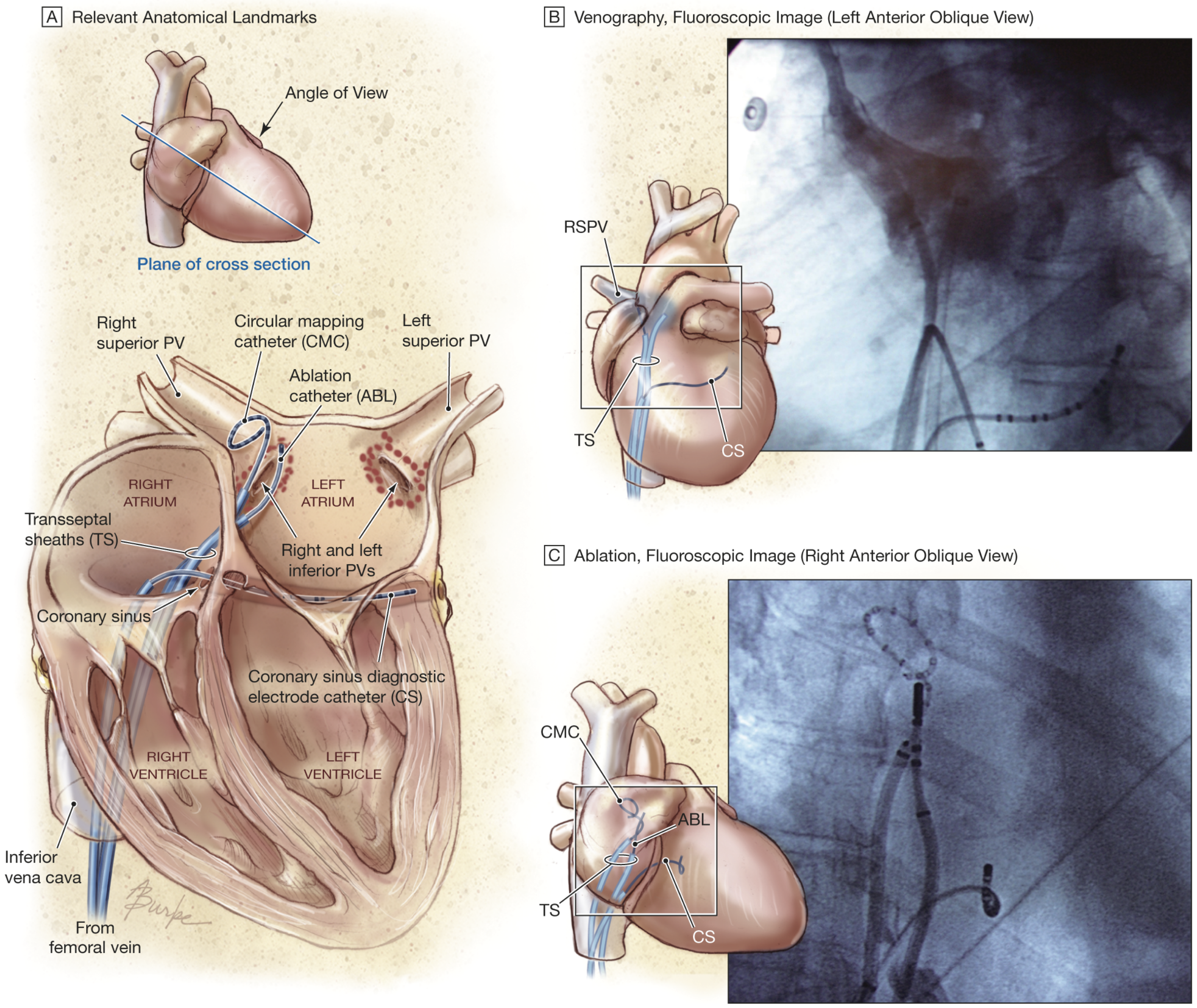

A Relevant Anatomical Landmarks
Angle of View
Plane of cross section
Right superior PV
Circular mapping catheter (CMC)
Left superior PV
Ablation catheter (ABL)
RIGHT ATRIUM
LEFT ATRIUM
Transseptal sheaths (TS)
Right and left inferior PVs
Coronary sinus
Coronary sinus diagnostic electrode catheter (CS)
RIGHT VENTRICLE
LEFT VENTRICLE
Inferior vena cava
From femoral vein
B Venography, Fluoroscopic Image (Left Anterior Oblique View)
RSPV
TS
CS
C Ablation, Fluoroscopic Image (Right Anterior Oblique View)
CMC
ABL
TS
CS

Figure 1.7: Catheter ablation for atrial fibrillation (Figure from [96]).

the blood flow is from the left atria to the right atria. ASD can be treated via catheter-based procedures.

Different from ASD, cyanotic CHD is a type of congenital heart defect in which the oxygen-rich blood in the left atria is mixed with the de-oxygen blood in the right atria or the de-oxygen blood directly bypasses the lung and enters the systemic circulation. One typical cyanotic CHD is dextro-transposition of the great arteries (d-TGA), in which the aorta and the pulmonary artery are transposed. As a result, the de-oxygen blood in the left atria enters the systemic circulation directly and the oxygen-rich blood flows to the lung. To alleviate this case, a hole is made between the left and the right atria to allow the oxygen-rich blood into the left atria and then to enter the systemic circulation as well. Sometimes the hole or an ASD is widened by a balloon catheter to increase the amount of oxygen in the left atria [88].

Atria fibrillation and other cardiac arrhythmias

Cardiac dysrhythmia is a category of cardiovaccular diseases in which the heart beat is irregular: faster (above 100 times per minute) or slower (below 60 times per minute) than normal, caused by an electrical activity disorder. Many types of cardiac dysrhythmia can be treated by catheter ablation [108]. Among them, the most common one is arial fibrillation (AF).

AF is often associated with a fast heart rate and symptoms such as heart palpitations, fatigue or shortness of breath. AF is usually caused by abnormal electrical impulses in the roots of the pulmonary veins. These impulses affect the normal electrical activity in the right atrium and furthermore cause irregular electrical conduction of the ventricular impulses which generate the beating of the heart [160, 97].

Other cardiac dysrhythmias which can be treated by catheterization include supraventricular tachycardia (SVT) and atria flutter (AFL). SVT is also associated with a fast heart rate caused by irregular electrical activity above the ventricles [59]. AFL is an irregular heart rhythm in the atria and may occur spontaneously in normal hearts. AFL is normally associated with a fast heart beat and frequently degenerates into atrial fibrillation [206, 97].

1.1.5 Imaging Modalities

The use of images from endoscopes, X-ray, ultrasound, MR and CT, is playing an increasingly important role in MIS. For cardiac image guided MIS, imaging can be classified into pre-operative medical imaging and intra-operative imaging. Pre-operative imaging, acquired from modelities such as magnetic resonance imaging (MRI) and computed tomography (CT), is used to model and analyze the morphology appearance or motion of the organ of interest prior to interventions for diagnostic purpose. This enables treatment planning or visualization for the subsequent intervention. Intra-operative imaging refers to images acquired during the intervention, such as endoscopic, fluoroscopy and echocardiography images. These images are mainly used for navigation and intra-operative visualization of the surgical scene in real-time. This real-time visualization of surgical scenes in cardiac MIS is the topic of this thesis.

Normally, images from pre-operative and intra-operative imaging techniques contain noise and artifacts. Along with the rapid development of image analysis and computer vision techniques, it is possible to automatically extract, enhance and highlight organs or structures of interest. As a result, computer-aided visualization can help the clinicians to improve the accuracy and speed of observation. In the context of cardiac catheterization, the most useful information is the position of the catheters in relationship to the morphology of the heart. An automatic segmentation and tracking of the catheter in interventional images during catheterization enables a better visualization. This can aid the physician in MIS interventions by highlighting the catheter location in real-time in the images. In the following we will review the most commonly used intra-operative imaging modalities for catheterization and compare their advantages and disadvantages.

X-ray Fluoroscopy

X-radiation (X-rays) is a form of electromagnetic radiation with a wave length between 0.01 to 10 nm, a frequency between 3×10^{16} to 3×10^{19} Hz and an energy between 100 eV to 100 keV [93].

X-ray fluoroscopy is an imaging technique that obtains images of the internal structures of a patient in real-time by using different X-ray absorptions of different internal organs or tissues. An X-ray fluoroscope consists of an X-ray source and an X-ray image intensifier, between which the patient is placed. The X-ray image intensifier converts X-rays into a visible image. Modern X-ray system usually encompasses the X-ray source and the intensifier within a C-arm set-up. Sometimes X-ray systems include two C-arms (with two X-ray sources and two intensifiers) to enable a bi-plane imaging. Fig 1.8 shows an example of such a bi-plane X-ray intensifier. The X-ray energy is absorbed to a larger extend by bone than by soft tissues. As a result the X-ray energy reaching the intensifier through bone is less than through soft tissues. This difference can be visualized by intensifier [93].

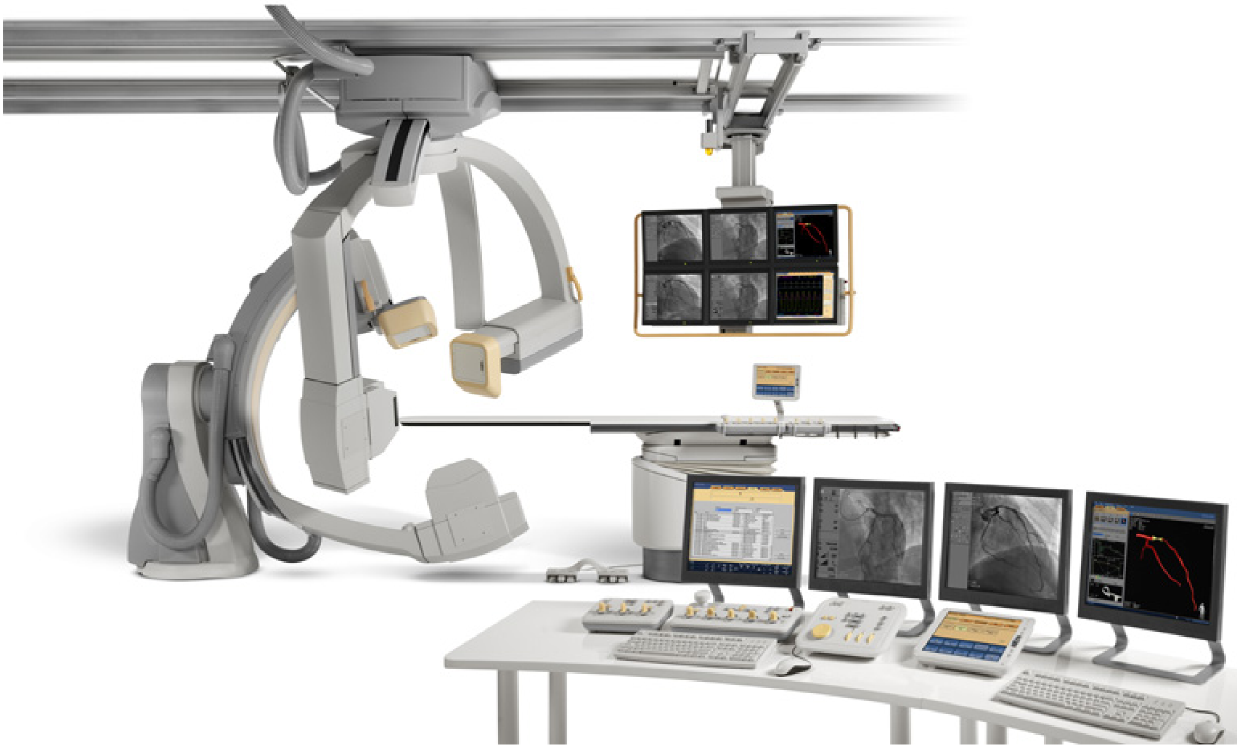


Figure 1.8: An example of bi-plane X-ray intensifier (Allura Xper FD biplane systems developed by Philips) (Figure from [163]).

For most clinical applications in cardiac catheterization, such as catheter ablation for atrial fibrillation (AF), X-ray fluoroscopic imaging is the currently dominant intra-operative modality used during the intervention. Fluoroscopy imaging can be performed at frame rates up to 30 frames per second. Therefore it allows the location of the heart and interventional devices to be visualized in real time. However, 2D X-ray images can not visualize soft tissues in 3D. In addition, the imaging also poses a potential danger to patients, especially children, as a result of the exposure to ionizing radiation.

In a catheterization laboratory where standard catheter ablations are carried out in a hospital, C-arm based X-ray image intensifiers are commonly used to obtain real-time fluoroscopic images during interventions. The C-arm mechanism enables the image intensifier to visualize the target area from arbitrary angles according to clinical requirements. This provides a way to acquire 3D spatial information about the target area despite the fact that only 2D images can be acquired at each point in time. However, 3D imaging comes at a high cost: images from different angles, from which 3D scene can be reconstructed, cannot be acquired simultaneously with only one

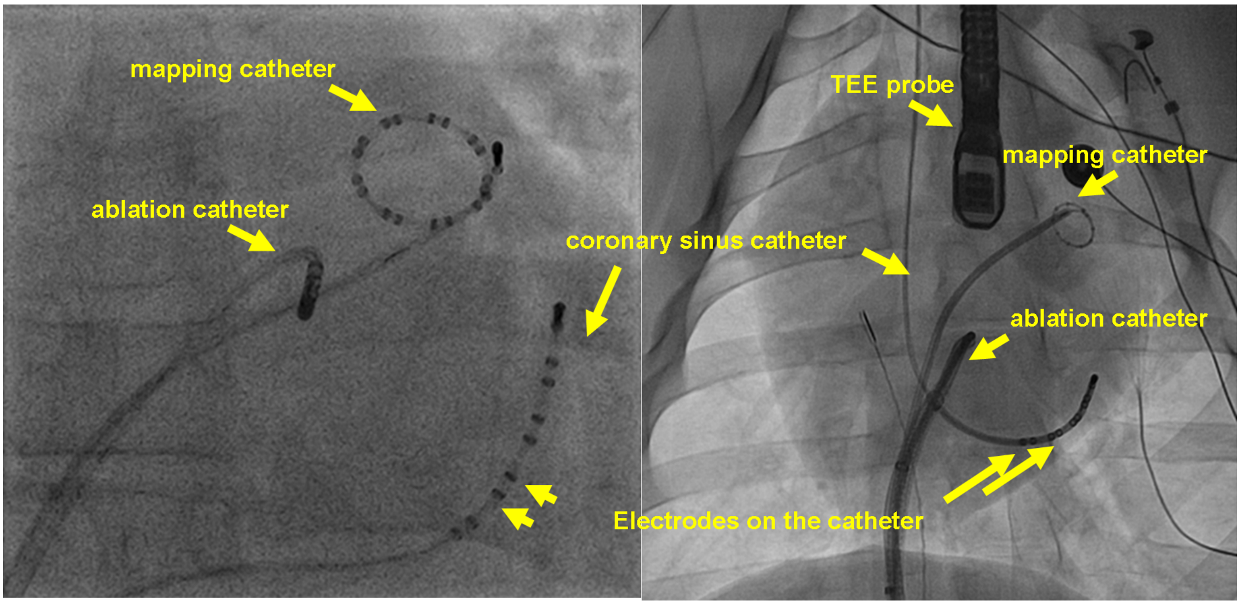


Figure 1.9: X-ray images of catheters for clinical (left) and porcine experimental (right) usages.

X-ray source. One exception are bi-plane X-ray intensifiers which have a pair of X-ray sources (an example is shown in Fig. 1.8). However, this also means that the radiation amount is doubled. Fig. 1.9 shows examples of the catheters in X-ray images. The left image shows a clinical image for a patient with a low dose X-ray exposure while the right shows an image for a porcine experiment with high dose X-ray exposure. It is obvious that the low dose image for clinical real-time use is more noisy than the image obtained with a high dose of X-ray exposure.

Echocardiography

One potential alternative imaging modality is echocardiography, wihch uses standard ultrasound images to create 2D or 3D images of the heart.

Ultrasound are sound waves with a frequency higher than the limit of the human hearing, which is approximately 20kHz. The frequency range for ultrasound in medical imaging usually is between 2 and 18 MHz and the energy exposure is less than 1 watt per square centimetre. An ultrasound wave is produced by a transducer and then travels through the body. The ultrasound impulse is reflected and echoed between the tissue interfaces and the transducer. The time difference and the strength of the reflected sound wave are measured. This information is turned into digital signals for imaging [15].

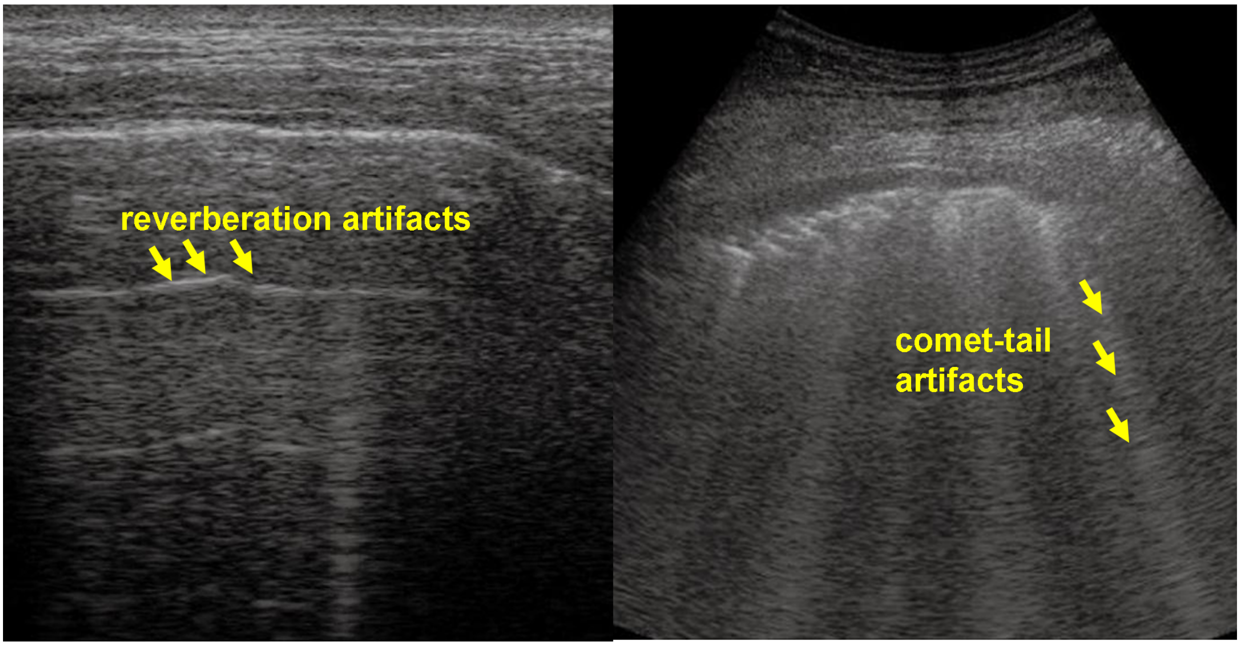


Figure 1.10: Artifacts in ultrasound images (Figure from [173]).

Compared with X-ray imaging, ultrasound imaging has the following advantages: (1) it can provide 3D imaging with 3D spatial information directly; (2) better soft tissue visualization which is required for many clinical cardiac interventions; (3) no harmful radiation. However, acoustic artifacts and a small field of view make the visualization of the interventional scene only by ultrasound extremely challenging [92]. Fig. 1.10 shows two types of common artifacts: (1) reverberation artifacts, which are caused by echoes between internal tissues rather than echoes between tissues, and (2) comet-tail artifacts, which is a special form of reverberation artifacts but usually caused by tiny structures [173].

Transthoracic echocardiography (TTE) uses an echocardiography transducer (or probe) placed on the chest wall (or thorax) to visualize the cardiac anatomy based on ultrasound. This is a non-invasive imaging technique that can be used to assess the morphology and function of the heart quickly and easily. However, because of the placement of the probe outside the body, the accuracy of the imaging can be limited, e.g., due to obesity, chronic obstructive pulmonary or chest-wall deformities between the probe and the heart [209]. It also has a limited ability to visualize cardiac structures at the back of the heart. Transesophageal echocardiography (TEE) is another type of ultrasound imaging in which the echocardiography transducer is inserted into the patient's esophagus. Thus it can provide direct visualization of the cardiac

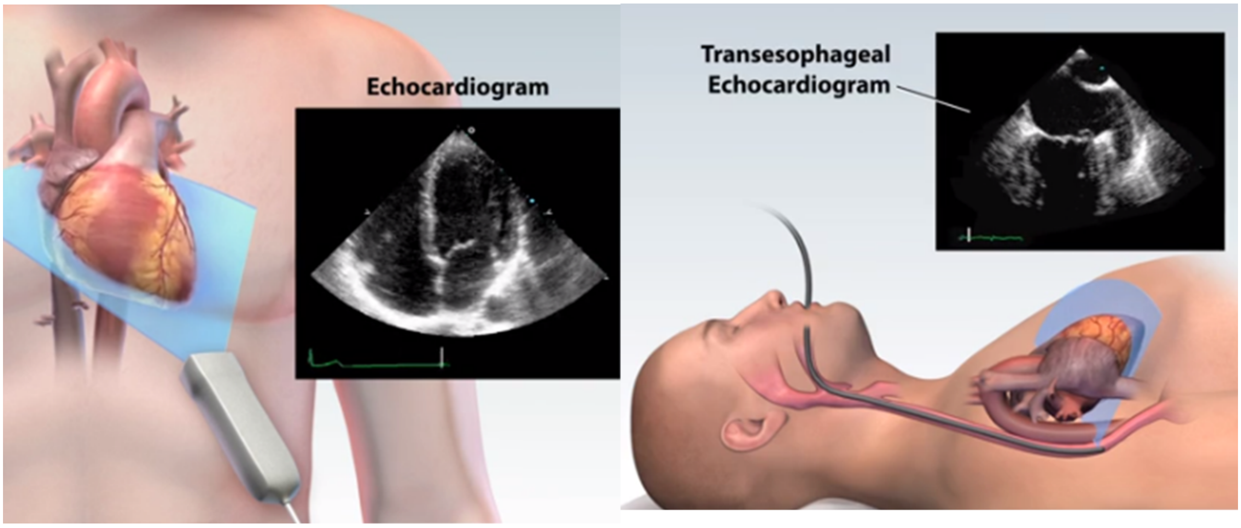


Figure 1.11: An comparison between TTE and TEE procedures (Figure from [136]).

structures and surgical instruments that are inside the heart without imaging through the chest wall. Compared with TTE, it may cause some degree of discomfort for the patient. However, the image quality is generally better than in TTE. The reason is that the ultrasound waves only needs to travel a short distance between the heart and the esophagus in TEE while the ultrasound waves need to traverse skin, fat, ribs and lungs in TTE. This leads to a significant attenuation of the ultrasound waves and thus weaker and noiser signals [209]. Fig 1.11 shows a comparison between the procedures of TTE and TEE.

Generally the visualisation and identification of catheters is particularly difficult in both types of echocardiography, even for experienced physicians, due to the acoustic artifacts, the small field of view and the general difficulty of interpreting ultrasound images. In addition, electrodes on catheters, which can be used to locate catheters in X-ray images, are difficult to identify in ultrasound images. Thus most of existing visualization approaches that are based on identifying the electrodes in X-ray images cannot be easily used in ultrasound imaging. New computer assisted techniques are needed to enhance the visualization of the catheters in ultrasound images.

The research reported in this thesis focuses on the development of novel techniques for improved catheter visualization in X-ray and ultrasound imaging. X-ray fluoroscopy and ultrasound are the main modalities to be used in this thesis. Thus the advantages and disadvantages of

Table 1.2: A comparison between X-ray and ultrasound modalities.

	X-ray	**Ultrasound**
Clinical use	Most catheterization interventions	Limited and only in some diagnostic procedures
View area	Large	Small
Imaging quality	High quality imaging, but noisy with low dose.	Lower image quality due to various acoustic artifacts
3D visualization	2D imaging technique. Possible 3D visualization via reconstruction with bi-plane systems	TEE can provide direct 3D visualization
Soft tissue visualization	Soft tissue invisible	Soft tissue visible
Ionizing radiation	Uses ionizing radiation and poses potential danger to pregnant women and children	Exposure to ultrasound wave is not harmful.

both modalities are compared and discussed in detail here. This comparison is summarized in Table 1.2. Fig. 1.12 also shows images for these two modalities. The entire catheter in X-ray images can be clearly identified while the same catheter can only be visualized partly in the ultrasound images and they are also difficult to identify. The electrodes are invisible in ultrasound images.

Other Modalities

Intravascular ultrasound (IVUS) is another ultrasound imaging technique in which an ultrasound probe is attached to the tip of a specially designed catheter which is inserted into the blood vessels to record in-vivo images of the endothelium of vessels. IVUS has a clear advantage of a better visualization of the details inside the coronary artery lumen. However, the main drawbacks of using IVUS in clinical practice include the high cost of this clinical usage, the high risk caused by the specialized catheter and the long time of the procedure [201]. Fig. 1.13 shows an example of an IVUS catheter (left) and an IVUS lumen image (right).

Another new emerging intravascular imaging technique is optical coherence tomography (OCT), which uses the magnitude and the echo time of the light instead of ultrasound to visualize microstructures like the lumen area of vessels or the anterior segment of the eye and the retina at a

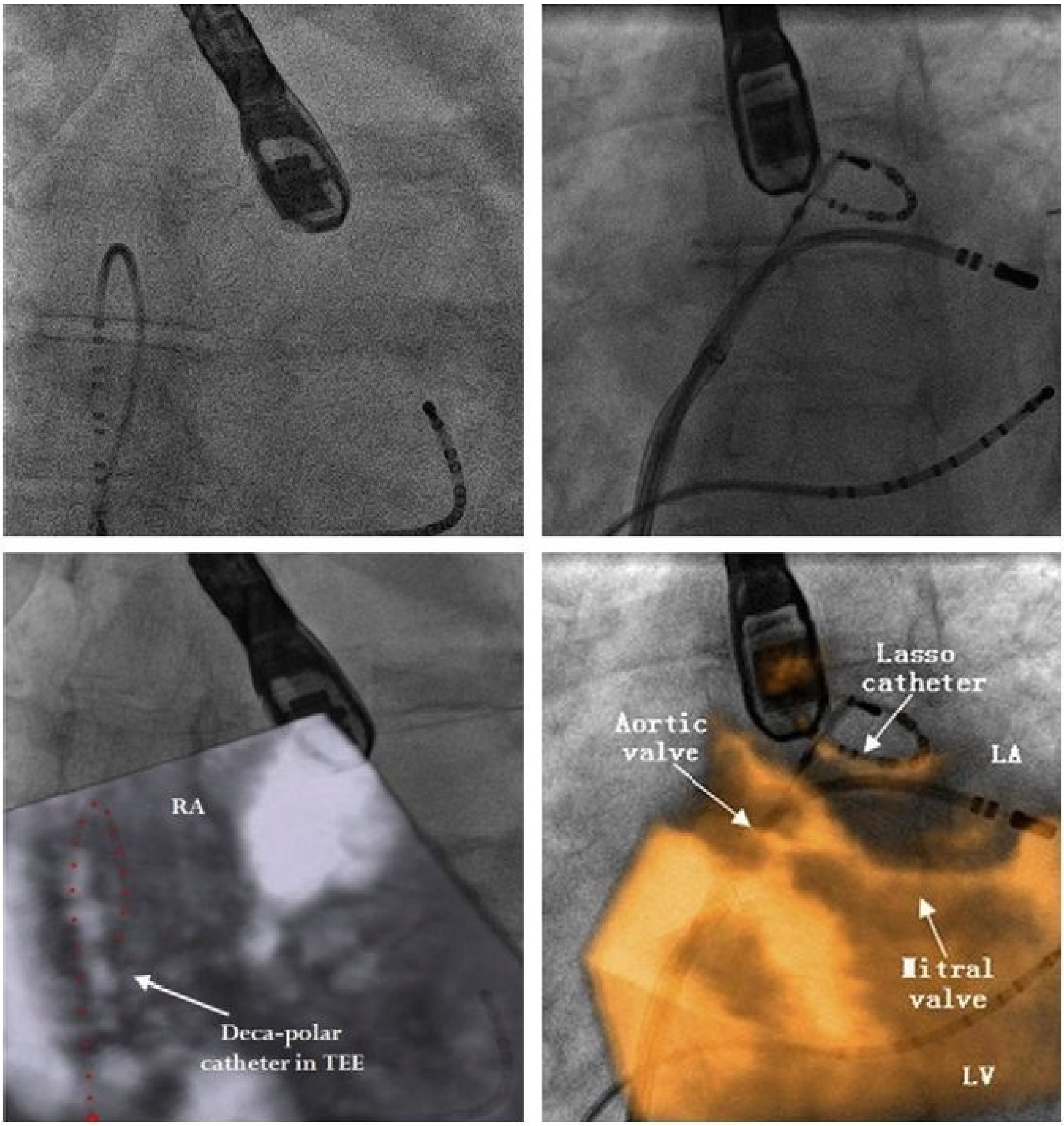


Figure 1.12: Examples of orginial X-ray images (top) with the corresponding ultrasound volume visualized (bottom) (Figure from [208]).

high resolution. OCT has a higher resolution than IVUS, with the tradeoff of a smaller imaging depth. As a result, the field of view for OCT image is relatively small. The resolution and the imaging depth of an OCT system depend on the light frequency. Generally, the limited imaging depth and the view field makes this application only suitable for detailed diagnostic assessment of micro-structure diseases [21]. Fig. 1.14 shows an example of a OCT catheter with its corresponding image. In summary, both OCT and IVUS are limited to imaging of vessels and cannot be used for the cardiac chambers.

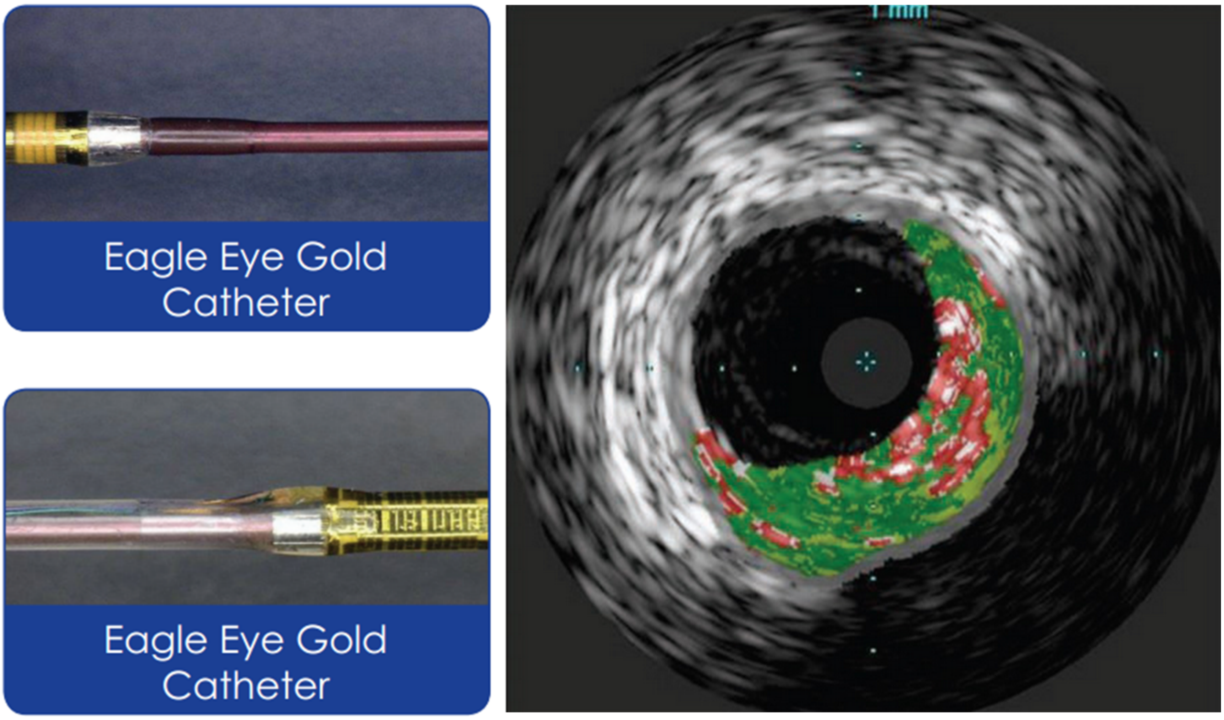


Figure 1.13: An example of two types of IVUS catheters and one of example of the IVUS lumen image showing vessel plaque (Figure from [201]).

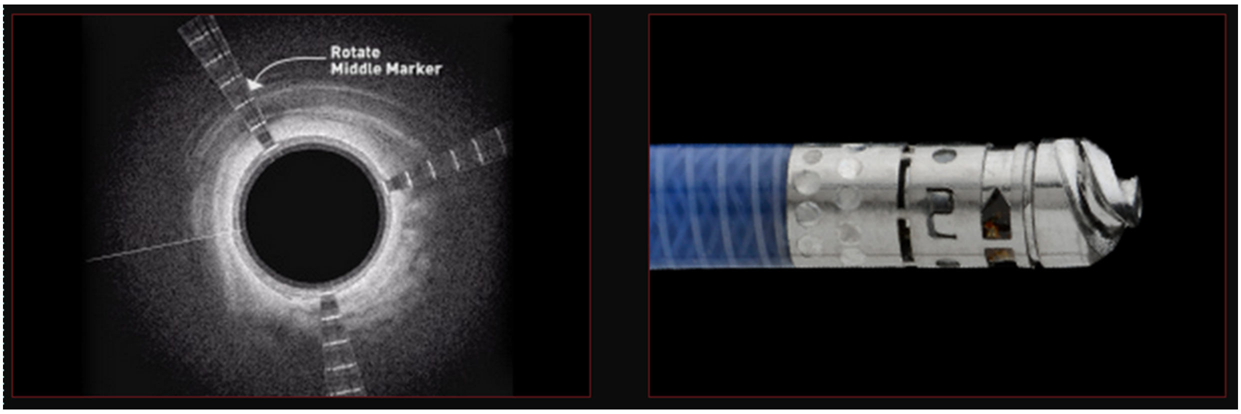


Figure 1.14: An example of an OCT catheter (right) and its corresponding image (left)(Figure from [201]).

Electroanatomic mapping (EAM) is a yet another emerging imaging technique that can localize the position and the orientation of the catheter tip, and construct a road map of the catheter in real-time by recording intracardiac electrical activation related to anatomical locations during a catheter ablation procedure. An EAM procedure is often companied with fluoroscopy imaging before the road map is constructed. However, the use of fluoroscopy is limited after the road map constructed. Thus, one general advantage of EAM is its ability to reduce the usage of fluoroscopy during the intervention. Other advantages of this technology include an increase

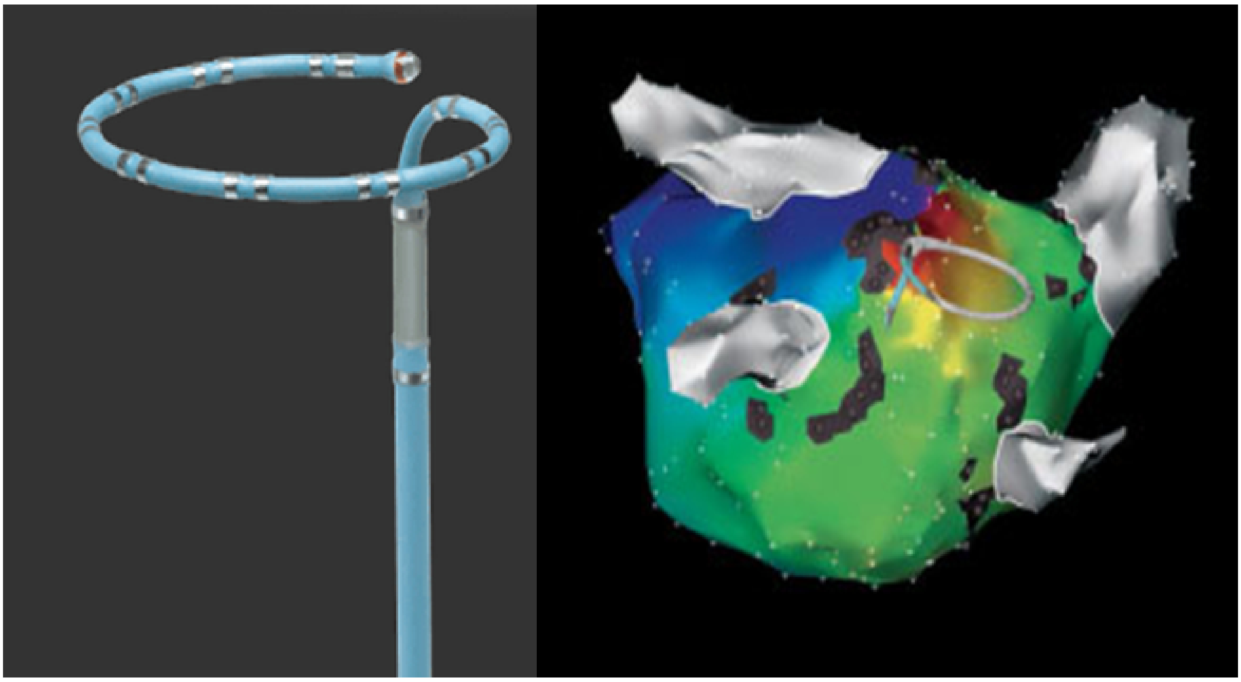


Figure 1.15: An example of the EAM specialized lasso catheter (left) and its imaging under EAM mapping (right) (Figure from [23]).

of the accuracy and easy reacquisition of the position for re-ablation (since the road map has already been reconstructed and it is straightforward to re-locate the ablation target using this road map). However, the spread of this new technology is limited by its cost and the strong dependence on specialized hardware platforms. Most of the EAM systems required specially designed catheters rather than normal ones [22]. Fig. 1.15 shows an example of an EAM system [23].

Computed tomography (CT) is a imaging technique that uses X-ray to reconstruct a 3D volume data. This is enabled by rotating a pair of an X-ray source and an intensifier around the target to form a sequence of images from various angles. Then these images are integrated into a 3D volume using a reconstruction algorithm. Cone beam computed tomography (CBCT) is a medical imaging technique that uses cone-beam shape X-rays to construct a computed tomography instead of the fan-beam used by the conventional CT [175]. The cone-shape beam enables a 3D visualization at one rotation while the conventional CT can only obtain a slice at one rotation. CBCT is most commonly used for diagnosis and can be used in stent placement interventions [20]. Besides the capability of 3D imaging, it shares most of the disadvantages of X-ray fluoroscopy such as ionizing radiation or the lack of contrast of soft tissue. In addition,

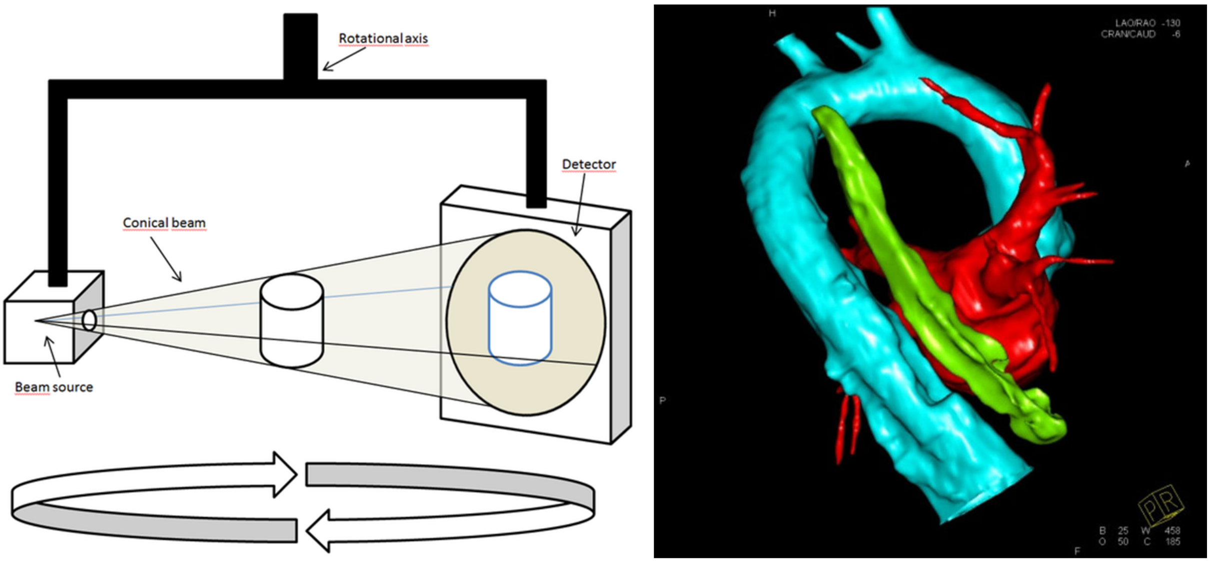


Figure 1.16: The imaging principle of CBCT (left) and an example of 3D image (right) (Figure from [211, 184]).

since the time of acquisition is relatively long, the fast motion of the heart will have a negative influence on the reconstruction accuracy. This drawback limits the application to real-time interventions, especially with large motions [156]. Fig. 1.16 shows the imaging principle (left) and a 3D reconstructed volume by CBCT (right). In this volume the reconstructed left atrium, aorta and esophagus is marked in red, blue and green, respectively.

Interventional magnetic resonance imaging (IMRI) is a Magnetic Resonance (MR) imaging technique used in interventional radiology procedures. The obvious advantage of IMRI over X-ray based imaging is its lack of ionizing radiation for the patient and the operator, and the advantage over ultrasound is its high quality images. However, the strong magnetic fields of the MR scanner require surgical tools that are specially designed. This limits the IMRI to be widely used in catheterization procedures [161].

1.2 Challenges

As X-ray fluoroscopy and ultrasound imaging are used in most of cardiac interventions, the detection and tracking of catheters for cardiac catheterization are mainly based on these two modalities.

For X-ray images, although catheters are visualized with a high quality, the detection and tracking of catheters are still challenging. A catheter being tracked is a non-rigid object. Unlike a rigid object, a non-rigid object needs more parameters to model the tracking. In addition, the catheter motions are complicated, with a combination of motions along both transversal and longitudinal direction. This means that the search space of each parameter is also large. Moreover, normally more than one catheter is in the view and multiple catheters can interfere with each other. The large amount of noise in the low dose images for clinical use also has negative consequences. All of the above factors make the problem challenging. Finally, catheterization is a real-time application that imposes strong constraints on the speed of the tracking algorithm.

Therefore in order to enable soft tissue visualization, the addition of ultrasound imaging would be beneficial. For ultrasound, the segmentation and tracking is even more challenging since the imaging quality is worse than in X-ray images. Since 3D TEE ultrasound is used, the search space is larger than 2D imaging modalities such as X-ray fluoroscopy. In addition, a great number of acoustic artifacts exist in this large search space. Thus, direct catheter detection in ultrasound is not feasible. In order to make the task easier, constraints should be introduced to reduce the search space and eliminate the interference of the artifacts. This can be done by combining the corresponding X-ray and ultrasound images for the tracking. A framework is required to integrate these two modalities and this framework must be fast and tolerant to registration errors.

Another challenge is to avoid X-ray exposure as much as possible to a large extent. This requires the catheter extraction to be performed in ultrasound images without the aid of X-ray images. In this case, the only available prior information is the location of the catheter in the last frame. When the temporal consistency is used, the catheter segmentation problem is transformed into a catheter tracking problem. Compared with X-ray imaging, the frame rate in US images is lower and the view of field is smaller. This means the motion of the catheters is larger and more unconstrained. The complexity of the catheter motion, as well as the low ultrasound imaging quality and the severe acoustic artifacts, aggravate the tracking problem, especially when taking constraints on computational cost into account.

1.3 Contributions

The main contributions of this thesis include the following:

- A novel and fast catheter segmentation and tracking strategy in X-ray images is proposed. There are a large number of studies addressing the problem of catheter segmentation and tracking in X-ray images. Most of them focus on catheter electrode detection instead of detecting the entire catheter. Some of the approaches address the entire catheter tracking via discrete optimization. However, the complex of catheter motion pattern and the mutual interference between multiple catheters are usually not fully taken into account. In our research, we propose a strategy to deal with a more complicated motion involving large longitudinal movements. The novel features of our tracking strategy include a fast path growing based catheter extraction, adaptability to large longitudinal movements for individual catheter tracking and the capability to address the problem of mutual interference from multiple catheters during tracking. A combination of edge and ridge detectors is used within image patches to estimate the catheter orientation and the center of local catheter fragments. These features can resolve the intersection of multiple catheters in X-ray images. A Kalman filter is used to extract the entire catheter by incorporating these features.

- The second contribution of the thesis is a catheter extraction system for ultrasound images assisted by the corresponding X-ray images. Based on previous work on 2D-3D registration of X-ray and ultrasound images, our system can successfully bridge the gap between these two modalities. This strategy enables the ultrasound imaging to be combined with X-ray images during the intervention. A better visualization of soft tissues is possible through the use of ultrasound images. Based on the registration of ultrasound and X-ray images, the segmentation or tracking results from the X-ray images are employed to reduce the search range for catheter location in the ultrasound images. Searching for the catheter in this reduced space means that severe acoustic outliers outside the search range can be eliminated. This also accelerates the catheter extraction algorithm, compared with the extraction without X-ray images. In this reduced search space features can be detected. A graphical model is defined to organize these features. The optimal path in this graphical model can be identified to fit the catheter curve in ultrasound images. To the best of our knowledge, this work is the first to address the problem of the segmentation of the entire catheter from ultrasound images for cardiac catheterization procedures.

- We also proposed a catheter tracking strategy using only ultrasound images. This strategy significantly reduces the dependence on X-ray imaging and thus the risks from ionizing radiation. Without X-ray images, the only clue help to reduce the search range is the location of the catheter in the last frame. We refer to this type of prior information as temporal consistency. Temporal consistency is used to identify the outliers that are obviously inconsistent with catheter extraction results from previous frames. However, even if temporal consistency is taken into account, the complicated catheter motion involving both transversal and longitudinal motion is still a challenging problem. The tracking methodology we have developed for 2D X-ray images cannot be simply extended to 3D ultrasound images because the computational cost would increase dramatically. In order to address this problem, the entire tracking process is decoupled into two stages: transversal tracking and longitudinal tracking. The strategy guarantees the tracking along both longitudinal and transversal directions with high speed. In addition, the motion model of the catheter is designed as a periodic motion model since it can approximate the periodic beating of the heart. The parameters of this motion model are being tracked online by using a Kalman filter. The motion model is estimated based on the information from previous frames. The estimated motion model can be used to predict the location of the catheter in the next frame more accurately. A more precise prediction of the catheter location in the next frame can reduce the likelihood of a tracking failure as a result of large motion. Furthermore it can reduce the amount of X-ray exposure since a tracking failure normally requires additional X-ray images to re-initialize the tracking. The proposed system is implemented using a GPU platform. To the best of our knowledge, this system is the first catheter tracking system for 3D ultrasound images which can satisfy the real-time requirements (above 10 fps).

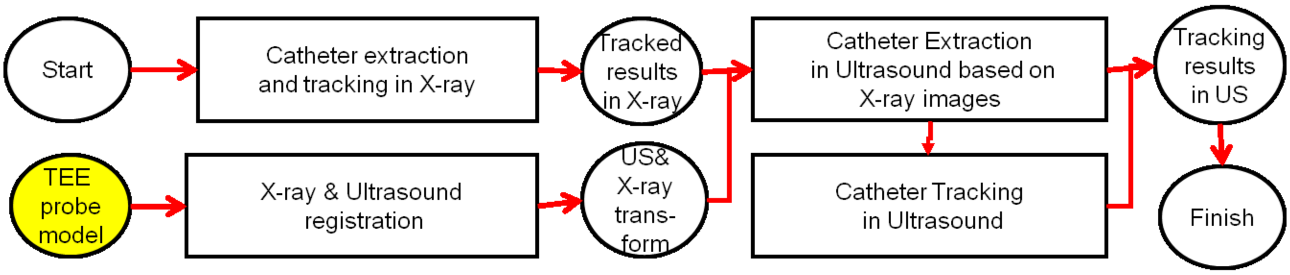

Start
Catheter extraction and tracking in X-ray
Tracked results in X-ray
Catheter Extraction in Ultrasound based on X-ray images
Tracking results in US
TEE probe model
X-ray & Ultrasound registration
US& X-ray trans- form
Catheter Tracking in Ultrasound
Finish

Figure 1.17: A system integrating all contributions. Yellow indicates processing that can take place off-line.

All of the proposed contributions can be integrated as a complete system as shown in Fig. 1.17. The online procedure starts with the catheter extraction and tracking in X-ray images. Independently, the off-line constructed probe model can be used to calculate the transformation between the corresponding X-ray images and ultrasound images by registration. Then the result of the registration, as well as from the catheter tracking in the X-ray images, can be used to extract the catheter in the corresponding ultrasound volume. Finally, the extracted result from one ultrasound frame can be used to initialize the tracking for the following ultrasound frames without the need for X-ray images. The relationship between all proposed algorithms will be summarized in the conclusion chapter.

1.4 Thesis Outline

In the following chapter, we will review the background and the state-of-the-art that is related to our research. The third chapter will introduce the catheter segmentation and tracking for X-ray images. The catheter segmentation algorithm can extract the entire catheter from each frame independently. It does not require the information from previous frames. It can be used to initialize a tracking system or to recover a tracking failure. A tracking algorithm is also developed to use the information from previous frames. From our evaluations, the extraction algorithm can preserve the entirety of the catheter. It is robust to more complicated motions. The tracking algorithm can address multiple catheters cases, including catheter overlapping and mutual interferences. This is important since cases of multiple catheter cases are quite common in clinical interventions. The fourth chapter will introduce the catheter segmentation in ultrasound images based on the corresponding X-ray images. In this chapter the X-ray imaging and ultrasound imaging are related by a 3D-to-2D registration algorithm. Then the catheter extracted from X-ray images can be projected to the corresponding ultrasound images. This information enforces a constraint on the search space for catheters in the ultrasound images. With this catheters are segmented from this constrained space. We proposed graph model-based algorithms to extract the catheter in the ultrasound images. Our system is tolerant to the registration error between X-ray images and ultrasound images. This framework is validated based on both phantom and clinical data. The fifth chapter focuses on the real-time catheter tracking only in ultrasound images. X-ray imaging is not required for each single frame and the X-ray exposure can be maximally reduced. Approaches addressing the tracking of catheter in 2D X-ray images are not applicable to 3D ultrasound volume, mainly due to the increase in computational complexity. Thus we proposed a strategy which uses a reduced search space to search for possible locations of the catheter. Our experiments based on both porcine and clinical data indicate that our system can address more complicated motion and simultaneously preserve the entirety of the catheter being tracked. With the GPU platform, the entire system can meet the real-time requirement (above 10 fps). The last chapter will give a brief conclusion, including a summary of the contributions, the potential problems and possible future work.

Chapter 2

Background

This chapter reviews the related work and the state-of-the-art in the field of segmentation and tracking. We start with a general investigation of computer vision techniques relevant to our application. In the second part of the chapter we focus on techniques for catheter segmentation and tracking, as well as a brief introduction of hardware accelerators.

2.1 Feature detection for segmentation and tracking

As the first step for both segmentation and tracking, feature detection is an important step since it can directly affect the performance of the entire system. Good features can significantly enhance the robustness and furthermore improve the accuracy. In some cases, features, such as corners, blobs, tubular structures, are clearly visible and they can be detected by designing corresponding detectors. However, in other cases, features are less clearly visible. In many cases, a feature is in form of a high-dimensional vector. This vector-based feature usually encodes the information from the neighbourhood of a pixel or voxel. Then sufficient data are required to learn the nature of the feature. Supervised learning can be used to determine the decision hyper-plane or probabilistic distribution for either segmentation or tracking applications. In some cases, unsupervised learning can also be used to transform the original feature into a more useful feature for segmentation or tracking.

2.1.1 General Feature Detection

Detectors dealing with basic geometric features such as blobs, corners, edges or tubular structures in an image normally produce scalar-valued measurements characterizing the strength of the feature. Then the corresponding features are detected by thresholding the output of the detector. Usually such scalar-valued detectors are expected to be fast, robust and accurate. For speed, detectors are usually designed with low computational cost such as using integral images [16]. Furthermore, if a large number of pixels in a small neighbourhood have a detector value above the threshold, then only one pixel (with the maximal value) is selected to represent. This is referred as non-maximum suppression [121]. For robustness, detectors are expected to be invariant to the changes of illumination, rotation, scale and affine transforms. This can be achieved by feature normalization, principle orientation, scale space [121] and shape adaption [138], respectively. In addition, interpolation can be carried out to allow feature detection with sub-pixel accuracy [121].

Corners

The Harris corner detector [82] is a classic feature extraction method but one of the most useful methods to extract stable corner points from images. The Harris corner detector is based on the eigenvalues of a structure tensor matrix H_{harris} in form of:

$$H_{harris} = \sum_{u,v \in N(x,y)} \omega(u,v) \begin{pmatrix} I_x(u,v)^2 & I_x(u,v)I_y(u,v) \\ I_x(u,v)I_y(u,v) & I_y(u,v)^2 \end{pmatrix} \tag{2.1}$$

Here $I_x(u,v)$ and $I_y(u,v)$ are the image intensity derivatives along x and y axis on pixel (u,v). $N(x,y)$ denotes a neighbouring patch of (x,y). $\omega(u,v)$ denotes the weights (normally using a Gaussian Kernel) for averaging within a small patch. A corner is identified if both two eigenvalues of the corresponding matrix have a large positive value. The Harris corner detector has been widely used in the field of computer vision and robotics due to its stability, straightforward and fast calculation, as well as its high accuracy. Alternative corner detectors which have a similar performance to the Harris corner detector have been reported in several studies [181, 185, 168].

