COMPUTER VISION BASED ARTICULATED MOTION UNDERSTANDING

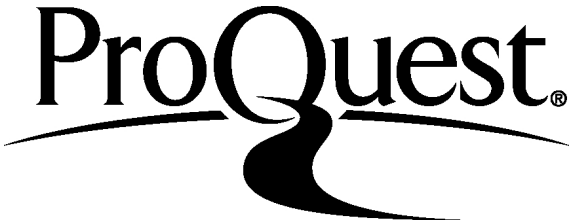
by

Suren Kumar
Feb 2016

A dissertation submitted to the
Faculty of the Graduate School of
the University at Buffalo, State University of New York
in partial fulfilment of the requirements for the
degree of

Doctor of Philosophy

Department of Mechanical and Aerospace Engineering



The thesis of Suren Kumar was reviewed by the following:

Dr. Venkat Krovi
Professor, Mechanical and Aerospace Engineering, State University of New York (SUNY), Buffalo
Thesis Advisor, Committee Chair

Dr. Jason Corso
Associate Professor, Electrical Engineering and Computer Science, University of Michigan, Ann Arbor
Thesis Co-Advisor, Committee Member

Dr. Manoranjan Majji
Assistant Professor, Mechanical and Aerospace Engineering, State University of New York (SUNY), Buffalo
Committee Member

Dr. Michael J. Jones
Senior Principal Member Research Staff, Mitsubishi Electric Research Laboratories (MERL), Boston
Outside Reader

Acknowledgments

I am sitting in a Starbucks in Ann Arbor and it seems surreal that I am about to write the final and probably the most important piece of my thesis. At the end of this seemingly long journey, first and foremost, I would like to acknowledge the contribution of my thesis advisor, Dr. Venkat Krovi. On the one hand, you gave me enormous freedom to do my research, while at the same time, you steered me to a course correction whenever I went astray. Thanks for helping me articulated my thoughts and correcting my research papers at the last minute. I would like to thank my co-advisor and current postdoc advisor, Dr. Jason Corso for inspiring me to do bold research work even if it may not lead to publications and acting as a "snow-plough" on multiple research papers.

I am thankful to my internship advisor and committee member, Dr. Micheal Jones for providing reasearch feedback and helping me in improving the thesis draft. I would also like to thank my committee member, Dr. Manoranjan Majji for being available for discussions and my thesis presentation. I would like to extend my sincere gratitude to Priyanshu Agarwal, who inspired me by his work ethic and collaborated on tracking and human pose estimation part of this thesis. My dear friend and colleague, Vikas Dhiman collaborated on the articulation estimation and language part of this thesis. Thanks for allowing me to express my daily frustrations with the lack of progress in my research during our time together in Buffalo. I would also like to thank Javad Sovizi for collaborating on pose estimation work and being a cheerful companion, especially during the last part of my thesis. I am also thankful to Matthias and Mark for reading a chapter of my thesis and providing corrections.

I would like to thank my friends in Buffalo including Santhosh, Ratna, Kali, Rohit, Glenn, Seungkook Jun, Xiaobo, Dipanshu, Manish (the running mate), Manish, Tarun, Utkala, Prerna, Venkat, Kumar, Ujjwala, Abhishek and other AID Buffalo volunteers for making my life outside the lab quite cheerful. Thanks to friends in Boston including Sagun, Katie, Eytan, Mayank, Mayuri, Darshana, Anshul, Nitin, and Siva for giving me reasons to promptly leave my office at

5PM during my internship. I would also like to thank friends in Princeton including Harish and Sree for our various fun outings together.

My dear fiancee, Aahana deserves a special mention for helping me through this entire process till the end. She helped me in writing the introduction and painstakingly corrected the grammatical mistakes in the entire thesis. But more importantly, her emotional support and love helped me stay afloat when everything seemed to go downhill. Lastly, I would like to thank my parents for their sacrifices in ensuring that I got a great education.

Table of Contents

Acknowledgments **iv**

List of Tables **x**

List of Figures **xi**

Abstract **xviii**

Chapter 1
Introduction **1**
1.1 Organization 3
1.2 Impact 6
1.2.1 Research Contribution 6
1.2.2 Selected Publications 7
1.2.2.1 Journal 7
1.2.2.2 Refereed Conference 8
1.2.3 Code Released 9
1.2.4 Dataset Released 9

Chapter 2
Articulated Object Detection and Tracking **10**
2.1 Introduction 10
2.2 Related Work 11
2.2.1 Single Object Tracking 11
2.2.2 Multiple Object Tracking 11
2.3 Research Contribution 13
2.4 Product of Tracking Experts 14

2.4.1 Object Detection . . . 16
2.4.2 Point Feature Based Tracking . . . 17
2.4.3 Background Subtraction . . . 19
2.4.4 Motion Continuity . . . 19
2.4.5 Dense Optical Flow . . . 20
2.5 Multiple Object Tracking . . . 21
2.5.1 State Based Tracking Representation . . . 21
2.5.2 Occlusion Detection and Track Resolution . . . 21
2.5.3 Interpolation Based Trajectory Estimation . . . 24
2.5.4 Entry/Exit Estimation . . . 24
2.6 Experiments . . . 26
2.6.1 Surgical Tool Tracking . . . 26
2.6.2 Person Tracking . . . 29
2.7 Discussion . . . 30
2.7.1 Product of Tracking Experts (PoTE) Model . . . 30
2.7.2 Point Feature Based Tracking . . . 31
2.7.3 Group Based Tracking and Identity Resolution . . . 31
2.7.4 Failure Modes Observed . . . 31
2.8 Conclusion . . . 32

Chapter 3
Pose Estimation **34**
3.1 Introduction . . . 34
3.2 Surgical Pose Estimation . . . 35
3.3 Previous Work . . . 38
3.3.1 Computational Complexity . . . 39
3.3.2 Regression Methods . . . 39
3.4 System Overview . . . 41
3.4.1 Visual Features . . . 42
3.4.2 Gaussian Process Regression . . . 43
3.5 Improving Prediction . . . 45
3.6 Experiments . . . 46
3.6.1 Execution Time . . . 48
3.6.2 Filtering . . . 49
3.6.3 Generalization Beyond Learned Data . . . 50
3.7 Future Work . . . 50
3.8 Conclusion . . . 51

Chapter 4
Articulation Estimation **53**
4.1 Introduction . . . 53
4.2 Related Work . . . 55
4.2.1 Structure from Motion . . . 55
4.2.2 Direct Motion Sensing Approaches . . . 55
4.3 Articulation Model . . . 56
4.3.1 Articulation Classification . . . 57
4.3.2 Rigid Body Articulation Classification . . . 58
4.3.2.1 Prismatic . . . 60
4.3.2.2 Revolute . . . 61
4.3.2.3 Plane Constrained Motion . . . 61
4.3.2.4 General Rigid Body Motion . . . 61
4.3.2.5 Static . . . 61
4.3.3 Point Particle Articulation Classification . . . 62
4.4 Temporal Structure . . . 62
4.4.1 Choosing Order . . . 64
4.5 Articulation Model Estimation . . . 64
4.6 SLAM for Dynamic World . . . 65
4.6.1 Time update . . . 67
4.6.2 Measurement Update . . . 68
4.6.3 Dynamic World Representation . . . 68
4.7 Articulated EKF SLAM . . . 69
4.7.1 Robot Motion Model . . . 69
4.7.2 Observation Model . . . 70
4.7.3 Jacobian Computation . . . 71
4.8 Results . . . 72
4.8.1 Configuration Estimation . . . 72
4.8.2 Temporal Order . . . 73
4.8.3 Articulation Estimation . . . 74
4.8.4 Dynamic World SLAM . . . 75
4.8.4.1 Qualitative Analysis . . . 76
4.8.4.2 Quantitative Analysis . . . 76
4.9 Conclusion . . . 78

Chapter 5
Representation in Language **79**
5.1 Language Output . . . 79
5.1.1 Detection and Tracking . . . 81
5.1.2 Pose Attributes . . . 82

5.2 Can Language inform Vision? . . . 82
5.3 Model . . . 85
5.3.1 Paired Sparse Model . . . 85
5.3.2 Compositional Sparse Model . . . 87
5.4 Features . . . 89
5.4.1 Visual . . . 89
5.4.2 Audio . . . 90
5.5 Experiments and Results . . . 91
5.5.1 Dataset . . . 91
5.5.2 Visualization . . . 92
5.5.3 Reproduction Evaluation . . . 92
5.5.4 Generalization Evaluation . . . 94
5.6 Conclusion . . . 97

Chapter 6
Conclusion **99**
6.1 Overview of Results . . . 100
6.1.1 Tracking . . . 100
6.1.2 Pose Estimation . . . 100
6.1.3 Articulation Estimation . . . 101
6.1.4 Vision and Language . . . 101
6.2 Applications . . . 102
6.2.1 Surveillance . . . 102
6.2.2 Surgical Semantic Feedback . . . 102
6.2.3 Motion Planning in Dynamic Environments . . . 103
6.2.4 Minimum Time Scene Understanding . . . 104
6.2.5 Language Grounding . . . 104
6.3 Future Work . . . 105
6.3.1 Theoretical Development . . . 105
6.3.2 Experimentation . . . 106

Appendix A
Additional Derivations **107**
A.1 Upper Bound on Point Tracker Error . . . 107

Bibliography **109**

List of Tables

2.1 Accuracy using Baseline and PoTE tracker 27
2.2 Summary of quantitative accuracy results on PETS 2009 Dataset (S2.L1 walking scenario). 30

3.1 Tool pose estimate angular accuracy in degrees using different visual features 48
3.2 Average error in tool opening angle with lighting variations . . . 50

4.1 Error metrics for center and radius error 73
4.2 Comparison of localization error for two different SLAM algorithms 77

5.1 V-measure distance matrix between the feature representation. v_1 and v_2 represent RGB and Fourier descriptor features, respectively, a_1 and a_2 represent the feature extracted from first and second audio segment. 91
5.2 Shape and Color Exemplars in the dataset 92

List of Figures

1.1 Overview of the overarching problem addressed in the thesis. Streaming data from visual sensors can be used to identify the objects present in the scene or understand articulated structure without any prior knowledge of the objects. However, typically articulated structure is implicitly estimated via detection of an object category. Both tracking and pose output can be exploited to generate language which can in turn be used to look for specific objects or poses in the scene, (b) Demonstration of "door" detection, (c) Exemplar of pose estimate of the door which is directly related to its language representation "Open" and (d) Shows the information extracted from the environment based on articulation estimation which separates objects based on motion and finds their motion axis. 4
(a) Overview 4
(b) Tracking 4
(c) Pose 4
(d) Articulation 4

2.1 System Flow Diagram 14
2.2 Two tracking experts on sides and resulting PoTE model in middle with Gaussian probability density function contours on right. First tracking expert has associated Gaussian with mean = $[245, 270]^T$ and variance = $diag([16.66, 25])$, second tracker has associated Gaussian with mean = $[255, 275]^T$ with same variance, when combined using PoTE model results into a Gaussian with mean = $[250, 272.5]^T$ and variance =$diag([8.33, 12.5])$ 16
2.3 Flow diagram of tracking using KLT. 18

2.4 State based representation of entities in the tracking framework. Multiple group of entities might form which are tracked as groups, groups might split entirely into individual entities or few entities might leave the group. 22
2.5 Formation and fragmentation of tracked sub-groups in 'S2.L1 walking' scenario in the PETS 2009 Dataset S2 for People Tracking and the associated state space as explored in the video. (Please view in color). The Entities in flowchart on the right hand side are represented by the same color as the bounding boxes in frames on the left hand side. Note that because of significant overlap (E_1, E_2, E_3 combine to $E_{1,2,3}$ in frame 37) several entities combine to form a single entity which is collectively tracked. The separate entities may re-split from a grouped entity (as $E_{4,5}$ split to E_4 and E_5 in frame 39) given enough observations. 24
(a) Frame 13 24
(b) Frame 37 24
(c) Frame 39 24
(d) Frame 54 24
2.6 Interpolation based trajectory estimation to track objects through an interaction process 25
(a) Coarse Tracking 25
(b) Fine Tracking 25
2.7 Learned HOG Templates with a representative bounding box for different tool types 27
(a) Visual Image of Tool 27
(b) Visual Image of Clamp 27
(c) Learned HOG of Tool 27
(d) Learned HOG of Clamp 27
2.8 Tracking results for "Tool" and "Clamp" on various surgical operation videos in proposed dataset. (Please view in color) 28
2.9 Tracking results for "S2.L1 walking" scenario in the PETS 2009 Dataset S2 for People Tracking highlighting the various concepts of our tracker including entry, exit, and multiple tracking. (Please view in color) 33
(a) Frame 3 33
(b) Frame 13 33
(c) Frame 16 33
(d) Frame 17 33
(e) Frame 18 33
(f) Frame 26 33

(g) Frame 37 . . . 33
(h) Frame 41 . . . 33
(i) Frame 73 . . . 33
(j) Frame 98 . . . 33
(k) Frame 109 . . . 33
(l) Frame 121 . . . 33
(m) Frame 130 . . . 33
(n) Frame 131 . . . 33
(o) Frame 132 . . . 33
(p) Frame 245 . . . 33
(q) Frame 249 . . . 33
(r) Frame 265 . . . 33
(s) Frame 269 . . . 33
(t) Frame 332 . . . 33

3.1 Flow chart of the tool pose estimation framework . . . 37
3.2 (a) Linear Regression yields the model $y = mx + c$ with $m = 0.9644$ and $c = 1.5891$ (b) Quadratic model yields the model $y = ax^2 + bx + c$ with $a = 0.0534$, $b = -0.04781$ and $c = 7.1231$ (c) Bayesian Regression fit uses a zero mean Gaussian observation noise with variance 10 and a zero mean Gaussian with diagonal variance 5 prior on slope and intercept. The estimation process results in a mean parameter estimate of 0.9867 slope and 1.1797 intercept. (d) GPR with constant mean function, Gaussian likelihood and Squared Exponential covariance function . . . 41
(a) Linear Regression . . . 41
(b) Quadratic Regression . . . 41
(c) Bayesian Linear Regression . . . 41
(d) GPR . . . 41
3.3 An example image with tool and corresponding HOG feature. The two different parts of the tool show up distinctly in the orientation histogram. . . . 42
3.4 Gaussian Process regression with 3 sigma bounds plotted with true value of tool opening angle . . . 46
3.5 Customized Box Trainer Setup Retrofitted with Optical Reflective Markers . . . 47
3.6 True, Regression mean estimate and filtered estimates of tool opening angle . . . 49

3.7 Representative video frames for a "dark" sequence in the collected dataset obtained using GPR to estimate tool opening angle. First row shows the image frame, second row shows the orientation and opening angle of the left tool and the third row shows the orientation and opening angle of the right tool using LBP features. 51
(a) Frame 52 51
(b) Frame 139 51
(c) Frame 209 51
(d) Frame 217 51
(e) Frame 285 51
(f) Frame 52 51
(g) Frame 139 51
(h) Frame 209 51
(i) Frame 217 51
(j) Frame 285 51
(k) Frame 52 51
(l) Frame 139 51
(m) Frame 209 51
(n) Frame 217 51
(o) Frame 285 51
3.8 Left and right tool opening angles for a "dark" sequence in the collected dataset obtained using GPR 52
(a) Left Tool Opening Angle 52
(b) Right Tool Opening Angle 52

4.1 Example of a prismatic articulated object. 54
4.2 Demonstration of articulated joints considered in this work at two different time steps. Revolute and prismatic joints are 1 DOF joint while motion on a plane is a 2 DOF joint. 58
4.3 Graphical Model of the general SLAM problem. The known nodes are darker than the unknown nodes. 67
4.4 Estimation of configuration parameters for a 2D landmark in revolute motion centered at point $(2, 2)$ with radius 1. Gaussian noise of 0.01 variance in both X and Y directions. Joint estimation yields a revolute motion centered at $(2.18, 2, 21)$ with a radius of 0.83, while separate configuration estimation yields a revolute joint centered at $(2.05, 1.88)$ with a radius of 1.10 74

4.5 Comparison of EKF filtering based state estimation for various orders of a motion parameter. For displaying purposes, we only show the zeroth order derivative state from all the different motion models. 75
(a) Zero Order 75
(b) First Order 75
(c) Second Order 75
4.6 Frames at different time intervals of our simulation. Color of a landmark at a particular frame is the weighted sum of colors assigned to each motion model. The weights used are the probability of the landmark following that particular motion model and estimated by our algorithm. We also show the predicted trajectory of a landmark according to the estimated motion model. . . . 76
4.7 Demonstration of Articulated EKF algorithm at various time steps. At each time step, we plot the robot's true state with a triangle, and the estimation of the robot's mean and covariance SLAM states is shown by an ellipse. 77
(a) Frame 1 77
(b) Frame 6 77
(c) Frame 8 77
(d) Frame 9 77
(e) Frame 41 77
(f) Frame 42 77

5.1 An example image from MS COCO with its associated captions . 80
5.2 Tracked Person with Generated Sentence 'Person moves downwards' 81
5.3 Tracked Person and Cart with Generated Sentence 'Person interacts with cart.' 81
5.4 Generated results of Open/Close attribute on surgical videos . . . 82
5.5 Our overarching goal is to improve human-robot/robot-robot interaction across sensing modalities while aiming at generalization ability. Multi-modal compositional models are important for effective interactions. 83
5.6 Mapping the physical concepts from visual domain such as color, texture and shape to the spoken language domain 87
5.7 Fourier Representation of a triangular shape with 2 and 10 fourier harmonics 90
(a) Segmented Image 90
(b) 2 Harmonics 90

(c) 10 Harmonics 90
5.8 For the audial utterance *blue halfcircle*, (a) generated image by mapping from audial to visual domain. (b), (c) retrieval of top 4 color and shape neighbors by both models. For both the models, a feature representation of audial utterance is mapped to visual representation which includes color and shape representation. This visual representation is then used to find the nearest neighour in the entire dataset to generate the color and shape neighbors in (b) and (c) respectively. 93
(a) Map 93
(b) Color Neighbours 93
(c) Shape Neighbours 93
(a) Map 93
(b) Color Neighbours 93
(c) Shape Neighbours 93
5.9 Comparison of correct retrievals by two different algorithms compositional and non-compositional. Left image shows the retrieval of shape features, while right shows that of color. 94
5.10 Generalization performance result depiction for audial utterances (a) *blue circle* (b) *green rectangle* (c) *red square* (d) *yellow halfcirle* . . . 95
(a) Blue Circle 95
(b) Green Rectangle 95
(c) Red Square 95
(d) Yellow HalfCircle 95
(a) Blue Circle 95
(b) Green Rectangle 95
(c) Red Square 95
(d) Yellow HalfCircle 95
5.11 Confusion matrices for generalization experiments evaluated by human subjects. Rows are for different features: colors and shapes. Columns from left to right are four different experiments (1) Images generated by compositional model are evaluated by humans with *unbiased* questions like "Describe the color and shape of this image" from fixed set of choices (2) Paired model with *unbiased* questions. (3) Compositional model with biased questions like "Is the shape of generated image same as the given example image?" 4) Paired model with biased questions. 96

6.1 Surgical semantic feedback from tracking and attribute generation. Two different surgical instruments “Tool 1” and “Clamp” are detected and tracked in images. This tracked bounding box is used in an attribute generation framework, such as open/close, which can be used as feedback to the surgeon. 103

Abstract

Articulated objects have components that are joined together with a kinematic joint which allows them to move with respect to each other. As robots move from industry floors to indoor environment and work in collaboration with humans, it is vital for robots to understand the articulated structure of the environment. For example, to open a door, a robot needs to find a door in the environment, estimate the rotation axis and then take appropriate control action to open the door.

We consider the hierarchy of representation of articulated objects starting from a bounding box to pose and further to finding the articulation itself using only the vision sensors (RGB/RGBD cameras). For object tracking using bounding box representation, we propose Product of Tracking Experts Model to use various object trackers that focus on specific motion and appearance characteristics of the object. For pose estimation and tracking, we propose an observation model using Gaussian Processes which is combined with motion-continuity models to track object pose over time. We show connections to the human language output that can be extracted from each level of representational hierarchy. Towards the end we demonstrate how language itself can be used to help vision by exploiting the compositionality of language. The thesis will present various applications ranging from surveillance, surgical safety feedback to Simultaneous Localization and Mapping (SLAM) in dynamic environments.

Chapter 1

Introduction

As robots move into our indoor environments, it is vital for them to have the visual sensing and reasoning capabilities to effectively communicate and collaborate with humans. Coupling visual sensing with the subsequent inference, computational vision techniques provide a critical tool for semantic understanding of the environment. As a result, enabling autonomous agents, along with the curiosity to understand human vision, has motivated a significant amount of research in computer vision. From a sensing perspective, there has been a steady growth in the diversity, complexity, breadth and capability of sensing modalities. Vision sensors are preferred over other sensors because they are cheap and have a wide field of view with fast update rates and just like human vision, these sensors allow us to passively sense a scene compared to other sensing modalities such as tactile sensing which requires contact. On the computational front, we have come a long way from early figure-ground analysis [110] to face detection [146], vision based autonomous driving [49], and optical navigation of extraterrestrial vehicles.

