

TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Informatik  
Lehrstuhl für Informatikanwendungen in der Medizin & Augmented Reality / I16

# Machine Learning for Medical Instrument Detection and Pose Estimation in Retinal Microsurgery

Mohamed Alsheakhali

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen  
Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr. Michael Gerndt  
Prüfer der Dissertation:  
1. Prof. Dr. Nassir Navab  
2. Prof. Dr. Farida Cheriet,  
Polytechnique Montréal, Canada

Die Dissertation wurde am 19.12.2016 bei der Technischen Universität München  
eingereicht und durch die Fakultät für Informatik am 19.05.2017 angenommen.

---

## Abstract

Instrument detection and pose estimation has attracted great interest in retinal microsurgery. Automatic detection of the instrument parts and estimating the instrument pose promote many applications to guide the surgeon in the operation room. One important application is the automatic positioning of Optical Coherence Tomography (OCT) scans to estimate the distance between the detected instrument tip and the retina to minimize the damage during the surgery. There are many other applications that employ the pose of the instrument such as activity recognition and surgical workflow analysis. This work addresses the problem of detecting the instrument parts (tips and/or the joint point) to estimate the pose and provide the OCT device with the required parameters to position its scans accordingly. At first, we detect the instrument tip along with shaft orientation. Then, we move to detect the three parts, which are two forceps tips and the joint point. Finally, we detect, in addition to forceps parts, the shaft orientation, and hence, we get all parameters needed for OCT positioning.

The primary contributions of this work are fourfold. In the first, we propose to use color information in conjunction with geometric structure of the instrument shaft to localize the instrument tip and the shaft's orientation. In the second approach, we propose a discriminative method to detect the instrument connecting point and the orientation. In this method, Convolutional Neural Network (CNN) is designed to detect the instrument parts separately, while a regression forest is trained to work on top of the CNN predictions in order to localize the joint point and estimate the instrument orientation in one step. The forest is trained on joint structural features of multiple instrument parts. In our third contribution, we formulate the problem as a regression task to predict the locations of the instrument left and right tips in addition to the joint point in 2D images. We introduce a new pipeline to incorporate only the reliable parts in the localization process. For that end, the training in this pipeline is done in a heuristic way by associating the features of the samples in the vicinity of the instrument parts with guiding information to improve the localization accuracy. Additionally, the pipeline integrates a module for the automatic recovery which is needed in cases of low images quality and instrument disappearance. In the fourth contribution, a Conditional Random Field (CRF) model of the instrument is proposed. This model employs the regression forest for unary detections which represents the confidence of each hypothesis in the image space. Prior information is modeled as potential functions to express the kinematic constraints of the instrument structure. The model predicts the locations of each part of the instrument as well as the shaft orientation. Therefore, this work presents different techniques to assist the surgeon in minimally invasive procedures. These techniques are not limited to retinal microsurgery but also can be applied to laparoscopic surgery.

---

---

## Zusammenfassung

Das Erkennen und die Lageschätzung von chirurgischen Instrumenten ist von großem Interesse im Bereich der retinalen Mikrochirurgie, denn das automatische Erkennen von Instrumententeilen und das Einschätzen der Pose ist die Grundlage vieler computergestützter Hilfestellungen für den Chirurgen während einer Operation. Zu den wichtigen Anwendungen gehört das automatische Positionieren der optischen Kohärenztomografie (OCT), welches das Einschätzen des Abstandes zwischen der detektierten Instrumentenspitze und der Retina ermöglicht und somit das Risiko einer Verletzung durch unabsichtlichen Kontakt minimiert. Weitere Anwendungsmöglichkeiten sind die Aktivitätserkennung und die objektive Arbeitsablaufanalyse im Operationsraum. Diese Dissertation behandelt das Detektionsproblem der Referenzpunkte des chirurgischen Instrumentes, gegeben durch die Spitzen und deren Verbindungspunkt, um die Pose abzuschätzen und dadurch dem OCT Gerät die für das Positionieren der Abtastung benötigten Parameter zu liefern. Im ersten Schritt wird nur die Instrumentenspitze und dessen Orientierung detektiert. Danach konzentrieren wir uns auf die Erkennung von den drei Referenzpunkten, die durch die zwei Spitzen und das Verbindungsstück gegeben sind. Schließlich wird zusätzlich zu den genannten Punkten auch die Orientierung des Schafts erkannt, um alle benötigten Parameter für die OCT Positionierung zu bekommen.

Der wesentliche Beitrag dieser Arbeit ist vierfältig: zunächst schlagen wir vor, die Farbinformation in Verbindung mit der geometrischen Struktur des Instrument Schafts zu benutzen, um die Instrumentenspitze und die Orientierung des Schafts zu schätzen. Im zweiten Ansatz schlagen wir eine diskriminative Methode vor, um den Verbindungspunkt und die Orientierung des Instruments in einem Schritt zu ermitteln. In dieser Methode wurde ein Convolutional Neural Network (CNN) entworfen, um die einzelnen Instrumententeile zu lokalisieren. Basierend auf den CNN Vorhersagen wurde ein Regression Forest trainiert, der die Verbindungsstelle und die Orientierung des Instruments in einem Schritt lokalisiert. Der Forest wurde auf gemeinsamen strukturellen Merkmalen der mehreren Instrumententeile trainiert. Im dritten Beitrag dieser Arbeit formulieren wir das Problem als Regressionsaufgabe, um zusätzlich zu den Verbindungspunkt auch die zwei Instrumentenspitzen in den 2D Bildern zu voraussagen. Wir stellen eine neue Algorithmen-Pipeline vor, in der nur zuverlässige Teile eingebunden werden. Um dies zu erreichen, wurde das Training in dieser Pipeline auf heuristische Weise durchgeführt, in dem die Merkmale der Stichprobe in der Nähe der Referenzpunkte mit leitenden Informationen assoziiert wurden, welche die Lokalisierungsgenauigkeit verbessern. Weiterhin wurde ein Modul für die automatische Korrektur integriert, das im Falle von schlechter Bildqualität und Instrumentenverschwinden notwendig ist. Im vierten Beitrag wird ein Conditional Random Field (CRF) Modell vorgestellt. In diesem Ansatz werden Regression Forests für eine unäre

---

Detektion eingesetzt, die die Wahrscheinlichkeit jeder Hypothese im Bildraum repräsentieren. Vorinformation werden als Potential Functions modelliert, um die kinematischen Nebenbedingungen der Instrumentenstruktur auszudrücken. Dieses Modell sagt die Position jedes Referenzpunktes des Instruments sowie die Schaftorientierung voraus. Daher präsentiert diese Arbeit verschiedene Techniken, um den Chirurgen in minimal-invasiven Verfahren zu unterstützen. Diese Techniken sind nicht auf retinale Mikrochirurgie beschränkt, sondern können auch auf laparoskopische Chirurgie angewendet werden.

---

---

## Acknowledgment

After four years of research for this thesis, the first person I would like to cordially thank is prof. Nassir Navab, the supervisor of my thesis. I cannot find enough words to express my appreciation to his support, motivational words and kind guidance. I am very proud to be one of his group (CAMP).

Secondly, I want to thank my parents and wife for supporting me in the moments of difficulties. My thanks go also to my daughter Nada who was born one month before submitting this work. Thanks also to my brothers, sisters and all my family who have been waiting so long for this great moment.

Furthermore, I would like to thank our project manager at Zeiss, Dr. Abouzar Eslami, for his supervision and guidance. Additionally, I would like to thank my wonderful colleagues Hessam Roodaki, Mehmet Yigitsoy, Shadi Albarqouni, Loic Peter, Nicola Rieke, Chiara di San Filippo, David Tan, Federico Tombari and Vasileios Belagiannis for their collaboration and the longtime of discussion and interaction.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgement</b>	<b>vi</b>
<b>Contents</b>	<b>vii</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Algorithms</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Medical Background . . . . .	2
1.2 Retinal Microsurgery . . . . .	3
1.3 Optical Coherence Tomography (OCT) . . . . .	3
1.4 Motivation . . . . .	4
1.5 Problem Statement and Challenges . . . . .	6
1.6 General Applications . . . . .	8
1.7 Contributions . . . . .	9
1.8 Thesis Outline . . . . .	10
<b>2 Machine Learning Background</b>	<b>13</b>
2.1 Random Forests . . . . .	14
2.1.1 Introduction . . . . .	14
2.1.2 Decision Trees . . . . .	15
2.1.3 Random Forests Concept . . . . .	17
2.2 Convolutional Neural Networks . . . . .	18
2.2.1 Basic Concepts . . . . .	18



## CONTENTS

---

2.2.2	Gradient Descent and Error Backpropagation . . . . .	21
2.2.3	Convolutional Neural Network Concept . . . . .	24
2.2.4	CNN Layers Types . . . . .	24
2.3	Conditional Random Fields . . . . .	28
2.3.1	Graphical Modelling . . . . .	29
2.3.2	CRF's Potential Functions . . . . .	32
2.3.3	Output Variables Inference . . . . .	32
2.3.4	Parameter Learning . . . . .	34
<b>3</b>	<b>Related Work</b>	<b>37</b>
3.1	Intensity-Based Instrument Tracking . . . . .	37
3.2	Instrument Tracking by Detection . . . . .	39
3.3	Instrument Pose Estimation . . . . .	39
3.4	State of the Art Techniques Summary . . . . .	40
<b>4</b>	<b>Color Information and Geometric Modelling</b>	<b>43</b>
4.1	Introduction . . . . .	43
4.1.1	Instrument Segmentation . . . . .	44
4.1.2	Structural Information Integration . . . . .	45
4.1.3	Instrument Tip and Centerline Detection . . . . .	46
4.1.4	Instrument Tip Tracking . . . . .	47
4.2	Experiments and Results . . . . .	48
4.2.1	Retinal Microscopic Datasets . . . . .	48
4.2.2	Datasets Evaluation. . . . .	49
4.3	Conclusion . . . . .	50
<b>5</b>	<b>Using Deep Learning for Articulated Instrument Detection</b>	<b>51</b>
5.1	Introduction . . . . .	51
5.2	Problem Formulation . . . . .	52
5.2.1	Unary Potentials Using CNN . . . . .	52
5.2.2	Pairwise Potentials . . . . .	55
5.2.3	Regularization Term . . . . .	55
5.3	Experiments and Results . . . . .	56
5.3.1	Public Dataset . . . . .	57
5.3.2	Zeiss Dataset . . . . .	58
5.4	Conclusions . . . . .	60

---

CONTENTS

---

<b>6</b>	<b>Deep Architecture for Instrument Pose Estimation</b>	<b>63</b>
6.1	Introduction . . . . .	63
6.2	Problem Formulation . . . . .	64
6.2.1	Deep Network Architecture . . . . .	64
6.2.2	Loss Function . . . . .	65
6.3	Experiments and Results . . . . .	65
6.3.1	Public Dataset . . . . .	65
6.3.2	Zeiss Dataset . . . . .	67
6.4	Conclusions . . . . .	70
<b>7</b>	<b>Instrument Pose Estimation Based on Reliable Hough Voting</b>	<b>71</b>
7.1	Introduction . . . . .	71
7.2	Hough Forest . . . . .	72
7.3	Proposed Method . . . . .	73
7.3.1	Forceps Joint Classification . . . . .	73
7.3.2	Pose Estimation . . . . .	74
7.4	Tracking and Recovery . . . . .	76
7.5	Experiments and Results . . . . .	77
7.5.1	Public Dataset . . . . .	78
7.5.2	Zeiss Dataset . . . . .	79
7.5.3	Laparoscopic Dataset . . . . .	81
7.6	Discussion . . . . .	82
7.7	Conclusions . . . . .	85
<b>8</b>	<b>CRF-Based Model for Forceps Pose Estimation</b>	<b>87</b>
8.1	Introduction . . . . .	87
8.2	Problem Formulation . . . . .	88
8.2.1	Unary Potentials . . . . .	90
8.2.2	Binary Translation potentials . . . . .	90
8.2.3	Ternary Potentials . . . . .	91
8.2.4	Quaternary Rotation Potential . . . . .	92
8.2.5	Inference of the Instrument Pose . . . . .	92
8.3	Experiments and Results . . . . .	94
8.3.1	Zeiss Dataset . . . . .	94
8.3.2	Public Dataset . . . . .	96
8.3.3	Laparoscopic Dataset . . . . .	98
8.4	Results Discussion . . . . .	99
8.5	Conclusions . . . . .	100

## CONTENTS

---

<b>9 Conclusion and Outlook</b>	<b>103</b>
9.1 Summary and Findings . . . . .	103
9.2 Limitations . . . . .	104
9.3 Future Work . . . . .	104
<b>A Instrument Type Detection Using CNN</b>	<b>107</b>
A.1 Proposed method . . . . .	107
A.1.1 Problem Formulation . . . . .	107
A.1.2 Convolutional Neural Network Structure . . . . .	108
A.2 Results . . . . .	109
A.2.1 Eye Phantom Datasets . . . . .	109
A.2.2 Real Microsurgery Datasets . . . . .	110
A.2.3 Results Analysis . . . . .	110
A.3 Conclusion . . . . .	111
<b>Bibliography</b>	<b>113</b>

# List of Tables

3.1	Methods Comparisons :(T = tip, LT = left tip, RT = right tip, JP= joint point, VT = work on vitrectomy, FC = work on Forceps, I = automatic Initialization, O = estimate orientation, R = real time) .	41
6.1	Strict PCP scores for different $\alpha$ values for public dataset sequences.	68
6.2	Strict PCP scores for different $\alpha$ values for Zeiss dataset sequences.	69
8.1	Strict PCP scores for $\alpha = 0.5$ on Zeiss Dataset . . . . .	94
8.2	Strict PCP scores for $\alpha = 0.5$ on Public and Laparoscopic(Lap) datasets . . . . .	99

LIST OF TABLES

---

# List of Algorithms

8.1 Inference Algorithm . . . . .	93
-----------------------------------	----

LIST OF ALGORITHMS

---

# List of Figures

1.1 Human eye cross-sectional view . . . . .	2
1.2 Retinal Microsurgery carried out on pig’s eye at Zeiss Laboratory	3
1.3 Left: Human eye interior view, Right: Posterior image after looking through microscope lens placed on top of eye lens . . . . .	4
1.4 OCT imaging system . . . . .	5
1.5 (Left) Microscopic Image with two OCT scans and three detected points labeled in cross signs, (Right) OCT depth information along each OCT scan. . . . .	6
1.6 Different instruments appearances of different surgeries . . . . .	8
1.7 Different views for the same pig’s eye surgery at Zeiss laboratory	8
2.1 Decision Tree Example: At each node a question is asked and the samples go either left or right until a leaf node is reached. The leaves store class distribution. . . . .	15
2.2 Instrument Example illustrating the concept of deep learning architectures. . . . .	19
2.3 Illustration of a single neuron. . . . .	20
2.4 Example of multi-layer perceptron. . . . .	21
2.5 Example explaining the concept of gradient descent. . . . .	23
2.6 A Convolutional Neural Network with one convolution layer, one pooling layer, two fully connected layers, and output layer of four classes. . . . .	25
2.7 Graphical Model: An undirected graphical model that would correspond to the medical instrument joints. . . . .	29
2.8 Factor Graph: This graph is a representation of the undirected graph model shown in Figure 2.7. The circles are random variables and the black boxes are potential functions or factors representing the dependencies between variables. . . . .	31
4.1 RGB image with its L*a*b* Transformation . . . . .	44
4.2 Color information extracted from a* channel . . . . .	45



LIST OF FIGURES

---

4.3	Edge Image . . . . .	45
4.4	(a) The detected Hough lines in the Edge image. (b) The same Hough lines obtained from the edges image and superimposed on the refined a* channel.(c) The tool model where the mid-line should lay on each Hough line. . . . .	46
4.5	Some cases where the instrument detected edges are not parallel.	47
4.6	Random samples from different datasets with different conditions. The first top row is from the first dataset where the red component is prominent and the instrument is evenly illuminated. The second bottom row is from the second dataset where the green component is prominent, and the instrument is unevenly illuminated . . . . .	48
4.7	Instrument tip detection accuracy measurements. . . . .	49
4.8	Instrument centerline detection accuracy measurements. . . . .	49
5.1	The designed CNN: Filters sizes = 5x5, Pooling size= 2x2. The numbers of features channels are 20 at layer 1 and 50 at layer 2. . . . .	52
5.2	Patches samples of size 28 × 28, where each row was chosen from a different class: center, shaft, and background respectively. . . . .	53
5.3	(a) Instrument example. (b) CNN output example . . . . .	54
5.4	Regression Random Forest Model learned on joint features of point pairs which represent a configuration . . . . .	56
5.5	Results for each sequence of the public dataset when trained and tested on separate sequences. The bounding box is centered on the detected instrument’s center. . . . .	57
5.6	The results for the full dataset, when learned on the first halves from each sequence and tested on the second halves . . . . .	58
5.7	The objective function and error curves after each epoch of CNN training from the full public dataset . . . . .	59
5.8	The results based on the Angular Threshold for both cases . . . . .	59
5.9	The results for the full Zeiss dataset. . . . .	60
5.10	Samples of the results showing the detected joint point and estimated centerline. . . . .	60
5.11	The angular thresholds results for the full dataset. . . . .	61
5.12	The objective function and error curves after each epoch of CNN training from the full Zeiss dataset . . . . .	61
6.1	Deep convolutional architecture for instrument pose estimation: Convolutions use different kernel sizes as indicated inside each box. . . . .	64
6.2	Results for each sequence of the public dataset, when learned and tested on separate sequences. . . . .	66

LIST OF FIGURES

---

6.3	Accuracy Thresholds results for testing the model that trained from all sequences . . . . .	67
6.4	The objective function and error curves after each epoch of CNN training from the full public dataset . . . . .	67
6.5	the results for the full dataset, when learned on the first halves from each sequence and tested on the second halves. . . . .	68
6.6	The objective function and error curves after each epoch of CNN training from the full Zeiss dataset . . . . .	69
7.1	The whole pipeline of the proposed method. . . . .	74
7.2	Accuracy Thresholds performance: The results for the three sequences when each trained from half of the images and the testing was done on the second half. . . . .	77
7.3	Accuracy Thresholds performance: The results for the three sequences when trained from the first halves of the images together and tested on the second halves. . . . .	78
7.4	Strict PCP scores for the full dataset using our proposed method and TPOS. . . . .	78
7.5	Four sequences from Zeiss dataset with different instrument types and different light conditions. . . . .	79
7.6	Accuracy Thresholds performance: The results of the four sequences when each was trained from half of the images and tested on the second half. . . . .	80
7.7	Accuracy Thresholds performance: The results for the four sequences when trained from the first halves of the images together and tested on the second halves. . . . .	81
7.8	Two unseen sequences from Zeiss Dataset, each sequence was taken from different surgery. . . . .	82
7.9	Threshold accuracy for detecting left, right, and center points of the instrument in sequence 5 . . . . .	82
7.10	Threshold accuracy for detecting left, right, and center points of the instrument in sequence 6 . . . . .	83
7.11	Strict PCP scores for the full Zeiss dataset. . . . .	83
7.12	(a) a qualitative example of the estimated pose for laparoscopic dataset, (b) pixel-wise accuracy of predictions for each of the three forceps joints, (c) strict PCP scores for left and right gripper parts predictions. . . . .	84
8.1	(Left) Target pose estimation, (Right) The factor graph for the Forceps: 4 variables (left (L), right (R), center (C), and shaft (S)) are used with different types of constraints are presented with different edge colors: black (translation), green (rotation), red (relative length), and blue (consistency) . . . . .	89

LIST OF FIGURES

---

8.2	Connectivity modeling using Bézier curves where the dashed lines are orthogonal vectors and the position of the control point $p$ is placed along one of those vectors with different displacements from the center point. . . . .	90
8.3	Eight samples from each sequence of Zeiss dataset with pose estimation . . . . .	95
8.4	The accuracy threshold scores for left, right and center points respectively . . . . .	96
8.5	Angular Threshold scores for Zeiss sequences. . . . .	97
8.6	Threshold accuracy for each of the public sequences separately . . . . .	97
8.7	Threshold accuracy for laparoscopic dataset . . . . .	98
8.8	Accuracy threshold for different forceps joints of the public (full and separate sequences) and laparoscopic (Lap) datasets. . . . .	98
8.9	Angular Threshold scores for Public and Laparoscopic sequences . . . . .	99
A.1	Convolutional Neural Network for Tool Type Detection. . . . .	108
A.2	Four different types of surgical instruments. . . . .	109
A.3	Confusion matrix of eye phantom group. . . . .	109
A.4	Three different tool types with different poses. . . . .	110
A.5	Confusion matrix for real microscopic images. . . . .	110

LIST OF FIGURES

---

# Chapter 1

## Introduction

Retinal Microsurgery is among the most delicate operations, in which a micro-precision handling is required in tasks such as retinal membrane peeling. Carrying out such surgeries requires manipulating retina surface with medical instruments such as vitrectomy or forceps. An efficient feedback for the distance between the instrument tip and the retina is a demanding requirement to minimize tissue damage caused by unintentional touch of retina. This distance can't be estimated from only microscopic optical images. Advances in computer science, mathematics and physics over the last decades have stimulated the development of new imaging technologies such as Optical Coherence Tomography (OCT). This imaging technology has been equipped and integrated with nowadays generation of ophthalmic surgical microscopes. The usage of OCT technology in these microscopes has allowed for the visualization of sub-retinal structure information [11] and the segmentation of the retinal anatomical layers [62] which are less than  $10\mu m$  thick. Moreover, OCT imaging allowed for the development of retinal pathologies diagnosis. The capabilities of OCT imaging can benefit as well retina treatment and surgery in many cases. One of the research fields which attracts a lot of attention over the last years is the estimation of the instrument depth information. Extracting such information constantly over time requires assistance of instrument detection and tracking algorithms to localize landmarks for OCT device. However, the development of such algorithms is still a challenging task due to the complex operation environment as well as to the structure-less characteristic of the instrument itself. In this thesis, we address the problem of medical instrument detection, tracking and pose estimation in retinal microsurgery. Our ultimate goal is to estimate the instrument joints coordinates in 2D image space. Extracting these coordinates in real time performance paves the way for many applications to guide surgical interventions.

In this chapter, we begin first with medical background about retinal diseases and microsurgery. Next, OCT imaging for retinal microsurgery is presented with the motivation and problem statement. Finally, we list the contributions of this work and give the outline for the overall structure of this thesis.

## 1.1 Medical Background

Microsurgery has been applied as treatment operation for many eye diseases affecting the retina. Epiretinal membrane (EM), which is also called macular pucker, is one of those diseases caused by aging process, diabetes, previous trauma or post vitreous detachment (PVD) [122]. EM has been described for the first time by Iwanoff [54] in 1865 who showed that this ocular pathology affected 7% of the population who are over 50 years old. To explain the effect of this pathology, let's consider the eye structure shown in shown in Figure 1.1. The vitreous is the transparent gel that fills the blank space in the center of the eyeball between the lens and retina. As time goes by, changes in vitreous can cause a number of problems in the eye including wrinkling of the retina. The wrinkling is due to a membrane covering the surface of the retina. This membrane is formed as a result of immune system response which forces the retinal cells to converge in the macular area. The membrane, which is attached to the surface of the retina, has a tendency to contract. Therefore, it causes the retina to wrinkle and results in distortions of vision within the macula area which has the finest details of vision. The distortions can change the perceived objects dimensions and create a field dependent aniseikonia [18] which cannot be treated with optical glasses [93]. The treatment of EM is accomplished by removing or peeling the surface membrane by microsurgery. The first step in this procedure is to replace the vitreous, and then the surgeon, with fine forceps, can grab the edge of the membrane delicately and remove it from the eye. In cases where the membrane edges are difficult to be recognized, a cut on the thickest part of the membrane is created with micro-vitreoretinal blade [99]. Hence, the created cut is used as the starting point for the peeling operation which is performed by forceps instrument. The movement of the forceps during peeling should be in a circular fashion in order not to damage the retinal tissue. This treatment would enable the retina surface to get smooth back and improve the vision again. Even though surgery is not usually recommended due to its complications such as bleeding in the eye and cataracts [5], it is the only effective solution when distortions of vision are severe.

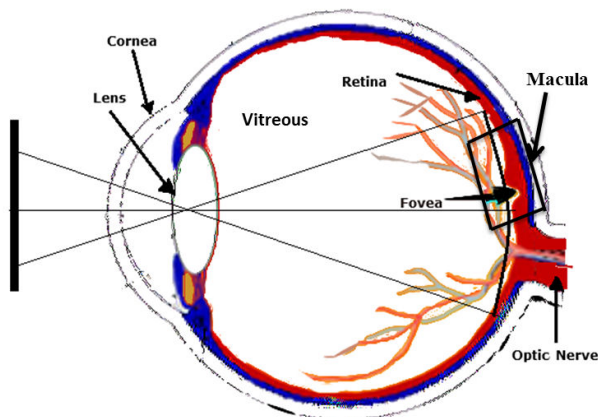


Figure 1.1: Human eye cross-sectional view

## 1.2 Retinal Microsurgery

Microsurgery is the treatment of small parts of the human body which the surgeon can't access with naked eyes. Handling these parts requires high optical magnification which performed using high-precision microscopes. For retinal microsurgery as shown in Figure. 1.2, surgeon places microscope lens on top of eye's lens, and retinal surface can be seen through these lens. The manipulation of retinal tissue is carried out by hand-held instrument which could be a forceps or vitrectomy.



Figure 1.2: Retinal Microsurgery carried out on pig's eye at Zeiss Laboratory

Retinal Microsurgery starts by the creation of three access ports, labeled  $P1$ ,  $P2$  and  $P3$  in the white area of the eye as shown in Figure 1.3 (Left). The first one  $P1$  is called the infusion port which used to pump fluid into the eye to replace the vitreous removed from it. The other two access ports,  $P2$  and  $P3$ , are used to access retina tissue into the vitreous cavity. They allow to remove vitreous from the eye as well as to access the macula and the rest of the retina. Surgeon uses one of the ports to insert light pipe, while the other port is used to insert the peeling instrument. High resolution microscope lens are placed on top of eye lens where under high magnification the surgeon can access the posterior area of the eye. In this area, the surgeon can see through microscope the retina tissues and vessels as shown in Figure 1.3 (Right), and peel the target membrane off the eye. After the completion of the peeling operation, the ports are easily removed and the eye is sealed up without any sutures. The most delicate part of this surgery is the peeling operation which requires a special care to access retina surface. To increase safety and minimize retina damage, instrument depth information should be maintained over surgery time. Therefore, the assistance of OCT imaging would be the promising technique to accomplish such a task.

## 1.3 Optical Coherence Tomography (OCT)

Optical Coherence Tomography (OCT) is a powerful imaging modality which can generate cross-sectional images with high resolution for small size

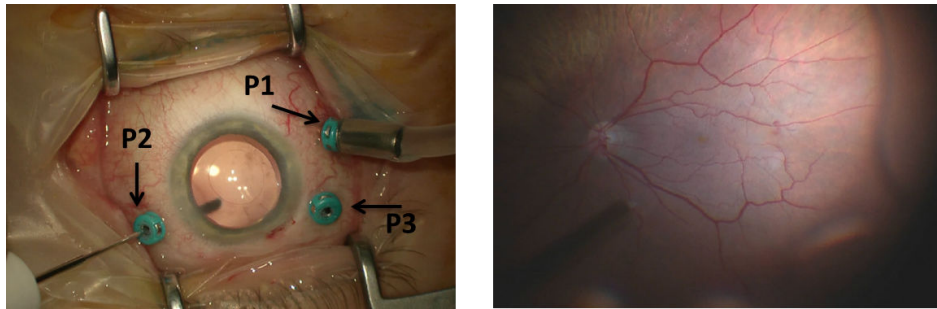


Figure 1.3: Left: Human eye interior view, Right: Posterior image after looking through microscope lens placed on top of eye lens

tissues. It is a non-invasive imaging technique which uses near-infrared light with high penetrating abilities into the scattering medium. This allows it to capture fine details in the range of micrometers of tissue structure. Having these features, OCT is widely used in different applications for ophthalmology including early diagnosis, detection and tracking of diseases.

A typical OCT imaging system is shown in Figure. 1.4. It consists mainly of light source, beam splitter, reference mirror and photo detector [9]. The working principle is based on low coherence interferometry [87]. The source emits a light beam to the object being imaged. Once the light reaches the beam splitter, it is splitted into two paths: one towards the reference mirror and the other to the object. Most of the light is scattered once hitting the object. However, the reflected lights from both paths are collected on a photo detector which shows the interference pattern. This pattern shows a high interference signal when the reflected beams from both paths have traveled roughly the same optical length. The profile of such signal, called A-scan, shows the location of structures within the object of interest along one axial dimension. Obtaining a cross-sectional image, called B-scan, can be achieved by getting A-scans for a series of object's samples. The formation of the axial A-scans and cross-sectional B-scan depends on the OCT imaging system. The first widely used system is called Time Domain system (TD-OCT), in which the reference mirror is moving in a linear way to change the reference optical path. This setup would allow the detection of structures at different distances by matching their optical path length with the adjustable reference path. Therefore, the detected signal consists of a combination of a DC component and an interference component carrying depth information of the sample being imaged. The other system is called Fourier Domain OCT (FD-OCT). In this imaging system, there is no need to move the reference mirror or any other part, and the photo detector is replaced with a spectrometer. This allows for higher speed imaging in comparison with TD-OCT systems.

## 1.4 Motivation

While OCT imaging can be applied to stationary tissues to get depth information, applying it to a moving instrument requires using detection



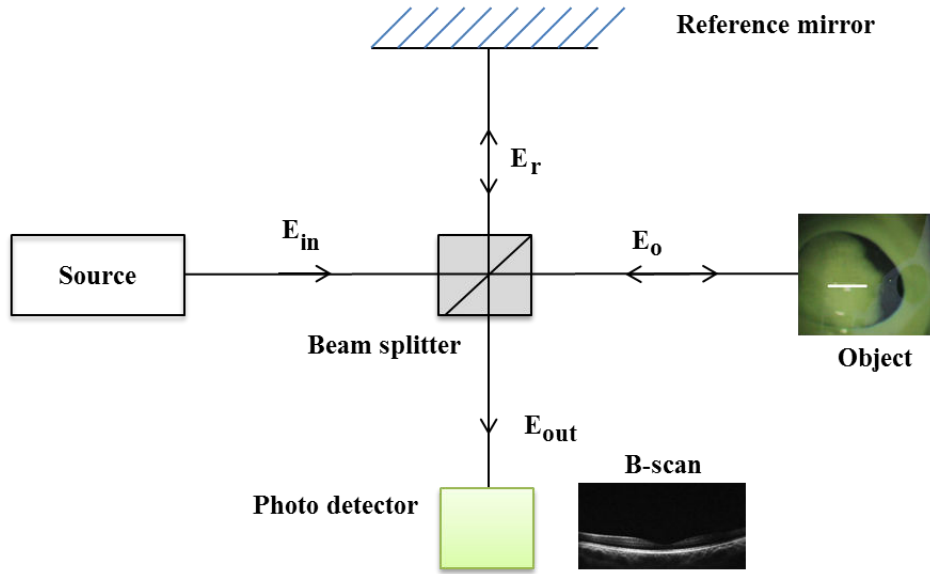


Figure 1.4: OCT imaging system

and tracking algorithms in advance. These algorithms have been addressed from different perspectives based on surgery type, available modalities, and number of used cameras (i.e. Monocular or stereo). Numerous studies for robotic-assisted surgery [4, 7, 101, 110, 38, 123] have been done to track medical instruments for minimally invasive procedures. Using stereo camera [119] has been also employed for instrument tracking in laparoscopic surgery to handle the limitations of the single view imaging. The equipment in recent ophthalmic surgical microscopes allows the usage of OCT images in additional to the optical ones. Even though most of the instrument detection and tracking methods [67, 105] can provide visual assistance during surgery by localizing instrument tip, still these methods tend to fail at real in-vivo surgery. Additionally, they can't extract all parameters required for the full benefit of OCT scans in order to achieve minimally invasive procedures. The current trend to minimize retina damage during surgery is to integrate OCT with reliable and real time instrument detector. OCT device requires prior extraction of some reliable points on the instrument body serving as landmarks for OCT imaging scans. The benefit of these scans is to have depth information at the landmarks locations in the 2D microscopic images. Therefore, a tangible feedback is given during surgery about the distance between the retina tissue and the instrument's part being scanned. To elaborate the interaction between OCT scans and real time detector, Let us consider Figure 1.5. The function of the real time detector is to estimate the coordinates of the points  $A, B$ , and  $C$ , labeled as cross signs in Figure 1.5 (Left), in the 2D image space. Therefore, the estimated coordinates are given to the OCT device in order to position the OCT scans accordingly. Figure 1.5(Left) shows two OCT B-scans. The first one, labeled with white color, passes through the detected points  $A$  and  $B$ , and the corresponding OCT depth image is shown in Figure 1.5 (Right:Top). It is obvious that the two jumps in the horizontal intensity profile along the retina surface correspond to the instrument

two tips  $A$  and  $B$ , which also reflect the distances from each tip to the retina surface. The second depth image shown in Figure 1.5(Right:bottom) is associated with the blue OCT B-scan in Figure 1.5(Left). Here the jump in the retinal surface intensity profile reflects the distance between the connecting point  $C$  and the retina surface. Therefore, augmenting the scene with depth information for the most interesting points to the surgeon requires reliable, robust and real time detection algorithms. The more landmarks we can extract using specific detectors, the more reliable estimated depth information we get, and hence, the safer the procedure is. In this thesis, different approaches are proposed for medical instrument detection and tracking in order to initialize and to work interactively with the OCT imaging. While the tracking and detection algorithms run on microscopic 2D images, OCT scans give the third dimension which is a promising way towards minimally invasive procedures. Furthermore, in this thesis, the proposed methods go beyond tracking of a single point in 2D images to more complicated task which is the pose estimation of articulated forceps used in the peeling operation. The pose estimation would give the coordinates of different joints of the forceps and estimate the state of the instrument (i.e. open or close). Most importantly, it localizes the instrument tips which grab the surface membrane from the retina. Finally, the ultimate goal of these approaches is to step forward in the direction of computer-assisted surgery to minimize the unintentional damage during retina surgery.