Blobs

In addition to corners, blobs are another type of salient points commonly used. Most of the blob detectors are based on the Laplacian of Gaussians (LoG) [116], the difference of Gaussians (DoG) [117, 121], or the determinant of the Hessian matrix (DoH) [115, 16]. Given an input image $I(u, v)$ and a Gaussian kernel $G(u, v, s)$ at the scale of s, the result of the convolution $L(u, v, s)$ is defined as:

$$L(u, v, s) = G(u, v, s) * I(u, v) \tag{2.2}$$

Assume the second partial derivatives of $L(u, v, s)$ along x and y are denoted by $L_{xx}(u, v, s)$, $L_{xy}(u, v, s)$ and $L_{yy}(u, v, s)$, then the Laplacian of Gaussian (LoG) is defined as

$$\bigtriangledown^2 L(u, v, s) = L_{xx}(u, v, s) + L_{yy}(u, v, s) \tag{2.3}$$

The value of the LoG can be used to detect blobs.

The difference of Gaussian (DoG) uses the difference of two Gaussian convolution values at different scales to approximate the value of the LoG. One assumption of this approximation is that $L(u, v, s)$ satisfies the diffusion equation:

$$L_s(u, v, s) = 0.5 \bigtriangledown^2 L(u, v, s) \tag{2.4}$$

Here $L_s(u, v, s)$ is the derivative of $L(u, v, s)$ along the scale change s and can be approximated by the difference of $L(u, v, s)$ at various scale s.

In addition to LoG and DoG, descriptors based on the Hessian matrix can be used to detect blobs. The Hessian matrix is the matrix of second derivative and can be written as:

$$H_{hessian} = \begin{pmatrix} L_{xx}(u, v, s) & L_{xy}(u, v, s) \\ L_{xy}(u, v, s) & L_{yy}(u, v, s) \end{pmatrix} \tag{2.5}$$

The determinant and the trace of the Hessian matrix are often used as a detector. The Hessian matrix is also used in Speeded-Up Robust Features (SURF) detector [16]. One advantage of

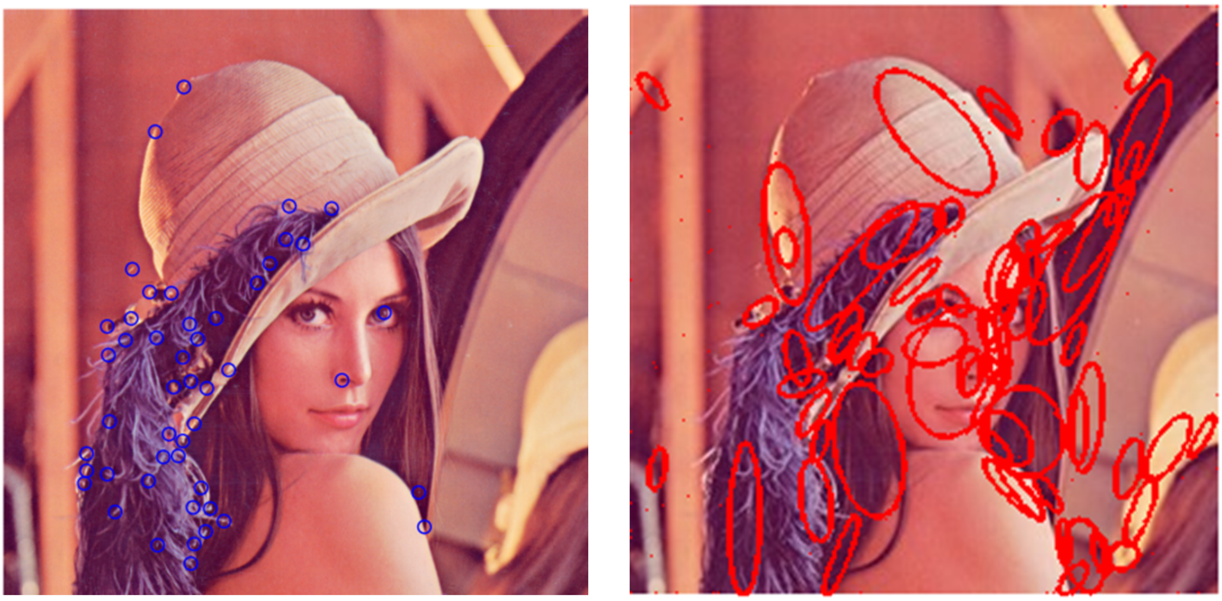


Figure 2.1: A example showing Harris corners on the left and DoG blobs.

SURF is that it calculates the elements of the Hessian matrix based on an integral image. The integral image can be used to approximate the DoG value in an efficient way [16]. This strategy significantly speeds up the detection process and is suitable for real-time applications. Fig 2.1 shows an example of Harris corners (Left) and DoG blobs (Right).

In order for salient point detectors to detect corners or blobs, a basic requirement is that the detector should be invariant to changes of orientation, scale and affine transforms. In [16, 121], orientation invariance is guaranteed by calculating the principle orientation of the local patch and scale invariance is achieved by constructing a scale space pyramid. Affine invariance is generally obtained by iteratively calculating a second-moment matrix to realize shape adaptation [138, 139, 118, 141, 199].

Edges

Edges are another category of basic features being widely used. The edge detection problem has been well researched. Most approaches use first-order derivatives of the intensity for edge detection [36, 60]. Assume that the image gradient along x and y are denoted by $I_x(u, v)$ and

$I_y(u,v)$ at (u,v), then the following responses are calculated:

$$I_{mag}(u,v) = \sqrt{I_x(u,v)^2 + I_y(u,v)^2} \tag{2.6}$$

$$\varphi(u,v) = atan2(I_x(u,v), I_y(u,v)) \tag{2.7}$$

Here $I_{mag}(u,v)$, the gradient magnitude, can be used to threshold edges. $\varphi(u,v)$ is the direction in the direction normal to the edge.

Tubular Structures

Most detectors for tubular structures are used in applications such as road segmentation from 2D aerial images [187, 112] or vessel segmentation from 3D medical images [69, 174, 109, 102, 113]. Compared with edge detectors, most tubular structure detectors are based on second order derivatives, which form a Hessian matrix as shown in eq. 2.5. A 2D tubular structure usually corresponds to a 2D Hessian matrix that has a principle eigenvalue with a large absolute value and a second eigenvalue close to zero. For 3D tubular structures, two eigenvalues have a large absolute value, and the third eigenvalue is close to zero. We assume that the eigenvalues in 2D(3D) are ordered as follows: $|\lambda_1| < |\lambda_2|(< |\lambda_3|)$. Both blobs and tubular structures can be identified via the eigen-system of the Hessian matrix. Table. 2.1 summarizes the geometric patterns and the corresponding eigenvalues. Fig 2.2 shows an example of edge detection and tubular structure detection in an X-ray angiogram of the heart.

For parametric models such as lines, circles, ellipses, and curves, the detection problem in the original image space can be transformed into a detection problem in parameter space, which is also called the Hough space. Each set of the parameters are weighted proportionally to the number of supporting pixels in the original image. Then the detected shape, which is modelled by the parameters with a large weight, is extracted [57, 83]. However, the computational cost of the detection via Hough transform is high if the shape being detected is modelled with a large number of parameters. One typical application for the Hough transform is the line detection. Since lines can be modelled with only two parameters, the line detection based on the Hough

Table 2.1: Geometric patterns and the corresponding eigenvalues $|\lambda_1| < |\lambda_2| (< |\lambda_3|)$ (Table from [69])

2D		3D			Patterns
λ_1	λ_2	λ_1	λ_2	λ_3	
		L	L	H^{-}[a]	plate-like structure (bright)
		L	L	H^{+}	plate-like structure (dark)
L	H^{-}	L	H^{-}	H^{-}	tubular structure (bright)
L	H^{+}	L	H^{+}	H^{+}	tubular structure (dark)
H^{-}	H^{-}	H^{-}	H^{-}	H^{-}	blob-like structure (bright)
H^{+}	H^{+}	H^{+}	H^{+}	H^{+}	blob-like structure (dark)

[a] $H^{+\backslash -}$ means a high absolute value with a positive\negative sign. L means a low absolute value close to zero.

transform is quite efficient and has been widely used in many applications [3, 130, 66]. Fig. 2.3 shows an example of line detection through its 2D Hough space. The left part of the figure shows two lines in the original space and the right part of the figure shows the voting result in the Hough space modelled by two parameters of each line (the distance to the origin and angle between the line and the X-axis).

2.1.2 Feature Detection using Machine Learning

The general feature detection framework described previously can be only used for simple features or shapes. These features are parametric features that can be identified based on prior knowledge. Usually the detection or classification of such features is performed by thresholding the output of the corresponding detector. However, more complex sets of features are difficult to represent and the detection of these features is more complex. In addition, it is difficult to explicitly represent the prior knowledge necessary to encode these more complex features. In these cases, an alternative approach is to learn either an more efficient representation of the feature or a probabilistic distribution in the feature space.

In many cases features represent vectors in a high-dimensional space. This vector usually encodes the local information within the neighbourhood of the point of interest. One commonly used feature in computer vision is the intensities within a local patch directly, usually normalized [68], of either the original image or the gradient image. Other features (also referred as

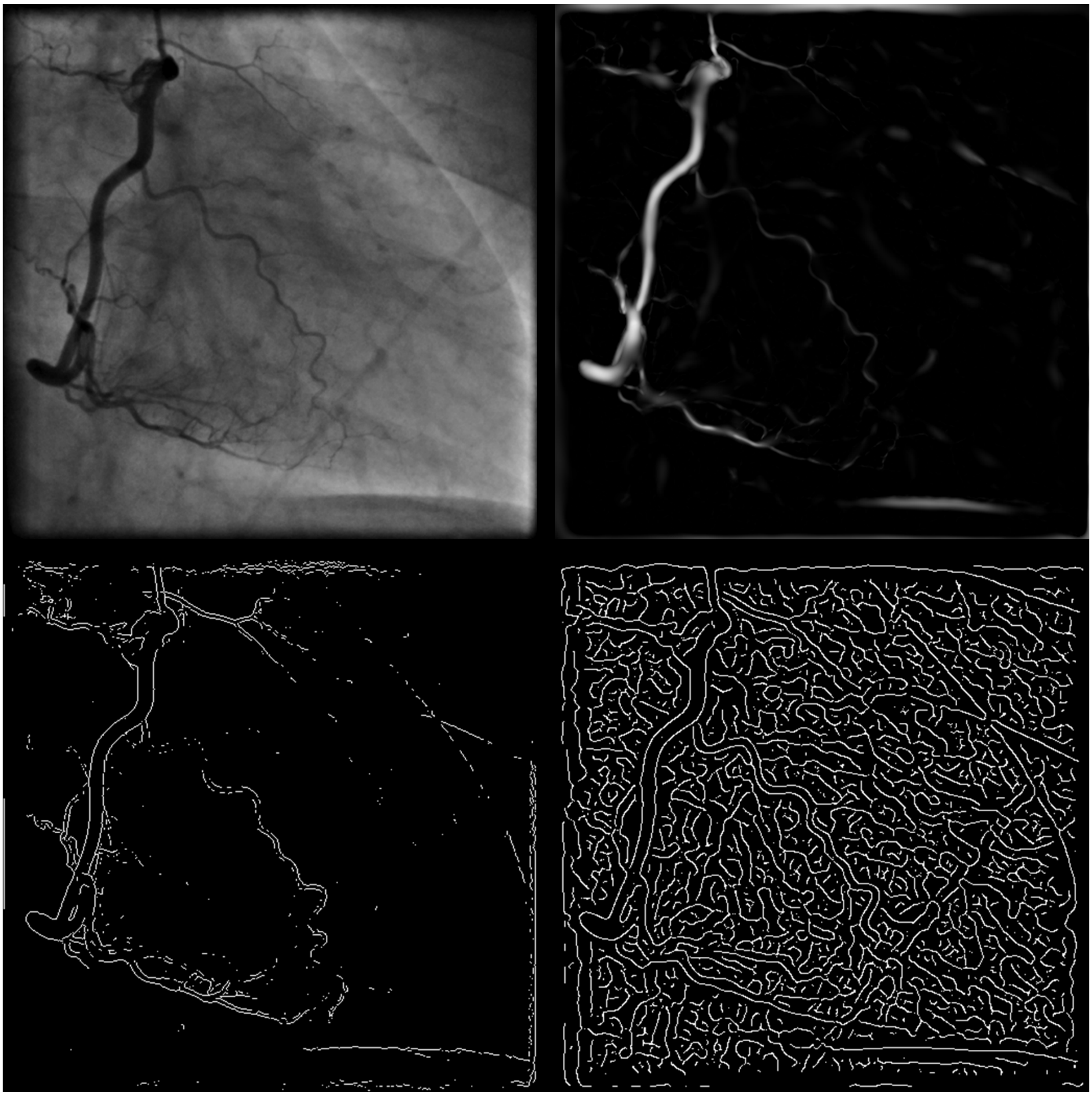


Figure 2.2: An example of edge detection with Sobel (bottom left) and Canny (bottom right) operators, and tubular structure detection (top right), compared with the original image (top left).

descriptors) [121, 16, 140, 53] organize the local information in a more efficient way. We give two examples of the most popular descriptors in Fig. 2.4 and Fig. 2.5: The scale-invariant feature transform (SIFT) [121] defines a descriptor based on the statistics of gradients. A local patch near the key point is aligned with an primal orientation calculated previously. This patch is divided into small cells (4×4 in the figure). For each cell, it consists of a number of pixels. The gradient directions for these pixels are used to construct an 8-bin histogram for this cell. Then the 4×4 gradient histograms for all cells in the local patch are concatenated to form a 128D

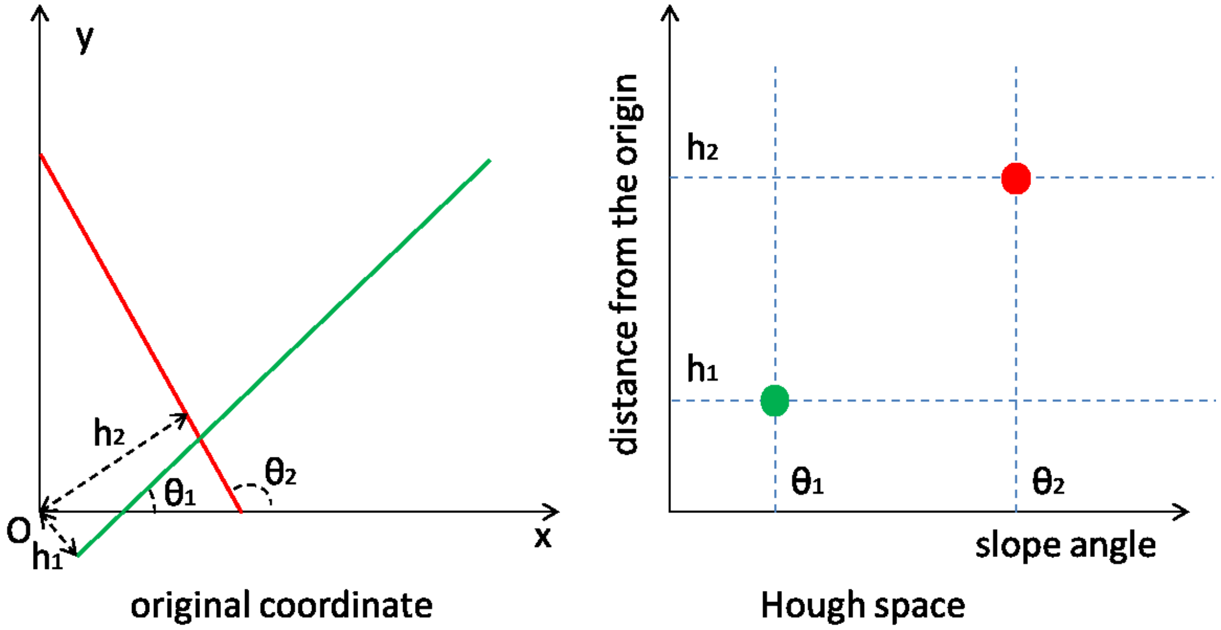


Figure 2.3: An example of Hough line detection with two lines in the original coordinate (left) and the corresponding voting in the Hough space.

vector. Fig. 2.4 shows an example of SIFT descriptor. The left shows the patch consisting of 4×4 cells, with the centre (the yellow cross) on the key point and x-axis aligned with the primal orientation calculated previously (the yellow arrow), the black arrows in each cell indicate the strength of each bin corresponding to the 8-bin histogram. The right sub-figure shows an example of a cell of which the histogram is generated based on the local gradients denoted by red arrows. The Histogram of Gradients (HoG) [53] gives a more general framework to construct a feature descriptor based on gradients. A full gradient image is divided into small cells and a histogram based on gradient orientations is constructed. Then a block, which consists of a number of cells, is defined. The feature of this block is constructed by concatenating all histograms corresponding to the cells belonging to this block. In contrast to SIFT, which can only describe the local neighbourhood of a key point, HoG features can be used to describe a whole image by concatenating the features of all blocks in this image. Fig 2.5 shows a basic approach for the calculation of HoG. The left sub-figure shows cells with the corresponding orientation distribution, which is similar to SIFT descriptors. The right sub-figure shows the histograms corresponding to all cells within a local block. All this histogram can be concatenated to form a feature describing this block.

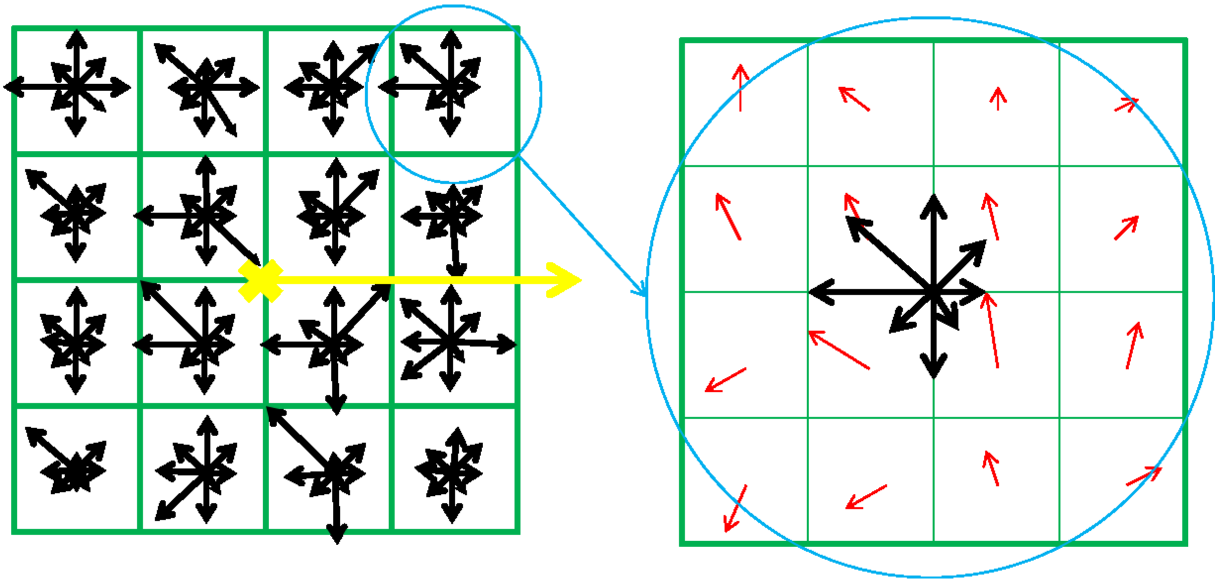


Figure 2.4: SIFT descriptor calculation with cells (left) and an example of one cell calculated based on local gradients (right).

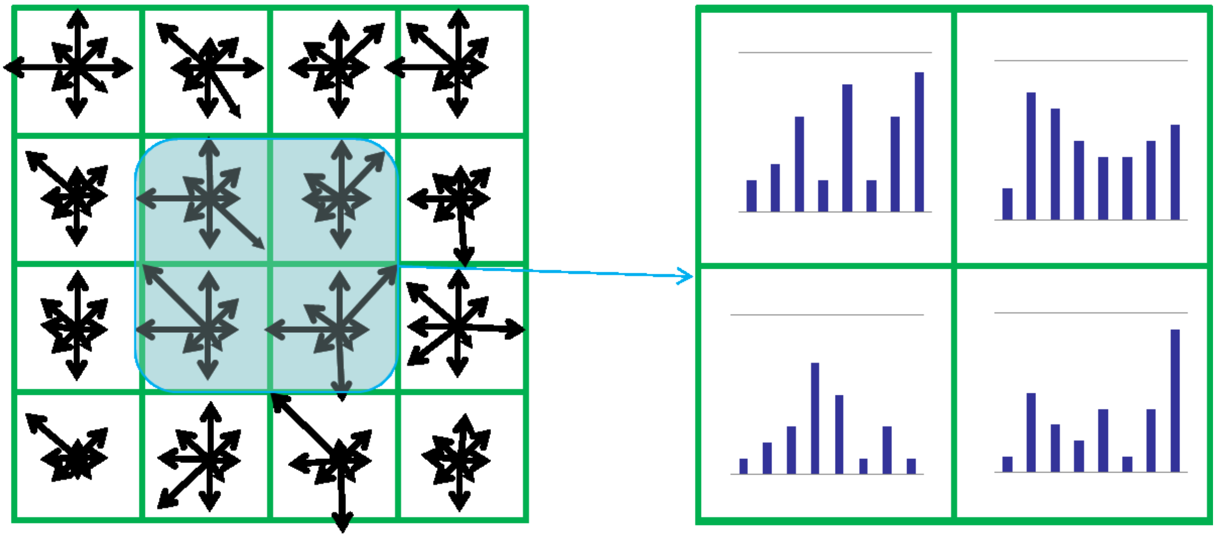


Figure 2.5: HoG descriptor calculation with cells (left) and the histograms based on the gradient statistics (right).

Given a set of vector-based measurements, which are called features in this context, a discrete (classification) or continuous (regression) function is learned based on a set of available training data. This function takes the features as inputs and the output of this function, normally a scalar value, indicates the class label or a regression value. For example the class labels could be ’no feature detected’ or ’feature detected’. The learning process can be carried out in either a supervised or unsupervised way depending on whether the labels of the training data are used during the learning.

In most of the classification or regression cases, some available training data are manually annotated with labels and a supervised learning is performed. Commonly used supervised learning methods include support vector machines (SVMs) [189, 40], decision trees [188], k-nearest neighbours algorithms (k-NN) [51] and artificial neural networks (ANNs) [221]. SVMs are one of the most popular supervised learning methods for classification. They determine a hyperplane to separate data with different labels by maximizing a margin (the distance between two hyperplanes which give a boundary to the positive data and the negative respectively). SVMs can be used to solve both linear and non-linear problems. Fig. 2.6 shows an simple 2D linear SVM case in which the hyperplane equation is denoted by

$$\mathbf{w} \cdot \mathbf{x} - \mathbf{b} = \mathbf{0} \tag{2.8}$$

Here $\mathbf{w}$ is the normal direction of this hyperplane and $\frac{b}{\|\mathbf{w}\|}$ denotes the offset of the hyperplane to the origin along $\mathbf{w}$. For a positive or a negative class, the value of $\mathbf{w} \cdot \mathbf{x} - \mathbf{b}$ is above 1 or below -1, respectively. $\mathbf{w} \cdot \mathbf{x} - \mathbf{b} = \pm\mathbf{1}$ can be considered as two margin hyperplanes and the distance between these two hyperplanes is $\frac{2}{\|\mathbf{w}\|}$. Thus, $\|\mathbf{w}\|$ is to be minimized in order to maximize the margin in form of the distance between these two hyperplanes.

To improve the classification performance, especially to address the problem of overfitting, a group of classifiers can be integrated and organized instead of a single classifier. This is the basic idea of ensemble learning. Typical ensemble learning techniques include bagging [31], boosting [71, 176], probabilistic boosting trees (PBT) [198] and random forests [32, 52]. Bagging is an ensemble learning in which a number of subsets of the data (bags) are sampled and a number of classifiers are trained based on each subset. Then the final output of bagging is the average of the outputs of all trained classifiers. Boosting is a method to organize a number of weak classifiers to form a strong classifier. Each classifier is assigned a weight and these weights are used to average and combine all weak classifiers. Probabilistic boosting tree employs both the idea of decision tree and boosting. One of the most popular ensemble learning is random forests which are a group of decision trees. A bagging strategy is used to select a number of subsets of the data and an individual decision tree is trained based on each data

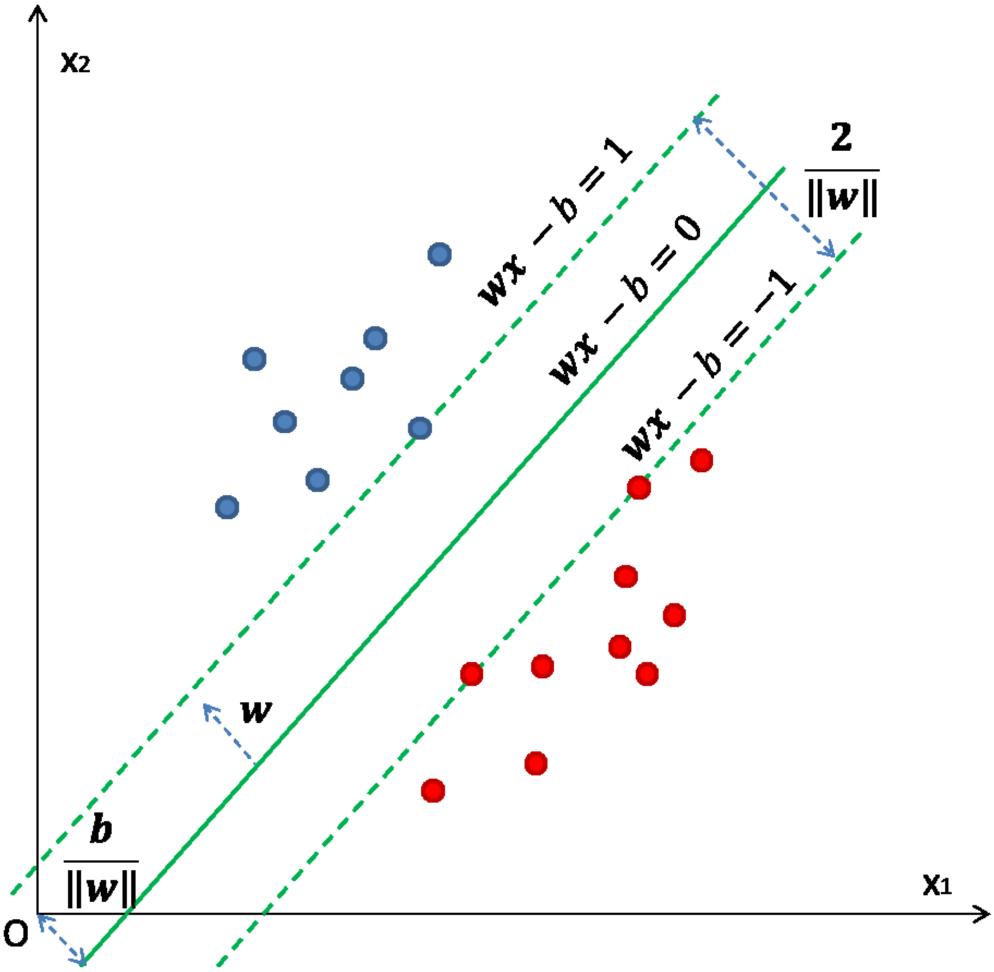


Figure 2.6: A simple 2D linear SVM case.

subset. However, to eliminate the correlation among different trees, during the training of each individual tree, a subset of features is randomly selected to form the tree. Random forests can also be used for unsupervised learning [52]. Another advantage of ensemble learning is that it is easier to carry out incremental learning by updating the weight of each weak classifier. Thus these method can be used for online learning [157, 158, 170]. For example in [200, 77], to address the tracking problem, various types of features can be considered as weak learners and measurements are evaluated by the combination of these features with the corresponding weights. In each iteration, new training data are used to update the weights of these features and eventually the most salient features have the highest weights during the training process.

Many of high dimensional features are redundant. A redundant feature space is not only high in computational cost, but also affects the general performance of detectors. Thus, the transformation of high dimensional features to a more compact and low-dimensional representation is of central importance in the field of machine learning. Principal component analysis (P-CA) [58] is a widely-used method to reduce dimensionality by analyzing the singular values of the corresponding covariance matrix. Manifold learning [177, 195, 17, 204, 43, 178] has been proposed to retain the useful dimensions in a manifold space to preserve useful information in a low-dimension representation. Sparse coding [127, 128, 114], as well as low rank techniques [129], are other alternative ways to achieve a sparse feature representation by matrix analysis. Using sparse learning as an example, the original features are written as a matrix $Y_{m\times n} = (y_1, \ldots, y_n)$ with the size of each feature vector n and the dimension m. Then an over-complete dictionary $D_{m\times k}$ with the size of the dictionary k as well as the corresponding sparse representation $X_{k\times n} = (x_1, \ldots, x_n)$ is found by minimizing the following function:

$$\min_{D_{m\times k}, x_i} \sum_{i=1}^{n} (\|y_i - D_{m\times k} x_i\|_2^2 + \lambda S(x_i)) \tag{2.9}$$

Here $S(x_i)$ is a sparsity constraint (usually the L_0 or L_1 norm). Then the new representation $\widetilde{Y}_{m\times n}$ can be obtained by applying

$$\widetilde{Y}_{m\times n} = D_{m\times k} X_{k\times n} \tag{2.10}$$

A dictionary can be learned using method proposed in [4]. Through sparse coding, the original features can be represented by a sparse code of coefficients. A mechanism transforming the original features to a sparse code can be considered as a sparse encoder. Compared to the original feature, a sparse code reduces the redundant information while preserving useful information. If a number of sparse encoders are stacked (the output of one sparse encoder is used as the input to another sparse encoder) to form a hierachical network, then a deep network is constructed [18]. Fig. 2.7 shows an example of two layer networks with the original feature in blue, the first layer coding in green and the second (the final) coding in red. The yellow layer is a classifier based on the sparse coding in red. The final sparse code is expected to preserve more compactly valuable information than the original features or other intermediate spare codes. This code can be considered as a new feature created by unsupervised training. In contrast

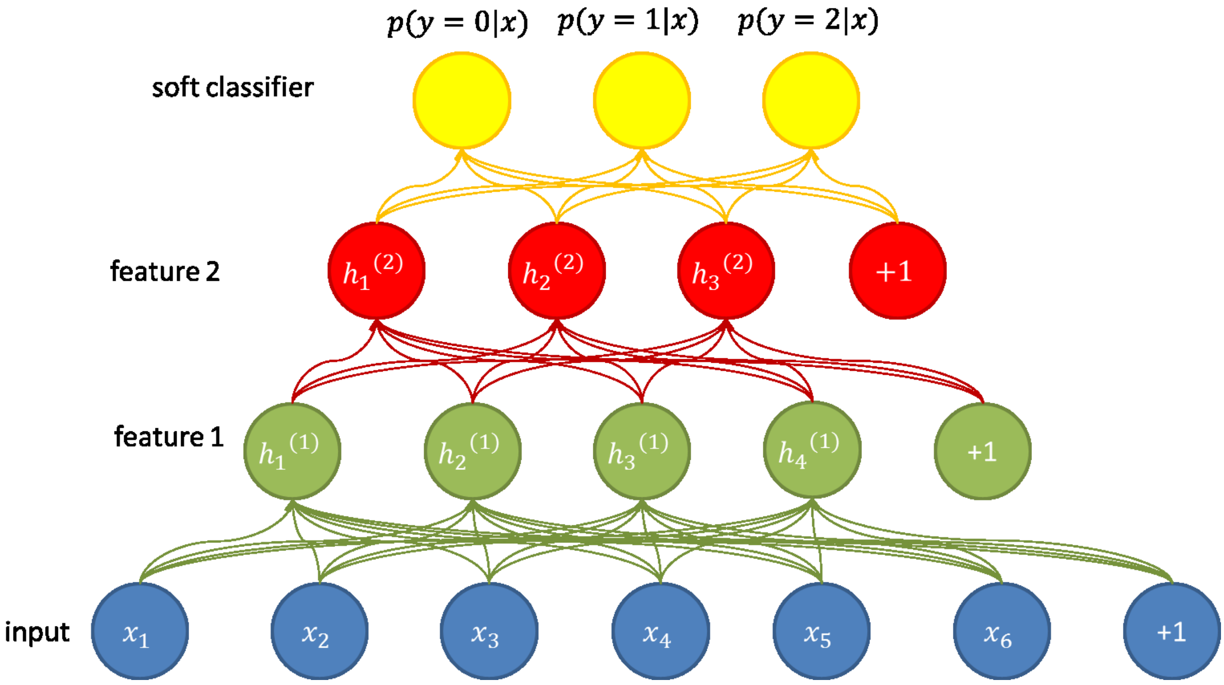


Figure 2.7: An example of a two-layer stacked auto-encoder network with a softmax classifier.

with the original feature, the new feature usually achieves better performance when it is used for supervised learning. Conversely, the original feature can be recovered from a sparse code. This process is called decoding. The recovered feature from a sparse code is expected to be noise-free since the source, the sparse code, is expected to be more compact and without redundant information. Deep learning algorithms [18], which fit such an architecture are becoming increasingly popular for their capability to establish a network architecture to create a good feature representations in an unsupervised fashion. Deep neural networks (DNN) [19, 110], restricted Boltzmann machines (RBM) [171] and deep belief networks (DBN) [26] can provide alternative architectures to the stacked sparse encoder. However, a large number of training data are required when deep learning approaches are used.

2.2 Optimization and Filtering

2.2.1 Optimization

No matter how local detection results are obtained, false positives and false negatives always exist in real-world situations. To address this problem, other constraints such as smoothing have to be used. This allows the formulation of an optimization problem. In the field of computer vision, in addition to conventional continuous optimization based on gradient descent, discrete optimization approaches are popular since they can reach the global solution efficiently and will not be trapped in local minima. Commonly, graphical models are used to model the objective function to be optimized [194].

For example, assume a graphical model is denoted by $G =< \mathbf{V}, \mathbf{E} >$ with a vertex set $\mathbf{V}$ and an edge set $\mathbf{E}$. Let us further assume that objective function to be solved can be modeled by a set of variables. Each variable corresponds to a vertex in the graph and the relationship between any two variables is modeled as an edge linking the two corresponding vertices. In addition, we assume that all of the variables share the same discrete and finite value space $\mathbf{L}$ and each value in this space will be viewed as a label. The labels (or values) of all variables are represented as a vector $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n})$. Then, the discrete optimization problem is to find the optimal label for each variable. The optimal label assignment is expected to correspond to a global minimum of the cost function in form of:

$$\min_{\mathbf{x}}(\sum_{p \in \mathbf{V}} u_p(x_p) + \lambda \sum_{p,q \in \mathbf{E}} w_{p,q}(x_p, x_q)) \tag{2.11}$$

This form of the cost function consists of a unary term $u_p(x_p)$ and a pairwise term $w_{p,q}(x_p, x_q)$. The unary term uses constraints from each individual vertex and the pairwise term enforces constraints on a pair of vertices (an edge). The exact form of the unary term and pairwise terms depend strongly on the particular application.

In image segmentation [180, 30], graph-cut based optimization plays an increasingly important role to solve this type of optimization problem [29]. The 'maximizing flow and minimizing

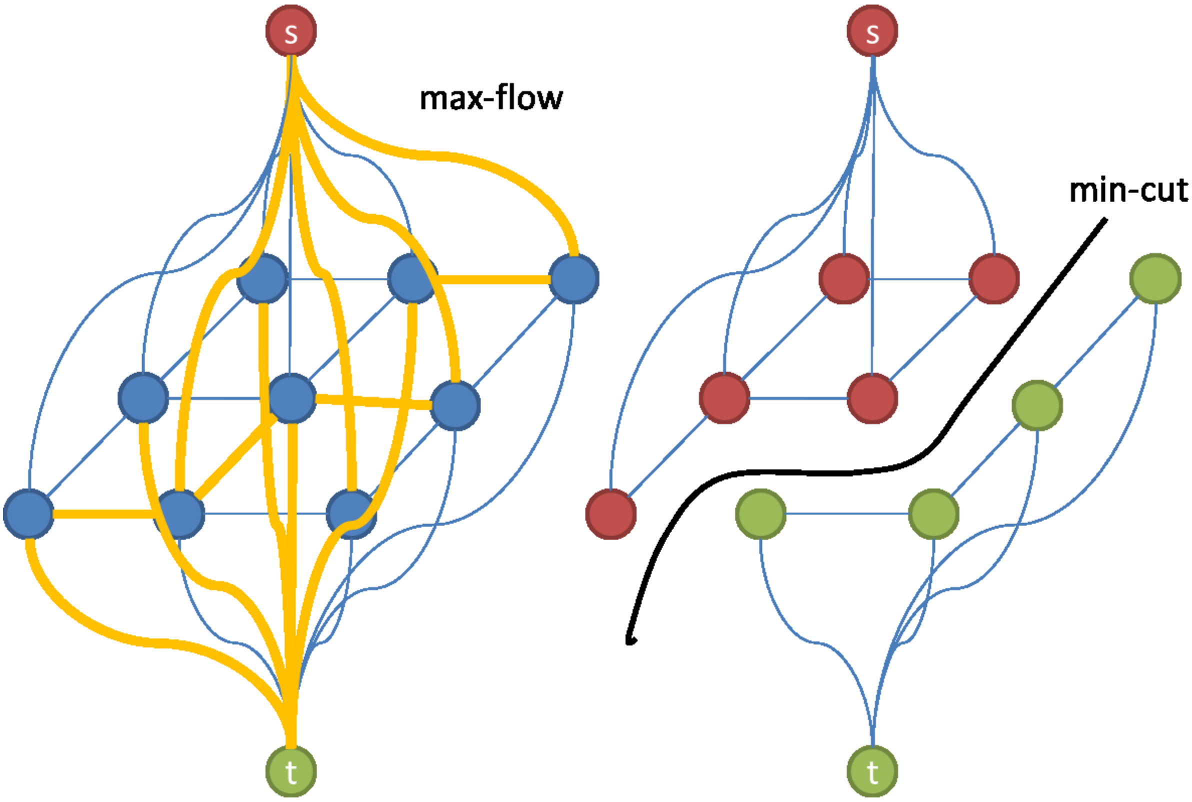


Figure 2.8: Binary label classification based on max-flow/min-cuts optimization.

cuts'(max-flow/min-cuts) algorithm, which is proposed in [28, 105], is one of the most widely used approaches to solve the optimization problems described above. Fig. 2.8 shows the process of max-flow/min-cuts solution: Given a graphical model with both vertices and weighted edges. We assume all vertices in a graph are linked to two additional vertices: a source s or a sink t by additional edges. Imagine that there is a flow from the source to the sink, through all edges. Each edge has a maximal flow capacity equal to its weight. If we increase the flow in the network some edges will reach their maximum flow capacity. These edges compose a set, which is referred as a $s - t$ cut, separating the vertices into two sets, corresponding to two binary labels. This technique can also solve multi label optimization problems [28, 105].

To further accelerate the optimization process, a primal-dual strategy is introduced in [107] and the fast primal-dual (fast-pd) algorithm is proposed. The key idea is to transform the objective function in Eq. 2.12 into the following binary optimization problem with the argument set

$\mathbf{x} = \{x_{p,a}, x_{pq,ab} : p, q \in \mathbf{V}, a, b \in \mathbf{L}\}$:

$$\min_{\mathbf{x}}(\sum_{p\in\mathbf{V}}(\sum_{a\in\mathbf{L}} u_p(a)x_{p,a}) + \lambda \sum_{p,q\in\mathbf{E}}(\sum_{a,b\in\mathbf{L}} w_{p,q}(a,b)x_{pq,ab})) \tag{2.12}$$

s. t. the following constraints:

$$\sum_{a\in\mathbf{L}} x_{p,a} = 1 \tag{2.13}$$

$$\sum_{a\in\mathbf{L}} x_{pq,ab} = x_{q,b} \tag{2.14}$$

$$\sum_{b\in\mathbf{L}} x_{pq,ab} = x_{p,a} \tag{2.15}$$

Note that the value of each argument in $\mathbf{x}$ is either 1 or 0. Thus this is a typical integer programming problem. $x_{p,a} = 1$ means that label a is assigned to vertex p and $x_{pq,ab} = 1$ means that labels a and b are assigned to vertex p and q respectively. Eq. 2.13 indicates that only one label is assigned to each vertex. Eq. 2.14 and Eq. 2.15 preserves the consistency between the values of $x_{p,a}$, $x_{q,b}$ and $x_{pq,ab}$. The fast primal dual algorithm aims to accelerate the optimization by solving the dual problem of this integer programming problem. Other approaches include random walks [79].

In addition to graph-cut approaches, there are other approaches to solve this type of optimization problem. An early way to solve this problem is using belief propagation (BP) techniques [65], which are expensive in terms of computational cost. In [70] an improved loopy belief propagation (LBP) is proposed to accelerate the optimization process. At the same time, message-passing has been proposed to address the same type of optimization problem. The tree-reweighted message passing (TRW) algorithm, which is the most popular version, is o-riginally proposed in [202]. The basic idea of TRW is to decompose the graphical model into several components in form of simple tree structures. Then the optimization of each individual component is solved by belief propagation. Finally all components are integrated again by averaging the weights of vertices shared by different components in the entire graphical model. The convergence of TRW has been shown to be guaranteed by an improved version proposed in [103]. In [106], the dual decomposition has been introduced to enhance the performance of the message passing approaches. A comparison between LBP, TRW and graph-cuts can be found in [194].

2.2.2 Filtering

In tracking problems, the measurements of features may be obtained sequentially instead of simultaneously. In this case, an optimization framework is only applicable to a frame for which the measurements have been obtained at the same time. In the case of measurements sequentially obtained from different frames, filtering techniques, which are often used in the field of robotics [196], can be employed. These filtering techniques can predict and update a set of pre-defined states sequentially using new measurements through a system model and a measurement model.

Kalman filters [207], are often used to address dynamic problems with linear models and Gaussian noise. Two basic stages are included in a Kalman filter framework: prediction and updating. Assume that the state vector under estimation at the $k-1$th time is denoted by $\hat{\mathbf{x}}_{k-1|k-1}$, with its corresponding uncertainty covariance matrix $\hat{\mathbf{P}}_{k-1|k-1}$. During the prediction stage at time k, the state vector $\hat{\mathbf{x}}_{k|k-1}$ as well as the corresponding uncertainty matrix $\hat{\mathbf{P}}_{k|k-1}$ are predicted based on the system model and the information at the $k-1$th time; then the measurement $\mathbf{y_k}$ at time k is obtained and the estimation of the state vector and the covariance matrix are updated into $\hat{\mathbf{x}}_{k|k}$ and $\hat{\mathbf{P}}_{k|k}$ based on a measurement model and new arriving measurements. Fig 2.9 shows the overview of the framework. The numerical calculation of covariance matrix is sometime unstable since it is not always inversible. The squared root decomposition has been introduced into the calculation process to address this problem [11]. To accelerate the calculation when the dimension of the state vector is very large, the information matrix which is the inverse of covariance matrix is calculated instead of the covariance matrix.

The sparse extended information filter (SEIF) [197] achieves high computational efficiency. The basic idea of SEIP is using a sparse information matrix (many low-value elements are set to zero) to approximate the real information matrix with a small tradeoff in accuracy. The basic Kalman filter assumes linear models. To address non-linear problems, the extended Kalman filter (EKF) [196] and Unscented Kalman filter (UKF) [203] have been proposed. However,

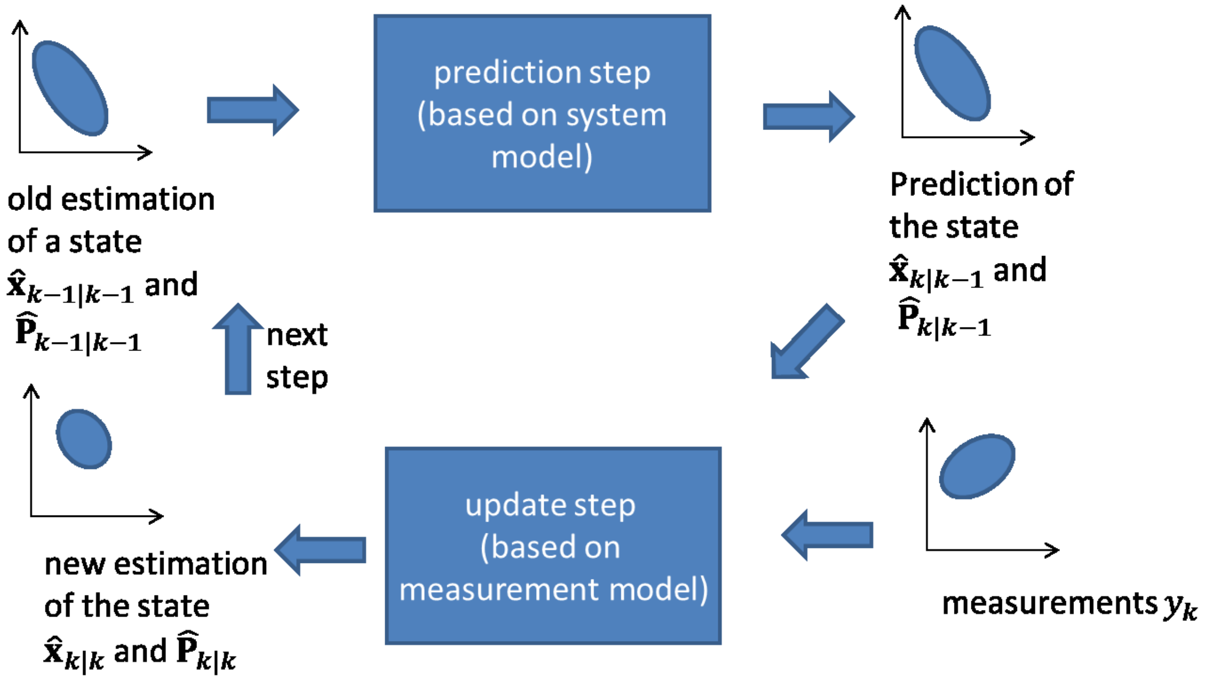

prediction step (based on system model)
Prediction of the state $\hat{\mathbf{x}}_{k|k-1}$ and $\widehat{\mathbf{P}}_{k|k-1}$
old estimation of a state $\hat{\mathbf{x}}_{k-1|k-1}$ and $\widehat{\mathbf{P}}_{k-1|k-1}$
next step
update step (based on measurement model)
measurements y_k
new estimation of the state $\hat{\mathbf{x}}_{k|k}$ and $\widehat{\mathbf{P}}_{k|k}$

Figure 2.9: Kalman filter framework.

one limitation of the Kalman filter is that it assumes that the noise of either the system model or the measurement model follows a Gaussian distribution. However, this assumption is not always applicable.

Another filtering techniques are particle filters [6], also called Sequential Monte Carlo (SMC). Compared with a Kalman filter, they make no assumption on the noise distribution. The key idea of particle filter is to use sampled particles (which are a group of sampled state vectors) to approximate the probability distribution. During the prediction stage, a certain number of particles are sampled at the current time k based on the previous state $k-1$ their corresponding probability distribution, as well as the system model. Then new measurements are used to evaluate each particle and re-weight them, based on the measurement model. The re-weighted particles are used to approximate the likelihood. Finally, based on the likelihood, the particles are re-sampled to form a new probability distribution for the current time k. Fig. 2.10 shows an brief framework of particle filter. More discussions on the details of applying particle filters can be found in [164, 120, 37, 56, 24, 25].

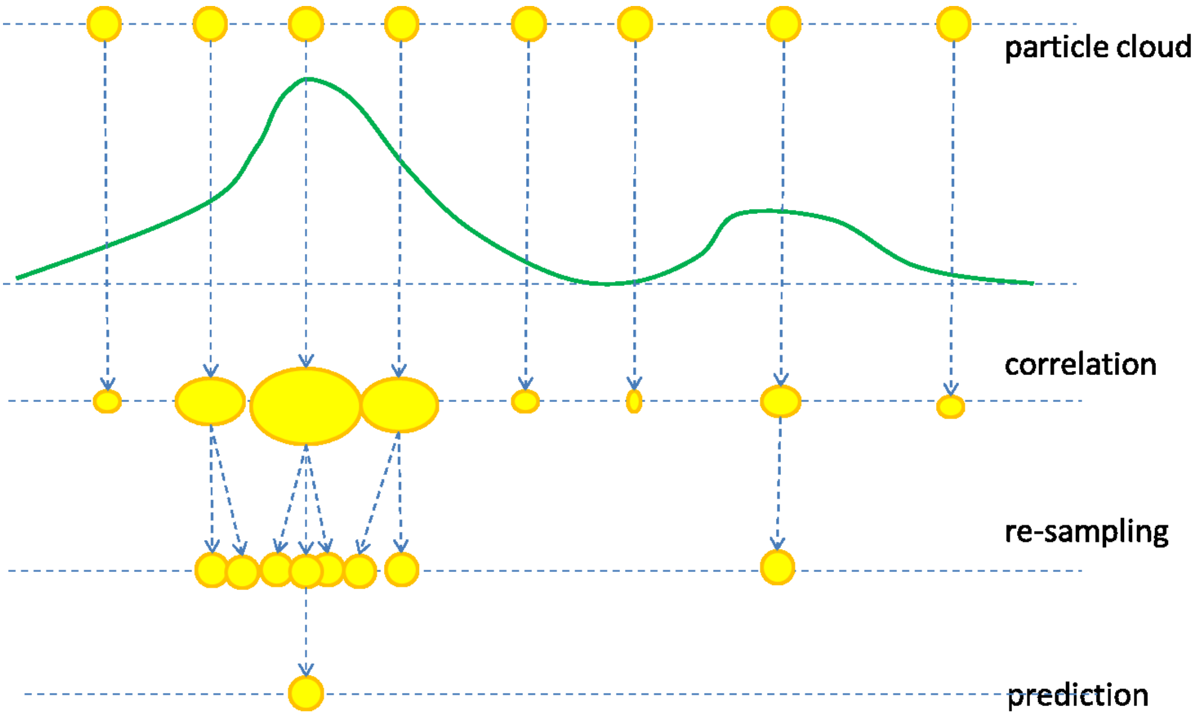


Figure 2.10: Particle filter framework.

2.3 General Segmentation and Tracking Applications in Computer Vision

There exists a vast amount of literature on image segmentation and tracking. Therefore we only review some of the key approaches. Emphasis is placed specifically on contour and curve segmentation and tracking, which is directly related to our research.

2.3.1 Segmentation

Image segmentation, as one of the basic problems in computer vision, has thrived since the very beginning of computer vision. The methodologies for segmentation vary based on different applications. Early efforts have mainly focused on simple methodologies such as clustering [220, 12], edge detection [119], histogram-based techniques [155] and adaptive thresholding [13, 14]. Clustering aims to segment images by grouping similar structures together. The clusters can also be merged and split iteratively to refine the segmentation [91]. Segmentation can also be refined by region-growing, which assumes that a seed region is manually identified from which the segmentation can be grown [42, 2].

Expectation-Maximization Algorithm and Hidden Random Markov Field

If the image to be segmented is characterized by properties that are known a-priori and can be modeled using a small number of parameters, algorithms such as the Expectation-Maximization algorithm (EM), can be employed [225].

The EM algorithm is an estimation strategy to estimate the values of unobservable states $\mathbf{y}$ and the parameters of the probability functions θ given the observable states $\mathbf{x}$ by iteratively maximizing the likelihood which is in form of:

$$L(\mathbf{x};\theta) = \mathbf{p}(\mathbf{x} \mid \theta) \tag{2.16}$$

The EM algorithm can be divided into two steps: the Expectation step (E-step) and the Maximization step (M-step). During the E-step, the distribution of the unobservable states $\mathbf{x}$ under conditions $\mathbf{y}$ and the current estimation of θ^t is estimated. Then the expectation of the log likelihood under the conditional distribution $\mathbf{y}$ is represented as:

$$Q(\theta) = E_{\mathbf{y}|\mathbf{x},\theta^{\mathbf{t}}} \log(L(\mathbf{x}, \mathbf{y}; \theta)) \tag{2.17}$$

In the subsequent M-step the model parameters θ are updated by maximizing $Q(\theta)$:

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta) \tag{2.18}$$

Hidden Markov models (HMM) are Markov chains or MRFs with states which cannot be directly observed. HMMs have been successfully used in speech recognition [166] and handwritten character recognition [111]. Fig. 2.11 shows an example of hidden Markov model in which $\mathbf{x}$ denote the unobservable states and $\mathbf{y}$ denotes the observations. The state transition probabilities are given by $\mathbf{a}$ and $\mathbf{b}$.