Perhaps, the greatest advances are yet to come from exploiting the spatio-temporal and Markovian structure inherent in physical world. The world, as we observe, is Markovian in nature. The historical state along with the current action gives us an intuition about what might happen in the future. While there is a significant body of work to incorporate temporal aspect, there are still major

gaps in the way temporal information is integrated with the static aspect of visual data. The major datasets in computer vision, such as Pascal VOC [39], Imagenet [129], and NYU-Depth [104], have enabled tremendous progress in all the major realms of computer vision including image classification[79], object detection [51], and image segmentation [104]. However, these datasets all consist of static images due to which a lot of temporal context is lost. Simple extension of static image techniques to videos, such as Histograms of Oriented Gradients (HOG) to HOG-3D [72], Convolutional Neural Network (CNN) in $2D$ to CNN-3D [62] have been proposed. Even the quintessential temporal tasks, such as object tracking, which involve finding temporally consistent tracks of objects in videos have been predominantly addressed by employing appearance models [160]. Although, adding the temporal context via motion models has been shown to improve tracking [27]. Indeed, if a picture is worth a thousand words, a video should be worth millions. *In this thesis, we demonstrate how incorporating the spatio-temporal aspect with static visual processing enables an utilizable interpretation of the physical environment.*

Another important aspect that has been missing in contemporary vision research is the use of structure, specifically articulated structure. A simple way of thinking about articulated structure is to relate it with the physical constraints on the motion of an object. For example, automobiles are constrained to move on a planar surface. These constraints on an object's motion or, in other words, the articulated structure is often implicitly known if an object can be visually identified. Even the human pose estimation problem routinely assumes the knowledge of articulation of the various limbs of the human body. However, for unidentified objects or for robots to understand the physical world without any prior knowledge, we need to propose a framework to understand articulations.

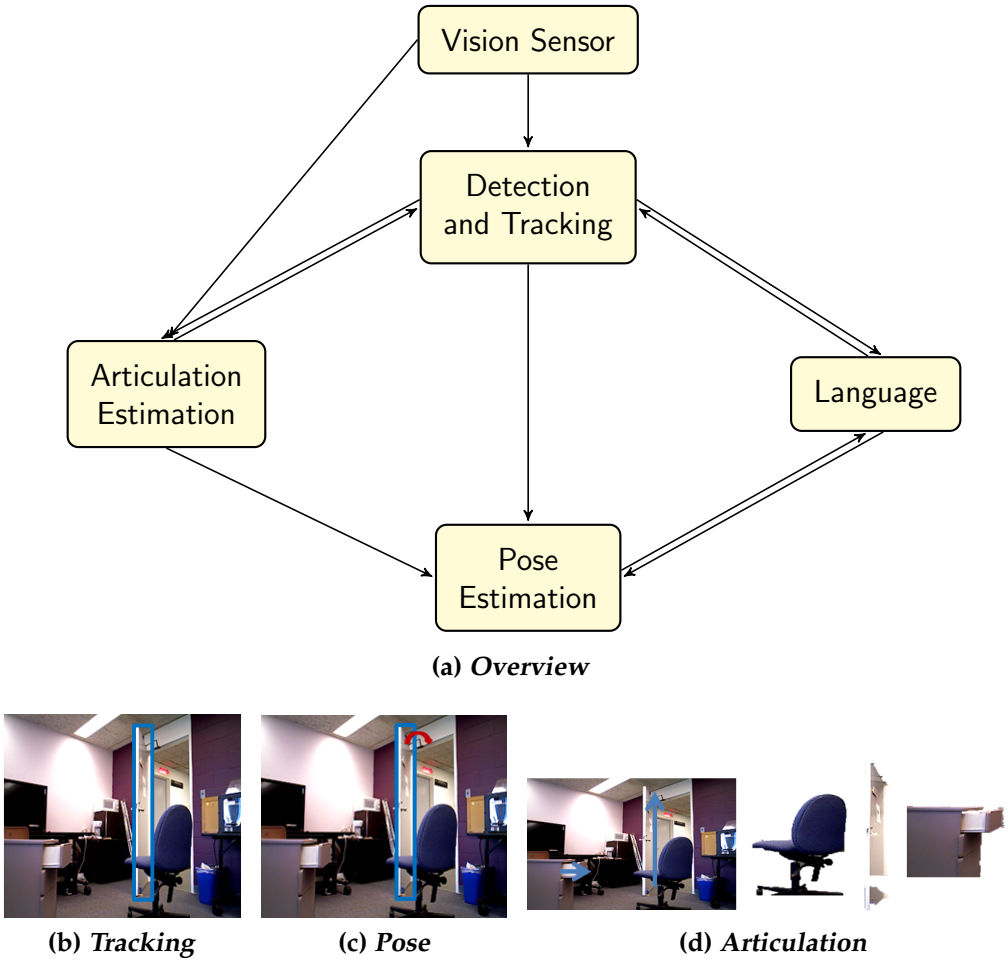
At the outset, we define articulated objects as the objects that have parts that can move with respect to each other (e.g. doors, humans, automobiles) in contrast to static non-articulated objects (e.g. buildings, rivers). For example, to know the location of a door, one only needs to measure the angle of the door about a hinge instead of knowing the full 6 degrees of freedom afforded to a rigid object. It is well known that knowledge of structure/representation helps

in a variety of inference problems in robotics and machine learning [35]. Structure not only enables prediction (for example, where the door might be at the next time instant?) but also in identifying the useful parts of incoming sensor data. Deeper understanding of articulated objects is critical for a variety of other computer vision problems. For example, articulated body pose estimation provides an useful mid-level feature representation for traditional computer vision tasks such as action recognition [162]. Yao et. al [169] obtained an improved accuracy for action recognition even with noisy pose estimates.

In this thesis, we explore, exploit, demonstrate and evaluate how an understanding of articulated structure can enhance a robot's performance of various tasks.

1.1 Organization

The overarching problem addressed in this thesis is to endow robots equipped with a vision sensor with the ability to understand articulated objects and communicate with humans in natural language about those objects and the environment. This problem spans multiple research frontiers in computer vision, machine learning, robotics, artificial intelligence and natural language processing. To elucidate this point, consider an autonomous robot following a human command "Open the door". A robot without any prior knowledge would first need to know how to find a door (object detection), keep track of the position of the door (object tracking), understand the word "open" (pose estimation), infer the structure of human language (language understanding) and then act while maintaining its position with respect to the door (Simultaneous Localization and Mapping (SLAM)). An added complexity to the problem would be if the robot had no previous knowledge to identify the structure of the door in which case the robot would need to carry out articulation estimation. It must be noted here, that the vision sensors that we consider are ones that can generate a stream of color (and depth) images. Figure 1.1 schematically depicts the problems addressed in this thesis including detection and tracking, pose estimation, articulated estimation and representation in language. In this thesis, we exploit the nature of articulated objects to implement the quintessential "sense-think-act" paradigm for robotic systems in unstructured environments.



(a) ***Overview***

(b) ***Tracking*** **(c)** ***Pose*** **(d)** ***Articulation***

Figure 1.1: Overview of the overarching problem addressed in the thesis. Streaming data from visual sensors can be used to identify the objects present in the scene or understand articulated structure without any prior knowledge of the objects. However, typically articulated structure is implicitly estimated via detection of an object category. Both tracking and pose output can be exploited to generate language which can in turn be used to look for specific objects or poses in the scene, (b) Demonstration of "door" detection, (c) Exemplar of pose estimate of the door which is directly related to its language representation "Open" and (d) Shows the information extracted from the environment based on articulation estimation which separates objects based on motion and finds their motion axis.

This thesis is organized as follows. Chapter 2 describes our work in articulated object detection and tracking using only a RGB camera. Detection and tracking refers to finding a bounding box around an object consistently over time in an image stream. Object tracking can be primarily divided into single

object tracking and multiple object tracking. Single object tracking methods attempt to maintain a temporally consistent bounding box around the object in image plane while dealing with appearance changes and noise in environment [160]. The typical approach is to capture essential parts of the object's appearance that are invariant to noise, easy to detect and match [170]. Instead of designing a perfect tracker, we track objects by combining outputs from multiple single object tracking algorithms. Multiple object tracking approaches on the other hand explicitly model interaction between various objects apart from using single object tracking mechanisms. There are two predominant approaches; i) tracking-from-detection [115] which attempts to stitch together temporally consistent tracks of objects by using large number of detection boxes, and ii) bottom-up methods that model the type of constraints in the scene to generate tracks of objects [21]. We propose a multiple state-space based interaction model to constrain the uncertainty associated with groups and resolve the ambiguities given sufficient observations.

In Chapter 3, we take the resulting bounding box around an object and get a pose estimate of the object using the articulation model known *a priori*. Pose estimate here refers to configuration of joints in an articulated body, such as, angle of a door, location of chair on a ground plane. The pose estimation research can be divided into two major parts based on the end result, i) $2D$ approaches that seek to find extents of each joint of articulated body in an image [119], and ii) $3D$ methods that find configuration of joints in $3D$ space either by using joint angles or Euclidean coordinates [6]. Each of these types of pose estimates can be obtained by using a model based optimization approach [1], learning based methods [90] or a combination of these two methods. Model based optimization methods assume a model of the articulated object and fit that model against incoming data to find the configuration of articulated object that best explains the data. On the other hand, learning based methods seek to learn a functional mapping from data to pose using ground truth data. We propose developing a learning based algorithm that generalizes on the novel data by using a motion model consistent with temporal evolution of pose.

In Chapter 4, we breakdown the assumption of knowing the articulated model beforehand and estimate the model using the image stream. We demon-

strate that it is possible to use these estimates and subsequently the resulting pose estimate to map dynamic scene. Johansson demonstrated the efficacy of human vision in performing articulation estimation by the moving lights display experiment [64]. The results encouraged a significant amount of research into replicating the human-like performance for articulation estimation. One of the earliest methods of articulation estimation used structure from motion to estimate the homogeneous transformation between camera and a rigid object and then build kinematic chains [165]. Recent introduction of depth cameras has allowed us to skip the structure-estimation problem and directly focus on the motion in the scene [114]. We take a multiple model approach which resolves the articulation model given sufficient information from observations.

Chapter 5 summarizes our early attempts at generating a language output and subsequently using the compositional nature of the language to drive vision. Inspired by the recent performance boost on image classification tasks using Convolutional Neural Networks (CNN), a number of researchers have explored the use of CNN for extracting features and converting images and videos to language [145]. One of the key benefits of language is the natural compositional ability to stitch together ideas about the physical world by combining words. This compositional ability has been harnessed to ground language to physical perception [102]. In contrast, we propose a generative model which can exploit the compositional nature of language to develop a invertible mapping between vision and language.

In the final chapter, we list some applications of the work outlined in this thesis and directions for future research.

1.2 Impact

1.2.1 Research Contribution

The major research contributions that were made as part of this thesis are as follows:

Product of Tracking Experts

We proposed a Product of Tracking Experts (PoTE) model for probabilisti-

cally merging results from various time-varying probabilistic/non-probabilistic trackers.

Gaussian Processes Based Pose Tracking

We proposed a real-time pose tracking algorithm by only using visual data with Gaussian Processes Regression and motion continuity model integrated into a Kalman Filter.

Articulated Structure Identification

We proposed an articulated structure estimation algorithm that explicitly uses temporal models which enable prediction. This is demonstrated by its incorporation into an algorithm for dynamic environment Simultaneous Localization and Mapping (SLAM).

Compositional Sparse Learning for Language and Vision

We proposed a framework for exploiting compositional nature of language to enable vision and vice-versa for robots to understand human language as manifested via physical world.

1.2.2 Selected Publications

A full list of publications can be accessed at the Google Scholar page of the author at `https://scholar.google.com/citations?hl=en&user=rLA6HfUAAAAJ`. Here we list the significant publications that have been used in writing of this thesis.

1.2.2.1 Journal

1. Suren Kumar, Pankaj Singhal, and Venkat Krovi. Computer-vision-based decision support in surgical robotics. *IEEE Design and Test*, 32(5):89–97, Oct 2015

2. Priyanshu Agarwal, Suren Kumar, Julian Ryde, Jason J Corso, and Venkat N Krovi. Estimating dynamics on-the-fly using monocular video for vision-based robotics. *IEEE/ASME Transactions on Mechatronics*, 19(4):1412–1423, 2014

3. Suren Kumar, Javad Sovizi, and Venkat N Krovi. Motion models and gaussian process regression based observation model for pose estimation. *In Preparation*, 2015

1.2.2.2 Refereed Conference

1. Suren Kumar, Javad Sovizi, M.S. Narayanan, and Venkat Krovi. Surgical tool pose estimation from monocular endoscopic videos. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 598–603, May 2015

2. Suren Kumar, Madusudanan Sathia Narayanan, Pankaj Singhal, Jason J Corso, and Venkat Krovi. Surgical tool attributes from monocular video. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4887–4892. IEEE, 2014

3. Suren Kumar, Vikas Dhiman, and Jason J Corso. Learning compositional sparse models of bimodal percepts. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014

4. Suren Kumar, Madusudanan Sathia Narayanan, Pankaj Singhal, Jason J Corso, and Venkat Krovi. Product of tracking experts for visual tracking of surgical tools. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 480–485. IEEE, 2013

5. Priyanshu Agarwal, Suren Kumar, Julian Ryde, Jason J Corso, and Venkat N Krovi. Estimating human dynamics on-the-fly using monocular video for pose estimation. In *Robotics: Science and Systems*, 2012

6. Priyanshu Agarwal, Suren Kumar, Julian Ryde, Jason J Corso, and Venkat N Krovi. An optimization based framework for human pose estimation in monocular videos. In *Advances in Visual Computing*, pages 575–586. Springer, 2012

1.2.3 Code Released

Object Tracking

Our code for detection and tracking of multiple objects was privately released to UC Berkeley and Toyon Corporation as part of performing teams of DARPA Mindseye Program.

Language and Vision

Our code for relating vision and language using compositional sparse models has been publicly released at `https://bitbucket.org/surenkum/bimodal_sparse`

Articulation Detection

Our code for articulation structure detection and EKF articulation SLAM will be made public upon publication at `https://bitbucket.org/surenkum/aae`

1.2.4 Dataset Released

Detection, Tracking and Attributes

We released a dataset consisting of 12 videos of real surgeries performed using Intuitive Surgical Inc.'s Da-Vinci robot with manual annotations for tools and their attributes (open/closed and Blood Stained/ Not Stained). This data and associated code is released under GNU General Public License and can be downloaded from `http://mechatronics.eng.buffalo.edu/research/rMIS_SkillAssessment/PoTE_DataSet.html`

Chapter 2

Articulated Object Detection and Tracking

2.1 Introduction

Articulated object detection and tracking is one of the most fundamental and essential component of any semantic understanding framework. The task of detection and tracking refers to finding a temporally consistent $2D$ bounding box around an object in the image. They are two important problems in tracking multiple objects in videos, i) Tracking a single object with time-varying appearance and (self-)occlusions in the environment and ii) Modelling and resolving individual object tracks upon interactions between multiple objects.

There are two important components of a single object tracking algorithm, motion propagation and appearance model. Motion model predicts the probable locations of the object in next frame given the data upto the current frame. Appearance model finds significant aspects of the object (logo, edges, points etc.) that enable separation of the object from the rest of the environment. Tracking algorithms use a combination of motion propagation model and appearance to first narrow down the search region by predicting locations of the object in the current frame and then fine-tuning that location by matching via appearance information. In this Chapter, we take a different approach to single object tracking. We propose to combine output from multiple single object trackers

which we call as experts to generate a track agreed upon by all the experts.

On modelling interactions between objects, we propose a time-evolving state-space based model for multiple object tracking, where multiple interactions potentially occur between tracked entities. We model entry/exit and interaction of multiple entities in unconstrained complex scenes using uncalibrated monocular camera. Our model does not make restricting assumption about type of interaction or behaviour during interaction by representing maximum associated uncertainty with the tracks given simple motion constraints. This results in a simple yet efficient representation of groups, formation and breaking of groups when observations are available.

2.2 Related Work

2.2.1 Single Object Tracking

Object tracking is a complex task due to self-occlusions, occlusion by other objects in the scene, entry and exit from scene, changes in illumination, collision between objects, motion blur, far scale and articulation in objects [170]. Significant advances have been made in single object tracking by building better appearance models using online boosting [52], online boosting with priors [53], generative and discriminative combination for adapting [68], multiple instance learning [8]. For further details on single object trackers, we refer the reader to a recent survey article [160].

2.2.2 Multiple Object Tracking

Single object tracking algorithms do not naturally extend to multiple object tracking perspective. Typical multi object tracking algorithms use a *bottom-up approach* for target representation and localization while coping with changes in the appearance of the tracked targets, or a *top-down approach* where data-association and filtering is performed to deal with object's motion. Ramanan et al. [119] present a bottom-up approach which first builds a model of appearance of each person in a video and then tracks by detecting this model in

each frame. Top-down approaches model the task of multiple object tracking as a network flow optimization problem employing techniques such as dynamic programming [16], multiple hypothesis tracking [121], linear programming relaxation [63]. However, there is a fundamental assumption of being able to reliably detect all the objects in each frame, even in unconstrained environments. This assumption is not trivial given state-of-the-art in object detection on unconstrained videos. Dollar et al. [32] observed degradation of performance of all state-of-the-art detectors under partial occlusion or when human is at far scale when tested on Caltech Pedestrian Dataset. Although training on a particular dataset helps, it comes at the risk of being too specific to a dataset. Using motion saliency for detection free segmentation of people in a video [46] has shown promise, but such methods would fail for stationary objects.

Tracking for detection using hierarchical Gaussian process latent variable model (hGPLVM) for explicitly modeling temporal coherency in a walking cycle has also been used to perform coupled detection and tracking [5]. However such techniques do not generalize well to videos which are unconstrained in terms of the activity being performed in the scene. Other recent work has focused on merging the bottom-up and top-down approaches using discriminative appearance-based affinity modeling with hierarchical tracklet association [92], and tracking-by-detection using class-specific pedestrian detector to localize people along with a motion-model-based particle filtering to predict target locations [21]. Fundamentally, these approaches do not model interactions and rather treat them as noise.

To model interactions, [58] proposes using a social force model to explain the change in behavior of a pedestrian due to environment and other people. Linear Trajectory Avoidance (LTA)[112] proposes predicting expected point of closest approach and use this estimate to model decision making of pedestrians. These approaches seek to actively model the intentions of pedestrians. Motion and learning based methods for interactions have also been proposed for tracking. Kratz et. al [76] learns Hidden Markov Models (HMM) on motion patterns in a video by dividing into spatial locations. Xiong et. al[161] proposes estimating the location of a target from supporting features by learning autoregressive models. However, these approaches involve many parameters that either need

to be learned apriori or estimated while tracking simultaneously. In contrast, we model interactions using a passive approach that reasons about interactions by increasing the associated position uncertainty. We perform inference and thus reduce position uncertainty only when suitable observations are available. We model interaction between entities using a adaptive state based representation of tracking and reinforce appearance-based cues to resolve identity, at the track level which is robust to noise in individual frames. This circumvents the problem of tracking individual entities within a group during interaction phase.

2.3 Research Contribution

Point Features Based Tracking - We introduce entity tracking using point feature correspondences by voting-based strategy to establish geometry of the tracked bounding box. This voting-based strategy is particularly effective due to the large number of features being tracked thus, effectively reducing the effect of noise in point feature tracking.

Product of Tracking Experts - We introduce a model to probabilistically merge outputs from various single object trackers probabilistically.

State Space for Interactions - Since the state of the world is not known beforehand, the best we can do is dynamically estimate the actors/agents in the scene and then track them over time. However, a fixed and concrete tracking system would not be able to adapt to the various general ways in which the people/objects can be interacting within the scene.

The elegance of dynamically adapting the state space as new frames are processed is that we can add a rich set of top-down constrains within a common framework. The type of constraint that we explore in this paper is the group interaction and then separation, but other rich constraints are also possible. For example, a person walking his/her dog or a man pushing a cart, or a woman carrying a bag.

Overall our tracking framework with its associated model does not use calibrated/stationary camera, scene model, knowledge about ground plane.

2.4 Product of Tracking Experts

We adapt a time evolving Product of Experts (PoE) [59] model to optimally fuse hypothesis from various trackers at each instant in time to propose a Product of Tracking Experts (PoTE). We consider each tracker $T_1, T_2, ..., T_K$ as experts for predicting the location of target center. Product of experts model for tracking ensures that the resulting model for track is explained by all the experts. A high level flow chart of proposed Product of Tracking Experts (PoTE) is shown in Figure 2.1. Our method bootstraps from high confidence detection to start tracks of various objects. We specifically obtain a very low false positive rate by increasing the detection threshold on learnt object detector.