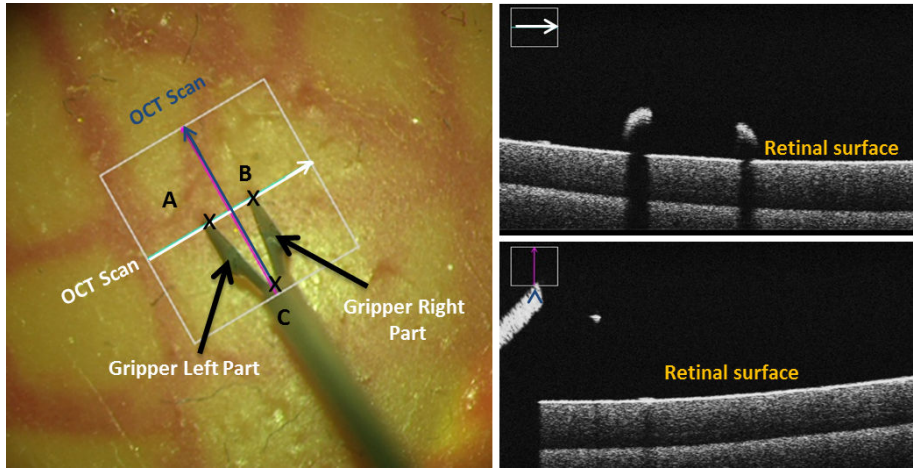


Figure 1.5: (Left) Microscopic Image with two OCT scans and three detected points labeled in cross signs, (Right) OCT depth information along each OCT scan.

## 1.5 Problem Statement and Challenges

In this thesis, we address the problem of medical instruments detection, tracking and pose estimation in retinal microsurgery. We regard this problem as the first and most important step in the process of estimating the distance between the instrument parts and the retina. Once the instrument joints

have been detected in real time, the OCT device would position the B-scans automatically to the detected target, and the depth would be estimated and presented to the surgeon during live surgery. However, detecting the instrument joints in the 2D image space is the most challenging issue in this framework. These challenges are due to many factors:

**Illumination variation :** Illumination variation has complex effects on the image of an object[24]. In retinal microsurgery, light pipe is completely controlled by the surgeon who directs it towards different parts of the retina. Moreover, the distance between the light pipe and the retina changes continuously according to the movement of the surgeon's hand. The changes in the light conditions introduce spurious edges and result in large variations which are extremely difficult to learn or analyze. Figure 1.6 shows different light conditions which result in light reflections along the instrument body and different appearance of the retinal surface.

**Cluttered non-static background :** The presence of vessels, optical disk, light pipe, instrument shadow and retinal lesions turn to have a significant influence on the performance of most existing detection and tracking algorithms. Moreover, continuous movements of both the background and the instrument complicate the creation of separate model for each. Additionally, the movement in the background is not only due the internal fluid movements, but also the eyeball itself is free to move in the eye cavity during surgery which makes the retina movement out of control.

**Instrument modelling :** The medical instrument can be described as a texture-less object [72] which doesn't have a fixed geometric shape like for example human face. Modelling structure-less object is more difficult and it is of high interest in machine vision [51, 70, 76, 97]. In contrast with human face detection, medical instruments consists of edges and corners which can appear at any orientation and with different opening degrees as shown in Figure 1.6. This doesn't suppose any clear geometric shape of the instrument at hand. Moreover, the instrument is moving in 3D space during surgery while we can access only 2D images. Hence, some parts might be occluded based on the rotation of the instrument in the 3D eye space which makes modelling the instrument very complicated task. In addition of being structure-less object, metallic instrument body could be highly affected by light reflection that causes some parts of the instrument to be totally invisible. Missing parts of the object would make the detection task more challenging in general.

In this thesis, most of the proposed methods address the problem of tracking the instrument relying on tracking by detection. Even though intensity-based tracking approaches [67, 88] of instrument have attracted special attention in medical imaging, still we believe tracking by detection turns to be the most promising solution in computer-assisted surgery. Unlike intensity-based tracker, tracking by detection algorithm can handle the manual initialization problem with no need of surgeon's input. This is regarded as a key advantage to bring like these computer-assisted techniques into existence. To highlight this advantage,

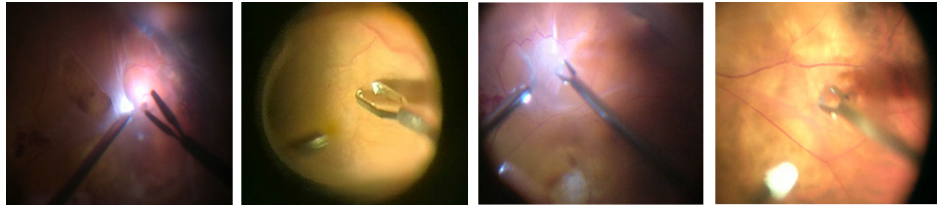


Figure 1.6: Different instruments appearances of different surgeries

let us consider the influence of manual initialization on the progress of the in-vivo surgery. Figure 1.7 shows retinal microsurgery done on pig's eyes at Zeiss laboratory which shows how busy the surgeon is during the operation time. In normal retinal microsurgery, surgeon uses both hands to hold with one the light pipe and with the other the peeling instrument while accessing the retina through microscope lens. Moreover, he uses his foot for turning the light on/off, switching the OCT on/off, changing light filters, turning the keratoscope on/off, ... etc. Therefore, manual initialization introduces a serious problem from the clinician's prospective which requires interrupting the surgeon to provide more input information at the expense of other tasks. Hence, it represents overhead and stress for the surgeon and prolongs the operation time. Robust and reliable computer-based assistance is a very demanding requirement to keep the surgeon focusing on the tasks at hand without interruption.

This thesis introduces efficient solutions for instrument pose estimation by detecting the instrument joints using state of the art computer vision methods and tracking these joint over time without interrupting the surgeon.

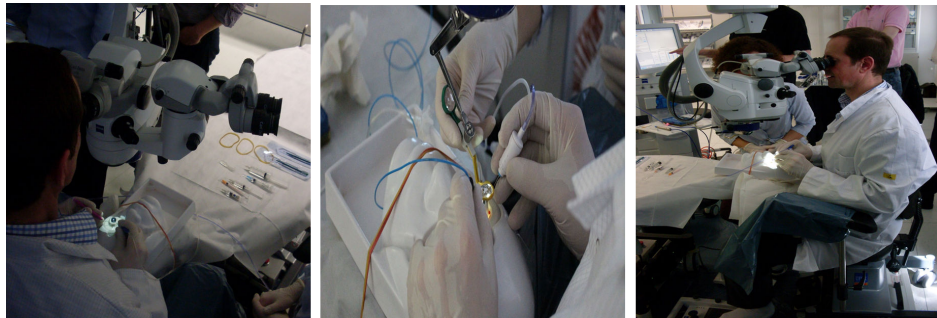


Figure 1.7: Different views for the same pig's eye surgery at Zeiss laboratory

## 1.6 General Applications

Real time detection and tracking of medical instrument joints can pave the way for advanced computer-aided support. One example is the automatic positioning of the intraoperative OCT (iOCT) during the in-vivo retinal surgery. In the current framework, surgeon has to position manually OCT B-scans to the position of interest. Automatic positioning allows OCT scans to follow specific parts of the instrument constantly and with no need for human intervention.

Since the potential damage during the peeling operation occurs as a result of instrument tips touch to the retina, following the tips has more interest for minimally invasive procedures. Therefore, the localization of the instrument tips rather than the instrument connecting point enables the estimation of the distance between the instrument tips and the retina tissue. Additionally, visualizing this distance to the surgeon has a significant impact to minimize the damage in this delicate operation. This distance can be visualized close to the instrument tips [91] so the surgeon doesn't need to switch between different displays. Furthermore, estimating the instrument orientation can optimize for positioning the OCT scans to achieve the maximal benefit of OCT imaging.

Robot-assisted vascular microsurgery [47, 55] is another interesting area to benefit from the instrument tip detection and pose estimation. In this kind of microsurgery, it is required to place a  $2 - 3 \mu m$  glass micropipette inside retinal vessels ( $20 - 130 \mu m$ ). The micropipette must remain in the vessel for up to few minutes in order to inject drugs or take pressure measurement after the insertion [55]. While many commercial manipulators are available, the eye geometry doesn't accept their use [82, 49]. Integrating the instrument tip detection in 2D images together with depth information using iOCT gives the tip coordinates in the 3D eye space. Therefore, it can guide the micropipette insertion into the target vessel. This can be accomplished by defining the intended depth to go into the vessel to the robot after positioning the micropipette appropriately to that vessel. The solutions presented in this thesis are easy to integrate with robot software and can't be hampered by the eye geometry.

Understanding the scene and activities during surgery has attracted special attention in interventional imaging. Activities understanding can't be achieved by detecting only a single point of interest like for example the instrument tip or the connecting point. Therefore, the aim of the pose estimation is to give an understanding of the big picture and to relate the detections of different parts to each other. Therefore, pose estimation can identify a specific state of the instrument required to perform a certain task during surgery. For example, the knowledge of the positions of the two forceps tips and the forceps connecting point provides us with the forceps opening degree. Following the estimated opening degrees over time helps in understanding the activity being done by the forceps (i.e. grabbing, releasing... etc.). Moreover, pose estimation applications can be extended from understanding the scene to deciding the appropriate action based on the activity.

Surgical workflow analysis is another application of instrument tracking and detection algorithms. Retina microsurgery passes through different phases from the beginning of the operation till the end. In certain phases, the surgeon might not need any instrument. Being able to recognize the appearance and disappearance of the instrument can assist in automatic detection and identifying of certain phases of the surgical workflow.

## 1.7 Contributions

To achieve our objectives, we introduce a number of novel methods for instrument detection, tracking and pose estimation in real in-vivo retina

microsurgery. The primary contributions can be summarized as follows:

- We investigated the problem of medical instrument tracking in real time. We make use of the metallic characteristic of the instrument to propose color-based segmentation approach. Once the segmentation is done, geometric constraints are imposed to localize the instrument tip and the orientation of the shaft. A database of hundreds of images was created from real retina microsurgery for the purpose of results evaluation.
- We focus on the problem of instrument detection by employing the powerful detection capabilities of the deep learning. The instrument is regarded as an articulated object where a probability map is obtained for each of its parts using deep learning-based discriminative model. Moreover, the orientation of the instrument shaft is predicted from the estimated maps by regressing the shaft end points.
- We investigated the problem of instrument pose estimation. We define the pose here to be the positions of the instrument joints in the 2D images. A discriminative Hough-based model is proposed to regress the instrument joints. To this end, training the model is done in a heuristic way so at testing time only reliable samples can participate in the joints predictions. Tracking is implicitly done by doing the predictions at the video frame rate speed while making use of the temporal information.
- The problem of the pose estimation is defined in different way so we can predict not only the joint coordinates but also we can estimate the orientation of the shaft. A novel CRF-based model is proposed to localize the instrument joints relying on part-base detectors and geometric 2D instrument structure priors. With this model, most of the important parameters for the OCT device are predicted. Therefore, instrument joints can be scanned with properly oriented OCT B-scans.

## 1.8 Thesis Outline

We provide an overview of each chapter of this thesis along with the related published or under submission work.

**Chapter 2.** We present the machine learning background of this thesis. In particular, we go through random forests, deep learning, and conditional random fields (CRFs) which form the base of our proposed algorithms.

**Chapter 3.** We give an overview of the work done in the area of instrument detection, tracking and pose estimation. The achievements and limitations of these methods are discussed.

**Chapter 4.** In this chapter, a color based segmentation method is introduced. The method localizes the instrument tip and orientation by imposing geometric

constraints to optimize for the instrument tip in the segmentation map. Related work:

- Alsheakhali, M., Yigitsoy, M., Eslami, A., Navab, N. (2015, March). Surgical tool detection and tracking in retinal microsurgery. In SPIE Medical Imaging (pp. 941511-941511). International Society for Optics and Photonics.
- Alsheakhali, Mohamed, et al. "Real Time Medical Instrument Detection and Tracking in Microsurgery." *Bildverarbeitung fuer die Medizin 2015*. Springer Berlin Heidelberg, 2015. 185-190.

**Chapter 5.** In this chapter, the deep learning is used to detect each part of the articulated instrument. The detection maps are employed to refine the results based on joint structural information of the defined parts. A regression forest is used for this refinement after being trained from joint structural features. Related work:

- Alsheakhali, Mohamed, Abouzar Eslami, and Nassir Navab. "Detection of articulated instruments in retinal microsurgery." 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI). IEEE, 2016.

**Chapter 6.** In this chapter, the deep learning is used to regress the instrument joints coordinates within the 2D image space. A new deep network is proposed to model the pose estimation as a regression deep learning.

**Chapter 7.** The work of this chapter aims at estimating the pose of the instrument. The Hough forest is employed for this purpose by re-implementing the classification and regression phases of this forests in a cascade way. In this way, we can integrate our scheme to activate the automatic recovery process after any tracking failure. Only reliable parts are involved in the training and testing processes to cast votes during the prediction of the joints coordinates. Related work:

- Alsheakhali, Mohamed, Abouzar Eslami, Hessam Roodaki, and Nassir Navab. "Instrument Pose Estimation Using Reliable Part-Based Voting in Retinal Microsurgery". The 8th International Conference on Information Processing in Computer-Assisted Interventions (IPCAI) (submitted)

**Chapter 8.** In this chapter, we model the dependencies between the instrument joints using Conditional Random Field (CRF). In this model, the random forest is used as the part-based detector (i.e. unary potentials). Higher order dependencies are implemented to model the translation, rotation and scale changes of the instrument. The CRF model is trained in this work to infer the configuration of the instrument which is considered the estimated pose.

- Alsheakhali, Mohamed, Abouzar Eslami, Hessam Roodaki, and Nassir Navab. "CRF-Based Model for Instrument Detection and Pose

## CHAPTER 1. INTRODUCTION

---

Estimation in Retinal Microsurgery". *Computational and Mathematical Methods in Medicine*, vol. 2016, Article ID 1067509, 10 pages, 2016. doi:10.1155/2016/1067509.

**Chapter 9.** We conclude our work by presenting our findings, the limitations of the proposed methods and our suggestions for future work.



## Chapter 2

# Machine Learning Background

Machine learning has become one of the most essential sources of information technology and data analysis. The demand for machine learning has grown over the last decades according to the increasing amount of data becoming available from clinical and industrial devices. The analysis of data became a necessity to understand the behavior of any system and to extract the most significant piece of information influencing its performance. To that end, machine learning algorithms find a mapping from input signals to output values [21]. As an example from medical applications, images of cells represented by intensity values can be considered as input data along with their labels (e.g. has a disease or not), where machine learning algorithm task is to find a mapping between intensity values and existence of that disease. Direct mapping from input to output is often a very complex task [31]. The complexity comes from the potential non-linear mapping required to accomplish the task at hand and the embedded noise into the input samples. Moreover, defining the most discriminative features is not an easy task and has a high impact on the accuracy of the mapping techniques. Furthermore, mapping might require handling missing or unbalanced input data. This is why using linear mapping can't always discover the relation between input and output. Therefore, non-linear and complex mapping functions are needed to be trained.

Generally, machine learning algorithms use training set to find the appropriate mapping. The training set consists of  $n$  input vectors  $X = \{x_1, x_2, \dots, x_n\}$  along with optional output vectors  $Y = \{y_1, y_2, \dots, y_m\}$ . During training, the mapping parameters, which is also called the model parameters, are adapted to optimize the mapping between input training data and output vectors [21]. This mapping has the form  $Y_i = f(X_i)$ . The quality of this trained model can be identified by applying the mapping function with the trained model parameters on new unseen samples. Those samples are called the testing set and the ability to map inputs from testing set to the correct associated outputs determines the generalizability of the model.

Machine learning algorithms can be categorized into many groups. However, the mostly used algorithms are related to either supervised or unsupervised learning. In supervised learning [21], each training sample is associated with an output vector. This vector can be a set of labels e.g. (color, digit, healthy

or not,... etc), where each label is a discrete value which we are interested to predict during testing new samples. This type of supervised learning is called classification. On the other hand, when the output has a continuous value to be predicted, then the problem is defined as a regression problem. A regression example could be the prediction of the location of tumor within the image space or to predict the area or the volume of a certain organ of human body.

In unsupervised learning [21], the training data has only the data inputs without association with any output vectors. Unsupervised learning has different applications, such as data clustering and density estimation of the input data. Clustering algorithms try to discover the similarities between input samples and group them accordingly. During testing, new samples are assigned to one group based on only their features similarity. For density estimation algorithms, estimating the distribution of the input within the feature space is the main objective.

Additionally, there are other categories using unlabeled data in supervised learning problems. This is called semi-Supervised learning [124] where part of the data is labeled. The learning algorithm tries to cluster the unlabeled data with the guidance of the labeled samples. Combining unlabeled samples with small amount of labeled data in one learning algorithm can produce more improvement in the detection accuracy [27].

In this thesis, different supervised learning algorithms have been employed for classification and regression problems. We focus on Random Forests [25] for classification and regression in this work. Basically, Random Forests model the posterior distributions using the extracted hand-crafted features, and they are presented in section 2.1. Automatic feature extraction using Convolutional Neural Networks (CNNs) [59] has also been investigated. CNNs, which are types of artificial neural networks, are designed to find a non-linear representation of the input data relying mainly on convolutional operations. CNNs are presented in details in section 2.2. In most machine learning applications, the outputs of classifiers or regressors can be integrated with graphical models to impose geometrical constraints on a certain object. Conditional Random Fields (CRFs), which are types of probabilistic graphical models, are used in this thesis to model the kinematics of the instrument. Modelling and inference using CRFs are explained in section 2.3.

## **2.1 Random Forests**

### **2.1.1 Introduction**

In the last few years, Random Forests [25] have been applied in many different tasks, including classification, regression, semi-supervised learning and density estimation, where their achieved performance has been proven to be the state-of-the-art in many applications. Random Forests gain their power from its ability to combine several weak learners into one strong learner. Each weak learner is working on random subsets of the whole available dataset during model training stage. Weak learners are trained independently from each other and can run in a parallel way. Moreover, relying on several such learners

gives Random Forests the ability to generalize the trained model. Furthermore, injection of randomness during training makes these forests robust and highly scalable to large datasets and improves the generalization feature. Basically, a Random Forest is an ensemble of several de-correlated decision trees (weak learners), which will be covered in the following section.

### 2.1.2 Decision Trees

Decision trees can be defined as "a set of questions organized in a hierarchal manner" [32]. The decision tree can be seen as an acyclic graph where the direction of data flow is from the root node to the leaves. The goal of the decision tree is to find a relation between observations and output classes. It divides the observations into subgroups where each subgroup is used to build a local model characterized by the class distribution in this group which is called the posterior distribution. Each internal node stores a decision function which can be a simple test question. Depending on the answer of that question, the input data sample goes down to either left or right node. At the new node, a new test is applied and the data sample continues going down until a terminal node is arrived which is called a leaf node. Each leaf node stores the class distribution of the samples arrived this node during the training stage. For example, Let's consider the decision tree shown in Figure 2.1 where the input features are denoted by  $F = (x_1, x_2) \in \mathbb{R}^2$ . At testing time, each sample passes down to a particular leaf node based on the values of  $x_1$  and  $x_2$ . The leaves nodes store the distribution of three classes. Therefore, the output of the decision tree for each sample is either the whole distribution or only the class which has the maximum probability in that distribution. In this case the random forest is called classification random forest.

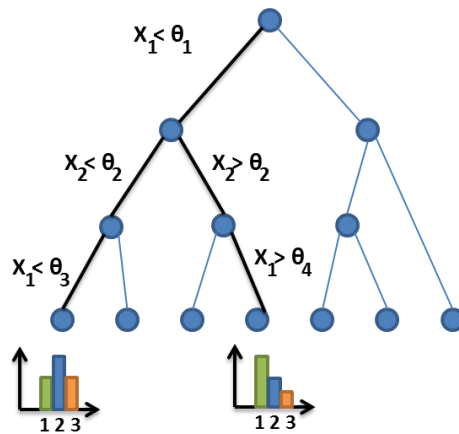


Figure 2.1: Decision Tree Example: At each node a question is asked and the samples go either left or right until a leaf node is reached. The leaves store class distribution.

Formally, we define an input data sample as a multi-dimensional vector  $\mathbf{v} = (x_1, \dots, x_d) \in \mathbb{R}^d$ . At each internal node, a decision function, parametrized by  $\theta_i$ ,

is applied on the features vector  $\mathbf{v}$  and based on the outcome of the decision function which is given by:

$$h(\mathbf{v}, \theta_i) = [\mathbf{v} \cdot \varphi > \tau] \in \{0, 1\}, \quad (2.1)$$

the input sample chooses which direction to follow towards the terminal node. This implies that  $\dim(\mathbf{v}) = \dim(\varphi)$  and  $\varphi$  specifies the shape of the splitting function. This function can be a hyperplane which functions as a threshold to separate the samples based on the entire features. In other cases, the splitting considers only a subset of features especially when the input features vector is sparse. This thresholding decision divides the input space  $S$  at each internal node into two disjoint output spaces  $S^L$  and  $S^R$ .

At testing time, given an unknown data sample described by its feature vector  $\mathbf{v}$ , the classification of the data sample starts by using the decision function at the root node to direct the sample down. After arriving a terminal node by following a certain path depending on the outcome of different decision functions, the class posterior is calculated depending on the samples distribution stored at the arrived leaf node.

At the training stage, a set of input data samples with their known outcome (i.e. class labels in classification forest) is presented to the tree. This set is called the training set. At this stage, the nodes receives this set needs to learn the parameter  $\theta_i$ . For this end, A set of decision functions  $\theta_j$  is generated, either following predefined rules, or by random assignment of  $\theta_j$  from a range of possible values. Each decision function  $\theta_j$  is evaluated using objective function to quantify how good this function is to split the data samples. The best decision function is the one which maximizes the objective function for this set of samples. This decision function is stored at the internal node to be used later for testing new samples. As denoted in [32], the most common form of the objective function employs the concept of entropy and information gain. The entropy of a discrete variable  $X$  with  $b$  outcomes is given by:

$$H(X) = - \sum_{i=1}^n p(x_i) \cdot \log_b p(x_i). \quad (2.2)$$

The information gain of splitting the data set  $S$  arriving the parent node using the split  $\theta_i$  is calculated based on the entropy equation and given by:

$$IG(S, \theta_i) = H(S) - \sum_{s \in \{S_L, S_R\}} \frac{|s|}{|S|} H(s), \quad (2.3)$$

where  $S_L$  and  $S_R$  are the two subsets formed by using the split  $\theta_i$ , and each subset goes to one subtree of the parent node. For regression forest, the gain is calculated in different way based on the variability of the outcome which will be explained with more details in Chapter 7. Decision trees use the information gain to find the best split at each internal node. Once the parameters of best splits are computed and stored, an unknown input sample can be classified following equation 2.1. These learned parameters are the optimal within the set of randomly drawn decision functions. However, it doesn't mean that it will give the optimal classification results for the whole dataset. Moreover, if the data

used for training is not linearly separable, the learned parameters might not be the optimal. Therefore, to improve the learning to find the optimal parameters, more complex decisions functions are used. These complex decision functions could use arbitrary lines in 2D for linear data separation, or conic sections [32] for non-linear data separation. However, a single classifier might not be robust or even sufficient for classification process. Collecting classification results from different classifiers learned from other randomly drawn subsets of data gives more confidence and robustness. Therefore, the concept of the random forests is presented in the following section.

### 2.1.3 Random Forests Concept

A random forest is an ensemble of  $T$  independent decision trees. It has been applied in many applications like face recognition [117] and achieved large success. In random forest ensembles, randomness is introduced during the training of each tree (weak classifier). Next, two popular ways of injecting randomness into trees are presented [32]:

1. Randomized node optimization: The parameters of the decision function, as mentioned before, can be drawn randomly from a range of possible thresholds. In this case, each tree node optimizes the parameters by testing a subset of the entire range of thresholds. The amount of randomness for all nodes in the tree can be controlled using a hyper-parameter, which should be the same for all trees in the forest.
2. Randomized training set sampling: In this way, randomness is introduced to the training data set instead of the decision function. Therefore, each tree can build a weak model by training only from a subset of the entire training set. Each subset is drawn randomly from the entire dataset. Bagging is one possible approach following this way and achieved high training efficiency [32].

Random forest uses several weak classifiers trained using any of the randomness ways so that the overall output of classification or regression process can be defined using all weak classifiers jointly. The output can be computed either by averaging the individual tree posteriors:

$$p(c|\mathbf{v}) = \frac{1}{T} \sum_{t=1}^T p_t(c|\mathbf{v}), \quad (2.4)$$

or by multiplying the trees outputs:

$$p(c|\mathbf{v}) = \frac{1}{Z} \prod_{t=1}^T p_t(c|\mathbf{v}), \quad (2.5)$$

where  $Z$  is a normalization factor, and  $T$  is the number of trees in the Random Forest [32].

Trees depth and the number of trees in a Random Forest are the most influencing hyper-parameters on the prediction accuracy. A higher number of trees leads to better prediction accuracy. However, it increases the computations cost [20]. Therefore, selecting the appropriate number of trees should compromise between the accuracy and speed requirements. Tree depth is an important hyper-parameter for generalization and to avoid problems like over-fitting and under-fitting. This parameter is highly connected with the minimum number of samples required for splitting. A higher value of the minimum number of samples leads to smaller trees. Therefore, this would reduce the prediction accuracy and at the same time reduce the risk of over-fitting [20]. Random Forest has another hyper-parameter which is the sufficient gain to stop splitting of the samples. This parameter ensures sufficient homogeneity level in the class labels arriving that node which helps to avoid over-fitting problem.

In this thesis, Random Forests are used as a classification model for medical instrument part-based detectors. Moreover, it is used to regress the instrument joints in the pose estimation problem. In both cases, hand-crafted features are used as the input feature vector. In the next section, we present an automatic feature extraction tool which called Convolutional Neural Network (CNN).

## 2.2 Convolutional Neural Networks

Convolutional Neural Network (CNN) is the most popular form of deep learning. It provides levels of abstraction and representation of data in deep architectures. In medical microsurgery, this abstractions can be understood as categories (i.e. "Instrument tip", "Open state", "Peeling operation", ...etc) or as features which represent the mapping of the input data as shown in Figure 2.2.

Here, the input could be an image or a patch of the image where the task of CNN network is to map this form of input to the features representation. The mapping starts at low level representation, which transform the input image to a feature vector representing the edges and corners amount in small patches in the image as shown in Figure 2.2. On top of this representation, higher level representation of the features is implemented where it transform the edges and corners information into more complicated structure-based representation of the image. At the highest level, objects and activities are identified. The main advantage of CNN and deep learning in general is the automatic discovery of abstractions from low level features to high level representations without the need of manual feature engineering [16].

### 2.2.1 Basic Concepts

CNNs follow the concept of artificial neural networks which are inspired by biology and mimic the human brain functionality [74]. As the human brains are made up of neurons connected with each other to do some tasks, neuron is the basic component or building block of the neural network. The structure of a

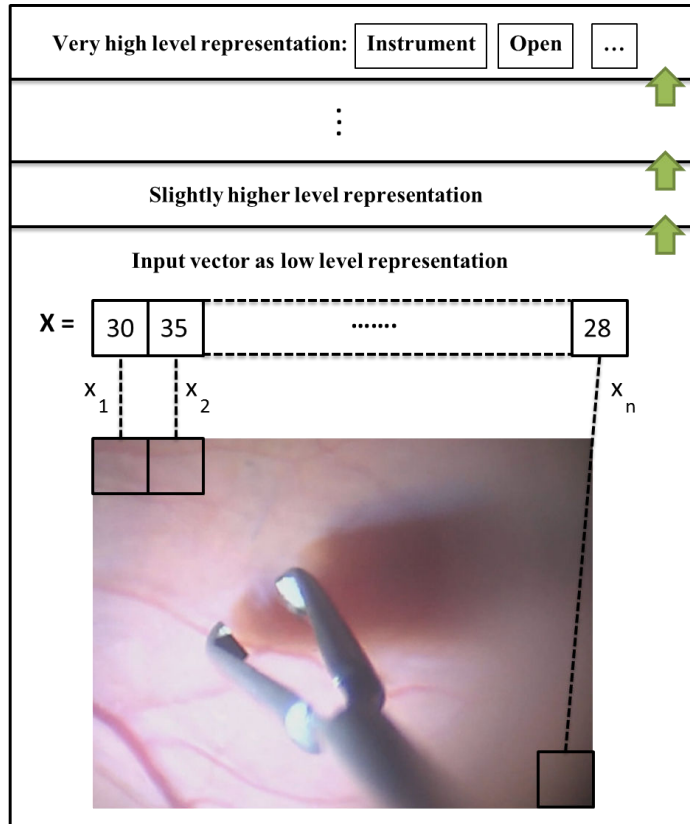


Figure 2.2: Instrument Example illustrating the concept of deep learning architectures.

single neuron is shown in Figure 2.3 where it has the input  $x_i$  and computes the output  $z$  as follows:

$$z = f\left(\sum_{i=1}^3 w_i x_i + b\right), \quad (2.6)$$

where the parameters  $w_i$  are the weights,  $b$  is the bias and  $f(\cdot)$  is a nonlinear activation function [21]. The output  $z$  of the neuron is also called activation. Therefore, every input  $x_i$  is weighted, afterwards, the weighted inputs are summed up together with the bias. The function  $f(\cdot)$  is then applied to the accumulated value to get the final output  $z$ . The importance of using nonlinear function is to be able to find nonlinear mappings between input features vector and the desired output. The most common choices of nonlinear activation functions are the logistic sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.7)$$

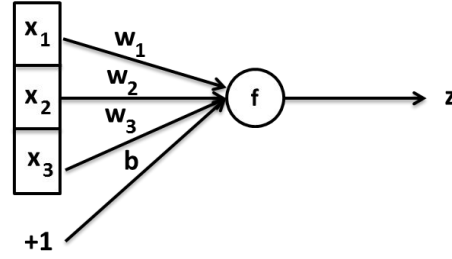


Figure 2.3: Illustration of a single neuron.

or the tangent function:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.8)$$

A neural network consists of a combination of neurons, where the output of one neuron serves as input to others in the next layer as shown in Figure 2.4. This model is called multi-layer perceptron or feed-forward neural network, in which the neurons are arranged into layers. In this model, the neurons of each layer are fully connected with the neurons in the next layer without forming any cycles or loops. The bias units are labeled with " + 1" and parametrized with  $b_i$  which add small shift values to the input weighted combinations and it is not part of the network inputs. Only  $x_i$ 's are considered the input layer,  $z$  is the output and all the layers in between are called hidden layers. The output  $z$  is computed depending on the values of the  $x_i$  inputs, bias values, and the network parameters  $w_i$ . For example, in the perceptron model in Figure 2.4, the output  $z$  is given by:

$$z = f\left(\sum_{i=1}^3 w_i^{(2)} a_i^{(2)} + b^{(2)}\right), \quad (2.9)$$

where  $a_i^{(2)}$  are the activations of the hidden units in the second (hidden) layer and  $b^{(2)}$  is the bias at this layer. Each  $a_i^{(2)}$  value is computed based on the bias and inputs from the previous layer. To clarify this, we show how to compute each of  $a_i^{(2)}$  values. For example  $a_1^{(2)}$  is given by:

$$a_1^{(2)} = f\left(\sum_{i=1}^3 w_{1i}^{(1)} x_i^{(1)} + b_1^{(1)}\right), \quad (2.10)$$

$a_2^{(2)}$  and  $a_3^{(2)}$  can be computed in the same way. Substituting the values of  $a_i^{(2)}$  into equation 2.9 gives the final output  $z$  in terms of the input  $x_i$  and the network parameters  $(W, b)$ . In this example,  $W^{(1)} \in \mathbb{R}^{3 \times 3}$ ,  $W^{(2)} \in \mathbb{R}^{1 \times 3}$ ,  $b^{(1)} \in \mathbb{R}^3$  and  $b^{(2)} \in \mathbb{R}^1$ .



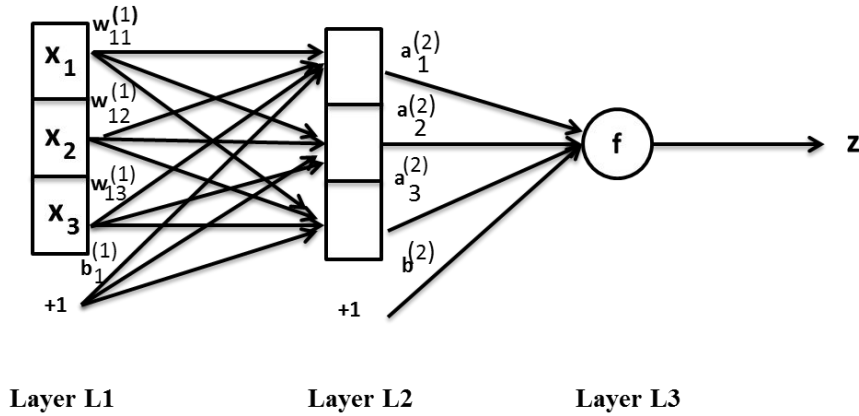


Figure 2.4: Example of multi-layer perceptron.

The process of computing the output of this network is called forward propagation[58] due to the way that inputs are forwarded from one layer to the next one through the network until the output is calculated. The example of the multilayer perceptron in this section can be generalized to any number of hidden layers between the input and output layers. Therefore, deep architectures are characterized by adding multiple hidden layers, and at each layer different number of neurons can be chosen. The deeper the network is, the higher the number of network parameters would be used. Those parameters, in most cases, are randomly initialized and need to be updated in each iteration during network training. Gradient descent and error backpropagation is the most common algorithm to learn the parameters in neural networks.

## 2.2.2 Gradient Descent and Error Backpropagation

The main issue in neural networks is the number of parameters  $\theta = (W, b)$  and how to define suitable parameters values for a specific problem [21]. One approach to learn the network parameters is called gradient descent and error propagation. The concept of this approach is to compute the error between the desired network output and the actual output of the model. This error is propagated back through the network to update the weights according to the gradient descent algorithm.

Regarding the network error computation, an appropriate error function is designed to measure the difference between the desired and actual outputs, where the objective of the training is to minimize the defined error function. Let us denote this error function by  $E(\theta)$ . One option to find the best parameters is to differentiate the error function and solve for the equation  $\nabla E(\theta) = 0$ . However, this option doesn't work when the error function has many local minima which makes it a non-convex problem [21]. Therefore, iterative numerical methods can be the alternative option to find a good solution. In neural network, initial

parameter vector  $\theta^{(0)}$  is chosen, and after each step this vector is updated by:

$$\theta^{(t+1)} = \theta^{(t)} + \Delta\theta^{(t)}, \quad (2.11)$$

where  $t$  is the iteration step [58] and  $\Delta\theta^{(t)}$  is the update value. Among the different approaches proposed to find this update value, gradient descent computes it to be a small step in the direction of the negative gradient and is given by:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla E(\theta^{(t)}) \quad (2.12)$$

where  $\alpha > 0$  is called the learning rate and it controls the step size towards the minimum. After a number of iterations, the parameters  $\theta$  are enough updated so a minimum (possible solution) is reached. The concept of the gradient descent algorithm is illustrated in Figure 2.5. The value of the derivative  $\nabla E(\theta^{(t)})$  at both points  $A$  and  $B$  in Figure 2.5 leads to move  $\theta$  value in the direction of the greatest decrease of the error function. Since the derivative at point  $A$  is negative, this would increase  $\theta$  value, hence, it would move to the minimum peak. On the other hand, the positive derivative at point  $B$  would decrease  $\theta$  value and force it to move towards the minimum peak too. The step size that  $\theta$  follows towards the minimum peak depends on the learning rate  $\alpha$ . A small value of  $\alpha$  leads to a good convergence of the algorithm but with higher number of iterations which makes the convergence slow. On the other hand, with too large value, the algorithm might diverge and never reach the minimum peak [65, 17]. In a simple gradient descent algorithm, the entire training dataset is used in order to compute the error function  $E(\theta)$ . Therefore, the computation of  $E(\theta)$  is needed for each step to compute the new values of the parameter vector  $\theta^{(t+1)}$ , which is computationally expensive process. In contrast, single training sample, instead of the entire dataset, is used to evaluate  $E(\theta)$  using stochastic gradient descent algorithm. This sample is chosen either by random selection or by cycling through the dataset.