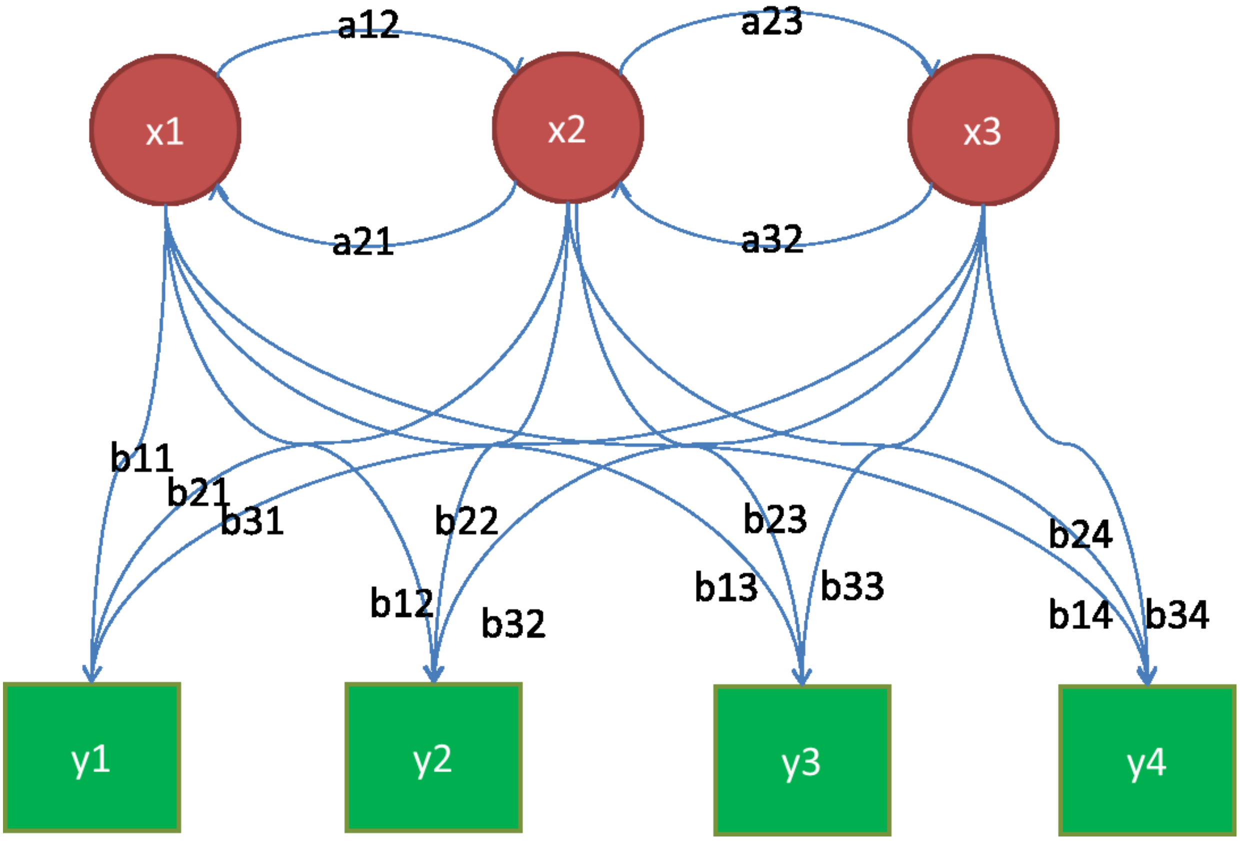


Figure 2.11: An example of hidden Markov model with hidden states **x** in red circles, observable states **y** in green boxes and state transition probabilities **a** and **b**.

In [225], a hidden MRF is defined as the combination of unobservable states $\mathbf{y} = \{y_i, i \in S\}$ representing the image segmentation labels for each pixel, and observable states $\mathbf{x} = \{x_i, i \in S\}$ representing the intensities of each pixel. S denotes an index set. The form of conditional probability distribution of $\mathbf{x}$ under $\mathbf{y}$ is known (normally a Gaussian distribution is used) but the parameters of the probability distribution θ is unknown. Then the EM algorithm is used to estimate the parameters θ and the hidden states $\mathbf{x}$ iteratively.

For Markov random fields, the segmentation can also be transformed to an optimization problem. In such cases, the arguments to be optimized are the labels of each pixel indicating the segmentation categories. The cost function encodes both constraints from the image intensities and the model itself (i. e. the smoothness for a contour curve). Usually this optimization cannot be solved efficiently by conventional continuous gradient-based optimization since it can be easily trapped in local minima. In contrast, discrete optimization, such as graph-cut [27, 169, 30], is popular in such applications [64, 223, 80, 216].

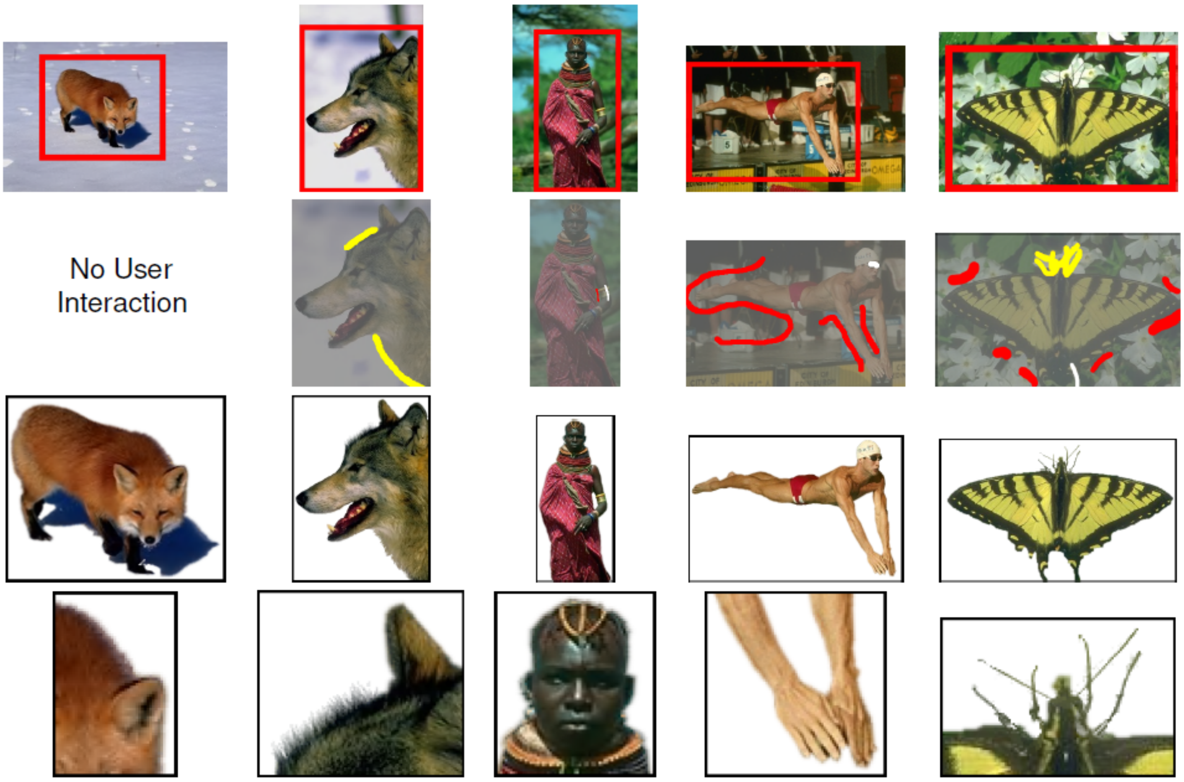


Figure 2.12: Examples of GrabCut applications. (Figure from [172]).

The GrabCut method [169] is one particularly successful example of segmentation approaches based on graph cuts. GrabCut is an interactive segmentation: First, a bounding box around the object to be segmented is specified by user. Then, a Gaussian mixture model is used to estimate the color distribution of the object. This model is used to construct a MRF with an objective function that encourages smoothness of the segmentation across neighboring pixels. The whole segmentation process can be iteratively refined by max-flow/min-cuts optimization approaches until convergence. Fig. 2.12 shows an example of the GrabCut segmentation. The forms of the cost functions which can be solved by the original graph-cuts are limited. However, many improvements have been made to make graph-cuts approaches applicable to a more general form of cost functions [104, 143, 167, 5].

2.3.2 Tracking

Optical Flow

In object tracking, most of studies focus on the tracking of objects in natural images. Many early studies use optical flow (OF) to measure the relative displacements between two adjacent frames [90, 122]. Optical flow is a method that calculates the relative motion between two adjacent image frames. One of the basic assumptions of OF is that when a point at pixel (x, y) at t-th frame moves to another pixel $(x + \Delta x, y + \Delta y)$ at $t + \Delta t$th frame, then the intensity remains constant:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{2.19}$$

Another assumption is that when using a Taylor expansion, the higher order derivative terms (above 2) are close to zero. Then the above equation can be approximated as follows:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \tag{2.20}$$

By applying Equ. 2.19 to Equ. 2.24 the basic optical flow equation can be obtained:

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0 \tag{2.21}$$

$V_x = \frac{\Delta x}{\Delta t}$ and $V_y = \frac{\Delta y}{\Delta t}$ denote the relative displacements in x and y between two adjacent frames for this pixel. However, the relative displacements cannot be obtained by only one optical flow equation. This is referred as the aperture problem. In order to compute the local displacements, a large number of optical flow equations at all pixels are used. Usually the relative displacements at pixels in a neighborhood are assumed to vary smoothly. Then the displacements for each pixel can be solved. These displacements can be used to track the object across different frames or analyze the motion of the object. Fig 2.13 shows the optical flow on an image (left) and the strength of the flow (right).

The use of optical flow techniques is limited to only small motion tracking. The reason is that for large motions, the assumption that the higher order derivatives of the Taylor expansion are

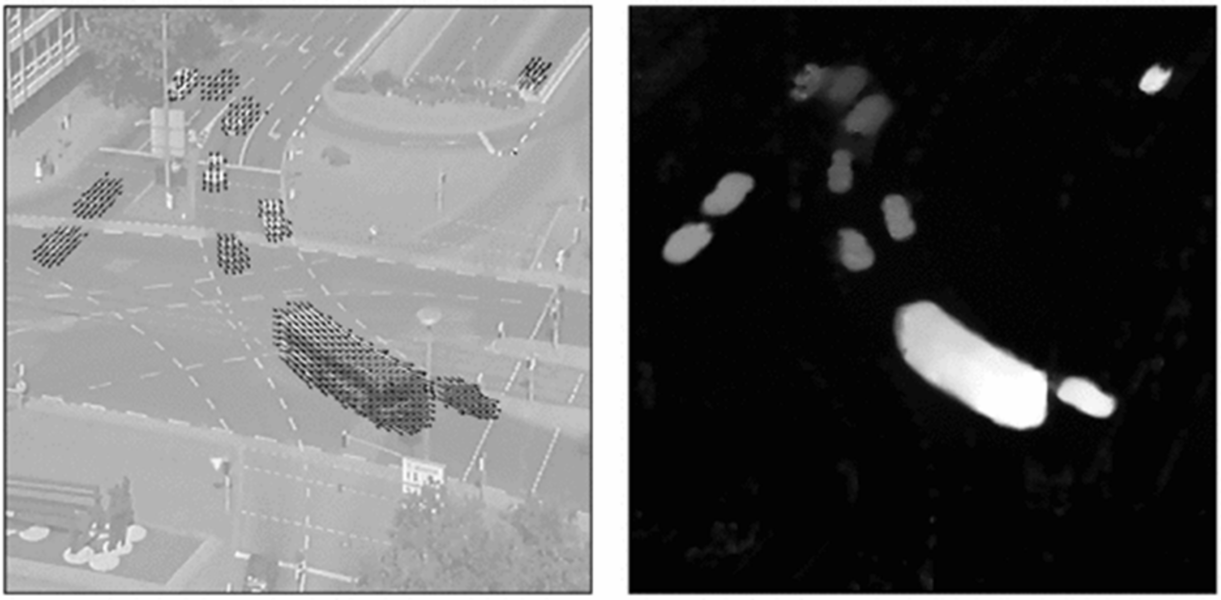


Figure 2.13: An example of the optical flow of an traffic scene (left) and the magnitude of the flow (right) (Figure from [35]).

close to zero is not valid. However, there are methods to address the large motion problem [35]. Other problems include the flow discontinuity at the boundary of different objects. This has been discussed in [222]. A recent evaluation on OF can be found in [10].

Learning-based tracking

As a result of the increased popularity of machine learning methods, learning-based approaches have been intensively studied for its application to object tracking. Boosting is one of the most commonly-used learning-based approaches to be used for tracking [76, 78, 9] due to its well understood theoretical underpinnings [71] and its simple updating mechanism for online learning [76, 78].

In the traditional boosting algorithm the strong learner $h^{strong}(x)$ is a linear combination of N weak learners $h_n^{weak}(x)$ with corresponding weights α_n:

$$h^{strong}(x) = sign(\sum_{n=1}^{N} \alpha_n h_n^{weak}(x)) \tag{2.22}$$

Usually α_n can be estimated by minimizing the classification error for each weak learner. However, to do this, all data samples are required to be available during training. This limits the applicability of bootsing in online learning scenarios where only one sample is available at each time.

To address this [76] develop an online boosting approach: Instead of combining weak learners directly, a strong learner is constructed by a linear combination of N weak selectors $h_n^s(x)$ in form of:

$$h^{strong}(x) = sign(\sum_{n=1}^{N} \alpha_n h_n^s(x)) \tag{2.23}$$

A selector is defined based on a set of M weak learners and the selector selects the weak learner with the lowest classification error in this set. The usage of selectors allows the strong classifier to be available at any time. In [78], a semi-supervised online boosting approach is proposed to make the tracking more robust. In this framework the data without labels can also be used to update the system. A distance is defined on two unlabeled data. During the online training process, the tracker is updated not only by the classification error for the labelled data but also by these distances for unlabeled data. Fig. 2.14 shows an example of the tracking results from [76].

To address the drift problem in a tracking system, a framework consisting of tracking, learning, and detection (TLD) is proposed for real-time tracking applications [98]. TLD is a framework that addresses the problem of drift which affects the performance of a tracker in long sequences. As Fig. 2.15 shows, the motion of the object is estimated by the tracker. Simultaneously, the detector, which works independently from the tracker, locates all similar appearances of the object by running the detector across the whole image. The learning component assesses the performance of both the tracker and the detector. Based on their performances, it generates training examples to update the detector in order to avoid false positives and false negatives matches which have been observed.

Random forests have also been used in a tracking framework to online estimate Hough transforms. This is also referred as Hough forests in [72, 75]. During the training stage, a random forest is trained by using image patches with a class label and an offset to the centroid of image. During the detection stage, the test image patches are passed through the random forest. The

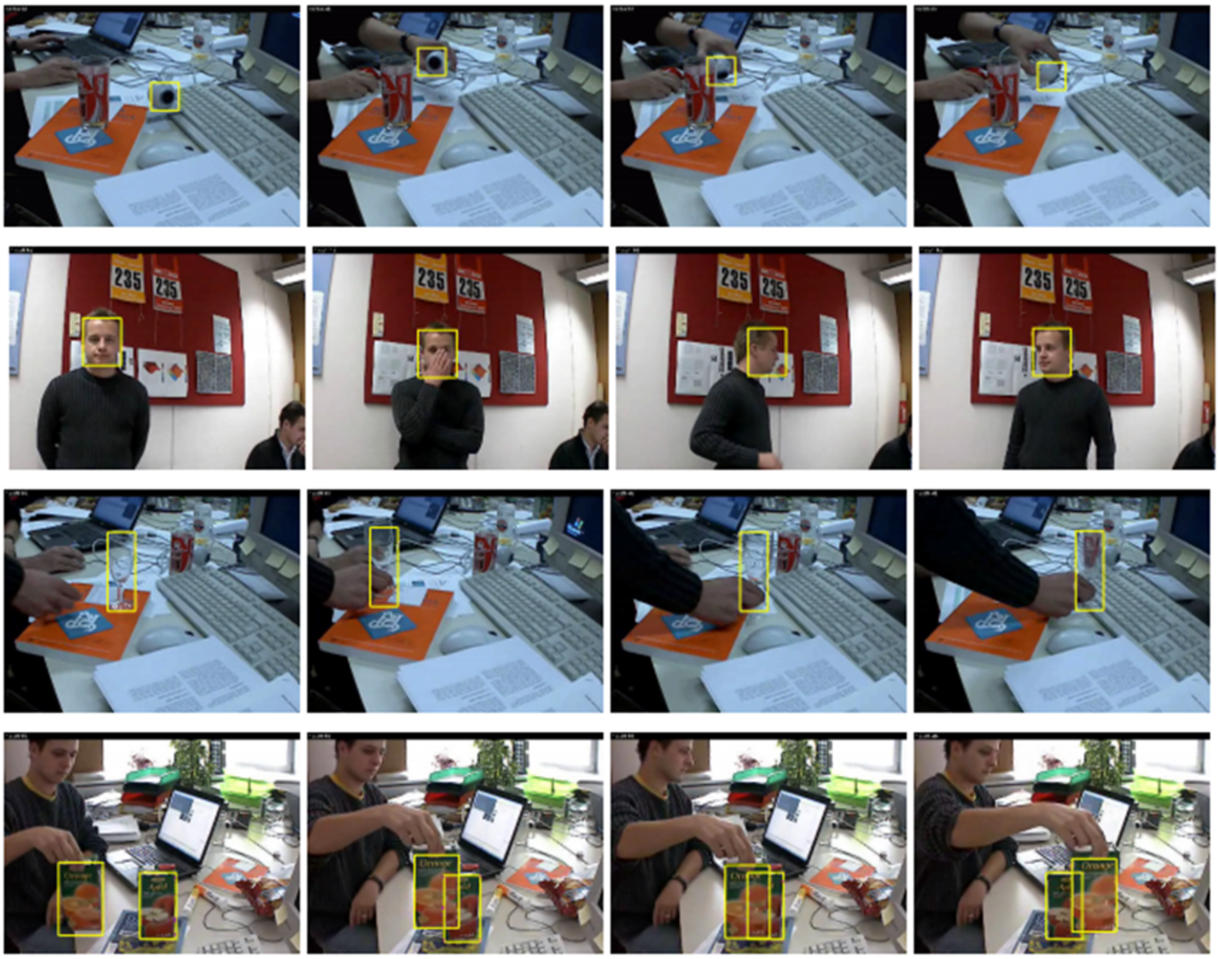


Figure 2.14: Tracking results based on online boosting (Figure from [76]).

detection result is calculated by Hough voting on the outputs of the forest.

Non-rigid tracking

Non-rigid object tracking can be viewed as the problem of curve or contour tracking. A significant amount of research has been carried out in this area, much of which focuses on active contour models (or *snakes*) [99]. These models aim to fit a contour to the image by minimizing an energy function which is of the form:

$$\min_{\mathbf{\Gamma}(s)} \int_0^1 (E_{internal}(\mathbf{\Gamma}(s)) + E_{image}(\mathbf{\Gamma}(s)) + E_{user}(\mathbf{\Gamma}(s)))ds \tag{2.24}$$

Here $\mathbf{\Gamma}(\mathbf{s})$ denotes the parameterization of the contour. $E_{internal}$ forces the contour to be smooth while E_{image} attracts the contour towards features in the image. E_{user} allows the

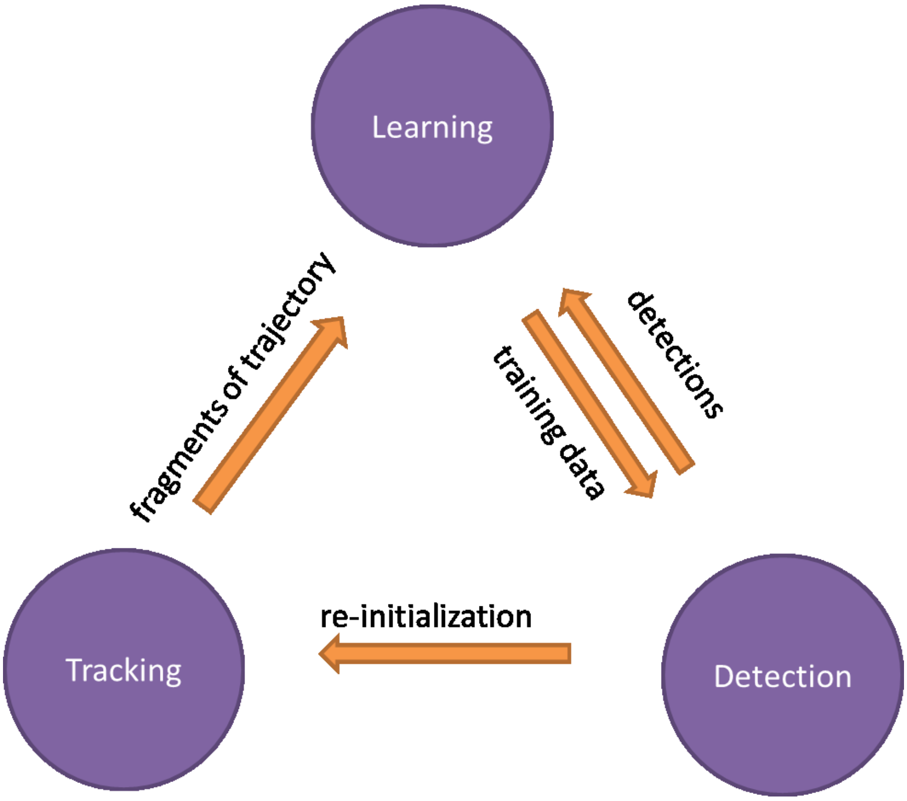


Figure 2.15: The overview of the TLD framework.

incorporation of user interaction in form of constraints. The solution to the above equation is usually found via iterative optimization techniques which can be time consuming. Active shape models (ASM) [48] construct a contour model represented by a set of landmarks on this contour, along with the probability distribution of these landmarks. The non-rigid motion of this contour can be represented by the displacements of these landmarks. During the training stage, the contours for all shapes are aligned in a local coordinate system. Then the distribution of each landmark in this local coordinate is learned by using principle component analysis (PCA). During the detection stage, the global transform of the model to the image, as well as the local displacements for all landmarks, is estimated. Fig 2.16 shows an example of cardiac contour fitting by ASM in ultrasound images. The concept of ASMs has been successfully extended to active appearance models (AAM) which use both shape and appearance information of the object being tracked [47]. In addition, filter-based methods have been used for curve tracking. In [95] the parameters of a contour model being tracked are updated based on a particle filter.

Mean shift [44, 45, 46] is another real-time tracking approach which can track a non-rigid

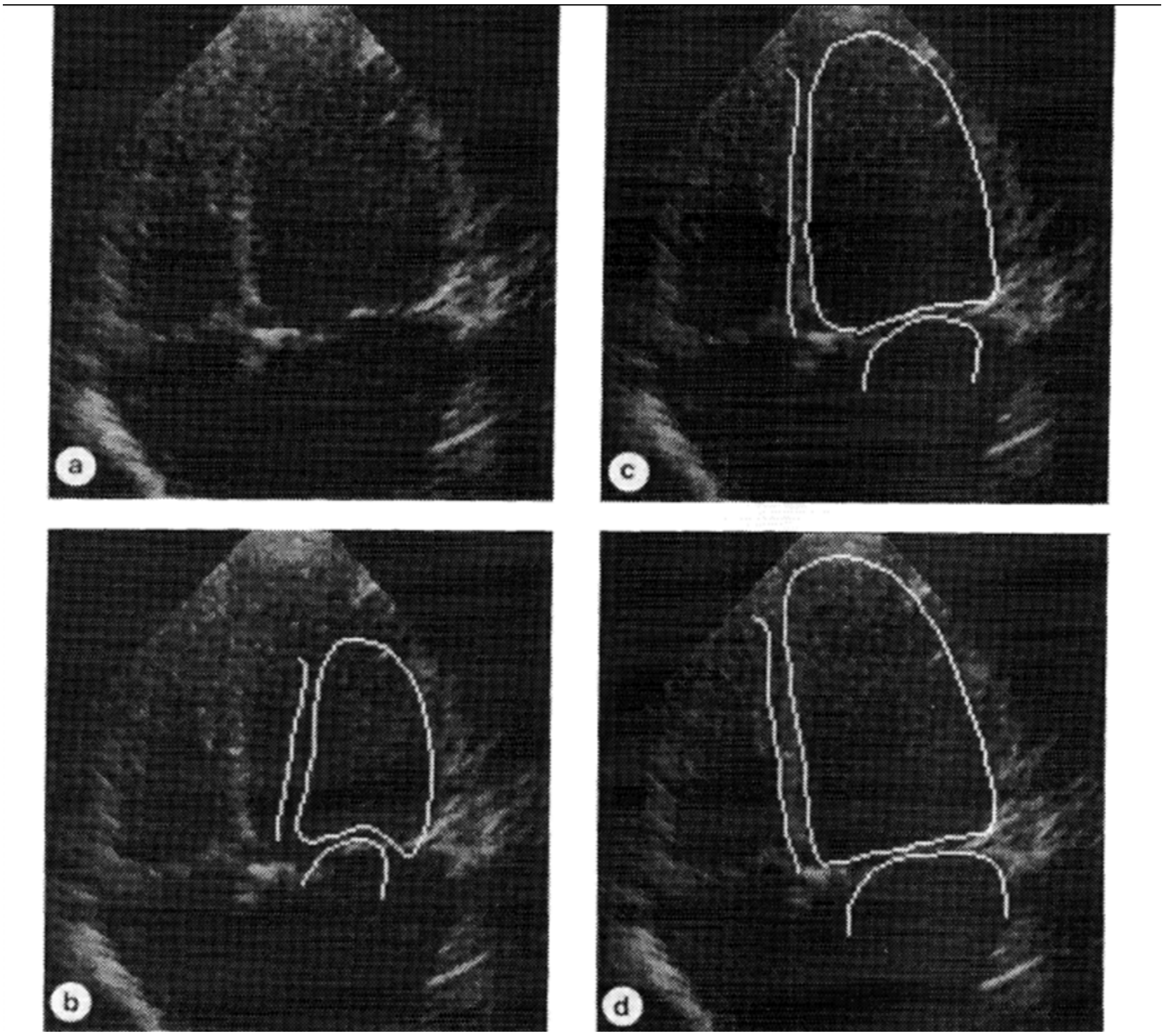


Figure 2.16: Tracking of the heart in ultrasound images using ASMs (Figure from [48]).

object within a local patch. The mean shift is an iterative method to find the peak of a density function (which is unknown) based on a finite number of sampled data. Assume x_i denotes one sampled data point and x is the current estimation of the peak. Then the mean shift is estimated by:

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)} \tag{2.25}$$

Here $K(x_i - x)$ is a kernel function (ususally a Gaussian) which determines the weight of each sampled data point x_i. $N(x)$ denotes the neighborhood of x. The value of x is updated with $m(x)$ and the whole process is repeated until x and $m(x)$ are close to each other. When mean shift is used in visual tracking, the object to be tracked is modeled as a template, which is a histogram based on intensities. Then during tracking, on each pixel the intensity histogram is constructed and compared with the template. Then a local confidence map is constructed based on the similarities at each pixel to the template. The peak of the confidence map is identified by the mean shift procedure. However, since mean shift is a patch-based approach, it can only address deformation within a local patch.

For large deformations, part-based models [62] have been proposed. These models can establish the relationship between different parts of the object to be tracked [63]. In the part-based model, a coarse global detector is used to approximately identify the object. Simultaneously, multiple local detectors that correspond to the parts of the object are used. Both the global detector and local part detectors are evaluated at each pixel of the image. Then the detection result at one pixel is evaluated based on both the responses of all detectors and the spatial relationship between detected positions. Then a confidence map, which indicates the potential locations of the object to be tracked, is constructed. Fig. 2.17 shows an example of pedestrian detection based on part-based model.

More recent contour-based tracking techniques for non-rigid objects are increasingly designed for particular applications such as face tracking [62, 39, 218] or body tracking [182, 100, 224]. However, the model of a face or body usually assumes a limited degree of freedom. In addition salient features such as eyes for faces or arms for bodies are available. Thus the tracking approaches for face and body cannot be extended to the tracking of other non-rigid objects. In addition, curve tracking has been formulated as a discrete optimization problem. Then a large number of discrete optimization approaches [29, 30, 107, 104, 103] can be used. One typical example [86] for curve tracking is based on fast-pd algorithm. The proposed curve tracking algorithm was also tested on guide-wire tracking based on X-ray images. This work will be detailed later.

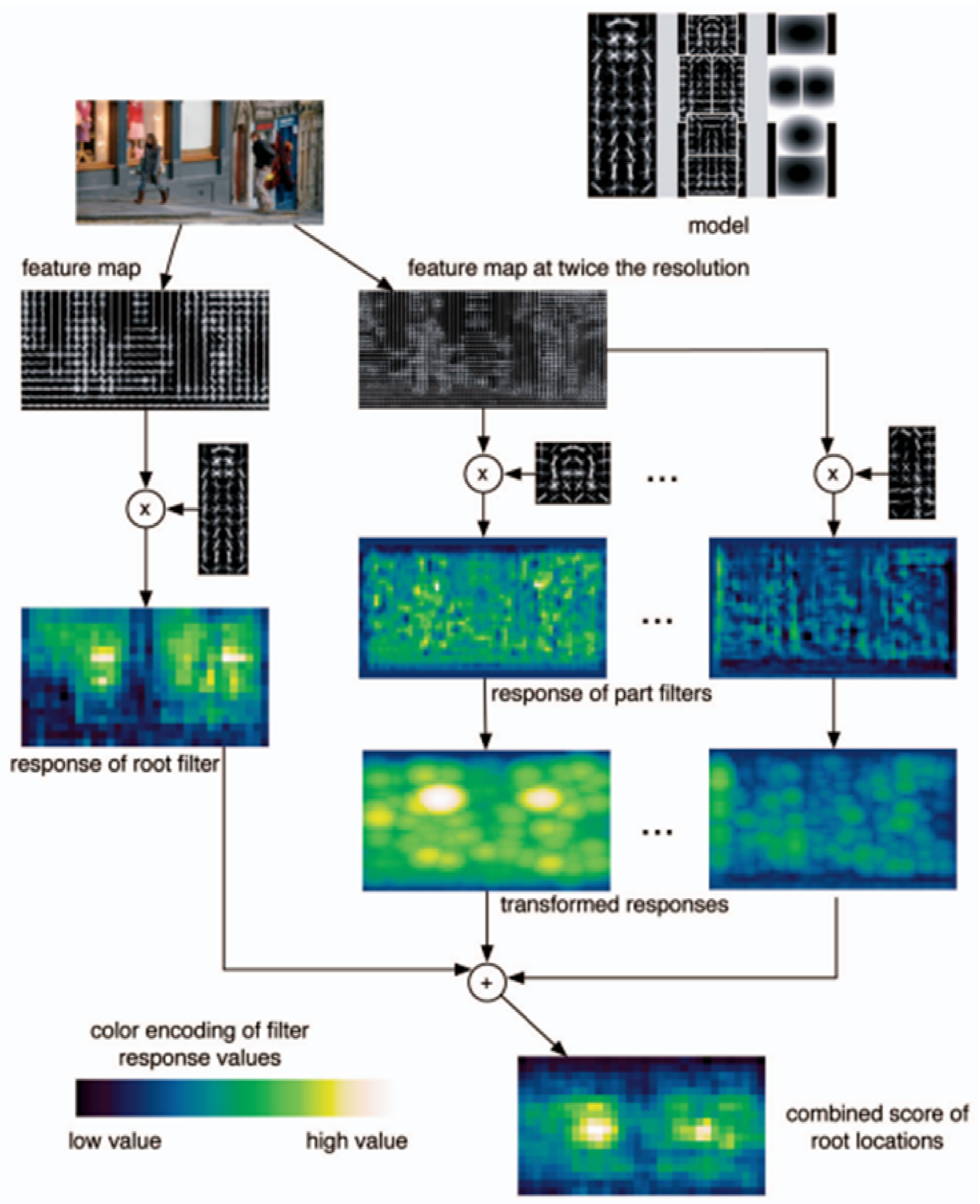


Figure 2.17: Pedestrian detection based on part-based model (Figure from [63]).

2.4 Catheter Segmentation and Tracking for Cardiac Interventions

2.4.1 Catheter Segmentation and Tracking in the X-ray Images

In the context of catheter tracking the topic of catheter tip detection has attracted a lot of attention since it has significant clinical value. The catheter tip has normally one or more electrodes and therefore has a very characteristic appearance in X-ray images. This enables a variety of feature detectors to be used. For example, the detection of the lasso catheter tip has been studied given the unique round shape of the catheter tip [34]. In this work the pose of the tip is reconstructed via ellipse fitting of its projection in a single biplane X-ray image. The catheter tip is then tracked through the remaining frames. For the tracking of coronary sinus (CS) catheter tips an electrode-based detector has been proposed in [125]. This detector first identifies blob-like structures based on an analysis of the Hessian matrix. The tip is identified by selecting a number of blobs with the lowest value of a pre-defined cost function. This work has been extended for the detection of lasso catheters and ablation catheters [124]. In [213] a probabilistic boosting tree (PBT) is used to train detectors for both the catheter tips and electrodes. Based on these results a number of hypotheses regarding the position of the tip in the next frame are generated. The hypothesis with the highest probability is selected based on a Bayesian formulation.

In a recent paper a multiple tip tracking algorithm has been proposed [142]. This approach is more robust to overlapping catheters and cases of occlusions. In this approach, blobs are first detected. Then outliers are removed based on spatial and geometric constraints. After that, the remaining blobs are clustered together. Then the blob detection near each detected blobs is repeated with a lower threshold. The purpose of this is to recover those blobs which were missed during the first blob detection. Finally, a number of hypotheses are generated based on these measurements and a greedy algorithm is used to score each hypothesis. The detection result is estimated by applying the hypothesis with the highest score. As a result, the selected blob clusters can be considered as the catheter tips and the corresponding blobs represent the electrodes on the tips. Fig. 2.18 shows an example of the detection results of three catheter tips, marked with different colors.

Most of the earlier approaches that focus on catheter tracking in X-ray images are specific to the catheter type. Therefore these approaches cannot be easily extended to other types of catheters. In addition, only salient features such as the tip or electrodes are used in these approachs. There is a small number of approaches that focus on the tracking of the whole catheter. Nearly all of these approaches focus on guide-wire tracking. In [205], a probabilistic

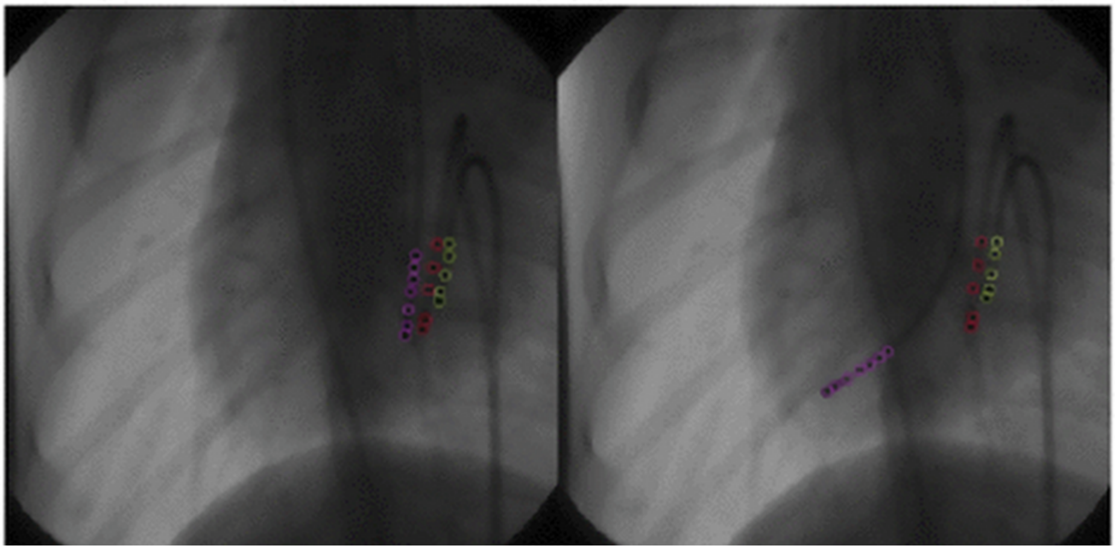


Figure 2.18: Catheter tip detection based on the method in [142]) (Figure from [142]).

framework based on the combination of learning-based and appearance-based features, as well as a hierarchical and multi-resolution tracking strategy has been proposed. In this work a probabilistic boosting tree (PBT) is trained for detection. The hierarchical tracking approach includes rigid and non-rigid tracking stages, corresponding to a coarse-to-fine process. In [162] the guide-wire tracking problem is addressed using a discrete optimization strategy in which the energy function is minimized. The energy function consists of a data term providing image intensity constraints and a regularization term that preserves the length of the curve. The data term is constructed using support vector regression in order to learn the relationship between the tracking error and intensity-based features. In later work [89], another discrete Markov Random Field (MRF) optimization was proposed for hybrid tracking. A combination of both a tubular structure detector and a salient landmark detector is used to enhance the robustness and the accuracy.

One typical representation of the guide-wire tracking based on discrete optimization was proposed in [86]. In this work a discrete optimization framework using MRFs and fast Primal-Dual (fast-PD) optimization is developed to track catheters. The catheter is modeled as a B-spline curve, which is a parametric representation of a non-linear curve. The location and the shape of a B-spline curve are determined by the labels (corresponding to displacements) of each control point that minimize an energy function. In Fig. 2.19 the control points are denoted by

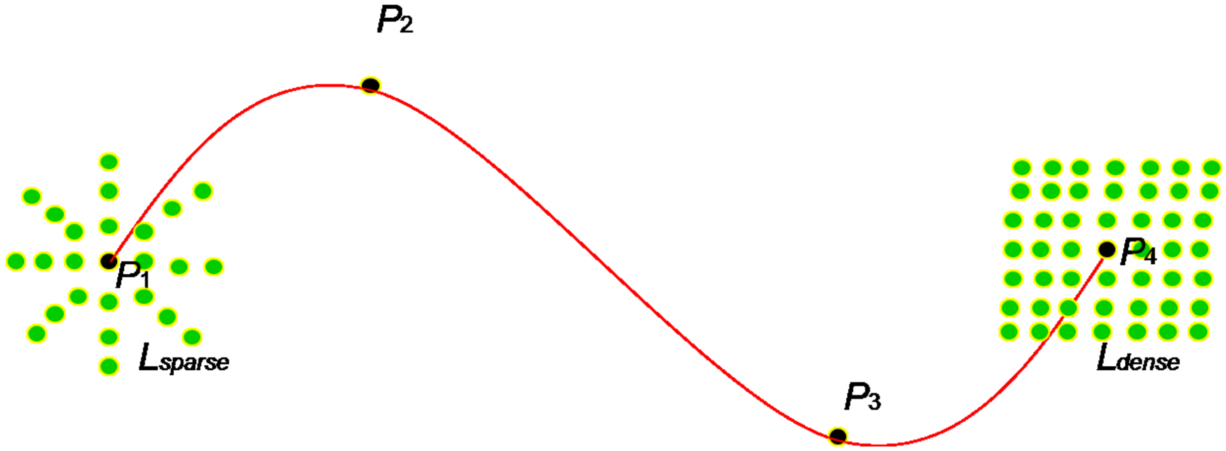


Figure 2.19: Illustration of the curve tracking based on discrete optimization on multiple label assignment.

$P_i, i = 1, \ldots, 4$. For each control point, the location in the next frame is searched within a local region. This region is the same as the label space since each local coordinate in the region corresponds to a label. Two types of label space, denoted by L_{sparse} and L_{dense}, are illustrated in the figure. The sparse label space is recommended when real-time performance is required while the dense label space is used when accuracy is more important. The corresponding energy function consists of only a pairwise term since an assumption that each position of the curve is determined by the nearest two control points is made. This pairwise term is a linear combination of a data term introducing image intensity information and a regularization term preserving the length.

In the recent literature there is a clear trend for guide-wire tracking techniques based on discrete optimization techniques [85]. These strategies are fast and can find a near-optimal solution with limited computational complexity. In [85] a number of different discrete optimization strategies are compared. Fast-PD is one of the recommended approaches. However, these discrete optimization approaches can only track a fixed portion of the catheter. This cannot compensate for large longitudinal motions of the catheter. The reason for this is that these optimization approaches assume a limited search range along the longitudinal direction for each control point of the B-Spline model. Large longitudinal motion is likely to cause the control points out of the search range.

2.4.2 Catheter Segmentation and Tracking in the Ultrasound Images

With the exception of studies focusing on tracking of the heart [219] or the valves of the heart [146] in ultrasound images, most previous work has used passive markers attached to surgical instruments [190, 147, 145] for tracking in ultrasound images. However, this is not suitable for catheter tracking given the small size of the catheter tip. Other previous approaches have tracked biopsy needles without additional markers [55, 226]. However, these algorithms are designed for rigid needles and cannot be easily extended to track long, non-rigid catheters.

Very few studies have directly addressed the problem of catheter segmentation and tracking in ultrasound. One recent study has targeted catheter segmentation in US [38] based on a pre-defined physical model. In the proposed method, the linear catheter tip is detected using on a Hough transform. This model focuses on the tip of the catheter and assumes a rigid nature of the catheter tip. Fig. 2.20 shows the catheter tip and its corresponding ultrasound image. The flexible part of the catheter is not tracked in their work. Although the final purpose of our catheter tracking is to localize the tips, it is important to first segment the whole catheter in order to constrain the tip location.

However, recently several methods that register the X-ray images to ultrasound images [73, 144, 92, 74, 87] have been developed. Thus, the information from the corresponding X-ray images can assist in detecting or segmenting structures in ultrasound.

2.5 Hardware Acceleration

Inherent parallelism exists in many applications of image processing, computer vision and medical imaging. In order to enhance the real-time performance, different hardware accelerators can be used to speed up the entire algorithm in addition to efficiency improvements in the algorithms itself. Currently, Graphics Processing Units (GPU) and Field-Programmable Gate Arrays (FPGA) are most widely used accelerators to realize the parallelism programming. More

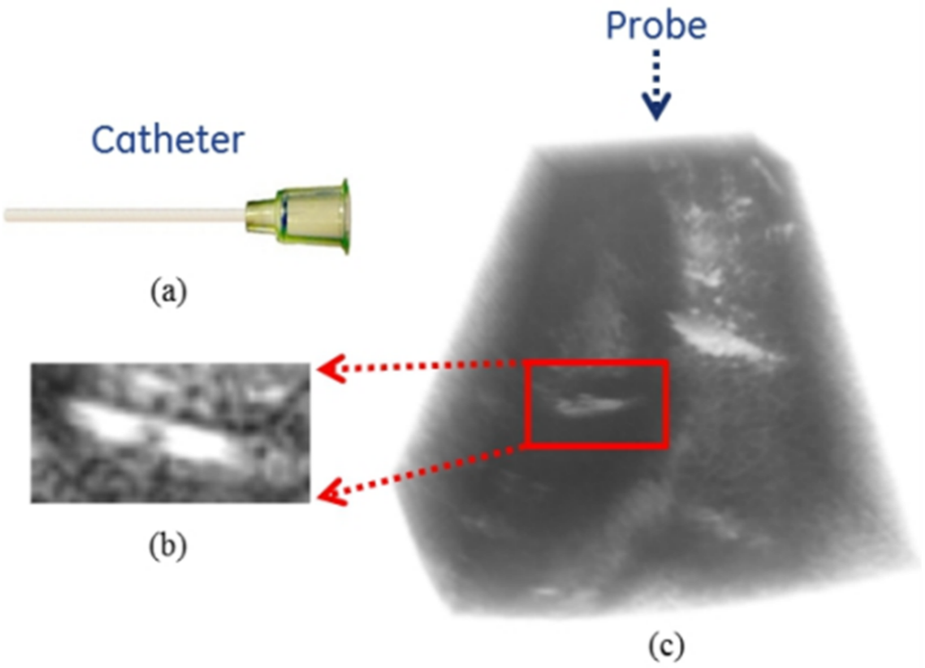


Figure 2.20: Catheter tip and its corresponding ultrasound image in [38] (Figure from [38]).

recently, many CPUs can also support multi-threading. In the following we will give a brief introduction to these techniques.

2.5.1 GPU Acceleration

The definition of Graphics Processing Units (GPU) was firstly proposed by Nvidia [152] as a single chip processor with integrated transformation, lighting, triangle setup/clipping, and rendering engines that is capable of processing a minimum of 10 million polygons per second [151]. Although the original purpose of the development of GPUs have been to manipulate computer graphics, GPUs have been widely used to accelerate many applications such as scientific computing, analytics, engineering, given the its efficient parallel computing ability. Compared with the traditional singe-core CPUs, which are capable to quickly process sequential serial tasks with a high frequency, GPUs have a large number of cores on the chip which can process in parallel multiple tasks simultaneously and efficiently [153]. The Single Instruction, Multiple Threads (SIMT) architecture is used for GPU parallel programming. Instructions are issued in serial while multiple threads can execute each instruction in parallel [149]. Nvidia has also released its own GPU programming interface, Compute Unified Device Architecture (CUDA) [150]. Besides CUDA, the Open Computing Language (OpenCL) is another interface for GPU programming aiming at cross-platform inter-operability [101].

2.5.2 FPGA Acceleration

A Field-Programmable Gate Array (FPGA) is a semiconductor device which can be programmed for customized applications based on a matrix of configurable logic blocks (CLBs). FPGAs allow reprogramming in case that the design evolves [217]. Due to its high flexibility in design, FPGAs have been widely used in a variety of application fields such as aerospace, automotive wired and wireless communications, audio, security, video and imaging [217]. Compared with the fixed microprocessor-based CPUs or GPUs, FPGAs are based on reconfigurable logic. This provides high flexibility for users in allocating hardware infrastructure resources and trading off the overall performance based on a particular application [41, 49]. Usually a typical FPGA consists of the following components [217]: (1) Configurable logic blocks (CLBs) which are the basic logic units and can perform elementary calculations for a single task independently; (2) basic select I/O that can support a number of I/O standards and organize these I/O operations in banks; (3) interconnects that link each CLBs to other CLBs, I/Os or global signals; (4) embedded block RAM memory and (5) digital clock management. VHDL and Verilog are common programming languages for FPGA design [1, 54].

2.5.3 CPU vs GPU vs FPGA

A number of comparisons have been performed between CPUs, GPUs and FPGAs based on a variety of applications [41, 49, 7, 50]. There is no general conclusion regarding the comparison of performance across these platforms since most of the evaluations are based on very specific applications. The best choice for hardware acceleration is highly dependent on the particular application. Generally, for a sequential algorithm with high dependency among different modules, CPUs are preferred given its high operational frequency and low data transfer latency from and to on-chip cache memory. Although currently most CPUs support multiple cores, their ability to perform computations in parallel is still not comparable to GPUs or FPGAs since the number of cores is limited. GPUs are an ideal choice for applications which are inherently parallelizable and the memory accesses are inter-dependent since it has a huge number of cores. However, fine-grained parallelism is difficult to realize with GPUs since the memory access is not efficient in GPUs. The size of local memory for each thread and the size of shared memory for each block are both limited. Data residing in local or shared memory cannot be cross-shared by the threads or blocks. On-chip global memory can solve this issue. However, it has a much longer latency than shared and local memory. In addition, control flow instructions will not be efficiently performed in GPUs. In comparison, FPGAs are a practical tool that support fine-grained parallelism with a low memory access latency. A high flexibility is provided to designers to customize the design and manage the tradeoffs for particular implementations. However, the lower operational frequency compared to both CPUs and GPUs should not be ignored [123, 41, 49, 7, 50].

Chapter 3

Fast Catheter Segmentation and Tracking in X-ray Fluoroscopic Images

3.1 Introduction

X-ray fluoroscopy is the standard imaging modality for cardiac catheterization procedures. Fast and reliable catheter segmentation and tracking enables better visualization and more accurate localization of the catheter. The fast segmentation of catheters can be used either for the initialization of the tracking or for the recovery from tracking failures. In the first frame of an X-ray sequence, there is little or no prior information available to support fully automatic segmentation. Therefore manual user interaction is required to specify which of the catheters is to be tracked. In this chapter we introduce a fast semi-automatic segmentation algorithm based on *Kalman filter-based path growing* and patch analysis. Compared to previous work which mainly focuses on catheter tips or other salient features, this strategy can segment a wide range of catheters with a high speed. This strategy is also used during the tracking stage which is also described in this chapter. When large longitudinal motion causes only a small proportion of the catheter to be trackable, this strategy can be used to extend the segmentation to the more difficult parts of the catheters.

For the tracking stage, a framework based on discrete optimization using the Fast Primal-

Dual (fast-PD) algorithm is proposed in order to solve the curve fitting problem efficiently. A graphical model is used to model the catheter as a smooth curve for the discrete optimization problem. For the tracking of individual catheters without interference from other catheters, we propose a new cost function that achieves robust and accurate tracking results. The cost function includes terms that not only preserve the appearance, but also preserve the length, curvature and distribution of points of the tracked catheter.

If there are multiple catheters within the same view, these can cause problems for tracking algorithms: For example, if two catheters are close to each other, there exists a possibility that the tracking may only detect one catheter. To address this, the relative distances between multiple catheters are added as soft constraints during the optimization. This can be achieved by creating dynamic edges in a graphical model. The relative distance between a pair of two control points is encoded as the weight of the corresponding dynamic edge in this model. These soft constraints are efficient clues to avoid the tracking failures caused by mutual catheter interference and occlusions.

3.2 Catheter Extraction in X-ray Images

The catheter extraction in the X-ray images is a key stage for the initialization of a tracking algorithm. Fig. 3.1 illustrates each stage of the proposed extraction method: Given one frame of the X-ray sequence (Fig. 3.1 (a)), the steps of the proposed algorithm are as follows: First, a SURF detector [16] is used to detect blob-like structures, i.e. the electrodes on the catheter. Blob-like structures (including real electrodes and outliers) detected by the SURF detector are highlighted in yellow in Fig. 3.1 (b). Starting from each detected blob, a Frangi vesselness filter is then used to determine whether a catheter is passing through the local patch. Fig. 3.1 (c) shows the results after apply this vesselness enhancement filter. If the patch is assumed to contain a catheter, a more detailed patch analysis is used to compute the principal direction of the catheter and to determine a potential landmark of the centerline in the next patch via a *Kalman filter-based path growing* framework. Fig. 3.1 (d$\sim$h) shows a patch located on a potentially relevant blob (d), edge points and their gradients being extracted within the patch (e), edge points classified into two different groups corresponding to the two sides of the catheter (f), edge points (after RANSAC outlier removal) used to calculate the path growing direction and to update the centerline (g), and the final result using patch analysis and Kalman filtering (h). In the following we will describe these steps in more detail.

3.2.1 Blob Detection and Tubular Structure Detection

Most catheters have electrodes on the tip, which appear as blob-like structures in the X-ray images. To detect these, the method of Speed Up Robust Features (SURF) [16] is used (Figure 3.1(b)). Meanwhile, the flexible shaft of the catheter requires a tubular structure detector. For this we adopt Frangi's vesselness filter [69] (Figure 3.1(c)).

Both the calculation of the SURF detector and Frangi's vesselness filter require the calculation of Hessian matrix, which is in form of:

$$\mathbf{H} = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} \tag{3.1}$$

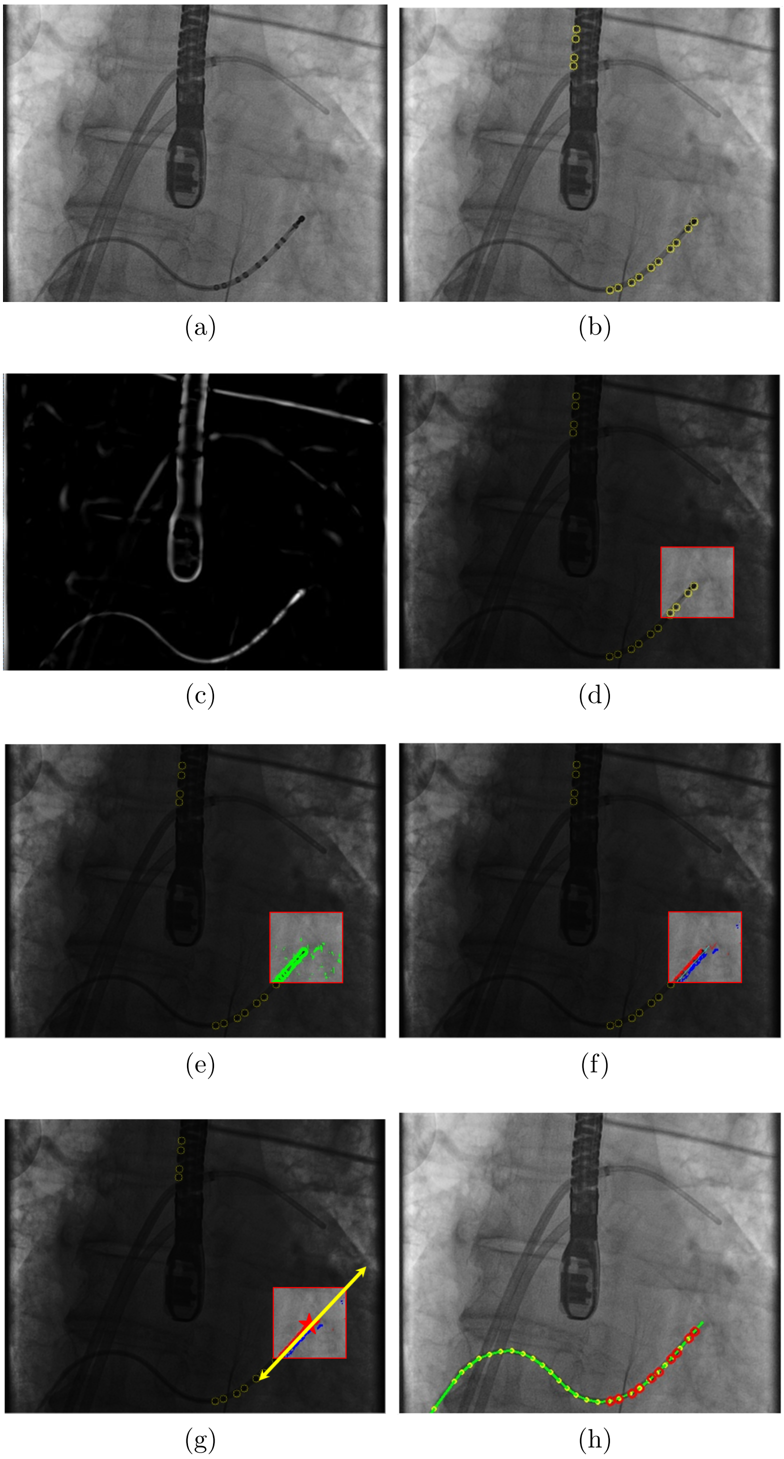
Here the elements of the Hessian, I_{xx}, I_{xy} and I_{yy}, are the second order partial derivatives of the image intensities along the two directions of the image plane x, y. Assume the eigenvalues of Hessian matrix are λ_1 and λ_2 with the order $|\lambda_1| < |\lambda_2|$. Then the response function of the SURF detector, R_{surf}, is defined as:

$$R_{surf} = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{3.2}$$

Note that D_{xx}, D_{xy} and D_{yy} are the approximated values of I_{xx}, I_{xy} and I_{yy} of the Hessian matrix computed using integral images.