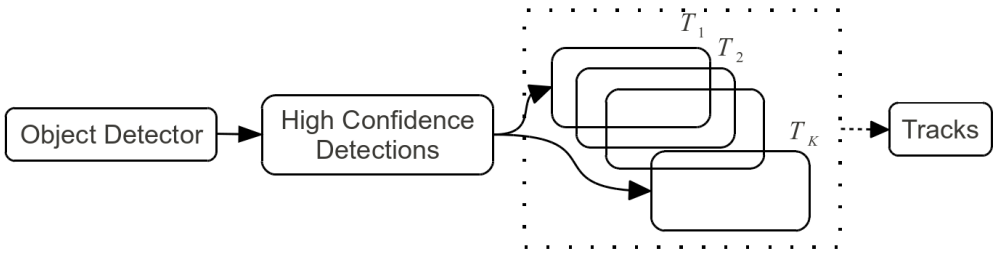


Figure 2.1: System Flow Diagram

Each entity is tracked independently by various trackers $T_1, T_2, ..., T_K$. These trackers could be based on either discriminative (data association techniques [45],detector confidence etc.) or generative (particle filter [108], Kalman Filter [155], KLT [97] etc.) along with a combination of generative and discriminative techniques [68], [21]. Tracking solely by using either generative or discriminative approaches in unconstrained scenes is hard because generative approaches make assumptions about the motion of entity whereas discriminative approaches make assumption of having a robust detector.

Let θ_k be the parameters associated with the probability distribution of each expert. Probability of any point $\underline{x}$ to be the true center of a bounding box as explained by all the expert trackers is given by Equation 2.1.

$$p(\underline{x}|\theta_{T_1}, \theta_{T_2}, ..., \theta_{T_K}) = \frac{\prod_{k=1}^{K} p_k(\underline{x}|\theta_k)}{\int \prod_{k=1}^{K} p_k(\underline{x}|\theta_k) \mathrm{d}\underline{x}} \tag{2.1}$$

Denominator in Equation 2.1 is a normalization constant and can be ignored to choose best $\underline{x}$. This model ensures robust tracking because it allows to incorporate (or leave out) arbitrary number of trackers. For example, for a discriminative classifier, detection score is commonly used to guide tracking. This classifier can be included in the tracking mix by modeling its distribution using an indicator function, which determines if this detection score is greater than a predetermined threshold. Tracking frameworks employing particle/Kalman filter provide a probability distribution as output which is ideal for this method. Additionally, an individual tracker is not required to give a probabilistic output but is only required to give a bounding box which could then be modeled as a probability distribution using Equation 2.9. Breitenstein et al.[21] propose using continuous class confidence density because current object detectors such as Histogram of Oriented Gradients (HOG) based detectors [29] provide a score at discrete spatial locations and scale. This could be easily incorporated in the presented framework. Hence, our proposed framework is an extensible method of combining results from different types of trackers.

If all the experts have normal distribution with $\underline{\mu_k}, \mathbf{\Sigma_k}$ as mean and covariance matrix, the resulting best location of center of bounding box $\underline{x}$ can be obtained analytically because product of independent normal distributions yields a normal distribution.

$$p(\underline{x}|\theta_{T_1}, \theta_{T_2}, ..., \theta_{T_K}) = \frac{\prod_{k=1}^{K} \frac{1}{2\pi|\mathbf{\Sigma_k}|^{\frac{1}{2}}} \exp(-\frac{1}{2}[\underline{x} - \underline{\mu_k}]^T \mathbf{\Sigma_k}^{-1}[\underline{x} - \underline{\mu_k}])}{\int \prod_{k=1}^{K} p_k(\underline{x}|\theta_k) \mathrm{d}\underline{x}} \tag{2.2}$$

The resulting Probability Density Function (pdf) can be obtained after some algebraic manipulation as

$$p(\underline{x}|\theta_{T_1}, \theta_{T_2}, ..., \theta_{T_K}) \sim \mathcal{N}(\underline{\mu}, \mathbf{\Sigma}), \text{ where}$$
$$\mathbf{\Sigma}^{-1} = \sum_{k=1}^{K} \mathbf{\Sigma_k}^{-1}, \underline{\mu} = \mathbf{\Sigma} \left(\sum_{k=1}^{K} \mathbf{\Sigma_k}^{-1} \underline{\mu_k} \right) \tag{2.3}$$

Intuition behind PoTE model is shown in Figure 2.2. We exploit combination

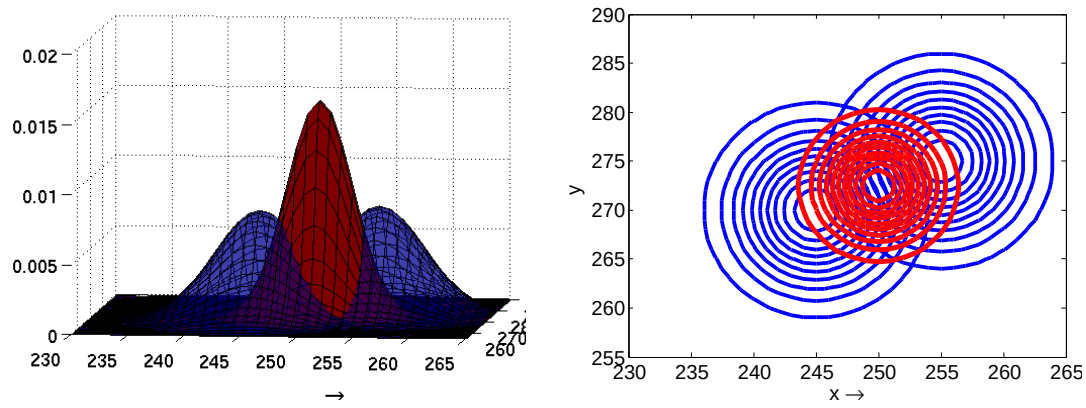


Figure 2.2: Two tracking experts on sides and resulting PoTE model in middle with Gaussian probability density function contours on right. First tracking expert has associated Gaussian with mean = $[245, 270]^T$ *and variance* = $diag([16.66, 25])$*, second tracker has associated Gaussian with mean* = $[255, 275]^T$ *with same variance, when combined using PoTE model results into a Gaussian with mean* = $[250, 272.5]^T$ *and variance* =$diag([8.33, 12.5])$

of various single object tracking algorithms to generate an overall track for a object. Now, we describe some of the individual tracking experts that we use in our work.

2.4.1 Object Detection

We use Deformable Parts Model (DPM) [41] based object detectors for articulated object detection from images. DPM object detector essentially captures the shape of the object by employing a model consisting of star-structured pictorial structure which links root of an object to its parts using deformable springs. Hence this model captures articulation and allows for learning a detector for various object configurations. DPM based detector (*det*) provides many bounding boxes BB_{det} along with their corresponding detection score τ as output. A detection is included in tracking experts mix only if it is deemed reliable, which is evaluated by an indicator function. This indicator function $\mathcal{I}$ is defined in

Equation 2.4 as

$$\mathcal{I} = \begin{cases} 1 & \text{if}(\tau \geq \tau_{thresh}) \wedge (BB_{det} \bigcap BB^{e}_{t-1}) \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

In Equation 2.4, τ_{thresh} is a predetermined threshold on detection scores for including only reliable detections. Additionally, only relevant detections are considered to track an object/entity e on current frame (time t) by evaluating whether a particular detected bounding box BB_{det} intersects with the bounding box of the entity in last frame (time $t-1$). If multiple detections have indicator function $\mathcal{I}$ as 1, we select the bounding box with the maximum score. Additional ways of selecting a bounding box could be based on velocity and size information [21]. Once a particular detection bounding box is selected, the detector is modeled as an expert by using Equation 2.9.

2.4.2 Point Feature Based Tracking

Kanade-Lucas-Tomasi (KLT) tracker, introduced by Lucas and Kanade [97], is a point feature tracker extensively used for computer vision tasks. This algorithm finds good spatial features to track by locating Harris corners in an image. To track a particular feature, a window centered on feature point in current image is matched in next image by Newton-Raphson method of minimization. We use Stan Birchfield's [18] implementation of KLT to achieve tracking of feature points. KLT based tracker uses bounding box of an object/entity in current frame to identify the region to be tracked in subsequent frames. Process of tracking using KLT is pictorially depicted in Figure 2.3.

To initialize the KLT tracker, we evaluate feature points inside the bounding box in current frame. Let $x_{BB} = [x_B(t), y_B(t), w_B(t), h_B(t)]$ is the axis aligned bounding box, where $[x_B(t), y_B(t)]^T$ is left top corner of bounding box in the image and $(w_B(t), h_B(t))$ specify width and height of the bounding box respectively at frame t. Geometric location of each feature point is obtained relative to top left point of the bounding box, thus encoding relative geometrical location

Figure 2.3: Flow diagram of tracking using KLT.

of all the features with respect to bounding box.

$$\begin{aligned} G_x(t,j) &= (x_B(t) - x_f(t,j)), \\ G_y(t,j) &= (y_B(t) - y_f(t,j)) \end{aligned} \tag{2.5}$$

$G_x(t,j), G_y(t,j)$ stores relative (x,y) location of j th feature on frame t. In second step, these features are tracked in next image using KLT. Each tracked feature carries the geometrical relationship from previous frame and votes for current location of bounding box by assuming that collection of large number of features can diminish the effect of noise in location of bounding box using Equation 2.6, where $(x_B(t+1,j), y_B(t+1,j))$ is location of the top-left corner of the bounding box as predicted by j th feature on frame $t+1$.

$$\begin{aligned} x_B(t+1,j) &= (G_x(t,j) + x_f(t+1,j)), \\ y_B(t+1,j) &= (G_y(t,j) + y_f(t+1,j)) \\ x_B(t+1) &= \sum_{j=1}^{J} w_j x_B(t+1,j), \\ y_B(t+1) &= \sum_{j=1}^{J} w_j y_B(t+1,j) \end{aligned} \tag{2.6}$$

where w_j is the weight associated with each feature point as obtained from the normalized objective residual in KLT minimization such that $\sum_{j=1}^{J} w_j = 1$. Width and height of the bounding box are updated based on the tracking output from the previous time step.

2.4.3 Background Subtraction

We use connected-component analysis on the background subtracted images to obtain probable location of an object in next image. All the resulting blobs are then filtered to remove noise by comparing their area with the input bounding box. Each blob ($i = 1, ...I$) is processed and considered probable in current frame only if its area is greater than a minimum ratio AR_{min} times the input bounding box area A_B and less than a maximum ratio AR_{max} times input bounding box area (2.7).

$$(A(i,t) > AR_{min} \times A_B) \wedge (A(i,t) < AR_{max} \times A_B) \tag{2.7}$$

where $\wedge$ stands for logical conjunction. Minimum and maximum ratio are kept relaxed so that the true blob is never lost. If no blob qualifies this filtering criteria, it implies that the object was stationary and hence its bounding box would be same as the bounding box in previous frame. If multiple blobs are found after filtering, the best blob is selected by area overlap ratio using (2.8).

$$\arg\max_{I} \frac{(A_{SBB}(i,t) \bigcap A_{BB})}{(A_{SBB}(i,t) \bigcup A_{BB})} \tag{2.8}$$

where $A_{SBB}(i,t)$ is the set of pixels belonging to bounding box of i^{th} blob in t^{th} frame, A_{BB} is the set of pixels belonging to the tracked bounding box in the previous frame.

2.4.4 Motion Continuity

We use a first order Markov dynamics model to predict the state estimate of entity E_i in next frame as E_i^+ . State of each entity E_i is represented by a bounding box $E_i \equiv [x_{CB}, y_{CB}, w_B, h_B]^T$ in each frame. We hypothesize location of the bounding box to be normally distributed in the image plane with its centroid $[x_{CB}, y_{CB}]^T$ as mean, and width (w_B) and height (h_B) as uncertainty in its loca-

tion (6 $\times$ variance).

$$\mu = [x_{CB}, y_{CB}]^T, \Sigma = \frac{1}{6}\begin{bmatrix} w_B & 0 \\ 0 & h_B \end{bmatrix} \tag{2.9}$$

We use a constant velocity motion model with position and velocity of the center of the bounding box as state. We model acceleration ($a = [a_x, a_y]^T$) as white noise ($a \sim \mathcal{N}(0, \mathbf{\Sigma_a})$), where $\mathbf{\Sigma_a}$ is estimated using the finite difference in a local temporal window.

$$E_i^+ = \begin{bmatrix} \mathbf{A} & [0]_{2\times2} \\ [0]_{2\times2} & \mathbf{A} \end{bmatrix} \begin{bmatrix} x_{CB}(t) \\ \dot{x_{CB}}(t) \\ y_{CB}(t) \\ \dot{y_{CB}}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} & [0]_{2\times1} \\ [0]_{2\times1} & \mathbf{B} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{2.10}$$

where $\mathbf{A} = \begin{bmatrix} 1 & \Delta t \\ 0 & 0 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix}$, and Δt is the time between two successive frames. It can be observed that the state transition matrix of discrete state space (Equation 2.10) is constant and hence the motion model as an expert has a normal distribution with center of bounding box in previous frame as mean and acceleration as standard deviation.

2.4.5 Dense Optical Flow

This tracker is based on extracting dense optical flow [23] and predicting the bounding box in next frame. Optical flow measures apparent motion of each pixel between two images assuming that its brightness remains constant in both images. We start tracking by using the detections with confidence measure above a given threshold τ_{thresh}. In each frame, we obtain the optical flow between two frames for all the pixels belonging to the desired bounding box. The location of bounding box in next frame is a result of flow in all the pixels and is approximated by the mean flow of all the pixels. Width and height of the bounding box is updated based on tracking output from the previous associated detection with current object track.

2.5 Multiple Object Tracking

Our approach to multiple tracking represents tracks of each entity with dynamically evolving states. We represent interactions between different entities by using state space techniques. Individual entity tracks are estimated using PoTE model which optimally fuse hypothesis from various trackers at each instant in time.

2.5.1 State Based Tracking Representation

We use a state based representation of entities being tracked which explicitly models interaction and entry/exit problem. Multiple target tracking based on state representation of entities is developed to allow a natural means for detecting collision and occlusion among multiple objects. This representation is able to handle complex interaction of entities throughout the video taking care of the entry and exit of each entity. A representation of multiple interacting entities is shown in Figure 2.4. Entities E_1 to E_{K_1} interact as the node $E_{1,2,\ldots,K_1,k+1}$ in frame $k+1$. Furthermore, additional $K_2 - K_1$ entities merge with these entities as the node $E_{1,2,\ldots,K_2,k+l_1}$ on frame $k+l_1$. All these entities are tracked collectively during occlusion and then split at the node $E_{1,2,\ldots,K_2,k+l_2}$ in frame $k+l_2$ and are uniquely tracked after occlusion. For the rest of this chapter, we drop the frame number from the notation of entities. The entities (E_j) that do not interact with other entities in the scene are tracked independently.

Multiple tracks are started from the frame with disparate object detections with high detection scores.

2.5.2 Occlusion Detection and Track Resolution

Occlusion detection is vital for generative tracking approaches as used in current implementation because otherwise tracking starts to drift and produce erroneous results. We develop an occlusion module which performs the tasks of occlusion detection and maintaining identity of an entity in case of an occlusion. Occlusion is detected when two tracks intersect based on pascal measure [32] as

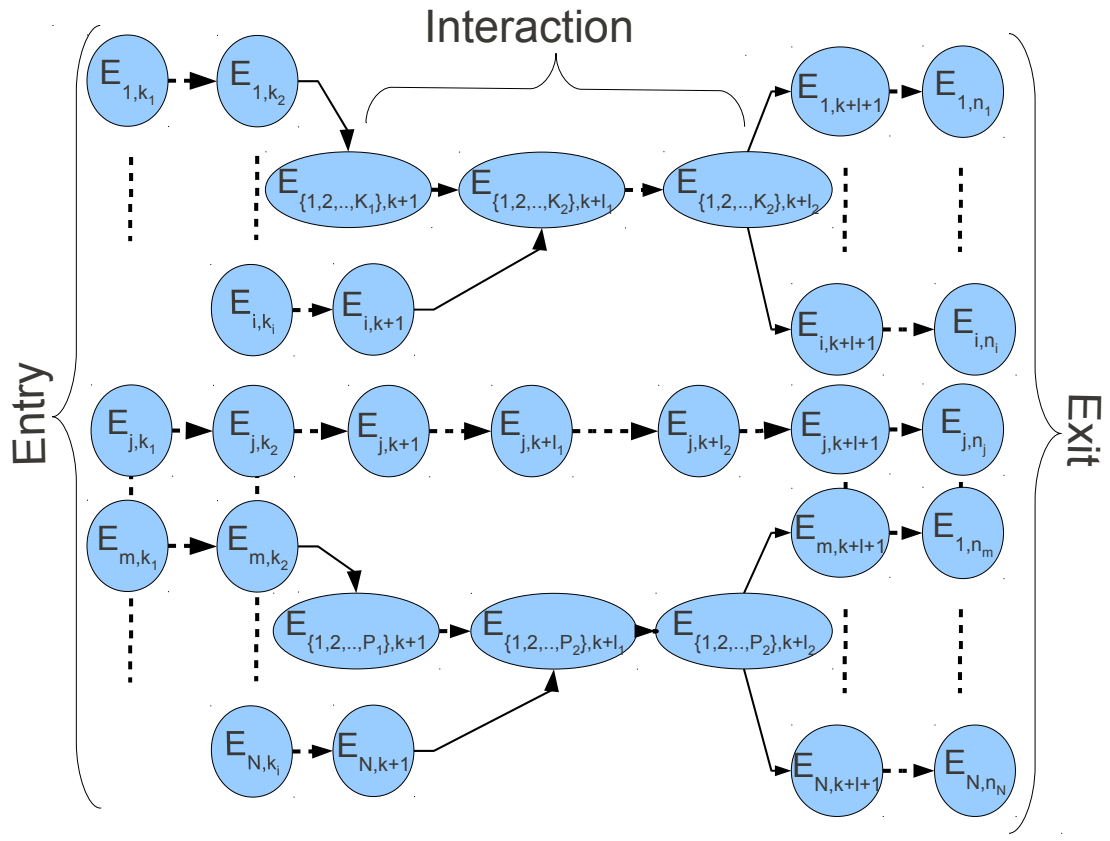


Figure 2.4: State based representation of entities in the tracking framework. Multiple group of entities might form which are tracked as groups, groups might split entirely into individual entities or few entities might leave the group.

in Equation 2.11, where E_S is set of all single entity states.

$$Pascal(E_i, E_j) = \frac{E_i \cap E_j}{E_i \cup E_j}, E_i, E_j \in \{E_S\} \tag{2.11}$$

All intersecting entities are represented as a combined state $E_k \in E_I$, where E_I is the set of interacting states. For a pre-existing group of nodes, we evaluate the area intersection ratio of an single entity using Equation 2.12, which evaluates the percentage of area of a single node that falls within a group node area.

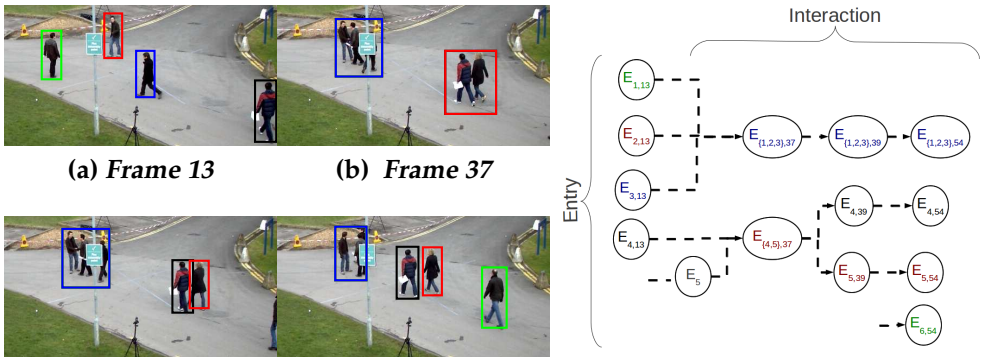
$$Aint(E_k, E_i) = \frac{E_k \cap E_i}{E_i}, E_i \in \{E_S\}, E_k \in \{E_I\} \tag{2.12}$$

This interaction model assists in imposing constraints on the states of fused nodes. We explore groups of objects interacting and then separating as the constraints imposed using interacting states. When a group of people intersect, we represent their state estimate as the combined uncertainty of all merged states.

$$\hat{E_k} = \mathrm{I}(\hat{E_i} \cup \hat{E_j} \cup ...), \forall i, j, .. \in k \tag{2.13}$$

We represent the position of fused node $\mathrm{I}(.)$ using a spatial Gaussian which spans $\mu \pm 3\Sigma$ of individual merged nodes. It is similar to representing the combined state as maximum entropy distribution of states. If a node in the k^{th} frame intersects with m other nodes, then all these nodes are merged into one sub-group node. Hence, all the individual nodes are replaced by this new node whose boundaries include extrema of all fused nodes. We predict the position of individual entities using the motion model of each individual node using the first order dynamics model as in Equation 2.10 and then spatially fuse state estimate using Equation 2.13.