**Backpropagation:** The backpropagation algorithm computes the gradients of the error function  $E(\theta)$  with respect to all network parameters. The gradients are used by the gradient descent to update the parameters values. Assuming that we have the training dataset samples  $X = \{x_1, x_2, \dots, x_n\}$ , along with their corresponding output labels vector  $Y = \{y_1, y_2, \dots, y_n\}$ , the sum of squares error function can be defined as:

$$E(\theta) = \frac{1}{2} \sum_{i=1}^n \|y_i - z_{\theta}(x_i)\|^2 \quad (2.13)$$

where  $z_{\theta}(x_i)$  is called the actual output of the feedforward network, and  $y_i$  is the desired output. The idea of the backpropagation algorithm is to compute an error term for each neuron to express its responsibility of any error in the output. This error is easy to compute for the output layer by measuring the difference between the actual and desired outputs as given in equation 2.13. The

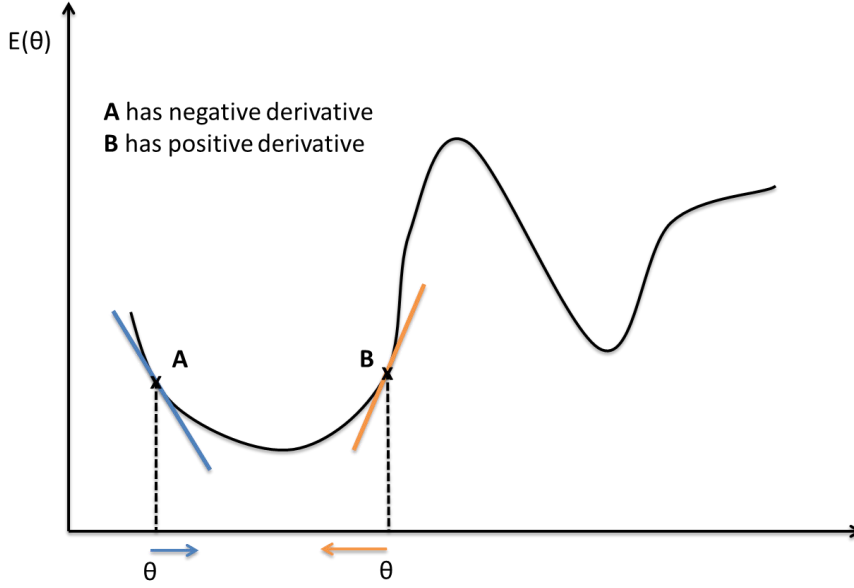


Figure 2.5: Example explaining the concept of gradient descent.

error is propagated one layer back to compute the error at that layer based on backpropagation error formula [21]:

$$\sigma_j^{(l)} = \hat{f}(z_j^{(l)}) \sum_k w_{kj}^{(l+1)} \sigma_k^{(l+1)} \quad (2.14)$$

where  $\sigma_j^{(l)}$  is the  $j^{\text{th}}$  error at layer  $l$  and  $\hat{f}(\cdot)$  is the inverse of the function  $f(\cdot)$  at the same layer. The error is back-propagated until it reaches the first layer. Therefore, the derivative of the error function with respect to the network parameters for the network in Figure 2.4 can be given by:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial z} \frac{\partial z}{\partial a^i} \frac{\partial a^i}{\partial w_{ij}} \quad (2.15)$$

where the derivatives  $\frac{\partial z}{\partial a^i}$  and  $\frac{\partial a^i}{\partial w_{ij}}$  can be directly computed by referring to Eq. 2.9 and Eq. 2.10. After computing the derivative of the error function with respect to each network parameter at the current iteration, the parameters are updated for the next iteration by:

$$w_{ij} = w_{ij} - \alpha \frac{\partial E}{\partial w_{ij}} \quad (2.16)$$

The bias vector  $b^{(i)}$  is updated in the same way, and this update process is repeated for a number of iterations until the best parameters which minimize the error function are obtained.

### 2.2.3 Convolutional Neural Network Concept

A Convolutional Neural Network (CNN) consists of at least one convolutional layer followed by the standard multilayer perceptron network. A CNN is designed as an extension of the neural networks in terms of the types of layers being used in the network. However, it follows the same working principles to find the optimal network parameters to minimize the error function. The architecture of the CNN is designed to take the advantage of working on 2D input data. This is achieved by using small-size kernels applied to the 2D images to produce different representation of the input. The output of applying these kernels can be followed by a form of pooling operators which results in translational invariant features. Applying a series of convolutional and pooling layers produces a new nonlinear representation with smaller size than the input. This representation can be fed into a normal neural network and trained with less number of parameters, which is a great advantage of using CNN. The reason behind this reduction of the number of parameters is due to processing the input with small kernels instead of applying fully connected layers directly.

The input to a CNN network is a  $r \times c \times h$  image where  $r \times c$  are the width and height of the image and  $h$  is the number of image channels, e.g. an RGB image has  $h = 3$  and a gray image, as shown in Figure 2.6, has  $h = 2$ . Assuming that the first convolutional layer has  $k$  filters (or kernels) of size  $n \times n$ , the output would be  $k$  feature maps, each is obtained by convolving the input image with one filter. The size of each feature map is  $(r-n+1) \times (c-n+1)$ . The parameters to be optimized at each convolutional layer are the weights  $w_1, w_2, \dots, w_{n \times n}$  of the  $k$  feature maps. The output of the first convolutional layer is subsampled with mean or max pooling layer. The pooling is performed over  $p \times p$  contiguous regions in the output feature maps where only the maximum or the average of each region is kept in the output images. The pooling layer doesn't influence the number of feature maps but it reduces their sizes by  $\frac{1}{p}$  times in each dimension if the stride step equals  $p$ . The aim of the pooling layer is to make the features invariant to small translations and to reduce the image size. The reduction in the image size would reduce significantly the amount of processing computations needed at the next layer while preserving the important features during subsampling. Next to the first convolution and pooling layer, a number of layers can still be added depending on the problem at hand. These layers also can be a combination of convolution, pooling, normalization or rectified nonlinear units.

Generally, the more convolutional layers we add to the network, the higher computation time is needed to process the image and the more parameters are required. Once the feature maps sizes get small enough after a certain number of convolution and pooling layers, those maps are processed by fully connected layers as shown in Figure 2.6. Unlike the convolutional layers, fully connected layers have higher number of parameters but need less computation.

### 2.2.4 CNN Layers Types

CNNs support many layers types implementations which form the basic blocks of the network architecture:

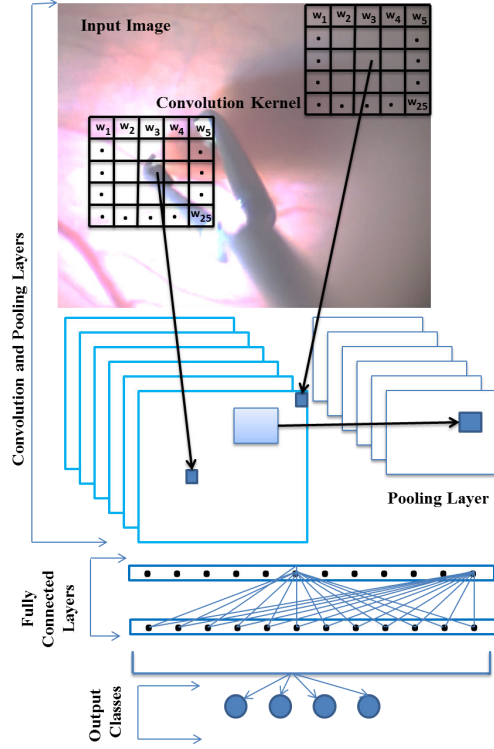


Figure 2.6: A Convolutional Neural Network with one convolution layer, one pooling layer, two fully connected layers, and output layer of four classes.

### Convolutional Layer:

This is the main layer of CNNs and computes the convolution of the input image  $x \in \mathbb{R}^{H \times W \times D}$  with a bank of  $K$  filters  $f \in \mathbb{R}^{H' \times W' \times D \times K}$  to produce output  $y \in \mathbb{R}^{H'' \times W'' \times K}$  as follows [116]:

$$y_{i,j,k} = b_{i,j,k} + \sum_{h=1}^{H'} \sum_{w=1}^{W'} \sum_{d=1}^D f_{h,w,d,k} \cdot x_{i+h,j+w,d} \quad (2.17)$$

where  $b_{i,j,k}$  is the bias. The values of the  $K$  filters are initialized randomly or based on a certain distribution. The size of the output image depends on the padding way and stride step of the convolution operation.

**Spatial Pooling Layer :**

It computes the average or the maximum response of a group of contiguous features in a feature map  $x$  within a patch  $P$  as follows [116]:

$$y_{i,j,d} = \frac{1}{H'W'} \sum_{1 \leq h \leq H', 1 \leq w \leq W'} x_{i+h,j+w,d}, \quad (2.18)$$

or

$$y_{i,j,d} = \max_{1 \leq h \leq H', 1 \leq w \leq W'} x_{i+h,j+w,d}, \quad (2.19)$$

where  $P$  has the size of  $H' \times W'$ . Pooling aims at preserving the dominant feature within a set of contiguous features. This has the advantage of making the features invariant to small translation variations. Moreover, depending on the chosen pooling stride, the size of resultant feature maps is much smaller than the size of the input maps. Therefore, less processing computations are needed at next layers.

**Local Response Normalization :**

It performs cross-sectional normalization where it is applied at each spatial location and to groups of feature maps as follows [116]:

$$y_{i,j,d} = x_{i,j,d} \left( k + \alpha \sum_{u \in G(u)} x_{i,j,u}^2 \right)^{-\beta}, \quad (2.20)$$

where  $k, \alpha, \beta$  are hyper-parameters and  $G(u) \subset \{1, 2, \dots, D\}$  is a corresponding subset of input feature maps. This operator produces an output image with the same size as the input map.

**Batch Normalization:**

Batch normalization works in different way from other CNN blocks. It performs normalization across images or feature maps in a batch. Let us assume the batch has  $T$  images or feature maps, then  $x, y \in \mathbb{R}^{H \times W \times K \times T}$ ,  $w \in \mathbb{R}^K$ ,  $b \in \mathbb{R}^K$  and the output feature map is computed as [116]:

$$y_{i,j,k,t} = w_k \frac{x_{i,j,k,t} - \mu_k}{\sqrt{\sigma_k^2 + \epsilon}} + b_k, \quad (2.21)$$

where  $\mu_k$  and  $\sigma_k^2$  are the mean and variance across the maps in a batch and computed as:

$$\mu_k = \frac{1}{HWT} \sum_{i=1}^H \sum_{j=1}^W \sum_{t=1}^T x_{i,j,k,t}, \quad (2.22)$$

$$\sigma_k^2 = \frac{1}{HWT} \sum_{i=1}^H \sum_{j=1}^W \sum_{t=1}^T (x_{i,j,k,t} - \mu_k)^2, \quad (2.23)$$

**Rectified Linear Unit (ReLU):**

This activation function has been proven to be more robust than sigmoid and hyperbolic tangent functions [46, 77] and it is defined as:

$$y_{i,j,k} = \max(0, x_{i,j,k}), \quad (2.24)$$

where  $x$  is the input to the neuron at location  $(i, j)$  of feature map  $k$  and  $y$  is the corresponding output. This unit is used to avoid gradient vanishing problem [52]. This problem is defined as the difficulty found in training and parameter tuning of neural network using gradient-descent methods and backpropagation. In this problem, the parameters of the network are updated based on their contribution in the output error. With normal sigmoid and hyperbolic tangent functions, the output at each layer is limited into a small output range (i.e.  $[0, 1]$  for sigmoid and  $[-1, 1]$  for hyperbolic tangent). Therefore, large changes in input will be mapped to a small range in the output. This problem becomes worse by adding more layers on top of each other. As a result, the gradient will be very small at the output layer regardless of the amount of input changes especially at early layers. Therefore, the network training will be very slow in updating the parameters at early layers.

This problem can be avoided by using activation function which doesn't limit the output to a small range of values. Since ReLU activation function has this property, it is considered a good solution for gradient vanishing problem as well as it shows more practicality than other activation functions in this orientation.

**Dropout:**

Dropout is a technique to address the problem of overfitting in deep neural networks. Overfitting can happen as a result of limited training data or high connectivity in the fully connected layers [100]. In fully connected layers, the output of each neuron in one layer is connected as input to each neuron in the next layer. In this case, training of the network would be slow and some dominant neurons might have influence on other neurons which leads to overfitting problem. Dropout is introduced to reduce the influence of a single node by switching off one or more neural network nodes during training. In this case, the learned weights of the nodes become in-sensitive to the weights of the other nodes. Therefore, dropout helps the network to generalize better. It switches the nodes off by removing it temporarily from the network, together with all of its input and output connections. Simply, each neuron is kept in the network with a probability  $p$  and removed with a probability  $1 - p$ . The process of switching a neuron on/off is independent of other neurons. Therefore, when

the dropout is applied, the activity in Eq.2.10 becomes:

$$a_1^{(2)} = f\left(\sum_{i=1}^3 w_{1i}^{(1)} p_{1i}^{(1)} x_i^{(1)} + b_1^{(1)}\right), \quad (2.25)$$

where  $p_{1i}^{(1)}$  are the retention probabilities of neurons at the first layer, which are regarded as gating 0 – 1 Bernoulli variable [10]. At test time, there is no dropout and all the neurons are involved in decision making. However, Eq.2.25 incorporates a retention probability  $p$  for each neuron. This probability is used to scale-down the outgoing weights of that neuron proportionally to the retention period.

Dropout forms a regularization term which can significantly increase generalization possibilities for a wide variety of classification problems compared with other regularization methods. Moreover, it can be generally applied to other graphical models such as Boltzmann Machines [95].

These blocks of deep learning network make it a powerful training tool with high capabilities to handle overfitting, gradient vanishing and other problems. Furthermore, CNNs have the ability to model wide range of variations in scale, rotations and light changes in data. A CNN network has been designed in this thesis to detect different parts of an articulated medical instrument. However, in most cases, we need post-processing techniques to cope with false detections. Therefore, relying on geometrical constraints would further improve the performance. This leads us to introduce Conditional Random Fields (CRFs) as powerful graphical models to apply such these constraints.

## 2.3 Conditional Random Fields

An articulated object can be defined as a set of  $N$  parts, where each part corresponds to a random variable  $y_i$ . These parts are related to each other based on the structure of the object. Hence, the purpose of the object pose estimation is to employ the relations between different parts to predict the vector  $y = (y_1, y_2, \dots, y_N)$  of random variables from the observations  $x$ . The observed random variable  $x_i$  can be considered as the received signals or extracted features. In the context of medical imaging applications, in particular the medical instrument pose estimation, those observations are extracted as features from different parts forming the medical instrument. The dependencies between these parts (random variables) are encoded using probabilistic graphical model (PGM). More specific to the task of pose estimation of the medical instruments, the goal of the PGM is to infer a particular pose  $y$  among the hypothetical poses in a huge search space. Each pose  $y$  is represented by a set of 2D image coordinates.

There are many types of graphical models [57] used to represent the relations between dependent and/or independent random variables. One class of these models, which has been applied in pattern recognition and machine learning applications, is called Conditional Random Field (CRF). Given an observation  $x$



of the random variables  $y$ , CRFs model the conditional distribution  $p(y|x)$ . For many applications, modelling using CRF has many helpful properties [14]:

1. Modelling using CRF is a problem-specific task, which means the definition of dependencies between the random variables and observations are varying according to the problem.
2. Unlike generative models, CRFs, which are discriminative models, don't require the estimation of the joint probability distribution  $p(y, x)$  which is a difficult task.
3. The model doesn't rely on prior model  $p(x)$  of the observations, where the dependencies between features make it also a difficult task to build such a prior model.

Therefore, a CRF offers a computational efficient solution among the graphical models, in addition to its simple structure. The main three parts of designing and using CRFs are modelling, inference, and parameters estimation.

### 2.3.1 Graphical Modelling

A graphical model is a way to express conditional or joint multivariate probability distributions. It brings together probability and graph theories in a powerful framework for statistical modeling. This is why it is also called probabilistic graphical models *PGM*. It expresses the conditional dependencies between random variables and the relations between these random variables and the observations. Based on the type of these relations, the type of the graphical model is defined which can be one of many types such as Bayesian networks (directed graphical model) or Markov Random Fields (undirected graphical model). The type of the graphical model defines the distributions by means of a graph  $G = (V, E)$ , where  $V$  is the set of vertices which correspond to random variables and  $E$  is the set of edges denoting dependencies between the variables as shown in Figure. 2.7.

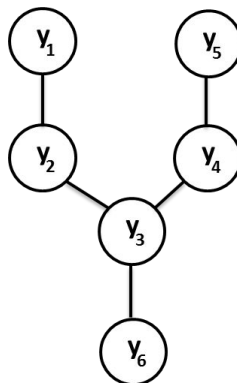


Figure 2.7: Graphical Model: An undirected graphical model that would correspond to the medical instrument joints.

Following the notation in [102], we define the probability distribution  $p$  over the random variables  $V = X \cup Y$ , where  $X$  is the input random variables observations, and  $Y$  is the output random variables. In our work, the domain of each random variable would be the 2D pixel coordinates within the image plane. The image space forms a discrete output domain  $\Lambda$  that each output variable's state is derived from. Let us assume an assignment  $x_i$  of the observations  $X$  is to assign a value to the  $i^{\text{th}}$  variable of  $X$ . Furthermore,  $x_a$  can denote an assignment to a subset of the observation  $X$ , where  $a \subset X$ . The probability distributions can be expressed as a product of factors which define the interactions between random variables. Assuming the factor is represented by  $\Psi_a(x_a, y_a)$ , then the distributions  $p(x, y)$  in an undirected graphical model or CRF can be defined in terms of these factors  $F = \{\Psi_a(x_a, y_a)\}$ , where  $\Psi_a : \Lambda^{|a|} \rightarrow \mathbb{R}^+$ , and can be written as [102]:

$$p(x, y) = \frac{1}{Z} \prod_a \Psi_a(x_a, y_a), \quad (2.26)$$

The constant  $Z$  is called the normalization factor or the partition function and defined as:

$$Z = \sum_{x, y} \prod_{a \in F} \Psi_a(x_a, y_a) \quad (2.27)$$

The computations of  $Z$  is in general intractable [57] and used to transform all the real values of the factors into probabilities and many algorithms have been proposed for approximating it. Each potential function is assumed to have the form:

$$\Psi_a(x_a, y_a) = \exp \left\{ \sum_k \lambda_{a,k} f_{a,k}(x_a, y_a) \right\}, \quad (2.28)$$

for some real-valued parameters  $\lambda_{a,k}$ , and a set of feature functions  $f_{a,k}(x_a, y_a)$ . This form defines a specific distribution over  $V$  from the exponential family which parameterized by  $\lambda_{a,k}$ . The feature function  $f_{a,k}(x_a, y_a)$  models the dependencies between random variables and observations and it can be defined in different ways including indicator functions. Moreover, the factorization of the distribution in 2.26 can be expressed in a different form called the factor graph.

A factor graph is a way to represent the factorization of the probability distributions  $p$  by a mean of graph form. The graph here has the triplet  $(V, E, F)$  representation as shown in Figure 2.8, where the variables describing the factor  $f_{a,k}$  are included in the graph. In this type of graphs, the random variables are represented by circle nodes and the factors by box nodes. The factors or potential functions show the dependencies between random variables, while the edges show the variables involved in these dependencies or relations. Each potential function can operate on any number of random variables. The number of variables connected at the factor box defines the type of the potential function, which can be unary, binary or higher order.

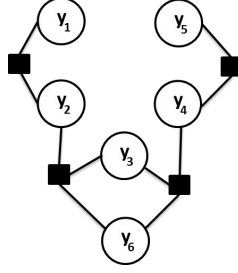


Figure 2.8: Factor Graph: This graph is a representation of the undirected graph model shown in Figure 2.7. The circles are random variables and the black boxes are potential functions or factors representing the dependencies between variables.

A conditional probability distribution  $p(y|x)$  for a factor graph  $G$  over random variables  $Y$  and observations  $X$  is considered a **Conditional Random Field (CRF)** iff for any assignment  $x$ , the distribution  $p(y|x)$  satisfies the Markov property with respect to the graph  $G$ :  $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$ , where  $w \sim v$  means that  $w$  and  $v$  are neighbors in  $G$  [61].

The defined conditional distribution by a set of factors  $F = \{\Psi_a\}$  that belong to  $G$  can be written as :

$$p(y|x) = \frac{1}{Z} \prod_{\Psi_a \in G} \exp \left\{ \sum_{k=1}^K \lambda_{a,k} f_{a,k}(x_a, y_a) \right\} \quad (2.29)$$

where  $K$  is the number of feature functions for each factor  $\{\Psi_a\}$ . These factors are partitioned to cliques in order to estimate the parameters of each factor. A clique in the graph  $G$  is a set of vertices connected with each other such that they form a complete subgraph in  $G$  [22]. Using the maximal clique rule, the factor graph  $G$  is partitioned into  $C = \{C_1, \dots, C_Q\}$ . Each clique  $C_q$  corresponds to set of factors having some feature functions  $\{f_{q,k}(x_t, y_q)\}$  parameterized by  $\lambda_{q,k} \in \mathbb{R}^{K(q)}$ , where  $K(q)$  is the number of feature functions in a clique  $q$ . Based on clique factorization, the conditional random field distribution can be written as:

$$p(y|x) = \frac{1}{Z(x)} \prod_{C_q \in C} \prod_{\Psi_c \in C_q} \Psi_c(x_c, y_c; \lambda_q) \quad (2.30)$$

where the normalization function  $Z(x)$  is defined as:

$$Z(x) = \sum_y \prod_{C_q \in C} \prod_{\Psi_c \in C_q} \Psi_c(x_c, y_c; \lambda_q). \quad (2.31)$$

Finally, each factor  $\Psi_c$  can be written as:

$$\Psi_c(x_c, y_c; \lambda_q) = \exp \left\{ \sum_{k=1}^{K(q)} \lambda_{q,k} f_{q,k}(x_c, y_c) \right\}. \quad (2.32)$$

Based on Eq.2.30, a conditional random field (CRF) is described by a set of potential functions along with their parameters  $\lambda = \{\lambda_{q,k}\}$ , where  $\lambda \in \mathbb{R}^D$ . In the context of medical applications, instrument pose estimation is specified by means of potential functions where the parameters  $\lambda$  are assumed to be estimated before starting the inference step of the final predictions of the instrument joints coordinates in the image space. Inference in CRFs is also related to energy minimization problems. However, predicting the instrument joints coordinates for the random variables in CRFs requires finding the maximum posterior probability which is equivalent to energy minimization [78].

### 2.3.2 CRF's Potential Functions

The potential functions as shown in Figure 2.8 are designed to model the dependencies between random variables  $Y$  and observations  $X$ . In other words, they model the relations between hidden and output variables which could differ from one application to another. Within the context of medical instrument pose estimation, many functions have been designed to model unary, binary, ternary and quaternary factors. The unary potential functions generally used to model the relation between input observations  $X$  and output variables  $Y$  (i.e. the relation between the appearance of image patches and the existence of any instrument part). In this thesis, we denote the unary functions by  $\phi_i(y_i, x)$ . Binary potential functions are designed in this work to model the relations between output variables  $Y$ . Some of these functions are used to model the temporal information by means of parts movements between frames. Others are used to model connectivity along gripper parts of the instrument to give preference for some hypotheses in order to localize the tips more precisely. We will denote the binary functions by  $\psi_{i,j}(y_i, y_j)$ . Ternary potential functions, which is denoted by  $\psi_{i,j,k}(y_i, y_j, y_k)$  in this thesis, have been modelled to impose more constraints on the output random variables  $Y$  to increase the robustness of the proposed model in cases of highly cluttered background. Finally, we modeled also a quaternary potential function denoted by  $\psi_{i,j,k,l}(y_i, y_j, y_k, y_l)$ , and as an example for this potential type is the modelling of the instrument parts rotations distribution. Pictorial structures [40] can be seen as a variant of the potential functions, but still they model the relations between the graph variables by different representation. Graphical models represent the first step in modelling using CRFs, the next steps are inference and parameters estimations which are presented next.

### 2.3.3 Output Variables Inference

The prediction of the output variables  $Y$  of the conditional random field is called inference. Inference is performed on the factor graph using the probabilistic graphical model to predict the output variables  $Y$  based on the observations  $X$ . The factor graph can be employed in two different methods to predict variables:

1. Maximum a posterior (MAP) inference: the goal of this inference is to predict the labels of the output random variables  $Y$  based on the

observations  $X$  and the learnt parameters  $\lambda$ . In this inference, the labels that maximize the posterior probability of Eq. 2.30 are obtained by:

$$\hat{y} = \underset{y}{\operatorname{argmax}} p(y|x, \lambda). \quad (2.33)$$

In medical instrument pose estimation,  $\hat{y}$  is the predicted instrument joints vector, which also represents the most probable configuration of the instrument pose based on MAP inference.

2. Probabilistic inference: the goal of this inference is to estimate the partition function  $Z(x)$  and also the marginal distributions of the factors. Therefore, this inference turns the problem to marginalization instead of maximization. The marginalization in this type of inference is obtained by:

$$\sum_y p(y|x, \lambda), \quad (2.34)$$

and would estimate all the marginals of instrument in terms of the instrument parts.

Many algorithms have been proposed to address the inference problem for general factor graphs. In this case, the problem is known to be NP-hard [98]. As a CRF is based on modelling the problem using factor graph, it can use the inference algorithms that are applied in graphical models in general [57]. However, the complexity of the graph has a high impact on the inference algorithms efficiency and the convergence time to find the optimal solution. Therefore, imposing more constraints on the graph structure would reduce the inference complexity and make the problem tractable. For the exact inference problem, the most commonly used algorithms are the forward-backward algorithm [83] and Viterbi algorithm [42]. However, the exact inference algorithms can be applied only to graphs without loops which have tree structures.

In graphs with loops, exact inference can be inefficient and computationally expensive even though it is possible after transforming the graph into a tree-structure one. Junction tree algorithm [63] can be used for this kind of transformations to enable exact inference for predictions. However, the transformations made by [63] rely on clustering the variables into huge clusters which requires exponential time to infer the solution. To solve the exact inference problem and make it possible for real time applications, the approximate inference has been used. Many algorithms have been proposed in the literature for approximate inference [57]. Most of the algorithms proposed in this regard can be categorized mainly into either Monte Carlo or variational algorithms. Monte Carlo algorithms, such as Markov Chain Monte Carlo (MCMC) methods [57], are computationally expensive and based on sampling to find an approximations of the distribution of interest. Formulating the inference problem as an optimization problem is a part of the variational methods and an approximation of the solution is found by minimizing an energy function. A well known variational algorithm is belief propagation which also can estimate exact solution for tree-structured graphs using max-product algorithm [57]. However, for loopy graphs, an approximate solution is estimated

which is not guaranteed to be the optimal one. For medical instrument pose estimation, due to the use of smaller number of random variables than being used for human pose estimation, we choose evolutionary algorithms [92]. These algorithms combine the sampling and solving optimization problem to find an approximate solution with real time performance. A well known algorithms from the evolutionary algorithms are the **genetic algorithms** [92, 48] which are used in this thesis for instrument pose inference. Genetic algorithms use mechanisms inspired by biological evolution including reproduction, mutation, recombination, and selection. The first step in these algorithms is to encode the chromosomes or individuals. A set of configurations are sampled and encoded as genes of chromosomes to build the initial population. The quality of each individual is computed based on fitness function which assigns to each individual a probability of survival in the next population. This function forms our optimization function which the algorithm is designed to maximize. The optimization of this function requires producing of a new generation and selecting of the best individuals (highest probabilities). After the selection process is done, crossover is applied between pairs of individuals to generate new offspring. This operation swaps some genes between the individuals after selecting crossing points for recombination. The swapping operation can produce some offsprings with higher survival probabilities than their parents. These offsprings are considered better solutions. Therefore, these algorithms run in an iterative way until they produce the best solution or stop after a fixed number of iterations. To avoid trapping at a local maximum, mutation process is applied with certain probability after crossover operator by changing values of some genes. Changes can be done in a random or heuristic way to ensure convergence during optimization. Evolutionary algorithms perform well approximations for all types of problems because no constraints are imposed on the fitness landscape.

The task of inference comes after modelling the problem using factor graph and estimating the parameters  $\lambda$ . Next, parameter learning is presented.

### 2.3.4 Parameter Learning

Parameter Learning in CRF is also called parameter estimation. These parameters,  $\lambda$  in Eq. 2.30, should be selected to maximize the similarity between the conditional probability distribution and the true distribution. To measure the similarity between two distributions, Kullback-Leibler (KL) divergence [60] is used. Following the notation in [102], we denote the training samples  $D = \{x^{(i)}, y^{(i)}\}_{i=1, \dots, N}$ , and they are assumed to be independent and identically distributed (i.i.d). There are many ways to estimate the parameters in CRF model:

**Maximum Likelihood** : Maximum Likelihood can be used to learn CRF parameters from training dataset in a probabilistic manner. The concept of this way is to maximize the training data samples that fit a certain model. The likelihood can be replaced with log-likelihood since the log function is monotonically increasing and can't change the location of the maximum points. The log-likelihood, which is called also the conditional log-likelihood, can be

written as:

$$L(\lambda) = \sum_{i=1}^N \log p(y^{(i)} | x^{(i)}; \lambda). \quad (2.35)$$

By substituting the conditional probability distribution from Eq. 2.29 into Eq. 2.35 and adding the regularization function  $Z(x)$ , we get a differentiable function  $L(\lambda)$ . This function has no closed form solution due to the difficulty in computing  $Z(x)$ , which requires each single training instance. However, it can be optimized in an iterative ways using gradient descent methods or Newton's method [19]. Newton's method has more computational cost due to the computations of the Hessian matrix at each iteration. Less computational methods can be used for Hessian approximations such as BFGS [19] and Limited-memory BFGS (LM-BFGS) [68]. In general, for complex graphs with high orders dependencies, the computations become more expensive, and the likelihood optimization might not be possible. Many other approximations of it have been proposed like surrogate likelihood [103].

**Margin-based estimation :** This is another common way to estimate the parameters  $\lambda$  of CRF. Again, assume we are given a set of training samples  $D = \{x^{(i)}, y^{(i)}\}_{i=1, \dots, N}$  of unknown distribution of CRF. Furthermore, assume a loss function  $\xi$  which computes the difference between the predictions  $\bar{y}$  of the labels, and their corresponding ground-truth  $y$  for a training instance. The parameters  $\lambda$  are learned in a way to minimize the loss function  $\xi(y, \bar{y})$ . The loss function  $\xi$  can be minimized using structural risk minimization [114]. A regularized empirical risk can be written as:

$$R(f) + \frac{C}{S} \sum_{i=1}^N \xi(y^{(i)}, f(x^{(i)})) \quad (2.36)$$

where the first term is the regularization term which helps to avoid overfitting during training, and the second term is the empirical estimation of the expected risk. To estimate the parameters  $\lambda$ , we need to find a good estimator function  $f$  which minimizes the loss function  $\xi$ . Structured support vector machine (S-SVM) training [113] is considered a feasible choice for the solving the minimization in Eq. 2.36. The regularization constant  $C > 0$  is a hyper-parameter.





## Chapter 3

# Related Work

The problem of medical instrument detection and tracking has been addressed in diverse research areas for different types of surgeries. Among these surgeries, retinal microsurgery is still the most delicate and challenging one. The challenges are mainly due to the limited accessibility to the retina tissue which should be accessed using only tiny instrument through microscope lens. Moreover, the small size of the instrument, the eye geometry and the complications in the surgery environment prevent the usage of special sensors for tips localization. Therefore, a heavy burden is put on processing the images obtained from different imaging devices (e.g. optical or OCT devices) to assist surgeons by visual localization of the instrument tips or by augmenting the scene with important information such as the distance of the instrument tips to the retina.

Different approaches have been proposed to tackle the detection and tracking problems. While some approaches formulate the problem as an intensity-based tracking, many other approaches employ dedicated detectors for certain parts of the instrument. Pose estimation techniques can be seen as another task working on top of detection and tracking algorithms. In this chapter, state of the art approaches from each category are presented with their achievements and limitations.

### 3.1 Intensity-Based Instrument Tracking

In this category, the instrument is detected and tracked based on color information [37, 36, 79] of the instrument and the artificial markers [50, 121], or even by tracking intensity values without transforming the image into other color models. Richa et al. [86] proposed to use weighted mutual information with gradient-based tracking [34] to track instrument through a stereo microscope. The joint probability matrix of the mutual information is weighted to make the similarity measure invariant to illumination changes during the search for the instrument tip. In their experiments, a fixed reference image and static background are assumed and only vitrectomy instrument is

used for evaluation. In other tracking algorithms [84], intensity values have been employed to find alignment parameters between target and reference images. After a good match is found, new patches are added to a database to be used for online tracking. The algorithm allows growing the region of interest to discover new regions and to make the tracking more robust. However, the algorithm can only localize a bounding box around the instrument tip in laparoscopic images.

Online learning [67] has achieved state-of-the-art performance on retinal and laparoscopic surgeries. This work builds up a database of positive and negative samples of the target instrument during its movement. The instrument is tracked by a modified work of Lucas Kanade tracker [71]. Moreover, the tracker is enforced by intensity-based cascade detectors for fast recovery after tracking loss. The algorithm has the ability to localize only the center point of forceps instrument. However, the algorithm can't validate the created database and can't guarantee that intensity-based detectors would work properly in case of significance changes of illumination. Moreover, the performance is highly dependent on good initialization after the disappearance of the instrument.

Active Testing [45, 108] has been employed to model the instrument tracking and detection in a unified framework[106]. The basis for this method is to model the instrument localization as a sequential entropy minimization problem to estimate 3DOF parameters required to localize the instrument tip. During the parameters optimization phase, the instrument appearance and intensity values have been highly used to build the trained models. Even though the method shows high accuracy of instrument tip prediction on eye phantoms, no evaluation has been carried out on real in-vivo retinal surgery. Moreover, the huge amount of training required to build different distributions limits the method's applicability and generalization.