The response function of Frangi's vesselness filter, R_{frangi}, is defined as:

$$R_{frangi} = \begin{cases} 0, & \text{if } \lambda_2 > 0 \\ \exp(-\frac{2\lambda_1^2}{\lambda_2^2})(1 - \exp(-\frac{\lambda_1^2+\lambda_2^2}{2c^2})), & \text{otherwise,} \end{cases} \tag{3.3}$$


(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 3.1: Catheter extraction in an X-ray image (see text for details).

Here c is normally set to half of the maximal value of $\sqrt{\lambda_1^2 + \lambda_2^2}$.

After the definition of the response functions for both blobs and tubular structures, τ_{surf} and τ_{frangi} are used to threshold the output of these response functions.

3.2.2 Patch Analysis

A patch is initially placed on each detected blob (Figure 3.1(d)) and later on the position determined by the Kalman filter, as detailed in the following section: Given a small square patch of the image, we determine whether there is a section of catheter inside using Frangi's vesselness filter. If the patch contains a catheter, a centerline is fitted to approximate the catheter segment. Information based on the fitted centerline is used for the *Kalman filter-based path growing* in the next step. However, Frangi's vesselness filter does not produce an accurate direction and position of the catheter fragments within the patch. This is because of performance reasons. To make the entire system as fast as possible, the Hessian matrix, on which Frangi's vesselness filter is based on, is calculated in an approximate fashion by using integral images instead of using the standard procedure proposed in [69]. However, the performance of Frangi's vesselness filter is very sensitive to the accuracy of the corresponding Hessian matrix. Thus, instead of calculating the direction and position by the vesselness filter, we analyse the statistical properties of the edge points and their corresponding intensity gradients. Our basic assumption is that the section of the catheter inside the local patch is straight. The patch analysis can be divided into edge point extraction, point classification, outlier removal and centreline fitting.

Edge point extraction

First, the border of the catheter in the patch is detected using Sobel edge features [60] based on two Haar filters along the x and y directions in the integral image. The filter responses of the two directions at a point i are collected into vector $\mathbf{d} = (d_{ix}, d_{iy})$. Then the magnitude R_i and the orientation θ_i of the vector $\mathbf{d}$ are calculated. By thresholding the magnitude of $\mathbf{d}$ with τ_{edge} and selecting the maximum response in a neighbourhood, a set of candidate edge points are identified: $\mathbf{e}_i = (R_i, \theta_i)$ (Figure 3.1(e)).

Point classification

Based on the candidate edge points, a histogram of (θ_i mod π) is constructed. We use mod π for the histogram so that parallel but opposite edges map to the same histogram bin. The maximum value in the histogram identifies the most likely direction of the catheter. Two angles are then defined from this maximum: θ^+ and θ^- so that $\theta^+ - \theta^- = \pi$. All edge points are then classified as catheter edges if they are within a certain range of these angles, usually $\pm\pi/6$. Edge points are therefore classified into three categories:

$$\mathbf{e}_i \in \begin{cases} \mathbf{E}^+, & \text{if } \theta_i \in [\theta^+ - \pi/6, \theta^+ + \pi/6] \bmod 2\pi \\ \mathbf{E}^-, & \text{if } \theta_i \in [\theta^- - \pi/6, \theta^- + \pi/6] \bmod 2\pi \\ \mathbf{E^0}, & \text{otherwise.} \end{cases} \tag{3.4}$$

Here $\mathbf{E}^+$ and $\mathbf{E}^-$ represent opposite edges of the catheter (Figure 3.1(f)) and $\mathbf{E^0}$ contains all other edge points.

Before calculating θ^+ and θ^-, a mask is used to filter the histogram and set bins to zero if these correspond to directions identified as 'unlikely' via prior knowledge (if prior knowledge exists). Patches are analysed one by one in the order indicated by the centreline direction. The prior knowledge can be the output direction from the last patch analysis in an adjacent patch. The purpose of this is to ensure a consistent direction with the direction of the centreline in the previous patch. This adds robustness for patches which contain two or more catheters intersecting each other.

Outlier removal

Because of the assumption that the section of the catheter inside the local patch is straight, the edge points can be modelled using line equations. Thus each set $\mathbf{E}^+$ and $\mathbf{E}^-$ can be modelled with a line equation corresponding to one edge of the catheter within the local patch. Assuming that the proportion of outlier points is below 50%, Random Sample Consensus (RANSAC) [67] is a good choice to remove these outliers. In our application, two points (the minimal required quantity to model a line equation) are sampled and a line is fitted through them. The distances between this line and the other points are subsequently calculated. If the number of points in the vicinity of the line is above a threshold, the line is accepted and the supporting points are retained while the others are removed as outliers. Outliers are moved from sets $\mathbf{E}^{\pm}$ to set $\mathbf{E^0}$.

Centerline fitting

After outliers are eliminated from the two sets, a centreline equation is determined using $\mathbf{E}^{\pm}$ (Figure 3.1(g)). This centreline equation encodes direction and position information and can be written as $ax + by + c = 0$. However, points from $\mathbf{E}^{\pm}$ are not exactly located on this centreline but located on two borders of the catheter in form of

$$ax + by + c \pm d = 0. \tag{3.5}$$

Here d denotes the offset of the border lines of the catheter to the centreline. Given the 2D coordinates $(x_i^{\pm}, y_i^{\pm})^T$ of the points from $\mathbf{E}^{\pm}$, the coordinates can be rewritten as $\mathbf{v}_i = (x_i^{\pm}, y_i^{\pm}, 1, \pm 1)^T$. Finding the centreline can then be easily transformed to a linear optimization problem. We then find the solution $\mathbf{p} = (a, b, c, d)^T$ that satisfies

$$\min_{\mathbf{p}} \|\mathbf{Ap}\|_2, \text{ subject to } \|\mathbf{p}\|_2 = 1. \tag{3.6}$$

Here $\mathbf{A} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n^+ + n^-})^T$ and $n^{\pm}$ is the total number of points in set $\mathbf{E}^{\pm}$. This optimization problem can be solved using Singular Value Decomposition (SVD). After the parameters of the centreline are determined, the direction θ_p is denoted by:

$$\theta_p = \begin{cases} \arctan(-a/b) \bmod \pi, & \text{if } b \neq 0 \\ \pi/2 & \text{if } b = 0. \end{cases} \tag{3.7}$$

The intersection $\mathbf{x}_p = (x_p, y_p)^T$ of the centreline and the perpendicular line through the centre of the patch, which is the new updated position, is given by:

$$\mathbf{x}_p = \begin{pmatrix} b & -a \\ a & b \end{pmatrix}^{-1} \begin{pmatrix} bx_c - ay_c \\ -c \end{pmatrix}. \tag{3.8}$$

Here $\mathbf{x}_c = (x_c, y_c)^T$ is the centre of the patch.

Fitting of multiple line equations

In a clinical scenario, multiple catheters may be present in one view simultaneously. Furthermore it is likely that two or more catheters are quite close to each other or intersect each other. Even if only one catheter is present, it may have sharp turns and in the corresponding local patches the line fitting may fail. In such cases, multiple line equations may be required simultaneously within one patch. A possible solution is to repeat the above procedure iteratively: In each iteration we perform the above line fitting and the corresponding supporting edge points $\mathbf{E}^+$ and $\mathbf{E}^-$ are then removed from the histogram. Then, a different line equation is fitted in the next iteration. This procedure is repeated until a stopping criteria is reached: The stopping criteria depends on the numbers of edge points in $\mathbf{E}^+$ and $\mathbf{E}^-$ and on the number of iterations. The borders of both sides of of the catheter should be equally clear. In other words, the edge points supporting the two borders of the catheter (corresponding to sets $\mathbf{E}^\pm$ respectively) should have approximately equal size. Therefore, if the ratio of the numbers of points $\mathbf{E}^+$ and $\mathbf{E}^-$ is approximatly 1 ($> \tau_{multi}$ in our application) and the sum of these two numbers is above a threshold n_{multi}, the line equation is accepted. If the fitted line is accepted and the number of the iteration is below 3, then the loop continues; otherwise it stops.

Parallel lines within one patch

Sometimes multiple catheters are both close to each other and have the same directions. As a result there are several parallel lines within the same local patch. The above strategy cannot

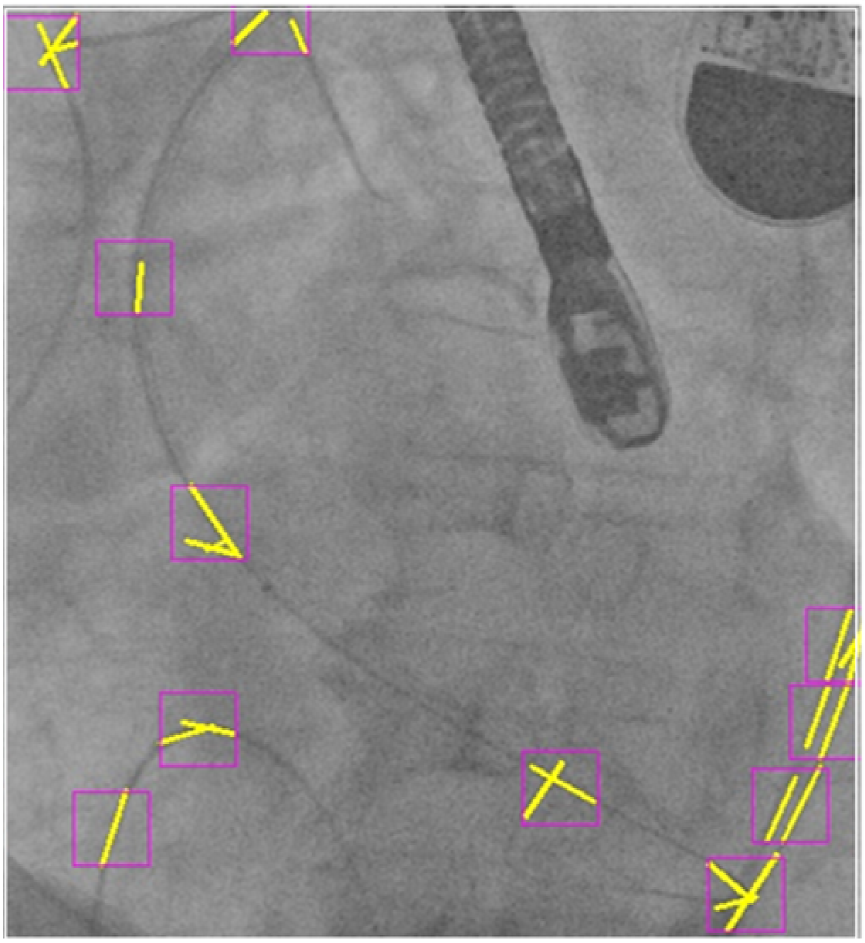


Figure 3.2: An example of multi-line fitting patch analysis with multiple line segments detected in yellow.

work in this case. The reason is that the supporting edge points for parallel catheters are within the same bin in the orientation histogram. Thus they need to be separated first in order to identify the edges of different catheters. Given a direction θ^+, the 2D spatial positions of all points in $\mathbf{E}^+$ and $\mathbf{E}^-$ are projected onto the axis that corresponds to the direction of θ^+. Then the distribution of the projected points on this axis serves a clue for further clustering: points clustered together are considered to belong to one catheter, and points with a large gap on the axis are considered to belong to different clusters or catheters. We show an example of one image containing patches with multiple catheters in Fig. 3.2.

3.2.3 Kalman Filter-based Path Growing

Given a set of initial candidate blobs detected using SURF, the catheter can be extracted by expanding it from each blob to form the whole catheter. At each step of the expansion or growth, the centreline fitting via local patch analysis is carried out. The measurements from the previous step (direction and updated position of the catheter centreline) are employed to locate the next patch. If a patch yields no fitted line or the border of the image has been reached, the process is terminated (Figure 3.1(h)). This process can be formulated within a Kalman filtering framework. The system model of the Kalman filter is defined as:

$$\begin{cases} x_{k+1} = x_k + \triangle t(v(\omega_k \cos\theta_k)), \\ y_{k+1} = y_k + \triangle t(v(\omega_k \sin\theta_k)), \\ \theta_{k+1} = \theta_k + \triangle t\omega_k, \\ \omega_{k+1} = \omega_k + \triangle t\sigma_k. \end{cases} \tag{3.9}$$

The state vector is given by $\mathbf{x}_k = (x_k, y_k, \theta_k, \omega_k)^T$. Here x_k and y_k denote positions of the patch centres while θ_k and ω_k denote angular displacement and velocity at k-th time respectively. The state vector is updated at every step as well as its corresponding covariance matrix $\mathbf{P}_k$. $\triangle t$ is the sampling time and $\sigma_k \sim N(0, \sigma^2)$. We use v to denote the instantaneous velocity, which is then defined as a function of angular velocity ω_k,

$$v(\omega_k) = (v_{max} - v_{min}) \exp(-\alpha\omega_k^2) + v_{min}. \tag{3.10}$$

$v(\omega_k)$ should be neither too large nor too small, and within $[v_{min}, v_{max}]$. α (usually set to 1) denotes how sensitive the velocity is in response to a change of ω_k. This means that the step size for the catheter growth is proportional to the size of the instantaneous angular velocity. As a result, landmarks are sampled more densely at sharper turns of the catheter and more sparsely on smoother sections. The elements of the state vector θ_k and $\mathbf{x}_k = (x_k, y_k)^T$ are directly measured by the values of θ_p and $\mathbf{x}_p = (x_p, y_p)^T$ obtained from the patch analysis.

The system and measurement equations of the Kalman filter can then be written as follows:

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \sigma_k, \tag{3.11}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{V}_k, \tag{3.12}$$

with the system matrix

$$\mathbf{A}_k = \begin{pmatrix} 1 & 0 & -\sin\theta_k(S_k + \triangle t v_{min}) & -2\alpha\omega_k \cos\theta_k S_k \\ 0 & 1 & \cos\theta_k(S_k + \triangle t v_{min}) & -2\alpha\omega_k \sin\theta_k S_k \\ 0 & 0 & 1 & \triangle t \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.13}$$

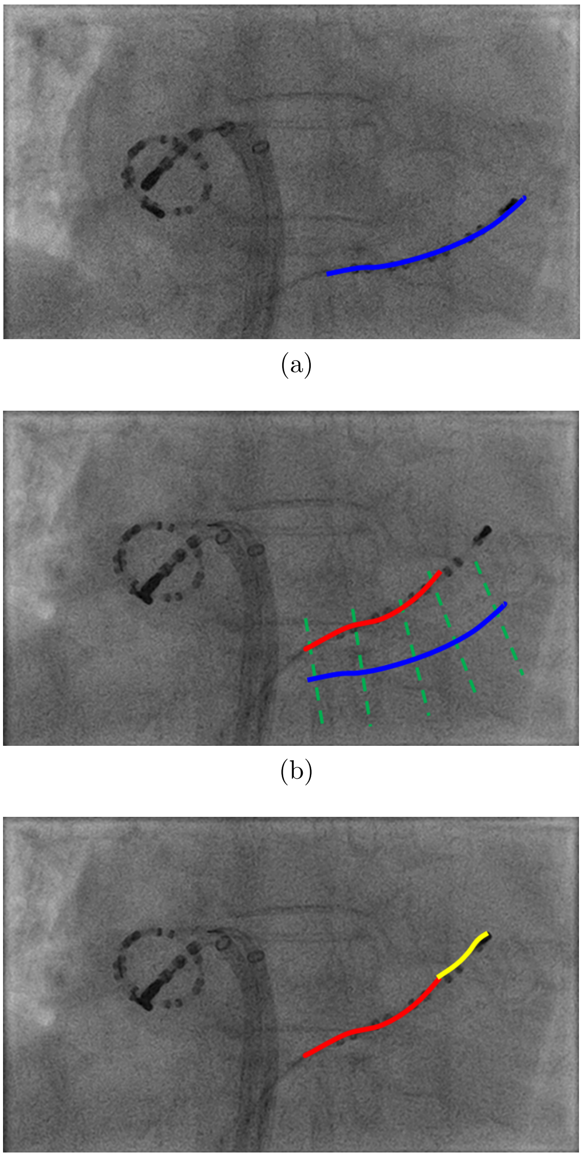
and measurement matrix

$$\mathbf{H}_k = [\mathbf{I}_3 | \mathbf{0}]. \tag{3.14}$$

Here $S_k = \triangle t(v_{max} - v_{min}) \exp(-\alpha\omega_k^2)$ and $\mathbf{B}_k = (0, 0, 0, \triangle t)^T$. We assume that the measurement uncertainties for the position and the direction are within 5 pixels and around 15°, respectively. Based on this assumption we can write $\mathbf{V}_k$ as a diagonal matrix with entries $(5, 5, 0.25)$ along its diagnonal. The coordinates and the direction obtained from the initial patch analysis are used to initialize the first three states in the state vector x_0, y_0 and θ_0. ω_0 is set to zero. We also make another assumption that the initial uncertainty of the position is within 5 pixels and the uncertainty of both the angular displacement and the angular velocity is within 15°. Then we can define the corresponding covariance matrix $\mathbf{P}_0$ as diagonal matrix with diagonal entries $(5, 5, 0.25, 0.25)$. By applying the extended Kalman filter, a sequence of sorted displacements $(x_k, y_k)^T$ representing the potential catheter can be generated.

3.2.4 Catheter Extraction Applied to Tracking in X-ray Sequences

Normally the catheter extraction in each frame independently suffers from instability because it does not take the temporal consistency between consecutive frames into account. In order to make the segmentation in subsequent frames more robust and consistent, information from previous frames can be used for the catheter extraction. This transforms the segmentation problem into a tracking problem. A method for catheter tracking in X-ray images is presented in the following. In our application, the main requirement is to track the catheter as completely as possible. To achieve this, the algorithm described in this section first uses the previously proposed fast-PD method to track movements transversal to the catheter. This alone is sufficient when the length of the catheter in the field-of-view is not changing. Secondly, a *Kalman filter-based path growing* method, the same as proposed in the last section, is used to extract parts of the catheter that are obscured. This strategy ensures that the entire catheter is always tracked as much as possible.



(a)

(b)

(c)

Figure 3.3: Catheter tracking in X-ray images .

Fast-PD Catheter Tracking

Fast-PD [107] is a fast discrete multi-label approach for the efficient optimization of Markov Random Fields (MRF). In our application, the catheter is defined by a set of key points and can be considered as a graph in which two consecutive points are linked by an edge. Each point defines a search region in the next frame and each potential location of the catheter in the search region corresponds to a discrete label. Fig. 3.3 (a) shows an example of an X-ray frame with the result of the catheter extraction marked in blue. Fig. 3.3 (b) shows the next frame with the projection of the result from the previous frame in blue and the new result estimated via fast-PD optimization in red. The dotted lines in green represent the search range of the optimization. All of the pixels in the search region of any key point define a label space L. During the tracking, the energy function that is minimised using fast-PD is defined as

$$f(\mathbf{x}) = \sum_{i=1}^{n}(u_i(x_i)) + \gamma \sum_{i=1}^{n-1}(d(x_i, x_{i+1})). \tag{3.15}$$

Here $\mathbf{x} = \{x_1, x_2, \cdots, x_n\}$ are the possible positions in the search region for each point. Each position corresponds to a label assignment. The unary term $u_i(x_i)$ and the pairwise term $d(x_i, x_{i+1})$ are defined as:

$$u_i(a) = \begin{cases} 0, & \text{if } Patch(a) = 1 \\ 1 - S(a), & \text{if } Patch(a) = 0, \end{cases} \tag{3.16}$$

$$d(a, b) = \sqrt{((x_a - x_b)^2 + (y_a - y_b)^2)}. \tag{3.17}$$

The unary term enforces the constraints from the image itself and the pairwise term enforces the constraints from the model, e.g., the smoothness of the curve. Given $a, b \in L$, the unary term depends on the value of $Patch(a)$, which checks if there is at least one catheter passing through the local patch. This can be done by via the patch analysis introduced previously. If there is no catheter, i.e. $Patch(a) = 0$, the unary term is defined by a similarity function $S(\cdot)$, in our case the normalized cross-correlation between the previous and current image. The pairwise term $d(a, b)$ is the planar distance between two locations in the search area corresponding to the two labels a, b. γ is a parameter that balances the unary and pairwise terms and is chosen manually. The result is a segmentation of the catheter in the new frame accounting for motion that is transversal to the catheter (see Figure 3.3(b)). However, sometimes some parts of the catheter are outside of the search range, which can lead to an incomplete extraction of the catheter. Thus we use the *Kalman filter-based path growing* again to recover the undetected parts of the catheter.

Kalman filter-based path growing for Tracking

A large search region for the fast-PD optimization would significantly slow down the tracking algorithm. In our application, we have chosen to ignore the search range in the longitudinal direction and retain that of the transversal direction because the transversal direction is more important. However, the trade off is that a large amount of motion in the longitudinal direction of the catheter could cause the catheter to move out of the search range of the fast-PD optimization. To address this, *Kalman filter-based path growing* is introduced in Section 3.2.3 to compensate for this.

The fast-PD optimization yields a sequence of sorted points corresponding to potential catheter locations for the next frame. Starting from the two end points of the catheter, each point is checked for whether there is a catheter passing through it using the function $Patch(x)$. For each point with $Patch(x) = 0$ the point is removed and the next adjacent point is checked; for each point with $Patch(x) = 1$ the point is used as the new end point and the process is terminated. As a result, the catheter may be shortened but will have reliable points $(Patch(x) = 1)$ as end points. The *Kalman filter-based path growing* is then used to detect new sections, starting from the end points, which allows for the recovery of the entire catheter, even if there is a larger proportion of the catheter in the current frame than in the previous frame. Figure 3.3(c) shows that an untracked part of the catheter in yellow has been recovered by the *Kalman filter-based path growing*.

3.3 Catheter Tracking in X-ray Images

The previous section of this chapter describes a fast catheter extraction algorithm which can be used to segment catheters at each individual frame. The extraction algorithm can be performed at different frames independently. From now on we discuss the catheter tracking framework, in which catheter can be segmented from the current frame based on the information from previous frames.

3.3.1 Image Enhancement

In each X-ray frame, several measurements are computed based on the pre-processing of the original X-ray image I. Four types of measurements in form of an enhanced image calculated from I are obtained: tubular structures I_{tub}, blob-like structures I_{blob}, orientations I_{ori} for tubular structures and enhanced catheter tips I_{tip}. The intensity values of all derived images are normalized within the range $[0, 1]$. I_{tub} and I_{blob} are calculated via the Hessian matrix at one or a limited number of scales. The range of scales is pre-defined and the scale is controlled by a Gaussian kernel with standard deviation σ_{blur}. More precisely, a smoothed image is computed as $I_{blur} = I * G(0, \sigma_{blur})$ where $G(\mu, \sigma)$ is a Gaussian kernel with mean μ and standard deviation σ. Then the Hessian matrix $\mathbf{H}$ is defined as

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 I_{blur}}{\partial x^2} & \frac{\partial^2 I_{blur}}{\partial x \partial y} \\ \frac{\partial^2 I_{blur}}{\partial x \partial y} & \frac{\partial^2 I_{blur}}{\partial y^2} \end{pmatrix} \tag{3.18}$$

Here the elements of the Hessian, the second order partial derivatives, can be approximated by finite differences. The catheter tip enhancement map I_{tip} is calculated using Frangi's vesselness filter [69]:

$$I_{tip} = \begin{cases} 0, & \text{if } \lambda_2 > 0 \\ \exp(-\frac{2\lambda_1^2}{\lambda_2^2})(1 - \exp(-\frac{\lambda_1^2+\lambda_2^2}{2c^2})), & \text{otherwise,} \end{cases} \tag{3.19}$$

with two eigenvalues λ_1 and λ_2 satisfying $|\lambda_1| < |\lambda_2|$. c is normally set to half the maximal value of $\sqrt{\lambda_1^2 + \lambda_2^2}$. Meanwhile, an orientation map I_{ori} is calculated from the normalized eigenvector $\mathbf{v}_2$ of the Hessian matrix,

$$I_{ori} = \varphi_{ori}. \tag{3.20}$$

Here φ_{ori} is calculated from the eigenvector $\mathbf{v}_2 = (\cos(\varphi_{ori}), \sin(\varphi_{ori}))$. The blob map I_{blob} can be calculated by the determinant of the Hessian matrix as proposed in [16]:

$$I_{blob} = \det \mathbf{H} = \lambda_1 \lambda_2. \tag{3.21}$$

Using I_{tub} and I_{ori}, a steerable filter can be designed to enhance tips of the tubular structures at point $\mathbf{p}$:

$$I_{tip}(\mathbf{p}) = |\frac{\sum_{\triangle p_x, \triangle p_y \in \Omega} I_{tub}(\mathbf{p}^{'}) G(\triangle p_x, \sigma_{tip}) sign(p_y)}{\sum_{\triangle p_x, \triangle p_y \in \Omega} G(\triangle p_x, \sigma_{tip}),}| \tag{3.22}$$

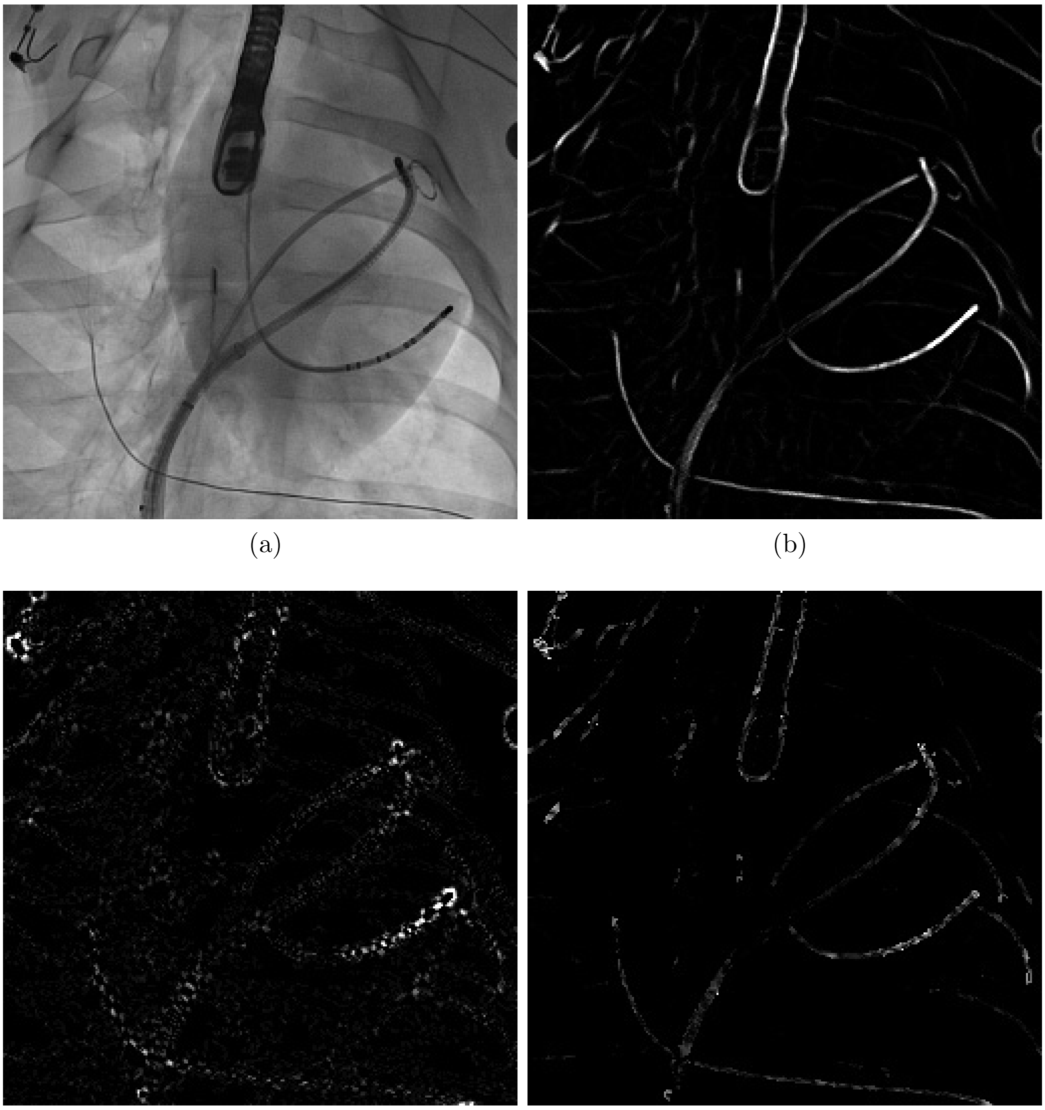
with

$$\mathbf{p}^{'} = \mathbf{p} + [\mathbf{v}_x, \mathbf{v}_y] \triangle \mathbf{p}. \tag{3.23}$$

Here $\Omega = [-5, 5]$ defines the size of the patch. The x-axis of this patch($\mathbf{v}_x$) is set to align with the angular direction φ_{ori} from I_{ori}. The y-axis $\mathbf{v}_y$ is perpendicular to $\mathbf{v}_x$. The local coordinates in the patch are represented by $\triangle \mathbf{p} = [\triangle p_x, \triangle p_y]^{'}$. As before $G(0, \sigma_{tip})$ is a Gaussian kernel with zero mean and variance $\sigma_{tip} = 5$. $sign(*)$ is the sign function which returns 1, 0 or -1 in case of positive, zero and negative arguments respectively. Fig. 3.4 shows examples of the different pre-processed images.

3.3.2 Overview of the Discrete Optimization Framework

The catheter tracking problem is transformed into a discrete optimization problem using a Markov Random Field (MRF) formulation. The MRF in our application is in form of a graph model $G =< V, E >$ which is related to a B-spline representation of the catheters or the contour of the ultrasound probe. The nodes in this graph correspond to the control points of the B-splines. For each B-spline, the control points at two ends are specified and belong to a



(a)

(b)

(c)

(d)

Figure 3.4: Examples for an original image (a), its corresponding tubular map (b), blob map (c), and tip map (d).

terminal node set V_c. All other control points belong to the set V_n. V_c and V_n comprise the complete node set V of the graphical model. The edges E link pairs of nodes. These edges consist of two types of edges that depend on the relationship of the nodes. If two nodes are adjacent control points of the B-spline, then an edge is created and this edge is added to the set E_s. The relative distance of two nodes is important especially when these two nodes are close to each other and they belong to two parallel catheters or two parts of the same catheter. In these cases the relative distances between the nodes should be preserved, whether they belong to the same B-spline or not. Thus, a dynamic edge is added. This type of edges is denoted by

E_r. E_s and E_r comprise the complete edge set E of the graphical model. The term 'dynamic' is used to indicated that the edge set E_r is changing from frame to frame, depending on the distribution of the control points. If a node in V_c is connected by only one edge of E_s and its position is far not on the image border, then this node is identified as a catheter tip and added to a subset $V_t \subseteq V_c$. In some cases, one catheter is represented by more than one B-spline curve fragment and they are linked together to represent the whole catheter.

For each catheter, catheter fragment, or the contour of the ultrasound probe, $i = 0, \dots, m-1$, a B-spline curve is denoted by

$$\mathbf{S}_i(t) = \sum_{j=0}^{n_i-1} \mathbf{c}_{i,j} B_{i,j}(t), \ t \in [t_{i,min}, t_{i,max}]. \tag{3.24}$$

Here the control points $\mathbf{c}_{i,j}, i = 0, \dots, n_i - 1$ belonging to V, with $\mathbf{c}_{i,0}$ and $\mathbf{c}_{i,n_i-1}$ belong to V_c. For each node $\mathbf{c}_{i,j}$, the updated location is obtained by a local movement offset $\triangle\mathbf{c}_{i,j}$ within a 2D search range $\{-R_{range} \leq x \leq R_{range}, -R_{range} \leq y \leq R_{range}\}$. We assume that each pixel in this search range corresponds to one label. Thus obtaining the correct movement offset for each control point is equivalent to assigning the correct label to this point. Therefore, the problem of curve tracking is transformed to a multi-label assignment problem, which can be solved via discrete optimization. The label space comprises all pixels in the search range. The optimization objective function is defined as

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) = (1-\gamma)\sum_{p \in V}(u_p(x_p)) + \gamma \sum_{p,q \in E}(w_{p,q}(x_p, x_q)). \tag{3.25}$$

Here $\mathbf{x}$ denotes all labels assigned to each node p in V. x_p denotes the location of the control point p. The objective function consists of an unary term $u_p(x_p)$ and a pairwise term $w_{p,q}(x_p, x_q)$ linked with a parameter γ. In the following sections, details about the objective function are presented.

3.3.3 Objective Function

The unary term $u_p(x_p)$ in (3.25) in our application consists of only a single unary data term that attracts x_p to the tip of the catheters. The pairwise term $w_{p,q}(x_p, x_q)$ in the objective function in (3.25) is a combination of several terms, including a pairwise term $w^0_{p,q}$, a length continuity preserving term $w^1_{p,q}$, a landmark distribution term $w^2_{p,q}$, a shape continuity term $w^3_{p,q}$, and a relative distance preservation term $w^4_{p,q}$. The pairwise terms can be summarised as

$$w_{p,q}(x_p, x_q) = \sum_{u=0}^{4} \alpha_u w^u_{p,q}(x_p, x_q), \tag{3.26}$$

with $\sum_{u=0}^{4} \alpha_u = 1$. The coefficients $\alpha_u, u = 0 \cdots 4$ are heuristically determined.

Unary Data Term

The B-spline used in our application is a clamped B-spline, which interpolates its interval control points with two fixed end control points. The unary data term is only defined for nodes in V_t. The purpose of this term is to attract each node in V_t to the location with a high intensity value in the catheter tip map I_{tip}:

$$u_p(x_p) \quad = \quad \begin{cases} (1 - I_{tip}(x_p)), & \text{if } p \in V_t \\ 0, & \text{otherwise.} \end{cases} \tag{3.27}$$

Pairwise Data Term

The main task of the pairwise data term is to locate the catheter through pixels with high intensity values in the maps I_{blur}, I_{tub} and I_{blob}. The main difference to the unary data term is that the pairwise term focuses on the whole catheter rather than only on the catheter tips. Different types of similarity measurements are used for I_{tub}, I_{blob} and I_{blur}. I_{tub} detects general tubular structures and I_{blob} detects specific blob-like electrodes. I_{blur} measures the original intensities after smoothing. The pairwise data term is only defined for two adjacent control points of the same B-spline. The nodes p and q can also be denoted by the j-th and $j + 1$–th control points of the B-spline. We make the following assumption: If two consecutive control points j and $j+1$'s labels are x_p and x_q, then any other control point k of the same B-spline is labelled as $x_p, k < j$ and $x_q, k > j$. The reason for this is to make other control points' movements consistent with these two consecutive control points. The motion offsets corresponding to labels x_p and x_q is denoted by $\triangle \mathbf{c}_{i,j}$ and $\triangle \mathbf{c}_{i,j+1}$, respectively. Thus the resulting B-spline is defined by

$$\hat{\mathbf{S}}_i(t) = \mathbf{S}_i(t) + \sum_{k=0}^{j} \triangle \mathbf{c}_{i,j} B_{i,k}(t) + \sum_{k=j+1}^{n_i-1} \triangle \mathbf{c}_{i,j+1} B_{i,k}(t). \tag{3.28}$$

The catheter observations, or *landmarks*, are sampled densely in the range $[t_{i,min}, t_{i,max}]$ with a step size $\triangle t = 5$. The k-th landmark corresponds to $t_k = \min(t_{i,min} + k\triangle t, t_{i,max})$ with $k = 1, \ldots, m_i$ where $m_i = \lceil (t_{i,max} - t_{i,min})/\triangle t \rceil$. Finally the pairwise data term is defined as

$$w^0_{p,q}(x_p, x_q) = \frac{\sum_{k=0}^{m_i} N^i_{j,j+1}(t_k)(1 - I_m(\hat{\mathbf{S}}_i(t_k)))}{\sum_{k=0}^{m_i} N^i_{j,j+1}(t_k)} \tag{3.29}$$

with

$$I_m(\hat{\mathbf{S}}_i(t_k)) = \beta_0 I_{blur}(\hat{\mathbf{S}}_i(t_k)) + \beta_1 I_{tub}(\hat{\mathbf{S}}_i(t_k)) + \beta_2 I_{blob}(\hat{\mathbf{S}}_i(t_k)), \tag{3.30}$$

where $\beta_0 + \beta_1 + \beta_2 = 1$. $N^i_{j,j+1}(t_k)$ is a weight parameter. Normally a large weight is allocated to a landmark which is near to the corresponding control point. The weight can be calculated by:

$$N^i_{j,j+1}(t) = \frac{B_{i,j}(t) + B_{i,j+1}(t)}{\sum_{k=0}^{n_i-2}(B_{i,k}(t) + B_{i,k+1}(t)).} \tag{3.31}$$

Pairwise Length Continuity Term

Using only intensity information causes the tracking to be sensitive to noise and outliers. This can be addressed by incorporating prior knowledge into the tracking. Such prior information can be the length, the landmark distribution, and the shape of the catheter, or the relative distance between different catheters. One problem that occurs frequently is that the tracked curve can collapse into a single point if no regularization constraints are applied. To address this problem, previous work [86] introduced a length continuity preserving regularization term.

The length of the B-spline curve $\hat{\mathbf{S}}_i(t)$ is calculated by

$$l(t) = \int_{t_{i,min}}^{t} \|\hat{\mathbf{S}}_i'(\tau)\| d\tau,\ t \in [t_{i,min}, t_{i,max}]. \tag{3.32}$$

Here $\hat{\mathbf{S}}_i'(\tau)$ denotes the first derivative of $\hat{\mathbf{S}}_i(\tau)$. Note that this derivative is in vector form. The length continuity preserving regularization term is defined by ensuring that the 2-norm of the derivative vector at each landmark is equal to that of the B-spline in the previous frame. This term is defined for adjacent control points which have an edge that belongs to E_s:

$$w_{p,q}^1(x_p, x_q) = \frac{\sum_{k=0}^{m_i} N_{j,j+1}^i(t_k)\rho(\|\hat{\mathbf{S}}_i'(t_k)\|, \|\mathbf{S}_i'(t_k)\|)}{\sum_{k=0}^{m_i} N_{j,j+1}^i(t_k).} \tag{3.33}$$

The function $\rho(x, y)$ from [86] is used to measure how close the two scalars x and y are:

$$\rho(x, y) = (1 - x/y)^2. \tag{3.34}$$

Pairwise Landmark Distribution Term

We also add a length continuity regularization term that prevents the catheter from collapsing. However, in most of the cases, the length of the catheter is changing rapidly due to longitudinal motion. The length continuity regularization term causes the tracking to respond to the changes in catheter length rather slowly. Thus, in our application, another regularization term using the second order derivatives of the B-spline curve is added. This term allows catheter length changes but enforces that the distance between adjacent landmarks remains constant:

$$w_{p,q}^2(x_p, x_q) = \left| \frac{d\|\hat{\mathbf{S}}_i'(\tau)\|}{d\tau} |_{\tau=t_k} \right|. \tag{3.35}$$

Note that $\frac{d\|\hat{\mathbf{S}}_i'(\tau)\|}{d\tau} |_{\tau=t_k}$ denotes the value of $\frac{d\|\hat{\mathbf{S}}_i'(\tau)\|}{d\tau}$ at the point $\tau = t_k$. This term is defined for all edges in the edge set E_s.

Shape Continuity Term

In addition to length information, the shape of the catheters from the previous frames can also be used as a constraint in the current frame. The purpose of the shape continuity term is to rule out outliers caused by image noise or other catheters. The shape continuity term is defined by forcing the curvature of the catheter at each point to be equal to that of the corresponding point in the previous frame:

$$w^3_{p,q}(x_p, x_q) = \frac{\sum_{k=0}^{m_i} N^i_{j,j+1}(t_k)\rho(\|\hat{\mathbf{S}}^{"}_i(t_k)\|, \|\mathbf{S}^{"}_i(t_k)\|)}{\sum_{k=0}^{m_i} N^i_{j,j+1}(t_k).} \tag{3.36}$$

Here $\rho(x, y) = (1 - x/y)^2$, as defined previously. Again, this term is defined for all edges in the edge set E_s.

Relative Distance Preservation Term

All of the above regularization terms work on individual catheters. However, in clinical cases, it is common that multiple catheters are present in the same view. In such a situation, it is likely that two catheters or two sections of the same catheter can interfere with each other. Especially when two catheters or two catheter fragments are parallel and close to each other, then one of these two catheters may be incorrectly identified on the other catheters. In such cases, the relative distance between these two catheters should be used. To address this, every pair of nodes in set V, except for adjacent nodes of the same curve, is checked for the probability of overlap, based on their distance and collinearity. For a node (control point) j of the i-th B-spline, a position of this curve, $\mathbf{S}_{i,j}(\bar{t}_{i,j})$ at $\bar{t}_{i,j}$, which this control point j has the most influence on, is identified. The value of $\bar{t}_{i,j}$ can be obtained by solving:

$$\bar{t}_{i,j} = t_{index} \tag{3.37}$$

With

$$index = \arg\max_{k=0}^{m_i} B_{i,j}(t_k). \tag{3.38}$$

In this equation, the value of $\bar{t}_{i,j}$ is selected among a finite set $t_k, k = 0 \ldots m_i$ corresponding to m_i landmarks of the curve i. The B-spline kernel $B_{i,j}(t)$ for control point j should achieve the highest value at $t = \bar{t}_{i,j}$.

We assume that nodes p and q correspond to the j_1-th node on the i_1-th B-spline and the j_2-th node on the i_2-th B-spline, respectively. We use x_p/x_q to denote the label of node p and q. Since each label corresponds to a displacement in a local coordinate related to a search region, we use $\triangle\mathbf{c}_p$ and $\triangle\mathbf{c}_q$ to denote the displacement nodes p and q. Then the updated B-spline is obtained as

$$\widetilde{\mathbf{S}}_{i_{1/2}}(t) = \mathbf{S}_{i_{1/2}}(t) + \sum_{k=0}^{n_{i_{1/2}}-1} \triangle\mathbf{c}_{p/q} B_{i_{1/2},k}(t). \tag{3.39}$$

After this definition, if the nodes p and q satisfy the following conditions, there is a high risk that curves i_1 and i_2 will have negative interference on each other:

$$|\cos < \mathbf{S}_{i_1}(\bar{t}_{i_1,j_1}) - \mathbf{S}_{i_2}(\bar{t}_{i_2,j_2}), \mathbf{S}'_{i_{1/2}}(\bar{t}_{i_{1/2},j_{1/2}}) > | < \tau_{angle1}, \tag{3.40}$$

$$|\cos < \mathbf{S}'_{i_1}(\bar{t}_{i_1,j_1}), \mathbf{S}'_{i_2}(\bar{t}_{i_2,j_2}) > | > \tau_{angle2}, \tag{3.41}$$

with $\bar{t}_{i_1,j_1}$ calculated from eq. 3.38. Here we use $\tau_{angle1} = 0.7, \tau_{angle2} = 0.8$. $< *, * >$ denotes the angle between two vectors. These conditions select nodes which are both close to each other and parallel to each other. For such a pair of nodes, an edge is constructed to link them and this edge is referred as relative distance preserving edge (RDP edge). A RDP edge is added to an edge set E_r. Note that for different frames, the elements in E_r may differ. If nodes p and q are selected to construct an RDP edge, and the labels of these nodes are assigned to x_p and x_q respectively. The weight of this edge is determined by:

$$w^4_{p,q}(x_p, x_q) = \beta_r \delta_r e^{-\alpha_r |dis(\widetilde{\mathbf{S}}_{i_1}(\bar{t}_{i_1,j_1}), \widetilde{\mathbf{S}}_{i_2}(\bar{t}_{i_2,j_2}))|}, \tag{3.42}$$

$$\delta_r = |dis(\widetilde{\mathbf{S}}_{i_1}(\bar{t}_{i_1,j_1}), \widetilde{\mathbf{S}}_{i_2}(\bar{t}_{i_2,j_2})) - dis(\mathbf{S}_{i_1}(\bar{t}_{i_1,j_1}), \mathbf{S}_{i_2}(\bar{t}_{i_2,j_2}))|, \tag{3.43}$$

with $\alpha_r = 1.0$ and $\beta_r = 10.0$. Here $dis(*, *)$ is defined as

$$dis(\mathbf{S}_1, \mathbf{S}_2) = |\frac{\|\mathbf{S}_1 - \mathbf{S}_2\|}{\cos < \mathbf{S}_1 - \mathbf{S}_2, \mathbf{S}_1^{'} + \hat{s}\mathbf{S}_2^{'} >}|, \tag{3.44}$$

$$\hat{s} = sign(\cos < \mathbf{S}_1^{'}, \mathbf{S}_2^{'} >). \tag{3.45}$$

Note that $\mathbf{S}_i^{'}(t)$ is the derivative of $\mathbf{S}_i(t)$ in term of t. The term $\cos < \mathbf{S}_1 - \mathbf{S}_2, \mathbf{S}_1^{'} + \hat{s}\mathbf{S}_2^{'} >$ evaluates how parallel the curves are at these two control points while $\|\mathbf{S}_1 - \mathbf{S}_2\|$ evaluates how closed these two control points are in distance. The function $dis(\mathbf{S}_1, \mathbf{S}_2)$ measures both the distance and the extent of parallel tangential directions for the pair of nodes. All of the above defined terms are then combined together into a global objective function in eq. (3.25), which is optimized using the fast-PD discrete optimization approach.

3.4 Experiments

We evaluate the two algorithms that we have developed in this chapter: the catheter extraction algorithm in Section 3.2 and the catheter tracking algorithm in Section 3.3. Since the data we used are different for these two evaluations, we first introduce the data sets and then describe the experimental settings in each section separately.

3.4.1 Evaluation of the Catheter Extraction Algorithm

The purpose of developing the catheter extraction algorithm is to improve the tracking system by either (re)-initialization or post-processing. Thus the evaluation of the extraction algorithm is based on a tracking application. As a result, a comparison is performed between several tracking algorithms with and without the extraction algorithm. In this section, we compare three algorithms: (1) A tracking algorithm based on discrete optimization, previously proposed in [86]; (2) the catheter extraction algorithm used to segment catheters from each individual frame independently, and (3) a combination of (1) and (2) with our catheter extraction algorithm as a post-processing step.

Porcine data (consisting of five sequences and nearly 1000 frames) and patient data (consisting of four sequences, nearly 2000 frames) were used in our experimental evaluation. The image size is 512×512 and the image resolution is $0.43mm \times 0.43mm$ for porcine data and $0.34mm \times 0.34mm$ for patient data. All X-ray images were acquired using a C-arm fluoroscopy imaging system. For each frame, three X-ray images were taken at different angles. All of the porcine data used high dose X-ray for improved image quality. The patient data were acquired during trans-catheter aortic valve implantation procedures using low dose X-ray. All images contain a guide wire as well as a catheter. The ground truth catheter locations in the X-ray images were marked manually. The frame rate of the porcine X-ray sequences is 15 fps and approximately 8 to 10 frames cover one heart beat. The frame rate for the patient X-ray sequences is 30 fps and approximately 30 to 35 frames cover one heart beat. The main source of motion is cardiac motion rather than respiratory motion. Experiments were performed on an Ubuntu Linux system with a 3.40GHz CPU, 8GB RAM desktop computer, without GPU acceleration. The algorithm has been implemented in C++ and complied with the GNU Compiler Collection (GCC), using the default optimization option. The default option does not take into account multiple core parallelization for CPUs. Thus the evaluation of computational speed is based on a single-core. A higher speed can be expected when using both the highest optimization option and multiple core compilation. For all CPU implementations, image intensities are presented as 16-bit integers (shorts) or 32-bit floating-point (floats) numbers.

The following performance metrics were defined to evaluate the speed, accuracy and robustness of each algorithm:

- **Average frame rate:** The ratio of the number of frames and the computing time is the average frame rate. This measures the real-time capabilities;

- **Average tracking error:** For each landmark i, the shortest distance d_i to the ground truth is calculated. Then, a threshold ρ is used to select correctly tracked landmarks with $d_i \leq \rho$. The average d_i of the correctly tracked landmarks is defined as the tracking error;

- **Incorrect tracking percentage (ITP):** ITP is defined as the percentage of incorrectly tracked landmarks of the total number of landmarks. ITP indicates the reliability of the tracking results;

- **Failed tracking percentage (FTP):** For each landmark in the ground truth, the minimum distance to the tracked curve is calculated. The landmark is considered as successfully tracked if the distance is below ρ. Thus, FTP is defined as the percentage of ground truth landmarks that are not successfully tracked given the total number of landmarks in the ground truth. This evaluates to some extent the whole catheter can be tracked;

- **Ratio of failed tracked frames (ROF):** If FTP or ITP in a single frame is 100%, we regard this tracking as a failure. The ratio between the number of frames where a tracking failure is detected and the total number of frames indicates how often re-initialisation is needed. It also indicates the stability and robustness of the system.

Parameter Tuning

Parameters were selected based on either prior information or experiments across a range of possible values. Some of them, such as the scale of the SURF features, are dependent on the data and are selected based on prior knowledge for each dataset, e.g. the size of the catheter. Others were selected based on experiments and can be used for other data, such as the threshold of the SURF response, the threshold of the tubular score, the threshold for the edge points in patch analysis, the size of the patch for patch analysis, and the maximum and minimum growth step size for *Kalman filter-based path growing*. We summarize some key parameter choices in Table 3.1.

It can be observed that the scales for both the vesselness and SURF detectors are rather limited. This is because the width of the catheter can be considered as prior information in order to limited the number of scales which have to be used. This allows the algorithm to be fast. Another critical setting is the search range of the fast-PD tracking framework. We set the 1D search range in the transverse direction of the catheter to 10 pixels. By defining a

Table 3.1: Selected parameter values for the evaluation of the extraction algorithm

parameter	value	parameter	value
Scales for both Frangi's and SURF detectors [a]	4, 6	Threshold for the SURF response (τ_{surf})	0.01
Threshold of the tubular score (τ_{frangi})	0.1	Threshold of the edge points in patch analysis (τ_{edge})	0.005
Size of the patch for patch analysis (in pixels)	10	Trade-off between unary and pairwise terms (γ)[b]	0.1
Maximum step size for *Kalman filter-based path growing* (v_{max})	10	Minimum step size for *Kalman filter-based path growing* (v_{min})	3

[a] The scales correspond to the standard derivation of the Gaussian kernel during the calculation for the Hessian matrix for both Frangi and SURF detectors. [b] Based on the accuracy, ITP and FTP performance, which are the most important metrics, the trade-off value between unary and pairwise terms can be selected in the range $\gamma = 0.05 - 0.2$.

sufficiently large search region, the tracking system can capture large displacements including abrupt catheter position changes. However, a larger search region means a higher likelyhood of two or more overlapping catheters in the search region. This could confuse the tracking system and manual corrections would be frequently necessary. Another drawback is that the computational costs increase if the search region is enlarged. Using a 1D search space rather than a 2D search space means the search cannot capture motion in the longitudinal direction. This strategy makes the tracking fast but unable to track longitudinal movements. However, the tracking of longitudinal movements is not our objective during the fast-PD optimization since the longitudinal movements can be tackled completely by *Kalman filter-based path growing.*

Results

Fig. 3.5 shows the comparison results for different ρ (0.4-6.5) based on porcine data and clinical data. All plots show the threshold ρ in the horizontal axis. The plots in (a) to (d) show the performance in terms of error, ROF, FTP and ITP respectively for a value range of $\rho = 0.4 - 6.5$. The methods compared include our proposed method combining fast-PD and *Kalman filter-based path growing* extraction (red line), fast-PD only (blue line) and *Kalman filter-based path growing* extraction only (green line). The computational speed of the different approaches is

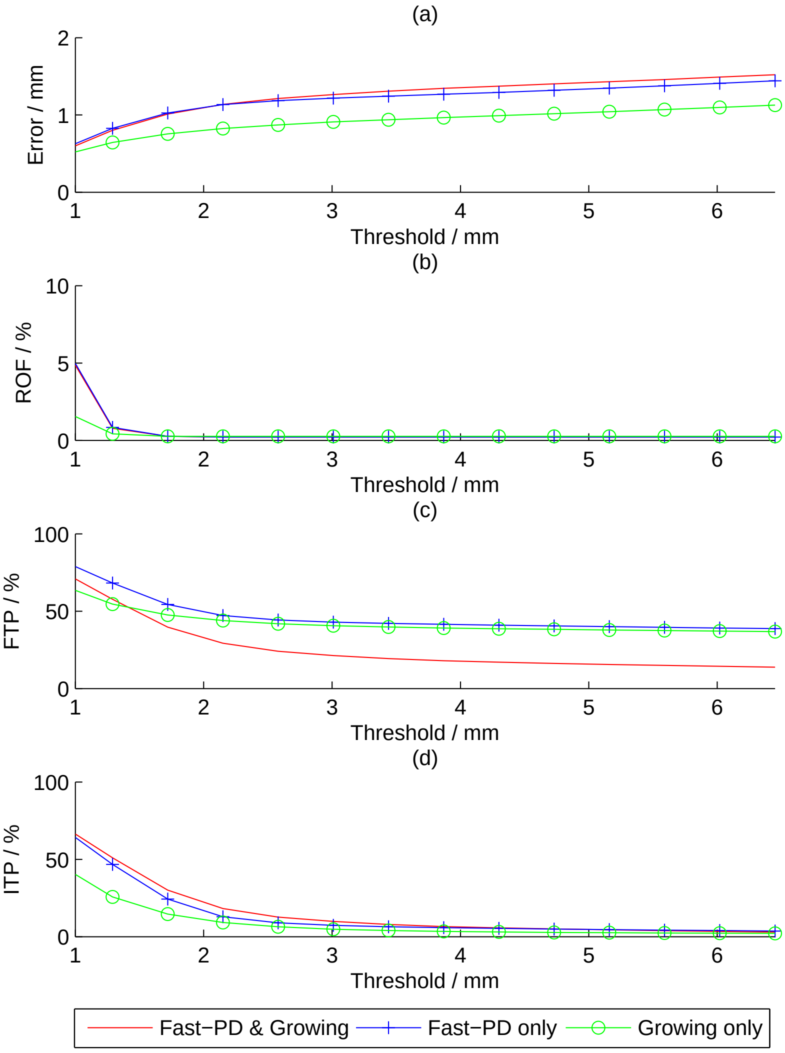


Figure 3.5: Comparison of experimental results on X-ray sequences.