This approach is similar to dynamics system approach where dynamics model is used to propagate states if no observations are available, which essentially keeps increasing the uncertainty in state [133]. Figure 2.5 depicts process of occlusion detection using intersections of different tracks. Frame 13 shows three different entities which are subsequently merged in frame 37. Each sub-group node stores the number of nodes and their prior tracks, that were merged in it. To resolve an occlusion problem, if any detections/observation O intersecting with a sub-group node is observed, this detection is associated to a prior track using appearance modeling. For the purpose of maintaining identity we solve the correspondence problem of tracks before and after occlusion using KL distance [118] on RGB histograms. RGB histograms for an individual entity track are updated at every instant of detected bounding box merged with the track. Furthermore, identity is only resolved when the number of objects before and after occlusion are same. The identity resolution as used in this thesis does not extend to all the possible scenarios and we refer the reader to recent person identification literature [4]. A sub-group node is completely fragmented if the number of detections intersecting with a sub-group node is equal to the number



(a) ***Frame 13*** **(b)** ***Frame 37*** **(c)** ***Frame 39*** **(d)** ***Frame 54***

Figure 2.5: Formation and fragmentation of tracked sub-groups in 'S2.L1 walking' scenario in the PETS 2009 Dataset S2 for People Tracking and the associated state space as explored in the video. (Please view in color). The Entities in flowchart on the right hand side are represented by the same color as the bounding boxes in frames on the left hand side. Note that because of significant overlap (E_1, E_2, E_3 combine to $E_{1,2,3}$ in frame 37) several entities combine to form a single entity which is collectively tracked. The separate entities may re-split from a grouped entity (as $E_{4,5}$ split to E_4 and E_5 in frame 39) given enough observations.

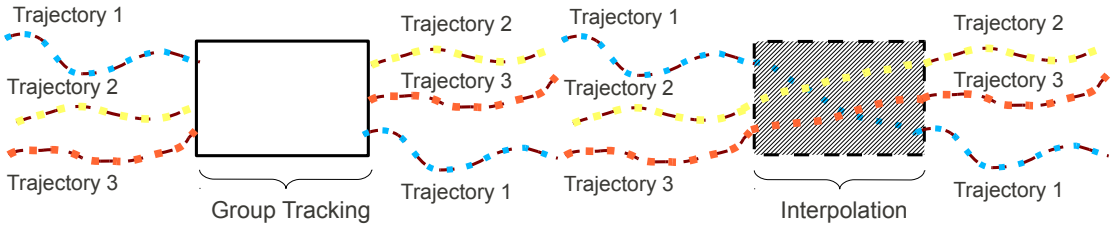
of nodes in a sub-group node.

2.5.3 Interpolation Based Trajectory Estimation

Figure 2.5 shows that when multiple objects interact in the scene, our algorithm first tracks sub-groups of the interacting objects (Coarse tracking). Figure 2.6 illustrates the interpolation based trajectory estimation framework. In order to interpolate, we first establish the identity of each object before sub-group formation and after sub-group fragmentation Figure 2.6a. We then interpolate the bounding boxes within each tracklet of sub-group nodes using constant velocity interpolation model Figure 2.6b. This model of group tracking is found suitable for all the tested datasets.

2.5.4 Entry/Exit Estimation

One of the challenging components in object tracking is the task of determining entry/exit from a frame. There are two ways in which an entity can exit/enter the scene, complete occlusion inside the frame by another object or background



(a) *Coarse Tracking* **(b) *Fine Tracking***

Figure 2.6: Interpolation based trajectory estimation to track objects through an interaction process

and exit from field of view of a camera. We use appearance- and motion-cues to estimate the entry/exit frame.

Motion cue modeled as velocity estimate of an entity using KLT is used to determine whether an entity became stationary or went outside of the field of view of camera. However motion cues alone cannot address the task because an object might appear stationary under a variety of circumstances,e.g. when completely occluded by another entity. So appearance cues are used to reinforce the decision about entry/exit of an entity, after motion cues have identified potential candidate entity for exit. Appearance is modeled using RGB histogram (16 bins in each color channel) of an image patch of the bounding box. We evaluate the histogram of the current, last detected bounding box on the current frame and the frame where last detection was merged with current track.

We use Kullback-Leibler (K-L) divergence as a measure of the distance between the two histograms. $score_1$ represents the KL distance between RGB histograms of bounding box of a track on last merged detection frame and current bounding box location evaluated on last merged detection frame. $score_2$ represents histogram distance between the track on last merged detection frame and current bounding box location evaluated on current frame. $score_3$ represents histogram distance between location of the track on last merged detection frame evaluated on current frame and current bounding box location evaluated on current frame. An entity is treated as exit from a scene if the distance ($score_2$) between the two histograms (detected object and the background) is greater than both $score_1$ and $score_3$.

2.6 Experiments

We experimentally evaluated our tracking algorithm on two different tracking tasks of surgical tool tracking and person tracking.

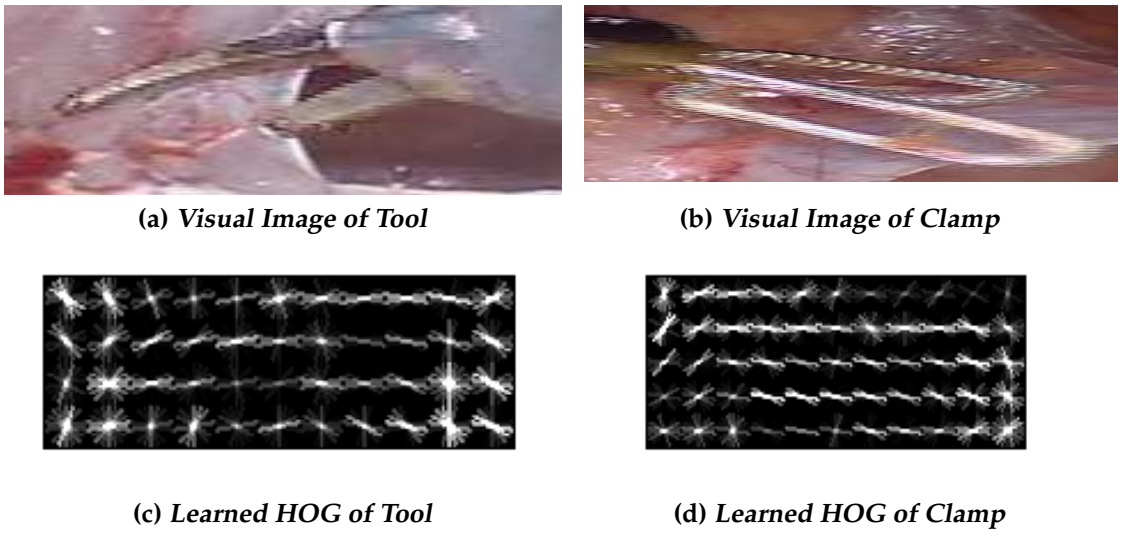
2.6.1 Surgical Tool Tracking

Increasingly surgeries are being performed by tele-operated devices on patients with remote manipulators through small incisions while providing the surgeon at master end with look-and-feel of an open surgery. Such robotic laparoscopic (or minimally invasive) procedures result in minimal pre- and post- surgical trauma and faster recovery for the patients. Recently there has been significant interest in video understanding techniques applied to recorded or online surgical video captured by tele-operated surgical devices. Tracking surgical tools in general has been used for a wide range of applications including safety, decision-support as well as skill assessment. To the best of our knowledge, there are no publicly available datasets for testing our tool tracking algorithm. Hence, we propose a new dataset consisting of 8 small sequences (1500 frames) for "Clamp" class and 8 sequences (1650 frames) for "Tool" class acquired while performing Hysterectomy surgery using da Vinci Surgical System (dVSS) to conduct our evaluation.

The proposed dataset has real-world video-sequences with various artefacts including tool articulations, occlusions, rapid appearance changes, fast camera motion, motion blur, smoke and specular reflections. This dataset was then manually annotated for the bounding boxes of the tools in every frame. The overall accuracy of our PoTE method is then evaluated using standard performance measures [100] by calculating True Positive (TP), False Positive (FP), True Negatives (TN) and False Negatives (FN). We treat a bounding box in image to be True Positive if the pascal measure (ratio of area intersection and area union) in image frame is greater than 0.5, which is commonly used for measuring accuracy of object detection detection methods [32]. The overall accuracy is evaluated as a ratio of sum of true positives and true negatives to a sum of true positives, true negatives, false positives and false negatives.

We learn a detector for both tool classes using HOG based object detection

methods [41]. Figure 2.7 shows HOG template model learned using ground truth annotations for both the tool classes. We test a baseline tracker using detection and KLT tracking for both tool classes on the proposed dataset and PoTE tracker with detection, optical flow and KLT as experts. The method with detection and KLT tracking is an example of single object tracking method where the tracker needs an initialization which is provided by detected bounding box in our case.



(a) ***Visual Image of Tool*** **(b)** ***Visual Image of Clamp***

(c) ***Learned HOG of Tool*** **(d)** ***Learned HOG of Clamp***

Figure 2.7: Learned HOG Templates with a representative bounding box for different tool types

Tool Type	Baseline	PoTE
Clamp	68.27%	**75.81**%
Tool	28.04%	**63.31**%

Table 2.1: Accuracy using Baseline and PoTE tracker

As shown in Table 2.1, our algorithm outperforms the baseline method on this challenging dataset for both the tool types. Baseline method's performance worsens on "Tool" class because of rapid perceived motion associated camera pose/zoom changes, articulation and tool motion. "Clamp" is usually kept stationary in surgery to hold the tissue while "Tool" is used to perform tissue cutting as can be seen from results in Figure 2.8.

The key benefits in our current implementation ensue from complementary nature of the two constituent probabilistically merged approaches – Point feature based tracking, which is robust for small motion and Region based tracking, which works well in case of significant motion. As a result, our tracker shows robust tracking for scenarios which involve unconstrained surgical tool activities as shown in Figure 2.8.

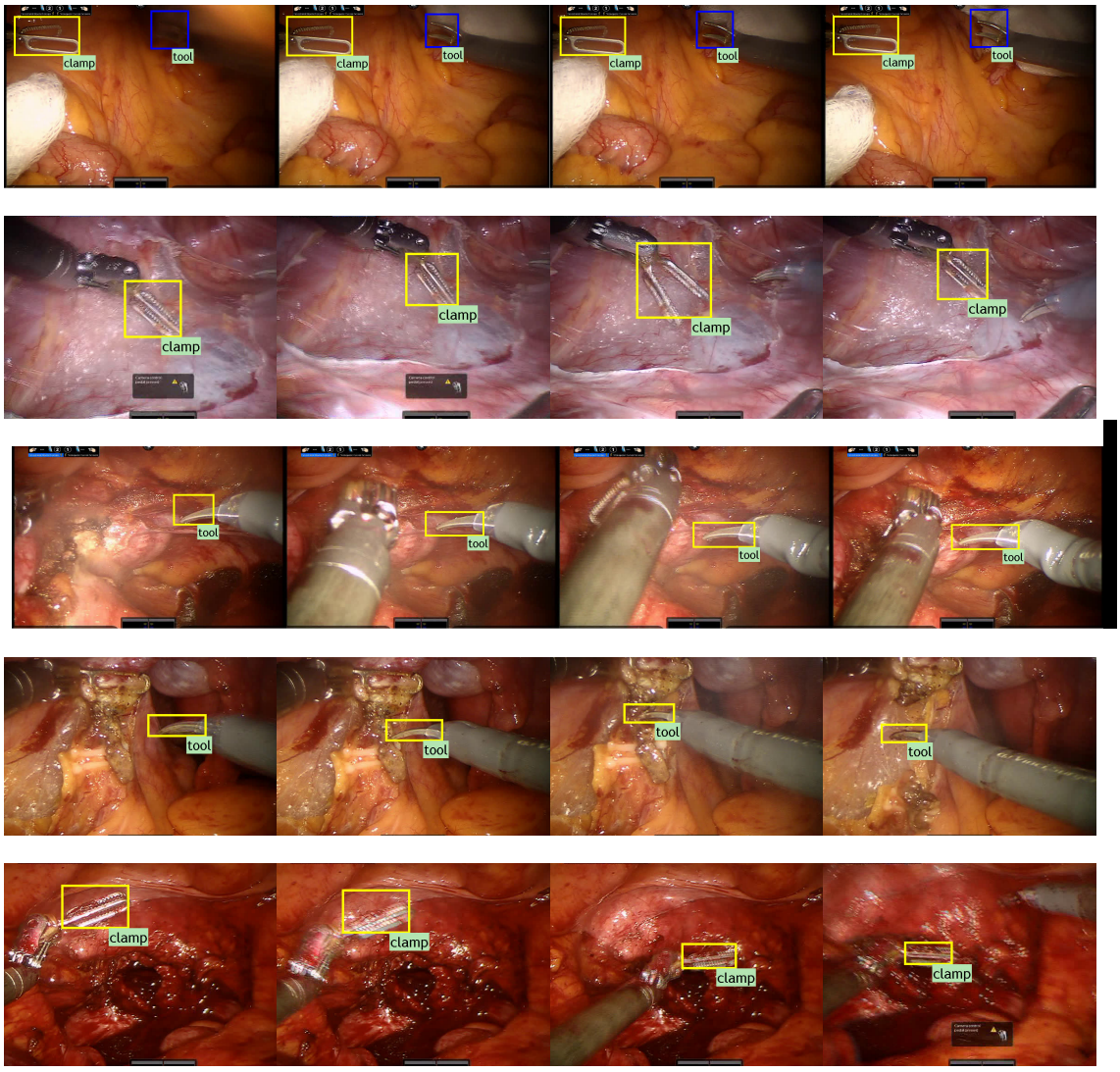


Figure 2.8: Tracking results for "Tool" and "Clamp" on various surgical operation videos in proposed dataset. (Please view in color)

2.6.2 Person Tracking

Person tracking is most well-studied articulated object tracking with obvious applications to surveillance, designing human-computer interfaces and extracting semantic information from videos and images. We use PoTE model with tracking experts based on object detector [41], KLT tracking [97], blob tracking [56] and motion propagation. We evaluate the proposed framework on the PETS 2009 and evaluated quantitative metrics to benchmark our algorithm against the state-of-the-art trackers.

PETS 2009 dataset - Performance Evaluation of Tracking and Surveillance (PETS) dataset[1] is a recent and popular dataset used for evaluating tracking performance. We use single camera view, View_001 (S2.L1 walking sequence) which has multiple interactions between various humans in the scene. We generate detection bounding boxes using poselet's based detector [20] which detects approximately 45% of all people with a detection threshold of 5.6, out of which 12% are overlapping with each other.

We follow current performance evaluation standard for tracking and use CLEAR metrics [17]. The multiple object tracking accuracy (MOTA) metric takes into account number of misses, false positives and number of mismatches, computed over total number of objects in all frames. The multiple object tracking precision (MOTP) measures the ability of tracker to precisely locate an object position in image frame. For measuring precision, we use PASCAL measure [32] with area overlap threshold of 0.5. We also evaluate performance metrics proposed by [158], which counts the number of mostly tracked (MT), partially tracked (PT) and mostly lost (ML) trajectories along with the number of track fragmentations (FM) and identity switches (IDS), to compare our performance against [7].

Table 2.2 shows that our results are comparable to state-of-the-art results. Figure 2.9 shows the visual tracking results. Frame 16 shows that a person is detected (red bounding box) which is then back-tracked in the preceding frames and is found to exit the scene in Frame 13. Frames 13-18 depicts that our motion-model based tracker works well with significant occlusion in the scene. Frames

[1] Available: `http://www.cvg.rdg.ac.uk/PETS2009/a.html`

Table 2.2: Summary of quantitative accuracy results on PETS 2009 Dataset (S2.L1 walking scenario).

Method	MOTP	MOTA	MT	PT	ML	FM	IDS	FN	FP
Yang et al. [166]	53.8%	75.9 %	-	-	-	-	-	-	-
Breitenstein et al. [21]	56.3 %	79.7 %	-	-	-	-	-	-	-
Andriyenko et al. [7]	76.1%	81.4 %	19	4	0	21	15	-	-
Proposed Method	**76.25%**	**82.8%**	20	3	0	14	12	-	-

121-132 illustrates that our identity resolution algorithm is able to accurately track the humans even in case of full occlusion. However, our constant velocity based interpolation (Section 2.5.3) for fine tracking within sub-groups fails when complex human motion is present within the sub-group (Frames 26, 73).

2.7 Discussion

In this section, we discuss key insights observed about single entity tracking model, interaction model and current implementation of this model. We also discuss how the individual parts of generative tracking methods affect the results.

2.7.1 Product of Tracking Experts (PoTE) Model

Good results on all the tested datasets show the effectiveness of proposed product of expert model. Using the Equation 2.9 to model an expert might not be generalizable to trackers which give noisy underestimated bounding box as output , due to biasing towards experts with smaller bounding box sizes (reduced $\mathbf{\Sigma}$). This was not the case in the current implementation because we merge only high confidence score detections. However, for a more general case, variance of an expert could be modeled as discrepancy of size of expert proposed bounding box with current bounding box of entity as is used in [21]. Current implementation of PoTE model does not explicitly capture the goodness of those experts that generate only a bounding box. This could be easily incorporated in current

framework itself by decreasing the variance for experts with high confidence scores. Furthermore, our model does not require all the experts to be good at tracking. The only requirement on each tracker is that the variance of a expert should be consistent with its actual performance.

2.7.2 Point Feature Based Tracking

Good tracking results on surgical data shows robust performance of point feature based tracking. For a rigid object motion, point feature based tracking generalizes well even in case of moving camera. However, it drifts after certain frames and produces erroneous results if not supported by a detection expert. We demonstrate an upper bound on the drift error in the proposed algorithm in Appendix A.1.

2.7.3 Group Based Tracking and Identity Resolution

We proposed a computationally simple solution for group tracking and identity resolution. Identity resolution using only appearance features can be difficult if person in group have similar appearance. Additional feature based on shape, texture can be used to generalize the identity resolution. However, on tested datasets, appearance based identity resolution was found to be adequate. Results on PETS dataset (Figure 2.9) illustrates that our identity resolution algorithm is able to accurately track the humans even in case of full occlusion. However, our constant velocity based interpolation (Section 2.5.3) for fine tracking within sub-groups fails when complex object motion is present within the sub-group.

2.7.4 Failure Modes Observed

We observe the following failure modes in current implementation of PoTE model.

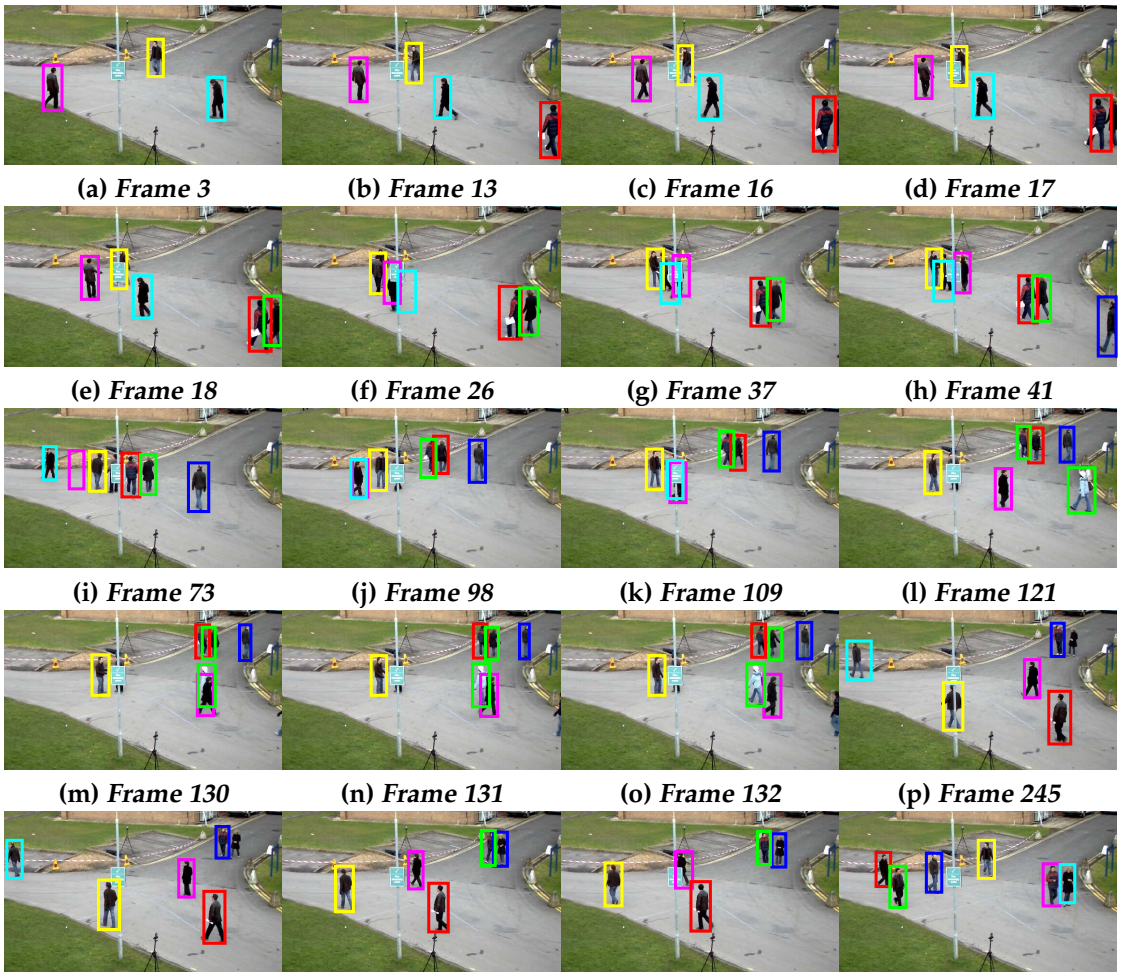
1. *Detections:* Since current framework relies on high score detections, the overall accuracy score reduces if no detections are available.