Riete et al. [85] proposed a solution to track Large Needle Driver (LND) tool by making use of the landmarks on the tool surface. Color, location and Gradient-based features have been associated with the landmarks for training random ferns. The 3D locations of the instrument are retrieved by matching the features tracks in the stereo camera using normalized cross correlation. The method achieves high localization accuracy for LND tools. However, the tracking can't run at the video frame rate due to the computational cost of extracting all these features. Moreover, the occlusions of some landmarks due to the instrument rotation might result in high localization error. Another approach [3] was proposed for articulated tool tracking in 3D laparoscopic images, in which the color information is used for instrument parts segmentation. The segmented regions are described by different statistical models in order to estimate the pose of the instrument in the 3D space. Optical flow is used for pose tracking from image to another. The approach has also the limitations of expensive feature extraction and its sensitivity to the light changes.

Generally, intensity-based tracking or color-based detectors can be fast and accurate for estimating the instrument pose. However, some changes in the instrument or background appearance have severe impact on the tracking and lead to frequent tracker loss. Therefore, a high burden is put on good re-initialization after tracking failures. Furthermore, combining features from different color models [85, 2, 3] makes the tracker slower than required for real in-vivo surgeries.

## 3.2 Instrument Tracking by Detection

Tracking by detection algorithms have the advantage of being able to handle instrument disappearance since the re-initialization is no longer treated differently. However, the time complexity depends on the detector complications.

Sznitman et al. [105] proposed to integrate gradient-based tracker with machine learning based detector. The detector computes the sums of the oriented edges for positive and negative samples to train the deformation in instrument shape. On the other hand, the tracker is used to favor close detections to the last predicted instrument position. Even though their approach achieved the state-of-the-art accuracy in predicting the forceps connecting point location, it is unable to detect the forceps tips which are of more interest from clinician prospective for minimally invasive surgery.

Further improvement using the same set of features as in [105] has been proposed in [107]. The instrument is modelled as an articulated object where all the instrument parts are located linearly in a row. The algorithm employs gradient boosted regression trees as an iterative optimization way to stop the algorithm early and reduce the computation time. Random Sample Consensus RANSAC [41] algorithm is used to fit a line through these detections and to estimate the instrument tip location. The algorithm shows good performance on vitrectomy instruments in retinal, pelvic and spine surgeries. However, it doesn't have any mechanism to handle forceps instrument to localize its two tips.

Chen et al. [29] uses structural and geometric features as the input vector for spiking neural network [35]. Gabor kernel is used to extract features at specific orientations using Laplacian of Gaussian (LoG) of the input image. The method is used to localize the instrument tip in laparoscopic images and achieved high detection accuracy. However, using the proposed network for instrument detection doesn't satisfy real time requirements.

The main challenging issue for the tracking by detection algorithms is to find the discriminative features which can be extracted at the video frame rate. Moreover, the refinement process on top of detections parts to localize the instrument joints coordinates has to meet the real time requirements as well. Generally, tracking by detection methods are slower than other methods and more challenging exists to make them running at the video frame rate. In this thesis, most of the proposed solutions follow this category since they provide more robust ways to avoid tracking failures in this delicate microsurgery.

## 3.3 Instrument Pose Estimation

The process of pose estimation aims to find 2D or 3D coordinates of instrument joints. It can be considered as a separate problem. However, most of the proposed methods employ it on top of tracking or detection operations.

A database of a 3D CAD model of the forceps was generated in [8], and the likelihood to the projected contour of the microscopic image is extracted to find a match from the database to estimate the forceps pose. The proposed method

employs color information and was evaluated only on synthetic data. Therefore, it is complicated to extract these contours in real in-vivo surgery having cluttered background and severe light changes. Instrument pose in [3] was estimated on top of intensity-based classifier output. This output is a set of segmented regions used to fit distribution models to estimate the 3D pose of the tool.

Rieke et al. [88] proposed to use intensity-based tracker [109] to locate a bounding box around the instrument. The tracker predicts the translational parameters to update the position of the bounding box. Within this box, the instrument joints are predicted using random forest [15]. The approach also achieved high accuracy rate in joint localization, but it cannot recover automatically after the tracker is lost. Moreover, the tracker needs a huge amount of training samples to handle changes in the instrument appearance.

### 3.4 State of the Art Techniques Summary

Table 3.1 shows a summary and comparisons among the state-of-the-art techniques in terms of whether they are able to localize instrument joints (one tip, Left tip, right tip, joint point), run at real time, work without manual initialization, handle forceps instrument and estimate the instrument orientation.

The table shows that the majority of the algorithms can't localize the left and right tips of forceps instrument despite of using forceps instrument for their evaluation. On the other hand, Rieke et al. [88] proposed the only approach used to localize these tips at real time performance, but they are missing the automatic re-initialization module to handle tracking loss. Therefore, in this thesis, the proposed solutions to the pose estimation problem are built on top of special detectors to localize forceps joints in real time while still being able to handle tracking loss problems. The ultimate goal of the proposed methods is to find a feasible solution to work on real in-vivo surgeries. While most of the proposed methods have not been validated on real surgeries, our proposed methods are evaluated on real retinal and laparoscopic surgeries in addition to publicly available datasets for quantitative comparisons with other state-of-the-art methods.

Table 3.1: Methods Comparisons :(T = tip, LT = left tip, RT = right tip, JP= joint point, VT = work on vitrectomy, FC = work on Forceps, I = automatic Initialization, O = estimate orientation, R = real time)

Method	T	LT	RT	JP	I	VT	FC	O	R
Richa et al. [86]	✓				✓	✓			✓
Reiter et al. [84]	✓					✓			✓
Reiter et al. [85]	✓			✓	✓	✓	✓	✓	
Baek et al. [8]	✓				✓		✓		✓
Rieke et al. [88]	✓	✓	✓	✓		✓	✓		✓
Li et al. [67]				✓		✓	✓		✓
Allan et al. [2]	✓				✓	✓	✓	✓	
Allan et al. [3]	✓				✓	✓	✓	✓	
Sznitman et al. [105]				✓	✓	✓	✓		✓
Sznitman et al. [107]	✓			✓	✓	✓	✓	✓	✓
Sznitman et al. [106]	✓				✓	✓		✓	✓
Pezzementi et al. [79]				✓		✓	✓	✓	
Chen et al. [29]	✓				✓	✓			

## CHAPTER 3. RELATED WORK

---

## Chapter 4

# Color Information and Geometric Modelling

### 4.1 Introduction

Instrument Detection and tracking in retinal microscopic surgery is a crucial part for the minimal invasive procedures due to the aforementioned challenges in the introduction chapter. The first approach towards detection and tracking of the instrument is to consider color information to segment it from the other objects in the background. As mentioned earlier in this thesis, artificial markers have been used for detection and tracking instruments in laparoscopic datasets [111] which requires a colored strip to be placed on the instrument shaft. The predefined color information is used to facilitate image segmentation and therefore to localize the instrument tip and the orientation of the shaft. Color information of natural landmarks has also been employed in other work [2] for articulated instrument modelling in laparoscopic surgery. However, using artificial or natural markers on forceps instruments used in retinal membrane peeling is quite difficult due to the tiny size of the instrument (*i.e.* 0.5mm in diameter). Moreover, most of the color based approaches [2, 3] need to extract color-based features from different channels to localize the instrument in 2D images. Therefore, these approaches don't meet real time requirements. Hence, their time complexity prevents their usage for medical applications which require real time processing capabilities.

On the other hand, depending only on the geometry of the instrument [118, 106, 79] definitely requires a prior shape information to be modelled. This information imposes some kind of constraints on the background, instrument entry point, shaft length, type of instrument or light changes. Some of these constraints can not be satisfied at real in-vivo surgery. As a result, such approaches might not be applicable in such surgeries. Moreover, changes in 2D retinal images from one image to another under severe light changes and gripper movements are often difficult to be modelled as a linear transformation. This is why most of the trackers fail in real surgery. Therefore, more complicated or deformation transformations are required to model the changes in the geometry

and appearance of instrument over time.

In this Chapter, both of color information and the geometry of the rigid part of the instrument are used in a new approach. The time complexity required by the color-based methods [2, 3] is avoided by relying only on one color model instead of combining color information from different channels of different color models. Therefore, to compensate for less color information, the geometry of the instrument rigid part is used for the optimization of the tip and orientation estimation.

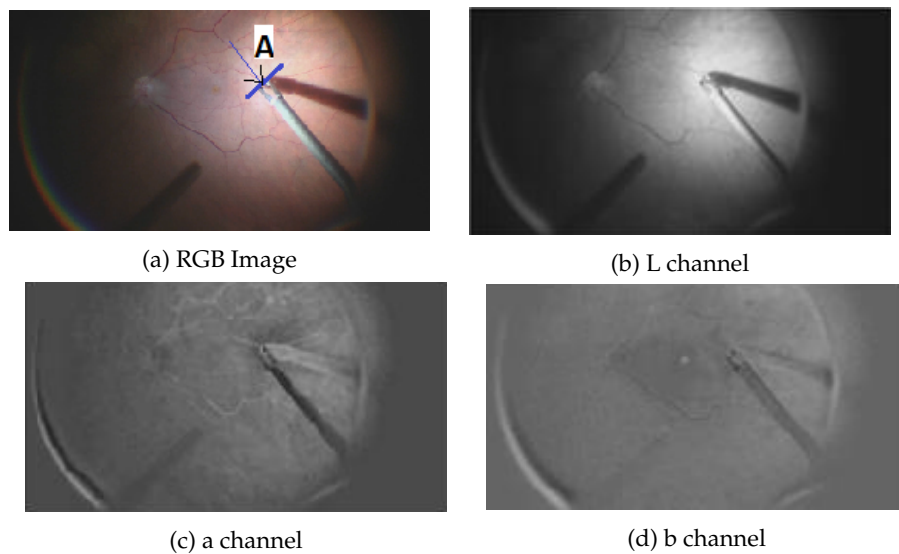


Figure 4.1: RGB image with its  $L^*a^*b^*$  Transformation

Herein, a new approach is proposed to detect and track the tool tip in microscopic images in real time surgery in a more precise fashion. We define the tool tip position as a point on the tool centerline where it touches the retinal tissues (shown in Figure. 4.1(a) as point A ). The  $L^*a^*b^*$  transform [53] is used mainly for the segmentation process as shown in Figure. 4.1, because it highlights the perceptual uniformity of the instrument which is characterized as a textureless object. Moreover, the nonlinear relations of  $L$ ,  $a^*$  and  $b^*$  channels are applied to mimic the nonlinear response of the eye. Therefore, the uniform changes in the perceived color in the eye can be obtained by uniform changes of components in the  $L^*a^*b^*$  color space. The structural information is considered to reduce the search space and to optimize for the potential instrument segment location in real time. Once the instrument segment is detected in a frame, the instrument centerline and the instrument tip are extracted and propagated to the next frame to make the detection and tracking much faster and accurate.

### 4.1.1 Instrument Segmentation

Based on experimental observations in  $L^*a^*b^*$  color space, the instrument is included in a small range of the lowest intensity values of the  $a^*$  channel within the retina region as shown in Figure 4.1 (c). Applying a thresholding function



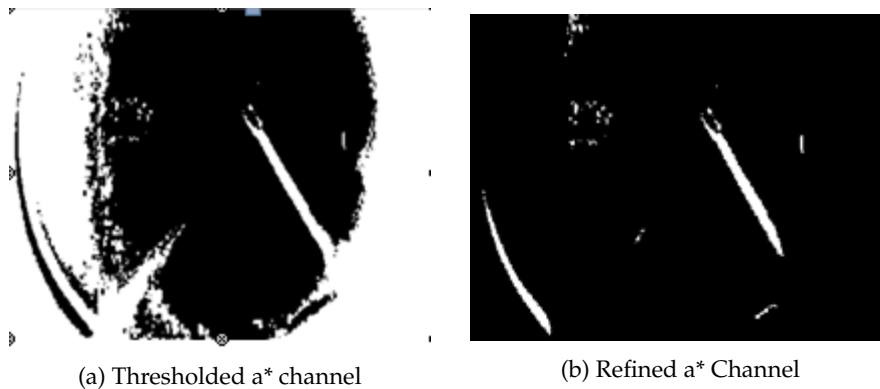


Figure 4.2: Color information extracted from a\* channel

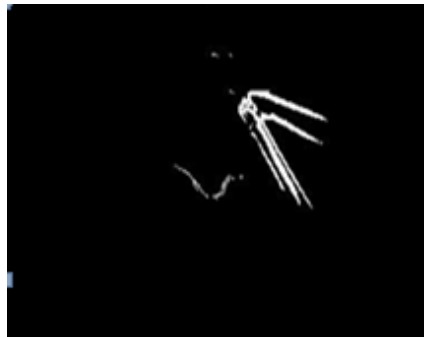


Figure 4.3: Edge Image

to filter out intensity values larger than 10% of the maximum intensity value of a\* channel gives the results shown in Figure.4.2(a). It shows that most of the instrument pixels are preserved in addition to some parts of the background. It is also worth noting the disappearance of the instrument shadow in this filtered image. This would avoid the confusion in detection between the instrument and its shadow. Moreover, more background's clutter can be removed easily by subtracting a thresholded L\* channel from the image in Figure.4.2(a) to produce the refined a\* channel as shown in Figure.4.2(b).

#### 4.1.2 Structural Information Integration

The refined a\* channel gives an image with many segmented objects. Prior information in different ways to give preference for some segments against others can be incorporated. Many of these segments could be discarded if they are not aligned with strong edges, or even if they aligned with bended edges, this is why the structural information with the refined a\* channel need to be integrated. The gradient of the green channel is used to get structural information, and after thresholding, it gives the output shown in Figure.4.3 which is called the edge image. The thresholding is important to eliminate the contribution of the background and other eye components in the edge image.

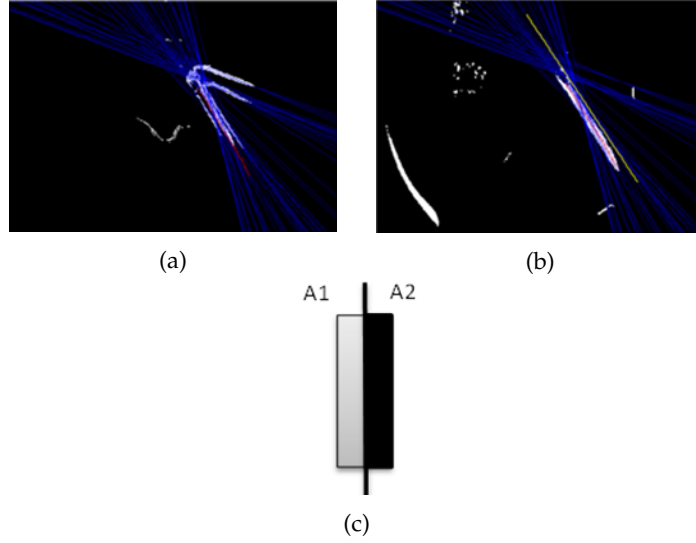


Figure 4.4: (a) The detected Hough lines in the Edge image. (b) The same Hough lines obtained from the edges image and superimposed on the refined a\* channel.(c) The tool model where the mid-line should lay on each Hough line.

From the edge image in Figure.4.3 and the refined a\* channel in Figure.4.2(b), the instrument object could be defined without resorting to the intensity values by just applying the probabilistic Hough transform [13] to detect the lines in the edge image as shown in Figure.4.4(a). The strongest 150 linear segments are extracted and superimposed on the refined a\* channel as shown in Figure.4.4(b). At each line, an instrument model as shown in Fig.4.4(c) is fit to find the instrument edge line. The model consists of two areas, where all white pixels in one area are given positive weight while the white pixels in second area are given a negative weight. The selected instrument edge line is the one which maximizes the cost function  $F$  given by Eq. 4.1.

$$F = w_1 * \sum_{p \in A_1} X(p) + w_2 * \sum_{p \in A_2} X(p) \quad (4.1)$$

where  $w_1$  and  $w_2$  are the weights given to the white pixels  $p$  of the refined a\* image  $X$  located in  $A_1$  and  $A_2$ , respectively. If the value of  $w_1$  is positive and  $w_2$  is negative, then Eq. 4.1 detects the right side of the tool shaft. The negative value of  $w_2$  is chosen to penalize the existence of the white pixels on the right side of the Hough lines. The yellow line in Figure. 4.4(b) is the detected line based on Eq. 4.1, and the white segment aligned to it is considered the instrument object.

### 4.1.3 Instrument Tip and Centerline Detection

The instrument centerline can be detected easily if the detected left and right instrument edges were parallel. Unfortunately, in most cases they are not parallel

due to the quality of the  $a^*$  channel which is affected by fast motion, image blur and the large illumination changes. As a result, the detected edges are not well aligned to the actual instrument edges. Some cases are depicted in Figure. 4.5. To overcome misalignment problem and to find the instrument centerline, it would be more robust to rely on the instrument segment itself aligned to one of the detected instrument edge line. For this end, the centers of masses for a bunch of lines  $\{L_i\}_{i=1}^m$  perpendicular to the detected line by Eq. 4.1 are computed which produce  $m \in \mathbb{R}$  candidate points. The center of mass  $(X_{L_i}, Y_{L_i})$  for each line  $L_i$  is calculated based on:

$$X_{L_i} = \frac{1}{n} * \sum_{j=1}^n x_j, \quad Y_{L_i} = \frac{1}{n} * \sum_{j=1}^n y_j \quad (4.2)$$

where  $n$  is the number of white pixels along one line  $L_i$  orthogonal to the instrument edge line, and  $x_j$  and  $y_j$  are the coordinates of these pixels. The instrument centerline is found by fitting a line to the  $m$  computed points using RANSAC [41]. The resultant line is the tool centerline. It forms a signal where the transition from the foreground to the background is the instrument tip position in case of using a vitrectomy instrument, but if the instrument has a forceps shape then the transition point is the connecting point and further processing is required to find the instrument tip position. The processing in this case is to start from the detected connecting point and find the connected components around the centerline on both sides. The farthest point in the connected component from the connecting point is kept, and the projection of this point perpendicularly on the centerline is the instrument tip position.

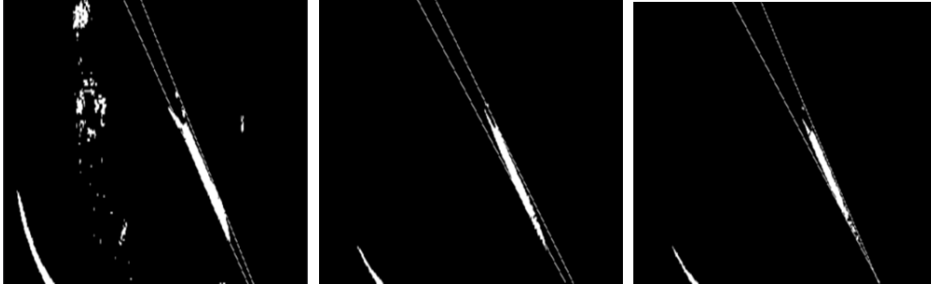


Figure 4.5: Some cases where the instrument detected edges are not parallel.

#### 4.1.4 Instrument Tip Tracking

Once the instrument centerline and the instrument tip are detected in the current frame, this information is propagated to the next frame. Therefore, there is no need to process the entire frame each time. Assuming the instrument tip position  $P_t$  and the instrument slope  $S_t$  have been detected at frame  $t$ , then the search for the candidate Hough lines at frame  $t + 1$  is limited to the lines within a rectangular box centered at  $P_t$  and tilted according to  $S_t$ . These candidates lines

are filtered out again to get rid of the ones which have a large slope difference from the instrument slope detected at the previous frame based on eq.4.3.

$$|S_{(t+1)}^{(i)} - S_t| < \epsilon \quad (4.3)$$

Where  $S_{(t+1)}^{(i)}$  is the  $i$ -th candidate line at frame  $t + 1$ , and  $\epsilon$  is a small value chosen empirically to be around 0.2.

## 4.2 Experiments and Results

This algorithm is implemented using C++ and OpenCV installed on a machine with Core-I7, 2.8GHz CPU, and it runs at 23 fps.

### 4.2.1 Retinal Microscopic Datasets

Two microscopic datasets for real human eye surgery have been used in order to validate the technique. The Datasets are captured by Carl-Zeiss Lumera 700 microscope, and 400 (1080X1920) images have been manually annotated for each dataset. The annotation includes the instrument tip position, and one point on the centerline of the shaft to calculate the instrument slope. The images were resized to one fifth of their original size during processing, while the validation and visualization both consider the original size. Fig.4.6 shows the detected instrument tip and centerline for samples from both datasets.

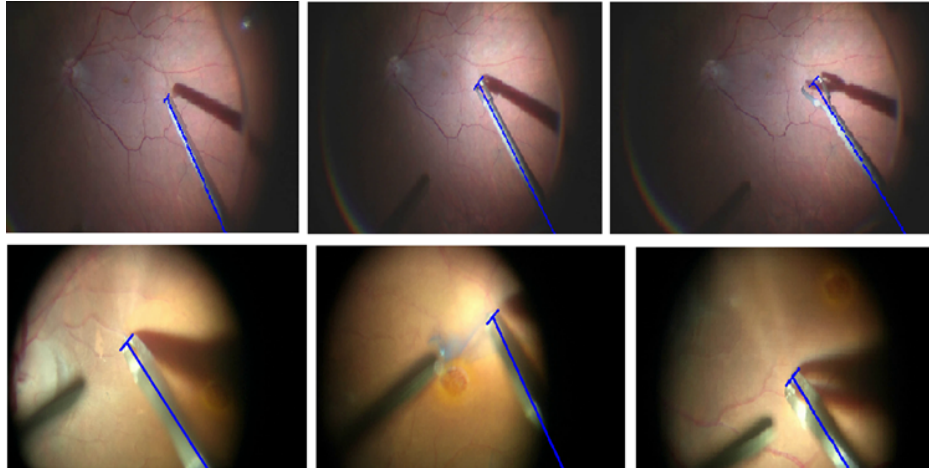


Figure 4.6: Random samples from different datasets with different conditions. The first top row is from the first dataset where the red component is prominent and the instrument is evenly illuminated. The second bottom row is from the second dataset where the green component is prominent, and the instrument is unevenly illuminated

### 4.2.2 Datasets Evaluation.

The model size is chosen empirically with width of 7 pixels on each side and height of 140 pixels. The weights  $w_1$  and  $w_2$  in Eq. 4.1 were chosen to be 1 and -5 respectively. The tracking box has the size of 20x80 pixels. The percentage of the images where the tool tip is correctly detected as a function of the accuracy threshold is calculated. For each accuracy threshold  $T_1$ , we consider the percentage of the images in which the detected instrument tip is at distance less than or equal to  $T_1$  pixels from the actual position based on the ground truth. This threshold varied from 5 to 50 pixels. From Figure.4.7, it can be noticed that the instrument tip positions have been correctly detected in 90 percent of the images within a threshold of only 20 pixels for the first dataset, which shows the high accuracy in detection. For the second dataset, the detection error is a bit higher due to the large illumination variations and the unevenly illuminated parts of the instrument, in addition to the nature of the images which are blurred in comparison with the first dataset.

Figure 4.7: Instrument tip detection accuracy measurements.

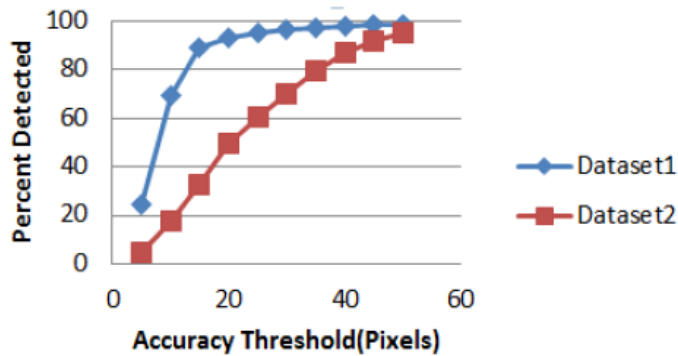


Figure 4.8: Instrument centerline detection accuracy measurements.

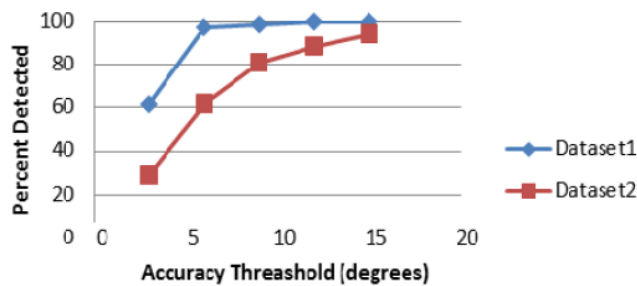


Figure.4.8 shows the accuracy of our method in detecting the centerline of the instrument. Another accuracy threshold  $T_2$  has been defined as the angular difference in degrees between the actual slope measured based on the ground truth and the detected slope of the instrument. The results show that in 90

percent of the images, centerlines were detected with angular error smaller than 6 degrees for the first dataset, and 12 degrees for the second dataset.

### 4.3 Conclusion

Herein, a new real time approach for the detection and tracking of the medical instrument in retinal microsurgery is presented. The approach detects the instrument tip position with high accuracy and without maintaining prior information about the instrument's gripper part. The results show the efficiency of using this approach to handle cases that contain not only the instrument but also blood vessels, instrument shadow and light pipe. The approach also works well regardless of blurring effects, and small lightning changes.

The datasets which used for evaluation exhibit most of the challenges issues. However, the ability of  $L^*a^*b^*$  transform to mimic the human eye perception way makes this approach an effective one to discriminate between the instrument and other components like its shadow and the light pipe. Even though these components have similar geometry, color information can filter them out. However, the quality of the  $a^*$  channel has an impact on the accuracy of the approach, and the accuracy gets lower in case of low light reflection on the instrument body. This means at severe illumination changes or no light focusing on the instrument, the segmentation of the instrument might not be reliable and have a negative impact on the tip detection. This leads us to investigate more robust techniques to cope with light changes and to sustain the accuracy detection of the current approach.

In next chapter, we investigate the performance of Convolutional Neural Networks (CNN) as a discriminative model to detect the instrument parts at the first step.

## Chapter 5

# Using Deep Learning for Articulated Instrument Detection

### 5.1 Introduction

In the previous chapter, the problem of instrument detection and tracking has been addressed relying on color information as our main features while the geometry structure is used to optimize for the instrument tip location and centerline estimation. In retinal microsurgery, the unexpected appearance changes and extreme deformation of the instrument makes the tracking a challenging task. Moreover, most of the hand-crafted features used to address the problem of instrument detection are a combination of intensity-based and simple structure features. These features might not be the perfect ones to cope with the unexpected changes during the surgery. Therefore, finding the optimal and most discriminative features plays the most essential role to enhance the practicality of the proposed method. To that end, we address the same problem in this chapter using deep learning to learn representation from raw data using non-linear transformations [16]. The most popular architecture for realizing deep learning is Convolutional Neural Networks (CNN) which has been applied in many applications in computer vision, speech recognition and natural language processing (NLP). In the field of computer vision, CNNs have been proposed for challenging problems, such as classification [59, 56, 73], regression [75], localization [26], detection [104], segmentation [94, 69, 90], feature extraction [28, 96], crowdsourcing [1] and pose estimation [112, 80, 66]. State-of-the-art results have been achieved using CNN in these tasks.

In the proposed approach a CNN is used to detect an articulated instrument. The network is trained and employed as a feature extractor to map the appearance of instrument parts to the appropriate class label. Testing the input image using this trained network produces a probability map for each part of the articulated instrument. The maps localize the potential instrument parts in the 2D image space and they are employed in a CRF model as unary potentials

outputs. Therefore, the search space for the instrument parts in 2D images can be limited to only seeds points of the highest probabilities in these maps. Hence, many approaches can work on the reduced space to predict the instrument joints. To preserve the robustness of the approach against illumination changes, structural information of the seeds points are employed to model the binary potentials in the CRF model. Therefore, instrument parts detection and the refinement of the final joints localization are integrated in a CRF model where the inference predicts the instrument center point coordinates and estimates instrument orientation in one step.

In the next section, the problem is formulated as a CRF inference problem. Next, the CNN architecture for unary potentials is presented and followed by modeling the pairwise potential and regularization term. Finally, we show the achieved results on public and Zeiss datasets.

## 5.2 Problem Formulation

The proposed approach models the instrument configuration  $Y$  as a simple graphical model using a CRF of two random variables where each variable  $Y_i \in Y$  corresponds to instrument part's coordinates. The two parts used in this approach are the instrument's center point and the instrument's shaft. Considering now an instance of the observation  $x \in X$  (i.e. instrument part features), and an instrument configuration  $y \in Y$ , the posterior becomes:

$$P(y|x, P) = \frac{1}{Z(x)} \prod_i^n \phi_i^{conf}(y_i, x) \cdot \prod_{(i,j) \in E} \psi_{ij}^{struct}(y_i, y_j, x_i, x_j) \cdot \prod_{(i,j) \in E} \psi_{ij}^{Temp}(y_i, y_j, P) \quad (5.1)$$

where  $Z(x)$  is the normalization factor,  $n$  is the number of instrument's parts and  $E$  is the edge connecting between one sample from the instrument's center part hypotheses and another sample from the shaft's hypotheses. Next, we define each term in Eq.5.1 with more details.

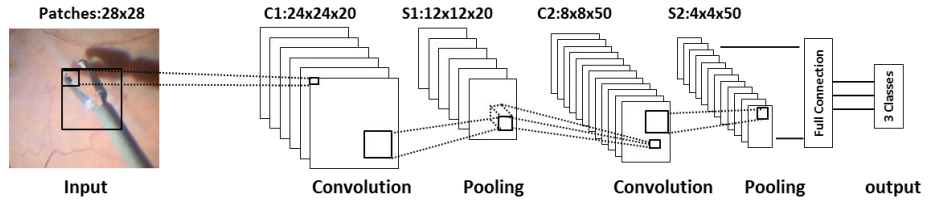


Figure 5.1: The designed CNN: Filters sizes = 5x5, Pooling size= 2x2. The numbers of features channels are 20 at layer 1 and 50 at layer 2.

### 5.2.1 Unary Potentials Using CNN

The unary potentials function  $\phi_i^{conf}(y_i, x_i)$  in Eq. 5.1 is designed to give a score for each pixel in the image indicating the confidence to which part it



belongs based on the observations. To detect each part and get a probability of its detection confidence, we use a modified LeNet [64] as a multiclass detector. The network is learned from examples  $S = \{p_i, c_i\}_{i=1}^N$ , where  $p_i$  is an  $r \times r$  image patch,  $N$  is the number of training patches and  $c_i \in \{1, \dots, C\}$  is a class label. Below, we describe the layers of our designed network shown in Figure. 5.1.

**Input Layer:** The input patch size to the network is  $28 \times 28$  where each patch has an associated label. We use different label for each of the instrument center part, instrument shaft part and background. Samples of these parts are shown in Figure. 5.2.

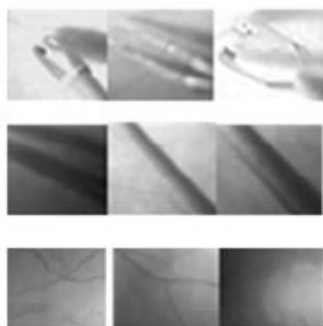


Figure 5.2: Patches samples of size  $28 \times 28$ , where each row was chosen from a different class: center, shaft, and background respectively.

**Convolutional layer C1:** The input patch is processed by convolutional kernels of size  $5 \times 5$  to produce feature maps of size  $24 \times 24$  pixels. Each unit in each feature map is connected to a 5 pixels cell in the input. We use at this layer 20 feature maps where one kernel produces one map using convolution operation with the input image. C1 layer contains 520 trainable parameters. The kernels at this layer are trained to capture low-level features like edges and corners.

**Pooling layer S1:** This layer is the first sub-sampling layer with 20 feature maps of size  $12 \times 12$ . Each unit in each feature maps is connected to  $2 \times 2$  cell in the corresponding feature map in C1 and the unit value is set to the maximum value in that cell. The cells in this network are non-overlapping. Therefore, feature maps have half the size of feature maps at C1 layer.

**Convolutional layer C2:** This layer works directly on pooling layer S1 output with convolution kernels of size  $5 \times 5$ . The number of feature maps is 50 and each unit of each map is connected to  $5 \times 5$  cell in the corresponding location in S1 layer. Therefore, each map size is  $8 \times 8$  and the number of trainable parameters at this layer is 25050. In contrast to C1 layer, C2 layer filters are trained to capture higher-level features in the object like curvatures or other structural components.

**Convolutional layer S2:** Sub-sampling is applied again on the C2 feature maps to generate the same number of feature maps but with half the size ( $4 \times 4$ ). Max

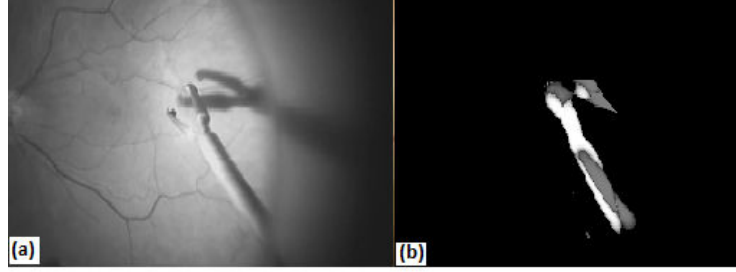


Figure 5.3: (a) Instrument example. (b) CNN output example

pooling is used which is functioning similar to S1 layer pooling.

**Fully connected layer:** This layer has 500 neurons and connected to each neuron in the output layer. Therefore, a new representation is obtained for the image patch based on the trainable weights of the entire network.

**Output layer:** We use softmax loss function to normalize the network output which consists of three neurons. The output  $z_i$  corresponding to neuron  $i$  is given by:

$$z_i = P(y = c_i | x) = \frac{e^{(\sum_j w_{ij} a_j^{(l)} + b_i)}}{\sum_{k=1}^3 e^{(\sum_j w_{kj} a_j^{(l)} + b_k)}} \quad (5.2)$$

where  $x$  is the input patch,  $a_j^{(l)}$  is the  $j^{\text{th}}$  neuron in the last layer preceding the output layer and  $w_{ij}$  is the weight along the connection between  $a_j^{(l)}$  and the output neuron  $z_i$ . Each output  $z_i$  estimates a probability of belonging to a class  $c_i$ . Therefore, softmax function serves as a normalization function as well as nonlinear operator. The error function used in this network is the multi-class cross-entropy error which is given by:

$$E(\theta) = \sum_{i=1}^m \sum_{j=1}^k \delta_{y,j} \ln(z_j) \quad (5.3)$$

where  $m$  is the number of patches,  $k$  is the number of classes and  $\delta_{y,j}$  is the Kronecker delta [21]:

$$\delta_{y,j} = \begin{cases} 0 & \text{if } (y \neq j) \\ 1 & \text{if } (y = j) \end{cases} \quad (5.4)$$

Once the error function is computed, the error is back-propagated into the network to update the weights. The network is trained from image patches to generate three probability distribution maps such that one map for each class. The background probability map is discarded, while some candidates with high probabilities are sampled from both the instrument's center probability map

(white color pixels in Figure. 5.3. (b)) and the instrument's shaft probability map (gray color pixels in Figure. 5.3 (b)) to form some instrument configurations hypotheses.