31.3, 31.3 and 1666.7 frames per second. From these figures, it is clear that the combined strategy achieves a significantly lower FTP with a trade-off of a higher ITP. The catheter growing-only strategy achieves the highest speed and the lowest error. The results are shown in Table 3.2 for $\rho = 2mm$. Fig. 3.6 and Fig. 3.7 show example frames with results of the three methods for porcine and clinical data respectively. Rows (a), (b), (c) and (d) correspond to the original data, the results of catheter growing extraction only, fast-PD only and the proposed strategy combining fast-PD and *Kalman filter-based path growing*. The tracked curves are marked in green. For each row, the six columns correspond to frame no. 1, 21, 41, 61, 81, and 101. The figures show also that the proposed extraction method combined with a tracking framework can extract the entire catheter even when the initialization is incomplete.

Table 3.2: Results on X-ray image sequences for the extraction and tracking algorithm.

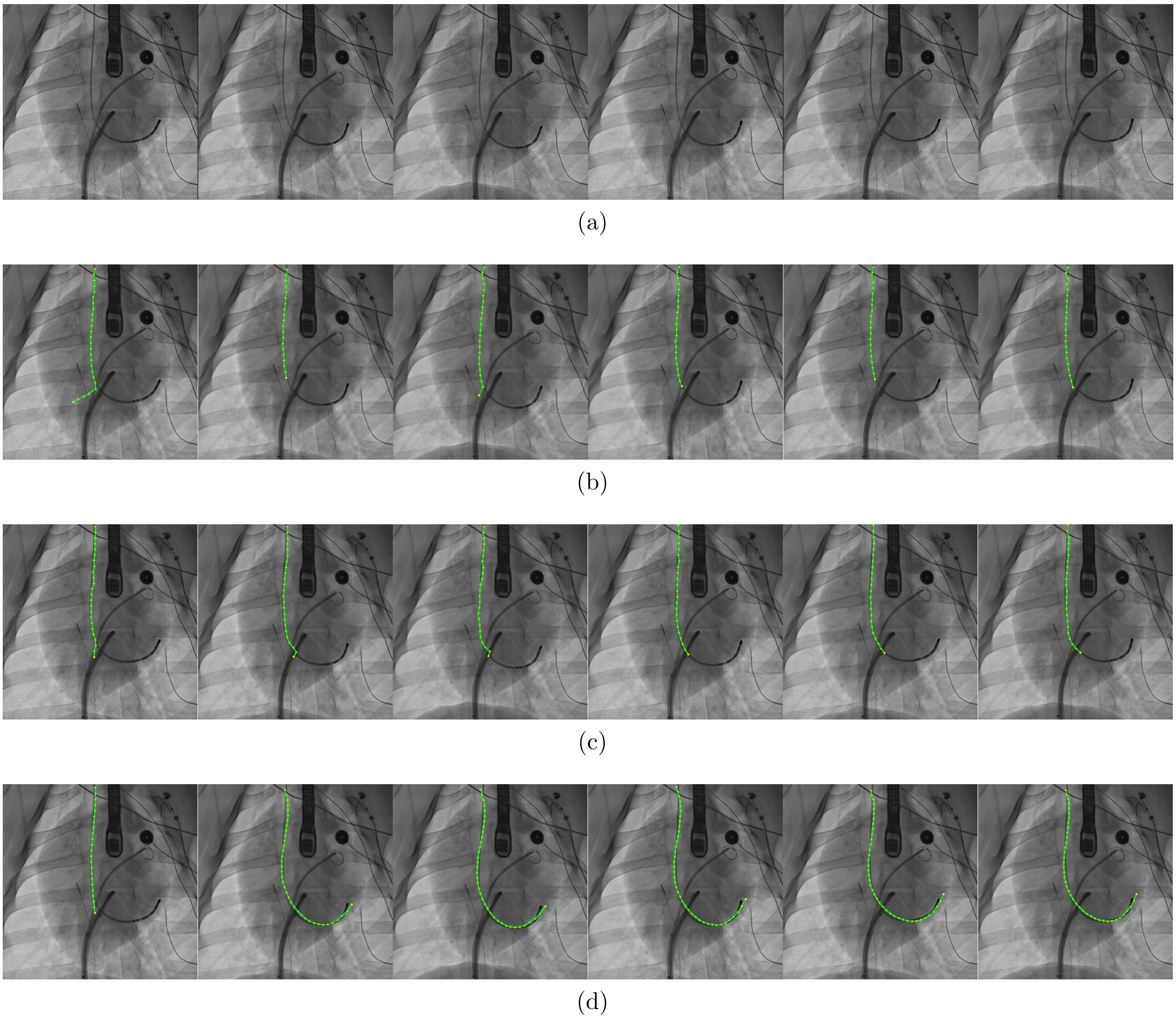
	Speed (fps)	**Error (mm)**	**FTP (%)**	**ITP (%)**	**ROF (%)**
fast-PD + Growing	31	1.137	29.2	18.2	0.21
fast-PD only	31	1.135	47.0	12.9	0.21
Growing only	>1000	0.82	44.0	9.2	0.26

In summary, the results indicate that the proposed extraction method based on a catheter growing strategy can quickly extract the catheter entirely with low error. This can be used for the initialization of the catheter tracking in an X-ray image sequence. However, it is not recommended to track catheters using the catheter extraction algorithm alone. Since every adjacent frame is segmented independently, no temporal consistency is enforced. Thus, using the catheter extraction algorithm for every frame individually will result in a less robust performance (higher ROF). Therefore, we have proposed to combine the catheter extraction strategy with a fast-PD tracking framework, which can adapt to changing catheter lengths. However, this approach tends to result in a higher ITP. An FTP below 40% can only be achieved by combining fast-PD and *Kalman filter-based path growing* strategies.

3.4.2 Evaluation of the Catheter Tracking Algorithm

In the previous section the catheter extraction algorithm is used for initialization of a tracking system or a post-processing step. However, this does not work in the case of two or more catheters. The *Kalman filter-based path growing* strategy may link two catheter fragments which could belong to different catheters. Thus, to track multiple catheters, the extraction algorithm introduced in the previous section is not used.

We tested our algorithm on 10 high dose X-ray porcine sequences, comprising around 700 frames with an original size of 512×512 pixels, with a resolution of 0.43mm×0.43mm. To improve the computational performance the input images were down-sampled to 256×256 pixels. Experiments were performed on a Nvidia GeForce GT 640 graphics card in a PC with a 3.40GHz CPU and 8GB RAM. The algorithm is implemented in C++ and CUDA and complied with



(a)

(b)

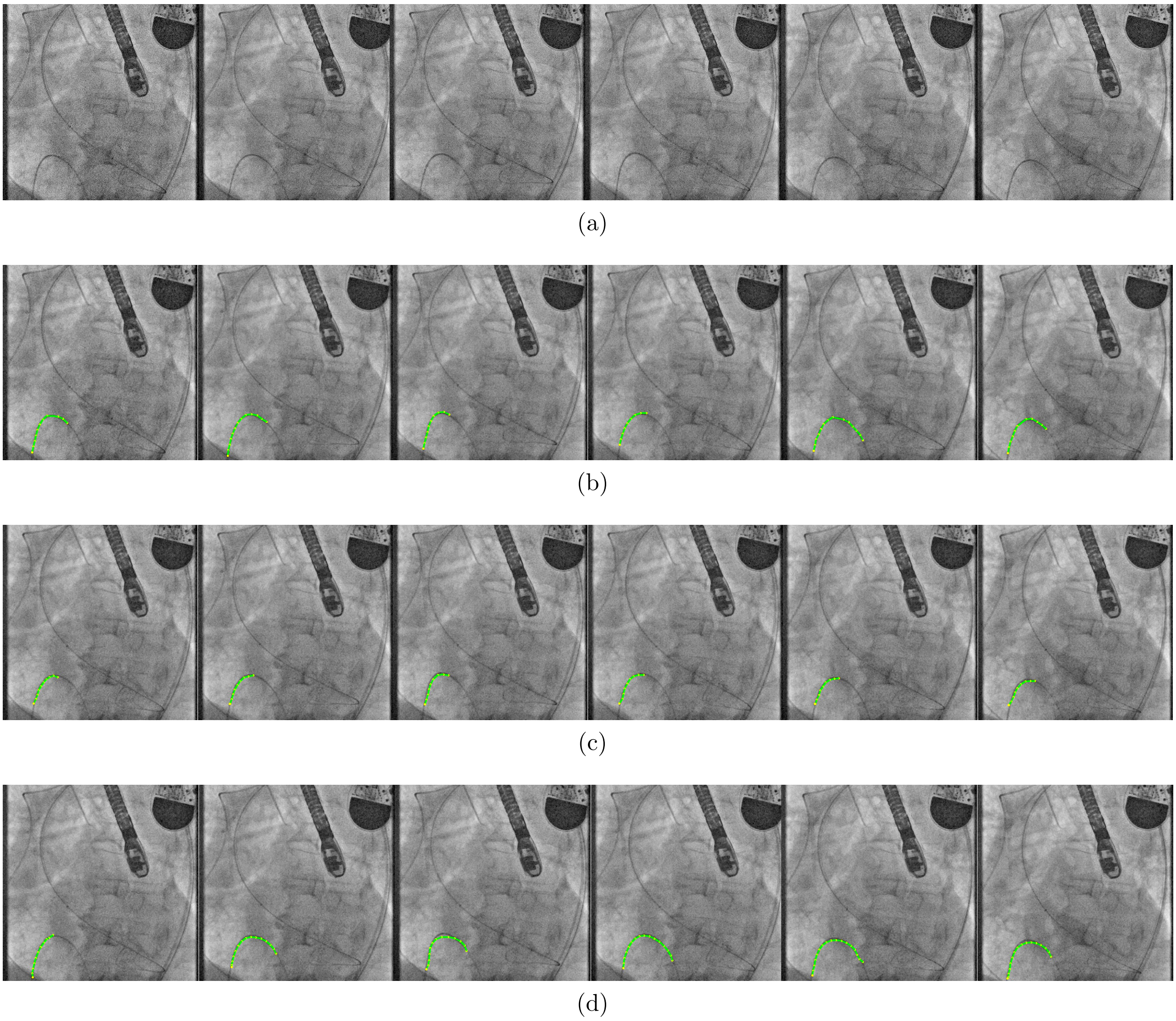
(c)

(d)

Figure 3.6: Examples of the tracking results in porcine X-ray data.

the Nvidia CUDA Compiler (NVCC) with default options. The GPU applications are optimized for maximal parallelization since the potential improvements in speed based on GPUs are expected to be significantly larger than for CPUs. For the GPU application, in order to improve the instruction throughput, the image intensities are represented by 64-bit integers rather than 32-bit floats. The precision is not affected by this. We compared two versions of the proposed algorithm, with and without the relative distance preservation (RDP) term for individual catheter tracking and multiple catheter tracking, In addition we also evaluated a previously proposed algorithm [86, 85] to demonstrate the improvements of individual catheter tracking and the ability for multiple catheter tracking.

The evaluation uses the following performance metrics:


(a)

(b)

(c)

(d)

Figure 3.7: Examples of the tracking results in patient X-ray data.

- **Average curve tracking error (CTE):** A landmark i of the tracking result is correctly tracked if the shortest distance d_i to the ground truth satisfies $d_i \leq \rho$. The average d_i among all correctly tracked landmarks is defined as the average curve tracking error;

- **Average catheter tip tracking error (TTE):** The average distance between the tracked catheter tips is compared with the ground truth catheter tips. This error is usually higher than CTE because no threshold is used;

- **Overlap percentage:** The length of the overlap between the tracking result and the ground truth compared the whole length of both the result and the ground truth, is defined as the overlap percentage. This measures the reliability of the tracking.

GPU acceleration

The implementation of our algorithm on a mixed CPU and GPU platform ensures a good computational performance when tracking multiple catheters. The allocation of the algorithm to the CPU or GPU depends on the nature of the parallelism. A hybrid CPU and GPU programming approach is used to perform some parts of the tracking on the GPU and other parts on the CPU. However, frequent switching between the GPU and CPU can cause significant overheads, i.e. data transfers between the CPU and GPU. This data transfer is often relatively slow and represents a bottleneck for real-time applications.

In our application, the pre-processing stages of the algorithm involve image blurring and the computation of the blob and tubular structure enhancement filters. In these pre-processing stages each pixel of the image can be computed independently. Thus, these computations are completely parallelizable and can be carried out on the GPU. They involve a data transfer from the CPU to GPU and GPU computation of the following images: the tubular structure image I_{tub}, the blob image I_{blob}, the orientation image I_{ori} and the enhanced tip image I_{tip}. Note that all of these processed images are stored on the GPU and there is no data copy from the GPU to the CPU. The reasons for this is that the following calculations based on these images are also carried out on the GPU.

The discrete optimization framework consists of two steps: Firstly, the computation of the cost term for each label (the unary term) and the computation of each label pair cost term (the pairwise terms); Secondly, the standard fast-PD optimization framework uses these terms as inputs to find the optimal solution. In our implementation, the fast-PD optimization is carried out on the CPU while the energy term computation can be fully parallelized on the GPU for both unary and pairwise energy terms. All energy term values are organized in form of tables and these tables have to be transferred from the GPU to the CPU, as input for fast-PD optimization that is carried out on the CPU. Although a GPU to CPU data transfer is needed, the approach is efficient because this table is much smaller than the image and can therefore be efficiently transferred.

Table 3.3: Parameter values for the evaluation of the tracking algorithm.

parameter	**value**	**parameter**	**value**
Scales for both Frangi and SURF detectors	4, 6	Threshold of the SURF response (τ_{surf})	0.01
Threshold of the tubular detector response (τ_{frangi})	0.1	Standard derivation of the Gaussian kernel for tip enhancement (σ_{tip})	5
Step size of landmark sampling ($\triangle t$)	5	Search range for each control point of B-spline model (R_{range})	5
Angular threshold for the selection of a pair of nodes with distance preservation term ($\tau_{angle1}, \tau_{angle2}$)	$(0.7, 0.8)$	Parameters for calculating the weight of distance preservation edge (α_r, β_r)	$(1.0, 10.0)$

[a] The scales correspond to the standard derivation of the Gaussian kernel during the calculation for the Hessian matrix.

Parameter Tuning

In order to optimize the performance of our algorithm, the parameter space of the algorithm must be explored in detail. Most of the parameters, such as the thresholds or the scales for tubular and blob detectors, or the search range for tracking, can be defined based on prior information. Table 3.3 gives a brief summary of these parameter values.

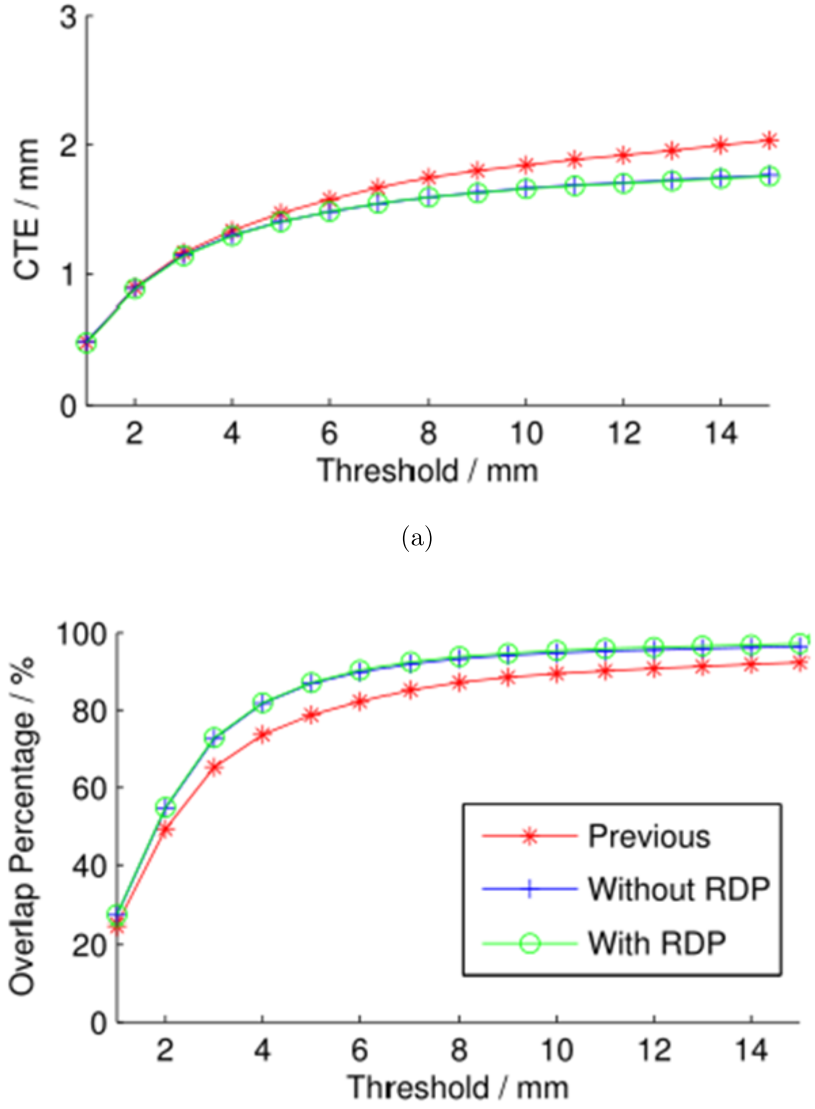
The weighting parameters for the different terms ($\gamma, \alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta_0, \beta_1, \beta_2$) are more difficult to select. These parameters are independent from each other. This makes the possible space of parameters extremely large. Furthermore, the optimal choice of parameters may depend on the actual image sequence. The motion pattern of the catheters, the frame rate or the intensity contrast, which may vary among different sequences, influence the choice of these parameters as well. This means that an optimal set of parameters for one sequence may not be optimal in another image sequence with different properties. Thus, we aim to find a set of parameters that represents a good compromise and works well for all tested image sequences.

To do this we firstly run an extensive grid search based on all available image data to obtain a set of parameters that achieves the best performance. Then in order to accelerate the parameter optimization process, we defined the range of all parameters to be within $[0, 1]$ and the resolution of the search grid to be 0.05. There are also linear constraints which are

$\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = \beta_0 + \beta_1 + \beta_2 = 1$. We found the best set of parameters to be $(0.5, 0.3, 0.3, 0.15, 0.15, 0.1, 0.4, 0.3, 0.3)$. Thus for most of cases, we use these parameters. Otherwise, we can adapt each parameter manually to achieve the best performance based on the output of the grid search. The following rules are used when we carry out parameter tuning for each individual sequence: (1) Increase γ when a more accurate tip tracking is required; (2) increase α_0 when image quality is high; (3) increase α_1 and decrease α_2 when longitudinal movements are small; (4) increase α_3 when the tracked object is more rigid; and (5) increase α_4 when the risk of overlapping catheters is high.)

Results

Fig. 3.8 shows a comparison of the CTE and overlap percentage for different values of ρ (1–15mm): From this, the improvements for CTE and overlap percentage of the proposed tracking system are visible for both individual catheter tracking and multiple catheter tracking, in particular when compared with the approach in [86, 85]. There is a very slight improvement when the Relative Distance Preservation (RDP) term is added. This is expected because the cases with a high risk of overlapping catheters are infrequent in our experimental data. The TTEs for the previous method, the proposed method with and without RDP are 10.9mm, 8.8mm and 8.1mm, respectively. The TTE is much higher than CTE because there is no threshold ρ applied. Given $\rho = 5mm$, our proposed approach (with RDP) achieves a CTE/TTE of 1.40/8.10 mm and an overlap percentage of 87.18%. A visual comparison is shown in Fig. 3.9. The sub-figures show the original frame (a), the result of [86, 85] (b), and the results of our method without RDP (c), and with RDP (d). The blue circles denote the nodes in V. Small yellow circles denote landmarks and each adjacent landmark is linked with a red edge. The pink lines denote the edges in E_r and the white lines denote the edges in E_s. The computation time of our tracking method is between 100-600 ms per frame depending on the parameter settings and datasets. The catheter extraction algorithm is much faster than the catheter tracking algorithm in X-ray images. During the tracking, the bottleneck for the computational speed is in the calculation of the values of objective function that is being optimized by the Fast-PD optimization. Besides the objective function calculation, the image pre-processing



(a)

(b)

Figure 3.8: Comparison of curve tracking error and overlap percentage for different error thresholds ρ.

and feature detection are the other main factors in terms of computational speed. Thus in the multiple catheter tracking module, image processing and feature detection, as well as the objective function calculation, are performed using the GPUs.

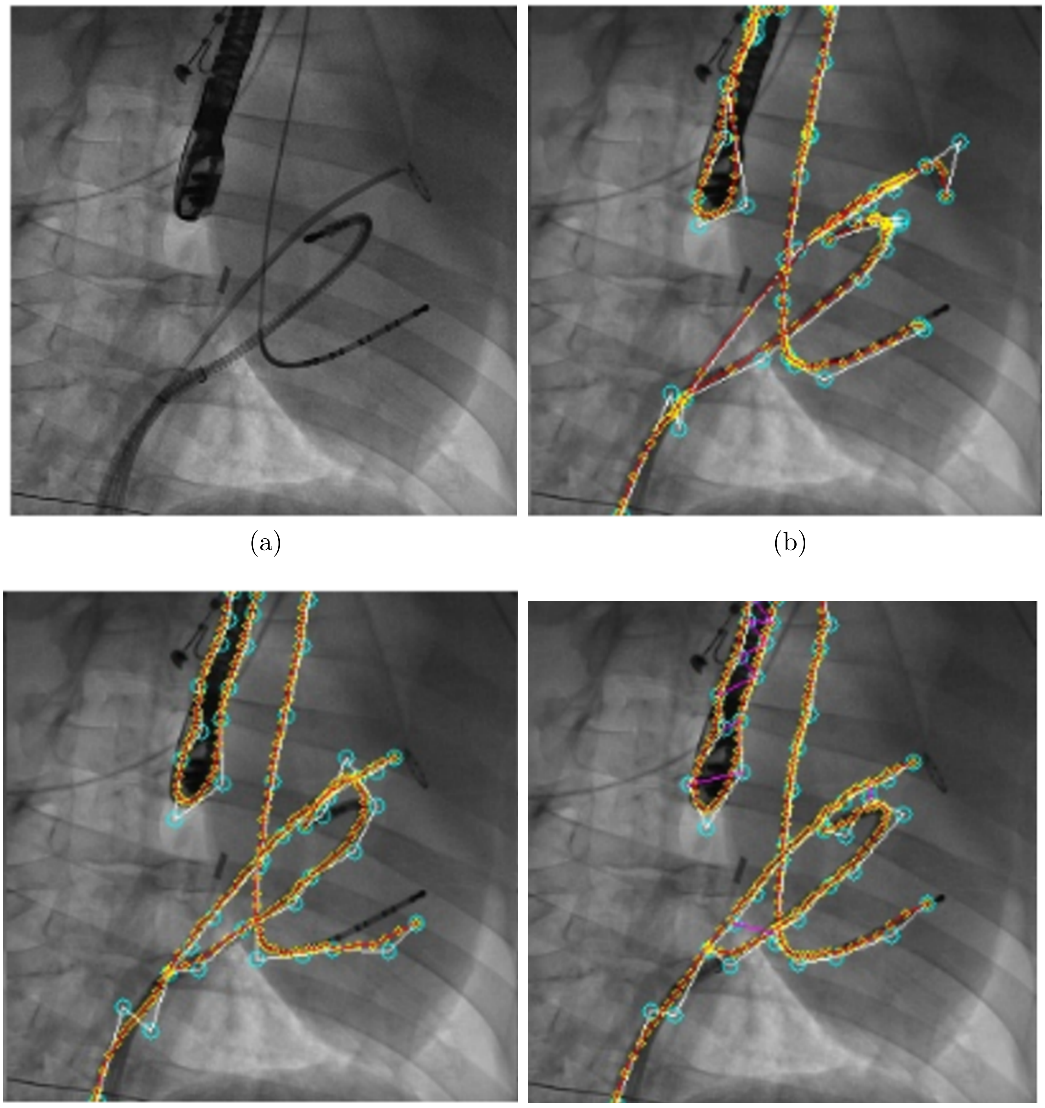


Figure 3.9: Example of the comparison for multiple catheter tracking with the original frame (a), the result from [86, 85] (b), and the results of our method without RDP (c) and with RDP (d).

3.5 Discussion

For catheter segmentation (extraction) in X-ray images, our proposed segmentation algorithm based on *Kalman filter-based path growing* can extract catheters without any prior information. This extraction can be performed at a speed of 100-600 ms per frame. It therefore satisfies the real-time requirements without the need to resort to hardware acceleration, i.e. the use of GPUs. The segmentation algorithm can be used for the initialization of catheter tracking, the recovery from tracking failures, or the extension of the proportion of the catheters being tracking. However, the start and end points for catheter growing are sensitive. Thus, manual interaction is needed to select the catheters to be segmented and tracked. Although these requirements reduce the proposed system from a fully automatic system to a semi-automatic system, it still significantly reduces the amount of manual work required.

During the catheter tracking, the proposed segmentation algorithm can be used to extend the proportion of the catheter, which may be 'lost' during the tracking stage. The tracking framework, which is based on fast-PD optimization, uses the consistency of adjacent frames. This makes the tracking more stable. However, the fast-PD optimization alone is unable to adapt the tracking when the length of the visible catheter changes due to longitudinal movements. Thus, a balance is sought between the fast-PD optimization and the *Kalman filter-based path growing* strategy in order to both preserve the consistency and to allow adaption to changes in the length of the visible cathether. From the perspective of our application, the *Kalman filter-based path growing* strategy has the ability to track the whole catheter but risks 'over-growing'. This means some non-catheter structures might be included in the catheter.

For the multiple catheter tracking approach, additional regularization terms are defined to address the challenge of multiple catheters in the same view. In this algorithm, it is recommended that the *Kalman filter-based path growing* extraction algorithm is not used. Because the extraction algorithm is very sensitive to mutual interferences of different catheters it can make the multiple catheter tracking unstable. Furthermore, the multiple catheter tracking strategy uses a large number of energy terms in the objective function. Thus it introduces a large number of coefficients that must be chosen in a heuristic fashion. This makes the parameter space large and difficult to tune. Although we propose values for these coefficients these may not be optimal for all cases. An online learning strategy for the parameter values could address this problem. Finally a large database of clinical patient data would help to evaluate our approach more thoroughly.

Chapter 4

Catheter Segmentation in Ultrasound Images using Registered X-ray Images

4.1 Introduction

The segmentation and tracking of catheters in ultrasound images is very challenging if no prior information is used. In this chapter we propose to use corresponding and registered X-ray images to assist with the segmentation of the catheters in the ultrasound images. A segmentation algorithm for catheters in X-ray images has been introduced in the previous chapter. This segmentation produces an estimate of the catheter in form of a B-spline curve. Given the spatial transformation between the X-ray and ultrasound modality, each curve in a 2D X-ray image corresponds to a curved surface in the corresponding 3D ultrasound volume. Thus the search space for the possible location of catheters in the ultrasound images can be dramatically reduced from the whole 3D volume to only a limited space surrounding this surface corresponding to the back-projected 2D catheter in the X-ray images. This strategy can reduce the interference of other catheter-like structures, which are far from the catheter location, and can reduce the computational cost for the segmentation in the ultrasound images.

In the following the spatial registration between the X-ray and ultrasound images will be used as a key step linking both modalities together. Their registration is estimated by aligning a pre-calculated transesophageal echocardiogram (TEE) probe model to its 2D projection in the corresponding X-ray images using the algorithm proposed in [73]. The transformation matrix, which describes a rigid transform, can be efficiently calculated by iteratively minimizing the re-projection error. The 2D-3D registration is not our contribution and therefore will be only briefly discussed (Section 4.2). With the resulting transformation matrix, we can determine a 3D curved surface in the ultrasound images that corresponds to the catheter curve in the X-ray images.

In Sections 4.3 and 4.4 two graphical models are introduced. These graphical models are used to address the challenge of catheter segmentation in ultrasound images with the search region for the catheter constrained by the back-projected curve surface. The first graphical model is based on a simple model with two distinct end nodes. In this model the catheter curve is modelled as the shortest path connecting the two end nodes. Although this model is easy to compute and has a small parameter search space, which makes it easy to tune, it is not flexible enough in more realistic clinical scenarios. We therefore propose a second model that is based on a hierarchical graphical model with two levels. In this model the catheter curve is modelled as a combination of the shortest path in the top-level graph and the longest path in the bottom-level graph. This is referred to as the optimal path in the following. In clinical cases this yields better results than those obtained from a simple single layer graph.

4.2 Registration of TEE and X-ray Fluoroscopy

The two imaging modalities are registered using the method proposed in [73]. The transformation matrix between X-ray image space and ultrasound image space is obtained by registering a previously acquired nano-CT image of the TEE probe to the projection image of the ultrasound probe in the X-ray image. In this registration process, digitally reconstructed radiographs (DRRs) are used to simulate the process of X-ray projection and projections of the 3D probe are generated. The best match between the DRR and the acquired X-ray projection image is then obtained by iteratively adjusting the transformation parameters (position and orientation) of the 3D probe image.

The transformation matrix $\mathbf{T}$ mapping each US volume from the US image coordinate system to the X-ray coordinate system consists of three terms:

$$\mathbf{T} = \mathbf{T}_{proj}\mathbf{T}_{probe}\mathbf{T}_{US}. \tag{4.1}$$

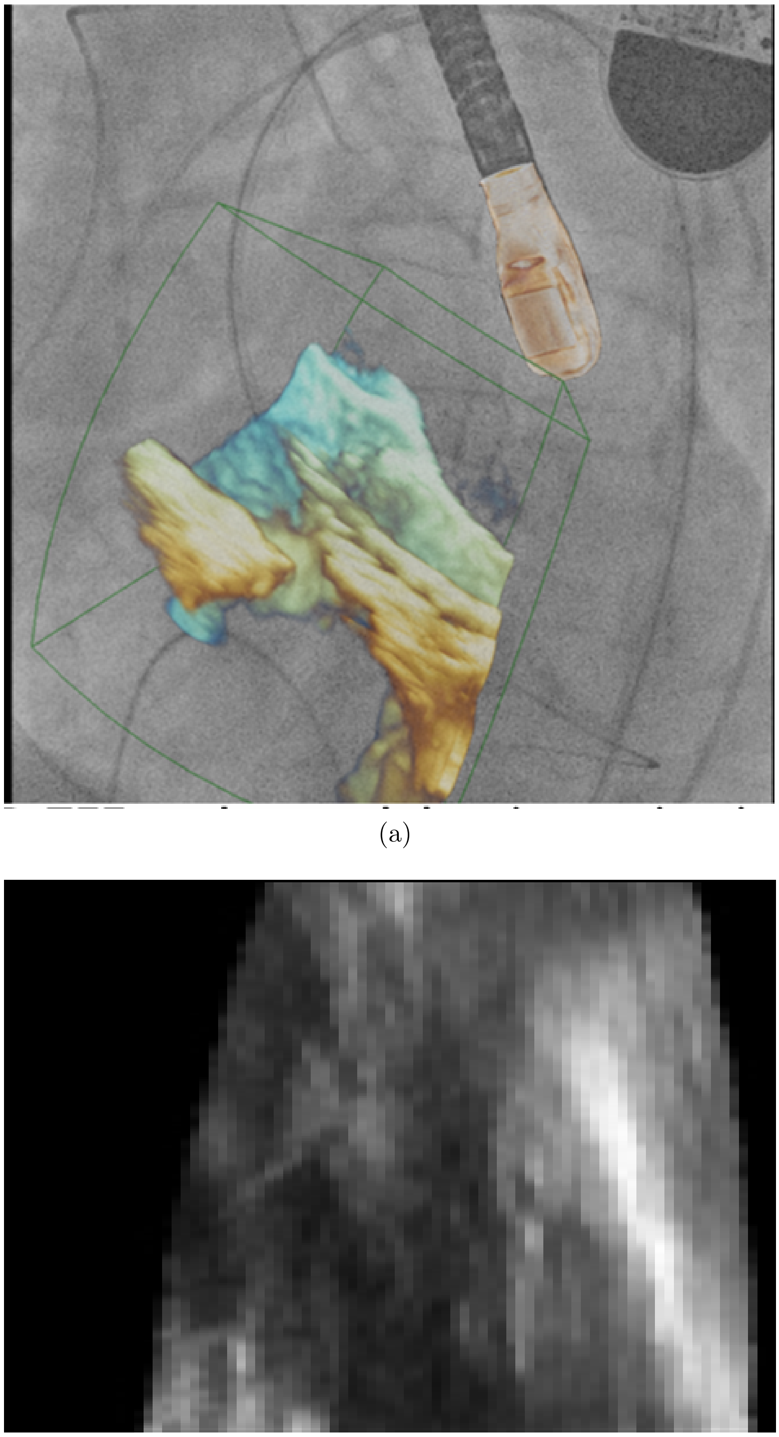
Here $\mathbf{T}_{US}$ transforms the 3D US volume to the US probe space, $\mathbf{T}_{probe}$ denotes the 3D position of the US probe in the 3D space of the X-ray scanner, and $\mathbf{T}_{proj}$ stands for the projection of the 3D position in X-ray space to the 2D X-ray image. $\mathbf{T}_{US}$ and $\mathbf{T}_{proj}$ can be acquired directly from the corresponding imaging devices or through calibration, while $\mathbf{T}_{probe}$ is calculated by the GPU accelerated 2D/3D registration algorithm described above. The 2D projection error for the X-ray images has been reported to be within 2mm in [73]. An example of a 2D/3D registration is shown in Figure 4.1(a). For further details regarding the registration algorithm the reader is referred to [73].

4.3 Catheter Extraction in Echocardiographic Images Using Shortest Path

4.3.1 Feature Detection in a Reduced Search Space

The extracted catheter in the X-ray image is represented by a set of sorted 2D landmarks and their homogeneous coordinates: $\mathbf{I}^i_{xray} = (x^i_{xray}, y^i_{xray}, 1)^T, i = 1, \ldots, n$ where n is the number of landmarks in the set describing the catheter. Suppose the voxels in the ultrasound image that correspond to each landmark in the X-ray image are denoted as $\mathbf{I}^i_{echo} = (x^i_{echo}, y^i_{echo}, z^i_{echo}, 1)^T, i = 1, \ldots, n$ then the relationship between the 2D X-ray landmark coordinate and its corresponding ultrasound voxel coordinate can be denoted by:

$$\mathbf{I}^i_{xray} = \mathbf{T}\mathbf{I}^i_{echo} \tag{4.2}$$



(a)

(b)

Figure 4.1: Registration of the ultrasound probe model to its X-ray projection (a) and an example of the curved surface in US corresponding to the catheter curve in X-ray (b).

Note that the transformation matrix $\mathbf{T}$ is a 3D to 2D projection matrix introduced in Equ. 4.1. Thus, for a fixed known X-ray coordinate $\mathbf{I}^i_{xray}$, there are an infinite set of points $\mathbf{I}^i_{echo}$ in the ultrasound image satisfying Equ. 4.2. These points comprise a back-projected ray in the ultrasound volume. Thus, any search strategy for the features of the catheter should focus on a search along each ray instead of the whole ultrasound volume. This can reduce the search cost dramatically. However, because of registration and catheter segmentation errors, the search space cannot be limited to just along the ray. Instead, the small space around each ray, within a small distance threshold, is also searched. The features, which we extract are SURF features as proposed in [16]. However, we use 3D SURF features rather than 2D SURF features as in the previous chapter. For each ray, the detected features are clustered if they are within a certain distance along the direction of the ray. Thus after searching along the i-th ray, a set of clusters $C_j^i, j = 1, \ldots, n^i$ is generated automatically, where n^i is the number of clusters found for the i-th ray. For each cluster C_j^i containing a set of feature points f_j^i, we compute its characteristic feature set S_j^i which is defined as:

$$S_j^i = \{f_{j,Xmax}^i, f_{j,Ymax}^i, f_{j,Zmax}^i, f_{j,Xmin}^i, f_{j,Ymin}^i, f_{j,Zmin}^i, f_{j,mean}^i\} \tag{4.3}$$

In this set, the first six points define a bounding box. The last feature point $f_{j,mean}^i$ is the mean of the previous six feature points, with a standard deviation d_j^i.

4.3.2 Catheter Extraction Based using a Graphical Model and the A$\star$ Algorithm

After a group of clusters has been generated, the next step is to identify which clusters are located on the catheter. For this, we have chosen an approach that builds a graphical model of the clusters and then finds the shortest path.

Clusters on nearby rays are connected by an edge considering each cluster C_j^i as a node in the graphical model. In detail, for a cluster C_j^i on the jth ray, all other clusters on rays $j+1$, $j+2$ are connected to this cluster. For each edge connecting two clusters (nodes), an initial weight $W_{i,j \to p,q}^{ini}$ of the edge from node C_j^i to C_q^p is calculated as a linear combination of the 3D Euclidean distance $d_{i,j \to p,q}^{ini}$ between the corresponding clusters and two standard deviations (defined in the last subsection) of both nodes d_j^i to d_q^p:

$$W^{ini}_{i,j\rightarrow p,q} = \alpha d^{ini}_{i,j\rightarrow p,q} + (1-\alpha)(d^i_j + d^p_q). \tag{4.4}$$

Here α is a parameter that must be set manually for different situations. In most of our experiments, we empirically set α to 0.5. $d^{ini}_{i,j\rightarrow p,q}$, which can be determined using the smallest distance between any pair of points from feature sets S^i_j to S^p_q. d^i_j and d^p_q are standard derivations calculated on sets S^i_j and S^p_q defined in e. q. 4.3.

Firstly, we set the original cost to connect two features equal to the distance between these two features. If two features are quite close to each other, the weight to connect them is a small positive value equal to the distance. However, for a path which consists of a large number of features which are close to each other, the accumulated cost for the whole path is still high, equal to the accumulated distance. In other words, if a path had no large gap and only consist of features with small intervals, the cost would still be high. The cost is dependent to the length of the catheter. In order to make the cost independent to the length, we assume that the original cost below a pre-set value (5 in our application) equals to zero. Thus, there is no cost to connect two nodes, which are close to each other. As a result, a path without large gap, even if the path is quite long, it will have a low overall cost.

The standard shortest path problem defines only one start node and only one end node in the graphical model. In our application, we first select all potential start nodes and all potential end nodes. If a node C^i_j near the border has no other node at rays $k < j$ connecting to it with zero-weight, then this node is selected as a potential start node. If a node C^i_j near the opposite border has no other node at rays $k > j$ connecting to it with zero-weight, then this node is selected as a potential end node. All potential start nodes are connected to a unique virtual start node with zero-weight and all potential end nodes are connected to a unique virtual end node also with zero cost. Thus the multiple start and end nodes are transformed to only one start and one end node. Therefore, the standard shortest path can be solved by the $A^\star$ algorithm [84]. Finally, smoothing is performed through fitting a B-spline curve.

4.4 Catheter Extraction in Echocardiographic Images Based on Optimal Path

In this section we introduce an alternative graphical model for the catheter segmentation in ultrasound volumes.

As mentioned before, we can use the registration transformation between the X-ray and US modalities together with the catheter extraction in the X-ray images to significantly reduce the search space for the potential catheter location in the US volume. This reduced search space corresponds to a region around a curved surface corresponding to the extracted catheter curve in the X-ray image (Figure 4.1(b)). This reduced space around the curved surface is first straightened to a rectangular volume. 2D SURF features are then detected on sampled cross-sectional 2D images extracted from this volume. Among these features, only a small proportion corresponds to the correct catheter. Then a two-layer hierarchical graphical model is used to organize all extracted features and the optimal path is identified in the graph. Features along this optimal path should correspond to the catheter. The optimal path is a combination of the longest path in the top layer graph and the shortest paths in each of the bottom layer graphs. The longest path helps to maximize the probability of the linked catheter fragments belonging to the complete catheter (rather than a shorter catheter segment or an imaging artefact) while the shortest path is used to select the features which represent each catheter fragment in a smoother way. The selected optimal path is considered as the candidate catheter extracted from the US volume.

4.4.1 Search Space Straightening and SURF Feature Extraction

The catheter extraction method used for the X-ray images proposed in the previous chapter cannot be extended to ultrasound images because the electrodes cannot be detected in the ultrasound images. In addition, the patch analysis used in the X-ray images becomes very computationally expensive if it is extended from 2D to 3D. The Frangi vesselness filter is also unsuitable because it is sensitive to artefacts in ultrasound images. In these images Frangi's vesselness filter detects too many outliers, which correspond to speckle and other imaging artefacts. This leads to many detected tubular structures with inconsistent longitudinal direction. If the longitudinal direction is roughly known, then the catheter will exhibit a blob-like appearance at the intersection of the transversal plane that is perpendicular to the longitudinal direction of the catheter. Thus a 2D blob-like detector, encoding prior knowledge about the longitudinal direction of the catheter can be more suitable for the detection of possible points on the catheter.

Given the extracted catheter in the corresponding X-ray image and the transformation matrix between these two modalities, a curved surface in the 3D US volume, which corresponds to the catheter in the 2D X-ray, can be extracted. The location of the catheter in the US image is ideally located within this surface. However, in practice, the location of the catheter is likely to be near this surface, but not exactly on it, because of catheter segmentation errors in the X-ray images and registration errors. Thus, the search for features should not be restricted to the 2D curved surface but should also be carried out along the direction perpendicular to the surface (see Figure 4.2). Because of the curved nature of the surface, perpendicular lines at different locations on the surface will be in different directions and may intersect each other. These intersections can cause an inconsistent ordering of candidate landmarks along these directions when compared with the original order. In order to cope with this, we define a single unified transversal search direction for every location on the curved surface. As a result, the transversal directions at all locations are parallel to each other and no intersections occur as Fig. 4.2 shows.

We re-write the transformation matrix for the 2D/3D registration as $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4]^T$. For the k-th landmark location $\mathbf{p}_k = (u_k, v_k)$ on the X-ray image, the corresponding ray equation in the 3D ultrasound volume is denoted by:

$$[\mathbf{t}_1 - u_k\mathbf{t}_4, \mathbf{t}_2 - v_k\mathbf{t}_4]^T\mathbf{X}^k = 0. \tag{4.5}$$

Here $\mathbf{X}^k = [x_1^k, x_2^k, x_3^k, 1]^T$ represents the coordinates of the voxels in the US volume corresponding to the k-th landmark in the X-ray image. The rays corresponding to locations on the catheter in the X-ray image form a curved surface $\mathbf{S}$. The optical axis $\mathbf{c}$ of the X-ray imaging

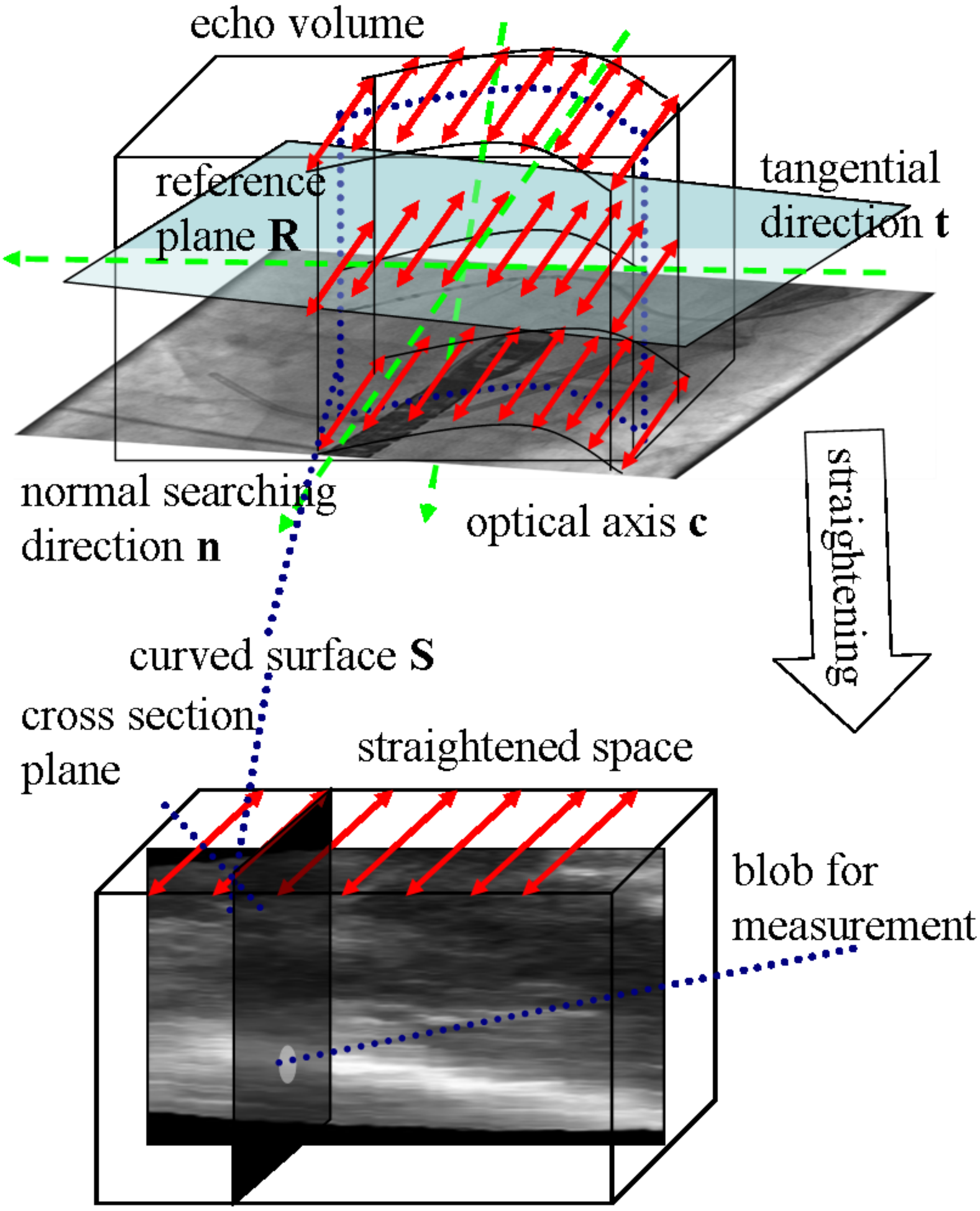


Figure 4.2: Search space straightening.

system in US space is normal to the X-ray plane. The plane which crosses the centre of the US volume with a normal vector along the optical axis **c** defines the reference plane **R**. The vector, in this plane **R**, connecting the projection of one end of the catheter to the other end, is defined as the longitudinal direction **t** of the catheter. Using this, the transversal direction **n** is defined perpendicular to both **c** and **t** (see Figure 4.2).

After the transversal search direction, **n**, and the reference plane, **R**, are defined, the search space surrounding the curved surface is straightened along the longitudinal direction, **t**, into a regular rectangular volume, **V**. First, the direction $\mathbf{v}^{\mathbf{k}}$ of each ray corresponding to a landmark on the catheter in the X-ray image, and the corresponding intersection $\mathbf{X}_0^k$ with the reference plane $\mathbf{R}$, are calculated using eq. (4.5). Then the ray equation can be denoted as $\mathbf{X}^k = \mathbf{X}_0^k + s\mathbf{v^k}$ where s is bounded by $[s_{min}^k, s_{max}^k]$. s_{min}^k and s_{max}^k correspond to the intersections of the ray with the border of the original ultrasound volume. Next, the value of each voxel in the straightened volume is calculated by

$$\mathbf{V}_{i,j,k} = I(\mathbf{x}_0^k + (j + s_{min} - 1)\mathbf{v}^k + i\mathbf{n}), \tag{4.6}$$

where $s_{max} = \max_k s_{max}^k$ and $s_{min} = \min_k s_{min}^k$. $I(\mathbf{x})$ is the intensity value of coordinate $\mathbf{x}$ in the original ultrasound volume. If $\mathbf{x}$ is outside the border of the volume, the value is set to zero. The search range along the transversal direction is also bounded to $[-\tau_{transversal}, \tau_{transversal}]$ where $\tau_{transversal}$ restricts the value of i. Eq. 4.6 maps voxels in the original US volume to their new locations in the straightened US space.

The longitudinal direction $\mathbf{t}$, along which the search space is straightened, corresponds to the z-axis of the new, straightened US volume $\mathbf{V}$. After a catheter has been transformed into the straightened space, it is likely that the catheter lies in the z-axis direction, with small changes in the other directions. Using this assumption, the $x-y$ cross-sectional plane through $\mathbf{V}$ will show a blob like structure at the position at which the catheter passes through the plane. However, in practice, the catheter is often perpendicular to the X-ray plane, so that the catheter lies along the y-axis and the $x-z$ cross-sectional plane shows a blob-like structure rather than the $x-y$ cross-sectional plane. Generally, the $y-z$ cross-sectional plane should not contain blobs. The SURF detector is a fast blob-like detector for 2D images. Thus, given a coordinate $\mathbf{X}$ in the straightened volume, we calculate the SURF detector response by examining both the $x-y$ and $x-z$ cross-section and select the higher of the two SURF detector values at that position. A set of candidate coordinates $\mathbf{X}_l^k$ with corresponding responses R_l^k for the l-th feature on the k-th ray, are retained for the next steps by defining a threshold τ_{surf}, .

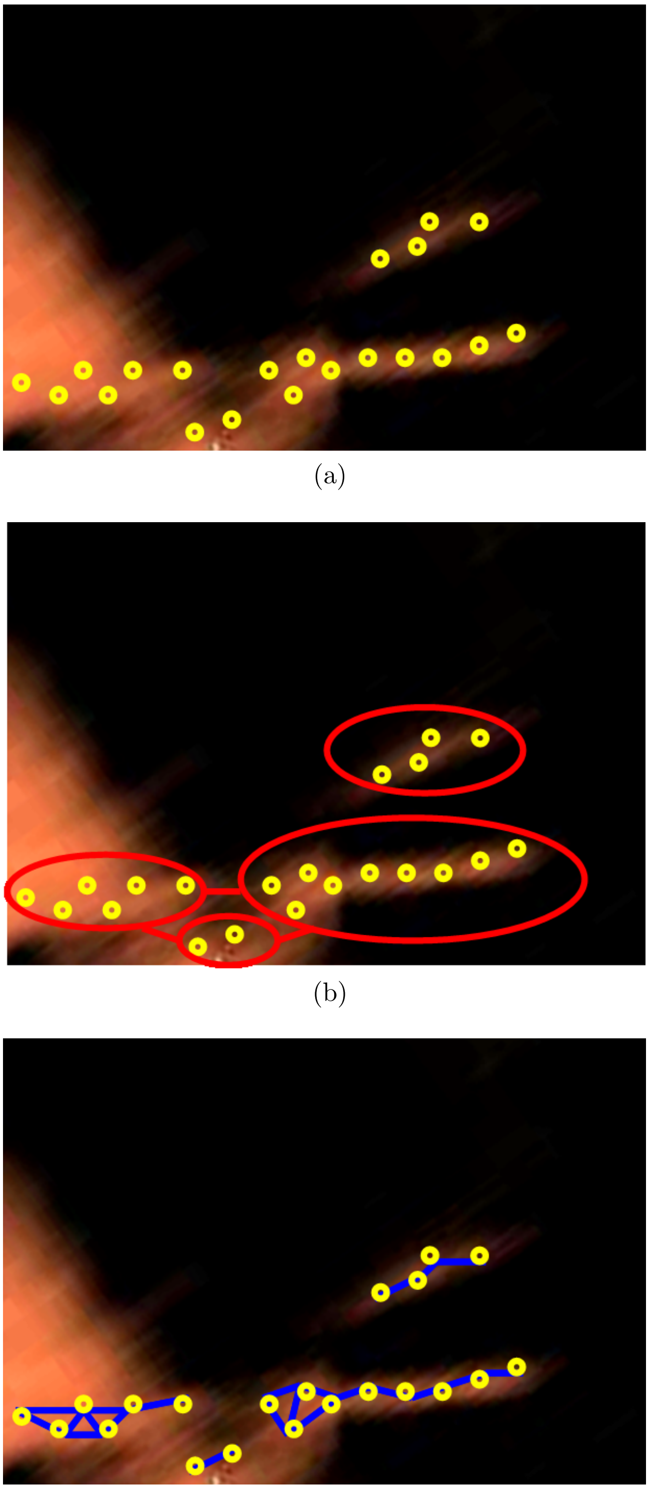
4.4.2 Two-Layer Hierarchical Graph Construction

Since we know approximately where the catheter may start and end (using the information from the X-ray images), it is possible to include this prior information via a graphical model with corresponding starting and ending nodes.