2. *Group Tracking:* Our group tracking fails on complex group interactions. These can be better handled by leveraging an online classifier model [21].

3. *Point Feature Tracking Drift:* As can be seen from our results on surgical dataset, point feature based tracking methods drifts during group interaction and if no qualified detections were available, tracking would have failed.

2.8 Conclusion

In this work, we proposed a Product of Tracking Experts framework to merge output from multiple trackers that specialize on a subset of motion and appearance characteristics. PoTE framework improved performance on a surgical dataset compared to a prototype baseline tracker. We also demonstrated a state space based framework for incorporating interaction constraints for tracking multiple objects in complex scenarios. The resulting framework has very few parameters and allows for incorporating multitude of constraints. Our framework improves on the state-of-the art on a multi-person tracking dataset demonstrating the effectiveness of the interaction model. As part of future work, we have also been exploring other constraints such as between a person and carried cart by representing these constraints using a spring model, which localizes both objects when either carried object or person is detected. Overall, this chapter lays a foundation upon which we built the pose estimation work in Chapter 3.



(a) *Frame 3* (b) *Frame 13* (c) *Frame 16* (d) *Frame 17*

(e) *Frame 18* (f) *Frame 26* (g) *Frame 37* (h) *Frame 41*

(i) *Frame 73* (j) *Frame 98* (k) *Frame 109* (l) *Frame 121*

(m) *Frame 130* (n) *Frame 131* (o) *Frame 132* (p) *Frame 245*

(q) *Frame 249* (r) *Frame 265* (s) *Frame 269* (t) *Frame 332*

Figure 2.9: Tracking results for "S2.L1 walking" scenario in the PETS 2009 Dataset S2 for People Tracking highlighting the various concepts of our tracker including entry, exit, and multiple tracking. (Please view in color)

Chapter 3

Pose Estimation

3.1 Introduction

Pose estimation is the process of measuring the current state of joints of an articulated body using data from a sensing device. In this chapter, we review major techniques of pose estimation that use visual sensing and present a novel pose tracking method which exploits regression based pose estimation. Before we proceed, its important to consider the distinction between $2D$ and $3D$ pose. $2D$ pose estimation using computer vision involves finding the (x, y) location of each individual joint in the image plane while $3D$ pose estimation requires estimation of joint angles or $3D$ joint location in inertial frame (assuming that geometry and articulation model is known *a priori*).

Pose estimation of articulated bodies using computer vision has found applications in various fields including virtual reality [106], visual surveillance [1], action recognition [169] and human motion understanding [67]. Information about pose combined with *a priori* knowledge of physical properties provides a complete description of an articulated rigid body. For example, using only the detection and tracking of a human in the image as presented in Chapter 2 is not sufficient to determine whether a human picked up an object which, however, can be readily understood using the pose information of human hands.

Pose estimation using visual sensing is an inverse problem which requires estimation of camera calibration parameters, camera pose and the state of the

articulated objects solely from the image observations of the articulated body. In most of the scenarios, this problem is ill-posed because of the degrees of freedom in the estimation problem, non-linear many-to-one camera projection models, (self) occlusions and variations in environmental conditions.

There are two major approaches for pose estimation; i) model based optimization [1] and ii) learning based regression methods [89]. Model based approaches can be further categorized into the ones that require discrete template and the ones that perform continuous estimation. The major criticism of the model based approaches is the requirement for accurate geometrical (and visual) description of an articulated body apart from the engendered computational complexity because of the curse of dimensionality. On the other hand, learning based approaches lack the ability to generalize beyond the learned data. We propose a pose estimation framework that generates uncertainty bounds in addition to the pose prediction using a learning based approach. The uncertainty estimates mitigate the performance degradation of learning based approaches on novel data by using motion continuity models to filter the pose estimates over time.

The rest of this chapter will focus on the surgical tool pose estimation problem. However, we note at the outset that none of the techniques presented in this chapter use information about the surgical scenario and hence are generalizable to any articulated pose estimation problem.

3.2 Surgical Pose Estimation

Surgical tool pose estimation has been proven to be useful for high- and low-level feedback tasks including safety-enhancement, semantic feedback and surgical skill assessment. Tool pose estimation using monocular camera input is a well-studied research problem as the monocular camera is one of the ubiquitous sensors across the spectrum of robotic devices. Current state-of-the art methods for visual tool pose estimation are computationally expensive and require elaborate geometric and appearance models of surgical tools.

We propose a tool pose estimation method that maps the visual bounding box to the 3D tool pose without any explicit knowledge of tool geometry using

Gaussian Process Regression (GPR). The proposed approach can be generalized to any surgical tool and provides tool pose estimates with a variance estimate in real-time. We demonstrate rigorous evaluation of the method under various conditions that might affect the estimation process. In order to evaluate the algorithm, we have instrumented a standard box trainer kit with two laparoscopic tools for simultaneous capture of ground truth poses and a video feed.

The present generation of surgical robotic systems (like Intuitive Surgical Inc.'s da Vinci^{TM}) are primarily tele-operated systems that rely on human-experts to perform surgeries [60]. Robotic surgical systems have achieved wide-spread adoption for a variety of surgical procedures because of reduced recovery time and integration of various automation technologies for improving patient outcomes. However, limitations such as restricted field of view, joint compliance, lack of kinesthetic feedback, lack of recommended clinical pathways etc., have compromised the safety of these devices [65]. In the light of numerous robotic-surgical accidents, there is a growing interest in improving the safety and decision support capabilities of such systems. A variety of sensing technologies have been proposed to improve the visual [82] and somatosensory feedback [157] being provided to the operating surgeon. This work focuses on improving the visual decision support system [83] available to an operating robotic surgeon. Specifically, we propose using real-time tool pose estimation to provide high-level feedback to the surgeon.

One of the ways to obtain end-effector pose is to use forward kinematics along with joint or motor encoders [71]. However, robotic devices have tool positioning inaccuracies due to joint compliance and complex articulations due to which conventional tracking or forward kinematics based approaches do not yield accurate pose estimates. Reiter et. al [122] observed errors of up to 1-inch in position accuracy using da Vinci^{TM} robot with forward kinematics based end-effector pose estimation. Tool pose estimation can also be addressed through a variety of other sensing modalities such as optical tracking systems, electromagnetic tracking systems and visual sensing mechanisms. Optical tracking systems need a clear line-of-sight to the surgical tool while electromagnetic tracking systems are affected by the presence of other metallic objects [143]. Sterilization, size, and placement requirements impose additional con-

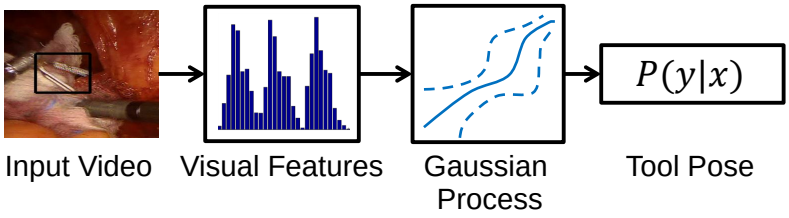


Figure 3.1: Flow chart of the tool pose estimation framework

straints on the use of the aforementioned sensing techniques. Because of these sensing limitations and instrumentation requirements, such pose sensing is limited in its applicability.

On the other hand, visual sensing via means of monocular or binocular sensing is a necessity in the design of tele-operated systems. Because of the wide spread availability, video-based approaches have gained wide acceptance as they provide a means to extract information using a 'computer-expert' viewing the scenes [149]. The research in surgical video understanding is further aided by concurrent research in the vision and robotics community such as the detection and tracking of humans for pose estimation and activity recognition [1]. Besides, the viability of obtaining real-time video feeds within an Operating Room (with minimal retrofitting) can result in a seamless integration of these methods to any existing systems. Hence, we propose using monocular video to estimate tool pose in real-time in order to develop a feedback mechanism for use in robotic surgery.

In order to guide our methodology, we consider two important goals for any surgical safety feedback algorithm; 1) real-time performance, 2) measure of confidence in estimates. Real-time performance is an operator requirement for a feedback mechanism to be integrated into any existing surgical device. Furthermore, a method that can provide confidence estimates is desirable for performing Bayesian risk based decision analysis. In order to satisfy these requirements, our framework models the tool pose as a regression problem using Gaussian Process Regression which maps the visual tool bounding box to the 3D tool pose estimate. Our framework generates a variance measure in addition to the mean estimate which yields itself to a variety of uncertainty analysis as demonstrated in this chapter.

3.3 Previous Work

Previous work in tool pose estimation literature can be broadly classified into two types; marker-less approaches that do not make any modification to the surgical tool and the approaches that use the assistance of external fiducials or markers on the tool. Various kinds of modifications in the tool end-effector have been proposed to achieve varying levels of tool pose. Early work focused on using color coding of instruments [154, 55] to segment the color-coded part of the tool in visual stereo/monocular imagery. However, such color markers assume separation of markers from the background, even with the variety of surgical procedures and different kinds of cameras [150]. Apart from color markers, optical and magnetic markers have been proposed to estimate tool pose [19]. But optical tracking requires direct line-of-sight while magnetic trackers are affected by presence of metallic objects in surroundings.

To overcome the limitations associated with marker-based-approaches, recent literature has focused on marker-less tool pose estimation. However, the current state-of-the-art uses different types of knowledge about the tool and its environment. Voros et. al [150] use knowledge of laparoscopic tool and environment (specifically the insertion point) to constrain the search process for edges corresponding to tools. Apart from using information about tool and environment, this work requires the tool stem to be visible in images [123]. Other predominant marker-less approaches require prior geometric and appearance knowledge of the tool end-effector [31, 123, 113]. Pezzementi et. al [113] used the appearance of different tool parts and a 3D model of the tool to estimate configuration by rendering and measuring consistency of 3D rendered model with tool appearance in image. Instead of working with the continuous configuration space, Reiter et. al [123] created tool templates corresponding to discretized kinematic joint configurations. Apart from requiring a computer-aided design (CAD) model and appearance model of the tool, such methods also have high computation requirements because of large number of image rendering necessary for objective function evaluation.

Some recent work by Bell et. al [15] predicts 6 degrees of freedom (DOF) change in endoscopic camera pose by computing optical flow over entire image

and using Artificial Neural Networks (ANN) to map the flow over a spatial partitioned grid to change in pose. However this framework assumes a static background and only produces a point pose estimate.

In contrast, our work does not require prior knowledge of tool parts appearance or detailed 3D models. We use generic visual features which are not specific to certain tool types and model pose estimation as a GPR problem which is solved in real-time during testing phase.

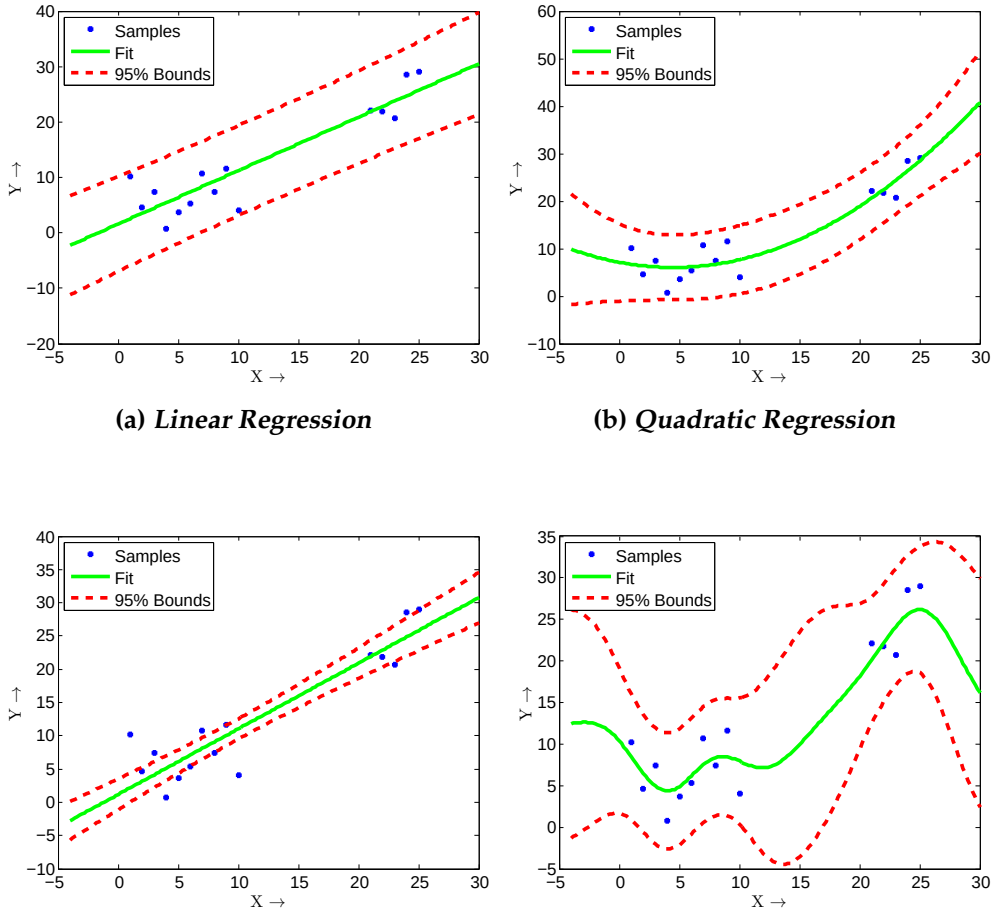
3.3.1 Computational Complexity

The pose estimation using the proposed method has a computational complexity of $O(n^3)$ during training step where n is the number of training data points. However, more importantly, during testing time the computational complexity is $O(n^2)$. On the other hand, let us consider the example of two model-based pose estimation algorithms. Agarwal et. al [1] use a combination of Golden Section, Powell's method and Genetic Algorithm for optimization. The complexity of Powell's method using golden section method is at least $O(N \log(\frac{1}{\epsilon}))$ [98] where N is the dimension of search space and ϵ is the convergence threshold. Computational complexity for the Genetic Algorithm is highly dependent on the nature of the optimization problem but is often polynomial order in time [130]. Hence the overall optimization process requires large number of optimization function evaluation which itself is computation inefficient because of the forward projection of CAD models to image frame. Reiter et. al [123] discretize each pose dimension and generate templates corresponding to each possible pose state configuration resulting in exponential $O(N^p)$ complexity where p is the number of discretizations in a pose dimension.

3.3.2 Regression Methods

The effectiveness of Gaussian Process Regression over traditional regression methods can be demonstrated using a dummy problem. Consider the problem of fitting a regression model on data generated from $y = x + 0.005x^2 + \epsilon$ where $x \in [1, 10] \cup [21, 25]$ is a positive integer and $\epsilon \sim \mathcal{N}(0, 10)$ is the random noise distributed with a 0 mean and 10 variance. To demonstrate the effective-

ness of GPR, we compare it with two methods of polynomial regression (linear and quadratic), and Bayesian Linear regression. Figure 3.2 demonstrates the results. We show the variance estimates that can be possibly generated using each regression method. Polynomial regression method yields point estimates of regression parameters (eg.: Slope and Intercept for linear regression) using which a prediction error can be obtained for large samples by applying t-tests to the estimates [33]. However as seen from the Figure 3.2(a),(b), assumption of an underlying polynomial model dictates the prediction even over the parts of sample space where one does not have any samples. Moreover, the prediction itself is incidental because of the underlying model one chooses. To get better variance estimates regardless of the sample sizes, one can use Bayesian methods. However even the Bayesian method (see Figure 3.2(c) for a linear case) suffer from limited expressivity of the underlying function. Contrasting these methods with the GPR (Figure 3.2(d)), GPR generates high variance estimates over the sample space with no training samples $x \in ([11, 20] \cup (-\infty, 0] \cup [26, \infty))$ while producing a mean function with minimal regularity assumptions.



(a) ***Linear Regression*** **(b)** ***Quadratic Regression***

(c) ***Bayesian Linear Regression*** **(d)** ***GPR***

Figure 3.2: (a) Linear Regression yields the model $y = mx + c$ *with* $m = 0.9644$ *and* $c = 1.5891$ *(b) Quadratic model yields the model* $y = ax^2 + bx + c$ *with* $a = 0.0534$, $b = -0.04781$ *and* $c = 7.1231$ *(c) Bayesian Regression fit uses a zero mean Gaussian observation noise with variance* 10 *and a zero mean Gaussian with diagonal variance* 5 *prior on slope and intercept. The estimation process results in a mean parameter estimate of* 0.9867 *slope and* 1.1797 *intercept. (d) GPR with constant mean function, Gaussian likelihood and Squared Exponential covariance function*

3.4 System Overview

Our method takes video frames with annotated bounding boxes around the surgical tools in the image frames as inputs. Generic visual features (x) are ex-

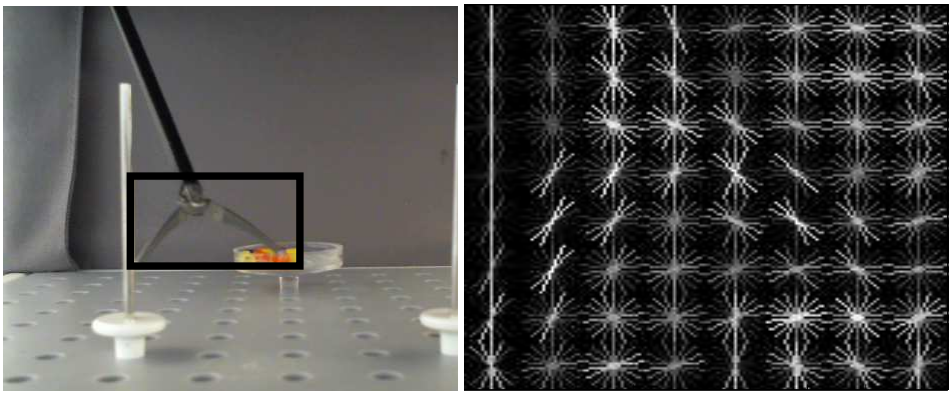


Figure 3.3: An example image with tool and corresponding HOG feature. The two different parts of the tool show up distinctly in the orientation histogram.

tracted from the image frame which are then mapped to tool pose (y) by learning a Gaussian Process. Tool pose y for the current problem is defined by 3 Euler angles representing orientation of tool stem and the end-effector opening angle which is observable solely from a bounding box around the end-effector.

3.4.1 Visual Features

A good image feature for pose estimation should ideally be unique for different poses of the tool and invariant to various sources of noise in imaging such as specular reflections, motion blur and partial occlusion. Instead of designing a robust feature which might restrict application of this framework to a certain application, we propose using generic visual features from which robust subspace can be identified independent of the application case.

For the current work, we consider using Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) which have been used previously for surgical video understanding [83]. LBP is a discriminative and computationally efficient way of modelling texture in an image [107]. HOG is one of the most widely used features in computer vision literature for object detection [29]. Figure 3.3 shows an image with bounding box around a surgical tool and the corresponding HOG visual representation. It can be seen that the resulting feature representation captures the shape of the tool.

3.4.2 Gaussian Process Regression

Since these generic visual features lie in a high-dimensional space, traditional regression frameworks such as linear, quadratic, ANN etc. will not perform well because of the need to estimate a high number of associated parameters. Traditional regression models do not provide reasonable uncertainty estimates and as a consequence these methods are unsuitable for Bayesian reasoning. Furthermore, the prediction itself is incidental because of the regression model one chooses.

GPR defines a distribution over function with inference taking place in the space of functions [120], thus avoiding the need to estimate the weights/parameters associated with traditional regression methods. Due to these properties, Gaussian process regression has been used in a wide variety of applications including human pose estimation [36], tracking [75, 117] and control [105]. We use GPR [120] to develop a pose estimation framework which is learnt using the ground truth marked tool image with corresponding pose. Fig.3.1 shows the overview of the proposed approach.

Let us denote the regression function by $f(x)$, which maps high-dimensional feature vector x to pose y. Given ground truth data from n observations, (X, Y), our framework seeks to estimate pose $y*$ for a new image with associated feature vector x^*,$p(y^*|x^*)$. We assume the measurement error to be additive in nature and independently and identically distributed according to a Gaussian distribution with zero mean and σ^2 variance, $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The measurement process can then be represented using Equation 3.1, where $f(x)$ represents the function that the Gaussian regression process seeks to infer from the space of functions.

$$y = f(x) + \epsilon \tag{3.1}$$

Following the notational representation in [120], we characterize function $f(x)$ with a Gaussian process which can be fully specified by a mean function $m(x)$ and covariance function $k(x, x^1)$ as represented in Equation 3.2.

$$f(x) \sim \mathbb{GP}(m(x), k(x, x^1)) \tag{3.2}$$

The covariance function captures the effect that any point in feature space x_p has on another point x_q. For the purpose of this work, we chose the commonly used form of covariance function, squared exponential, $\text{cov}(f(x_p), f(x_q)) = k(x_p, x_q) = \exp^{-\frac{1}{2l}(x_p - x_q)^2}$, where l captures the scale and in turn the smoothness of the function $f(x)$.