### 5.2.2 Pairwise Potentials

The pairwise potential function  $\psi_{i,j}^{struct}(y_i, y_j, x_i, x_j)$  is designed to model the relation between the instrument's two parts. To that end, we use the structural information associated with each configuration  $(y_i, y_j)$ , where  $y_i$  are the coordinates of a sample selected from the instrument's center probability map and  $y_j$  are the coordinates of a sample from the shaft candidates' map. Each configuration can be described by the joint features of its two parts. The descriptor for each instrument part is a feature vector extracted from a window centered at the corresponding coordinates in the colored image. To model the relation between the instrument's parts, we train a regression random forest on the joint HOG features of the two samples constituting the configuration. In the training phase, two points are annotated in each image as shown in Figure. 5.4 and the descriptor of the configuration is constructed from two patches around each of the points P1 and P2. Figure. 5.4 shows a positive and correct configuration of the instrument, while the negative configurations are chosen from random points from the background. During the testing phase,  $K$  configurations are randomly sampled from the highest probabilities of the unary distributions. Each configuration  $(y_i, y_j)$  is tested with the regression forest and the output represents the likelihood to the instrument structure. We can formulate the pairwise potential function using Eq.5.5.

$$\psi_{i,j}^{struct}(y_i, y_j, x_i, x_j) = \frac{1}{B} \sum_{b=1}^B S_b(x_{y_i}, x_{y_j}) \quad (5.5)$$

where  $B$  is the number of trees in the forest,  $(x_{y_i}, x_{y_j})$  are the extracted features of the configuration  $(y_i, y_j)$ , and  $S_b$  is the prediction assigned to the features from one tree. The predictions from all  $B$  trees are aggregated into a probabilistic manner to express the pairwise probabilities.

### 5.2.3 Regularization Term

The regularization term  $\psi_{i,j}^{Temp}(y_i, y_j, P)$  in Eq.5.1 is a pairwise potential function designed to prevent far jumps in detection and to give favor to a new configuration with a small difference in orientation from the detection in the previous frame  $P$ . We assume a Gaussian distribution over the orientations of the  $K$  sampled configurations at the frame, estimate the parameters  $\mu$ , and  $\sigma$  of this distribution, and propagate them from one frame to the next to give favor to small changes in the orientation. The regularization term is given in Eq.5.6.

$$\psi_{i,j}^{Temp}(y_i, y_j, P) = \mathcal{N}(\alpha(y_i, y_j) | \mu_p, \sigma_p) \quad (5.6)$$

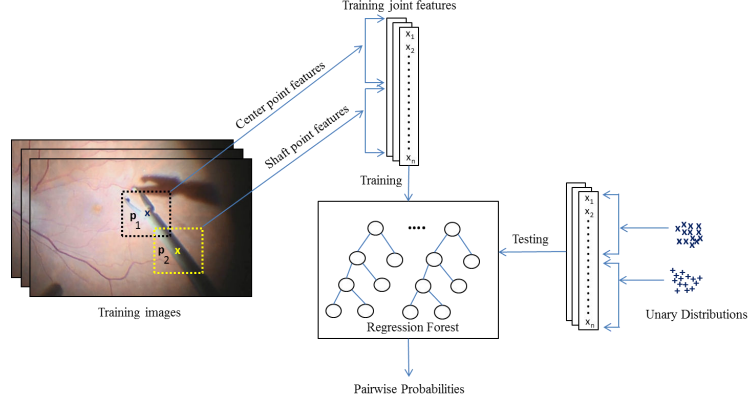


Figure 5.4: Regression Random Forest Model learned on joint features of point pairs which represent a configuration

where  $\alpha(y_i, y_j)$  is the orientation of one sampled configuration and  $(\mu_p, \sigma_p)$  are the estimated orientation and the standard deviation at the previous frame respectively. Finally, one configuration out of  $K$  sampled ones which maximizes the posterior probability given by Eq. 5.1 is chosen by heuristic iterations through the best candidates.

### 5.3 Experiments and Results

The experimental validation of the proposed algorithm is carried out on two different Retinal Microsurgery (RM) datasets: the first one is a public fully annotated dataset of three sequences of retinal surgery [105]. The second one is a new dataset, referred to as Zeiss dataset, comprising three real in-vivo RM surgeries with 1500 manually annotated images at 1920x1080 resolution each. These images are resized to one fourth of their original size for faster processing. The CNN is trained on the first half of the images from each sequence, where the patch size of  $50 \times 50$  pixels is extracted for each instrument part and from the background and resized to the CNN input size, as shown in Figure. 5.2. The learning rate of the network is empirically chosen of 0.001. The stochastic gradient descent uses batch size of 100 patches to update the network parameters in each iteration where the number of iterations is set to 100. The regression random forest is trained on the HOG features extracted from pairs of  $32 \times 32$  patches from the first half of the images. For the positive samples, one patch is centered at the instrument's center point, while the other at a point located at the instrument shaft centerline. The patches of the negative samples are chosen randomly from the background. The number of trees in the random forest is set to 20, and the candidates number  $K$  is set to 1000. The performance of the algorithm is evaluated by means of two different metrics: Accuracy Threshold score used by Sznitman et al. [105], and Angular Threshold score used in [6]. The accuracy threshold score gives pixel-wise prediction quality of the center point of the instrument, while the angular threshold expresses the quality of estimating the orientation in terms of degrees. Once the instrument center is

detected, the detection is bounded to a small Region of Interest (ROI) around the center. The re-initialization is done automatically by gradually expanding the ROI when some parts are missing in the detection.

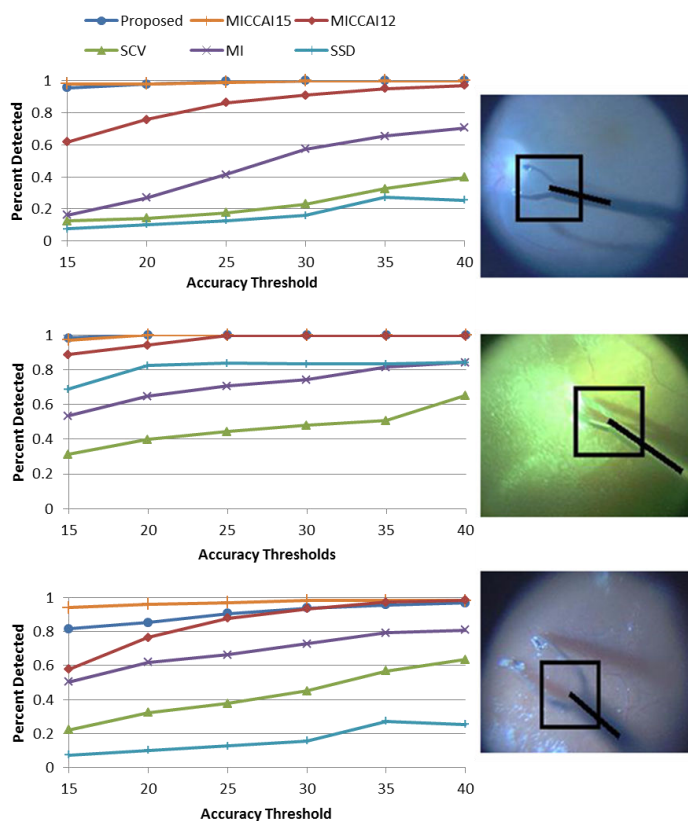


Figure 5.5: Results for each sequence of the public dataset when trained and tested on separate sequences. The bounding box is centered on the detected instrument's center.

### 5.3.1 Public Dataset

This dataset consists of three sequences of retinal microsurgery with total of 1170 images of  $640 \times 480$ . Augmentation of the training images is carried out by simple translation and rotation of the input patches and no downsampling is done on the images. We compare our method with state-of-the-art methods: MI [12], MICCAI15 [88], MICCAI12 [105], SCV [81], and SSD. We use accuracy threshold values as defined in [105] where the threshold varies from 15 to 40 pixels. At each threshold  $T$ , all predictions of center points coordinates within  $T$  pixels from its ground truth locations are considered correctly detected. First, we evaluate the algorithm for every sequence separately by training the CNN on the first half of the sequence and testing on the second half. The results are shown in Figure. 5.5. Then the training and testing are done on the full dataset

by training another CNN on the first halves of all sequences, and testing on the rest and the results are shown in Figure. 5.6. It shows the robustness of the proposed method which can work on the full dataset as good as on the separate sequences. The approach can correctly predict the center joint coordinates in 90% of the images at threshold of 20 pixels which is the instrument shaft diameter. Moreover, it shows comparable results to the state-of-the-art methods in terms of the detection accuracy. The most important part which influences the accuracy of the final predictions is the unary detections. These detections are produced from CNN trained from 50% of the data and validated on the remaining samples. We noticed from our experiments that the convergence of the energy function is very good as shown in Figure. 5.7 and the error remains at low value from epoch 60 to the end of the experiment.

The results in Figure.5.8 show the performance of our approach when varying the angular threshold from 3 to 24 degrees. First, each sequence was tested separately, and then the whole dataset was tested. The results show that in 90% of the images the centerline of the instrument is extracted with an angular deviation of less than 10 degrees as shown in Figure.5.8.

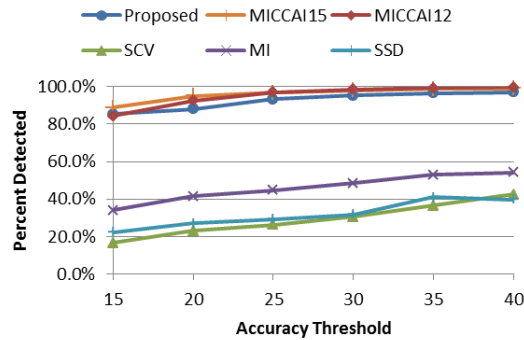


Figure 5.6: The results for the full dataset, when learned on the first halves from each sequence and tested on the second halves

### 5.3.2 Zeiss Dataset

Zeiss dataset represents more challenging cases where the images include clear appearance of vessels and instrument shadow in addition to the cluttered background. Moreover, it has severe light changes within each sequence of the set. We evaluate our approach by training on the first halves of the first two sequences and testing on the second halves in addition to the third unseen sequence. Since the average diameter of the tool shaft is 50 pixels for this dataset, we evaluate the pixel-wise prediction accuracy using thresholds values between 40 and 100 pixels. The angular threshold varies from 5 to 45 degrees. Some samples of the results from each of the sequences are shown in Figure.5.10. The result in Figure.5.9 shows the accuracy of our approach in detecting the instrument center. It shows that in 83% of the images the instrument center point has been correctly predicted within a threshold of 50 pixels. Moreover,

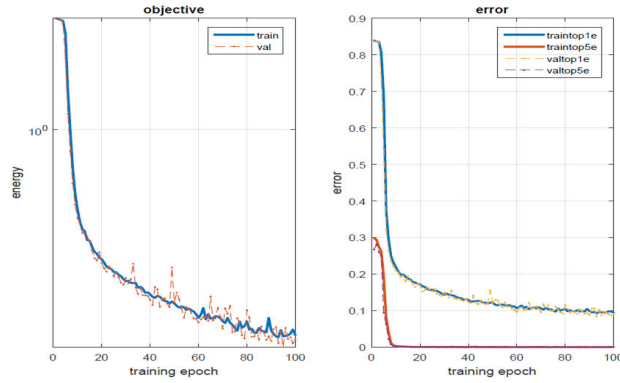


Figure 5.7: The objective function and error curves after each epoch of CNN training from the full public dataset

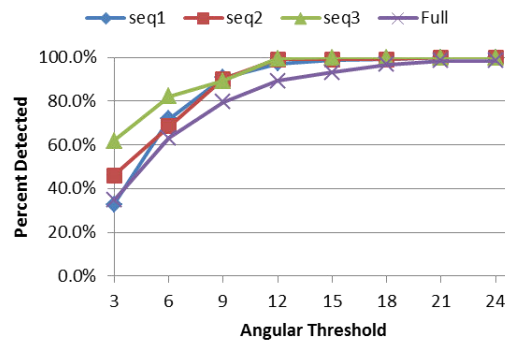


Figure 5.8: The results based on the Angular Threshold for both cases

the performance of the approach on the unseen sequence is very similar to the other sequences. Therefore, the approach shows its ability to generalize for other datasets. The angular threshold results for shaft orientation estimation are shown in Figure.5.11. Since the algorithm gives a favor to small changes in orientation, following the true orientation is slower than expected when there is fast motion, which accounts for high angular error in some cases compared to the public dataset. However, for the first and the third sequences, the centerline of the instrument is extracted with an angular deviation less than 20 degrees in 90% of the images. On the other hand, sequence 2 has more fast motion which influences the performance of the regularization term only. The designed CNN shows very good convergence of the objective function as well as the error remains at low levels starting from the training epoch number 40 as shown in Figure. 5.12. The results demonstrate that for most epochs the error is less than 0.1 and the energy function approaches zero.

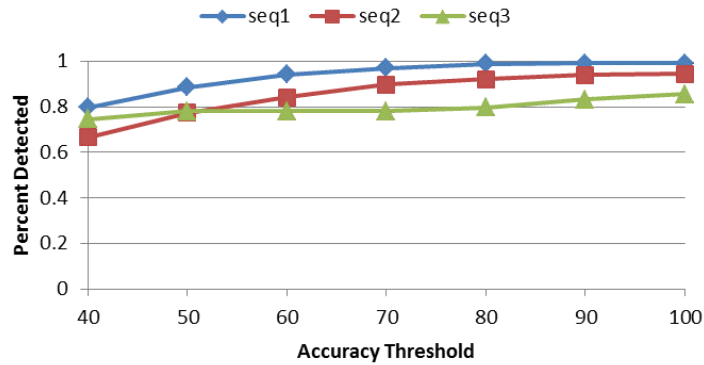


Figure 5.9: The results for the full Zeiss dataset.

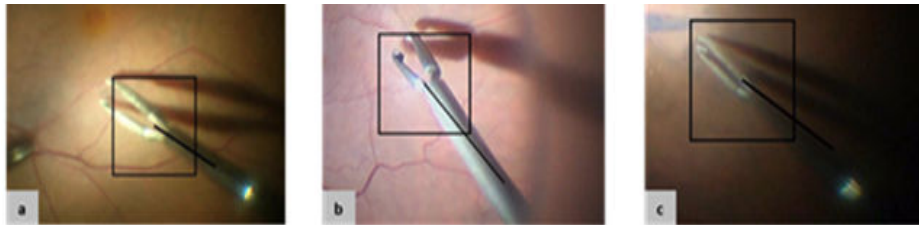


Figure 5.10: Samples of the results showing the detected joint point and estimated centerline.

## 5.4 Conclusions

In this chapter, a new approach has been presented for instrument center point detection and orientation estimation. The approach employs a CNN network for unary detections which are embedded into a CRF posterior function in a probabilistic manner. Some samples from the unary maps are used and their combinations are tested using regression forest to express the pairwise potential probabilities for the CRF model. The temporal information has been integrated into the model as a regularization term to ensure smoothness in instrument tracking from one frame to another. The approach demonstrates high performance in detecting the instrument center point and estimating the shaft orientation in challenging cases with severe light changes. However, using CNN for unary detections requires big amount of data for training and can achieve the real time requirements on GPU-based microscopes. Therefore, reducing the amount of data and maintaining the real time performance on CPU machines open up new prospects of advancement with this approach. Hence, considering the instrument as an articulated object and imposing more constraints on the relations among its parts are the most important messages we got from this method. In next chapters, we will see how to make use of this conclusion and employ the part-based detections in more efficient ways.



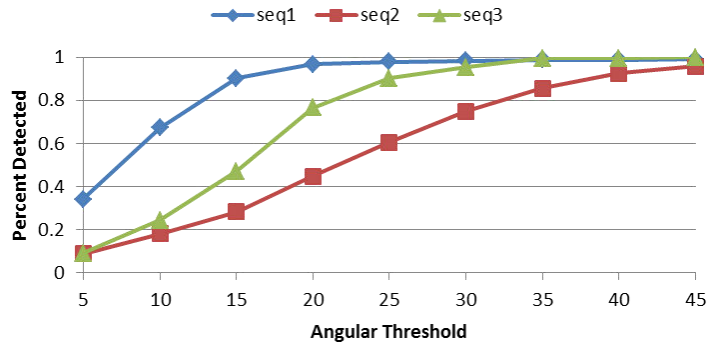


Figure 5.11: The angular thresholds results for the full dataset.

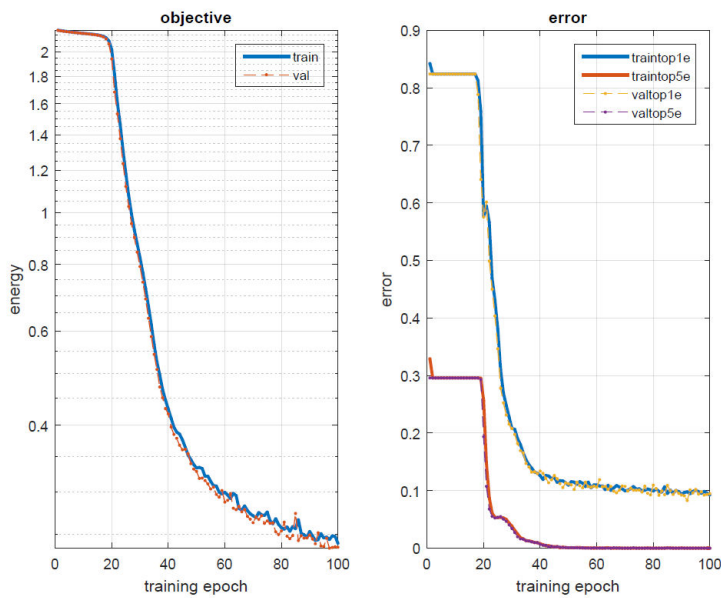


Figure 5.12: The objective function and error curves after each epoch of CNN training from the full Zeiss dataset

CHAPTER 5. USING DEEP LEARNING FOR ARTICULATED INSTRUMENT  
DETECTION

---

## Chapter 6

# Deep Architecture for Instrument Pose Estimation

### 6.1 Introduction

In this chapter, we pursue the utilization of deep learning architectures for the benefit of pose estimation in medical surgery. While the network designed in the previous chapter is used for part-based detection, here we focus on modelling our problem as a deep learning regression task working on the entire image. The main idea is to learn representations from the entire image in order to regress the coordinates of many instrument parts within the image space. The motivation of this work is to reduce the computation costs of using patch-based processing which requires processing thousands of patches in a sliding window way through CNN pipeline in order to classify each patch. Therefore, this work aims to estimate the instrument pose by localizing the instrument articulation points using only one forward propagation of the input. However, being able to predict many parameters relying only on a single-patch input requires the network to be deep enough to understand the relations among different image components. Using such deep regression networks has achieved state-of-the-art results for human pose estimation problem [112, 80, 66] and automated mitosis detection [30]. Moreover, regression CNN have demonstrated a good performance in estimating 2D/3D registration parameters [75].

In this chapter, we focus on the pose estimation of medical forceps instrument and model it as a regression problem. Deep regression CNN is employed for this task to train a model in end to end way to predict the locations of the instrument articulation points. Once trained, the network is fed with the entire image or a region of interest (ROI) containing the whole body of the instrument to estimate the pose in a single forward propagation step.

In the next sections, we present the designed regression CNN and evaluate its performance on different sequences from different real surgeries.

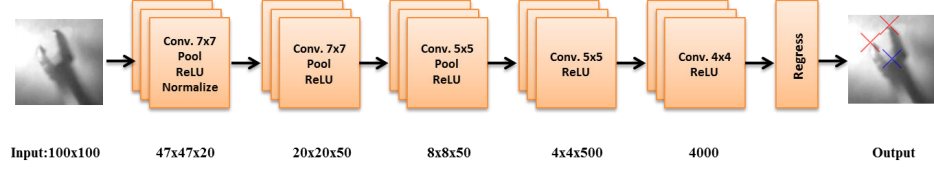


Figure 6.1: Deep convolutional architecture for instrument pose estimation: Convolutions use different kernel sizes as indicated inside each box.

## 6.2 Problem Formulation

In this section, we introduce a new CNN architecture to model the pose estimation as a regression problem. The input to the network is an image  $\mathbf{x} : \Omega \rightarrow \mathbb{R}$  and the output is a real valued vector  $\mathbf{y} = (y_1, y_2, \dots, y_N)$  of  $N$  instrument joints, where  $y_i \in \mathbb{R}^2$  is the coordinates of a single joint in the 2D image space. Given a training dataset  $\{(x_k, y_k)\}_{k=1}^K$  of  $K$  samples, the goal of the CNN is to find a representation function  $\phi(\cdot)$  that minimizes the difference between the output of the regression loss function and the desired output vector  $\mathbf{y}$  using backpropagation [65] and stochastic gradient descent [23]. The learnt parameters  $\theta$  of the mapping function  $\phi(\cdot)$  represents the network parameters needed to produce the output  $\hat{\mathbf{y}}$  given the input  $\mathbf{x}$  as:

$$\hat{\mathbf{y}} = \phi(\mathbf{x}; \theta), \quad (6.1)$$

where  $\hat{\mathbf{y}}$  is the estimated output vector representing the instrument pose. The network architecture used for this task is shown in Figure 6.1, where the input is an image and the output pose vector has three joints coordinates labeled in cross signs and overlaid on the input image.

Next, we present the details of the network architecture and the used loss function for our regression task.

### 6.2.1 Deep Network Architecture

The input to the network is an image of  $100 \times 100$  pixels and is convolved at the first layer with 20 kernels of  $7 \times 7$  pixels to produce 20 feature maps each has the size of  $94 \times 94$  pixels. The output of the convolution is normalized and goes through pooling and ReLU units to produce images of size  $47 \times 47$  pixels which are indicated on the first block in Figure 6.1. In the second block, we use again convolution, pooling and ReLU units with the same parameters as used in the former layers to produce 50 feature maps of sizes  $20 \times 20$  pixels. Next blocks have different units as illustrated in Figure 6.1 and at the end, a fully connected layer of 4000 neurons is integrated and connected to each neuron of the output layer.

## 6.2.2 Loss Function

The function of CNNs is to find a non-linear representation of the input image  $\mathbf{x}$ . To that end, we set the number of the output neurons at the last layer to be equal to the number of coordinates we want to regress, and we use the  $L2$  [112] loss function given in eq. 6.2 to compute the error between predicted and estimated joints coordinates.

$$E(\theta) = \frac{1}{2} \sum_{k=1}^K (y_k - \phi(x_k; \theta))^2 \quad (6.2)$$

This loss function is differentiable and the error can be easily computed and back-propagated using chain rule [83] as we explained in chapter 2.

## 6.3 Experiments and Results

Regression CNN has been validated on two different Retinal Microsurgery (RM) datasets: the first one is a public fully annotated dataset of three sequences of retinal surgery [105]. The second one is a Zeiss dataset, comprising three real in-vivo RM surgeries with 1200 manually annotated images. The learning rate of the network is empirically set to  $10 \times 10^{-6}$  and the momentum to 0.9. The stochastic gradient descent uses batch size of 100 patches to update the network parameters in each iteration where the number of iterations is set to around 150. The performance of the algorithm is evaluated by means of two different metrics: Accuracy Threshold score used by Sznitman et al. [105] as explained in previous chapter and the strict Percentage of Correct Parts (strict PCP) [39] which is a quality measure of the prediction of a part of an articulated object. The part is correctly predicted if the distances between its two predicted joints and their corresponding ground truth coordinates are less than a threshold  $\alpha \cdot R$ , where  $R$  is the ground truth part length, and  $\alpha$  is a fraction of that length. The algorithm is implemented using MatConvNet [115] and takes 2 – 3 hours to train each network, and 0.05 seconds to test each image on a normal *i7* personal computer.

### 6.3.1 Public Dataset

This dataset consists of three sequences of retinal microsurgery with total of 1170 images of  $640 \times 480$ . Training and testing are done on cropped patches around the forceps connecting point with size  $100 \times 100$ . The purpose is to evaluate the quality of the joints localization within the specified patches of the designed network. We compare our method with state-of-the-art methods: MI [12], MICCAI15 [88], MICCAI12 [105], SCV [81], and SSD. We use accuracy threshold values as defined in [105] where the threshold varies from 15 to 40 pixels. First, we evaluate the algorithm for every sequence separately by training the CNN on the first half of the patches and testing on the second half. The results are shown in Figure. 6.2. Then the training and testing are done on the full dataset by training one more CNN on the first halves of all patches, and testing on the rest and the results are shown in Figure. 6.3. In both cases, we

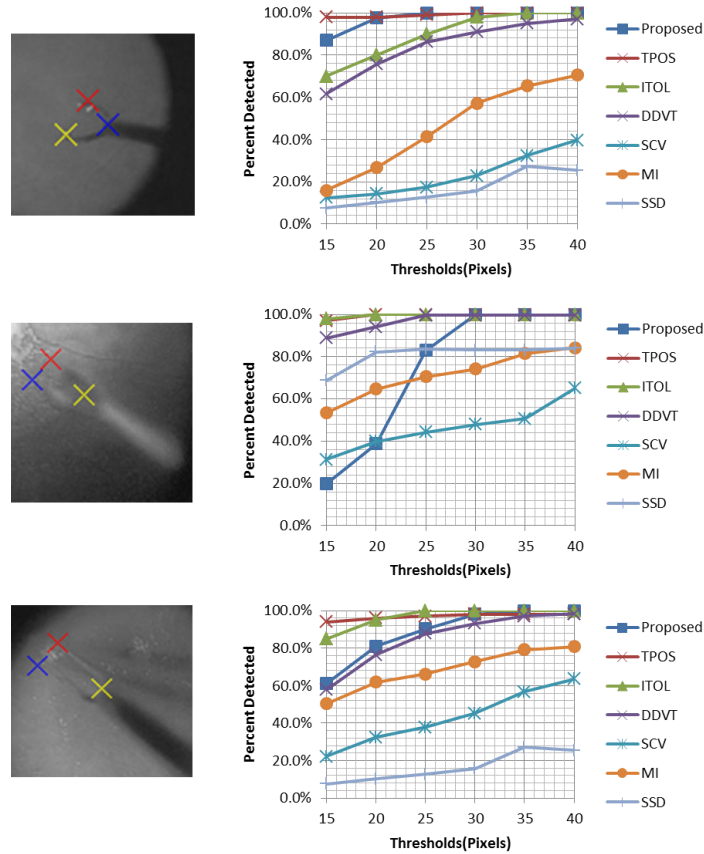


Figure 6.2: Results for each sequence of the public dataset, when learned and tested on separate sequences.

compare the detection accuracy of the instrument center point which shows comparable results to the others. However, the performance of our algorithm on the second sequence doesn't show very good results due to the length of this sequence. With only 100 patches, which is half of the sequence length, the network can't be well-trained to recognize all variations in test patches. In sequence 1, our approach achieves state-of-the-art performance, where the center point is correctly detected in 98% of the patches at accuracy threshold of 20 pixels. However, in the third sequence, we achieved the third score which can be attributed to the noise existing in the background of the testing patches which were not modelled in the training process. Therefore, as the network localizes the joints based on the entire image, existing of such noise would fire the wrong neurons and deactivate the right ones. Moreover, the pooling layer keeps the maximum or the average of the signals in block-wise manner, which might increase the dominance of the noisy signals in the subsequent layers. Therefore, it would have a negative influence on the final pose estimation. The objective function convergence of our model for the full dataset is shown in Figure 6.4 which proves that there is no overfitting in our model and the error remains constant at low values after 50 epochs.

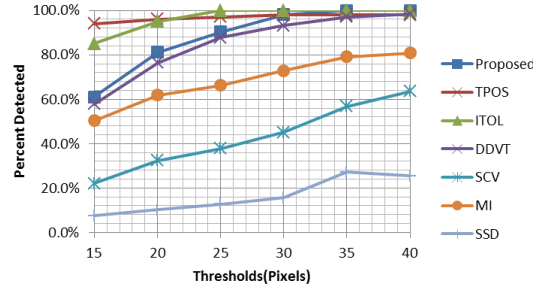


Figure 6.3: Accuracy Thresholds results for testing the model that trained from all sequences

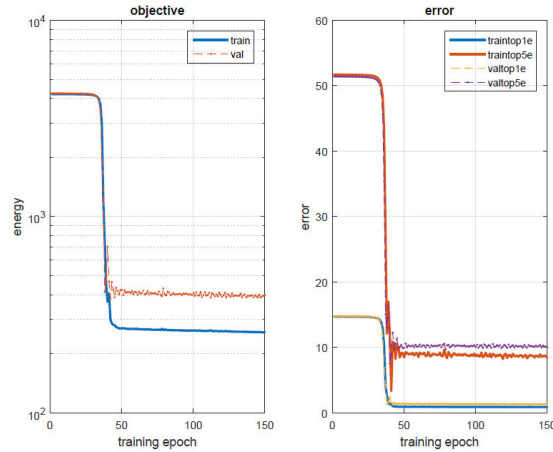


Figure 6.4: The objective function and error curves after each epoch of CNN training from the full public dataset

Strict PCP scores for detecting the left and right gripper parts for separate and full datasets are depicted in Table 6.1, which shows very good results on the first and third sequences while the performance is not that high for sequence 2 due to the aforementioned reasons. However, our network shows promising results to localize not only the center point but also the two tips of the forceps. The results demonstrate that at  $\alpha = 0.5$ , which is used in human pose estimation comparisons, the parts are correctly detected in all images of the first and the third sequences, and in 70% of the second sequence.

### 6.3.2 Zeiss Dataset

Zeiss dataset consists from three sequences where the images are captured at  $1920 \times 1080$  resolution. Due to the huge image size, cropped image around the instrument center point with size  $400 \times 400$  pixels are produced and resized to the standard size of our regression CNN input. We evaluate our approach by training on the first halves of the three sequences and testing on the second

Table 6.1: Strict PCP scores for different  $\alpha$  values for public dataset sequences.

$\alpha$	Seq1		Seq2		Seq3		Full	
	Left	Right	Left	Right	Left	Right	Left	Right
0.30	70	74	19	19	64	80	17	47
0.35	83	87	24	24	82	87	30	56
0.40	91	92	34	34	94	95	38	64
0.45	96	96	47	47	98	99	50	73
0.50	100	100	70	70	100	100	61	79
0.55	100	100	96	96	100	100	71	83

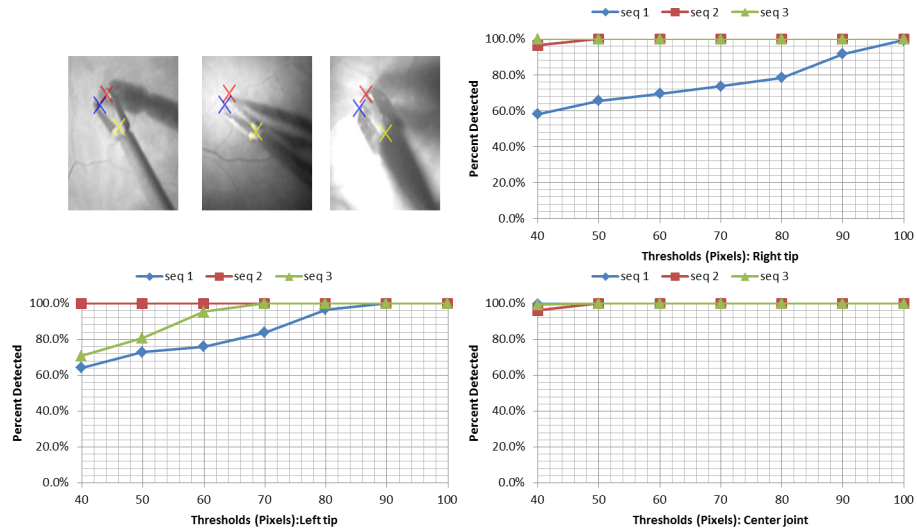


Figure 6.5: the results for the full dataset, when learned on the first halves from each sequence and tested on the second halves.

halves. Since the average diameter of the instrument shaft is 50 pixels for this dataset, we evaluate the pixel-wise prediction accuracy using thresholds values between 40 and 100 pixels.

Some samples of the results from each of the sequences are shown at the top left part of Figure.6.5. The result in Figure.6.5 shows the accuracy of our approach in detecting the instrument left, right, and center point. It shows that in almost all images of the three sequences, instrument center point has been correctly predicted within a threshold of 50 pixels. Moreover, at the same threshold, the network can correctly localize the instrument left and right tips in 80% and 93% of the images respectively. The high performance of the proposed network is emphasized by strict PCP scores, given in Table 6.2, which achieves correct left and right parts localization in around 97% and 93% of the images respectively. Moreover, Figure 6.6 proves the good performance of the networks which converges very well during training starting from the fiftieth epoch. The prediction error of some joints reduced from 50 pixels at the beginning of training to around 5 pixels at the last epoch.



Table 6.2: Strict PCP scores for different  $\alpha$  values for Zeiss dataset sequences.

$\alpha$	Seq1		Seq2		Seq3	
	Left	Right	Left	Right	Left	Right
0.30	71	65	98	97	69	98
0.35	74	68	24	24	72	100
0.40	78	71	100	100	79	100
0.45	85	74	100	100	91	100
0.50	95	78	100	100	95	100
0.55	99	87	100	100	100	100

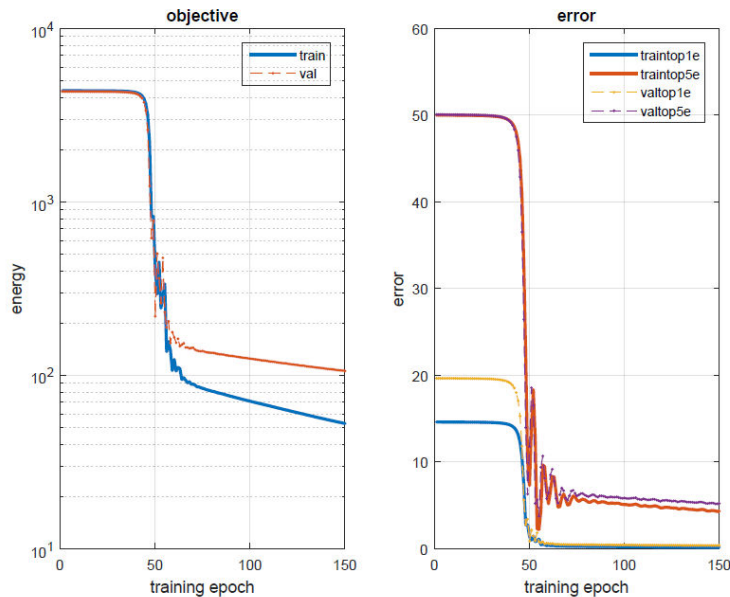


Figure 6.6: The objective function and error curves after each epoch of CNN training from the full Zeiss dataset

Generalization ability of such networks is limited to the amount of training samples and the amount of variations that should be augmented in the training samples. We tested our trained model on fourth unseen sequence with different background structures, and we observed that the strict PCP has scores of 57% and 60% for the left and right parts prediction at  $\alpha = 0.5$ . This result proves that modelling background variation should be considered in augmenting our training samples in order to maintain the performance as high as it is on the other three sequences. However, modelling the background variations is a challenging task due to the existence of vessels and other structures which can appear in different orientations, locations and scales. This is why part-based CNN networks, which doesn't need background modelling, has the tendency to be more accurate and less sensitive to the noise. However, the accuracy of the proposed network is very promising and can be improved further by learning more background variations.