Given a set of feature location $\mathbf{X}_l^k$ on each ray k, the features are first clustered into small clusters using a single linkage clustering technique [183] based on their spatial distance. Each cluster corresponds to a section of a catheter-like structure. Within each cluster, a low-level graph is constructed based on the feature locations within the cluster. The shortest path then corresponds to a potential section of the catheter. Given a set of catheter sections, a high level graph is then constructed considering each catheter section (or low-level graph) as a node. These two layers are connected via end nodes of the low-level graphs. In other words, considering two clusters to be connected, the intra-edge is constructed by connecting the end node of one cluster to the start node of the other cluster. By finding the longest path through this graph, a set of catheter sections are selected and linked together to form the complete catheter model. The whole path, consisting of one inter-cluster path and several corresponding intra-cluster paths, is finally selected as the potential catheter model (Figure 4.3). The purpose of this two-level hierarchical graph is to preserve the integrity of each catheter-like structure at a low level, but also to link them together in an appropriate way.

To achieve this, each ray and pair of consecutive rays is considered. If the distance between a pair of features on these ray(s) is below a threshold $\tau_{subgraph}$, then an edge is allocated to link the two feature locations with a cost corresponding to the distance between them. After single linkage clustering, the features are organized as a set of clusters $C_j, j = 1, \ldots, m$ with the features inside each cluster linked to each other directly or via other features.

Within a cluster, the features $\mathbf{X}_l^k$ (k is the ray index and l is the feature index on this ray) with no direct link to other features that have a smaller ray index than k are selected as the candidate start nodes. Features with no direct link to those with a larger index than its ray index are selected as the candidate end nodes. In other words, start nodes should have no other



(a)

(b)

(c)

Figure 4.3: The sub-figures show (a) the original features that have been detected (yellow dots in the figure); (b) the top-level graph with each cluster of features as nodes (red); and (c) the bottom-level graph. The original features are organized in a two-layer graph corresponding to each cluster where the features inside the cluster are nodes related by edges (blue) based on distances.

linked nodes in front of them and end nodes should have no linked nodes behind them. Only one pair of start and end nodes is selected among the candidates for each cluster by maximizing the distance between the start and end nodes. This strategy aims to maximize the length of the corresponding catheter fragment. Next, clusters are combined if one cluster i's start or end point A_i is very close to another cluster j's start or end point A_j, and their directions are almost co-linear. We assume that for cluster i/j, if the start (end) point is $A_{i/j}$, then the end (start) point is denoted by $B_{i/j}$. Then they should satisfy the following constraints:

$$e^{|\langle \overrightarrow{B_iA_i}, \overrightarrow{A_iA_j} \rangle|} + e^{|\langle \overrightarrow{A_iA_j}, \overrightarrow{A_jB_j} \rangle|} < \tau_{colinear}, \tag{4.7}$$

$$|\overrightarrow{A_iA_j}| < \tau_{cluster}. \tag{4.8}$$

Here $< \mathbf{a}, \mathbf{b} >$ represents the angle between vectors $\mathbf{a}$ and $\mathbf{b}$. The start and end points are swapped if the start point has a larger ray index than the end point after the clusters have been combined. Finally, a sub-graph with start and end nodes for each cluster is constructed. These sub-graphs are considered as the bottom layer graphs for the hierarchical graphical model.

Following this, the top layer graph is constructed by using only the start S_i and end E_i of each cluster i as nodes and establishing directed edges among them. The rule of linking edges is as follows: for each cluster i, an edge $e_{i,i}$ (intra-cluster edge) is created from the start to the end points. For any pair of the clusters i, j which have not been merged but are likely co-linear, an edge $e_{i,j}$ (inter-cluster edge) is created from the end point of one cluster to the start point of the other cluster. The result is a directed acyclic graph which can be used to find the longest path. The weight of an edge should penalize linked endpoints which are far from each other or the endpoints whose corresponding catheter fragments are less likely to be co-linear. Thus the weights $W_{i,j}$ allocated to each edge are chosen as:

$$W_{i,j} = \begin{cases} \|\overrightarrow{S_iE_i}\|^{\frac{\mathbf{avg}_p R_p^i}{\tau_{surf}}}, & \text{if } i = j \\ -\|\overrightarrow{E_iS_j}\|^{\frac{e^{|\theta_i|}+e^{|\theta_j|}+e^{|\omega_{i,j}|}}{3}} & \text{if } i \neq j. \end{cases} \tag{4.9}$$

Here $\mathbf{avg}_p R_p^i$ is the average value of the SURF detector responses among the features of the cluster. θ_i is the angle between $\overrightarrow{S_iE_i}$ and $\overrightarrow{E_iS_j}$ and θ_j is the angle between $\overrightarrow{E_iS_j}$ and $\overrightarrow{S_jE_j}$.

$\omega_{i,j} = \theta_j - \theta_i$. After the top layer graph is constructed, the next step is to find the optimal path.

4.4.3 Finding the Optimal Path

At this stage, a two-layer hierarchical graph has been constructed. The top layer is a directed acyclic graph. The start and end points of each cluster are nodes in the graph. The edges and their weights are defined by the co-linearity of nearby clusters according to eq. (4.9). The bottom layer contains several graphs, one for each cluster. The feature locations are the nodes and the edges are the directed links between two features within a threshold. The cost of the edge depends on the 3D distance between two linked nodes. The longest path is selected in the top-layer graph using the dynamic programming technique presented in [179].

Having obtained the longest path, the intra-cluster edges which link only the start and the end of each cluster should be interpolated with more nodes from the cluster. This can be achieved by finding the shortest path for each sub-graph corresponding to each intra-cluster edge on the longest path. This shortest path problem is solved using the $A^\star$ algorithm. Each intra-cluster edge of the longest path is then replaced by the corresponding shortest path. As a result, the optimized path not only links likely catheter segments, but also preserves and smooths the shape of each catheter segment. In order to extract more than one catheter in the same view, multiple optimal paths can be extracted. This is achieved by repeated localisation of the longest path on the top-layer graph. At the end of each iteration, the nodes that have been already selected as the active nodes on the path of a catheter are ruled out to ensure the extracted optimal paths do not overlap each other.

When multiple catheters overlap in the X-ray images, the corresponding search space in the ultrasound images would have multiple catheters in the field-of-view. In this case, the optimal path finding procedure can be performed iteratively to extract multiple catheter candidates. After each iteration, the weights of the extracted path can be altered in order to ensure that these paths are not extracted again during the next iteration. In our experience the situation where the images contain multiple catheters are rare and we have therefore not evaluated this approach in the remainder of the thesis.

4.4.4 Post-processing

The above described operations are all performed in the straightened space $\mathbf{V}$. In the final step of the catheter extraction, all detected catheter points are transformed back to the original ultrasound volume. These points are then used to define a B-Spline curve that models the catheter. The longitudinal direction $\mathbf{d_i}$ of each point i on the catheter is then computed using a 1st-order derivative of the B-Spline curve at i.

Because the amount of overlap of the X-ray and ultrasound images is sometimes limited by registration errors or only a small percentage of the catheter that is extracted from the X-ray, we can additionally use a catheter path growing step to extend the extracted part of the catheter to the un-extracted parts. Because the patch analysis framework previously used in Chapter 3 is computationally expensive in 3D, it cannot be used to obtain accurate measurements for a Kalman-based filter framework. However, since the catheter is smoother in 3D than in it is 2D X-ray images (the projective nature of 2D X-ray images can cause sharp changes of direction) a straightforward and simple strategy can be applied here: Before applying the catheter path growing method, the response of Frangi's vesselness filter for each point is checked, starting at each end point and working inwards. Points are discarded from the ends until a response is found that is above a threshold τ_{frangi}. This ensures that incorrectly tracked parts of the catheter that are caused by changes of the length or shape of the catheter in the field of view are ignored.

This is followed by a catheter path growing step: Starting from the current estimate of the end points of the catheter, the direction $\mathbf{v_i}$ for the growing is along $-\mathbf{d_0}$ for the first landmark and $\mathbf{d_n}$ for the last one. Given two adjacent landmarks $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$, the step size for the catheter growing is:

$$\triangle s_i = (M - N)\mathbf{v}_{i-1}^T \mathbf{v}_i + N. \tag{4.10}$$

Here M and N are the maximum and minimum size of growing step, which are set manually. This means that smaller steps are used when a sharp turn occurs and larger steps are used if the catheter is linear. The growing direction combined with the step size yield the next candidate position:

$$\mathbf{p}'_{i+1} = \mathbf{p}_i + \triangle s_i \mathbf{v}_i. \tag{4.11}$$

The final position of the next landmark is determined by locating the voxel within a defined neighbourhood (we set a 10 voxels range as neighbourhood) of $\mathbf{p}'_{i+1}$ with the largest response of Frangi's vesselness filter in this neighbourhood. The response of Frangi's vesselness filter is used because the 2D SURF blob detector can only be used in the straightened space but not in the original space. The catheter path growing is terminated when all of the candidates' responses within the neighborhood are below the threshold τ_{frangi}.

4.5 Experiments

4.5.1 Evaluation of the Single Layer Graphical Model

The single layer graphical model algorithm was first evaluated on a pair of phantom images, consisting of a 2D X-ray fluoroscopic image and a 3D TEE image. Additionally, it has been evaluated on real patient data, consisting a 321 frame X-ray sequence at 30 fps and a 216 frame TEE sequence at approximately 14 fps. The overlap between these two modalities lasts for approximately 10 seconds. All of the X-ray images, acquired with a Philips Allura Xper FD10 C-arm, have the same size of 512 × 512 pixels while the TEE images, acquired using a Philips X7-2t TEE probe, have a size of 144 × 160 × 208 voxels. The patient was undergoing an aortic valve implant via the minimally-invasive trans-aortic approach. 2D/3D registrations were carried out on a PC running Window 7, with a 2.66GHz CPU, 4GB RAM with a NVIDIA GeForce GTX 280 graphics card courtesy of James Houston, King's College London, at St. Thomas' Hospital. All other experiments were carried out on a PC running Ubuntu Linux with a 3.40GHz CPU and 8GB RAM. All algorithms are implemented in C++ and compiled with

GNU Compiler Collection (GCC), using the default optimization option. The evaluation of the computational speed is based on a single core. Image intensities are represented using 16 bit integers (shorts) or 32 bit floats.

Our experiments are concerned with computation time and accuracy of the catheter segmentation. The computation time for the catheter segmentation in the X-ray and US images are denoted by t_{xray} and $t_{ultrasound}$ respectively. The time required for the registration, t_{reg}, is less than 1s. The total time can be calculated as:

$$t_{overall} = \max(t_{xray}, t_{reg}) + t_{ultrasound}. \tag{4.12}$$

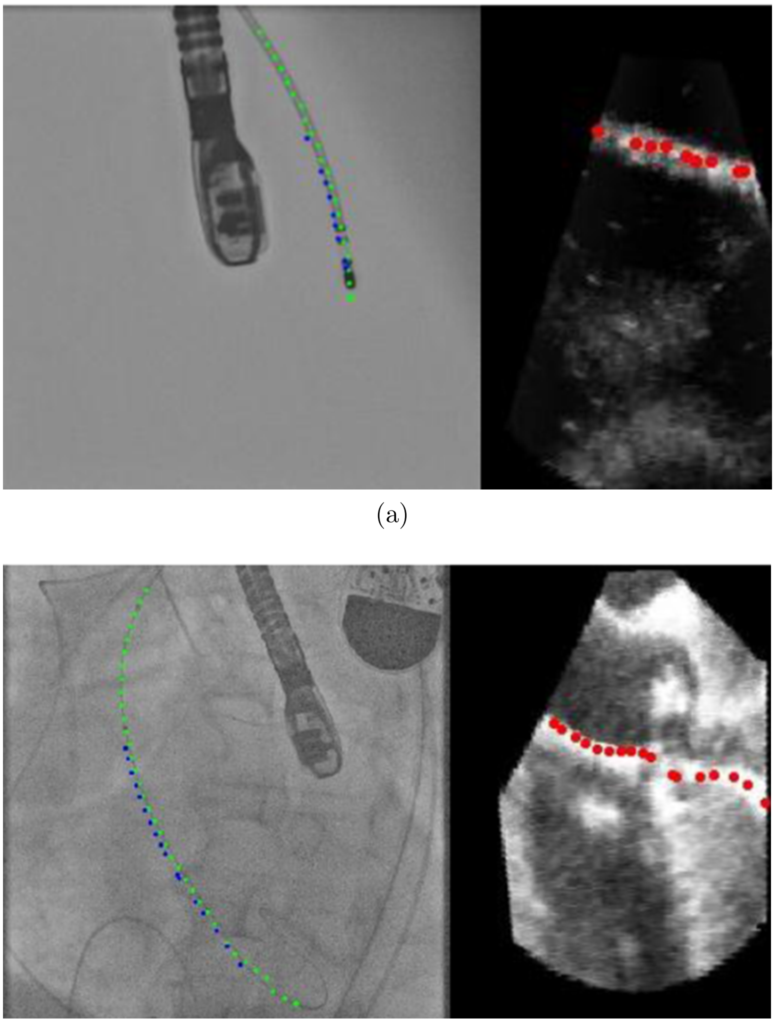
Because the SURF features are detected not only on the surface but within a margin around the surface, it is possible to evaluate the accuracy of the catheter segmentation via re-projection of the 3D points representing the catheter in the ultrasound image to the X-ray image and comparing them with the original 2D points extracted in the X-ray image directly. This re-projection error can be formulated as:

$$error = (\sqrt{\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{x}_i^{'}\|^2})/n. \tag{4.13}$$

Here $\mathbf{x}_i$ denotes the projection of the i-th point in the ultrasound image to the X-ray plane and $\mathbf{x}_i^{'}$ denotes the nearest point to $\mathbf{x}_i$ on the catheter detected in the X-ray image. This error includes both the segmentation error and the registration error.

The phantom experiment was undertaken in a tank of water. Only one ablation catheter and the ultrasound probe were put into the water. The procedure was simultaneously monitored by C-arm X-ray fluoroscopy and ultrasound. The catheter extraction is fast: 0.15s and 1.4s for the X-ray and ultrasound image respectively. The overall time $t_overall$ is around 2.4s with a re-projection error of 1.60mm. Fig. 4.4a shows some of the results using the phantom data. The ground truth for both X-ray and ultrasound images were obtained via manual annotation.

The average performance of the algorithm was evaluated using real patient data. The last ten

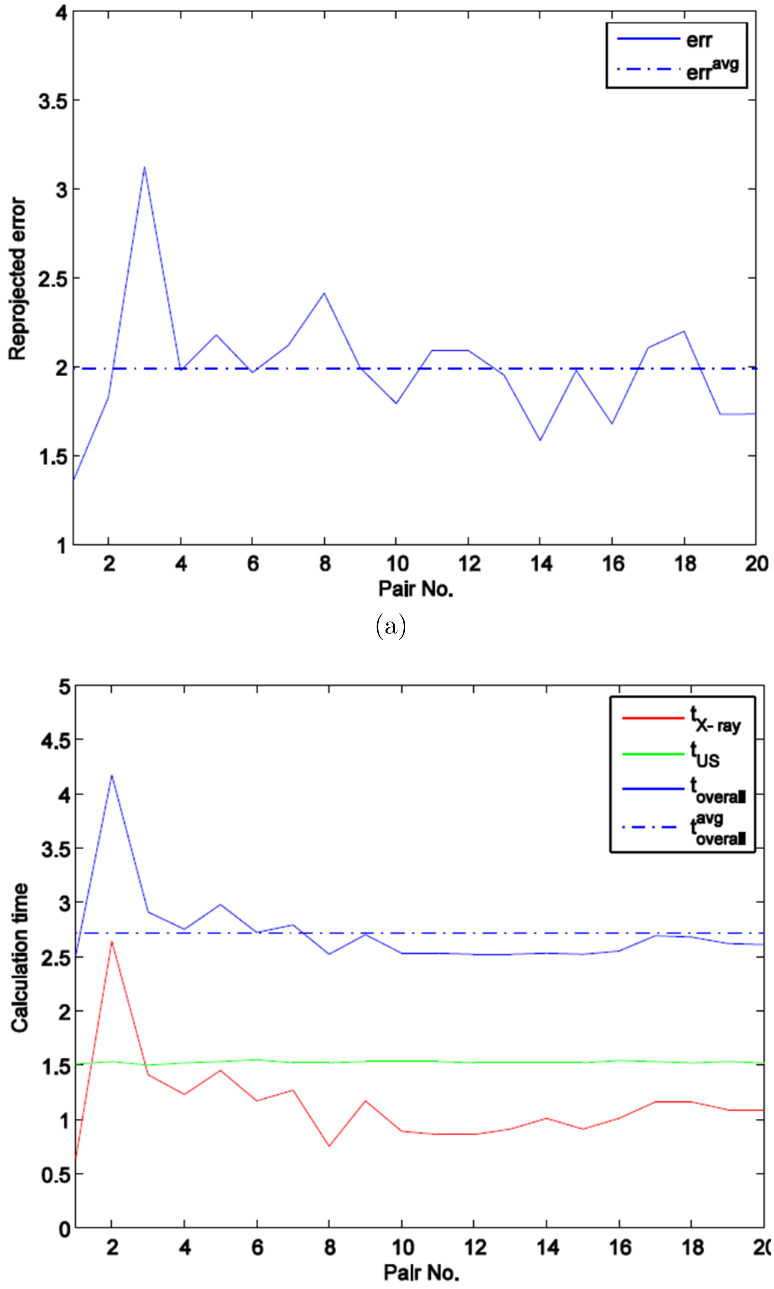


(a)

(b)

Figure 4.4: Examples of the catheter extraction using the single-layer graphics model: (a) from phantom images and (b) from real patient.

frames in the ultrasound imaging sequence (the registrations are more accurate) were paired with up to 3 X-ray frames, which were most likely to be their corresponding frames in time. 20 such pairs were formed and some of the segmentation results are shown in Fig. 4.5. The green points denote the original catheter extraction in the X-ray images while the blue points denote the back-projections from the ultrasound images to the corresponding X-ray images. The last frames of both X-ray and ultrasound were also paired and the results are shown in Fig. 4.4b. For the patient sequence, the average re-projection error is 1.99mm and the overall computation time is 2.72s.



(a)

(b)

Figure 4.5: Performance in 20 pairs from real patient data: (a) re-projection errors are on average 1.99mm. (b) Computation times are on average 2.72s.

4.5.2 Evaluation of the Hierarchical Graphical Model

For the evaluation of the hierarchical graphical model four sequences of clinical patient datasets were used in our experiments. The patient data were acquired during trans-catheter aortic valve implantation procedures. All of the X-ray images were acquired as low dose images. All images contain a guide wire as well as a catheter. The X-ray and ultrasound sequences data were

Table 4.1: Dataset Information

Modality	Number of Frames	Image Size	Resolution
X-ray	2000	512×512	0.34×0.34 mm^2
ultrasound	800	144×160×208	0.60×0.60×0.29 mm^3

synchronised by manually matching image frames. The ground truth catheter locations in the X-ray and ultrasound images were marked manually by an expert. The catheter is easy to locate in the X-ray images and therefore the intra-observer variability is low. However, the identification of a suitable ground truth in the ultrasound images was more challenging. We randomly sampled 40 frames from all datasets and for each frame the catheter was annotated five times. For each consecutive two annotations, the average error and the overlap percentage was calculated when the two annotations were compared. The overall error is 0.7mm and the overlap percentage is 87.3%. The frame rate of the X-ray images is 30 fps and approximately 30∼35 frames cover one heart beat. The frame rate of the ultrasound images is around 6∼7 fps and 6∼8 frames cover one heart beat. The main cause for catheter motion is therefore cardiac motion rather than respiratory motion. For the ultrasound images, we also estimate the motion between two adjacent frames given its low frame rate. The average motion along transversal and longitudinal directions are approximately 4mm and 6mm respectively. Because the frame rates of the X-ray sequences are higher than the frame rates of the corresponding ultrasound image sequences, one ultrasound image may correspond to two or three X-ray images rather than one single X-ray image. Thus although there are only 800 frames of ultrasound data, the number of corresponding X-ray and ultrasound image pairs consists of more than 2000 frames. Table 5.1 gives more details about the datasets used in this section. All experiments were performed on a PC running Ubuntu Linux system with a 3.40GHz CPU and 8GB RAM. All algorithms are implemented in C++ and compiled with the GNU Compiler Collection (GCC), with the default optimization option. Image intensities are represented using 16 bit integers (shorts) or 32 bit floats.

The following performance metrics were used to evaluate the algorithm's speed, accuracy and robustness:

- **Average frame rate:** The ratio of the total number of frames to the total time consumed is used to represent the average frame rate. This measures the real-time capabilities;

- **Average tracking error:** For each landmark i, the shortest distance d_i to the ground truth is calculated. Then a threshold ρ is used to select correctly tracked landmarks with $d_i \leq \rho$. The average d_i of correctly tracked landmarks is defined as the tracking error;
- **Incorrect tracking percentage (ITP):** ITP is defined as the number of incorrectly tracked landmarks divided by the total number of landmarks. ITP indicates the reliability of the tracking results;
- **Failed tracking percentage (FTP):** For each landmark on the ground truth, the minimum distance to the tracked curve is calculated. The landmark is considered as successfully tracked if the distance is below ρ. FTP is defined as the number of ground truth landmarks that are not successfully tracked divided by the total number of landmarks in the ground truth. This measure evaluates to what extent the whole catheter can be tracked;
- **Ratio of failed tracked frames (ROF):** If FTP or ITP is 100%, we treat this as a failure. The number of frames where a tracking failure is detected divided the total number of frames indicates how often re-initialisation is needed. It also indicates the stability and robustness of the system.

First, the proposed method is compared to the strategy without the catheter path growing during the post-processing to demonstrate the advantage of catheter growing in cases where there is large longitudinal motion in the ultrasound images. The advantages of the proposed similarity measurement are also demonstrated by comparison with other vesselness filtering methods, specifically Frangi's and Sato's vesselness filtering methods proposed in [69] and [174]. Finally, the overall algorithm is compared with the previously proposed method introduced in the last section, which is based the single layer graphical model. This comparison highlights the limitations of the single layer graphical model when applied to real-world clinical data.

When using the 2D SURF detector, only three scales ($\sigma = 4$, 6 and 8 for Gaussian smoothing) are used to calculate the responses of the SURF detector in order to speed up the computation.

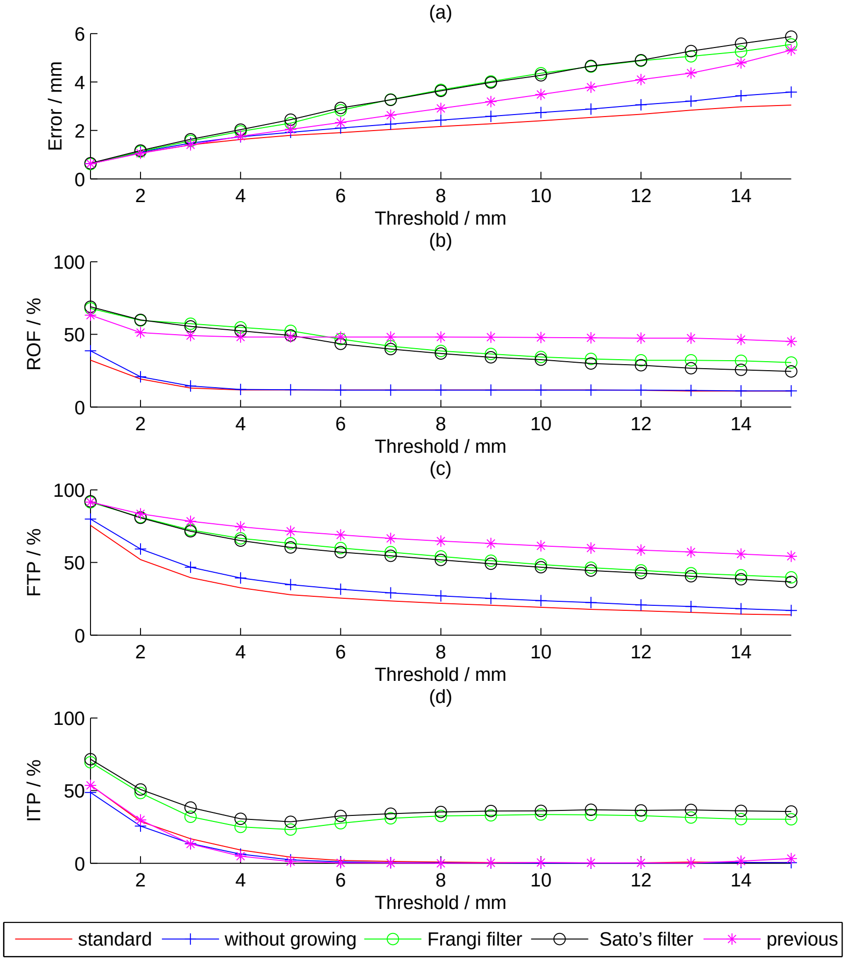


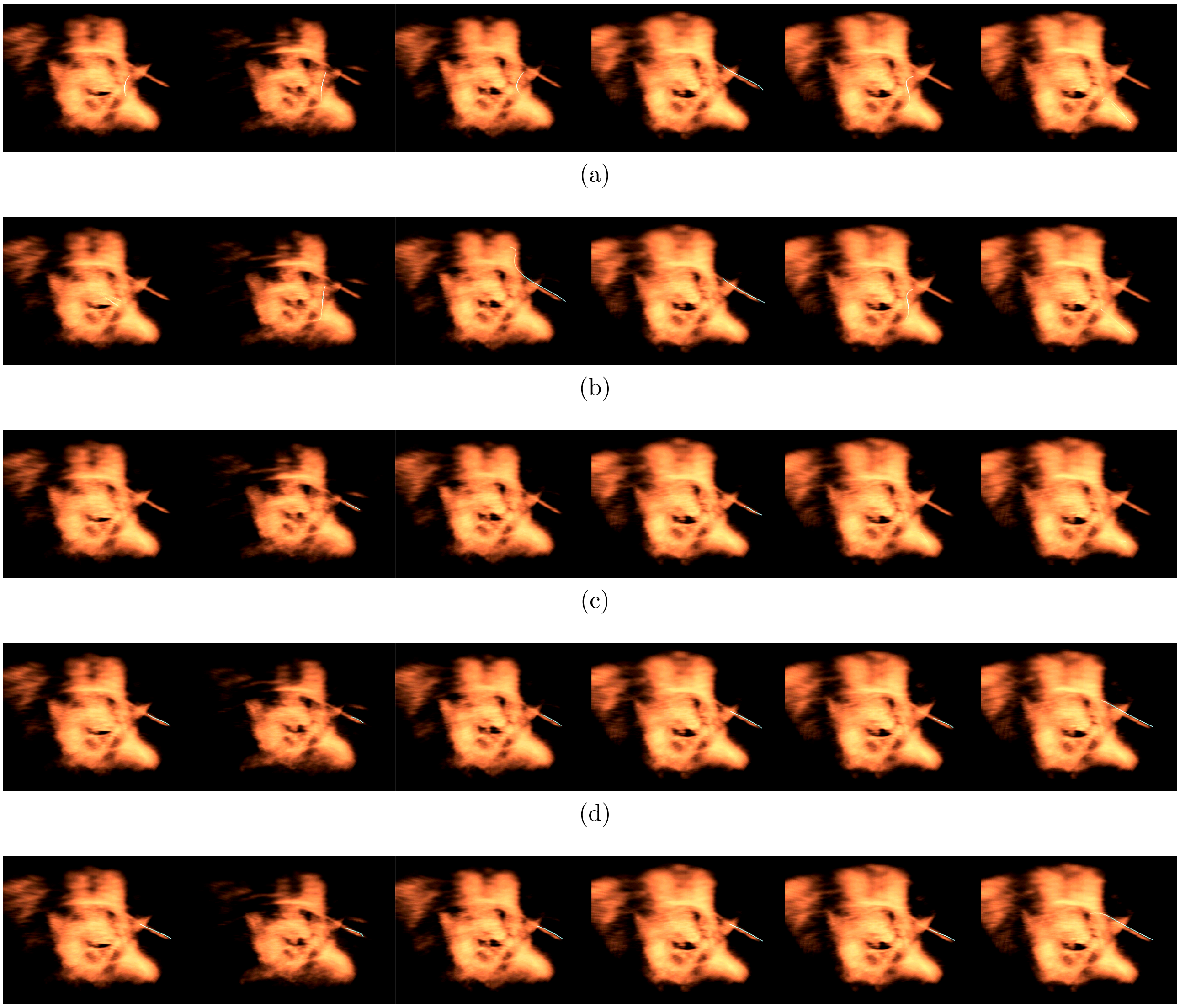
Figure 4.6: Experimental results for the ultrasound sequences.

The threshold $\tau_{surf} = 0.01$ is used to select feature locations. The search range $\tau_{transversal}$ along the transversal direction for each voxel on the curved surface crucially depends on the accuracy of the registration between X-ray and ultrasound images. Usually, it is set to 10 *pixels* to account for small registration errors, which means that the catheter is expected to lie on or very near the curved surface that is back-projected in the ultrasound image. In the case of poor registration or synchronization, the search range can be increased to 30, although there is a tradeoff between larger search ranges and increased computation times. At the stage of graph construction, the threshold for the minimum distances for the subgraph clustering is set to $\tau_{subgraph} = 10$. The threshold to decide on whether to merge any two clusters is set to $\tau_{colinear} = 1$ and $\tau_{cluster} = 20$. For the catheter path growing stage, the maximum and minimum growing steps are set to $M = 5$ and $N = 3$ respectively.

Fig. 4.6 shows the results of the evaluation in terms of tracking error, FTP, ITP and ROF. The FPS is 0.74, 1.35, 0.23, 0.23 and 2.00 frames per second for the proposed method with and without catheter path growing, Frangi's vesselness filtering, Sato's vesselness filtering, and the single layer graphical model. The vesselness filtering methods have the largest errors and highest ITP and are only better than the shortest path strategy in terms of ROF and FTP. The single layer graphical model performs much worse than the proposed strategy in all performance metrics except for ITP. The proposed method with or without catheter path growing strategy are the best two methods in almost any performance metrics and for any threshold value ρ. Considering only the catheter path growing strategy the use of the catheter path growing achieves lower errors, ROF and FTP. The trade-off is slightly increased computation times and very small increases in ITP.

Fig. 4.7 shows example results for the different methods. Rows (a), (b), (c), (d) and (e) correspond to the results using vesselness filtering [69, 174], the single layer graphical model and the proposed methods without and with catheter path growing. The segmented catheter is shown in green. For each row, the six columns correspond to frame no. 1, 21, 41, 61, 81 and 101. Given $\rho = 5mm$, our proposed method achieves a speed of 0.74 FPS and an error of 1.79 mm, with a FTP of 27.7%, ITP of 4.19% and a failure ratio of 11.6%. Although the performance is worse than that in X-ray images, the results are very encouraging given that the X-ray images have much more clearly defined borders of the catheter against the background while ultrasound images have much more artefacts.

Although the catheter path growing strategy can recover more of the catheter and extract parts of the catheter that are not visible in the X-ray images, the trade-off is a larger computational cost and a higher incorrect tracking percentage. Whether this approach should be used in clinical applications depends on whether the whole catheter is preserved in the straightened space. If not, this indicates that some parts of the catheter are not in the field-of-view of the X-ray images or that the registration between these two modalities is inaccurate.



(a)

(b)

(c)

(d)

(e)

Figure 4.7: Examples of the catheter tracking in ultrasound images from patients.

4.6 Discussion

Given the experimental results of different methods for catheter extraction in ultrasound images, we found that the 2D SURF blob detector performs better than vesselness filtering methods. The advantage of the blob detector in this application is that the direction of the catheter can be well estimated based on the result of X-ray tracking and the registration between X-ray and ultrasound images. If the catheter direction is known, it is easy to locate a series of planes through which the catheter will exhibit a blob-like appearance on the intersection of the cross-sectional planes and the catheter. A 2D SURF detector is thus more suitable to detect blob-like structures in these planes. The vesselness filtering techniques are more general and more flexible, so they can be used more widely. The filters usually estimate the direction of tubular structures by analyzing the eigenvalues of the Hessian matrix of second derivatives. However, if the image intensities are noisy and corrupted by artefacts such as speckles, the Hessian matrix estimated from the image is not reliable. Thus, the analysis on the eigenvalues does not produce an accurate estimate of the direction of the catheter. Vesselness filtering also ignores the prior information available from the X-ray segmentation. For these reasons, vesselness filtering is a less effective method for catheter segmentation. This is also supported by our experimental results. However, if a failure in the X-ray segmentation or in the X-ray to ultrasound registration occurs, prior information of the catheter direction would not be available for the ultrasound image volume, and in this case, vesselness filtering would be the only viable option.

The single layer graph model shows a good performance on phantom data, but fails in most of the frames in clinical patient data. The reason for this is the fact that in the clinical sequences, the catheter does not always follow the shortest path in the constructed graph if the start and end nodes cannot be fixed. Usually, the objective of the model is to maximize the likelihood of a path to represent the catheter. This maximization problem can be transformed to a longest path problem rather than a shortest path problem. Suppose that a catheter image comprises a set of broken fragments rather than a complete continuous catheter. The shortest path then corresponds to only one catheter fragment, whereas the longest path is able to segment the whole catheter. In the clinical sequences, the poor quality of ultrasound images means that this occurs frequently. On the other hand, to model each individual catheter fragment, for which we know the start and end nodes, it is more appropriate to use a shortest path model. This is because for each fragment, the purpose is to obtain a smooth representation of the catheter fragment. With known start and end points, this smoothing problem can be easily solved by finding the shortest path. In order to combine the two models together, a two layer hierarchical graphical model is proposed in which the top layer determines how to organize the catheter fragments, and the bottom layer determines how to achieve the best representation of each fragment.

The complete tracking system can run at a relatively fast speed but not in real-time (the real-time requirement is 10 fps at least for ultrasound images) on the platform on which we have carried out our experiments. However, since the computation for each pixel or voxel is independent, the system could easily be accelerated using GPU programming techniques. The computation time per frame is given by $t = \max(t_{xray}, t_{reg}) + t_{ultrasound}$. Here t_{xray}, t_{reg} and $t_{ultrasound}$ represent the computation times for the catheter extraction in the X-ray images, registration and catheter extraction from the ultrasound images, respectively. The extraction in the X-ray images and the registration can be carried out simultaneously. Often, the registration does not change significantly throughout the sequence, since the TEE probe is approximately stationary during the ultrasound acquisition. This means that for one frame, the registration transformation obtained from the first frame can be used, and the recalculation of the registration is unnecessary. Thus the total time depends mainly on the sum of t_{xray} and $t_{ultrasound}$. For t_{xray}, the average computation time is currently 0.03s, with nearly all of this time spent on the computation of candidate measurements in the search region. During this step, nearly 10^2 pixels in the search region are examined independently. These can be processed simultaneously. Thus t_{xray} could potentially be reduced significantly using GPU acceleration. Meanwhile, the potential improvement for the catheter extraction in the ultrasound images is even larger, given that the ultrasound image is a 3D volume. The different stages of the catheter extraction in the ultrasound images, the pre-processing and feature extraction, on both the original volume and the straightened volume, take 0.7s in total. This can be fully parallelized by examining each voxel independently and simultaneously on a GPU. The remainder of the computation, such as the catheter path growing, cannot be fully parallelized and require around 0.7s. However, the main cost is the calculation of feature measurements for each voxel and these computations could be carried out in advance and theiresults stored in memory. The average total time is $t_{ultrasound} = 1.3s$ using a CPU implementation. Given the easily parallelizable nature of the proposed algorithm, the required speed of 10 frames per second may well be achievable using a clever GPU or multi-GPU implementation.

Finally, the temporal consistency, which is a prior information from previous frames, is not used to constrain the search space of the next frame for ultrasound sequences. Currently, given one frame of ultrasound data, the algorithm does not take the previous location of the catheter into account. The catheter extraction for each ultrasound frame is carried out completely independently. In the next chapter, we introduce a framework of catheter tracking using only ultrasound sequences. Information from previous frames is used to provide constraints to the search space of the current frame.

Chapter 5

Real-time Catheter Tracking using Ultrasound without Fluoroscopy

5.1 Introduction

In this chapter we aim to develop a tracking system using ultrasound images only. Such a tracking system purely based on ultrasound images assumes temporal coherence of the catheter segmentation throughout an image sequence. Results from previous frames can be used to initialize the localization process in following frames instead of using the position of the catheter from corresponding X-ray images. Because X-ray assisted segmentation is no longer required for each frame, X-ray radiation exposure can be significantly reduced. However, the challenges of this approach are the severe interference of acoustic artefacts. A large number of false positives and false negatives in the catheter detection must not compromise the tracking accuracy. Therefore, the optimization framework needs to integrate additional constraints from both the catheter model itself and from the consistency of subsequent frames. This can be addressed by the fast discrete optimization framework proposed in [86]. Although this framework has been used for guide-wire tracking in X-ray sequences achieving satisfying results, the extension to ultrasound images has to overcome one main issue: computational latency.

The catheter is modelled as a B-spline represented by a set of control points. The discrete optimization framework defines a search range for each control point's location based on their previous location. The trade-off between the size of the search range and the computational cost indicates that the search range must be constrained in order to meet real-time requirements. However, if the search range for each control point is limited, then the curve tracking cannot recover rapid changes in the shape or the visible length of the catheter. Usually the search range covers a larger area in the transversal direction to the curve than in the longitudinal direction. Thus, catheter tracking can track transversal movements better than longitudinal movements, which causes the length of the catheter in the field of view to vary. The problem of computational cost becomes more severe when the imaging modality is extended from 2D X-ray to 3D ultrasound.

Besides this trade-off between the search range and the computational cost, another challenge is that fast catheter motion may cause parts of the catheter to move out of the search range in some frames. Tracking will fail in these cases. Thus, an online tracking failure detector is needed to re-initialize the tracking at the right time. The re-initialization can be done either manually or via the previously proposed X-ray assisted catheter segmentation. However, neither of these methods are desirable w.r.t. a minimal manual intervention effort and minimal X-ray exposure. Thus, in order to reduce the number of re-initializations, a search range should be chosen as close as possible to the true position of the catheter. Therefore a method of motion estimation has been developed to assist the prediction of catheter positions.

In this chapter, a real-time framework, which combines a fast discrete optimization for catheter fitting and a Kalman filter for online motion estimation, is proposed to address the problem of catheter tracking in ultrasound sequences for cardiac catheterization. The main algorithm consists of three main steps: (1) tubular structure measurement obtained in the region of interest; (2) discrete optimization for curve fitting given the tubular measurements; and (3) a Kalman filter for motion estimation and location prediction in the next frame. From the perspective of our application, the first and most important contribution is a system, which can track the catheter in an ultrasound sequence in real-time, with acceptable accuracy for the first time. Compared with the previous work in the last chapter, additional clinical data are added to evaluate it. From an algorithmic perspective, the discrete optimization framework is decoupled into two stages, transversal motion tracking and longitudinal motion tracking. The transversal motion tracking is addressed with the original optimization strategy in [86] with a search range defined in the plane normal to the curve. To address longitudinal motion, a new definition of the optimization strategy is developed (details are given in the Section 5.3.4). Additionally, a motion model is developed to estimate periodic motion of the heart. This is incorporated into the Kalman filter, which is used to accurately and robustly predict catheter positions in subsequent frames.

5.2 Measurements in the Region of Interest

5.2.1 B-spline Curve Representation of the Catheter

A cardinal B-spline $\mathbf{S}(t) = \sum_{i=0}^{n-1} \mathbf{c}_i B_i(t)$ is used to model a catheter using only one independent variable t. The basis functions $B_i(t)$ are defined for a range of $[-t_{margin}, t_{range} + t_{margin}]$ with $[0, t_{range}]$ corresponding to the catheter including a margin $\pm t_{margin}$ allowing for longitudinal movements. The parameters of the B-spline are the coordinates of the n control point locations $\mathbf{c}_i, i = 0, \ldots, n-1$.

5.2.2 Region of interest definition

To accelerate the tracking algorithm only voxels that are close to the approximate location of the catheter in a given frameare extracted for further processing. In order to extract the region of interest, the locations on the curve are first sampled over the range $[-t_{margin}, t_{range} + t_{margin}]$. Assuming we need m measurements on the curve, the sampling step can be defined as $\triangle t_d = (t_{range} + 2t_{margin})/(m-1)$. Thus the m sampled locations on the curve are $\{\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_{m-1}\}$ where $\mathbf{S}_j = \sum_{i=0}^{n-1} \mathbf{c}_i B_i(-t_{margin} + j\triangle t_d)$. A global longitudinal direction of the catheter, υ_{tan}, is defined as $\upsilon_{tan} = \mathbf{S}_{m-1} - \mathbf{S}_0$. For each sampled location, $\mathbf{S}_j, j = 0, \ldots, m-1$, a plane, $\mathbf{P}_j$, is defined containing this location with transversal direction υ_{tan}. The voxels on $\mathbf{P}_j$ that are in a defined neighbourhood of $\mathbf{S}_j$ are selected as a part of a region of interest (ROI). This ROI is converted to a Cartesian representation by defining a search range, R_{roi}, and the interval $\triangle l_{roi}$. The voxel at plane $\mathbf{P}_j$ with the plane coordinates $(x_j \triangle l_{roi}, y_j \triangle l_{roi}), x_j, y_j \in [-R_{roi}, R_{roi}]$ is then selected relative to the original $\mathbf{S}_j$ as a voxel in the Cartesian ROI with new the coordinates $(x_j+R_{roi}, y_j+R_{roi}, j)$. As a result, the surrounding voxels, which comprise the region of interest are unwarped along the main longitudinal direction υ_{tan}. This forms a new Cartesian space $V_{unwarped}$ from the original space $V_{original}$. Fig. 5.1 illustrates this process.

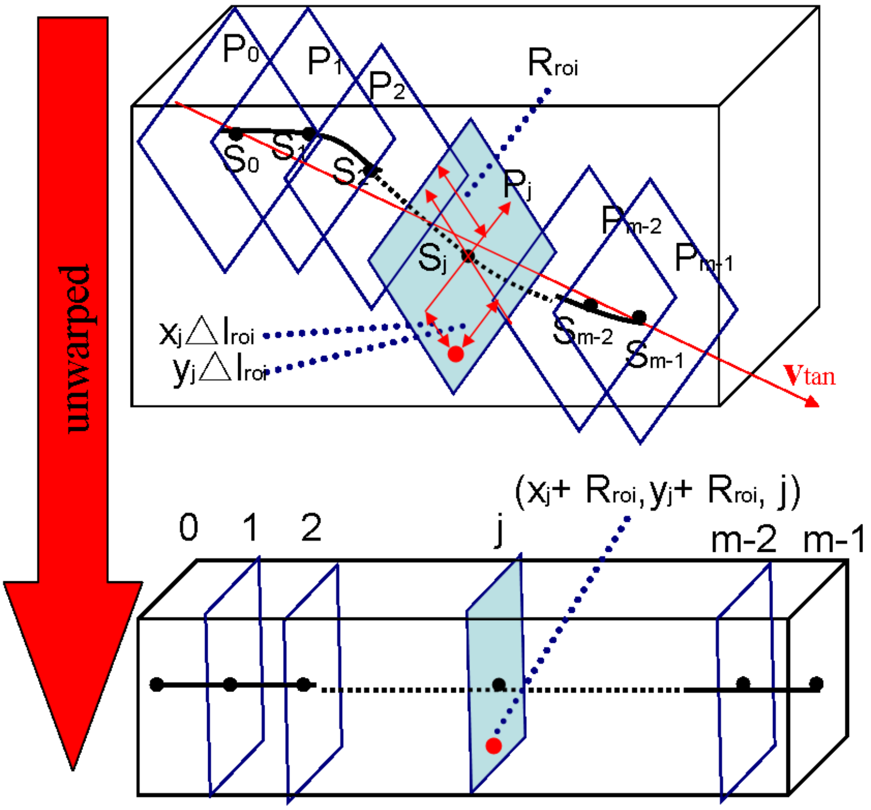


Figure 5.1: Search space definition and extraction of the ROI after unwarping.

5.2.3 Measurement of tubular structures

In our application, tubular structures in $V_{unwarped}$ are enhanced by applying the Frangi vesselness filter [69]. The thickness of the catheter can be used as prior information and as scale parameter for the vesselness filter. The scale is represented by standard deviation σ_{blur} of the Gaussian kernel used in Frangi's vesselness filter. Firstly, we blur the image using $V_{blur} = V_{unwarped} * G(\mathbf{0}, \sigma_{blur})$. Here $G(\mathbf{0}, \sigma_{blur})$ is a 3D Gaussian kernel with zero mean and standard deviation σ_{blur}. Since the Gaussian kernel is linearly separable, the 3D convolution can be efficiently calculated via consecutive 1D convolutions. The Hessian matrix is calculated from the blurred image V_{blur} and the Frangi's vesselness value is then computed using a difference operation:

$$\mathbf{H} = \begin{pmatrix} \triangle_{xx} V_{blur} & \triangle_{xy} V_{blur} & \triangle_{xz} V_{blur} \\ \triangle_{yx} V_{blur} & \triangle_{yy} V_{blur} & \triangle_{yz} V_{blur} \\ \triangle_{zx} V_{blur} & \triangle_{zy} V_{blur} & \triangle_{zz} V_{blur} \end{pmatrix} \tag{5.1}$$

where, e.g., $\triangle_{xy} V_{blur}$, is the second order differential operation on neighbouring pixels in V_{blur} along the x and y index directions. Other entries are defined in a similar way. This operation involves only each individual pixel with its adjacent pixels and thus the cost of the calculation is lower than for the Gaussian convolution.

The eigenvalues of the Hessian matrix are denoted by λ_1, λ_2 and λ_3 satisfying $|\lambda_1| < |\lambda_2| < |\lambda_3|$. The vesselness of tubular structures in the 3D volume are then enhanced by the function:

$$V_{tub} \quad = \quad \begin{cases} 0, & \text{if } \lambda_{2\ or\ 3} > 0 \\ \exp(-\frac{2\lambda_1^2}{|\lambda_2 \lambda_3|})(1 - \exp(-\frac{2\lambda_2^2}{\lambda_3^2}))(1 - \exp(-\frac{\lambda_{norm}^2}{2c^2})), & \text{otherwise} \end{cases} \tag{5.2}$$

where $\lambda_{norm}^2 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$. As described in Frangi's paper [69], the predefined parameter c depends on the intensity range of the image. Half the value of the maximum Hessian norm works in most cases. All above operations can be parallelized and implemented using our GPU acceleration platform.

5.3 Catheter Tracking based on Two-Stage Fast-PD Optimization

5.3.1 Overview of the Fast-PD Framework

The original Fast-PD framework was introduced in [86]. Here, we explain this framework for a better understanding of both the original idea and the improvements we have made.

In the original work, the problem of curve tracking is transformed to a multi-label assignment problem on a Markov Random Field (MRF), which is modelled as a graph. This problem can be solved by discrete optimization. Indeed, the curve, which is represented as a B-spline, corresponds to the graph $G =< V, E >$, with the control points of the B-spline as the nodes V and the links between adjacent control points as the edges E.

For a curve, the B-spline form is defined as:

$$\mathbf{S}(t) = \sum_{i=0}^{n-1} \mathbf{c}_i B_i(t),\ t \in [t_{min}, t_{max}] \tag{5.3}$$

where the n control points $\mathbf{c}_i, i = 0, \ldots, n-1$ belong to V. These control points determine the location and the shape of the curve. Thus, each control point's location is constrained within a pre-defined search range. Each pixel in the local search range corresponds to a label in a label space. The localization of the curve is then transformed to a multi-label assignment problem for each node in the graph model. The multi-label problem has been intensively studied in the field of computer vision within the context of segmentation problems. Discrete optimization is recommended as a fast solution to this category of problems. In [86], the Fast Primal-Dual (fast-PD) algorithm is employed because of its efficient computation. The objective function is defined as:

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) = (1-\gamma) \sum_{p \in V} (u_p(x_p)) + \gamma \sum_{p,q \in E} (w_{p,q}(x_p, x_q)), \tag{5.4}$$

Here $\mathbf{x}$ denotes a vector consisting of the label assignment to each node in V. x_p, is the pth element of the vector $\mathbf{x}$. It denotes the particular label assigned to node p. The objective function consists of an unary term $u_p(x_p)$ and a pairwise term $w_{p,q}(x_p, x_q)$ with a weighting term γ. The challenge is to define these two terms and to introduce constraints from both the tubular structure measurements and the physical properties of the catheter such as smoothness.

We begin with the definition of the terms in the objective function in e.q. (5.4) as given in [86]. The objective function consists of only pairwise terms, which are of the form:

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) = \sum_{i=0}^{n-2}((1-\lambda_1)C_1(x_i, x_{i+1}) + \lambda_1 C_2(x_i, x_{i+1})), \tag{5.5}$$

Here the labels x_i and x_{i+1} are assigned to the adjacent control points i and $i+1$. The data term $C_1(x_i, x_{i+1})$ measures the tubular enhancement at the location of the curve and the regularization term $C_2(x_i, x_{i+1})$ preserves the length of the catheter.

When the labels x_i and x_{i+1} are assigned to control points i and $i+1$, then any other control point j's label is also assigned to $x_j = x_i$ if $j < i$ or $x_j = x_{i+1}$ if $j > i+1$. The reason for this predefined rule is to make other control points' movements consistent with i and $i+1$. The local coordinate of the control point i, which corresponds to label x_i, is denoted by $\triangle\mathbf{c}_i(x_i)$. The updated new B-spline is then defined as:

$$\hat{\mathbf{S}}(t) = \mathbf{S}(t) + \sum_{j=0}^{i} \triangle\mathbf{c}_j(x_i)B_j(t) + \sum_{j=i+1}^{n-1} \triangle\mathbf{c}_j(x_{i+1})B_j(t) \tag{5.6}$$

Observations, or *landmarks*, are then sampled densely in the range $[t_{min}, t_{max}]$ with a step size of $\triangle t$. The k-th landmark corresponds to $t_k = \min(t_{min} + k\triangle t, t_{max})$ with $k = 0, \dots, m$ where $m = \lceil (t_{max} - t_{min})/\triangle t \rceil$. Finally, the pairwise data term is defined as:

$$C_1(x_i, x_{i+1}) = \frac{\sum_{k=0}^{m} N_{i,i+1}(t_k)(1 - V_{tub}(\hat{\mathbf{S}}(t_k)))}{\sum_{k=0}^{m} N_{i,i+1}(t_k)}, \tag{5.7}$$

where $N_{i,i+1}(t_k)$ denote weights calculated by:

$$N_{i,i+1}(t) = \frac{B_i(t) + B_{i+1}(t)}{\sum_{u=0}^{n-2}(B_u(t) + B_{u+1}(t))} \tag{5.8}$$

Using only image-based constraints has the problem that the tracked curve can easily collapse into a single point [86]. To address this problem, a length preserving regularization term

$$C_2(x_i, x_{i+1}) = \frac{\sum_{k=0}^{m} N_{i,i+1}(t_k)\rho(\|\hat{\mathbf{S}}^{'}(t_k)\|, \|\mathbf{S}^{'}(t_k)\|)}{\sum_{k=0}^{m} N_{i,i+1}(t_k)} \tag{5.9}$$

is defined in the original work [86]. Here $\hat{\mathbf{S}}^{'}(t_k)$ and $\mathbf{S}^{'}(t_k)$ are the first derivatives of $\hat{\mathbf{S}}(t_k)$ and $\mathbf{S}(t_k)$ respectively. The measurement function $\rho(x, y)$ forces x to be equal to y:

$$\rho(x, y) = (1 - x/y)^2 \tag{5.10}$$

Thus the length preserving function of this term is realized by forcing the 2-norm of the derivative vector of the new B-spline at each landmark to be equal to the one of the previous B-spline.

5.3.2 3D Catheter Tracking Framework

The previously described framework is not the original contribution of ours. Our work is based on it and improvements are made for both transversal and longitudinal motion tracking.