From the observation process as represented in Equation 3.1, by taking covariance on both sides and using the distribution of $f(x)$ for two different observations y_p, y_q, we get,

$$\text{cov}(y_p, y_q) = k(x_p, x_q) + \sigma^2 \delta_{pq} \tag{3.3}$$

where δ is the Kronecker delta function. Accumulating the data from all observations and using Equation 3.3, we obtain $\text{cov}(y) = K(X, X) + \sigma^2 I$, where $K(X, X)$ is the $n \times n$ matrix of covariances evaluated over the entire training data. Since we are interested in adapting the prior to the training data and eventually estimating the pose output at collection of unobserved features X^*, we model the joint distribution over the training and unobserved test outputs, $\mathbf{f}(X^*)$. Evaluating covariance over both training and test outputs using Equation 3.1, we get

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \mathbf{m}(X) \\ \mathbf{m}(X^*) \end{bmatrix}, \begin{bmatrix} K(X, X) + \sigma^2 I & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix} \right) \tag{3.4}$$

where $K(.,.)$ has the form of the covariance matrix evaluated at the corresponding inputs. Marginalizing out the observed ground truth data in order to obtain the conditional predictive distribution at test features, we get

$$\begin{aligned} \mathbf{f}^* | \mathbf{X}, \mathbf{y}, X^* &\sim \mathcal{N}(\bar{\mathbf{f}}^*, \text{cov}(\mathbf{f}^*)) \\ \bar{\mathbf{f}}^* &= \mathbf{m}(X^*) + K(X^*, X)[K(X, X) + \sigma^2 I]^{-1}(\mathbf{y} - \mathbf{m}(X)) \\ \text{cov}(\mathbf{f}^*) &= K(X^*, X^*) - K(X^*, X)[K(X, X) + \sigma^2 I]^{-1} K(X, X^*) \end{aligned} \tag{3.5}$$

By marginalizing over this joint distribution, we get the predictive distribution as represented in Equation 3.5. The mean value obtained at test feature value is essentially a linear combination of ground truth observations $\mathbf{y}$. As

mentioned earlier, we also obtain a variance estimate at each prediction in addition to the mean. Intuitively this algorithm gives high confidence estimates in the region of feature space with lot of ground truth data, and estimates with high variance in regions with sparse ground truth data.

3.5 Improving Prediction

Since the regression process provides a variance estimate in addition to the mean estimate, we propose multiple ways in which this information can be leveraged to improve prediction. Surgical actions have a certain degree of smoothness which can be used to smooth out mean estimates from regression process. The regression process by itself only considers the information extracted from a single frame which can then be used in a Kalman filter [156] setting to induce temporal filtering.

To ensure temporal continuity of the pose estimation framework, we consider second order motion continuity model as the dynamics model by treating acceleration as random noise. Consider a single k^{th} element of the pose state $y(k)$, the state evolution model can simply be represented as

$$\begin{bmatrix} y(k)_t \\ \dot{y}(k)_t \end{bmatrix} = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y(k)_{t-1} \\ \dot{y}(k)_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\delta t^2 \\ \delta t \end{bmatrix} \ddot{y}(k)_{t-1} \tag{3.6}$$

where $\dot{y}(k)_{t-1}$ and $\ddot{y}(k)_{t-1}$ represent first and second order derivatives of the state at previous time step $(t-1)$ and δt is the time step. Acceleration is modelled as zero mean Gaussian white noise $\ddot{y}(k)_t \sim \mathcal{N}(0, \sigma_a^2) \forall t$ with variance σ_a. The observation model is the Gaussian process regression framework as derived in Section 3.4.2 can be represented by Equation 3.7.

$$z_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y(k)_t \\ \dot{y}(k)_t \end{bmatrix} + v_t \tag{3.7}$$

In Equation 3.7, z_t is the mean prediction given by the regression framework which directly estimates the state $y(k)_t$ and v_t is the variance predicted by the regression framework which can be modelled as zero mean Gaussian distribu-

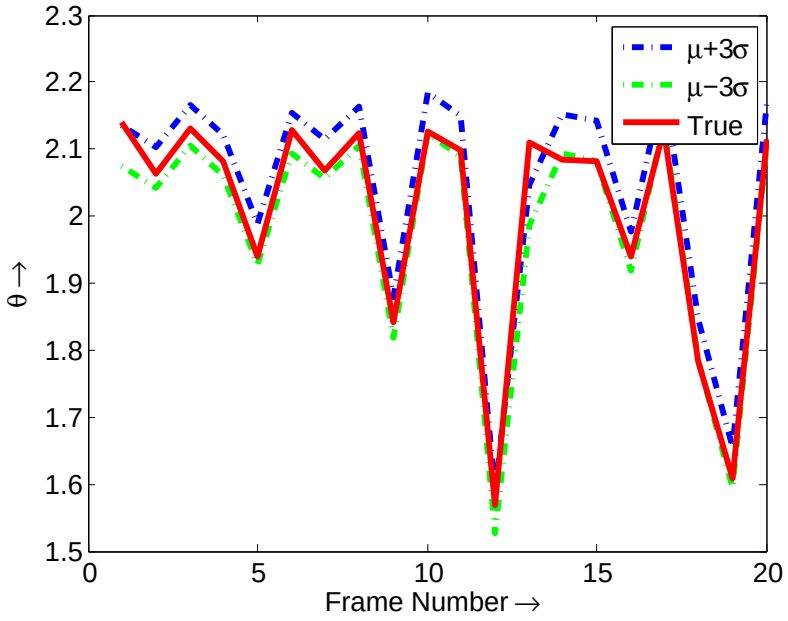


Figure 3.4: Gaussian Process regression with 3 sigma bounds plotted with true value of tool opening angle

tion at each time step.

To establish that the Gaussian process regression framework produces reasonable estimates of variance, we plotted the mean and 3 sigma bounds of variance estimated against the ground truth values. As shown in Figure 3.4, true value of the pose predominantly lies within the 3 sigma bounds. This proves that Gaussian process regression captures the variations present in feature space with respect to pose. This time varying variance along with mean prediction is used for temporal filtering using the Kalman Filter [156].

3.6 Experiments

To obtain the ground truth data for training and testing of the proposed framework, we used a standard box trainer kit with two laparoscopic tools as shown in the Fig.3.5. The box trainer was instrumented with NaturalPoint^{TM} Optitrak compatible markers to obtain ground truth tool pose data. Both moving and fixed part of a tool are instrumented with 3 markers which are tracked at 50 frames per second (fps). The motion data include columns of 3D coordinates of

individual markers along with 3D centroid location and orientation of tracked rigid bodies (computed from a set of markers associated with each such rigid body). The end-effector angle is directly related to the angle between normals of the planes attached to fixed and moving parts of the tool. A commercial web cam served as an endoscopic camera and was used to record images with a resolution of 640×480 pixels at 15 fps from inside the box trainer for our analysis.

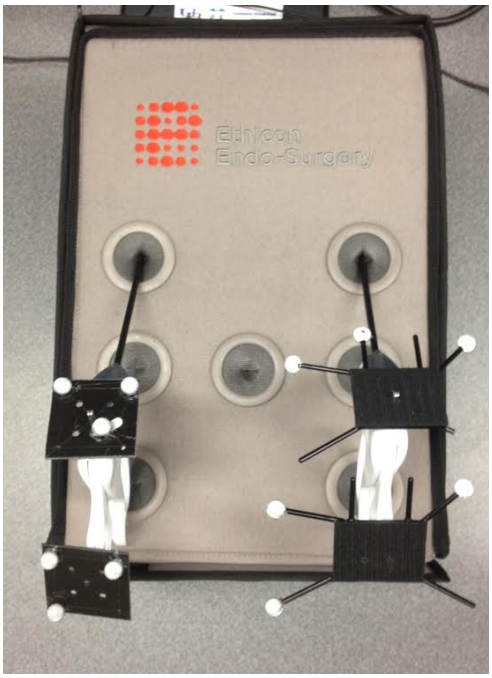


Figure 3.5: Customized Box Trainer Setup Retrofitted with Optical Reflective Markers

Experimental data was collected for simulating pick and place tasks. Our entire dataset had 4346 different tool poses with sensing noise such as motion blur due to fast tool opening, partial occlusions due to other tools and lighting variation. Each tool in the image plane had an associated bounding box. We use the proposed framework to map a visual bounding box to tool pose using various features. We estimated 4 angles for each tool (3 for tool stem orientation and 1 for end effector opening angle) and computed an error measure using cosine distance. We performed 5 fold cross validation wherein 4 parts of the data are used for training the Gaussian process regression which is then tested on the remaining data not used for training. Hyper-parameters for regression were estimated from training data by minimizing negative log marginal likelihood

Features	Orientation Error	Opening Angle
HOG	2.42°	2.49°
LBP	1.92°	2.48°

Table 3.1: Tool pose estimate angular accuracy in degrees using different visual features

with respect to hyper-parameters [120].

The part of the image corresponding to the bounding box of the tool was extracted and resized to a fixed size of 64×48 to ensure that visual features have same dimensionality during training and testing. HOG feature was extracted with a cell size of 8 resulting in a 1488 dimensional feature vector [144]. LBP was extracted with a cell size of 8 resulting in 696 dimensional feature vector.

Table 3.1 shows the average angular difference in degrees by converting the average cosine distance back to degrees. It is observed that both the visual features produce highly accurate pose estimates. Even with low observability for tool stem orientation due of the shape of tools, the angular discrepancy is lower because of the nature of laparoscopy wherein the insertion point of the tool strongly constrains the orientation of tool stem. On the other hand, tool opening angle demonstrates the effectiveness of the proposed framework because tool opening angle can vary from 0 to 90 degrees. Such accuracy in tool opening angle makes the proposed framework directly applicable to objective evaluation of surgical expertise by understanding "grasp" and "move" motion during surgery [66].

3.6.1 Execution Time

Our framework was real-time during the testing phase when tested on a desktop with Intel (R) Xeon (R) CPU E5620 @ 2.40 Ghz processor and the entire code base being run in MATLAB R2010b. On average, our framework took 0.046 sec per frame for bounding box extraction, resizing and evaluating visual features and 0.0011 sec per frame for Gaussian process regression prediction. In contrast, the current state-of-the-art methods [122] are not real-time even while performing computations using dedicated graphics hardware.

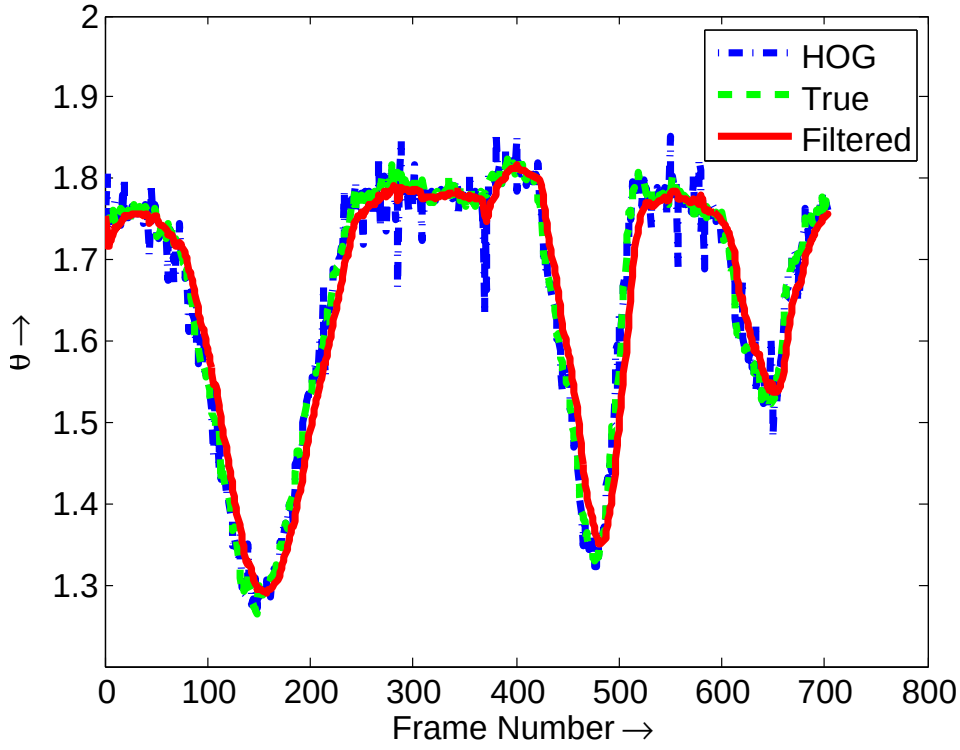


Figure 3.6: True, Regression mean estimate and filtered estimates of tool opening angle

3.6.2 Filtering

We made effective use of variance estimate provided by our framework to perform temporal smoothing on regression estimates as described in Section 3.5. Figure 3.6 shows the temporal smoothing using the Kalman filter when tested using predictions with HOG based Gaussian process regression estimates for tool opening angle. As demonstrated, while the true values are smooth, the Gaussian process regression estimates are noisy as the model does not incorporate any notion of time. The temporally noisy values of regression estimates are filtered to generate smooth variations of opening angle upon filtering. To further improve the temporal filtering, better motion models can be incorporated in Kalman filtering process.

Features	Left Tool	Right Tool
HOG	7.04°	5.63°
LBP	8.98°	6.92°

Table 3.2: Average error in tool opening angle with lighting variations

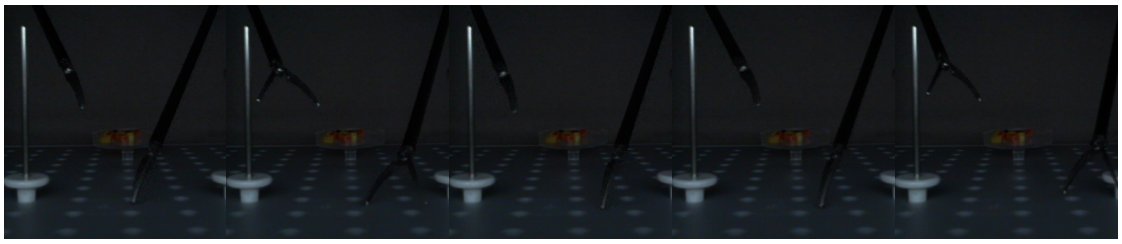
3.6.3 Generalization Beyond Learned Data

To replicate some of the effects of visual sensing noise, we performed one of the experiments with low lighting. To test the generalization performance of the proposed framework, the Gaussian process regression was never trained with low light condition. Figure 3.8 shows some of the characteristic frames and results for the low light sequence. Table 3.2 lists the angular discrepancy in left and right tool opening angles for both the visual features. Both the features generate a reasonably accurate estimate while HOG feature generalizes better compared to the LBP feature. State-of-the-art model template based methods [122] generate discrete templates with variations of $\pm 5^\circ$ for wrist pose because of the curse of dimensionality (pose lies in a high dimensional space).

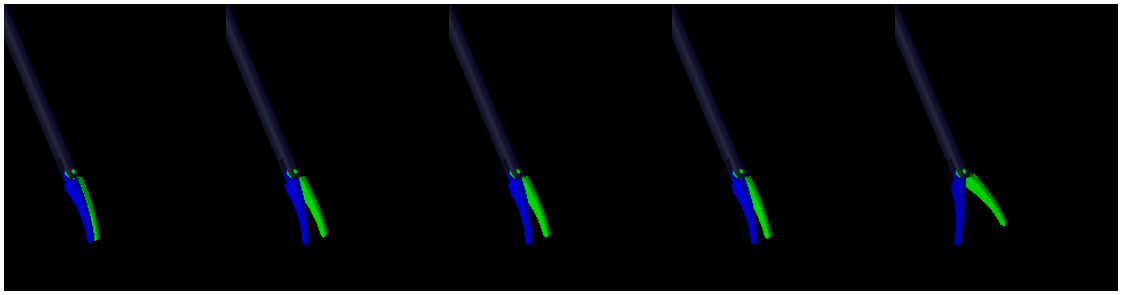
3.7 Future Work

We have proposed a generic real-time pose estimation framework with good generalization capabilities. This work opens up various directions of research including incorporation of activity specific models [48], [1] within the Kalman filtering framework to improve temporal tracking performance. Recently, Toshev et. al [141] proposed using Deep Neural Networks to estimate human pose from images. This prediction can be used a mean function for GPR process to yield a CNN-GPR regression. The predicted variance regression estimates can also be utilized for better understanding of predictions capabilities of the proposed framework. A high variance in a particular estimate reflects the sparsity of the training data in the neighbourhood of that particular pose. This can be leveraged to design experiments to further improve the accuracy of regression estimates.

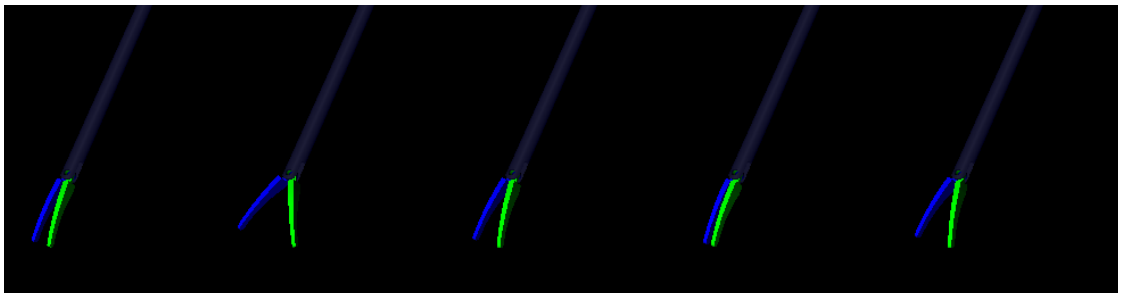
Furthermore, appropriate feature space to use as input in GPR can be chosen



(a) *Frame 52* (b) *Frame 139* (c) *Frame 209* (d) *Frame 217* (e) *Frame 285*



(f) *Frame 52* (g) *Frame 139* (h) *Frame 209* (i) *Frame 217* (j) *Frame 285*



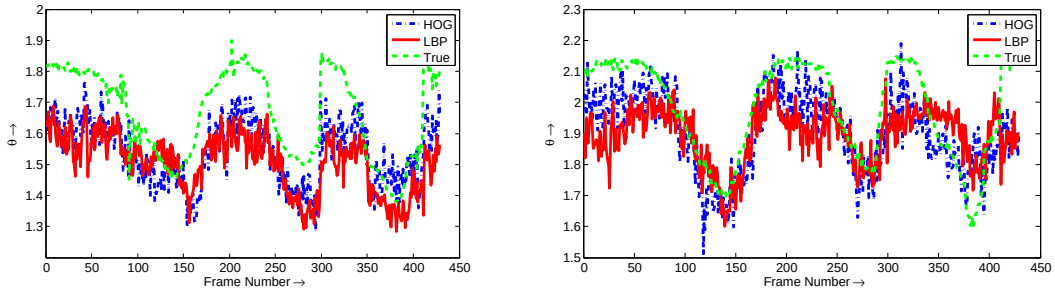
(k) *Frame 52* (l) *Frame 139* (m) *Frame 209* (n) *Frame 217* (o) *Frame 285*

Figure 3.7: Representative video frames for a "dark" sequence in the collected dataset obtained using GPR to estimate tool opening angle. First row shows the image frame, second row shows the orientation and opening angle of the left tool and the third row shows the orientation and opening angle of the right tool using LBP features.

by using negative log likelihood of observation $\log(p(y|X))$ in addition to a feature complexity term [135].

3.8 Conclusion

We proposed a real-time method to predict tool pose using generic visual features that are not specific to any tool or environment setting. Our Gaussian



(a) ***Left Tool Opening Angle*** **(b)** ***Right Tool Opening Angle***

Figure 3.8: Left and right tool opening angles for a "dark" sequence in the collected dataset obtained using GPR

process regression based framework not only predicts the regressed value but also generates reasonable estimates of variance. Experimental results using a customized box trainer demonstrate the effectiveness of the proposed approach for tool pose prediction with a variety of sources inducing visual sensing noise such as motion blur and partial occlusion. The visual features along with the learned regressors generalize well to changes in ambient lighting. The notion of using regression generated variance is demonstrated to be useful for temporal filtering using the Kalman filter. The overall framework of pose estimation is non-parametric and generalizable to novel data because of the proposed motion model based filtering. However, one last bottleneck in completing this thread of articulated body understanding is the assumption of an articulated structure. We seek to overcome this shortcoming in Chapter 4 where we demonstrate the estimation of the articulated model itself.