## 6.4 Conclusions

We presented a new regression CNN architecture that can localize the instrument joints within an image patch using one forward propagation step. The network regresses the instrument coordinates in real time speed and achieves good localization accuracy within a given patch. However, for full utilization of this network, it needs to be integrated with tracking algorithms to be constantly provided with the proper patch, or the patch in the next frame should be extracted based on the pose predicted in the current frame. Furthermore, re-initialization scheme is still needed to handle tracker deviations and instrument disappearance at real time performance.

## Chapter 7

# Instrument Pose Estimation Based on Reliable Hough Voting

### 7.1 Introduction

In the previous chapter, the problem of 2D instrument pose estimation within patches containing the entire instrument body has been investigated. In such holistic approach, the image patch should be extracted in advance from the entire big image using other detection or tracking methods. The idea of combining tracking or detection algorithms with pose estimation is to provide the surgeon with a convenient and robust method which can continuously follow the moving instrument at the video frame rate. This has the potential to minimize the operation time by minimizing the human intervention needed to handle tracking failure. Holistic approaches are characterized by their ability to predict the locations of instrument joints relying on the complete data of image patches. Therefore, considering the complete data for predictions at testing time introduces some difficulties to handle occlusions and noisy data [15]. Hence, the potential of such methods to generalize for unseen datasets from different surgeries can be low. Moreover, the detection accuracy achieved using holistic deep architecture in previous chapter is lower than part-based detection accuracy in chapter 5. In retinal microsurgery, where the precision of micro-millimeter is a demanding requirement, part-based approaches tend to be more promising solution as they are less sensitive to the noise. Moreover, these approaches can easily incorporate prior information to constrain the final prediction of the joints [120].

In this chapter, a new part-based approach is proposed to detect, track, and estimate the pose of forceps instrument with the ability to automatically handle tracking failures. In this work, the localization of the instrument joints doesn't rely on the direct output of parts classification process, but it depends on combining guidance information from all detected instrument parts. Therefore, the proposed approach introduces a new form of the Hough Forest [43] that

incorporates unlabeled reliable samples in the instrument joints localization process, and uses the classification statistics in the recovery process. In normal Hough forest, both classification and voting (regression) processes are trained using the same features samples. Each of these samples is associated with a class label. At testing time, each non-background classified pixel casts a vote. Having a large number of misclassified pixels creates a noise in the voting map and has the potential to generate a global maximum of voting at incorrect joints positions. The contribution of this work is to motivate the majority of the classified pixels to cast reliable votes to the right joints. This can be achieved by training the voting process of Hough forest in different way from the classification process. In the votes training process, samples in the neighborhood of the ground truth points are incorporated since they can cast reliable votes for the correct instrument joints coordinates even though they are not associated with any class label. The features of these samples are associated with different displacements votes to the ground truth coordinates. Therefore, at testing time, similar features can cast reliable votes regardless of the assigned label. This has the potential to create a global maximum of votes at each instrument joint coordinates and minimize the impact of non-reliable voters. Moreover, the proposed algorithm aims to remove the need of manual reinitialization after any tracking failure while being able to localize not only the forceps connecting point but also its two tips.

Next, we present the basics of Hough forests, then the details of our proposed framework of the classification and regression processes to learn Hough votes is presented. Finally, we show the results on retinal and laparoscopic datasets and compare the performance of our approach to the state-of-the-art methods on publicly available retinal dataset.

## 7.2 Hough Forest

Hough forests are special type of random forest, in which the trees are constructed from a set of patches  $\{P_i = \{I_i, c_i, v_i\}\}$ , where  $I_i$  is the appearance of the patch or the features extracted from it,  $c_i$  is the class label associated with the patch, and  $v_i$  is the patch offset or the vote of that patch for a certain object in the image [44]. The patches are collected from labeled dataset, where the class label is set to zero for the background patches, and non-zero for object patches. During training, each leaf node stores a proportion of the object patches arrived to this node and a list of their corresponding offsets. At test time, these offsets are used as votes to the object position in the Hough map. Therefore, this map is treated in a probabilistic manner to find the peak votes values which indicate the location of the target object in the original image. Hough forest presents a powerful tool for object detection and localization such as human detection [44, 43]. In these applications, the target object has sufficient structural information extracted from various parts to cast votes for the centroid of that object. However, for forceps joints localization, only limited amount of samples can cast reliable votes to the joints coordinates. Moreover, the forceps joints need to be detected in a very delicate microsurgery where the accuracy is a crucial issue. To get higher detection accuracy, the training is performed in a heuristic way by associating selective samples from the joints vicinity with different votes.

## 7.3 Proposed Method

The ultimate goal of our proposed approach is to predict the instrument's joints coordinates which represent the instrument pose in 2D image space. These joints are the two tips and the connecting point. The motivation of the proposed pipeline in Figure. 7.1 is based on the following observations:

1. In medical applications with strong illumination changes, intensity-based tracking or detection tends to fail. Structural information on the other hand provides more reliable features to track. Therefore, we don't rely on any intensity-based features for both classification and voting.
2. Most of the false positive detections of instrument joints are occurring around the true positive ones. Hence, more reliable predictions can be extracted if these false detections are involved correctly in the voting process. To achieve this, features of some points sampled from the regions expected to have those detections are employed in the voting training.

Therefore, our approach splits the training of the classification and voting processes into two phases. This enables us to augment the dataset with more variations for robust voting training. Moreover, the statistics of the classification process are utilized for triggering automatically the recovery process. The proposed method starts by classifying image pixels to detect the most reliable points within a Region of Interest (ROI) of the image. The detected points are used to cast votes to the instrument joints coordinates. The final pose estimation is the aggregation of votes on three 2D Hough voting maps, one for each joint. The details for each step will be given in the next sections.

### 7.3.1 Forceps Joint Classification

The goal of this step is to classify the pixels in an image as either eligible to vote for the instrument joints coordinates or non-eligible. The eligibility of each point is estimated based on the features extracted from a patch centered at that point. These features should also be similar to one of the instrument joints' features. Since the instrument parts are the most reliable structures to vote for their locations, the eligible pixels are considered the ones classified as instrument joints pixels, while the non-eligible ones are the background pixels which include vessels, instrument shaft, and other non-relevant structures. As shown in Fig. 7.1, a random forest [25] is used as a multiclass classifier, which is trained from samples  $\{P_i = \{x_i, c_i\}\}$  where  $x_i$  denoting the features extracted from a gray-scale image patch around any of the instrument joints along with the corresponding class label  $c_i \in \{1, 2, \dots, C\}$ .  $C$  is the number of classes used in training. In our method, we are interested in forceps instrument used in membrane peeling operations, and we chose the class labels to be left joint (tip), right joint (tip), connecting joint, and the background. The features extracted from each patch are Histogram of Oriented Gradients (HOG) [33] features. It is worth to note that only three points are annotated in each image as shown in Figure. 7.1, while all other pixels are considered theoretically background pixels. Therefore, to handle unbalanced data in the training phase, we choose random

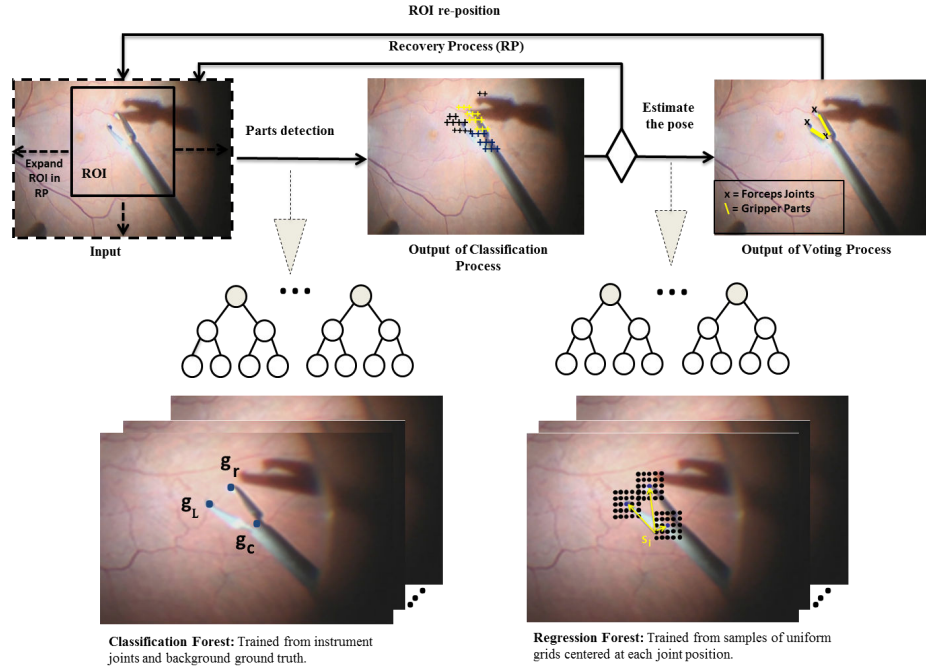


Figure 7.1: The whole pipeline of the proposed method.

samples for the background patches such that the numbers of foreground and background samples are equal. In the testing process, only pixels from the ROI are tested to be assigned class labels. The output is reliable points of the classification process which are classified as non-background points. Moreover, the reliable points start voting directly using the same extracted features as explained in the next section.

### 7.3.2 Pose Estimation

Given an input  $Z$  which is a set of points  $Z : \{z_i \in R^2, i = 1, 2, \dots, k\}$  representing the reliable detections coordinates after the classification process, where  $k$  is the number of these detections, the output of the pose estimation is  $O = \{J_i : J_i \in R^2, i = 1, 2, 3\}$  which represents the three coordinates of the instrument joints. To get more accurate votes, the training of the pose estimation is done based on uniform sampling of features from the vicinity of the instrument joints.

#### Training samples generation

Based on the observation that most of the false positive detections are around the true positives, a regression forest is trained from the ground truth annotated joints and other unlabeled nearby points to increase the reliability of voting in the pose estimation process. Given an image  $I$  from the training set along with its annotation vector  $\mathbf{g} = \langle \mathbf{g}_l, \mathbf{g}_r, \mathbf{g}_c \rangle$  where the vector elements represent the

2D coordinates of the left, right, and center joints respectively, we define three regular grids of  $n$  sampling coordinates  $\mathbf{S} = \{\mathbf{s}_i\}_{i=1}^n$ ,  $\mathbf{s}_i \in \mathbb{R}^2$ . Each grid is centered at one element of  $\mathbf{g}$  as shown in Fig. 7.1. A sample  $\mathbf{s}_i$  from the grids is described by HOG features which denoted as  $\mathbf{F}_i$  and extracted from a patch centered at the coordinates of that sample. The regression forest is trained from all the three grids samples, where the features at each sample  $\mathbf{s}_i$  is associated with a three dimensional vote vector  $\mathbf{v}_i = \mathbf{s}_i - \mathbf{g} = \langle \mathbf{s}_i - \mathbf{g}_l, \mathbf{s}_i - \mathbf{g}_r, \mathbf{s}_i - \mathbf{g}_c \rangle$ . This vector is expressed as a displacement vector connecting  $\mathbf{s}_i$  to the three elements of the annotation vector  $\mathbf{g}$ . The training data extracted from  $m$  images are combined to create the data samples  $D = \{\mathbf{F}_i, \mathbf{v}_i\}_{i=1}^{n \times m}$  which form the input to the regression forest. It is obvious that the sampled coordinates are located in a region where many instrument joints detections are expected to occur there. However, these detections can still vote accurately based on their features. Moreover, the samples are collected from images having instruments with different scales, rotations, opening degrees and lighting conditions.

### Voting Training

During training, each internal node of the trees selects the feature which maximizes the Information Gain (IG), and splits the samples set  $D_p$  arriving to the parent node of each tree into two subsets  $D_l$  and  $D_r$  that goes to the left, and right child respectively. The Information Gain (IG) is obtained as:

$$IG(D_p, D_l, D_r) = H(D_p) - \sum_{k \in \{l, r\}} \frac{|D_k|}{|D_p|} H(D_k) \quad (7.1)$$

where  $H(D_k)$  is the entropy of all  $\mathbf{v}_i$  in  $D_k$ , and is obtained based on votes uncertainty as:

$$H(D_k) = \sum_{j \in \{l, r, c\}} \sum_{i=1}^{|D_k|} \|\mathbf{v}_{i,j} - \mu_j\|^2 \quad (7.2)$$

where  $|D_k|$  is the number of elements in  $D_k$ ,  $\mathbf{v}_{i,j}$  is the value of the  $j^{th}$  dimension of  $\mathbf{v}_i$ , and  $\mu_j$  is the mean of all  $\mathbf{v}_{i,j}$  in  $D_k$ . Consequently, the node stores the feature which maximizes the information gain, associated with the threshold used for splitting. The tree continuously splits the samples and grows down until at least one of the following stopping criteria is satisfied:

1. the maximum depth of the tree is reached;
2. the number of samples  $|D_p|$  is insufficient for further splitting of the data;
3. the information gain of the best split is too small;
4. the samples in  $|D_p|$  cast homogeneous votes which is measured using the variance of votes.

In this case, the node is considered as a leaf node and stores the mean and the standard deviation of each dimension of all  $\mathbf{v}_i$  that reached this node.

### Votes Accumulation

At testing time, given an image  $I$ , we start by classifying its pixels to either instrument joints, or background pixels. Starting from the root node of each tree, a non-background pixel  $\hat{\mathbf{s}}_i$  is tested again based on its already extracted features from a patch around it to cast a vote. The final vote  $\hat{\mathbf{v}}_i$  associated with a patch centered at coordinate  $\hat{\mathbf{s}}_i$  in the image is aggregated by taking the 20 percent of the predictions with the lowest standard deviations from all trees, and averaging the corresponding learned means. The pose estimated by the vote  $\hat{\mathbf{v}}_i$  is  $\hat{\mathbf{g}} = \langle \hat{\mathbf{g}}_\ell, \hat{\mathbf{g}}_r, \hat{\mathbf{g}}_c \rangle = \hat{\mathbf{s}}_i - \hat{\mathbf{v}}_i$ , which is a three dimensional vector, one for each instrument joint. Therefore, three 2D Hough vote maps are used to accumulate the votes from all patches classified as an instrument part in the image. Each Hough map is considered as an integral image, and divided into cells, where each cell accumulates the votes within it. The cell with maximum number of votes is considered the instrument joint. Hence, the final pose estimation is the 2D coordinates of the maximum cells from the three Hough maps.

## 7.4 Tracking and Recovery

At testing time, the ROI is set to the whole image at the first frame. Once the instrument pose is estimated in an image, the ROI shrinks to a small size (i.e. one fourth of the image size) around the instrument center point. In classification phase, only pixels within the ROI are tested. The tested pixels are sampled from a grid with small spacing between grid points. These spacing specify the density of pixels sampling during testing. The number of pixels classified as an instrument joints is maintained for each joint. If any joint is missing in the classification process, the recovery process is triggered automatically. Joints are missing in either:

1. poor image quality due to image blurring or fast motion,
2. partial or fully occlusions by light pipe or microscope lens, or
3. non-existence of the instrument in the image.

Those cases occur very rarely during retinal peeling operation. However, our method can recognize these cases and automatically launch recovery process by gradually expanding the ROI with higher grid spacing. Increasing the grid spacing allows sampling of fewer pixels in order to reduce feature extraction time and hence satisfy the real time requirement. The grid spacing size is set to 2 pixels initially, and increased to 3 in the recovery process.



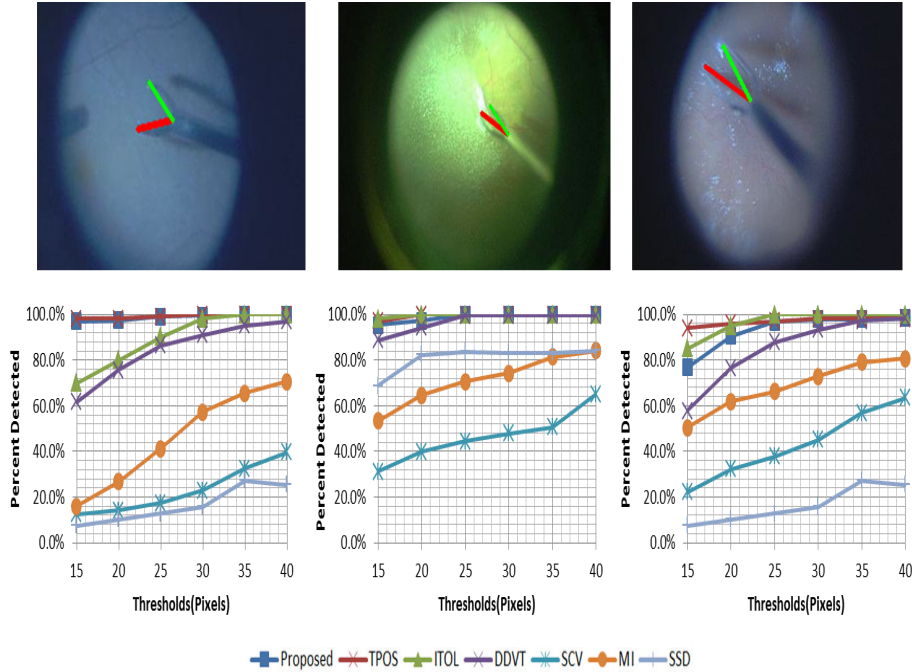


Figure 7.2: Accuracy Thresholds performance: The results for the three sequences when each trained from half of the images and the testing was done on the second half.

## 7.5 Experiments and Results

The proposed approach has been evaluated on two Retinal Microsurgery (RM) datasets. The first is a publicly available one [105], and the second is referred to as Zeiss dataset. The performance of the algorithm was evaluated using two different metrics. The first is the accuracy threshold score defined by Sznitmann et al. [105] which is a pixel-wise measure of the quality of the joints predictions in terms of mean square error. The second metric is the strict Percentage of Correct Parts (strict PCP) [39] which is a quality measure of the prediction of a part of an articulated object. The algorithm is implemented in C++ and runs at 18-fps on a normal Core-*i7* personal computer. For the classification forest, we use 50 trees with maximum depth of 25, while for pose estimation we use 20 trees with maximum depth of 30. The HOG features bin size is 9, and the patch size is 50x50 pixels. Additionally, the integral images are used for fast feature extraction in training and testing. Each grid size, which used in sampling coordinates during votes training, is 15x15 pixels. Additionally, the minimum number of samples  $|D_p|$  is set to 25, the minimum gain is 0.001, and the homogeneity of the votes is computed based on the standard deviation of the votes.

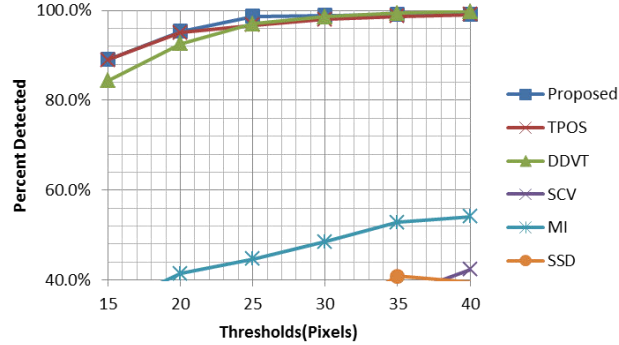


Figure 7.3: Accuracy Thresholds performance: The results for the three sequences when trained from the first halves of the images together and tested on the second halves.

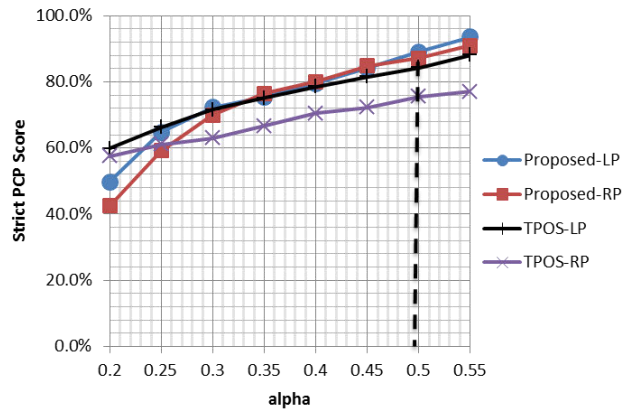


Figure 7.4: Strict PCP scores for the full dataset using our proposed method and TPOS.

### 7.5.1 Public Dataset

The public dataset has three different sequences of in-vivo vitreoretinal surgeries. It comprises 1171 images with resolution of 640x480 pixels. The sequences are different in terms of the lighting conditions, presence of the shadows, and the amount of noise in the background. We compared the performance of our method on these sequences to the state-of-the-art methods TPOS [88], ITOL [67], DDVT [105], SCV [81], MI [12], and SSD. For fair of comparison, the same setup as in the state-of-the-art methods has been followed in training and testing. We use the accuracy threshold to measure the prediction accuracy of the center point with thresholds between 15 and 40 pixels. Our method is evaluated firstly on each sequence separately by training on the first half of each sequence and testing on the second half of the same sequence. From the results in Figure. 7.2, we see in more than 96% of the images the center point is detected with error less than 20 pixels. Hence, this method demonstrates

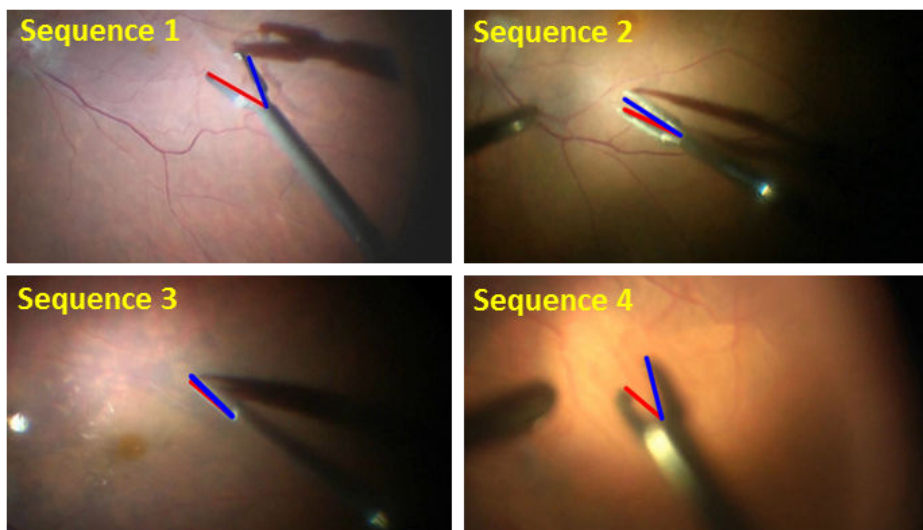


Figure 7.5: Four sequences from Zeiss dataset with different instrument types and different light conditions.

comparable results to the state-of-the-art methods with the advantage that no manual initialization is needed. Secondly, we evaluate on the full dataset by training on the first halves of all sequences, and testing on the second halves. In this case, the regression forest is trained from more reliable features extracted from the joints neighborhood. Hence, our method performs slightly better than the other methods and achieved high prediction accuracy where in 95% of the full dataset images the center point was detected with error less than 20 pixels as shown in Figure. 7.3. In Figure. 7.4, we compare the performance of our method with TPOS [88] using strict PCP score, by training on the first halves of the full dataset and testing on the second halves. The strict PCP scores for detecting the left (LP) and right (RP) gripper parts of the forceps are shown in Figure. 7.4, where at  $\alpha = 0.5$  (which used in human pose estimation comparisons) our approach achieves scores of 89%, and 87% for the left and right parts, respectively, while TPOS achieved 84%, and 75% for the same parts. Therefore, in term of the strict PCP scores, our method outperforms the state-of-the-art method TPOS on the full dataset, which goes in line with the generalization ability and robustness of the proposed method.

### 7.5.2 Zeiss Dataset

This dataset has four sequences of fully annotated images. The images are acquired by a Carl-Zeiss Lumera 700 operating microscope with a resolution of  $1920 \times 1080$  pixels at 25 fps scans. Each sequence has 600 images taken from real in-vivo surgeries, where we use only 200 images of each sequence for training, and test on the remaining. The images resolution was downsampled to one fourth of the original size to reduce time complexity and achieve real time performance. The images include forceps instruments with different types. They are taken in different lighting conditions and with various microscope

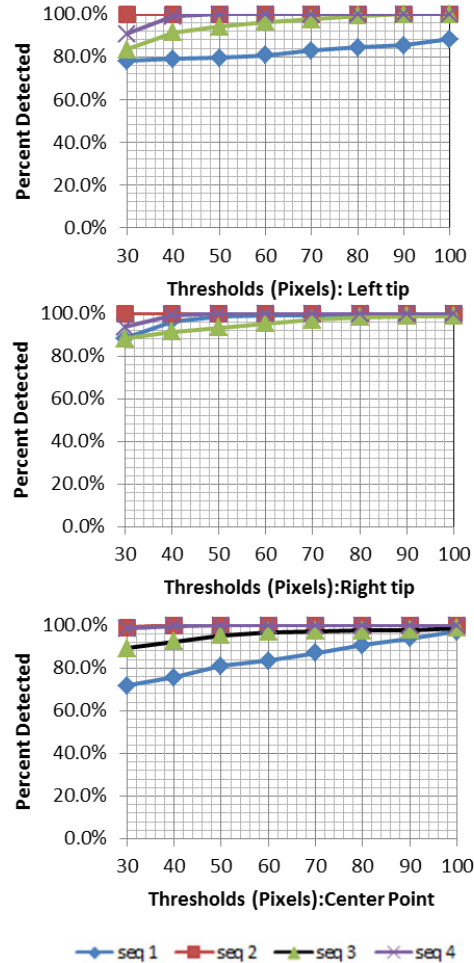


Figure 7.6: Accuracy Thresholds performance: The results of the four sequences when each was trained from half of the images and tested on the second half.

zooming factors. Additionally, they contain more challenging issues like the clear appearance of the blood vessels and instrument shadow as shown in Figure. 7.5. Since the instrument diameter is 50 pixels, we evaluate our approach with accuracy thresholds in the range from 30 to 100 pixels. The evaluation is done firstly on separate sequences, by training on only the first 200 samples, and testing on the remaining 400 images from the same sequence. The results in Figure. 7.6 show the percentage of correct predictions of the left, right, and center joints for each sequence separately. Then we evaluate using the full dataset as we did in the public dataset and the accuracy thresholds scores for each joint are depicted in Figure. 7.7. The method shows its ability to work robustly on the full datasets as well as on separate sequences. For generalizability purpose, we use the model of the full dataset to test our method performance on two unseen sequences, shown in Figure. 7.8. No sample of these sequences is included in the training and they are taken from different surgeries. Each of these sequences has 400 images. The accuracy threshold scores are shown in Figure.

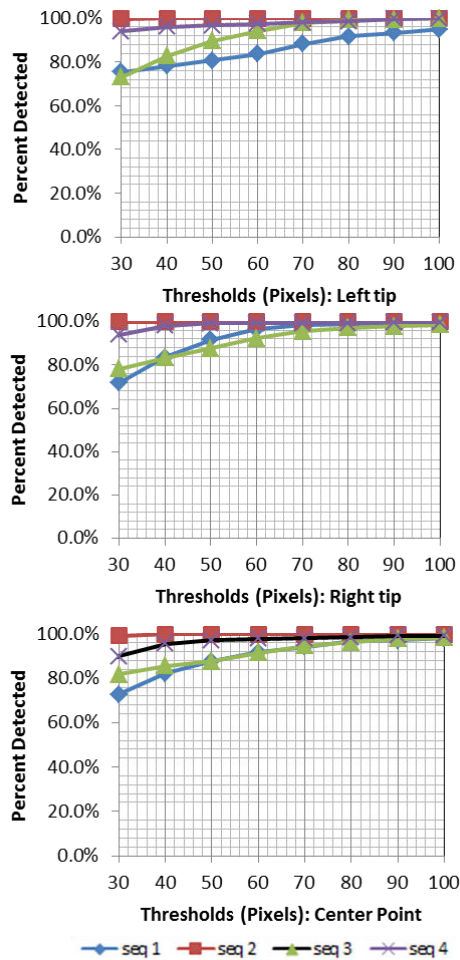


Figure 7.7: Accuracy Thresholds performance: The results for the four sequences when trained from the first halves of the images together and tested on the second halves.

7.9, and Figure. 7.10, which show high detection accuracy for most of the joints. However, for sequence 6, due to high light reflection on the right tip, its detection accuracy slightly decreased. The strict PCP scores for the full dataset is shown in Figure. 7.11. The left and right parts of the forceps are detected correctly at  $\alpha = 0.5$  in 92% of the images.

### 7.5.3 Laparoscopic Dataset

Finally, we show that our approach can be applied as well for laparoscopic dataset. This dataset comprises 1000 images and available on YouTube<sup>1</sup>. In this dataset, two forceps instruments are used to perform the surgery, where one of them is just for fixation and with static pose while the other does the peeling

<sup>1</sup><https://www.youtube.com/watch?v=IVp1sgjQ5To>

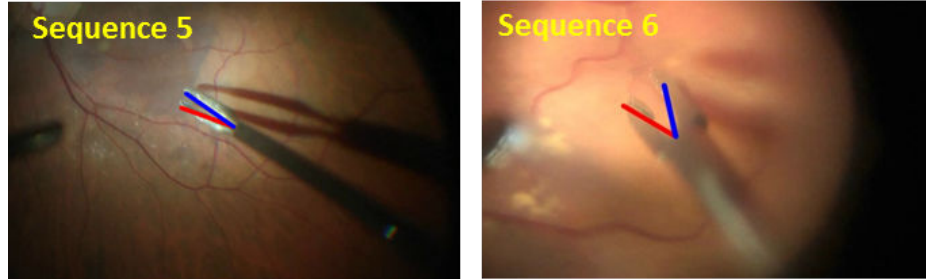


Figure 7.8: Two unseen sequences from Zeiss Dataset, each sequence was taken from different surgery.

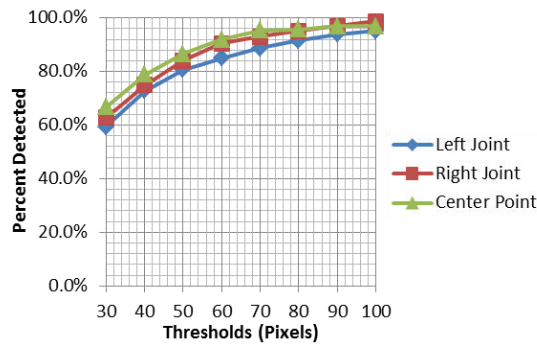


Figure 7.9: Threshold accuracy for detecting left, right, and center points of the instrument in sequence 5

operation. To evaluate our approach on this dataset, the first half of the images are used for training, while the second half for testing. The challenges in this dataset are mainly the extreme changes in the instrument structure which were not seen in the training images. Moreover, the background is more cluttered with other structures. The evaluation is performed on the non-static forceps using also accuracy thresholds and strict PCP with different values of  $\alpha$ . The threshold values are chosen from 15 to 40 pixels since the forceps shaft width in this dataset is 20 pixels. Figure. 7.12 (b) shows a good performance of our method to detect each joint and estimate the final pose. The results show that in more than 70% of the images, the tips are detected with error less than 20 pixels. Detecting the center point of this forceps is more challenging due to the severe changes and the expandable nature of the joint point. However, in 60% of the images, it is detected with error less than 20 pixels. The strict PCP scores are shown in Figure. 7.12 (c) for the left and right gripper parts of the forceps.

## 7.6 Discussion

The proposed method demonstrates promising results in retinal microsurgery to detect not only the connecting point of the forceps, but also its two tips. The accuracy of this method has been obtained using few



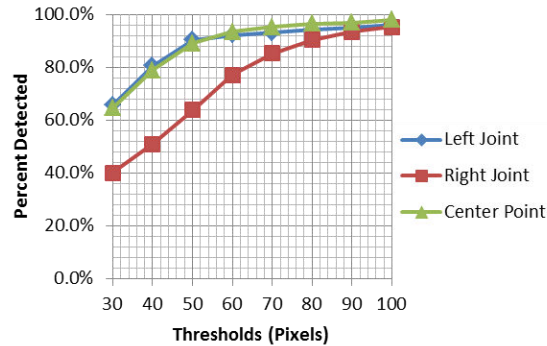


Figure 7.10: Threshold accuracy for detecting left, right, and center points of the instrument in sequence 6

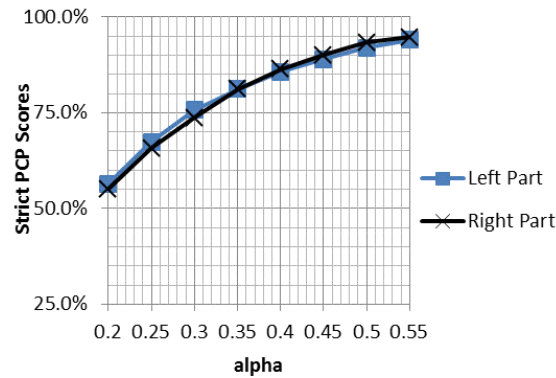


Figure 7.11: Strict PCP scores for the full Zeiss dataset.

samples for training, and relying only on structural features. Unlike other works done in this field, our method uses only one third of the data for training, and the performance is shown on the other two thirds in addition to new unseen sequences. Evaluating in this way is more convenient from the clinician’s perspective who expects to use the trained model for other new and longer surgeries. According to the availability of the ground truth annotations, we used 600 images from each sequence for quantitative evaluation. However, the approach runs on longer sequences with the same performance. Moreover, the method is able to recover automatically in cases of low quality images or missing of any forceps joint during the classification process. This implies that our approach can handle outliers cases like fast motion of the instrument, and working very close to the boundary of the image. In these cases, the recovery process might result in some inaccurate predictions to handle these outliers instead of interrupting tracking and initialize manually to start over. This is why the methods TPOS [88] and ITOL [67] perform slightly better on the third sequence of the public dataset in which the instrument goes in and out at least 2 times. These methods don’t have automatic recovery scheme and resort to the manual initialization to handle like this situation while our algorithms can cope

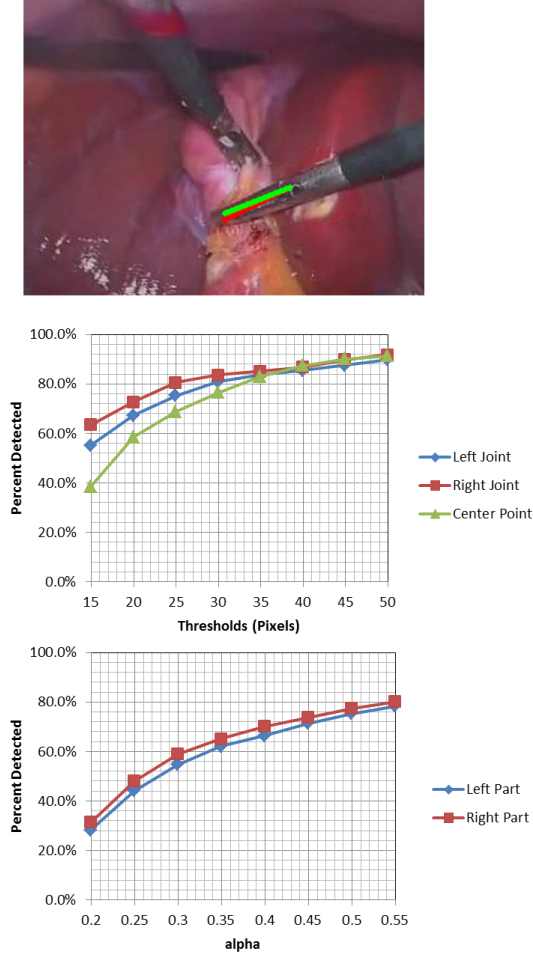


Figure 7.12: (a) a qualitative example of the estimated pose for laparoscopic dataset, (b) pixel-wise accuracy of predictions for each of the three forceps joints, (c) strict PCP scores for left and right gripper parts predictions.

with it at the expense of few inaccurate detections. For the same reason, in term of the PCP scores, TPOS [88] performs better at  $\alpha = 0.20$  while for most other values of  $\alpha$  our method outperforms theirs. We compared with only TPOS, since they are the only ones who predict the joints coordinates during tracking. The setup of the retinal microsurgery allows the use of only one instrument during the operation. However, our method can be easily extended to detect and track multiple instruments without incurring extra computational time. This can be achieved by processing the resultant Hough maps to localize  $K$  maxima in each. Processing the Hough maps in this way can be seen as a post processing step and it doesn't change the main pipeline.