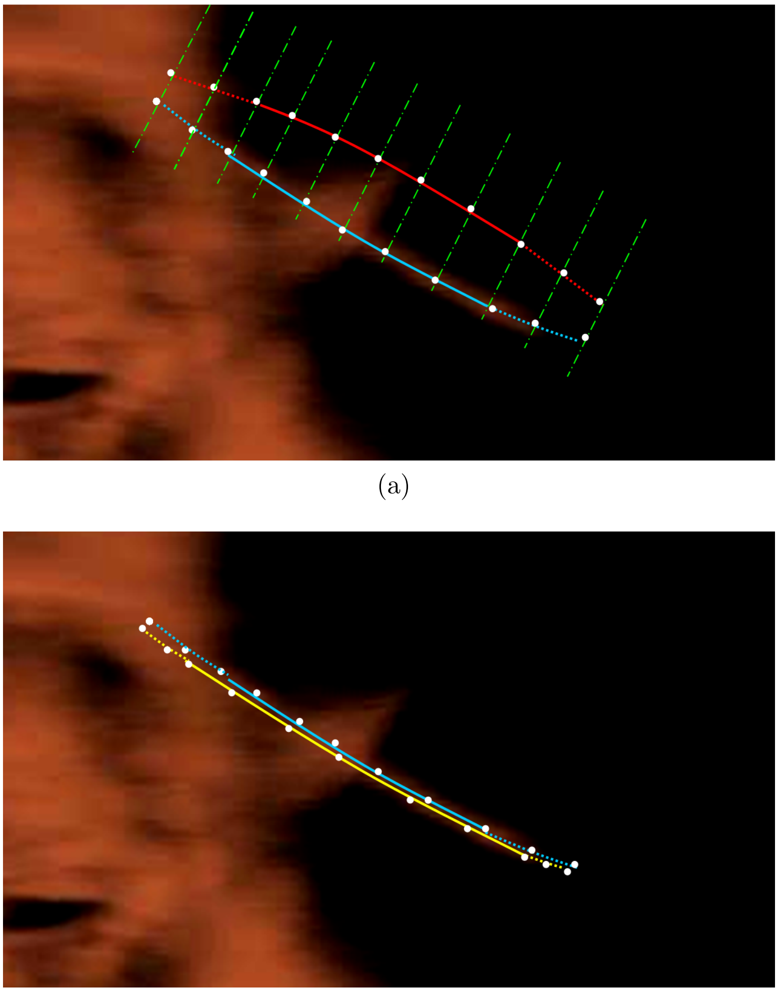
In [86] catheter tracking is performed in the 2D X-ray image and thus the label space is defined on a local 2D patch. However, when this strategy is directly extended to a 3D image, the computational complexity increases from $O(n^2)$ to $O(n^3)$. This makes real-time tracking impossible. Thus, we decouple the search space in 3D for each control point location into two sub-spaces: one 2D search range in the normal plane and one 1D search range in the longitudinal direction. The search range in the direction of the normal plane for each control point is limited in a predefined range $R_{fastpd1}$ with $\triangle l_{fastpd1}$, which is the step size of the measurements taken in this range. This is perpendicular to the main longitudinal direction of the catheter, which corresponds to the z-axis in V_{tub}. After the localization of the normal direction, the catheter position in the current frame should be almost correct and transversal motion should be successfully tracked. However, the catheter's end positions may be wrong, especially when large longitudinal movements occur. A second-stage optimization is carried out to adjust the curve in the 1D longitudinal direction and to track this additional motion. The optimization framework consists of: (1) tracking of the catheter in the transversal direction and (2) tracking in the longitudinal direction. As shown in Fig. 5.2, the red curve denotes the reference location defining the search space. The blue and yellow curves denote the result after tracking in the transversal and longitudinal directions respectively. The dots on the curves denote the distribution of the knots of the B-spline curve. The solid curves represent the sections of tubular structures while the dotted curves represent non-tubular structures. The computational costs of the transversal and longitudinal direction tracking processes are expected to be $O(n^2)$ and $O(n^1)$ respectively. Thus, this decoupling strategy reduces the computational cost from $O(n^3)$ to $O(n^2)$, which indicates its potential for real-time applications.

5.3.3 Transversal Catheter Tracking

The optimization for transversal motion tracking is based on the ideas of [86]. One of the key changes is that the search range for each control point is restricted to a plane normal to the longitudinal direction of the curve. The range $R_{fastpd1}$ is taken from the previous location of the corresponding control point. Another change is that when modelling the catheter as a B-spline curve $\mathbf{S}(t)$, the independent variable t is defined in $[-t_{margin}, t_{range} + t_{margin}]$ rather than in $[0, t_{range}]$. The range $[0, t_{range}]$ corresponds to the entire length of the catheter. $t_{min} = -t_{margin}$ and $t_{max} = t_{range} + t_{margin}$. The margin $\pm t_{margin}$ is included to allow for compensation of longitudinal movement. However, as one of our most important contributions, the longitudinal tracking optimization framework introduces an improved definition of the optimization framework in [86]. It is introduced in the next section.

5.3.4 Longitudinal Catheter Tracking

However, while transversal tracking can provide a good overlap with the tubularity measurements, the catheter's tips may be misaligned because of large longitudinal motion. In other words, if the range $[0, t_{range}]$ is not exactly aligned with the catheter positions $\mathbf{S}(0)$ and $\mathbf{S}(t_{range})$ the second stage optimization dealing with the longitudinal direction is required. However, the part of the curve derived from the tubular measurements has the same shape as the true



(a)

(b)

Figure 5.2: Tracking of transversal (a) and longitudinal (b) motion.

catheter. Any changes of the control point locations, no matter in which direction, are likely to change this shape. Thus the optimization representation in [86] is not appropriate for this case and it needs further improvements.

The overall longitudinal objective function to be optimized is:

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) = \sum_{i=0}^{n-2}((1-\lambda_2)C_1'(x_i, x_{i+1}) + \lambda_2 C_2'(x_i, x_{i+1}) + C_3'(x_i, x_{i+1})). \tag{5.11}$$

The definition of the three terms are explained below.

The curve is represented as a B-spline model which is a function $\mathbf{S}(t)$ of t over the range $[-t_{margin}, t_{range} + t_{margin}]$. One value of t within the range corresponds to the location of one point on the curve. The location and the shape of the entire curve depends only on the control points and it is independent of the variable t. Thus we can relocate the end points of the curve and the positions in between without changing the shape of the curve by constructing a new variable $t\prime$. The new generated curve $\mathbf{S}(t\prime)$ is expected to have the same shape but a different landmark distribution with the original curve $\mathbf{S}(t)$. Now we introduce how to construct the new variable $t\prime$.

The jth of m sampled observation points on the curve is located at $t = t_j = -t_{margin} + j\triangle t$. We assume the values of t_j are generated from another 1D cardinal B-spline $t(\tau) = \sum_{i=0}^{n-1} t_i^c B_i(\tau)$ and assume:

$$t_j = \sum_{i=0}^{n-1} t_i^c B_i(\tau_j). \tag{5.12}$$

Given that the values of t_j and τ_j are evenly sampled along the 1D axis τ, the n control points t_i^c can be calculated by solving linear equations. In this stage we are adjusting each control point t_i^c over a range $x_i \in [-T, T]$ where $[-T, T]$ is the label space and each x_i corresponds to one label in this space. The value of each x_i is selected every $\triangle l_{fastpd2}$ pixels in the range $[-T, T]$. The new $t\prime$ computed by the adjusted control points is then defined as:

$$t^{'}(\tau) = \sum_{i=0}^{n-1} (t_i^c + x_i) B_i(\tau). = t(\tau) + \sum_{i=0}^{n-1} x_i B_i(\tau) \tag{5.13}$$

When labels x_i, x_{i+1} are assigned to control points i and $i+1$, any other control point k's label is assigned to $x_k = x_i, k < i$ or $x_k = x_{i+1}, k > i+1$.A new $t\prime$ can be generated as:

$$t^{'}(\tau, x_i, x_{i+1}) = t(\tau) + \sum_{k=0}^{i} x_i B_k(\tau) + \sum_{k=i+1}^{n-1} x_{i+1} B_k(\tau) \tag{5.14}$$

The discrete form of $t\prime$ is denoted by:

$$t_j^{'}(x_i, x_{i+1}) = t^{'}(\tau_j, x_i, x_{i+1}) = t_j + \sum_{k=0}^{i} x_i B_k(\tau_j) + \sum_{k=i+1}^{n-1} x_{i+1} B_k(\tau_j) \tag{5.15}$$

A pair of labels x_i and x_{i+1} correspond to a curve of the form:

$$\mathbf{S}(t^{'}(\tau, x_i, x_{i+1})) = \sum_{i=0}^{n-1} \mathbf{c}_i B_i(t^{'}(\tau, x_i, x_{i+1})). \tag{5.16}$$

Substituting (5.14) into (5.16) gives:

$$\mathbf{S}(t^{'}(\tau, x_i, x_{i+1})) = \sum_{i=0}^{n-1} \mathbf{c}_i B_i(t(\tau) + \sum_{k=0}^{i} x_i B_k(\tau) + \sum_{k=i+1}^{n-1} x_{i+1} B_k(\tau)). \tag{5.17}$$

Note that the jth sample t_j corresponds to the independent variable τ_j satisfying (5.12). $C_1^{'}(x_i, x_{i+1})$ is then changed to incorporate the vesselness measurements:

$$C_1^{'}(x_i, x_{i+1}) = \frac{\sum_{j=0}^{m-2} N_{i,i+1}(t^{'}(\tau_j, x_i, x_{i+1}))G(\mathbf{S}(t^{'}(\tau_j, x_i, x_{i+1}))).}{\sum_{j=0}^{m-2} N_{i,i+1}(t^{'}(\tau_j, x_i, x_{i+1}))} \tag{5.18}$$

$G(\mathbf{S}(t))$ is defined as:

$$G(\mathbf{S}(t)) \quad = \quad \begin{cases} 1 - V_{tub}(\mathbf{S}(t)), & \text{if } 0 \leq t \leq t_{range} \\ V_{tub}(\mathbf{S}(t)), & \text{otherwise.} \end{cases} \tag{5.19}$$

Here $N_{i,i+1}(t)$ weights the corresponding term in a normalized way and ensures the measurements close to the nodes $i, i+1$ have higher weights. The definition was introduced in e.q. (5.8). $V_{tub}(\mathbf{S})$ denotes the normalized intensity of location $\mathbf{S}$ in the vesselness image. The purpose of the piecewise definition of $G(\mathbf{S}(t))$ is in order to guarantee that the curve section corresponding to $[0, t_{range}]$ is located on tubular structures and the sections of the curve between $[-t_{margin}, 0]$ and $[t_{range}, t_{range} + t_{margin}]$ are located on non-tubular structures.

The regularization term $C_2^{'}(x_i, x_{i+1})$, which ensures that the longitudinal motions of two adjacent points on the catheter-curve are similar, is defined as:

$$C_2^{'}(x_i, x_{i+1}) = \frac{\sum_{j=0}^{m-2} N_{i,i+1}(t^{'}(\tau_j, x_i, x_{i+1}))\|\mathbf{\Gamma}(\tau_{j+1}, x_i, x_{i+1}) - \mathbf{\Gamma}(\tau_j, x_i, x_{i+1})\|}{\sum_{j=0}^{m-2} N_{i,i+1}(t^{'}(\tau_j, x_i, x_{i+1}))}, \tag{5.20}$$

where $\mathbf{\Gamma}(\tau_j, x_i, x_{i+1})$ is defined as:

$$\mathbf{\Gamma}(\tau_j, x_i, x_{i+1}) = \mathbf{S}(t_j^{'}(x_i, x_{i+1})) - \mathbf{S}(t_j). \tag{5.21}$$

This term enforces an equidistant spacing of points $t^{'}(\tau)$ in $[-t_{margin}, t_{range} + t_{margin}]$.

The final regularization term $C_3^{'}(x_i, x_{i+1})$ guarantees the new catheter tip's $t^{'}$ are within the range $[-t_{margin}, t_{range} + t_{margin}]$. It is defined as:

$$C_3^{'}(x_i, x_{i+1}) \quad = \quad \begin{cases} 0, & \text{if } -t_{margin} \leq t^{'}(0), t^{'}(t_{range}) \leq t_{range} + t_{margin} \\ \infty, & \text{otherwise.} \end{cases} \tag{5.22}$$

Using all three terms above, the energy function is defined as in e.q. (5.11), which can be quickly solved by the Fast-PD framework. Tracking failures can be detected by thresholding the average intensity value of the pixels in V_{tub} which correspond to the resulting curve. We use 0.15 as the threshold.

5.4 Kalman Filter Framework

Usually, the search range for the transversal motion tracking is limited (within $R_{fastpd1}$ as previously described). If the catheter's transversal movements are extensive and fast, a regular search range is unlikely to recover the catheter. However, although the size of the search range cannot be extended to fulfil real-time requirements, the location for the catheter can be predicted using another reference tracking localization.

In most cases, the reference location is simply the location of the catheter in the previous frame. This clue is important but not sufficient for predicting the next position. One solution is to estimate the reference location by online estimation of the motion pattern. As one of the most widely used techniques, the Kalman filtering technique, can update a state vector online which may be used to reflect the motion pattern.

In our application motion is mainly caused by periodic cardiac movements. Breathing motion can be ignored due to its small influence on the catheter motion in ultrasound images. This is because the TEE probe, which is in the patient's oesophagus, is synchronized with the catheters in the chest. Although both are affected by respiration, their relative position stays constant. As cardiac motion has a predictable periodic pattern, a periodic model is considered to fit it.

To simplify the tracking system and to reduce the number of parameters, the periodic motion is modelled in 2D rather than in 3D. The 2D plane is normal to the longitudinal direction of the catheter and corresponds to the transversal motion tracking stage. This is justified because we only track the transversal movement between the real and the projected catheter and ignore longitudinal motion at this stage. One common way to model periodic motion is to use two sinusoidal functions corresponding to the x and y axes. The motion trace in this 2D plane is actually an ellipse (Fig. 5.3 right). We first define the state vector $\mathbf{x}^k$ at frame k as follows:

$$\mathbf{x}^k = (l_1^k, l_2^k, \varphi^k, \phi^k, \omega^k)^T, \tag{5.23}$$

This consists of the magnitudes l_1^k and l_2^k for the 2D normal plane, the phase difference φ^k, the angular displacement ϕ^k , and the angular velocity ω^k. l_1^k, l_2^k and φ^k define the shape of the ellipse along its major/minor axis. At the same time, the measurements of the measurement model are defined as the 2D locations $\mathbf{z}^k = (z_1^k, z_2^k)$ of this periodic motion. $\mathbf{z}^k$ is defined as the transversal offset vector from the current position of the catheter to the next position.(Fig. 5.3). ϕ^k and ω^k define the instantaneous state of the vector $\mathbf{z}^k$, which is shown in green in Fig. 5.3 (right). In the following $\mathbf{E}_n$ denotes an $n \times n$ identity matrix. The Kalman filter system and measurement model is then defined as:

$$\mathbf{x}^{k+1} = \begin{pmatrix} \mathbf{E}_4 & \mathbf{v} \\ \mathbf{0} & 1 \end{pmatrix} \mathbf{x}^k + \mathbf{R}_{model}, \tag{5.24}$$

$$\begin{pmatrix} z_1^k \\ z_2^k \end{pmatrix} = \mathbf{\Phi^k} \begin{pmatrix} l_1^k \cos(\phi^k) \\ l_2^k \sin(\phi^k) \end{pmatrix} + \mathbf{E}_2 r_{measured}, \tag{5.25}$$

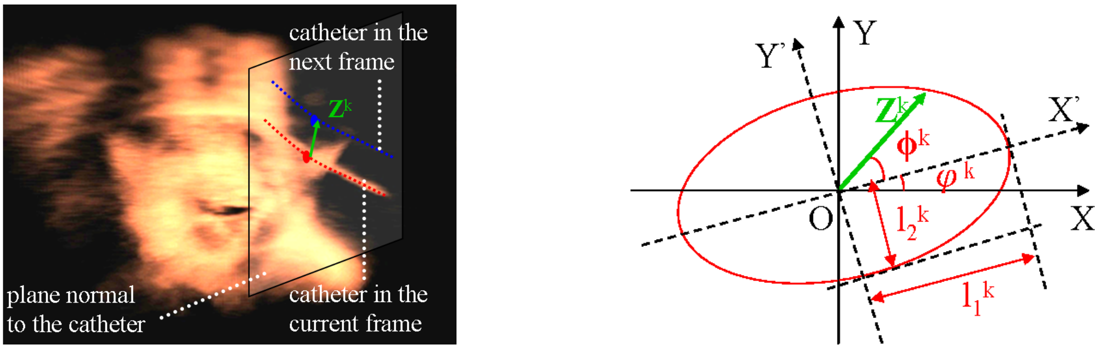

catheter in the next frame
Z^k
plane normal to the catheter
catheter in the current frame
Y'
Y
X'
X
O
Z^k
ϕ^k
φ^k
l_2^k
l_1^k

Figure 5.3: The definition of the state vector and the measurement model for the Kalman filter.

with

$$\mathbf{\Phi^k} = \begin{pmatrix} \cos(\varphi^k) & -\sin(\varphi^k) \\ \sin(\varphi^k) & \cos(\varphi^k), \end{pmatrix} \tag{5.26}$$

where $\mathbf{v} = [0, 0, 0, T]'$ is the sample time interval for T. $\mathbf{R}_{model}$ is the system covariance matrix and $r_{measured}$ indicates measurement errors, with $\mathbf{E}_2 r_{measured}$ as the measurement covariance matrix. When given an initial status vector $\mathbf{x}^0$ and its corresponding covariance $\mathbf{P}^0$, $\mathbf{x}^k$ and $\mathbf{P}^k$ a position for the kth frame can be obtained by the Kalman filter. This measurement model is motivated by the definition introduced in Fig. 5.3. In the left figure, a transversal plane is determined at the mid-point of the catheter with its normal direction along the longitudinal direction of the curve. Then the offset vector $\mathbf{z}^k$ (in green) is defined as the vector from the intersection of the current catheter (in red) with the transversal plane to that of the catheter's transversal plane in the next frame (in blue). In the right figure we model the offset vector $\mathbf{z}^k$ (in green) as a 2D periodic motion. Eq. (5.25) describes the transformation between two coordinate systems XOY $X'OY'$ (right).

However, a Kalman filter requires to specify the values of the state initialization $\mathbf{x}^0$, the state covariance $\mathbf{P}^0$, the system covariance matrix R_{model}, and the measurement error $r_{measured}$. Here we give a basic guide for the parameter choices. More details will be given in the next section.

One of our basic assumptions is that the tracked catheter is treated as the true location with a high confidence. Thus the measurement error $r_{measured}$ should be small. The cardiac motion can be modelled but does not exactly follow a regular periodic motion. The basic periodic parameters should vary within a limited range. The system covariance matrix is large but limited, depending on the actual motion in the data. More transversal motion will result in larger values in the matrix. The values of $\mathbf{x}^0$ and $\mathbf{P}^0$ can be determined from prior information such as the motion period and the motion magnitude. However, if the values of both $\mathbf{x}^0$ and $\mathbf{P}^0$ are reasonable (i.e. consistent with the prior information), their influence on the performance will decrease during the tracking process.

Before a new catheter location is computed with the Fast-PD optimization, the new states are first predicted via Eq.(5.24) followed by a prediction of the offset $\mathbf{z}^{k,k+1}$ with Eq.(5.25). When the location of the catheter in the previous frame is projected to the current frame, the longitudinal direction is approximated by calculating the difference vector between the location of the two ends of the curve. The normal vector of the transverse plane is then aligned with the longitudinal direction (Fig. 5.3). The new initial catheter estimate is obtained by moving the projected location in the transversal plane with the offset $\mathbf{z}^{k,k+1}$. After Fast-PD optimization, the new tracked catheter is used to update the offset $\mathbf{z}^{k+1}$. This new offset is then used to update the status vector through the standard extended Kalman filter framework. The transversal plane is also updated based on the new location of the catheter.

5.5 Experiments

The platform used for all experiments was a Linux PC with a 3.40GHz processor, 8GB RAM and a 384-core, 2GB Nvidia GeForce GT 640 graphics card for GPU acceleration. The image pre-processing, tubular enhancements, and the inputs for both transversal and longitudinal discrete optimization have been parallelized for computation on the GPU However, the core optimization of Fast-PD, as well as the Kalman filter module, are not ideal for parallelization because of dependencies between sub-steps. However, they can be computed efficiently on the CPU. The Compute Unified Device Architecture (CUDA) programming model is used for GPU programming. The GPU applications are compiled with the Nvidia CUDA Compiler (NVCC) with default options. The GPU applications are optimized for maximum parallelization since the potential improvements in speed based on GPUs is expected to be dramatically larger than

CPUs. For GPU the applications, in order to improve the instruction throughput, floating point data types are replaced by long integer data types for image intensity representations. The precision of the computation will not be affected since the long integer type is sufficient. In the tracking algorithm for ultrasound images, the main factors that affect the speed is the image processing and feature detection, as well as objective function calculation. However, these stages can be fully parallelized and these calculations are carried out on the GPUs.

The method was compared to several existing catheter tracking techniques. The main methods for comparison include our previous system proposed in [214, 215]. Another comparison method is the direct use of the optimization system in [86] for catheter tracking in ultrasound imaging. However, for real-time tracking, the label pool of this method is constrained to the transversal plane in the longitudinal direction of the catheter. Thus it is equivalent to the system we propose here without longitudinal tracking. We have not compared the direct use of the optimization system with a 3D search range, including the longitudinal direction, because this would cause the tracking system to be very slow and impossible to implement in a real-time application. The proposed methods with and without Kalman filter motion estimation were also compared to illustrate the influence of the motion estimation step. All algorithms were evaluated with different parameters. For the three Fast-PD based algorithms, which include the one with transversal tracking only, the one with both transversal and longitudinal tracking but without the Kalman filter, and the one with every module, each was evaluated on two setups: one (slow) setup with a large search range to evaluate its overall performance and another (fast) setup with a short search range to satisfy the real-time clinical requirement.

We defined the following performance metrics:

- **Average frame rate** The ratio of the total number of frames to the total time consumed is used to represent the average frame rate. It measures real-time capabilities;

- **Average tracking error** $err(\rho)$ For each catheter landmark i, the shortest distance d_i to the ground truth is calculated. A threshold ρ is used to select correctly tracked landmarks with $d_i \leq \rho$. The average d_i of correctly tracked landmarks is defined as the tracking error;

- **Incorrect tracking percentage (ITP)** $p_{itp}(\rho)$**:** This is defined as the ratio of the number of incorrectly tracked landmarks over the total number of landmarks on the tracking result. It measures the extent of false positives;

- **Failed tracking percentage (FTP)**$p_{ftp}(\rho)$**:** One landmark of the ground truth can be considered as failed if its minimal distance to the tracking result is above ρ. The FTP is defined as the number of incorrectly tracked landmarks divided by the total number of landmarks in the ground truth. It measures the system's ability to detect the whole catheter;

- **Overlap percentage** $p_{overlap}(\rho)$**:** This is the length of the overlap between the result and the corresponding ground truth as a percentage of the length of the union of the result and the ground truth;

- **Ratio of incorrectly tracked frames (ROF)** p_{rof}**:** The number of frames where a tracking failure is detected over the total number of frames.

5.5.1 Datasets

The data used to evaluate the method include 4 long (approximately 200 frames per sequence on average) clinical ultrasound sequences with corresponding X-ray fluoroscopic images, which is referred as Type A data, and 34 short clinical ultrasound sequences without X-ray images, referred as Type B. Both were acquired with a Philips X7-2t TEE probe, comprising more than 1800 frames in total. Various catheter types are included in these data. All data sets have a manual ground truth for the catheter position. These ground truths are manually marked by an expert. In the following experiments, the comparison is first made with methods [214, 215] based on Type A data. For other comparisons and evaluations, all of the data are used. In Table 5.1, we give details for each sequence.

To describe the extent of the catheter motion in these data, we compare two ground truths of two adjacent frames.

Table 5.1: Dataset Information

Seq.	**Size**	**Frames**	**Resolution (mm)**	**frames rate (fps)**	d_{cat}[a]	d_{nor}[b]	d_{tan}[c]
A1	144 × 160 × 208	216	0.60 × 0.60 × 0.30	14[d]	4.00	1.97	1.38
A2	144 × 176 × 208	216	0.51 × 0.50 × 0.30	14	4.58	1.39	2.12
A3	144 × 176 × 208	216	0.51 × 0.50 × 0.30	14	4.58	1.41	2.16
A4	144 × 176 × 208	216	0.51 × 0.50 × 0.30	14	4.58	1.35	2.70
B1	160 × 64 × 208	41	0.68 × 0.89 × 0.58	26	2.79	0.57	0.73
B2	160 × 64 × 208	55	0.56 × 0.75 × 0.48	30	3.35	0.37	0.15
B3	224 × 208 × 208	21	0.56 × 0.55 × 0.48	27	3.78	0.54	0.56
B4	160 × 64 × 208	65	0.56 × 0.75 × 0.48	30	3.34	0.42	0.27
B5	224 × 208 × 208	18	0.56 × 0.55 × 0.48	27	3.78	0.54	0.33
B6	192 × 176 × 208	17	0.35 × 0.33 × 0.24	12	6.56	2.29	2.34
B7	192 × 176 × 208	14	0.35 × 0.33 × 0.24	12	6.56	2.42	1.54
B8	192 × 176 × 208	15	0.38 × 0.36 × 0.24	12	6.11	2.37	2.09
B9	160 × 176 × 208	22	0.48 × 0.51 × 0.34	13	4.55	1.30	1.60
B10	128 × 128 × 208	21	0.61 × 0.65 × 0.43	17	3.54	1.32	1.94
B11	160 × 176 × 208	14	0.54 × 0.59 × 0.44	11	3.82	1.34	1.06
B12	160 × 64 × 208	49	0.69 × 0.89 × 0.58	26	2.77	0.69	0.61
B13	160 × 64 × 208	40	0.69 × 0.89 × 0.58	26	2.77	0.77	0.75
B14	160 × 64 × 208	49	0.57 × 0.75 × 0.48	30	3.33	0.92	0.64
B15	272 × 256 × 208	13	0.31 × 0.32 × 0.25	5	6.83	2.81	4.20
B16	224 × 208 × 208	31	0.56 × 0.55 × 0.48	27	3.78	0.71	1.26
B17	272 × 256 × 208	14	0.26 × 0.27 × 0.23	5	7.82	2.90	2.81
B18	224 × 272 × 208	16	0.35 × 0.32 × 0.30	6	6.18	2.56	3.78
B19	224 × 272 × 208	16	0.35 × 0.32 × 0.30	6	6.18	1.60	2.13
B20	160 × 64 × 224	22	0.74 × 0.97 × 0.63	24	2.56	1.31	0.99
B21	160 × 64 × 224	20	0.74 × 0.97 × 0.63	24	2.56	0.59	0.61
B22	160 × 64 × 224	30	0.74 × 0.97 × 0.63	24	2.56	0.91	1.75
B23	160 × 64 × 224	20	0.74 × 0.97 × 0.63	24	2.56	0.81	1.81
B24	160 × 64 × 224	26	0.74 × 0.97 × 0.63	24	2.56	0.52	1.03
B25	160 × 64 × 224	20	0.74 × 0.97 × 0.63	24	2.56	0.59	1.21
B26	160 × 64 × 224	29	0.74 × 0.97 × 0.63	24	2.56	0.71	1.82
B27	160 × 64 × 224	28	0.74 × 0.97 × 0.63	24	2.56	0.64	1.18
B28	160 × 64 × 224	45	0.74 × 0.97 × 0.63	24	2.56	0.79	1.54
B29	160 × 64 × 224	26	0.74 × 0.97 × 0.63	24	2.56	0.53	0.50
B30	224 × 144 × 224	29	0.35 × 0.35 × 0.31	12	5.92	1.52	2.23
B31	160 × 64 × 224	20	0.74 × 0.97 × 0.63	24	2.56	0.58	0.59
B32	160 × 64 × 208	25	0.81 × 1.05 × 0.68	23	2.36	0.76	1.24
B33	144 × 128 × 208	61	0.34 × 0.35 × 0.21	21	6.66	0.70	0.61
B34	144 × 128 × 208	60	0.34 × 0.34 × 0.21	21	6.72	0.64	0.49

[a] The estimated diameter of the catheter in voxels based on the estimated catheter diameter in mm (2mm) and the voxel size for each sequence.

[b] The estimated transversal motion per frame in mm.

[c] The estimated longitudinal motion per frame in mm.

[d] This figure for all Type A data is estimated based on prior information about the heart rate and the images. No direct frame rate can be read from the image header file.

The length of the ground truth is computed for every frame and the difference between two adjacent frames is a measure of the longitudinal motion of the catheter between these frames. The transversal motion error is calculated using the average tracking error performance metrics defined previously, except that all distances are averaged, not just those below a threshold. For each sequence, around 10∼20 consecutive frames were randomly sampled. Then the average length difference and the average transversal motion error are computed based on these sampled data.

5.5.2 Parameter Selection

During the stage of region of interest construction and the tubular structure enhancement, most parameters are independent of different dataset sequences and can be fixed. We have found that the catheter in all datasets can be modelled with $n = 10$ control points and $m = 40$ measurements. However, these two parameters of the catheter modelling can be adjusted for more complicated curves such as a longer length (more measurements (sampled landmarks) are needed) or a larger curvature (more control points are needed). All of the catheters are defined on a range of $t_{range} = 100$ with the margin $t_{margin} = 10$. The parameter $m = 40$ is also the length of the volume V_{tub}. The width and the height of V_{sub} are the same, with the value of $R_{roi}/\triangle l_{roi}$. In our application, we set $\triangle l_{roi} = 1$ and $R_{roi} = 15$ and these are fixed for every sequence. In the fast version of the algorithm mentioned above, $m = 20$ and $R_{roi} = 10$ are used.

The value of the parameter σ_{blur} is selected based on the diameter of the catheter in voxels, which can be considered as prior information. Regarding the search range for both transversal and longitudinal movements, the search ranges are $R_{fastpd1} = 5$ ($R_{fastpd1} = 3$ in the fast version) with the sample bin $\triangle l_{fastpd1} = 1$ for transversal movements, and $R_{fastpd2} = 10$ ($R_{fastpd2} = 5$ in the fast version) with the sample bin size $\triangle l_{fastpd2} = 1$ for longitudinal movements.

During the Fast-PD tracking stage, two parameters which cannot be determined by prior information are the weighting terms for the Fast-PD terms: λ_1 for transversal tracking and λ_2 for longitudinal tracking. For such parameters which cannot be determined based on prior information, we perform an exhaustive search in the range $0.1 \sim 0.9$ with a resolution of 0.1 for each sequence.

During the Kalman filter tracking stages, the parameters to be set include the state initialization $\mathbf{x}^0$ and state covariance $\mathbf{P}^0$, system covariance matrix R_{model} and measurement error $r_{measured}$. Assuming independence between different state elements, which causes $\mathbf{P}^0$ and R_{model} to be diagonal, results in 16 remaining parameters, which is a large number for tuning. To reduce the tuning space, we also make the following assumptions: we fix $\mathbf{x}^0$ and $\mathbf{P}^0$ since they have limited influence on the performance. We set l_1^0 and l_2^0 in $\mathbf{x}^0$ equal to the estimated transversal motion magnitude in Table 5.1 for each sequence. φ^0 is set to $\pi/2$ and ϕ^0 to zero. The angular velocity ω^0 is set according to the frame rate of each sequence, which approximates the value to $2\pi/n_{period}$. n_{period} is the approximate number of frames covering one cardiac period and can be estimated from the frame rate and the average cardiac heart rate. We set $\mathbf{P}^0$ equal to R_{model} since they correspond to the same variables in the state vector. With these assumptions, the number of parameters is reduced to 6. We also assume that l_1^k and l_2^k have the same covariance value q_1, and φ^k and ϕ^k also have equal covariance q_2, since they are intrinsically at the same quantity level. The angular velocity ω^k has a different covariance value q_3. Thus the total number of parameters that need to be adjusted has been significantly reduced to four: $(q_1, q_2, q_3, r_{measured})$. An exhaustive search based on the general evaluation function $f_{eval}(err(\rho), p_{overlap}(\rho), p_{rof})$ is defined as:

$$f_{eval}(err(\rho), p_{overlap}(\rho), p_{rof}) = 10err(\rho) + (1 - p_{overlap}(\rho)) + p_{rof} \tag{5.27}$$

The default values for $(q_1, q_2, q_3, r_{measured})$ are set to $(1.0, 0.7, 0.7, 0.06)$ for all sequences. Generally, we set $(\rho) = 5$ as this value is close to the limit of manual detection of incorrect tracking.

5.5.3 Parameter Sensitivity

According to the previous section, most parameters can be determined from prior information based on data properties. However, the weighting terms for Fast-PD tracking, which are λ_1

and λ_2 cannot be determined in this way. Thus we perform an analysis based on all values from the range $[0.1, 0.9]$ with a resolution of 0.1. Another set of parameters that require exhaustive search are $(q_1, q_2, q_3, r_{measured})$ for the Kalman filter. The prior information suggests default values of $(1.0, 0.7, 0.7, 0.06)$. We perform an analysis based on a range around these values. The evaluated performance metrics include the error $err(\rho)$, the overlap percentage $p_{overlap}(\rho)$, FTP, ITP, ROF and the computation time.

The first evaluation is the parameter λ_1 from 0.1 to 0.9. The algorithms with and without the transversal tracking stage are evaluated. Kalman filter based tracking is not evaluated here since it has no direct impact on the fast-PD optimization. For the variations with longitutinal tracking, λ_2 is set to an intermediate value of 0.5. From Fig. 5.4, it is obvious that including longitudinal tracking and a general definition of $\lambda_2 = 0.5$ achieves a better overlap percentage and lower FTP at any value of λ_1. This means that the improvement achieved by the longitudinal tracking approach is very robust even for various configurations of the transversal tracking method, $\lambda_1 = 0.1 \sim 0.9$. It is notable that the error is not improved since λ_1 does not control the accuracy. Other performances measures such as ITP and ROF are also not improved because they do not have a direct relationship with the longitudinal tracking module. The longitudinal tracking is used for extending the extraction to a catheter larger length, which is reflected in overlap ratio and FTP.

The second evaluation examines the other coefficient λ_2 from 0.1 to 0.9. As with the first evaluation, the Kalman filter module is not involved. For both, the algorithms with and without longitudinal tracking, the configuration of the transversal tracking module is set to $\lambda_1 = 0.5$. Given that the algorithm without longitudinal tracking is independent of λ_2, the resulting graphs for this algorithm (solid lines in Fig. 5.5) are horizontal lines which provide a default value for λ_2. From Fig. 5.5, we conclude that additional longitudinal tracking can achieve a robust improvement in overlap ratio and FTP with different values of λ_2. In other words, the proposed system is insensitive to changes of the λ_2 value. For other performance metrics, improvements are not guaranteed. This is similar to the conclusion from Fig. 5.4.

The last evaluation is made for the Kalman filter's parameter set $(q_1, q_2, q_3, r_{measured})$. In this

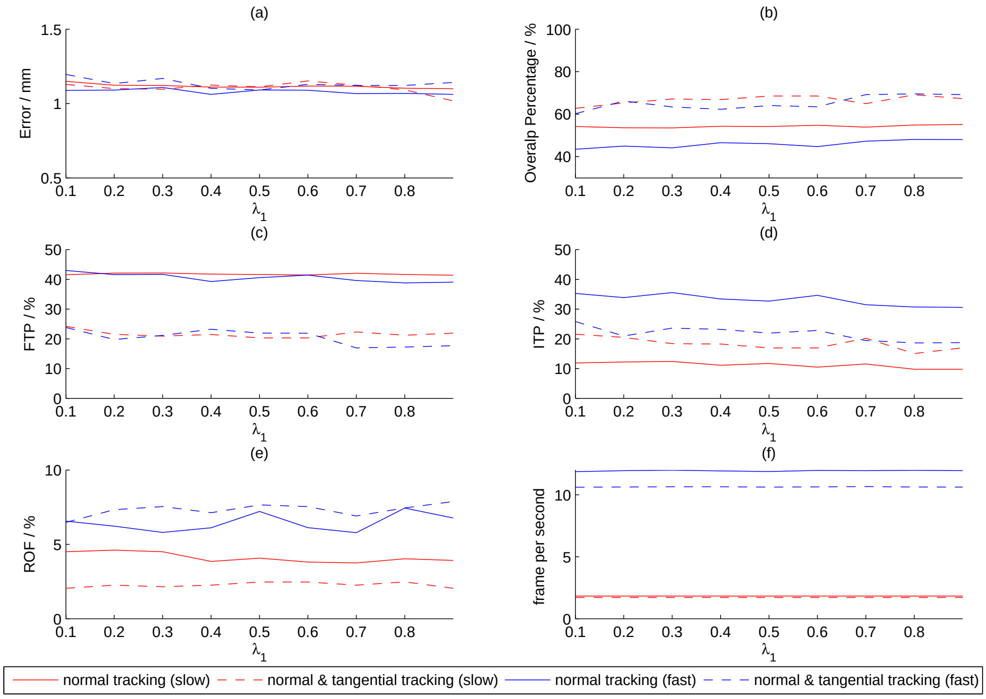


Figure 5.4: Sensitivity analysis for λ_1.

evaluation, only the algorithms with and without the Kalman filter module are evaluated. The longitudinal tracking module is used in all evaluated algorithms. The evaluation is performed in the range from $(1.0, 0.1, 0.1, 0.01)$ to $(10.0, 1.0, 1.0, 0.10)$ with a step size of $(1.0, 0.1, 0.1, 0.01)$. This range is chosen based on prior knowledge about the Kalman filter model and the error distribution. 64 values are randomly sampled in this range and the indices of these values are used for the horizontal axis. From Fig 5.6, the algorithms without the Kalman filter module (dashed lines), with $\lambda_1 = \lambda_2 = 0.5$, show horizontal lines since they are independent of the parameters. This horizontal line gives an evaluation reference. Note that the horizontal axis does not represent a particular dimension of the parameter vector $(q_1, q_2, q_3, r_{measured})$. It represents different combinations of all evaluated values in the vector. As this evaluation is only made to see the sensitivity, rather than to find the optimum value, the values of the parameter vector that the combination indices correspond to are not shown here. Generally we can see from the figure that the performance with the Kalman filter module is sensitive to changes in the parameters. The curve of the algorithms with the Kalman filter module show some peaks above the reference line but also some valleys below. This is not surprising since these parameters are strongly linked to the motion properties of each individual sequence. The ITP

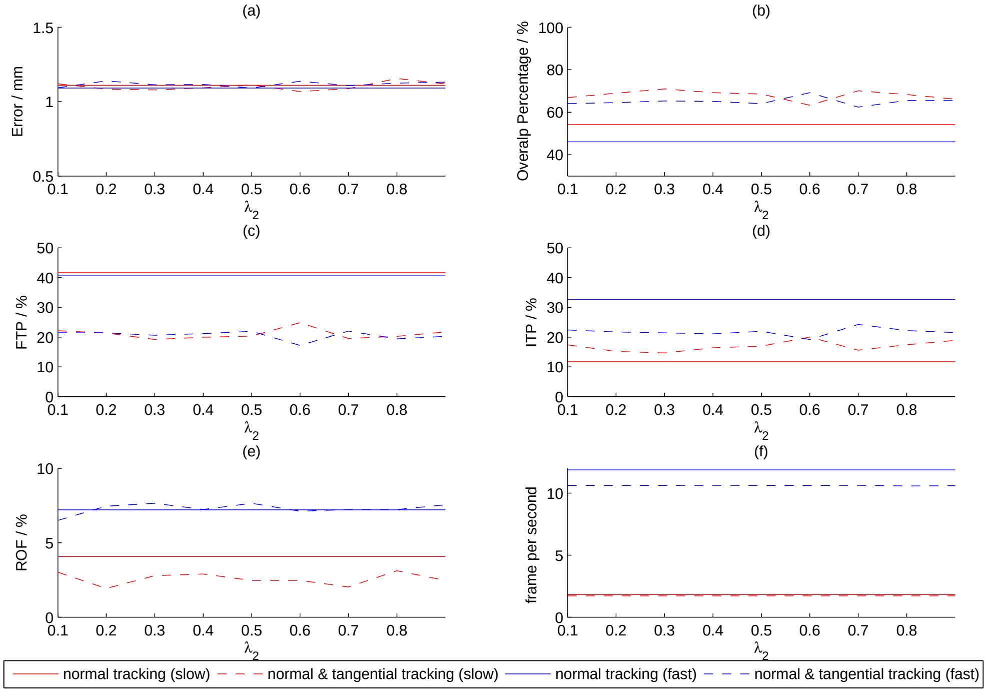


Figure 5.5: Sensitivity analysis for λ_2.

and FTP are slightly improved and furthermore the overlap ratio is also enlarged. The error is not improved by using the Kalman filter. The Kalman filter module can reduce the ROF and this is more obvious for the fast setup than the slow setup, since for the fast setup the algorithm with the Kalman filter achieves lower ROF more frequently than it achieves higher ROF. The reason for this is that the search range is limited for the fast setup. Thus the catheter is more likely to be out of range and hence the ROF will increase. In such cases, the Kalman filter's role is important to reduce the ROF.

5.5.4 Overall Performance Comparison

Comparisons were first made based on the four long sequences which have corresponding X-ray images. Here all of the algorithms can be compared at the same time. All methods are compared except the direct usage of Fast-PD without the longitudinal tracking and without Kalman filter. Additionally, there is one method from our previous work [214, 215] to be included. This method uses the X-ray tracking results to assist the catheter extraction in ultrasound images. However, this method does not account for the temporal consistency between frames. Fig. 5.7 shows a

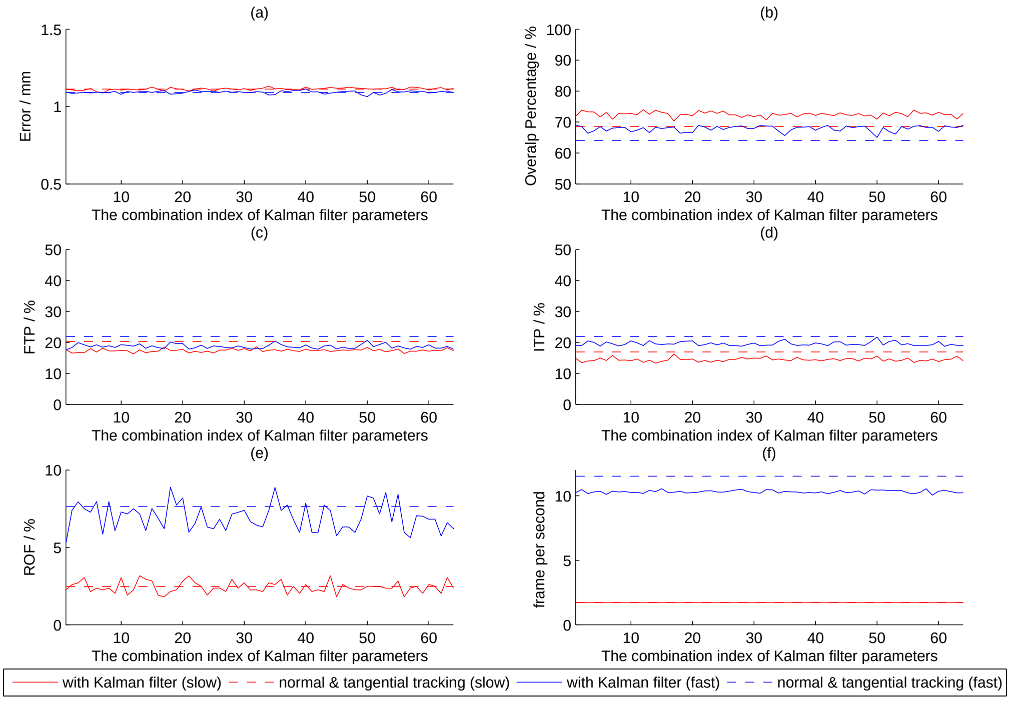


Figure 5.6: Sensitivity analysis on Kalman filter parameters.

comparison of the tracking error, ITP, FTP and overlap percentage for different ρ (1–15). The computation times for all approaches (in the same order as shown in Fig. 5.7) are 1.6, 11.8, 10.4, 10.5 and 1.35 fps. The ROFs of the first four approaches are 1.13, 6.57, 4.86 and 3.43%, respectively. It should be noted that the measured speed is calculated based on the time spent on the GPU plus the time spent on the CPU independently. The time needed for data transfer between GPU and CPU is not taken into account.

Another comparison was made without the method in [214, 215] so that all of the data are used, including the Type B data without corresponding X-ray images. Type B data cannot be used with our previous method [214, 215]. Fig. 5.8 shows a comparison of the performance metrics for different ρ (1–15). In the same order as in the figure legend, the computation times for all approaches are 1.7, 11.8, 10.5, and 10.3 fps and the ROFs are 2.18, 7.43, 7.21 and 4.09%, respectively. Note that the speed and the ROF are independent of ρ.

From these the comparison results we conclude that: (1) The proposed method is significantly faster, more accurate and more robust than the previous methods in [214, 215]. The reason is that the consistency between frames was not taken into account in the previous method causing

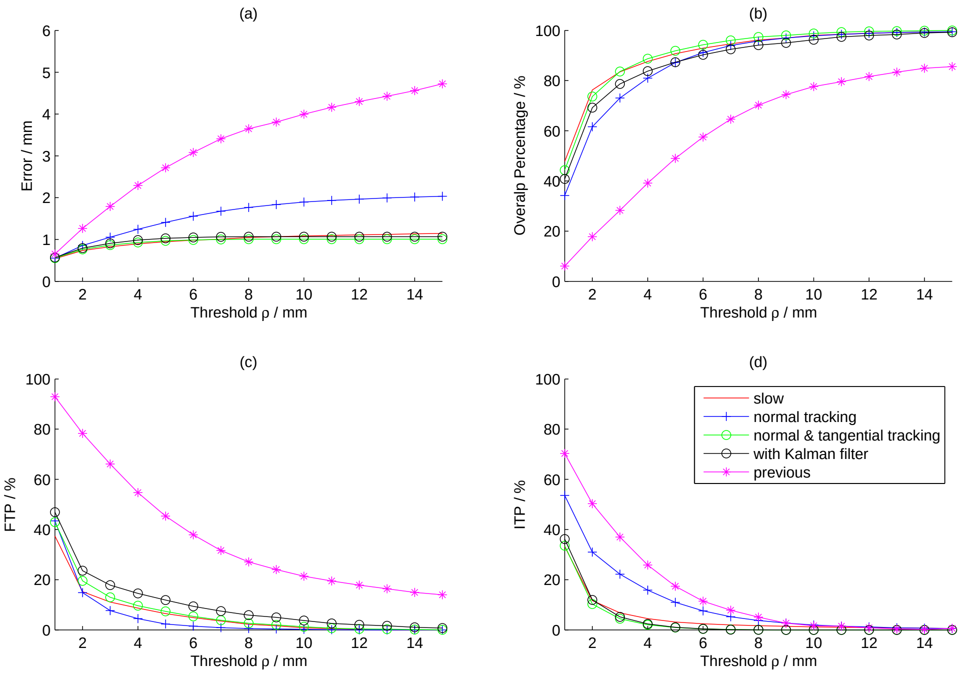


Figure 5.7: Comparison results only with Type A datasets.

a poor performance. (2) The proposed Fast-PD framework with longitudinal movement is more suitable to extract the entire length of the catheter with a smaller ITP, smaller FTP, and a larger overlap ratio. It can also improve the tracking accuracy considering the results from Type A datasets. However, the error has only reduced very slightly with Type B data. The reason is due to the length of the data sets. If the longitudinal tracking was not used, for data sets with more frames, the ITP percentage would eventually rise without longitudinal tracking. The resulting large proportion of incorrect landmarks would have a negative effect on the measured accuracy. Thus, the longitudinal tracking module can keep ITP percentage low and this affects the final accuracy indirectly. As most of the Type B data sets are short, this is not obvious. (3) With Kalman filtering, the ROF is reduced while other performance metrics do not significantly change.

Generally, based on both Type A and B data, if we set $\rho = 5mm$, the proposed method with the fast setup achieves a computation speed of 10.3 fps and an error of 1.07 mm, an overlap ratio of 78.4% and a failure ratio of 4.1% on all data sets. Fig. 5.9 shows examples of the tracked results on Type A data. The rows correspond to (a) the results using the previous method that

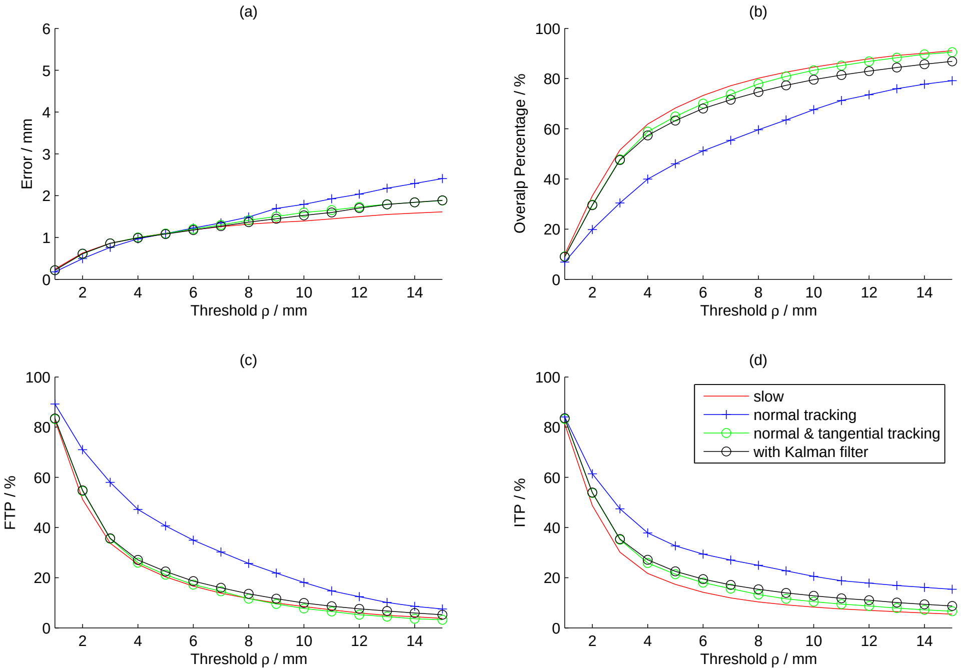
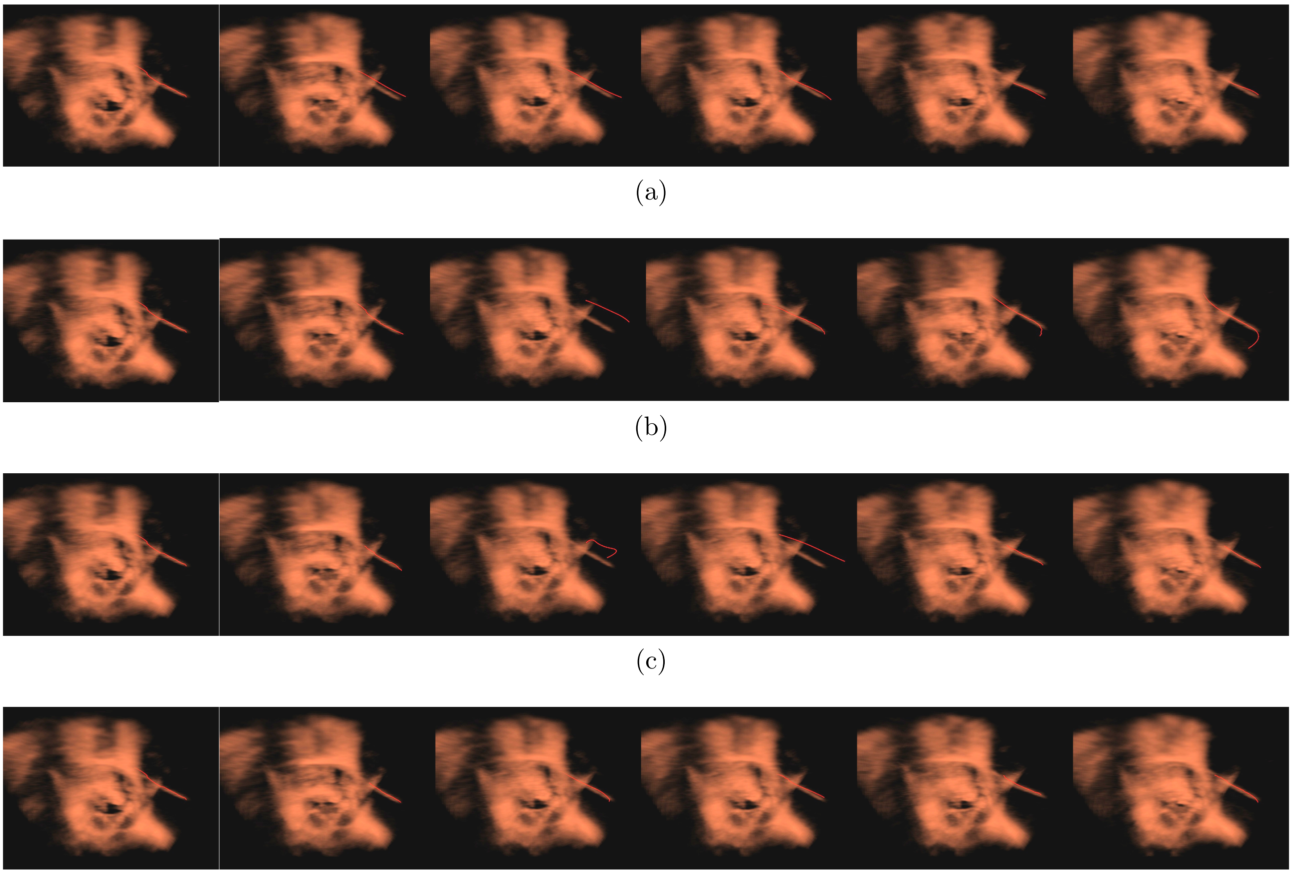


Figure 5.8: Comparison results with both Type A and Type B datasets.

uses X-ray images [214, 215], (b) the method with only transversal tracking, (c) the method with both transversal and longitudinal tracking but without Kalman filter prediction, and (d) the proposed method including all building blocks. The resulting catheter splines are marked in red. For each row, the six columns correspond to frame numbers 1, 32, 66, 74, 85 and 99. Fig. 5.10 shows examples of the tracked results on Type B data. The rows correspond to (a) the results using the method with only transversal tracking, (b) the method with both transversal and longitudinal tracking but without Kalman filter prediction, and (c) the proposed method including all improvements. The resulting catheter splines are marked in red. For each row, the six columns correspond to frame numbers 1, 4, 7, 10, 13 and 16.