Chapter 4

Articulation Estimation

4.1 Introduction

Imagine a robot moving in a typical living room environment and encountering indoor objects such as doors, drawers, chairs etc. We posit that in order for the robot to understand, map, or interact with such objects, the robot needs to be able to understand the articulation. Consider a garage door as shown in Figure 4.1, in order for a robot to interact with the door, it is vital to understand the axis of motion as shown in the figure. Notably, this understanding of articulated structure (in case of the garage door, the axis of motion) coupled with the motion parameter information (in case of the garage door, the opening length of door at current time instant) provides complete information about the articulated body. This articulated structure provides common sense constraints that can improve tracking over time. For example: In case of the garage door, we know that it will not rotate or fly away and hence the only important parameter to track over time is the length along the opening direction. Psychophysical experiments on human motion understanding have demonstrated that humans first distinguish between competing motion models (translation, rotation, and expansion) and then estimate the motion conditioned on the motion model [159].

Inspired by these psychophysical experiments, we propose a framework to first estimate articulated structure/motion model and then use the resulting models to build a map of the environment. Articulation estimation is also essen-

Figure 4.1: Example of a prismatic articulated object.

tial in order to decompose the scene to model robot-environment interaction. It provides a mechanism to understand the way objects, such as drawer, and refrigerator door, can be manipulated. Apart from the active robot interaction with environment, it is essential to know the articulated structure in order to represent the pose state (pose estimation task as presented in Chapter 3). We refer to information about joints such as the type of joints (e.g. revolute), number of joints and kinematic chain as articulated structure. The most common solution to this problem is to "detect" the objects (e.g. people [50]) using the sensor data and conditioned on the object detection, the articulation structure is known *a prior* (e.g. pose estimation for humans [168]). Because of the tremendous improvements in object detection over large datasets [79], solutions that address the problem of articulation structure have taken a backseat. However, we argue that despite the success in visual detection, object detection in unstructured environment still remains an open ended problem. The performance is further reduced on texture-less objects [25] which populate our indoor environments such as doors, drawers, chairs etc. Furthermore, machine learning based approaches only generalize to the objects in the training dataset which limits the applicability of "detection" based approaches to previously unseen objects.

4.2 Related Work

4.2.1 Structure from Motion

Using image motion to understand motion and structure in the scene is a historically well studied problem in computer vision. Ullman proposed that in non-degenerate cases under orthographic projection, three pictures of four points can determine structure and motion [142]. Tomasi and Kanade formalized Ullmans's idea and suggested a method to compute camera motion and image structure by tracking features in the images [140]. Motion and shape matrices were obtained by factorizing a matrix of feature tracks and enforcing the rank constraint of the rigid body motion along with metric constraints of a rotation matrix. Costeira and Kanade extended the factorization idea to segment and recover shape along with motion of multiple moving bodies in the scene [28] . The resulting motion of rigid bodies can be further analyzed to estimate kinematic chains and, hence, to yield articulated structures [165].

There are certain fundamental limitations to structure from motion approaches. First, the reliance on feature tracking methods such as KLT [97] is not suitable for indoor environments which may not have much texture. Secondly, motion orthogonal to the image plane is not modeled because image projection is approximated as affine projection [165]. With the discovery of cheap and commonplace hardware such as Kinect, there is a need to re-examine the traditional structure from motion idea. Since such hardware already provides depth for a feature point, one already has shape as estimated by traditional structure from motion. Also using depth, one can model the motion orthogonal to image plane. Furthermore, texture-less objects can be tracked better by adding depth edges to the tracking mix. There have been efforts at using depth information by simply using the depth and calibration parameters to directly represent the trajectory in Euclidean space (R^3) [114, 70].

4.2.2 Direct Motion Sensing Approaches

Another predominant class of methods to estimate articulated structure assumes that the motion information of individual parts is directly available. Placement

of markers such as ARToolKit [42], checker-board markers, infrared markers, etc. on various parts of the articulated body can yield good estimates of the motion transformation. The placement of markers removes the need of otherwise noisy feature-tracking from the structure estimation process [54, 136, 134, 57]. Another way to get better estimation of articulated motion is via active interaction of robot manipulating an articulated object [69, 57].

In contrast to state-of-the-art methods, we propose performing online articulation estimation. Online estimation not only enables changing beliefs with more observations but also allows for inclusion in online tracking and mapping algorithms. Prior work has relied on collecting data from demonstrations and performing articulation estimation offline. Recently, Martin et. al [101] have proposed a framework for online estimation, however, there is no explicit probabilistic measure for model confidence to select an articulation model. We have also added newer class of articulation models under consideration such as object moving on a plane. Our second major contribution consists of addressing the lack of temporal modelling (Ex:acceleration/deceleration of a door) in articulation estimation. We propose an explicit temporal model for each articulation type which is necessary to make good long-term future predictions. Temporal modeling of arbitrary order allows us to ; i) track new parts/objects that enter/exit the scene [101], ii) model the entire scene and as a result explore dependencies between neighboring objects, and iii) assimilating articulated object motion in Simultaneous Localization and Mapping (SLAM) [34]. To the best of our knowledge, this is the first work that addresses the use of articulated objects with arbitrary order temporal models within SLAM.

4.3 Articulation Model

We represent all the articulated motion in the world as

$$X(t) = f_M(C, q(t)) + \nu \tag{4.1}$$

where $X(t)$ is the observed motion of an object, $M \in \{M_j\}_{j=1}^{r}$ is one of the r possible motion models, C is the configuration space (e.g. axis vector in case

of a door), $q(t)$ represents the time-varying motion variables (e.g. length of prismatic joint, angle of door) associated with the motion model M and ν is the noise in observations. We are assuming that all the motion in the world can be explained by one of the pre-defined r possible motion models.

This kind of representation wherein non-time varying configuration parameters are separated from time-varying motion variables is beneficial in a multitude of ways: First, it allows for a unified treatment of various types of articulation because of a single and consistent representation of motion variables. Second, this representation can be robustly estimated from experimental data given the reduced number of parameters to be estimated. Furthermore, it often makes the estimation problem linear and as a consequence, convex.

A notable omission from our modeling of articulated systems as in Equation 4.1 is the input to the system such as torque acting on a door, force on a drawer etc. This modeling limitation is due to the passive nature of our sensing approach in addition to no prior information about the agents in the scene. To predict motion at the next time step $P(X(t+\delta t)|X(t))$ without modeling the input forces/torques (thus not using a dynamics model), we need to model the propagation of motion variables $P(q(t+\delta t)|q(t))$. Before we proceed to model the temporal evolution of motion variables, we consider the task of configuration estimation.

4.3.1 Articulation Classification

The configuration parameters in Equation 4.1 are entirely dependent on the type of articulated joint. In this section, we consider the problem of articulation identification from point correspondences over time. Rigid bodies can move in $3D$ space with $SE(3)$ configuration which is the product space of $SO(3)$ (Rotation Group for 3D rotation) and $E(3)$ (Translation using 3D movement). The full $SE(3)$ has 6 degrees of freedom (DOF) which are reduced when a rigid body is connected to another rigid body via a joint. For example, the configuration space for a revolute joint (1 DOF joint) can be assumed to be a connected subset of the unit circle. Figure 4.2 shows some of the articulated joint modeled in this work.

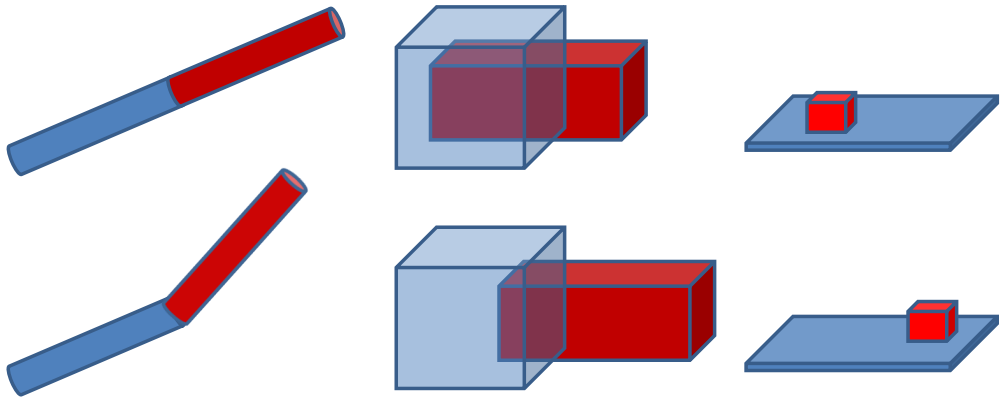


Figure 4.2: Demonstration of articulated joints considered in this work at two different time steps. Revolute and prismatic joints are 1 DOF joint while motion on a plane is a 2 DOF joint.

We consider two different types of articulation classification framework: i) Rigid Body Articulation Classification, and ii) Single Point Articulation Classification. This distinction is important because a rigid body articulation classification requires observation of at least 3 points on the same body over time. However, this approach is not suitable for tracking 1 point on the body or inclusion in feature based computer vision methods such as Extended Kalman Filter (EKF) SLAM.

4.3.2 Rigid Body Articulation Classification

We extend the factorization approach as described in [28] to 3-D track data available from a depth camera. We assume that a single object moves relative to a static camera on which features can be tracked from frame to frame. Following the notation in the paper, let us represent a point on the object as $p_i^T = [X_i, Y_i, Z_i]^T$ in the camera frame. In the current frame f, the position of the point in homogeneous coordinates can be represented as

$$s_{fi}^C = \begin{bmatrix} p_{fi}^C \\ 1 \end{bmatrix} = \begin{bmatrix} R_f & t_f \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} p_i \\ 1 \end{bmatrix} = \begin{bmatrix} R_f & t_f \\ 0_{1\times 3} & 1 \end{bmatrix} s_i \tag{4.2}$$

where R_f and t_f are the rotation and translation of the object from current frame to the frame in which object points are initially represented and s_i is the homogeneous representation of the point p_i in inertial frame. Assuming that we track N features over F frames, one can write

$$\begin{bmatrix} u_{11} & \dots & u_{1N} \\ . & & . \\ u_{F1} & \dots & u_{FN} \\ v_{11} & \dots & v_{1N} \\ . & & . \\ v_{F1} & \dots & v_{FN} \\ w_{11} & \dots & w_{1N} \\ . & & . \\ w_{F1} & \dots & w_{FN} \end{bmatrix} = \left[\begin{array}{c|c} i_1^T & t_{x_1} \\ . & . \\ i_F^T & t_{x_F} \\ j_1^T & t_{y_1} \\ . & . \\ j_F^T & t_{y_F} \\ k_1^T & t_{z_1} \\ . & . \\ k_F^T & t_{z_F} \end{array}\right] \begin{bmatrix} s_1 & . & . & . & s_N \end{bmatrix} \tag{4.3}$$

where (u_{fi}, v_{fi}, w_{fi}) is the location of a feature point in the current frame, vectors i_f^T, j_f^T, k_f^T are the rows of the rotation matrix R_f and $(t_{x_f}, t_{y_f}, t_{z_f})$ represent the components of the translation vector at time instant with frame f. Equation 4.3 can be represented in a accumulated form as

$$\mathbf{W} = \mathbf{MS} \tag{4.4}$$

where $\mathbf{W}$ represents the accumulation from trajectories of N points tracked over F frames, $\mathbf{M}$ contains all the information about the motion of the object present in the scene and $\mathbf{S}$ contains all the information about the shape of the object. Since the rank of the product of two matrices can not exceed the minimum of rank of individual matrices, it is clear that the maximum rank of W is 4 (Rank(M) = 4). Computing the singular value decomposition of $\mathbf{W}$, we get

$$\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^T \tag{4.5}$$

where $U \in R^{3F\times 4}$, and $V \in R^{N\times 4}$ are left and right real singular matrices and Σ is a 4×4 diagonal matrix of singular values. If $\mathbf{W}$ was full rank, we would have to consider N singular values. However, we only write out the components

corresponding to first 4 singular values because the rank of W is 4,. We factorize the matrix W as product of two matrices,

$$\hat{\mathbf{M}} = \mathbf{U}\Sigma^{\frac{1}{2}}, \hat{\mathbf{S}} = \Sigma^{\frac{1}{2}}\mathbf{V}^T \tag{4.6}$$

The factorization as defined in Equation 4.6 is not unique as any invertible 4×4 matrix A will lead to an alternate solution $\mathbf{M} = \hat{\mathbf{M}}A$, $\mathbf{S} = A^{-1}\hat{\mathbf{S}}$. To estimate this matrix A, we use the fact that the resulting matrix $\mathbf{M}$ is a homogeneous transform generating a solution for A using Equation 4.7

$$\begin{bmatrix} m_{00}^2 & 2m_{00}m_{01} & 2m_{00}m_{02} & m_{01}^2 & 2m_{01}m_{02} & m_{02}^2 \\ m_{10}m_{00} & m_{10}m_{01}+m_{11}m_{00} & m_{10}m_{02}+m_{12}m_{00} & m_{11}m_{01} & m_{11}m_{02}+m_{12}m_{01} & m_{12}m_{02} \\ m_{20}m_{00} & m_{20}m_{01}+m_{21}m_{00} & m_{20}m_{02}+m_{22}m_{00} & m_{21}m_{01} & m_{21}m_{02}+m_{22}m_{01} & m_{22}m_{02} \\ m_{00}m_{10} & m_{00}m_{11}+m_{01}m_{10} & m_{00}m_{12}+m_{02}m_{10} & m_{01}m_{11} & m_{01}m_{12}+m_{02}m_{11} & m_{02}m_{12} \\ m_{10}^2 & 2m_{10}m_{11} & 2m_{10}m_{12} & m_{11}^2 & 2m_{11}m_{12} & m_{12}^2 \\ m_{20}m_{10} & m_{20}m_{11}+m_{21}m_{10} & m_{20}m_{12}+m_{22}m_{10} & m_{21}m_{11} & m_{21}m_{12}+m_{22}m_{11} & m_{22}m_{12} \\ m_{00}m_{20} & m_{00}m_{21}+m_{01}m_{20} & m_{00}m_{22}+m_{02}m_{20} & m_{01}m_{21} & m_{01}m_{22}+m_{02}m_{21} & m_{02}m_{22} \\ m_{10}m_{20} & m_{10}m_{21}+m_{11}m_{20} & m_{10}m_{22}+m_{12}m_{20} & m_{11}m_{21} & m_{11}m_{22}+m_{12}m_{21} & m_{12}m_{22} \\ m_{20}^2 & 2m_{20}m_{21} & 2m_{20}m_{22} & m_{21}^2 & 2m_{21}m_{22} & m_{22}^2 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \\ a_{02} \\ a_{11} \\ a_{12} \\ a_{22} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.7}$$

where m_{ij} is the (i, j) the element of the matrix $\mathbf{M}$.

The estimated M matrix has information about the rotation and translation which can be used to classify the joint. In the rest of this section, we will describe how the information from the $\mathbf{M}$ matrix is necessary and sufficient to classify a joint.

Consider the motion of two points x_0, x_1 (represented in an inertial frame) on a rigid body at time t_0 and at some subsequent times t_1, t_2. The most general form of rigid body motion of a point can be represented by a rotation matrix $R_{t_0}^{t_1}$ and an associated translation vector $T_{t_0}^{t_1}$ from time instant t_0 to t_1:

$$x_0^{t_1} = R_{t_0}^{t_1} x_0^{t_0} + T_{t_0}^{t_1} \tag{4.8}$$

where the superscript denotes the time.

4.3.2.1 Prismatic

For points lying on a prismatic joint such as a drawer, rotation with respect to an inertial frame remains the same (or the rotation between two frames is identity $R_{t_0}^{t_1} = I$), resulting in $x_1^{t_1} - x_0^{t_1} = x_1^{t_0} - x_0^{t_0}$. This is essentially saying that the vector joining two points on a prismatic joint remains the same before and after

the motion.

4.3.2.2 Revolute

For further distinction between revolute and general motion, we need information from more than one time step. For points lying on a body undergoing revolute motion such as a door, the points have the same translation vector over time. Hence, estimating the translation vector from two time steps is a sufficient condition to classify a joint as revolute joint: $T_{t_0}^{t_1} = T_{t_1}^{t_2}$.

4.3.2.3 Plane Constrained Motion

Plane constrained motion is useful for characterizing motion of objects like chairs that can be translated on a plane and rotated about the normal to the plane. Let the plane be denoted by a point x_p lying on the plane and $\hat{n}$ being normal to that plane. Consider the case of a rigid body that has point $x_c^{t_0}$ in contact with the ground plane which moves to $x_c^{t_1}$. Since $x_c^{t_0}$ and $x_c^{t_1}$ both lie on the ground plane, we have

$$(x_c^{t_0} - x_0)^T \hat{n} = 0 \tag{4.9}$$

$$(x_c^{t_1} - x_0)^T \hat{n} = 0 \tag{4.10}$$

$$x_c^{t_1} = R_{t_0}^{t_1} x_c^{t_0} + T_{t_0}^{t_1} \tag{4.11}$$

By doing algebraic manipulation, we get $(R_{t_0}^{t_1} x_0 + T_{t_0}^{t_1} - x_0)^T \hat{n} = 0$.

4.3.2.4 General Rigid Body Motion

For general motion such as a drone that can rotate and translate anywhere in the space, both the rotation and translation matrix will be different.

4.3.2.5 Static

If the rotation matrix between two instances is identity and translation is zero, then the rigid body is stationary.

4.3.3 Point Particle Articulation Classification

For the point particle classification, we consider revolute, prismatic and static point types. To find the revolute joint involves finding a circle passing through the observations of the point over time. Similarly, we need to find a line passing through the point particle observation for the prismatic joint,. We will elaborate more on this estimation process in Section 4.8.1.

4.4 Temporal Structure

Articulation estimation provides us with configuration parameters of the articulated motion but one still needs to estimate the evolution of motion variables over time (e.g.- position of the object along an axis for prismatic joint). Temporal propagation of articulated bodies will require knowledge of dynamic model parameters (mass, friction etc.) apart from the external excitation (motor torque, force) applied to the system. Several approaches have been proposed for estimating these parameters that use ground truth trajectories to estimate inertial and friction parameters [38], but they assume a priori access to the object. Furthermore, the external excitation cannot be predicted as it might vary depending on the intention of agents.

The goal of our approach is to enforce a structure on the evolution of articulated motion without using any prior information specific to the current articulated body. We take our inspiration from neuroscience literature which posits that humans produce smooth trajectories to plan movements from one point to another in environment [44]. This smoothness assumption can be leveraged by using motion models that use limited number of position derivatives. Let us assume that $q(t)$ is the articulated motion variable (e.g. extension of a prismatic joint, angle of door along a hinge). The system model for a finite order motion in continuous time domain with $\mathbb{X}(t) = [q, q^{(1)}, ., ., q^{(n-1)}]$ (dropping the explicit time dependence of q and using superscript to denote the order of derivative)

as the state can be written as,

$$\begin{bmatrix} q^{(1)} \\ q^{(2)} \\ . \\ . \\ q^{(n)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & . & . & 0 \\ 0 & 0 & . & . & 0 \\ 0 & 0 & . & . & 0 \\ 0 & 0 & . & . & 1 \\ 0 & 0 & . & . & 0 \end{bmatrix} \begin{bmatrix} q \\ q^1 \\ . \\ . \\ q^{n-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ 1 \end{bmatrix} \eta \tag{4.12}$$

where $q^{(n)}$ denotes n^{th} order derivative of the motion variable and η is the noise. This state propagation model can be converted to a discrete time model as

$$\mathbb{X}(t+\delta t) = A\mathbb{X}(t) + B\eta \tag{4.13}$$

$$A = \begin{bmatrix} 1 & \delta t & \frac{\delta t^2}{2} & . & . \\ 0 & 1 & \delta t & . & . \\ 0 & 0 & 1 & . & . \\ 0 & 0 & 1 & . & . \\ 0 & 0 & . & . & . \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} \frac{\delta t^n}{n!} \\ \frac{\delta t^{n-1}}{(n-1)!} \\ . \\ . \\ \frac{\delta t^2}{2!} \\ \delta t \end{bmatrix} \tag{4.14}$$

where A is simply the matrix exponential $\exp\{A^c \delta t\}$ of the matrix representation A^c in continuous time and $B = (\int_0^{\delta T} \exp\{A^c \mu\} \mathrm{d}\mu) B^c$ with B^c being the corresponding continuous time representation.

The model considered in Equation 4.14 is essentially trying to predict the motion variable $q(t+\delta t)$ using the information at time step t. It is a finite order Taylor series expansion of the motion variable $q(t+\delta t)$.

$$q(t+\delta t) = q(t) + \frac{q^{(1)}}{1!}\delta t + \frac{q^{(2)}}{2!}(\delta t)^2 + ... + \sum_{k=n}^{\infty} \frac{q^{(k)}}{n!}(\delta t)^k \tag{4.15}$$

This representation hence assumes the differentiability of the motion variable. The approximation error is of the order $O((\delta t)^n)$. Various convergence studies can be done to choose the right order n for a given time duration δt, but here we study the physical aspects of the problem.