## 7.7 Conclusions

A new approach is presented in this paper to detect, track, and estimate the pose of forceps instrument. The algorithm exploits reliable unlabeled samples in the voting training process of the Hough forest, and selects reliable voters to predict the instrument joints coordinates during testing time. In this way, the algorithm reinforces the creation of global maximum of voting close to the joints coordinates. Moreover, the new implementation allows the benefit of classification statistics in automatic triggering of the recovery process. The approach demonstrates the ability to generalize to unseen and long sequences of in-vivo surgery and runs at real time performance. Comparing to the state-of-the-art methods, our method shows comparable results with no need of manual reinitialization.

CHAPTER 7. INSTRUMENT POSE ESTIMATION BASED ON RELIABLE  
HOUGH VOTING

---

## Chapter 8

# CRF-Based Model for Forceps Pose Estimation

### 8.1 Introduction

In the last chapter, the instrument pose is defined as the 2D coordinates of the two tips of the left and right gripper's parts of the forceps and its center joint coordinates. In many applications, the orientation of the instrument shaft plays an important role to direct the OCT B-scans in a certain direction. In the work of Hessam et al. [91], an OCT scan along the instrument shaft has been employed for augmenting the scene with depth information of the instrument tip. Having reliable information about the shaft orientation can be exploited in numerous applications such as augmented reality and actions understanding. Moreover, it optimizes the positioning of the OCT B-scans for the full benefit of OCT imaging. Estimating instrument orientation could be integrated with Hough voting algorithm developed in the last chapter to localize instrument shaft. However, it requires adding new Hough map to accumulate the votes for shaft as well as it would allow the shaft candidates detections to vote for the other instrument parts. According to the homogeneity and low structural variations along the instrument shaft, the involvement of shaft part in the Hough voting process would create a lot of noisy votes and negatively influence the other Hough maps for the other joints. This is simply due to the high similarity existing among the shaft samples features. These samples need to be associated with different displacement vectors to be involved reliably in the voting process. Associating similar features with different output vectors would be a source of confusion in supervised machine learning. Therefore, shaft's samples have been excluded from the training process in order not to involve non-reliable voters which might deteriorate the detection accuracy of the other joints. Subsequently, the shaft orientation would not be estimated using this voting scheme. This motivates us to step up our efforts to include the instrument orientation in the pose estimation while maintaining the accuracy of predicting the other three instrument joints as done in the Hough voting in the last chapter. Hence, it leads us to model the instrument detection, tracking and pose estimation in a different way. The new solution in this chapter models the pose

estimation problem as a Conditional Random Field (CRF). Our contribution is to consider the relations and dependencies among the instrument parts to impose kinematic constraints on the parts detections hypotheses. Different modules are used to model these relations which are regarded as our prior information of the instrument structure. Therefore, the algorithm maintains the part-based detection as in the Hough voting and integrates it with the prior information modules within the CRF framework. The modules are implemented to capture the translation, rotation, and scale changes among the parts.

In this chapter, we start with the problem formulation which gives the big picture of our designed CRF model. The basic designed modules are explained next. Then, we show how to use all components together in the inference process using genetic algorithms to estimate the final pose estimation. Finally, we show the performance of our algorithm on retinal and laparoscopic images.

## 8.2 Problem Formulation

In this work, medical instrument is modelled as a multi-part articulated object where each part can be detected separately. Depending on the used features, parts detections using most of machine learning classifiers can result in a large number of false detections especially for structure-less objects like medical instruments. However, these detections, including the true positive ones, form a new and reduced search space within the 2D image space which represents instrument part's hypotheses space. Therefore, the sought targets are just specific instrument part detections within the reduced space, such that these detected parts would represent the instrument pose. Prior information about the instrument parts and the relations between them are integrated on top of these detections together in one model in order to filter out the vast majority of false detections and to end up with the optimal instrument configuration. Prior instrument information can include the relative lengths of the parts, the angles between them, the gripper length, the possible movements of the joint, the possible changes of the current state, ... etc. Given different prior information models which expressed as probabilistic distributions and different potential instrument configurations, then the ultimate goal of our approach is to optimize for the best configuration (instrument pose as shown in Figure 8.1. (Left)) which maximizes the likelihood of the distributions of the prior models. To that end, the instrument in our method is modeled as a CRF of  $n$  random variables, and the factor graph of this model is shown in Figure 8.1.(Right). Each random variable  $Y_i$  corresponds to an instrument part, and the edges among these variables denote conditional dependence of the parts which can be described as a physical constraint. The instrument pose is given by the configuration  $Y = (Y_1, Y_2, \dots, Y_n)$  where the state for each variable  $Y_i \in \Lambda_i$  represents the 2D position of the instrument part, and is taken from the discrete space  $\Lambda_i \subset R^2$ . Consider an instance of the observation  $x \in X$  that corresponds to instrument parts features, a reference pose  $P$  and an instrument configuration  $y \in Y$ , the posterior is defined

as:

$$\begin{aligned}
 p(y|x, P) = & \frac{1}{Z(x, P)} \prod_i^n \Phi_i^{Conf}(y_i, x) \cdot \Phi_i^{Temp}(y_i, P_i) \cdot \prod_{(i,j)} \Psi^{Temp}(y_i, y_j, P_i, P_j) \\
 & \cdot \prod_{(i,j) \in E_{Trans}} \Psi^{Conn}(y_i, y_j) \cdot \prod_{(i,j,k) \in E_{RLen}} \Psi^{RLen}(y_i, y_j, y_k) \\
 & \cdot \prod_{(i,j,k) \in E_{Cons}} \Psi^{Cons}(y_i, y_j, y_k) \cdot \prod_{(i,j,k,l) \in E_{Rot}} \Psi^{Rot}(y_i, y_j, y_k, y_l) \quad (8.1)
 \end{aligned}$$

where  $Z(x, P)$  is the partition function, and  $\Phi^{Conf}(y_i, x)$  is the unary score function.  $E_{Trans}$ ,  $E_{RLen}$ ,  $E_{Cons}$ , and  $E_{Rot}$  are the graph edges that model the kinematic constraints among the instrument parts using different potentials functions.  $\Psi^{Conn}$  is a binary potentials functions to model the distances changes among the forceps gripper's end points based on the connectivity between the forceps center point and each of the tips.  $\Psi^{RLen}$ , and  $\Psi^{Cons}$  are ternary potentials functions to ensure consistency in the relative length of the left and right parts of the gripper, and whether they can be bounded by a small region in the image. The rotation potential function  $\Psi^{Rot}$  is defined to estimate the configuration likelihood based on the distribution describing the proper angles among the instrument parts. Once the forceps hypothetical parts are detected, different configurations from these hypotheses within a defined Region of Interest (ROI) are evaluated with the potential functions to select one configuration. This configuration is the one maximizing the posterior given in eq. 8.1 and it represents the forceps pose.

Next subsections, we present the unary potential which used to define some probable coordinates for instrument parts, followed by different types of potential functions to impose kinematic constraints on the instrument parts and represent our prior model of the instrument.

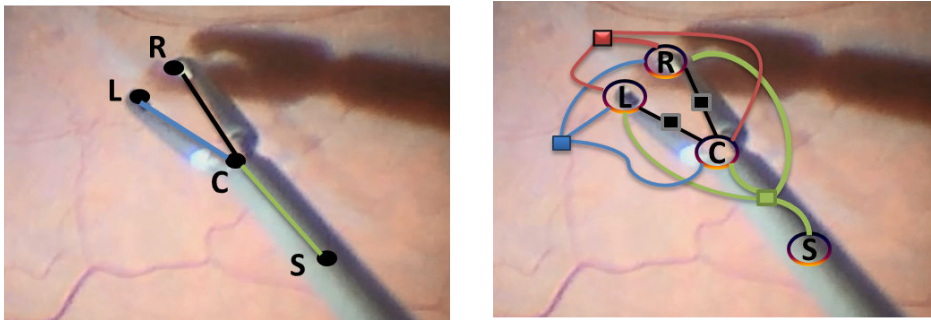


Figure 8.1: (Left) Target pose estimation, (Right) The factor graph for the Forceps: 4 variables (left (L), right (R), center (C), and shaft (S)) are used with different types of constraints are presented with different edge colors: black (translation), green (rotation), red (relative length), and blue (consistency)

### 8.2.1 Unary Potentials

The unary potential functions are designed to give a score for each instrument part hypothesis. Each hypothesis has a confidence value which is a probability assigned to the pixel in 2D images to express its degree of belonging to a specific instrument part. A regression forest is trained on histogram of oriented gradients (HOG) features for this purpose and regarded as a multiclass detector. The output of the regression forest is a class label prediction for each hypothesis and a confidence value. The number of class labels is set to the number of random variables in the CRF plus one for the background. The confidence value for each instrument part hypothesis is defined in eq.8.2.

$$\Phi^{Conf}(y_i, x) = \frac{1}{T} \sum_{j=1}^T \pi_j(x) \quad (8.2)$$

where  $T$  is the number of trees in the forest,  $\pi_j(x)$  is the probability assigned by one tree to  $y_i$  to express its belonging to a specific instrument part. The probability is given based on testing the features  $x$  associated with  $y_i$ . The term  $\Phi^{Temp}(y_i, P_i)$  favors joints hypotheses which are close to the last inferred joint  $P_i$  based on the spatial distance between them, as given by eq.8.3.

$$\Phi^{Temp}(y_i, P_i) = e^{\frac{-\|y_i - P_i\|_2^2}{2}} \quad (8.3)$$

Moreover, the temporal information is maintained for the orientation of each instrument part. Each part is defined by its two end joints, and eq.8.4 is included to penalize large changes in the part orientation in two consecutive frames.

$$\Psi^{Temp}(y_i, y_j, P_i, P_j) = e^{\frac{-\|\alpha(y_i, y_j) - \alpha(P_i, P_j)\|_2^2}{2}} \quad (8.4)$$

where  $\alpha(y_i, y_j)$  is the angle formed by the vector  $\vec{y}_i - \vec{y}_j$  with  $x$ -axis of the coordinates system.

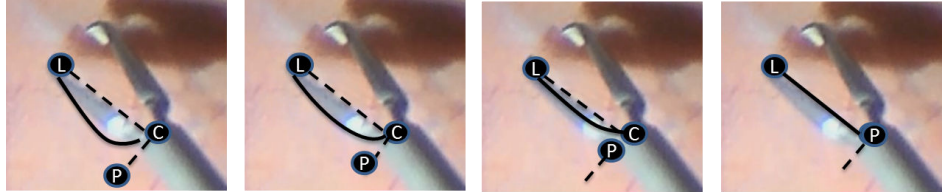


Figure 8.2: Connectivity modeling using Bézier curves where the dashed lines are orthogonal vectors and the position of the control point  $p$  is placed along one of those vectors with different displacements from the center point.

### 8.2.2 Binary Translation potentials

The distance between the tips and the center point changes at different scales and orientations. The translation potentials model these translations of the left

and the right tips to the center point by measuring the connectivity between the hypotheses of the instrument parts involved in the translational edges as shown in Figure.8.1(Right). For example, given one hypothesis  $y_i$  of the left part and one hypothesis  $y_j$  of the center part detections, the connectivity between them is computed along different quadratic Bézier curves controlled by the position of the control point  $P \in R^2$ , as shown in Figure.8.2. The curve in each image can be described using:

$$C_{y_i, y_j}^{(P)}(t) = (1-t)^2 y_j + 2(1-t)tP + t^2 y_i, \quad 0 \leq t \leq 1 \quad (8.5)$$

where the control point  $P$  is placed along the orthogonal vector to the vector  $(y_i, y_j)$  to get similar curvatures to most of the available grippers shapes. The displacement of the point  $P$  to  $y_j$  specifies the shape of the curve connecting  $y_i$  and  $y_j$ . By denoting this curve simply as  $C_{y_i, y_j}^{(P)}$ , the probabilistic connectivity along each curve is given by the following equation:

$$Conn(C_{y_i, y_j}^{(P)}) = \frac{1}{k^2} \sum_{j=1}^S |s_j|^2 \quad (8.6)$$

in which  $k$  is a normalization factor. The curve is assumed to consist of  $S \in R$  segments. Each segment  $s_j$  is a connected component of pixels along one curve. The connected components are extracted from the binary image created by thresholding the gradient image of the input microscopic image. The points  $y_i$  and  $y_j$  are overlaid on the binary image and considered strongly connected if at least one of Bézier curves aligned to the gripper edges curvature. This curve might consist of zero (not connected hypotheses where  $C_{y_i}^{y_j}(P)$  is set to  $\epsilon$  for numerical stability), one or many segments. Changing the position of  $P$  by different  $\Delta p$  values enables the algorithm to handle various types of forceps with different curvatures along the gripper. The connectivity measure in eq.8.6 is modeled to favor longer segments and penalize short ones in order to be robust in case of noisy images. The translation potential function keeps the maximum probability among all curves and it is defined in eq.8.7. A higher value of this probability means stronger connectivity, and higher potential of the hypotheses to belong to the gripper end points.

$$\Psi^{Conn}(y_i, y_j) = \max_{\Delta p} (C_{y_i, y_j}^{(P+\Delta p)}) \quad (8.7)$$

The connectivity along the left and right parts of the gripper are calculated in the same way but with different positioning of the control point  $P$ .

### 8.2.3 Ternary Potentials

The relative length function  $\Psi^{RLen}$  is used to model the relative length between the left and right gripper parts as a Gaussian distribution, and is given in eq.8.8. The function is designed to increase the algorithm robustness in case of false detections of structures like vessels near the instrument tips. The model parameters  $\mu_{i,j,k}^{RLen}$  and  $\sigma_{i,j,k}^{RLen}$  are estimated from the ground truth. Moreover, the

gripper length should be consistent with shaft length in the ROI from which the configurations are selected. Hence, the consistency function  $\Psi^{Cons} \in \{1, \epsilon\}$  is modeled to favor selected gripper parts with lengths less than half the size of the ROI side length. Otherwise, the output of the function is a small probability ( $\epsilon$ ) to penalize this configuration. In this way, the inconsistent combinations of parts hypotheses are penalized.

$$\Psi^{RLen}(y_i, y_j, y_k) = \mathcal{N}(|y_i - y_j|, |y_i - y_k| | \mu_{i,j,k}^{RLen}, \sigma_{i,j,k}^{RLen}) \quad (8.8)$$

$y_i$ ,  $y_j$ , and  $y_k$  are center, left, and right hypothesis respectively and they are chosen randomly from a specified ROI to initialize the optimization process as will be explained later.

### 8.2.4 Quaternary Rotation Potential

Any configuration  $y$  of the instrument forms an angles triple  $\theta = \{\theta_i, i = 1, 2, 3\}$  among its parts treated as random variables. The rotation potential in eq.8.9 models the relations between these random variables as a mixture of two multivariate Gaussian distributions. One distribution models the relation among the variables when the instrument is closed or is about to be closed, while the other distribution is for the open instrument with different degrees. The parameters for each distribution (the mean  $\mu_{i,j,k,l}^{R_n}$  and the covariance  $\Sigma_{i,j,k,l}^{R_n}$ ) are estimated from the ground truth, where  $n = 1$  for one distribution and  $n = 2$  for the other.

$$\Psi^{Rot}(y_i, y_j, y_k, y_l) = \sum_{n=1}^2 \mathcal{N}(\theta_{i,j,k,l} | \mu_{i,j,k,l}^{R_n}, \Sigma_{i,j,k,l}^{R_n}) \quad (8.9)$$

$y_i$ ,  $y_j$ ,  $y_k$ , and  $y_l$  are left, center, right and shaft hypothesis respectively.

### 8.2.5 Inference of the Instrument Pose

We used genetic algorithms [92] to infer an approximate solution which maximizes the posterior equation as:

$$\hat{y} = \underset{y}{\operatorname{argmax}} p(y|x, P) \quad (8.10)$$

The most important parts of the genetic algorithms are the representation of the chromosomes and the definition of the fitness function. Each chromosome consists of four genes  $\langle y_i, y_j, y_k, y_l \rangle$  representing one instrument configuration selected from the hypotheses space. The fitness function is set to the posterior function given in eq.1, which depends on the prior models  $p(y)$  of the instrument and the initial hypotheses probabilities given by the regression forest. The inference algorithm which is summarized in Algorithm 8.1 starts by random generation of 1000 configurations  $Y$  which considered the initial population. Among those configurations, the crossover is applied pairwise by interleaving



the genes at specific index to generate more variations from the current population. However, to enable the algorithm skipping local maxima during optimization, mutation operation is employed to replace random genes with others from the neighborhood. New configurations are created and evaluated using the fitness function. The configurations with highest scores are survived to the next generation. The solution is obtained after a fixed number of iterations or no convergence in two successive generations. In this case, the configuration with the highest score represents the instrument pose.

---

**input** :  $Y$ : List of configurations ( $Y^{(j)} : j = 1, \dots, N$ ), each  $Y^{(j)}$  represents one chromosome  
 $P(y)$ : Prior model of the instrument  
 $H$ : Hypotheses probabilities for each configuration  $y_i^j$  in  $Y$   
**output**:  $Y^{(i)}$ : Final configuration or the estimated pose

```

begin
  S for  $i \leftarrow 1$  to  $iterations$  do
    for  $j \leftarrow 1$  to  $Y.size$  do
       $Y^{(ind1)} \leftarrow Y[random\_configuration(\{1 \dots Y.size\})]$ 
       $Y^{(ind2)} \leftarrow Y[random\_configuration(\{1 \dots Y.size\})]$ 
       $crossover\_index \leftarrow random\_position(\{1 \dots Y^j.size\})$ 
       $crossover(Y^{(ind1)}, Y^{(ind2)}, crossover\_index)$ 
       $Mutation(Y^{(ind1)})$ 
       $Mutation(Y^{(ind2)})$ 
       $scores^{(2 \times j)} \leftarrow fitness\_evaluation(Y^{(ind1)}, P(y), H)$ 
       $scores^{(2 \times j + 1)} \leftarrow fitness\_evaluation(Y^{(ind2)}, P(y), H)$ 
       $population \leftarrow population \cup Y^{(ind1)} \cup Y^{(ind2)}$ 
    end
     $Y \leftarrow select\_top\_N\_configurations(Y, population, scores)$ 
     $Y^{(i)} \leftarrow argmax_Y(scores)$ 
    if ( $Y^{(i)} = Y^{(i-1)}$ ) break
  end
end

```

---

**Algorithm 8.1:** Inference Algorithm

Once the pose is estimated in the first frame, a reduced Region of Interest (ROI) is defined around the instrument center point to limit our detection space in the next frames. This ROI is expanded gradually when any instrument part is missing in the unary detections, or when the confidence from the inferred pose is low. Low confidence of the final solution after optimization happens when either: (1) low likelihood of the rotation distributions, or (2) the consistency potential output being small ( $\epsilon$ ). These cases mean either the solution cannot

have the normal forceps shape, or it has been formed from false detections in ROI, which requires the re-initialization to be triggered automatically by expanding the ROI.

### 8.3 Experiments and Results

The experimental validation of the proposed method is carried out on three different microsurgery datasets. The first is Zeiss dataset, which consists of eight sequences of surgeries performed on human eyes with frame resolution of  $1920 \times 1080$  pixels, downsampled to one fourth of the original size. The downsampling is done to reduce the amount of processing and achieve the real time requirements without affecting the detection accuracy. The second dataset is publicly available [105] with 1171 images of  $640 \times 480$  pixels. No downsampling is performed on this dataset. The third dataset is a laparoscopic surgery dataset with 1000 images available on YouTube <sup>1</sup>. The proposed algorithm is evaluated by estimating the pose of one of the instruments present in the laparoscopic surgery since the other instrument has a fixed pose. The performance of the algorithm was evaluated using three different metrics: (1) Accuracy threshold score defined by Sznitmann et al. [105] to measure the pixel-wise detection accuracy for each instrument joint, (2) the strict Percentage of Correct Parts (strict PCP) [39] for gripper parts detection accuracy, and (3) the angular threshold score defined in [6] to measure the accuracy of estimating the shaft's orientation. The algorithm runs at 15-fps for public and laparoscopic datasets and 18-fps for Zeiss datasets on a normal personal computer. For the regression forest 50 trees with maximum depth of 25 are used. The HoG features bin size is set to 9 and the patch size is  $50 \times 50$  pixels.

Table 8.1: Strict PCP scores for  $\alpha = 0.5$  on Zeiss Dataset

Zeiss Seq's	Seq1	Seq2	Seq3	Seq4	Seq5	Seq6	Seq7	Seq8
#Testing Images	590	400	400	400	200	400	200	200
Left PCP	91	99	98	98	92	85	96	75
Right PCP	93	99	99	99	93	94	97	76

#### 8.3.1 Zeiss Dataset

The algorithm has been evaluated on 8 sequences as shown in Figure.8.3, where each sequence is taken from different surgery with different conditions. To achieve maximum reliability in clinical use, only 200 images from the first 4 sequences were used for training. The testing was done on the remaining images from each sequence in addition to 4 other unseen sequences to prove the generalizability of the algorithm. The number of testing images from each dataset is listed in Table 8.1. Each training frame has 4 annotated points: left and right tips, center point and a point on the shaft centerline. 200 samples from

<sup>1</sup><http://www.youtube.com/watch?v=IVp1sgjQ5To>

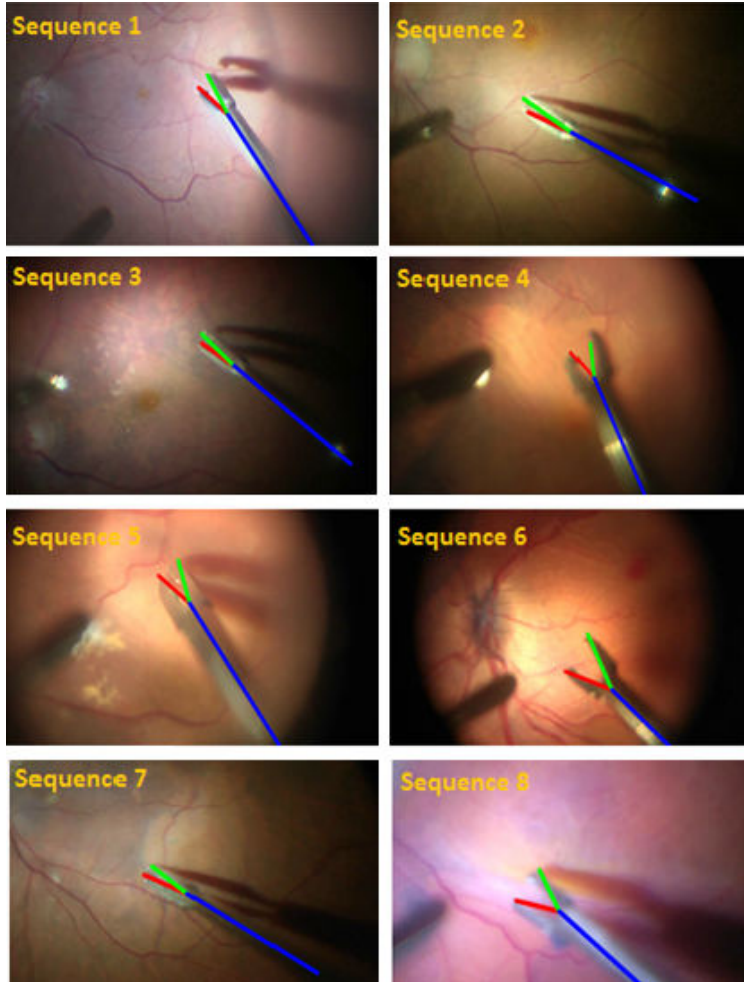


Figure 8.3: Eight samples from each sequence of Zeiss dataset with pose estimation

the training images are manually clustered to open and close states to estimate the parameters of the rotation Gaussian distributions. Since the instrument shaft diameter is 50 pixels, we evaluate using values between 20 and 80 pixels for the accuracy threshold. Figure.8.4 shows the percentage of correctly predicted locations for different joints of the instrument. The results show that in 90% of the testing images the tips are detected with less than 50 pixels (the shaft diameter) error. Moreover, the instrument center point is correctly detected in 85% of the images at the same threshold. The strict PCP scores of the left and right gripper's parts for  $\alpha = 0.5$  (which used for human pose estimation evaluation) for each sequence are depicted in Table 8.1.

The results demonstrate the high accuracy of localizing not only the joints but also the gripper parts, which are correctly detected in more than 90% of the entire dataset. Therefore, this proves the robustness of the algorithm and its ability to generalize to new sequences.

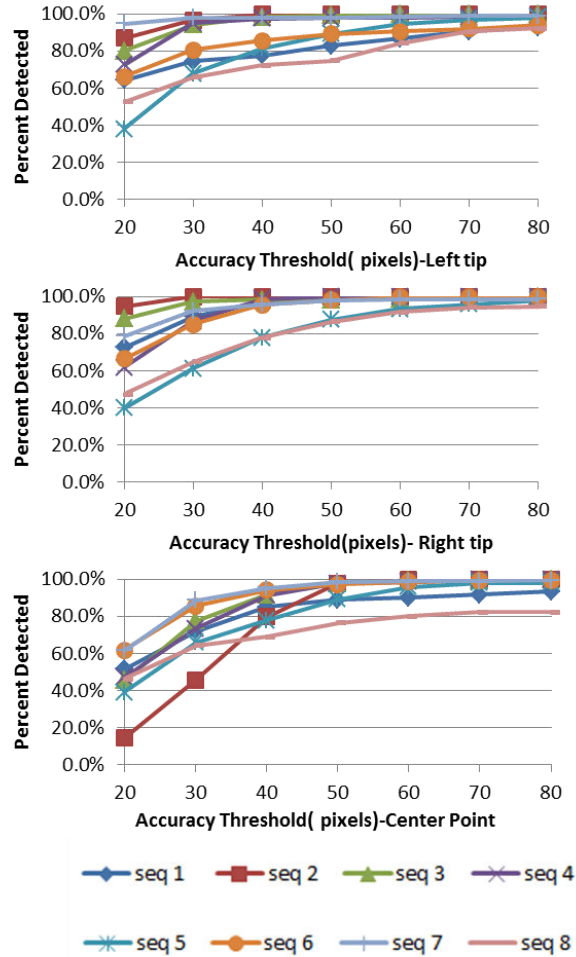


Figure 8.4: The accuracy threshold scores for left, right and center points respectively

Figure. 8.5 shows the performance of our algorithm to estimate the orientation of the shaft while varying the angular threshold from 3 to 24 degrees. It is evident that in about 90% of the images, the orientation is detected with deviation less than 15 degrees.

### 8.3.2 Public Dataset

The proposed method was compared with state-of-the-art methods: MC-15 [88], MC-14 [107], MC-12 [105], SCV [81], MI [12] and SSD. The evaluation includes two sequences of the public datasets. The third sequence is omitted, as in [107], due to its short length which makes it ill-suited for training purposes. In the first experiment, the training is done on the first half of each sequence separately and testing was on the second half. The detection

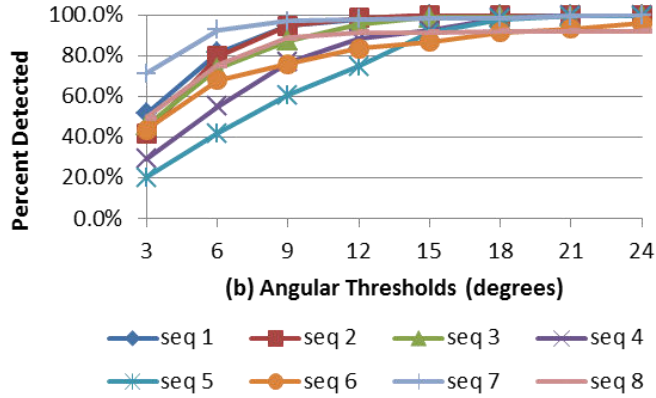


Figure 8.5: Angular Threshold scores for Zeiss sequences.

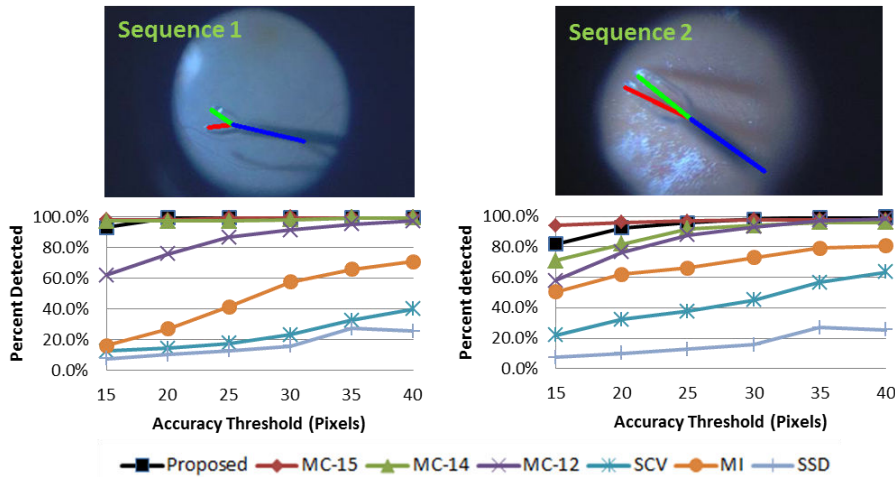


Figure 8.6: Threshold accuracy for each of the public sequences separately

accuracy of the center point is shown in Figure.8.6 which shows comparability of the proposed method to the state-of-the-art methods with the advantage of not requiring manual re-initialization. For example, at threshold of 20 pixels (the shaft diameter), the center point are detected correctly in more than 95% of the images in both cases. PCP scores show that the left gripper part is correctly detected in 97% and 93% of the images in sequence 1 and sequence 2 respectively, and the right part is correctly detected in 95% of the images in both sequences. The comparison in Table 8.2 with state-of-the-art method MC-15 [88], which is the only method that can locate the forceps tips, shows the comparability of our approach with the advantage of no manual re-initialization is required to handle tracking failures. Moreover, the accuracy threshold scores for detecting the two tips of the forceps in each sequence are depicted in Figure.8.8.

In the second experiment, the training is performed on the full dataset (the first two halves of the two sequences together) and the testing is done on the

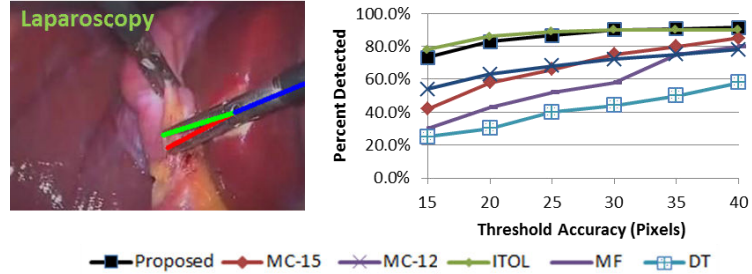


Figure 8.7: Threshold accuracy for laparoscopic dataset

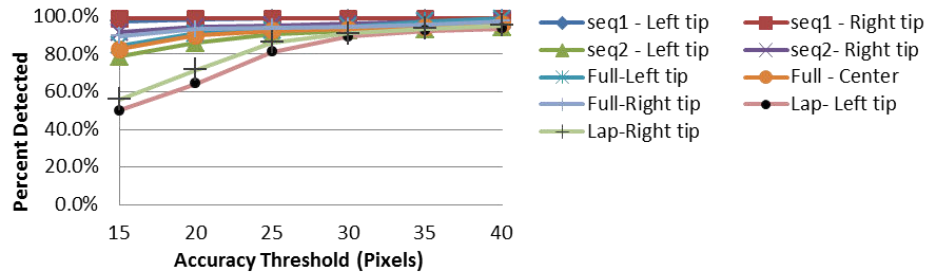


Figure 8.8: Accuracy threshold for different forceps joints of the public (full and separate sequences) and laparoscopic (Lap) datasets.

second halves. The performance of detecting forceps tips and forceps center point is shown in Figure.8.8 labeled with the prefix Full. The left and right gripper parts are correctly detected in 89% of the images of the entire dataset. The strict PCP scores for both experiments are listed in Table 8.2 and compared to MC-15.

### 8.3.3 Laparoscopic Dataset

We compared our performance with MC-15 [89], MC-12 [105], ITOL [67], MF [67] and DT [67]. Similar to these methods, training is done on the first half of the dataset, and the testing on the second half. Comparing the performance of our method in detecting the center point with the other methods using accuracy threshold is shown in Figure.8.7. It is obvious that our method outperforms most state-of-the-art methods, and achieves similar results to ITOL which is mainly an intensity-based tracking method and impractical for live surgery due to the required manual initialization. The accuracy threshold scores of detecting each tip is shown in Figure.8.8 while all other methods can't detect them in this challenging dataset. The PCP scores for detecting left and right gripper parts are 89% and 90% respectively as shown in Table 8.2 which demonstrate high detection accuracy of both gripper's parts .