5.6 Discussion

In this chapter we proposed a strategy for catheter tracking in ultrasound sequences. To enable its clinical application, our approach has been integrated as a module in a general framework: For example, for the first frame of the sequence and the frames immediately after a tracking



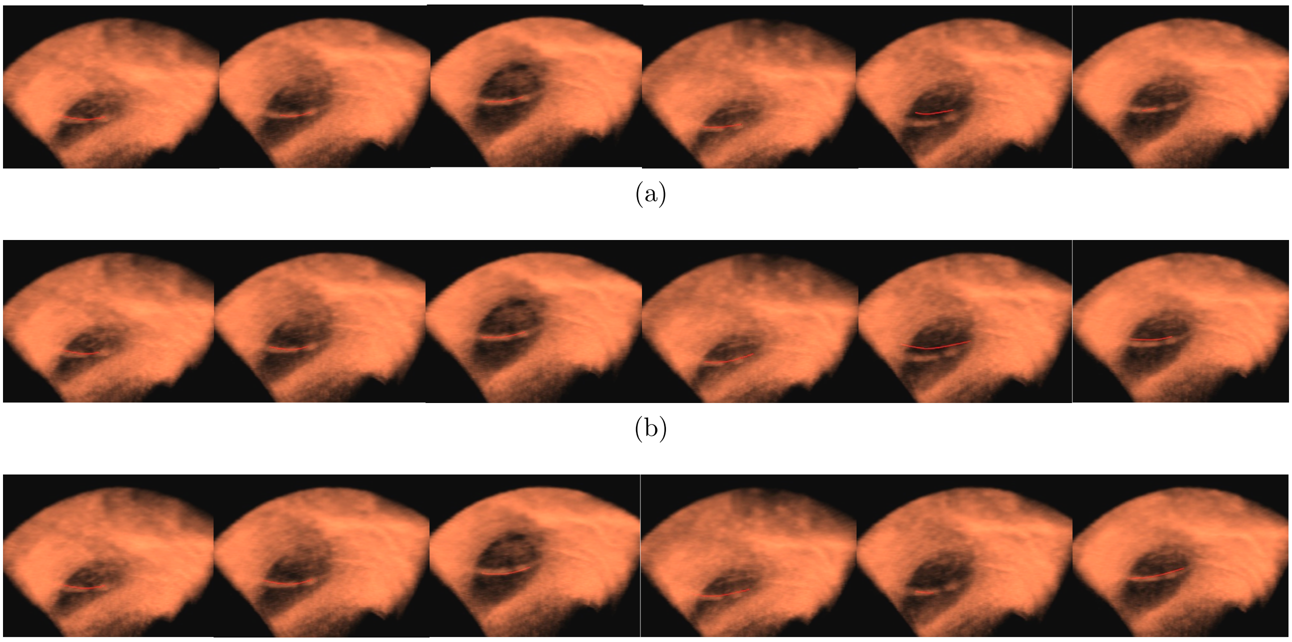
(a)

(b)

(c)

(d)

Figure 5.9: Examples of the tracking result on Type A (A1) sequence.



(a)

(b)

(c)

Figure 5.10: Examples of the tracking result on Type B (B7) sequence.

failure, our algorithm requires that the extraction result is provided by some other methods. Thus, in this framework, an automatic or semi-automatic catheter segmentation algorithm is required to give an initialization for our tracking algorithm. The initialization method proposed in our previous work [214] required X-ray to help segment the catheter in both X-ray and ultrasound images. Since catheter segmentation directly from ultrasound images is challenging due to the severe interference of acoustic artefacts, X-ray imaging could also be a solution to the initialization for the current algorithm. However, for the frames where the catheter can be easily tracked, the algorithm proposed in this paper can be used without X-ray exposure. This reduces the risk of radiation from X-ray compared to the previous approach.

Since tracking re-initializations are currently unavoidable, a method to automatically detect tracking failures is also investigated. We detect failures using the average tubular structure response of all sampled landmarks on the detected curve. The average response can detect if a tracking failure occurs in most cases. However, it neither takes the motion pattern of the catheter nor the tubular response distribution along the entire catheter length into account. In other words, if the motion pattern is more predicable, then the catheter position in the next frame will be more certain as well. Thus, if a catheter is detected far from its predicted position, then this can also be incorporated into the identification of a tracking failure. Furthermore, using the average tubular response ignores the possibility that a acceptable catheter tracking result may have a large incorrectly tracked proportion. However, this incorrectly tracked proportion may make the average tubular response low. As a result, this acceptable result will be considered as a tracking failure if we only use average tubular response to evaluate it.

Regarding the discrete optimization framework, the experimental evaluation shows that our strategy achieves a good robustness to parameters changes. However, one key requirement to guarantee this robustness is to ensure that the catheter spline curve is always smooth after tracking. The reason is that if the catheter is not smooth enough, it may not be consistent with the B-spline model so that the margin length at the end of the detected catheter might not overlap with the real catheter. In such cases, the undetected parts of the catheter could not be recovered during the longitudinal tracking stage because the undetected parts would be too far from the search space. To address this, we currently use the second derivative of the

B-spline model and add a constraint that the second derivatives of all the landmarks on the curve should be zero. This constraint aims to make the margin length's direction and curvature more consistent with the detected catheter length. However, this is still a potential problem if the curvature of the catheter tip is large and an insufficient number of observations are sampled on the catheter.

Compared with the discrete optimization, the Kalman filter is very sensitive to parameter settings. This is not surprising since most of these parameters, including the covariance matrix, are linked with the motion properties for each individual sequence. Thus, in order to apply the Kalman filter in the clinical context, the motion properties (which are strongly related to the properties of the imaging such as frame rate and resolution) should be specified as prior information and then used to set these parameters within the correct range. In our application, the motion model and the measurement model are straightforward and many assumptions have been made to simplify the filter system, including reducing the dimensionality of the parameter space. The states being tracked are mainly the transversal motion rather than longitudinal motion. The large transversal motion is the main factor responsible for tracking failures as it causes the catheter to be outside the search space. Although our simple Kalman filter system is successful, the full potential has not been explored. Thus in future work we will investigate using a more advanced model and more elaborated filtering technique to obtain a higher robustness to parameter changes and other interference from noise, outliers and modelling errors.

It is important to emphasise the real-time aspect of the application. The minimal requirement of the frame rate is at least 10 fps. By constraining the search space, our algorithm can satisfy this with a small trade-off for other performance metrics. However, to deal with more complicated cases, the trade-off may be more expensive. For example, to further accelerate our algorithm, the tubular structure enhancement is performed on the warped volume, which is much smaller than the original volume. A drawback of this is that it may introduce a very nonlinear space when the curvature of the catheter is high and the sampling varies compared to the original volume. In such cases, a curve which is smooth in the warped volume may have an unexpected non-smooth shape in the original volume. In addition, a low-resolution region will have insufficient observations and the curve in this region may lack accuracy. For high curvature catheters, the algorithm may need to use the original volume directly. Another trade-off involves the search range for the tracking in transversal direction, where the computational costs increase drastically with higher search radii. Thus, to achieve real-time performance, the search range can be limited or the search resolution can be reduced. A smaller search space will cause the catheter to be out of range more frequently and thus increase the failure ratio, while a lower search resolution will reduce accuracy and lead to a higher tracking error.

From the perspective of clinical applications, the most important part of the catheter to be tracked is often the tip. However, we have found that directly segmenting and tracking the tip in ultrasound images is extremely challenging. Thus, our approach is to first segment the entire catheter and then to localize the tip on this curve. This makes the tip detection easier since the tip must be located close to the end of the curve. However, due to the complexity of ultrasound image formation, the tip may not correspond exactly to the end point of the detected curve. Our algorithm therefore provides a starting constraint for a tip detection algorithm, which would be an additional step in our detection framework.

Chapter 6

Conclusion

6.1 Summary

In this thesis we have proposed three modules relating catheter segmentation and tracking in both X-ray and ultrasound images: (1) A fast catheter segmentation and tracking module for X-ray images; (2) A fast catheter segmentation algorithm in ultrasound images assisted by the registered X-ray images; and (3) A real-time catheter tracking framework in ultrasound images, without the use of X-ray images.

The first module is a combination of both catheter segmentation and tracking in X-ray sequences. One of our key contributions is the Kalman filter-based catheter extraction that has been used in both catheter segmentation and catheter tracking. The novel characteristic of this strategy is that it can extend the catheter extraction to the entire catheter from key landmarks. The proposed catheter segmentation based on this algorithm can run very quickly with a speed of more than 1000 frames per second (only 0.0006 seconds per frame). It has the low error of 0.82mm but with a high FTP of 44%, a low ITP of 9.2%, and a high failure ratio of 0.26%. During the catheter tracking the previous method, which is based on Fast-PD optimization, can achieve a speed of 31 fps and a error of 1.14mm, but with a higher FTP of 47%, a slightly lower ITP of 12.9%, and a failure ratio of 0.21%. However, with our extraction strategy, our combined method can achieve a lower FTP of 29.2%, an ITP of 18.2%. There is no trade-off in terms of speed, accuracy and the failure ratio. This Kalman filter-based extraction strategy has significantly reduced the failed tracking percentage, only with a small rise in the incorrectly tracking percentage. With regards to the multiple catheter tracking, our proposed system can achieve a CTE/TTE of 1.40/8.10 mm and an overlap percentage of 87.18%.

The second module consists of two strategies based on: (1) the shortest path in a single layer graph; and (2) the optimal path in a two-layer graphical model. The shortest path based strategy, although it achieves a speed of 2.4s per frame and a re-projection error of 1.60mm, only works well on phantom data and in some clinical datasets. Thus the second strategy is recommended for clinical cases. This strategy has been tested on different clinical datasets and achieves more convincing results. The experimental results show that it can track catheter motions in ultrasound images at 1.3 second per frame, with an error of less than 2 mm. Fewer than 4.5% of the tracked results are incorrect and more than 72% of the ground truth is tracked. Fewer than 12% of frames have a tracking failure. The path growing strategy for ultrasound tracking trades-off a small rise in the percentage of incorrectly tracked features (increased ITP), but is able to track a larger proportion of the catheter (reduced FTP). This is similar to the Kalman filter-based extraction strategy in X-ray tracking. This strategy makes the system more adaptive to cases where the catheter moves longitudinally and causes a change in the visible length of the catheter. The 2D SURF detector in the straightened space produces better measurements than the widely used vesselness filtering methods in our applications.

Lastly, we have proposed a real-time catheter tracking strategy based on ultrasound imaging for cardiac catheterization. A two-step optimization framework for both transversal and longitudinal tracking has been developed to ensure that the tracking algorithm is fast and reliable. In addition, a Kalman filter is used to reduce the tracking failures. The experimental results show that the proposed approach can track catheter motions at more than 10 frames per second, with an error of 1 mm approximately. To the best of our knowledge this is also the first study satisfying the real-time requirements (10 frames per second). The overlap ratio between the ground truth and the result is more than 75%. Around 4% of frames need to be re-initialized by X-ray tracking or manual intervention. It also shows that the longitudinal tracking module is very robust to parameter changes. The Kalman filter module is less robust to the corresponding parameter changes. However, clues from prior information can be employed to choose the right range of the parameters of the Kalman filter.

6.2 Applications

The direct application of the proposed modules can include an entire system combining all of the modules. The main purpose of this application is: (1) to introduce the ultrasound modality during the interventions to enrich the visualization of 3D soft tissues; (2) to reduce the X-ray exposure to minimize the harm to human health; and (3) to make the system more automatic and require less manually interventions. However, it is very challenging to discard X-ray modality completely and to fully automate the process. Thus, based on the algorithms proposed in this thesis, a strategy linking each module is proposed to construct an overall framework, as shown in Fig. 6.1.

Here, we separate the module using the X-ray modality into extraction and tracking modules. The extraction module takes manually marked landmarks as input and then extracts the entire catheter in the first frame or in the frame where the tracking in X-ray fails. Then the result can be used as an initialization for the tracking module, which uses the result from the previous frame as the input. After initialization, the tracking module can work in an isolated loop (the orange loop) until a failure has been detected. The results from both the extraction and tracking modules for X-ray images provide a catheter curve in 2D. Then, simultaneously the registration between X-ray and ultrasound can be carried out to obtain the transformation matrix, based on an off-line reconstructed US probe model. In a real application, this can be carried out once and then re-used in all following frames as our system allows some registration errors. Thus the time spent for the registration for every frame can be saved. The registration and the results from X-ray are used to extract the catheter in ultrasound image. The catheter extraction module in the ultrasound images can extract the catheter based on the corresponding X-ray and also provides an initialization to the last module: tracking for ultrasound images. After the tracking module for ultrasound images enabled, the system enters the red loop, where the

X-ray imaging is not needed until a tracking failure in the ultrasound images detected. Thus, if the red loop continues without tracking failure, the other modules are not required to compute, including the orange loop. This means that X-ray imaging is not used within the red loop.

However, when there is a failure detected in the ultrasound tracking, the system enters the orange loop again. The result from the last successfully tracked ultrasound frame can be projected back to the current 2D X-ray image. This projection can be used to initialize the orange loop again for the purpose of catheter tracking in X-ray images. When the orange loop is enabled, the result from the current X-ray image can be used for the catheter extraction in the ultrasound images (module 4) again. Then the output from this can also be used to initialize the red loop. If the red loop works without failure detected, then the orange loop is not required. If the red loop fails again, the tracking will stay in the orange loop. However, in the orange loop, if a tracking failure is detected as well for X-ray images, then the entire system will go to the beginning, manually pointing. During the whole process, manually interventions are only needed to identify the tracking failures in both X-ray and ultrasound images and to specify the catheter being tracked by clicking on some key points. This significantly reduces the manually work. Based on our experimental ultrasound data, the percentage of frames in which there is a tracking failure is around 4% . Theoretically during a four hour intervention the X-ray exposure can be reduced to less than 10 minutes. However, in clinical procedures, it is difficult to estimate the exact amount of X-ray exposure because the need for X-ray imaging does not only depend on the reliability of the tracking system in ultrasound images but also the need to better visualize some important structures via high quality, less artifacted X-ray imaging.

6.3 Future Work

The system we have proposed works well based on our tested datasets. However, though most of the datasets we acquired are from clinical procedures, the limited quantity of datasets tested means that these datasets cannot include every challenging aspect. In order to address these

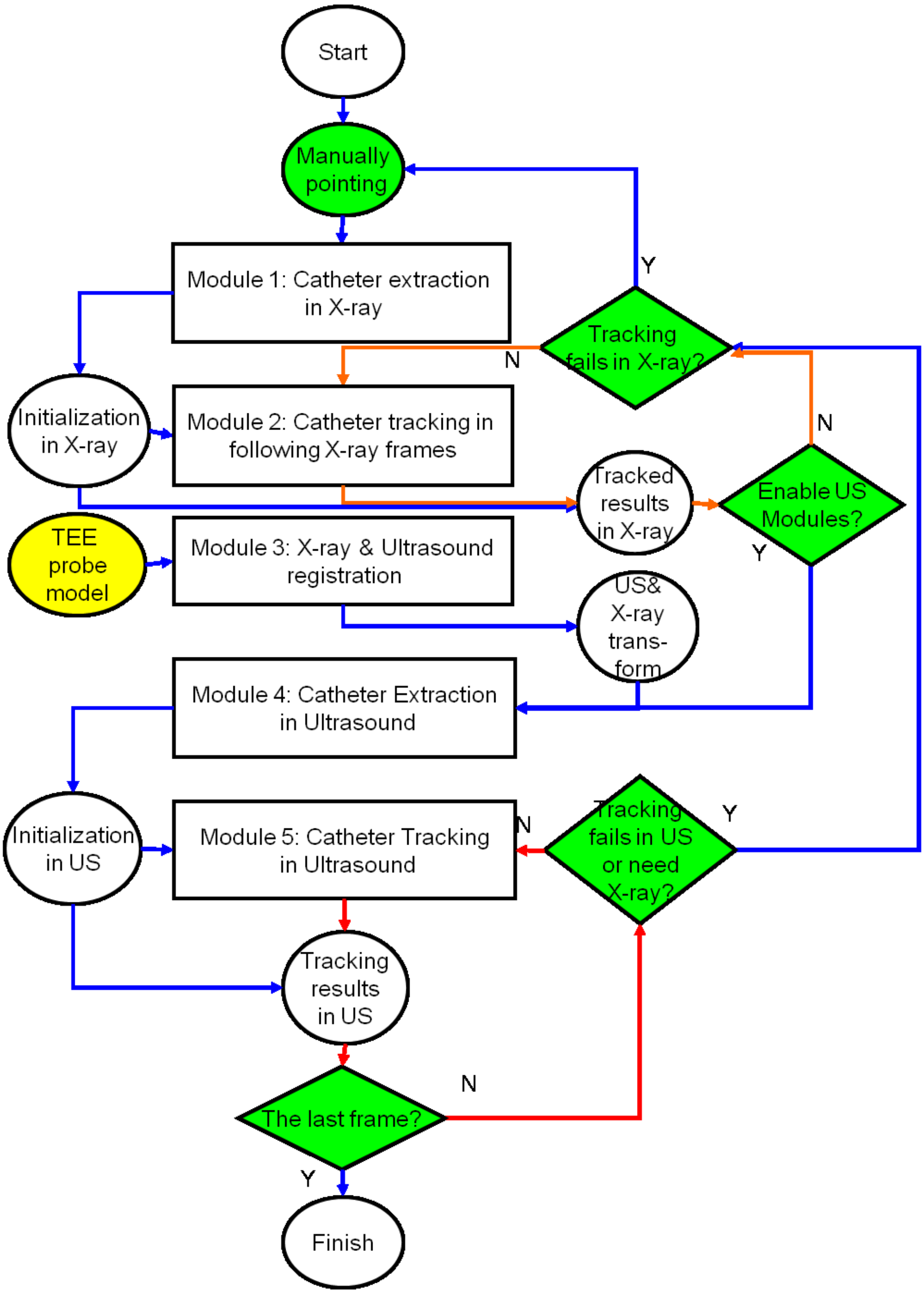

Start
Manually pointing
Module 1: Catheter extraction in X-ray
Y
Tracking fails in X-ray?
N
Initialization in X-ray
Module 2: Catheter tracking in following X-ray frames
N
Tracked results in X-ray
Enable US Modules?
Y
TEE probe model
Module 3: X-ray & Ultrasound registration
US& X-ray trans- form
Module 4: Catheter Extraction in Ultrasound
Initialization in US
Module 5: Catheter Tracking in Ultrasound
N
Tracking fails in US or need X-ray?
Y
Tracking results in US
N
The last frame?
Y
Finish

Figure 6.1: Green indicates manually interventions are needed. Yellow indicates processing that can take place off-line. The red and orange arrows corresponds to two loops which are referred as red loop and orange loop respectively.

potential problems, there are some several improvements that can be made.

The first issue is that in the field of view for either X-ray or ultrasound images more than one catheter is visible. Besides multiple catheters, there may also be other catheter-like structures in the view. An ideal framework needs to identify, segment and track multiple catheters simultaneously, as well as rule out outliers. In Chapter 3 we have proposed a multiple catheter tracking strategy for X-ray imaging. The proposed dynamic graph model can preserve relative distances in order to distinguish each catheter.

However, this strategy focuses on only the pixels of the catheters rather than the entire image. To illustrate this, consider in the first frame two catheters overlap but in the second frame they no longer overlap and separate. Now we are evaluating one incorrect result in the second frame: The estimated locations of the two catheters still overlap on one of the real locations of these two catheters, with the other real location undetected. Then, if we only evaluate the catheters which have being tracked, we would find the appearance of both tracked catheters is acceptable. This is because both these two catheters are located on one of the real catheter locations. Even one of them is incorrectly located. In this case, this incorrect result would be skipped without detection. However, if we focus on every pixel of the image, i.e., we measure the appearance of every pixel and calculate the spatial distance to the estimated locations, we will find that pixels on the other real location (which the estimated locations are far from) with a high similarity values but a large distance to the estimated locations. Based on this clue we can easily identify this incorrect result. However, evaluating every pixel in the entire image is expensive since the number of pixels in the whole image is larger than the number of pixels in the vicinity of the catheters. A suitable sampling strategy can be used to address it.

As for the ultrasound images, multiple catheter tracking is more challenging. To provide clues to identify each catheter, catheters extracted from X-ray images are used. However, this is insufficient since a catheter curve in X-ray image corresponds to a curved surface in the ultrasound volume. There could be more than one catheter candidate around this curved surface. Thus other constraints should be introduced as well. One possible clue is the temporal consistency between adjacent frames. Thus, the extracted result from the current frame should be close to the one extracted from the last frame. This has been considered in the catheter ultrasound tracking module when X-ray imaging is not needed. However, for the catheter extraction module combining both X-ray and ultrasound images, only constraints from the X-ray images are taken into account. Temporal consistency should be included as well in our future work.

Another issue regarding ultrasound imaging is the localization of catheter tips, which are more important in clinical practice. Currently the tip detection is linked to the curve extraction. In other words, the ends of the catheter curve are considered as the potential tip locations. This is not always true in the clinical practice since some parts of the catheter are not clearly visible due to the sound effects. Thus the ends of a curve may correspond to the point dividing visible and invisible parts of the catheter. To identify the correct catheter tip, the X-ray images should be used as a clue since the tip on X-ray images is better visualized. If X-ray images are not in use, which is the case in catheter ultrasound tracking module, other clues should be explored. For example, in ultrasound images, the tip of catheters often appears with a comet-tail artifact, which can be easily identified.

Catheters with high curvature shapes, e.g., the round tip of a lasso catheter, are also one limitation of our algorithm. For catheters with high curvature, a dense sampling of control points and dense observations can address this but it would incur a high computational cost. Thus this is not possible in the current framework. However, due to the rapid development of GPU hardware, this may not be a problem in the long term.

After the general framework has been established, several details of the framework still have potential for improvement. One specific issue is the measurements of the catheter in both X-ray and ultrasound images. The measurements which are used currently mainly depend on blob and tubular detectors. However, these detectors for either X-ray or ultrasound images have a large number of false positives and false negatives. To make an improvement, learning-based approaches can be used to train the detector off-line and then use be the trained detector online. However, in order to adjust the detector to the changes of the appearance online, fast incremental learning methods must be employed to update it. The field of machine learning provides a solid theoretical background to use such techniques. However, such learning-based algorithms require a large amount of training data.

Another issue that can be further improved is the modelling of the motion pattern of the catheter, which is consistent with the cardiac periodic motions. This model can provide more accurate prediction of the catheter location for the next frame. In our current framework, Kalman filtering is used to predict catheter location. However, this can be further improved by modelling the motion pattern off-line and then use of the learnt motion pattern in the online procedure. Machine learning techniques can be used to either train the motion pattern directly or to extract a good feature to describe the motion pattern. This feature can be tracked in addition to the location of the catheter to provide clues for motion estimation at each frame.

Bibliography

[1] IEEE standards interpretations: IEEE Std 1076-1987, IEEE standard VHDL language reference manual. *IEEE Std 1076/INT-1991*, pages 1–, 1992.

[2] Rolf Adams and Leanne Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[3] Nitin Aggarwal and W Clem Karl. Line detection in images through regularized hough transform. *IEEE Transactions on Image Processing*, 15(3):582–591, 2006.

[4] Michal Aharon, Michael Elad, and Alfred Bruckstein. -svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[5] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[6] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[7] Shuichi Asano, Tsutomu Maruyama, and Yoshiki Yamaguchi. Performance comparison of FPGA, GPU and CPU in image processing. In *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, pages 126–131. IEEE, 2009.

[8] Safuh Attar, John R Hankins, Stephen Z Turney, Mark J Krasna, and Joseph S

McLaughlin. Paraplegia after thoracotomy: report of five cases and review of the literature. *The Annals of Thoracic Surgery*, 59(6):1410–1416, 1995.

[9] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2011.

[10] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[11] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[12] Lauren Barghout and Jacob Sheynin. Real-world scene perception and perceptual organization: Lessons from computer vision. *Journal of Vision*, 13(9):709–709, 2013.

[13] Kees Joost Batenburg and Jan Sijbers. Adaptive thresholding of tomograms by projection distance minimization. *Pattern Recognition*, 42(10):2297–2305, 2009.

[14] Kees Joost Batenburg and Jan Sijbers. Optimal threshold selection for tomogram segmentation by projection distance minimization. *IEEE Transactions on Medical Imaging*, 28(5):676–686, 2009.

[15] Gilbert Baum. *Fundamentals of medical ultrasonography*. Putnam, 1975.

[16] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[17] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *The Neural Information Processing Systems*, volume 14, pages 585–591, 2001.

[18] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

[19] Yoshua Bengio and Samy Bengio. Modeling high-dimensional discrete data with multi-layer neural networks. In *The Neural Information Processing Systems*, pages 400–406, 1999.

[20] Goetz Benndorf, Benjamin Claus, Charles M Strother, Lee Chang, and Richard P Klucznik. Increased cell opening and prolapse of struts of a neuroform stent in curved vasculature: value of angiographic computed tomography: technical case report. *Neurosurgery*, 58(4):ONS–E380, 2006.

[21] Hiram G Bezerra, Marco A Costa, Giulio Guagliumi, Andrew M Rollins, and Daniel I Simon. Intracoronary optical coherence tomography: a comprehensive review: clinical and research applications. *JACC: Cardiovascular Interventions*, 2(11):1035–1046, 2009.

[22] Deepak Bhakta and John M Miller. Principles of electroanatomic mapping. *Indian Pacing and Electrophysiology Journal*, 8(1):32, 2008.

[23] Biosensewebster. Lasso NAV Eco Catheter, 2014. `https://www.biosensewebster.com/lassonav.php`.

[24] J-L Blanco, Javier González, and J-A Fernández-Madrigal. An optimal filtering algorithm for non-parametric observation models in robot localization. In *IEEE International Conference on Robotics and Automation*, pages 461–466. IEEE, 2008.

[25] Jose Blanco, Javier González, and Juan-Antonio Fernández-Madrigal. Optimal filtering for non-parametric observation models: applications to localization and SLAM. *The International Journal of Robotics Research*, 29(14):1726–1742, 2010.

[26] Y-lan Boureau, Yann L Cun, et al. Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, pages 1185–1192, 2008.

[27] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient ND image segmentation. *International journal of computer vision*, 70(2):109–131, 2006.

[28] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of

min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[29] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[30] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In *IEEE International Conference on Computer Vision*, volume 1, pages 105–112. IEEE, 2001.

[31] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[32] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[33] AR Brodbelt, JB Miles, PM Foy, and JC Broome. Intraspinal oxidised cellulose (surgicel) causing delayed paraplegia after thoracotomy–a report of three cases. *Annals of the Royal College of Surgeons of England*, 84(2):97, 2002.

[34] Alexander Brost, Rui Liao, Joachim Hornegger, and Norbert Strobel. 3-D respiratory motion compensation during EP procedures by image-based 3-D lasso catheter model generation and tracking. In *Medical Image Computing and Computer-Assisted Intervention*, pages 394–401. Springer, 2009.

[35] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36. Springer, 2004.

[36] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.

[37] Cristian Canton-Ferrer, Josep R Casas, and Montse Pardàs. Human motion capture using scalable body models. *Computer Vision and Image Understanding*, 115(10):1363–1374, 2011.

[38] Kunlin Cao, David Mills, and Kedar A Patwardhan. Automated catheter detection in volumetric ultrasound. In *International Symposium on Biomedical Imaging*, pages 37–40. IEEE, 2013.

[39] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014.

[40] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[41] Shuai Che, Jie Li, Jeremy W Sheaffer, Kevin Skadron, and John Lach. Accelerating compute-intensive applications with GPUs and FPGAs. In *Application Specific Processors, 2008. SASP 2008. Symposium on*, pages 101–107. IEEE, 2008.

[42] Li Chen, Heng-Da Cheng, and Jianping Zhang. Fuzzy subfiber and its application to seismic lithology classification. *Information Sciences-Applications*, 1(2):77–95, 1994.

[43] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[44] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[45] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 142–149. IEEE, 2000.

[46] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[47] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[48] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[49] Ben Cope, Peter YK Cheung, Wayne Luk, and Lee Howes. Performance comparison of graphics processors to reconfigurable logic: a case study. *Computers, IEEE Transactions on*, 59(4):433–448, 2010.

[50] Ben Cope, Peter YK Cheung, Wayne Luk, and Sarah Witt. Have GPUs made FPGAs redundant in the field of video processing? In *Field-Programmable Technology, 2005. Proceedings. 2005 IEEE International Conference on*, pages 111–118. IEEE, 2005.

[51] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, January 1967.

[52] Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2–3):81–227, 2012.

[53] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893 vol. 1, June 2005.

[54] Reference designer. Verilog tutorial for beginners, 2009. `http://www.referencedesigner.com/tutorials/verilog/verilog_01.php`.

[55] Bin Dong, Eric Savitsky, and Stanley Osher. A novel method for enhanced needle localization using ultrasound-guidance. In *Advances in Visual Computing*, pages 914–923. 2009.

[56] Arnaud Doucet, Nando De Freitas, Kevin Murphy, and Stuart Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[57] Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, January 1972.

[58] George H Dunteman. *Principal components analysis.* Number 69. Sage, 1989.

[59] EMedicineHealth. Supraventricular tachycardia, 2014. `http://www.emedicinehealth.com/supraventricular_tachycardia/article_em.htm`.

[60] Klaus Engel, Markus Hadwiger, Joe M Kniss, Christof Rezk-Salama, and Daniel Weiskopf. *Real-time volume graphics.* AK Peters, Limited, 2006.

[61] Franklin H Epstein and Russell Ross. Atherosclerosisan inflammatory disease. *New England journal of medicine*, 340(2):115–126, 1999.

[62] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[63] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[64] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[65] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54, 2006.

[66] Leandro AF Fernandes and Manuel M Oliveira. Real-time line detection through an improved hough transform voting scheme. *Pattern Recognition*, 41(1):299–314, 2008.

[67] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[68] David A Forsyth and Jean Ponce. *Computer vision: a modern approach.* Prentice Hall Professional Technical Reference, 2002.

[69] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multiscale vessel enhancement filtering. In *Medical Image Computing and Computer-Assisted Interventation*, pages 130–137. Springer, 1998.

[70] William T Freeman, Egon C Pasztor, and Owen T Carmichael. Learning low-level vision. *International journal of computer vision*, 40(1):25–47, 2000.

[71] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In Paul Vitnyi, editor, *Computational Learning Theory*, volume 904 of *Lecture Notes in Computer Science*, pages 23–37. Springer Berlin Heidelberg, 1995.

[72] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2188–2202, 2011.

[73] Gang Gao, Graeme Penney, Nicolas Gogin, Pascal Cathier, Aruna Arujuna, Matt Wright, Dennis Caulfield, Aldo Rinaldi, Reza Razavi, and Kawal Rhode. Rapid image registration of three-dimensional transesophageal echocardiography and X-ray fluoroscopy for the guidance of cardiac interventions. In *Information Processing in Computer-Assisted Interventions*, pages 124–134. Springer, 2010.

[74] Gang Gao, Graeme Penney, Yingliang Ma, Nicolas Gogin, Pascal Cathier, Aruna Arujuna, Geraint Morton, Dennis Caulfield, Jaswinder Gill, C Aldo Rinaldi, et al. Registration of 3D trans-esophageal echocardiography to X-ray fluoroscopy using image-based probe tracking. *Medical Image Analysis*, 16(1):38–49, 2012.

[75] Martin Godec, Peter M Roth, and Horst Bischof. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding*, 117(10):1245–1256, 2013.

[76] Helmut Grabner and Horst Bischof. On-line boosting and vision. In *IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 260–267. IEEE, 2006.

[77] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *BMVC*, volume 1, page 6, 2006.

[78] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *European Conference on Computer Vision*, pages 234–247. Springer, 2008.

[79] Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

[80] Leo Grady and Eric L Schwartz. Isoperimetric graph partitioning for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):469–475, 2006.

[81] William Grossman and Donald S Baim. *Grossman's cardiac catheterization, angiography, and intervention*, volume 1. Wolters Kluwer Health, 2006.

[82] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[83] Peter E Hart. How the hough transform was invented [dsp history]. *Signal Processing Magazine, IEEE*, 26(6):18–22, November 2009.

[84] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[85] Hauke Heibel, Ben Glocker, Martin Groher, Marcus Pfister, and Nassir Navab. Interventional tool tracking using discrete optimization. *IEEE Transactions on Medical Imaging*, 32(3):544–555, 2013.

[86] Tim Hauke Heibel, Ben Glocker, Martin Groher, Nikos Paragios, Nikos Komodakis, and Nassir Navab. Discrete tracking of parametrized curves. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1754–1761. IEEE, 2009.

[87] Tobias Heimann, Peter Mountney, Matthias John, and Razvan Ionasec. Learning without labeling: Domain adaptation for ultrasound transducer localization. In *Medical Image Computing and Computer-Assisted Intervention*, pages 49–56. Springer, 2013.

[88] Juerg Hodler, Gustav K von Schulthess, and Christoph L Zollikofer. *Diseases of the Heart, Chest & Breast: Diagnostic Imaging and Interventional Techniques.* Springer, 2007.

[89] Nicolas Honnorat, Régis Vaillant, and Nikos Paragios. Graph-based geometric-iconic guide-wire tracking. In *Medical Image Computing and Computer-Assisted Intervention*, pages 9–16. Springer, 2011.

[90] Berthold K Horn and Brian G Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981.

[91] Steven L Horowitz and Theodosios Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the ACM (JACM)*, 23(2):368–388, 1976.

[92] R James Housden, Aruna Arujuna, Y Ma, N Nijhof, Geert Gijsbers, Roland Bullens, Mark O'Neill, Michael Cooklin, C Aldo Rinaldi, J Gill, et al. Evaluation of a real-time hybrid three-dimensional echo and X-ray imaging system for guidance of cardiac catheterisation procedures. In *Medical Image Computing and Computer-Assisted Intervention*, pages 25–32. Springer, 2012.

[93] Jeffrey D Houston and Michael Davis. *Fundamentals of Fluoroscopy.* W.B. Saunders Company, 2001.

[94] The Heart Institute. Top cardiothoracic surgeons from all over the world learn the mcginn technique at the heart institue at staten island, 2011. `http://www.theheartinstituteny.com/pdf/THI_March_24th_Press_Release.pdf`.

[95] Michael Isard and Andrew Blake. Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[96] Marine JE. Catheter ablation therapy for supraventricular arrhythmias. *JAMA*, 298(23):2768–2778, 2007.

[97] Mark E Josephson. *Clinical Cardiac Electrophysiology: Techniques and Interpretations.* Wolters Kluwer Health/Lippincott Williams & Wilkins, 2008.

[98] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

[99] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[100] Roland Kehl, Matthieu Bray, and Luc Van Gool. Full body tracking from multiple views using stochastic sampling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 129–136 vol. 2, June 2005.

[101] Khronos. The open standard for parallel programming of heterogeneous systems, 2015. `https://www.khronos.org/opencl/`.

[102] Cemil Kirbas and Francis Quek. A review of vessel extraction techniques and algorithms. *ACM Comput. Surv.*, 36(2):81–121, June 2004.

[103] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[104] Vladimir Kolmogorov and Carsten Rother. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1274–1279, 2007.

[105] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[106] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. Mrf optimization via dual decomposition: Message-passing revisited. In *IEEE International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[107] Nikos Komodakis, Georgios Tziritas, and Nikos Paragios. Fast, approximately optimal solutions for single and dynamic MRFs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[108] Peter R Kowey, James Burke, and Adam Mohmand-Borkowski. *Clinical Management of Atrial Fibrillation*. Professional Communicatio, 2011.

[109] Karl Krissian, Gregoire Malandain, Nicholas Ayache, Regis Vaillant, and Yves Trousset. Model-based detection of tubular structures in 3D images. *Computer Vision and Image Understanding*, 80(2):130 – 171, 2000.

[110] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[111] Amlan Kundu, Yang He, and Paramvir Bahl. Recognition of handwritten word: first and second order hidden markov model based approach. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 457–462. IEEE, 1988.

[112] Ivan Laptev, Helmut. Mayer, Tony Lindeberg, Wolfgang Eckstein, Carsten Steger, and Albert Baumgartner. Automatic extraction of roads from aerial images based on scale space and snakes. *Machine Vision and Applications*, 12(1):23–31, 2000.

[113] Max WK Law and Albert CS Chung. Three dimensional curvilinear structure detection using optimally oriented flux. In *European Conference on Computer Vision*, pages 368–382. Springer, 2008.

[114] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.

[115] Tony Lindeberg. Scale-space theory: a basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 21(1-2):225–270, 1994.

[116] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[117] Tony Lindeberg. Scale Invariant Feature Transform. *Scholarpedia*, 7(5):10491, 2012.

[118] Tony Lindeberg and Jonas Gårding. Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure. In *European Conference on Computer Vision*, pages 389–400. Springer, 1994.

[119] Tony Lindeberg and Meng-Xiang Li. Segmentation and classification of edges using minimum description length approximation and complementary junction cues. *Computer Vision and Image Understanding*, 67(1):88–98, 1997.

[120] Jie Liu, Wilson Wang, and Fai Ma. A regularized auxiliary particle filtering approach for system state estimation and battery life prediction. *Smart Materials and Structures*, 20(7):075021, 2011.

[121] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[122] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

[123] Daniel Lustig and Margaret Martonosi. Reducing GPU offload latency via fine-grained CPU-GPU synchronization. In *HPCA*, volume 13, pages 354–365, 2013.

[124] YingLiang Ma, Nicolas Gogin, Pascal Cathier, R James Housden, Geert Gijsbers, Michael Cooklin, Mark O'Neill, Jaswinder Gill, C Aldo Rinaldi, Reza Razavi, et al. Real-time X-ray fluoroscopy-based catheter detection and tracking for cardiac electrophysiology interventions. *Medical physics*, 40(7):071902, 2013.

[125] YingLiang Ma, Andy P King, Nicolas Gogin, C Aldo Rinaldi, Jaswinder Gill, Reza Razavi, and Kawal S Rhode. Real-time respiratory motion correction for cardiac

electrophysiology procedures using image-based coronary sinus catheter tracking. In *Medical Image Computing and Computer-Assisted Intervention*, pages 391–399. Springer, 2010.

[126] Michael J Mack. Minimally invasive cardiac surgery. *Surgical Endoscopy And Other Interventional Techniques*, 20(2):S488–S492, 2006.

[127] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009.

[128] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.

[129] Ivan Markovsky. Structured low-rank approximation and its applications. *Automatica*, 44(4):891–909, 2008.

[130] Jiri Matas, Charles Galambos, and Josef Kittler. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 78(1):119–137, 2000.

[131] Joseph T McGinn, Saif Usman, Harry Lapierre, Vijayasimha R Pothula, Thierry G Mesana, and Marc Ruel. Minimally invasive coronary artery bypass grafting dual-center experience in 450 consecutive patients. *Circulation*, 120(11 suppl 1):S78–S84, 2009.

[132] Meddic.jp. Balloon atrial septostomy, 2012. `http://meddic.jp/balloon_atrial_septostomy`.

[133] Cook Medical. Biliary balloon-expandable stent, 2014. `https://www.cookmedical.com/product/-/catalog/formula-418-biliary-balloon-expandable-stent?ds=di_forb4_webds`.

[134] Cook Medical. Slip cath beacon tip hydrophilic catheter, 2014. `https://www.cookmedical.com/product/-/catalog/`

slip-cath-beacon-tip-hydrophilic-catheter?ds=
cbd56282-bfb9-43fc-93dc-acc1adec8dd3.

[135] Medicinenet. Coronary balloon angioplasty and stents (percutaneous coronary intervention, pci), 2014.
http://www.medicinenet.com/coronary_angioplasty/article.htm.

[136] Medmovie. echocardiogram, 2014.
http://medmovie.com/cvml/library_id/7556/topic/cvml_0096i/.

[137] Medtronic. misc cabg technique overview, 2009.
http://www.medtronic.com/mics/documents/MICS_CABG_Technique_Overview.pdf.

[138] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *European Conference on Computer Vision*, volume 2350 of *Lecture Notes in Computer Science*, pages 128–142. Springer Berlin Heidelberg, 2002.

[139] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[140] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, Oct 2005.

[141] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[142] Fausto Milletari, Nassir Navab, and Pascal Fallavollita. Automatic detection of multiple and overlapping ep catheters in fluoroscopic sequences. In *Medical Image Computing and Computer-Assisted Intervention*, pages 371–379. Springer, 2013.

[143] Hossein Mobahi, Shankar R Rao, Allen Y Yang, Shankar S Sastry, and Yi Ma.

Segmentation of natural images by texture and boundary compression. *International journal of computer vision*, 95(1):86–98, 2011.

[144] Peter Mountney, Razvan Ionasec, Markus Kaizer, Sina Mamaghani, Wen Wu, Terrence Chen, Matthias John, Jan Boese, and Dorin Comaniciu. Ultrasound and fluoroscopic images fusion by autonomous ultrasound probe detection. In *Medical Image Computing and Computer-Assisted Intervention*, pages 544–551. Springer, 2012.

[145] Jay Mung, Francois Vignon, and Ameet Jain. A non-disruptive technology for robust 3D tool tracking for ultrasound-guided interventions. In *Medical Image Computing and Computer-Assisted Intervention*, pages 153–160. 2011.

[146] Shelly T Nevo, Marijn Van Stralen, Albert M Vossepoel, Johan HC Reiber, Nico de Jong, Antonius FW van der Steen, and Johan G Bosch. Automated tracking of the mitral valve annulus motion in apical echocardiographic images using multidimensional dynamic programming. *Ultrasound in medicine & biology*, 33(9):1389–1399, 2007.

[147] Paul M Novotny, Jeff A Stoll, Nikolay V Vasilyev, J Pedro, Pierre E Dupont, Todd E Zickler, and Robert D Howe. GPU based real-time instrument tracking with three dimensional ultrasound. In *Medical Image Computing and Computer-Assisted Intervention*, pages 58–65. 2006.

[148] NuMED. Atrioseptostomy catheter, 2011. `http://www.numedforchildren.com/z5-fda.htm`.

[149] Nvidia. CUDA C programming guide. `http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#axzz3VT5PsGvE`.

[150] Nvidia. CUDA parallel computing, 2015. `http://www.nvidia.co.uk/object/cuda-parallel-computing-uk.html`.

[151] Nvidia. Graphics processing unit (GPU), 2015. `http://www.nvidia.com/object/gpu.html`.

[152] Nvidia. Nvidia global, 2015. `http://www.nvidia.com/content/global/global.php`.

[153] Nvidia. What is GPU accelerated computing, 2015.
`http://www.nvidia.co.uk/object/gpu-computing-uk.html`.

[154] Encyclopedia of surgery. Thoracotomy, 2014.
`http://www.surgeryencyclopedia.com/St-Wr/Thoracotomy.html`.

[155] Ron Ohlander, Keith Price, and D Raj Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8(3):313–333, 1978.

[156] Robert C Orth, Michael J Wallace, and Michael D Kuo. C-arm cone-beam ct: general principles and technical considerations for use in interventional radiology. *Journal of Vascular and Interventional Radiology*, 19(6):814–820, 2008.

[157] Nikunj C Oza. Online bagging and boosting. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2340–2345 Vol. 3, Oct 2005.

[158] Toufiq Parag, Fatih Porikli, and Ahmed Elgammal. Boosting adaptive linear weak classifiers for online learning and tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[159] Patient.co.uk. Minimally invasive surgery, 2013.
`http://www.patient.co.uk/doctor/minimally-invasive-surgery#ref-4`.

[160] Patient.co.uk. Atrial fibrillation, 2014.
`http://www.patient.co.uk/doctor/atrial-fibrillation-pro`.

[161] Kim Butts Pauly. Interventional magnetic resonance imaging. *Topics in Magnetic Resonance Imaging*, 16(5):337, 2005.

[162] Olivier Pauly, Hauke Heibel, and Nassir Navab. A machine learning approach for deformable guide-wire tracking in fluoroscopic sequences. In *Medical Image Computing and Computer-Assisted Intervention*, pages 343–350. Springer, 2010.

[163] Philips. Allura xper fd biplane systems, 2014.
`http://www.healthcare.philips.com/gb_en/products/interventional_xray/product/electrophysiology/imaging_systems/electrophysiology_biplanes.wpd`.

[164] Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[165] Manoj Pradhan. Minimally invasive cardiac surgery(MICS). `http://drmanojpradhan.com/minimally-invasive-cardiac-surgerymics`.

[166] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[167] Shankar R Rao, Hossein Mobahi, Allen Y Yang, S Shankar Sastry, and Yi Ma. Natural image segmentation with adaptive texture and boundary encoding. In *Computer Vision–ACCV 2009*, pages 135–146. Springer, 2010.

[168] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, pages 430–443. Springer, 2006.

[169] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[170] Amir Saffari, Christian Leistner, Jakob Santner, Martin Godec, and Horst Bischof. On-line random forests. In *IEEE International Conference on Computer Vision Workshops*, pages 1393–1400, Sept 2009.

[171] Ruslan Salakhutdinov and Geoffrey E Hinton. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pages 448–455, 2009.

[172] Amaia Salvador. Understanding grabcut, an interactive foreground extraction using iterated graph cuts, 2014. `http://bitsearch.blogspot.co.uk/2014/03/understanding-grabcut-interactive.html`.

[173] Sergio Sartori and Paola Tombesi. Emerging roles for transthoracic ultrasonography in pleuropulmonary pathology. *World journal of radiology*, 2(2):83, 2010.

[174] Yoshinobu Sato, Shin Nakajima, Hideki Atsumi, Thomas Koller, Guido Gerig, Shigeyuki Yoshida, and Ron Kikinis. 3D multi-scale line filter for segmentation and

visualization of curvilinear structures in medical images. In *CVRMed-MRCAS'97*, pages 213–222. Springer, 1997.

[175] William C Scarfe, Allan G Farman, and Predag Sukovic. Clinical applications of cone-beam computed tomography in dental practice. *Journal-Canadian Dental Association*, 72(1):75, 2006.

[176] Robert E Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[177] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[178] Matthias Scholz, Fatma Kaplan, Charles L Guy, Joachim Kopka, and Joachim Selbig. Non-linear pca: a missing data approach. *Bioinformatics*, 21(20):3887–3895, 2005.

[179] Robert Sedgewick and Kevin Wayne. *Algorithms (4th ed.).* Addison-Wesley Professional, 2011.

[180] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[181] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600. IEEE, 1994.

[182] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124, Jan 2013.

[183] Robin Sibson. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[184] Siemens. syngo dynact cardiac, 2014. `https://www.healthcare.siemens.co.uk/angio/options-and-upgrades/clinical-software-applications/syngo-dynact-cardiac`.

[185] Stephen M Smith and J Michael Brady. Susana new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[186] Spine-health. Minimally invasive surgery definition, 2014. `http://www.spine-health.com/glossary/minimally-invasive-surgery`.

[187] Carsten Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):113–125, Feb 1998.

[188] Dan Steinberg and Phillip Colla. Cart: classification and regression trees. *The Top Ten Algorithms in Data Mining*, 9:179, 2009.

[189] Ingo Steinwart and Andreas Christmann. *Support vector machines.* Springer, 2008.

[190] Jeffrey Stoll and Pierre Dupont. Passive markers for ultrasound tracking of surgical instruments. In *Medical Image Computing and Computer-Assisted Intervention*, pages 41–48. 2005.

[191] USC Cardiothoracic Surgery. Coronary angiography / angiogram, 2010. `http://www.indiahospitaltour.com/heart/coronary-angiography-angiogram-india.html`.

[192] USC Cardiothoracic Surgery. Minimally invasive cardiac surgery, 2014. `http://www.cts.usc.edu/minimallyinvasivecardiacsurgery.html`.

[193] Jasjit S Suri, Chirinjeev Kathuria, and Filippo Molinari. *Atherosclerosis Disease Management.* Springer, 2010.

[194] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, 2008.

[195] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[196] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[197] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8):693–716, 2004.

[198] Zhuowen Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *IEEE International Conference on Computer Vision*, volume 2, pages 1589–1596. IEEE, 2005.

[199] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[200] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–511. IEEE, 2001.

[201] Volcano. Ivus imaging: Overview, 2013. `http://eu.volcanocorp.com/products/ivus-imaging/#.U7AQSpRdWh0`.

[202] Martin J Wainwright, Tommi S Jaakkola, and Alan S Willsky. Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005.

[203] Eric A Wan and Rudolph Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.

[204] Chang Wang and Sridhar Mahadevan. Manifold alignment using procrustes analysis. In *Proceedings of the 25th international conference on Machine learning*, pages 1120–1127. ACM, 2008.

[205] Peng Wang, Terrence Chen, Ying Zhu, Wei Zhang, Shaohua Kevin Zhou, and Dorin Comaniciu. Robust guidewire tracking in fluoroscopy. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 691–698. IEEE, 2009.

[206] WedMD. Atrial flutter, 2014. http://www.webmd.com/heart-disease/atrial-fibrillation/atrial-flutter.

[207] Greg Welch and Gary Bishop. An introduction to the Kalman filter, 1995.

[208] What when how. The future of cardiac mapping (electrophysiology study of the heart: Mapping procedure) (cardiac arrhythmias), 2014. http://what-when-how.com/cardiac-arrhythmias-new-considerations/the-future-of-cardiac-mapping-electrophysiology-study-of-the-heart-mapping-procedu

[209] Wikipedia. Transthoracic echocardiogram, 2013. http://en.wikipedia.org/wiki/Transthoracic_echocardiogram.

[210] Wikipediac. Cath lab, 2014. http://en.wikipedia.org/wiki/Cath_lab.

[211] Wikipediac. Cone beam computed tomography, 2014. http://en.wikipedia.org/wiki/Cone_beam_computed_tomography.

[212] Wikipediac. Percutaneous coronary intervention, 2014. http://en.wikipedia.org/wiki/Percutaneous_coronary_intervention.

[213] Wen Wu, Terrence Chen, Adrian Barbu, Peng Wang, Norbert Strobel, Shaohua Kevin Zhou, and Dorin Comaniciu. Learning-based hypothesis fusion for robust catheter tracking in 2D X-ray fluoroscopy. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1097–1104. IEEE, 2011.

[214] Xianliang Wu, R James Housden, YingLiang Ma, Daniel Rueckert, and Kawal S Rhode. Real-time catheter extraction from 2D X-ray fluoroscopic and 3D echocardiographic images for cardiac interventions. In *Statistical Atlases and Computational Models of the Heart*, pages 198–206, 2012.

[215] Xianliang Wu, R James Housden, Niharika Varma, YingLiang Ma, Daniel Rueckert, and Kawal S Rhode. Catheter tracking in 3D echocardiographic sequences based on tracking in 2D X-ray sequences for cardiac catheterization interventions. In *International Symposium on Biomedical Imaging*, pages 25–28, 2013.

[216] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.

[217] Xilinx. What is a FPGA, 2015. `http://www.xilinx.com/fpga/index.htm`.

[218] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 532–539, June 2013.

[219] Lin Yang, Bogdan Georgescu, Yefeng Zheng, Yang Wang, Peter Meer, and Dorin Comaniciu. Prediction based collaborative trackers (PCT): A robust and accurate approach toward 3D medical object tracking. *IEEE Transactions on Medical Imaging*, 30(11):1921–1932, 2011.

[220] Yuhong Yang. Information theory, inference, and learning algorithms. *Journal of the American Statistical Association*, 100(472):1461–1462, 2005.

[221] B Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

[222] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*, pages 214–223. Springer, 2007.

[223] Charles T Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 100(1):68–86, 1971.

[224] Xiaoqin Zhang, Weiming Hu, Xiangyang Wang, Yu Kong, Nianhua Xie, Hanzi Wang, Haibin Ling, and Steve Maybank. A swarm intelligence based searching strategy for articulated 3D human body tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 45–50, June 2010.

[225] Yongyue Zhang, Michael Brady, and Stephen Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57, 2001.

[226] Yue Zhao, Hervé Liebgott, and Christian Cachard. Tracking micro tool in a dynamic 3D ultrasound situation using Kalman filter and RANSAC algorithm. In *ISBI*, pages 1076 –1079, 2012.

[227] Peter J Zimetbaum and Mark E Josephson. *Practical Clinical Electrophysiology.* Wolters Kluwer Health/Lippincott Williams & Wilkins, 2009.

Publications

[1] Xianliang Wu, R. James Housden, YingLiang Ma, Benjamin Razavi, Kawal S. Rhode, and Daniel Rueckert. Fast catheter segmentation from echocardiographic sequences based on segmentation from corresponding X-ray fluoroscopy for cardiac catheterization interventions. In *IEEE Transactions on Medical Imaging*, 2014 (to appear).

[2] Xianliang Wu, R. James Housden, YingLiang Ma, Kawal S. Rhode, and Daniel Rueckert. A fast catheter segmentation and tracking from echocardiographic sequences based on corresponding X-ray fluoroscopic image segmentation and hierarchical graph modelling. In *ISBI*, pages 951–954, 2014.

[3] Xianliang Wu, R. James Housden, YingLiang Ma, Daniel Rueckert, and Kawal S. Rhode. Real-time catheter extraction from 2D X-ray fluoroscopic and 3D echocardiographic images for cardiac interventions. In *STACOM*, pages 198–206, 2012.

[4] Xianliang Wu, R. James Housden, Niharika Varma, YingLiang Ma, Kawal S. Rhode, and Daniel Rueckert. Fast catheter tracking in echocardiographic sequences for cardiac catheterization interventions. In *STACOM*, pages 171–179, 2013.

[5] Xianliang Wu, R. James Housden, Niharika Varma, YingLiang Ma, Daniel Rueckert, and Kawal S. Rhode. Catheter tracking in 3D echocardiographic sequences based on tracking in 2D X-ray sequences for cardiac catheterization interventions. In *ISBI*, pages 25–28, 2013.