Figure.8.9 shows the performance of our algorithm to estimate the

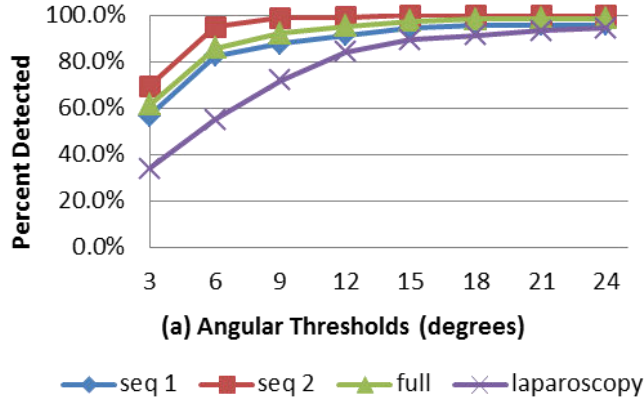


Figure 8.9: Angular Threshold scores for Public and Laparoscopic sequences

 Table 8.2: Strict PCP scores for  $\alpha = 0.5$  on Public and Laparoscopic(Lap) datasets

Public/Lap Seq's	Proposed				MC-15			
	Seq 1	Seq 2	Full	Lap	Seq 1	Seq 2	Full	Lap
Left PCP	97	93	89	89	95	97	N/A	N/A
Right PCP	95	95	89	90	97	95	N/A	N/A

orientation of the shaft for the public and laparoscopic datasets. The results show that the orientation is detected with deviation less than 12 degrees in 90% of the public images and 83% of the laparoscopic images.

## 8.4 Results Discussion

The proposed approach showed high accuracy of instrument joints localization in real time performance. This accuracy is attributed to modelling the dependencies between instrument parts as CRF model, while other methods don't consider these dependencies and rely only on individual parts detection. These dependencies are built on top of random forest outputs trained using only gradient-based (HOG) features to serve as unary detections functions. Unlike other intensity-based tracker methods, relying on HOG features makes our approach robust enough to illumination changes during surgery. Moreover, it reduces the amount of training samples needed for training large changes in instrument appearance. This is why, in the first dataset, our algorithm needs only 200 samples from only 4 sequence and it is able to run on testing images with 3 times the size of the training ones. Furthermore, the algorithm has been evaluated on 4 unseen sequences from different ophthalmic surgeries and its performance proves its capabilities to work on new sequences as well as the performance on the training data. Practically, it can run on even longer sequences since there is no need to train more samples to account for new illumination changes. Moreover, relying on detected structural parts using HOG

features brings new advantage to our method which is being able to sense some confidence signals. This feedback is employed for automatic recovery process, which is missing in most other methods, to localize again the instrument after its disappearance without surgeon's intervention.

The results also presented high PCP scores on most of Zeiss retinal sequences. However, in sequence 8, the PCP score is not as high as the other sequences due to the blurriness of the images which makes the detection of the gripper edges very difficult. Hence, the connectivity potential function would not be able to give fair preferences to the sampled configurations in this case. Hence, the accuracy is a little bit affected due to missing the contribution of one of the potential functions of our CRF model. Coming to the public dataset, PCP scores of our method show comparable results to MC-15 [88]. However, the advantage of our approach over it is the ability to work without stopping on these sequences, while in sequence 2, MC-15 [88] needs two manual re-initialization to handle instrument disappearance from the scene. Herein, we want to highlight a very important point which is handling outliers during surgery. Unlike MC-15 [88] and ITOL [67] which don't have any scheme to handle outliers, our algorithm builds its recovery process on top of structural parts detections in order to handle outliers cases which can interrupt the surgery if they couldn't be handled immediately. Those outliers compose of images without instrument, images with blurred instrument and images with partial occlusions in which the instrument is very close to the Field of View (FoV) boundary. Our algorithm implements some potential to recognize these cases and activate the recovery process automatically. Despite of its importance and feasibility, the recovery process is required to handle all challenging cases which might result in inaccurate predictions occasionally. These inaccurate predictions influence the accuracy only during outliers handling, but generally the algorithm can recover easily and maintain the performance of the state-of-the-art methods.

Comparing on laparoscopic dataset, our approach outperforms MC-15 [88] by at least 20% at most of the accuracy thresholds in localizing the instrument center point and achieves very close performance to ITOL [67]. However, ITOL can't detect the forceps two tips as well as it is just intensity-based tracking algorithm. Hence, our algorithm tends to be more robust and practical for real surgeries due to its ability to localize the instrument left and right tips with high accuracy.

One more important strength point of the proposed approach is the ability to estimate the orientation of the instrument shaft. Unlike other approaches, the orientation is treated as a part in our CRF model, and this characteristic makes our approach successful one for the full integration with OCT imaging to position OCT scans according to given coordinates and orientation. The angular threshold results show also high accuracy in estimating the instrument orientation in all sequences of the different datasets.

## 8.5 Conclusions

We presented a new approach for localizing the forceps tips and center point as well as estimating the orientation of its shaft. The approach models



the instrument detection, tracking and pose estimation as an inference problem using CRF model. It models the relations between instrument parts and maintains confidence values to know whether to keep tracking by detection or to trigger the recovery process automatically. The performance of the proposed approach has been evaluated on retinal and laparoscopic surgeries using three different metrics. One great advantage of the approach over state of the art methods is the ability to handle tracking failures in real time. Such circumstances occur often in real complex datasets and waste a lot of operation time. The algorithm generates all parameters needed for OCT device in order to position OCT scans automatically in real surgery since it locates not only the instrument tips, but also the instrument orientation. Experimental results demonstrate the efficiency, robustness and accuracy of our method in real in-vivo scenarios and its ability to work on long videos. Comparisons with the state of the art methods on public and laparoscopic datasets demonstrate comparable results with the advantage that no manual re-initialization is needed.



## Chapter 9

# Conclusion and Outlook

### 9.1 Summary and Findings

This thesis is devoted to the study of medical instrument detection, tracking and pose estimation in real delicate microsurgery. We investigate the problem of instrument pose estimation in 2D images to introduce new computer-assisted solutions. The main objective of the proposed solutions is to pave the way for the utilization of the intra-operative Optical Coherence Tomography (iOCT) in retinal microsurgery. Therefore, the proposed algorithms aim to find the OCT device parameters for the correct positioning of its light beams. Those beams consist of a set of lines with specific orientation where each line hits a specific point of interest in the retina 2D image. Once these points have been scanned in 2D images, their distances to the retina surface are estimated, which represent the third dimension inside the eyeball space.

Initially, in Chapter 4, we rely on color information together with a simple model of the rigid part of the instrument. The approach demonstrates a promising real time solution on real in-vivo surgeries to estimate the orientation of the instrument shaft in addition to predicting the tip coordinate. In chapter 5, the powerful capability of deep learning in object detection is employed to model the instrument as an articulated object. A CNN network is used as a discriminative model to localize the probable locations of instrument's parts. A regression forest is trained on joint structural features (i.e. HOG) to work on top of CNN predictions. The approach shows high detection rate on public and real in-vivo videos for real in-vivo surgery. Moreover, the technique can extract precisely the joint point of forceps instrument in addition to estimate the required orientation. CNNs have been further employed in chapter 6 to regress the locations of the instrument joints within the image space.

In Chapter 7, we prove the robustness of relying on HOG features not only for classification, but also for regressing the instrument joints coordinates in in-vivo operations. A Hough forest is trained from unlabeled samples in the vicinity of the annotated points to enforce voting to be close to the instrument joints during regression. The algorithm localizes the two tips of forceps instrument in addition to the center point. Furthermore, it shows its practicality on long

real surgeries and comparability to the state-of-the-art methods with no need for manual initialization to handle tracking failures.

In Chapter 8, we develop a CRF model for instrument pose estimation. The approach models the relations between the instrument parts and uses discrete optimization to find the best configuration to express the instrument pose. The inference process localizes not only the tips coordinates but also the instrument orientation which is a very important step to orient the OCT B-scans during surgery. It achieves the real time requirements and shows also high prediction accuracy on public, laparoscopic and retinal microsurgery.

## 9.2 Limitations

We work on the problem of instrument pose estimation in complex and delicate operations. Our experiments and evaluation is done on real datasets collected from clinical partners using Carl-Zeiss microscopes. The initial idea of using color and geometric modelling is working when the light is focused close to the instrument body. Although this condition is satisfied in most cases, still the movement of the hand-held light tube remains uncontrollable, which makes relying on color information not fully reliable. Therefore, this leads us to use CNN to explore discriminative features of different parts of the instrument in chapter 5 and chapter 6. These methods demonstrate better results and works on larger number of real videos. However, speeding up feature extraction in CNNs for real time applications requires GPU implementation. Therefore, it needs GPU-equipped microscopes. As a compromise, feature engineering is used to model the instrument structure. The extracted features are chosen to be robust enough to the light changes during surgery. This is why we rely on HOG features in chapter 7 for both classification and regression processes. Despite of being discriminative enough for the instrument joints, HOG features create a broad detection map for the instrument shaft. Hence, only the two forceps tips and the joint point can be estimated using Hough forests.

Finally, In chapter 8, the proposed technique can predict the instrument two tips, center joint point and the orientation using CRF model. However, prior information of the instrument structure is very simple and might not be very efficient to handle cases of partial occlusions and low quality images due to lens distortion and fast motion. However, these are common problems for part-based object detection methods in computer vision.

## 9.3 Future Work

We proposed a number of techniques to address the problem of detection, tracking and pose estimation of medical instrument in retinal microsurgery. However, we have not solved all problems related to the challenges that might be faced in real surgery. We still believe that the work can be improved in different ways. Relying on deeper architectures of deep learning has the potential to advance the problem of instrument pose estimation. Regardless of the time complexity needed by deep learning, the advance in hardware technology in the

future can lead to abroad spread of deep learning applications in medical fields. Moreover, more complex models can be integrated into our CRF model to cope with partial occlusions. It is worth here to mention that any new model to solve a particular problem can be easily integrated in our proposed CRF model. Finally, online learning has been attracting a lot of attention in medical applications over the last few years. We believe that online learning of structural changes or instrument's shape transformations over time can reduce the amount of data required for training and increase the robustness of tracking.

## CHAPTER 9. CONCLUSION AND OUTLOOK

---

## Appendix A

# Instrument Type Detection Using CNN

State-of-the-art methods for instrument detection, tracking and pose estimation have not employed instrument type as prior information to enhance the performance of such methods. Having the instrument type as prior information during the surgery allows the integration of different type-specific detection algorithms where the selection among them is done automatically. This also has the potential to improve the algorithm's performance during surgery which is carried out by instruments with wide variations. Moreover, since many instrument detection algorithms require modeling the shape of the tool or some of its parts, having prior information about the type allows incorporating many shape models for different instruments in one general framework.

In this work, we show the power of using CNN to extract such prior information by automatically detecting the instrument type in live surgery. A deep convolutional neural network is learned from different instrument patches to predict the class of the instrument during the surgery.

### A.1 Proposed method

#### A.1.1 Problem Formulation

Assume we have  $C_1, C_2, \dots, C_N$  of different instrument classes, where  $N$  is the number of classes. Given an image patch  $X$  that comprises one of the instrument types, determining the type is simply classifying this patch by assigning one class label to it, and it can be formulated by calculating the posterior probability  $P(C_i|X)$ .

To calculate this probability, a convolutional neural network shown in Figure.A.1 is learned from few examples. The examples are provided to the deep network in the form of pairs  $(T_i, C_i)$ , where  $T_i$  is the  $i$ -th  $32 \times 32$  input patch associated with its class label  $C_i$ . The network is learned to find a function  $F(T, C)$  that can predict the class for any unseen patch. To predict a class for a given new

input patch  $X$ , we input this patch to the network and get a class label based on Eq.A.1 which is computed by CNN.

$$p(c|X) = \max_i(p(C_i|X)) \tag{A.1}$$

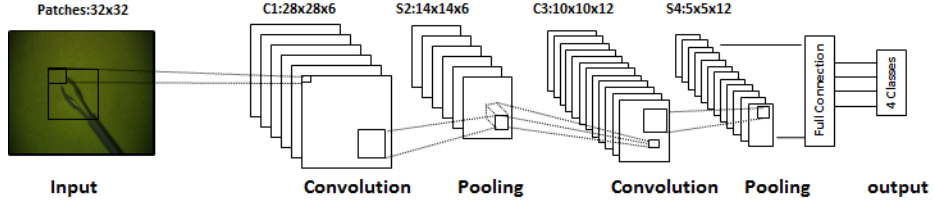


Figure A.1: Convolutional Neural Network for Tool Type Detection.

### A.1.2 Convolutional Neural Network Structure

The convolutional neural network are designed as shown in Figure.A.1, where it has two convolutions layers, two pooling layers and a fully connected neural network.

The input to the first convolution layer is a patch of size  $32 \times 32$  pixels, which includes the entire instrument head part. Therefore, the input patch should be resized to the standard input size of the designed network. The coefficients of the convolution kernel is randomly initialized, and the each convolution kernel has a size of  $5 \times 5$  pixels. Six convolution channels are used and the convolved images has the size  $28 \times 28$  pixels.

The output of the first convolution layer is fed into the first pooling layer, which uses a max-pooling operation to account for any small translations in the input patches. The output of this layer has the same number of features channels but with half the size.

The second convolution layer has the same kernel size and double the features channels, and another pooling layer is applied to the output of the second convolution layer to produce the features vector of size  $5 \times 5 \times 12$ . These features are the input to a fully connected neural network with  $N$  output neurons. The output probability at each neuron represents the confidence that the patch belongs to each class.

All the parameters of the neural network, in addition to the convolution kernel parameters are optimized iteratively using the stochastic gradient descent and the back-propagation algorithm based on Eq. A.2.

$$\operatorname{argmin} \frac{1}{n} \sum_{i=1}^n L(F(T_i; w_1, \dots, w_L), C_i), \tag{A.2}$$

where  $w_1, \dots, w_L$  are the network parameters and  $L$  is the loss function which computes the errors between the predictions and the desired outputs for  $\mathbf{n}$



training examples. In the designed network, the used loss function is the softmax function.



Figure A.2: Four different types of surgical instruments.





				
Patch Group	1	2	3	4
1	0.88	0.01	0	0.105
2	0	1	0	0
3	0	0.4	0.96	0
4	.015	0	0	0.985

Figure A.3: Confusion matrix of eye phantom group.

## A.2 Results

This approach is validated on two groups of datasets, one are images for eye phantom taken from Carl-Zeiss microscopes and the second are real microsurgery images taken from in-vivo video surgeries. The datasets compose of image patches with different illumination conditions, different orientations and different states of the instrument’s grippers. Before feeding the network with the patches from both cases, the patches are normalized to compensate for illumination changes during the operation.

The convolutional neural network used in this work is the MatConvNet [116] which is trained using the pack-propagation algorithm with 100 epochs and learning rate of 0.001.

### A.2.1 Eye Phantom Datasets

Four datasets from Carl-Zeiss Microscopes have been used for training and testing the proposed algorithm on four instrument types shown in Figure. A.2. For each dataset, 600 images are used to train the convolutional neural network on different type after pre-processing operation. The testing operation is performed on 200 images from each dataset. In 94 percent of the testing

images, the instrument type is correctly classified. The confusion matrix in Figure. A.3 shows the details.

### A.2.2 Real Microsurgery Datasets

Three datasets from real surgery videos have been used for training and testing on three instrument types shown in Figure.A.4. For each dataset, 400 images are used to train the convolutional neural network while testing is performed on 200 images. The images show challenging cases where they have clear presence of the instrument shadow and the retinal vessels. The results show that the algorithm could handle instrument with cluttered background, and in 87% of the testing images the instrument type is correctly classified. Figure. A.5 shows the confusion matrix of the results.

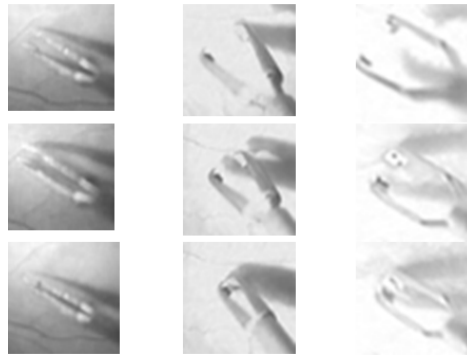


Figure A.4: Three different tool types with different poses.




			
Patch Group	1	2	3
1	0.995	0	.005
2	0.01	0.875	0.115
3	0.045	0.125	0.83

Figure A.5: Confusion matrix for real microscopic images.

### A.2.3 Results Analysis

In the first group, most of the confusions occur between type 1 and 4, and the reason behind that is the characteristics of the gripper tips. Both types, at some

orientations, look similar to each other and create false detections. Moreover, when the tools are in close state, it is very difficult to classify all the types correctly since most of the informative instrument structures are not clearly visible.

For the second group, the results are not as good as the first group due to the cluttered background and the severe changes in the surgery conditions such as blurring degrees and illumination changes. However, the classification precision is still high, and most of the misclassifications are due to the confusion between types 2 and 3 when the forceps is closed. In that case, the two gripper sides look parallel with small gap in between which makes it difficult to distinguish between them.

### **A.3 Conclusion**

The proposed approach shows promising results to extract prior information about instrument type during live surgery. The detected instrument class can be incorporated with other tracking and detection algorithms to improve the practicality and robustness and to reduce uncertainty. Using CNN in this direction demonstrates high classification accuracy at real time performance where it would be very difficult for any classical classifier to find discriminative features to capture fine difference between different instruments types. Integrating this approach with other tracking and detection methods would contribute significantly towards minimally invasive procedures.



# Bibliography

- [1] Shadi Albarqouni, Christoph Baur, Felix Achilles, Vasileios Belagiannis, Stefanie Demirci, and Nassir Navab. Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE transactions on medical imaging*, 35(5):1313–1321, 2016.
- [2] M. Allan, S. Ourselin, S. Thompson, D.J. Hawkes, J. Kelly, and D. Stoyanov. Toward detection and localization of instruments in minimally invasive surgery. *Biomedical Engineering, IEEE Transactions on*, 60(4):1050–1058, 2013.
- [3] Max Allan, Ping-Lin Chang, Sébastien Ourselin, David J. Hawkes, Ashwin Sridhar, John Kelly, and Danail Stoyanov. *Image Based Surgical Instrument Pose Estimation with Multi-class Labelling and Optical Flow*, pages 331–338. Springer International Publishing, Cham, 2015.
- [4] Max Allan, Steve Thompson, Matthew J Clarkson, Sébastien Ourselin, David J Hawkes, John Kelly, and Danail Stoyanov. 2d-3d pose tracking of rigid instruments in minimally invasive surgery. In *International Conference on Information Processing in Computer-assisted Interventions*, pages 1–10. Springer, 2014.
- [5] David Allen and Abhay Vasavada. Cataract and surgery for cataract. *British Medical Journal*, 7559:128–135, 2006.
- [6] Mohamed Alsheakhali, Abouzar Eslami, and Nassir Navab. Surgical tool detection and tracking in retinal microsurgery. In *SPIE 2015*, volume 9415, pages 1–6.
- [7] Keyvan Amini Khoiy, Alireza Mirbagheri, and Farzam Farahmand. Automatic tracking of laparoscopic instruments for autonomous control of a cameraman robot. *Minimally Invasive Therapy & Allied Technologies*, 25(3):121–128, 2016.
- [8] Y. M. Baek, S. Tanaka, K. Harada, N. Sugita, A. Morita, S. Sora, and M. Mitsuishi. Robust visual tracking of robotic forceps under a microscope using kinematic data fusion. *IEEE/ASME Transactions on Mechatronics*, 19(1):278–288, Feb 2014.
- [9] Ahmadreza Baghaie, Zeyun Yu, and Roshan M DâĂŹSouza. State-of-the-art in retinal optical coherence tomography image analysis. *Quantitative imaging in medicine and surgery*, 5(4):603, 2015.

## BIBLIOGRAPHY

---

- [10] Pierre Baldi and Peter J Sadowski. Understanding dropout. In *Advances in Neural Information Processing Systems*, pages 2814–2822, 2013.
- [11] Marcin Balicki, Jae-Ho Han, Iulian Iordachita, Peter Gehlbach, James Handa, Russell Taylor, and Jin Kang. Single fiber optical coherence tomography microsurgical instruments for computer and robot-assisted retinal surgery. In *MICCAI 2009*, pages 108–115.
- [12] Marcin Balicki, Eric Meisner, Raphael Sznitman, Russell Taylor, and Gregory Hager. Visual Tracking of Surgical Tools for Proximity Detection in Retinal Surgery. pages 55–66, 2011.
- [13] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.
- [14] Vasileios Belagiannis. *Human pose estimation in complex environments*. Dissertation, Technische UniversitÄt MÄijnchen, MÄijnchen, 2015.
- [15] Vasileios Belagiannis, Christian Amann, Nassir Navab, and Slobodan Ilic. Holistic human pose estimation with regression forests. In *AMDO 2014*, pages 20–30.
- [16] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.
- [17] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer, 2012.
- [18] Conrad Berens and Michael Loutfallah. Aniseikonia: A study of 836 patients examined with the ophthalmo-eikonometer. page 234–267. Transactions of the American Ophthalmological Society, 1938.
- [19] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.
- [20] Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.
- [21] Christopher M Bishop. *Pattern recognition and machine learning (information science and statistics)* springer-verlag new york. Inc. Secaucus, NJ, USA, 2006.
- [22] Immanuel M. Bomze, Marco Budinich, Panos M. Pardalos, and Marcello Pelillo. *The Maximum Clique Problem*, pages 1–74. Springer US, Boston, MA, 1999.
- [23] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, pages 177–187, Paris, France, August 2010. Springer.
- [24] Wendy L Braje, Daniel Kersten, Michael J Tarr, and Nikolaus F Troje. Illumination effects in face recognition. *Psychobiology*, 26(4):371–380, 1998.
- [25] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

## BIBLIOGRAPHY

---

- [26] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2488–2496, 2015.
- [27] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010.
- [28] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.
- [29] Chun-Ju Chen, WS-W Huang, and Kai-Tai Song. Image tracking of laparoscopic instrument using spiking neural networks. In *ICCAS 2013*, pages 951–955.
- [30] H. Chen, X. Wang, and P. A. Heng. Automated mitosis detection with deep regression networks. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 1204–1207, April 2016.
- [31] Adam Coates. *Demystifying unsupervised feature learning*. PhD thesis, Stanford University, 2012.
- [32] A Criminisi, J Shotton, and E Konukoglu. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. *Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114*, 5(6):12, 2011.
- [33] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893, 2005.
- [34] Amaury Dame and Eric Marchand. Accurate real-time tracking using mutual information. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 47–56. IEEE, 2010.
- [35] Arnaud Delorme and Simon J Thorpe. Face identification using one spike per neuron: resistance to image degradations. *Neural Networks*, 14(6):795–803, 2001.
- [36] C. Doignon, P. Graebling, and M. de Mathelin. Real-time segmentation of surgical instruments inside the abdominal cavity using a joint hue saturation color feature. *Real-Time Imaging*, 11(5-6):429–442, October 2005.
- [37] C. Doignon, F. Nageotte, and M. De Mathelin. Detection of grey regions in color images : application to the segmentation of a surgical instrument in robotized laparoscopy. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 3394–3399, Sept 2004.
- [38] Xiaofei Du, Maximilian Allan, Alessio Dore, Sebastien Ourselin, David Hawkes, John D Kelly, and Danail Stoyanov. Combined 2d and 3d tracking of surgical instruments for minimally invasive and robotic-assisted surgery. *International journal of computer assisted radiology and surgery*, 11(6):1109–1119, 2016.

## BIBLIOGRAPHY

---

- [39] Vittorio Ferrari, Manuel Marin-Jimenez, and Andrew Zisserman. Progressive search space reduction for human pose estimation. In *CVPR 2008*, pages 1–8.
- [40] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22(1):67–92, Jan 1973.
- [41] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.
- [42] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.
- [43] Juergen Gall and Victor Lempitsky. Class-specific hough forests for object detection. In *In Proceedings IEEE Conference Computer Vision and Pattern Recognition*, 2009.
- [44] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(11):2188–2202, November 2011.
- [45] Donald Geman and Bruno Jedynak. An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14, 1996.
- [46] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Aistats*, volume 15, page 275, 2011.
- [47] Matthew R Glucksberg, Robert Dunn, and Claudine P Giebs. In vivo micropuncture of retinal vessels. *Graefe's archive for clinical and experimental ophthalmology*, 231(7):405–407, 1993.
- [48] David E Golberg. Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989:102, 1989.
- [49] Kenneth Wayne Grace. *Kinematic design of an ophthalmic surgery robot and feature extracting bilateral manipulation*. PhD thesis, Northwestern University, 1995.
- [50] Martin Groeger, Gerd Hirzinger, and Klaus Arbter. *Motion tracking for minimally invasive robotic surgery*. INTECH Open Access Publisher, 2008.
- [51] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.
- [52] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.
- [53] Richard S Hunter. Photoelectric color difference meter. *Josa*, 48(12):985–995, 1958.



## BIBLIOGRAPHY

---

- [54] Arch Iwanoff. Beiträge zur normalen und pathologischen anatomie des auges. *Graefe's Archive for Clinical and Experimental Ophthalmology*, 11(1):135–170, 1865.
- [55] Patrick S Jensen, Kenneth W Grace, Rajpaul Attariwala, J Edward Colgate, and Matthew R Glucksberg. Toward robot-assisted vascular microsurgery in the retina. *Graefe's archive for clinical and experimental ophthalmology*, 235(11):696–701, 1997.
- [56] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. 2014.
- [57] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*.
- [58] Andrej Krenker, Andrej Kos, and Janez Bešter. *Introduction to the artificial neural networks*. INTECH Open Access Publisher, 2011.
- [59] 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.
- [60] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.
- [61] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [62] Andrew Lang, Aaron Carass, Matthew Hauser, Elias S Sotirchos, Peter A Calabresi, Howard S Ying, and Jerry L Prince. Retinal layer segmentation of macular oct images using boundary classification. *Biomedical optics express*, 4(7):1133–1152, 2013.
- [63] S. L. Lauritzen and D. J. Spiegelhalter. Readings in uncertain reasoning. chapter Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems, pages 415–448. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.
- [64] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [65] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.
- [66] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014.

## BIBLIOGRAPHY

---

- [67] Yeqing Li, Chen Chen, Xiaolei Huang, and Junzhou Huang. Instrument tracking via online learning in retinal microsurgery. In *MICCAI 2014*, pages 464–471.
- [68] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.
- [69] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- [70] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [71] 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.
- [72] A. Mahmood. Structure-less object detection using adaboost algorithm. In *Machine Vision, 2007. ICMV 2007. International Conference on*, pages 85–90, Dec 2007.
- [73] Christopher D Malon, Eric Cosatto, et al. Classification of mitotic figures with convolutional neural networks and seeded blob features. *Journal of pathology informatics*, 4(1):9, 2013.
- [74] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [75] Shun Miao, Z Jane Wang, Yefeng Zheng, and Rui Liao. Real-time 2d/3d registration via cnn regression. *Proc. IEEE Int’l Symp. Biomedical Imaging*, pages 1–4, 2016.
- [76] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.
- [77] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.
- [78] Sebastian Nowozin and Christoph H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3&4):185–365, 2011.
- [79] Zachary Pezzementi, Sandrine Voros, and Gregory D Hager. Articulated object tracking by rendering consistent appearance parts. In *ICRA 2009*, pages 3940–3947.
- [80] Tomas Pfister, Karen Simonyan, James Charles, and Andrew Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In *Asian Conference on Computer Vision*, pages 538–552. Springer, 2014.

## BIBLIOGRAPHY

---

- [81] Mark R. Pickering, Abdullah A. Muhiit, Jennie M. Scarvell, and Paul N. Smith. A new multi-modal similarity measure for fast gradient-based 2D-3D image registration. In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*, pages 5821–5824, 2009.
- [82] Constantin J Pournaras, Ross D Shonat, Jean-Luc Munoz, and Benno L Petrig. New ocular micromanipulator for measurements of retinal and vitreous physiologic parameters in the mammalian eye. *Experimental eye research*, 53(6):723–727, 1991.
- [83] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.
- [84] Austin Reiter and Peter K Allen. An online learning approach to in-vivo tracking using synergistic features. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3441–3446. IEEE, 2010.
- [85] Austin Reiter, Peter K Allen, and Tao Zhao. Feature classification for tracking articulated surgical tools. In *MICCAI 2012*, pages 592–600.
- [86] Rogério Richa, Marcin Balicki, Eric Meisner, Raphael Sznitman, Russell Taylor, and Gregory Hager. Visual tracking of surgical tools for proximity detection in retinal surgery. In *IPCAI 2011*, pages 55–66.
- [87] Stephen J Riederer. Current technical development of magnetic resonance imaging. *IEEE Engineering in Medicine and Biology Magazine*, 19(5):34–41, 2000.
- [88] Nicola Rieke, David Joseph Tan, Mohamed Alsheakhali, Federico Tombari, Chiara Amat di San Filippo, Vasileios Belagiannis, Abouzar Eslami, and Nassir Navab. Surgical tool tracking and pose estimation in retinal microsurgery. In *MICCAI 2015*, volume 9349, pages 266–273. Springer Verlag, 2015.
- [89] Nicola Rieke, David Joseph Tan, Chiara Amat di San Filippo, Federico Tombari, Mohamed Alsheakhali, Vasileios Belagiannis, Abouzar Eslami, and Nassir Navab. Real-time localization of articulated surgical instruments in retinal microsurgery. *Medical image analysis*, 2016.
- [90] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.
- [91] H Roodaki, K Filippatos, A Eslami, and N Navab. Introducing Augmented Reality to Optical Coherence Tomography in Ophthalmic Microsurgery. In *Mixed and Augmented Reality (ISMAR), 2015 IEEE International Symposium on*, pages 1–6. 2015.
- [92] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*.

## BIBLIOGRAPHY

---

- [93] Robert P Rutstein. Retinally induced aniseikonia: a case series. *Optometry and Vision Science*, 89(11):50–55, 2012.
- [94] Hannes Schulz and Sven Behnke. Object-class segmentation using deep convolutional neural networks. In *Proceedings of the DAGM Workshop on New Challenges in Neural Computation*, pages 58–61. Citeseer, 2011.
- [95] T Sejnowski. Learning and relearning in boltzmann machines. *Graphical Models: Foundations of Neural Computation*, page 45, 2001.
- [96] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.
- [97] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.
- [98] Solomon Eyal Shimony. Finding maps for belief networks is np-hard. *Artif. Intell.*, 68(2):399–410, August 1994.
- [99] Shalabh Sinha. *Minimally Invasive Vitreous Surgery: 20 Gauge to 27 Gauge*. JP Medical Ltd, 2013.
- [100] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [101] C Staub, G Panin, A Knoll, Robert Bauernschmitt, and German Heart Center Munich. Visual instrument guidance in minimally invasive robot surgery. *International Journal on Advances in Life Sciences Volume 2, Number 3 & 4, 2010*, 2010.
- [102] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4(4):267–373, April 2012.
- [103] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.
- [104] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2553–2561. Curran Associates, Inc., 2013.
- [105] Raphael Sznitman, Karim Ali, Rogério Richa, Russell H Taylor, Gregory D Hager, and Pascal Fua. Data-driven visual tracking in retinal microsurgery. In *MICCAI 2012*, pages 568–575.
- [106] Raphael Sznitman, Anasuya Basu, Rogério Richa, Jim Handa, Peter G, Russell H Taylor, and Gregory D J, B H. Unified detection and tracking in retinal microsurgery. In *MICCAI 2011*, pages 1–8.

## BIBLIOGRAPHY

---

- [107] Raphael Sznitman, Carlos Becker, and Pascal Fua. Fast part-based classification for instrument detection in minimally invasive surgery. In *MICCAI 2014*, pages 692–699.
- [108] Raphael Sznitman and Bruno Jedynek. Active testing for face detection and localization. *arXiv preprint arXiv:1003.5249*, 2010.
- [109] David Joseph Tan and Slobodan Ilic. Multi-forest tracker: A chameleon in tracking. In *CVPR 2014*, pages 1202–1209.
- [110] Russell H Taylor, Arianna Menciassi, Gabor Fichtinger, and Paolo Dario. Medical robotics and computer-integrated surgery. In *Springer handbook of robotics*, pages 1199–1222. Springer, 2008.
- [111] Oliver Tonet, Ramesh U Thoranaghatte, Giuseppe Megali, and Paolo Dario. Tracking endoscopic instruments without a localizer: A shape-analysis-based approach. *Computer Aided Surgery*, 12(1):35–42, 2007.
- [112] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.
- [113] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04*, pages 104–, New York, NY, USA, 2004. ACM.
- [114] Vladimir Naoumovitch Vapnik. *Statistical learning theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.
- [115] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.
- [116] Andrea Vedaldi and Karel Lenc. Matconvnet - convolutional neural networks for MATLAB. *CoRR*, abs/1412.4564, 2014.
- [117] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.
- [118] Sandrine Voros, Jean-Alexandre Long, and Philippe Cinquin. Automatic detection of instruments in laparoscopic images: A first step towards high-level command of robotic endoscopic holders. *The International Journal of Robotics Research*, 26(11-12):1173–1190, 2007.
- [119] Congcong Wang, R. Palomar, and F. A. Cheikh. Stereo video analysis for instrument tracking in image-guided surgery. In *Visual Information Processing (EUVIP), 2014 5th European Workshop on*, pages 1–6, Dec 2014.
- [120] Lichao Wang, Vasileios Belagiannis, Carsten Marr, Fabian J. Theis, Guang-Zhong Yang, and Nassir Navab. Anatomic-landmark detection using graphical context modelling. In *12th IEEE International Symposium on Biomedical Imaging, ISBI 2015, Brooklyn, NY, USA, April 16-19, 2015*, pages 1304–1307, 2015.

## BIBLIOGRAPHY

---

- [121] Guo-Qing Wei, Klaus Arbter, and Gerd Hirzinger. Real-time visual servoing for laparoscopic surgery. *IEEE Engineering in Medicine and Biology Magazine*, 16(1):40–45, 1997.
- [122] Junichi Yonemoto, Yuuko Noda, Nami Masuhara, and Shigeaki Ohno. Age of onset of posterior vitreous detachment. *Current opinion in ophthalmology*, 7(3):73–76, 1996.
- [123] Jiawei Zhou and Shahram Payandeh. Visual tracking of laparoscopic instruments. *Journal of Automation and Control Engineering Vol*, 2(3), 2014.
- [124] Xiaojin Zhu. Semi-supervised learning literature survey. 2005.