4.4.1 Choosing Order

Ideally, one would want to choose the motion variable order as high as possible to reduce the approximation error as represented in Equation 4.15, especially, for long-term behavior prediction which is necessary for motion planning or when sensors go blind. But higher order motion models will result in over-fitting because of the need to estimate more parameters from few initial samples. It also increases the filtering problem complexity (as described in later section) significantly as Kalman filtering involves matrix multiplication; hence, the computational complexity is atleast $O(N^2)$ where N is the length of the state vector. Furthermore, the error in estimating higher-order derivatives of a noisy signal increases exponentially with respect to derivative order.

However, there is a number of reasons why we might get away with choosing a smaller order of temporal variables. First, in classical mechanics, we only consider second order derivatives of position variables. Also humans minimize jerk [44] in their motion.

4.5 Articulation Model Estimation

We now consider the task of estimating the type of articulated model $M \in \{M_j\}_{j=1}^{r}$ out of r different models. This does not automatically follow from the configuration and motion variables estimation. For example, consider the case of a point particle moving in $2D$ space: one can fit a line, circle or assume it to be static. One can potentially use goodness-of-fit measures to estimate the appropriate model along with some heuristics. However, there are various limitations in comparing goodness-of-fit measures related to the number of free parameters in different models, noise in the data, over-fitting and number of data samples required [132]. Instead of picking a model at the initial time-step, we use a filtering based multiple model approach to correctly pick the model for a given object.

We assume that our target object/particle obeys one of the r ($r \in Z^+, r > 0$) different motion models. In current formulation we assume a uniform prior $\mu_j(0) = P(M_j), \sum_{j=1}^{r} \mu_j(0) = 1$, over different motion models for each individ-

ual object. This prior can be modified appropriately by object detection. For example, doors are more likely to have revolute joint. Motion model probability is updated as more and more observations are received [164]

$$\mu_j(k) \equiv P(M_j|\mathbf{Z}_{0:k}) = \frac{P(z_k|\mathbf{Z}_{0:k-1}, M_j)P(M_j|\mathbf{Z}_{0:k-1})}{P(z_k|\mathbf{Z}_{0:k-1})}$$
$$\mu_j(k) = \frac{P(z_k|\mathbf{Z}_{0:k-1}, M_j)\mu_j(k-1)}{\sum_{j=1}^{r} P(z_k|\mathbf{Z}_{0:k-1}, M_j)\mu_j(k-1)} \tag{4.16}$$

The probability of the current observation z_k conditioned over a specific articulated motion model and all the previous observations can be represented by various methods. This probability for an Extended Kalman Filter(EKF) based filtering algorithm is the probability of observation residual as sampled from a normal distribution distributed with zero mean and innovation covariance [164]. We pick a model when the probability of that model becomes greater than a specified threshold. As more and more observations are received, our estimation algorithm chooses a specific model for each target object.

Algorithm 1 provides a pseudo-code for estimating a specific motion model. The algorithm requires definition of various motion classes, a motion class threshold τ and observations. We decide on a specific motion model when the probability of that model as represented in Equation 4.16 is greater than the threshold τ.

4.6 SLAM for Dynamic World

To demonstrate the need of articulation estimation with explicit time dependence, we propose an algorithm for performing SLAM in a dynamic scene. Figure 4.3 shows the graphical model of the most general SLAM problem where x_k, u_k, z_k, m_k, v_k represent the robot state, input to the robot, observation by robot, state of the world, and action of various agents in the environment. Please note that m in small letter is used to refer to the landmark while M is used to refer to the type of motion model.

Basic SLAM algorithms assume the map $m_{k-1} \equiv m_k \equiv m$ to be static and model the combination of robot state and map x_k, m as the state of the esti-

Data: $\{M_j\}_{j=1}^{r}$, z_k, τ
Result: $\hat{M} \in \{M_j\}_{j=1}^{r}$, C, $P(q(t+\delta t)|q(t))$
initialization: $C_j = \{\}$, $M = \{\}$;
while $M = \{\}$ **do**
  forall the $M \in \{M_j\}_{j=1}^{r}$, **do**
    if C_j *is* $\{\}$ **then**
      Estimate C_j ;
      Estimate Temporal Structure ;
    else
      Propagate state using EKF ;
      Estimate $P(z_k|\mathbf{Z}_{0:k-1}, M_j)$;
    end
  end
  forall the $M \in \{M_j\}_{j=1}^{r}$, **do**
    Normalize to obtain $\mu_j(k)$;
    if $\mu_j(k) > \tau$ **then**
      $\hat{M} = M_j$
    end
  end
end

Algorithm 1: Estimating the correct motion model and associated configuration parameters and motion variables

mation problem [34]. The estimation problem only requires a motion model $P(x_k|x_{k-1}, u_k)$ and observation model $P(z_k|x_k, m)$. The observation model assumes the observations to be conditionally independent given the map and the current vehicle state. The goal of the estimation process is to produce unbiased and consistent estimates (the expectation of mean squared errors should match the filter-calculated covariance) [164].

For the current SLAM problem, the state consists of time-varying map (unknown input to the world by various agents) and the robot state. Hence, the full estimation problem can be posed as

$$P(x_k, m_k|\mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{V}_{0:k}, x_0, m_0) \tag{4.17}$$

Following the notation in the review paper on SLAM by Durrant-Whyte and Bailey [34], $\mathbf{Z}_{0:k}$, $\mathbf{U}_{0:k}$ and $\mathbf{V}_{0:k}$ represent the set of observations, robot control

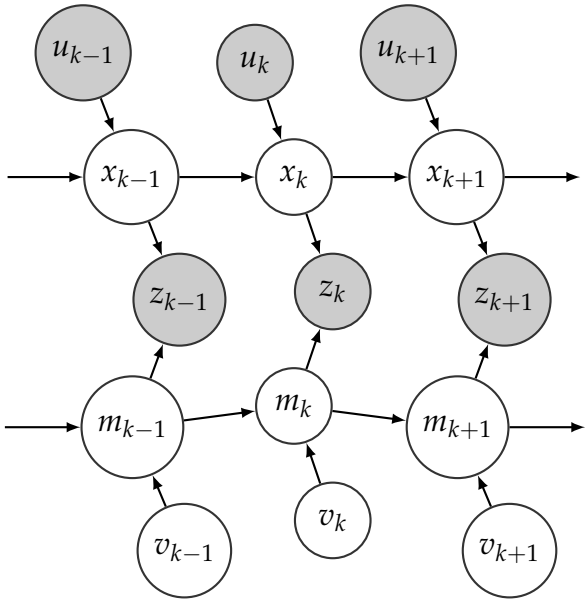


Figure 4.3: Graphical Model of the general SLAM problem. The known nodes are darker than the unknown nodes.

inputs and map control inputs from the start time to time step k. It is assumed that the map is Markovian in nature which implies that the start state of the map m_0 contains all the information needed to make future prediction if actions of various agents in the world $v_{k-1}, ..., v_{k+1}$ and their impact on the map are known.

4.6.1 Time update

The time update models the evolution of state according to the motion model. To write the equation concisely, let $A = \{\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{V}_{0:k}, x_0, m_0\}$, then

$$
\begin{aligned}
&P(x_k, m_k|A) = \\
&\int\int P(x_k, x_{k-1}, m_k, m_{k-1}|A)dx_{k-1}dm_{k-1} \\
&= \int\int P(x_k|x_{k-1}, m_k, m_{k-1}, A)P(x_{k-1}, m_k, m_{k-1}|A)dx_{k-1}dm_{k-1} \\
&= \int\int P(x_k|x_{k-1}, u_k)P(x_{k-1}, m_k, m_{k-1}|A)dx_{k-1}dm_{k-1} \\
&= \int\int P(x_k|x_{k-1}, u_k)P(m_k|x_{k-1}, m_{k-1}, A)P(x_{k-1}, m_{k-1}|A)dx_{k-1}dm_{k-1}
\end{aligned}
$$

$$= \int \int P(x_k|x_{k-1}, u_k) P(m_k|m_{k-1}, v_{k-1}) P(x_{k-1}, m_{k-1}|A) dx_{k-1} dm_{k-1} \tag{4.18}$$

The independence relationship in the derivation of the time update in Equation 4.18 is due to the Bayesian networks in Figure 4.3 in which each node is independent of its non-descendants given the parents of that node. Given the structure of time update, we need two motion models, one for robot, $P(x_k|x_{k-1}, u_k)$ and another one for the world, $P(m_k|m_{k-1}, v_{k-1})$. It can be clearly observed that $P(m_k|m_{k-1}, v_{k-1})$ for a static map is a Dirac Delta function and integrates out in Equation 4.18.

4.6.2 Measurement Update

The measurement update uses Bayes formula to update the state of the estimation problem given a new observation z_k at time step k. To write the equations concisely, let $B = \{\mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{V}_{0:k}, x_0, m_0\}$, then

$$\begin{aligned} P(x_k, m_k|B) &= \frac{P(z_k|x_k, m_k, A) P(x_k, m_k|A)}{P(z_k|A)} \\ &= \frac{P(z_k|x_k, m_k) P(x_k, m_k|A)}{P(z_k|A)} \end{aligned} \tag{4.19}$$

Equation 4.19, together with equation 4.18, defines the complete recursive form of the SLAM algorithm for a dynamic environment. The focus of current work is the representation of a map motion model to extend the standard SLAM algorithm with its static world assumption to a dynamic world.

4.6.3 Dynamic World Representation

The real world is dynamic in nature with a varying degree of motion such as a parking lot can be assumed to be temporarily stationary compared to a road which is always in motion. Previous literature to handle dynamic environments can be divided into two predominant approaches: i) detect moving objects and ignore them, and ii) track moving objects as landmarks [10]. In the first approach, using the fact that the conventional SLAM map is highly redundant,

the moving landmarks can be removed from the map building process [9]. In contrast, Wang et. al [151] explicitly track moving objects by adding them to the estimation state. However, their work assumed that the sensor measurement can be decomposed into observations corresponding to moving and static landmarks which requires a good estimate of moving and static landmarks to start with. Furthermore, it was assumed that the measurement of moving objects carries no information for the SLAM state estimation, implying that the map remains unchanged. A simple counter example consists of a moving door in an indoor environment clearly changing the map of the scene.

In this work, we demonstrate the SLAM algorithm for a feature based map. Furthermore, the overall framework is extensible to other kind of mapping algorithms. In feature based mapping, motion of each feature can be assumed to be independent, given the location of the feature at the previous time step. In dense mapping, a scene/map be decomposed into n different parts such as chair, door, etc., whose shape is known. The parts of the scene $m_k = \{b_k^i\}, 1 \leq i \leq n$, are assumed to move independently, and hence, the motion of the map can be represented as a collection of independent motion of the parts. The true motion model for each part of the scene is assumed to be one of the motion models $C \in \{C_j\}_{j=1}^{p}$ as represented in Section 4.3.1.

4.7 Articulated EKF SLAM

4.7.1 Robot Motion Model

We consider a robot with state $x_k = (x, y, \theta)^T$ at time k moving with constant linear velocity v_k and angular velocity ω_k. The state of the robot at the next time step can be represented as

$$x_{k+1} = \begin{pmatrix} x - \frac{v_k}{\omega_k}\sin\theta + \frac{v_k}{\omega_k}\sin(\theta + \omega_k \delta t) \\ y + \frac{v_k}{\omega_k}\cos\theta - \frac{v_k}{\omega_k}\cos(\theta + \omega_k \delta t) \\ \theta + \omega_k \delta t \end{pmatrix} + \mathcal{N}(0, R_k) \tag{4.20}$$

, where δt is the width of the time step and R_k is the error covariance of the noise (zero mean Gaussian). Error covariance can be derived by propagating

the noise through the robot motion model and projecting the input to the state space[138].

If the angular velocity is close to zero, the robot model as represented in Equation 4.21 will be ill-conditioned. The model with zero angular velocity is given by

$$x_{k+1} = \begin{pmatrix} x + v_k \delta t \cos(\theta) \\ y + v_k \delta t \sin(\theta) \\ \theta \end{pmatrix} + \mathcal{N}(0, R_k). \tag{4.21}$$

Following the approximations proposed by Thrun et al. [138], the angular and linear velocities are generated by a motion control unit $\hat{u}_k = (\hat{v}_k, \hat{\omega}_k)^T$ with zero mean additive Gaussian noise.

$$\begin{pmatrix} v_k \\ \omega_k \end{pmatrix} = \begin{pmatrix} \hat{v}_k \\ \hat{\omega}_k \end{pmatrix} + \mathcal{N}(0, M_k) \tag{4.22}$$

$$M_k = \begin{pmatrix} \alpha_1 \hat{v}_k^2 + \alpha_2 \hat{\omega}_k^2 & 0 \\ 0 & \alpha_3 \hat{v}_k^2 + \alpha_4 \hat{\omega}_k^2 \end{pmatrix} \tag{4.23}$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are the noise coefficients.

4.7.2 Observation Model

The robot measures range and bearing angle for the j^{th} landmark located at position $m_j = (m_{j,x}, m_{j,y})^T$ within robot's sensor field of view. Each observation can be written as

$$z_k^i = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \arctan(m_{j,y} - y, m_{j,x} - x) - \theta \end{pmatrix} + \mathcal{N}(0, Q_k) \tag{4.24}$$

where z_k^i is the i^{th} observation of the j^{th} landmark at time step k disturbed by zero mean Gaussian noise with covariance matrix Q_k.

4.7.3 Jacobian Computation

Extended Kalman Filtering (EKF) requires linearization of robot motion model to ensure that the state propagation maintains Gaussianity of the state distribution. In order to propagate the state, we need to estimate the Jacobian of the state propagation model with respect to the state at time step k. The state Jacobian can be represented as

$$J_{x_k}^{x_{k+1}} = \begin{pmatrix} 1 & 0 & -\frac{v_k}{\omega_k}\cos\theta + \frac{v_k}{\omega_k}\cos(\theta+\omega_k\delta t) \\ 0 & 1 & -\frac{v_k}{\omega_k}\sin\theta + \frac{v_k}{\omega_k}\sin(\theta+\omega_k\delta t) \\ 0 & 0 & 1 \end{pmatrix} \tag{4.25}$$

Furthermore, the error in the control space needs to be projected onto the state space, for which one needs to compute the Jacobian of state propagation model with respect to input u_k.

$$J_{u_k}^{x_{k+1}} = \begin{pmatrix} \frac{-\sin\theta+\sin(\theta+\omega_k\delta t)}{\omega_k} & \frac{v_k(\sin\theta-\sin(\theta+\omega_k\delta t))}{\omega_k^2} + \frac{v_k\cos(\theta+\omega_k\delta t)}{\omega_k} \\ \frac{\cos\theta-\cos(\theta+\omega_k\delta t)}{\omega_k} & \frac{-v_k(\cos\theta-\cos(\theta+\omega_k\delta t))}{\omega_k^2} + \frac{v_k\sin(\theta+\omega_k\delta t)}{\omega_k} \\ 0 & \delta t \end{pmatrix} \tag{4.26}$$

To assimilate each observation z_i^k, we need to compute the Jacobian of the observation model with respect to the overall SLAM state, consisting of the robot state as well as the motion parameters state associated with each landmark. However, for the i^{th} observation at time step k of landmark j, the only relevant entries in the Jacobian matrix are the derivative of observation with respect to the robot states and the motion parameters state associated with landmark j. The Jacobian of observation with respect to the robot state is

$$J_{x_k}^{z_k^i} = \begin{pmatrix} \frac{x-m_{j,x}}{\sqrt{q}} & \frac{y-m_{j,y}}{\sqrt{q}} & 0 \\ \frac{m_{j,y}-y}{q} & \frac{x-m_{j,x}}{q} & -1 \end{pmatrix} \tag{4.27}$$

and the Jacobian with respect to the motion parameters state is

$$J^{z_k^i}_{m_j} = \begin{pmatrix} \frac{m_{j,x}-x}{\sqrt{q}} & \frac{m_{j,y}-y}{\sqrt{q}} \\ \frac{y-m_{j,y}}{q} & \frac{m_{j,x}-x}{q} \end{pmatrix} J^{m_j}_{m(t)} \tag{4.28}$$

where $J^{m_j}_{m(t)}$ is the Jacobian of landmark observation with respect to the motion parameters state $m(t)$.

Data: $\mu_{t-1}, \Sigma_{t-1}, u_t, \{M_j\}_{j=1}^r, z_k, \tau$
Result: μ_t, Σ_t
Propagate Robot State and Covariance;
Propagate Landmarks State and Covariance;
forall the $z_k^i \in z_k$ **do**
  if $\hat{M} \neq \{\}$ **then**
    Estimate Motion Model($\{M_j\}_{j=1}^r, z_k, \tau$);
  else
    Assimilate Observation;
  end
end

Algorithm 2: Articulated EKF SLAM

4.8 Results

4.8.1 Configuration Estimation

We tested our configuration estimation for a variety of joint models. To elucidate the effectiveness of separation of motion parameters from configuration parameters, we consider the configuration estimation of revolute motion. Consider a point moving along a circle centered at $X_c = [x_c, y_c]^T$ with a radius r. According to our definition, configuration here refers to $C = [x_x, y_c, r]^T$, while the motion parameter is θ representing the time-varying angle of the moving point. For joint estimation, one first needs to assume the order of motion prior to configuration estimation. For the given case, we assume a commonly used constant-velocity motion model:

Estimation	Center Error	Radius Error
Joint	0.71	0.09
Separate	0.60	0.04

Table 4.1: Error metrics for center and radius error

$$X(t) = X_c + r[\cos(\theta_0 + \Delta T\omega), \sin(\theta_0 + \Delta T\omega)]^T \tag{4.29}$$

where $X(t)$ is the observed motion, θ_0 is the initial angle and ω is the constant angular velocity of the point. As can be observed the estimation problem resulting from Equation 4.29 is non-linear in θ_0 and ω. For the separate representation, the estimation problem reduces to the estimation of a circle from points lying on a circle, hence being linear. Figure 4.4 shows the estimation results. To estimate the resulting errors, we performed a Monte Carlo simulation and averaged errors across all the trials. Table 4.1 shows results from 500 Monte Carlo runs of the algorithm for the same problem. The error in estimation is Euclidean norm of the difference between estimated and true center and radius. It can be observed that separate estimation has considerably less error compared to joint estimation problem.

4.8.2 Temporal Order

After the evaluation of configuration parameters, various orders of motion models can be estimated from motion data. In order to perform this, we assume that there exists a function $G : (X(t), C) \mapsto m(t)$ to map observation and configuration data to motion parameters. This allows us to obtain motion parameters over time from which we can estimate various orders of temporal motion. We took the raw angular trajectory of a pendulum and fitted zeroth, first, and second order motion models. For a zeroth, first, and second order motion model, the state is θ, $[\theta, \dot{\theta}]$, $[\theta, \dot{\theta}, \ddot{\theta}]$ where $\theta, \dot{\theta}, \ddot{\theta}$. The motion parameter can be propagated to next time frame using the framework in Section 4.4. For the observation model, we assume the direct observation of motion parameter which is equivalent to observation of the body $X(t)$ after the configuration is estimated using

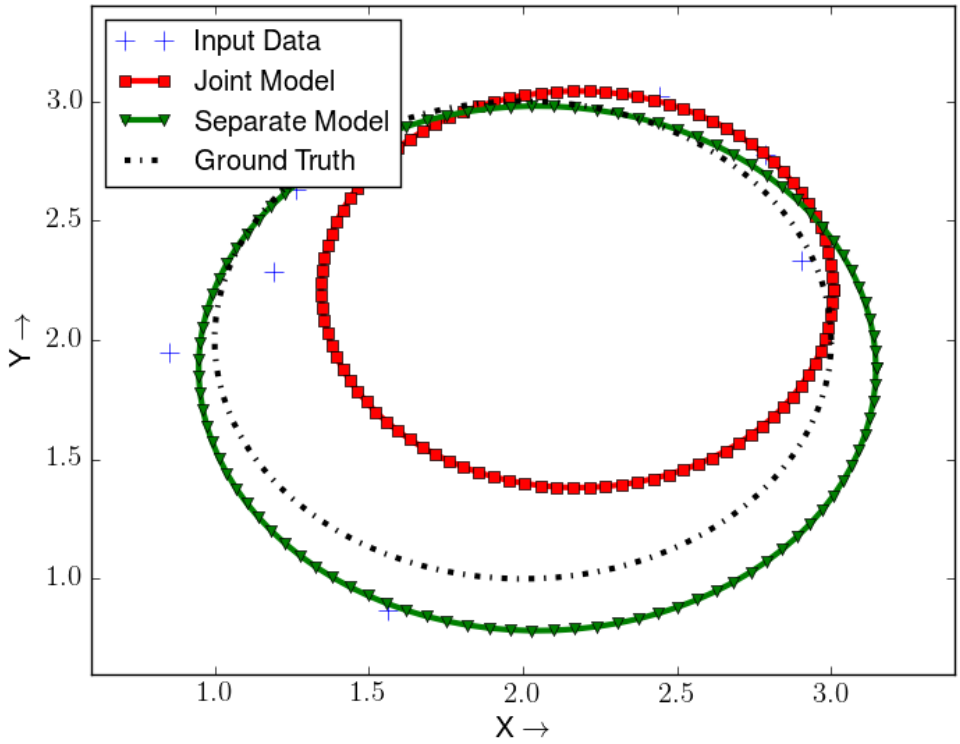
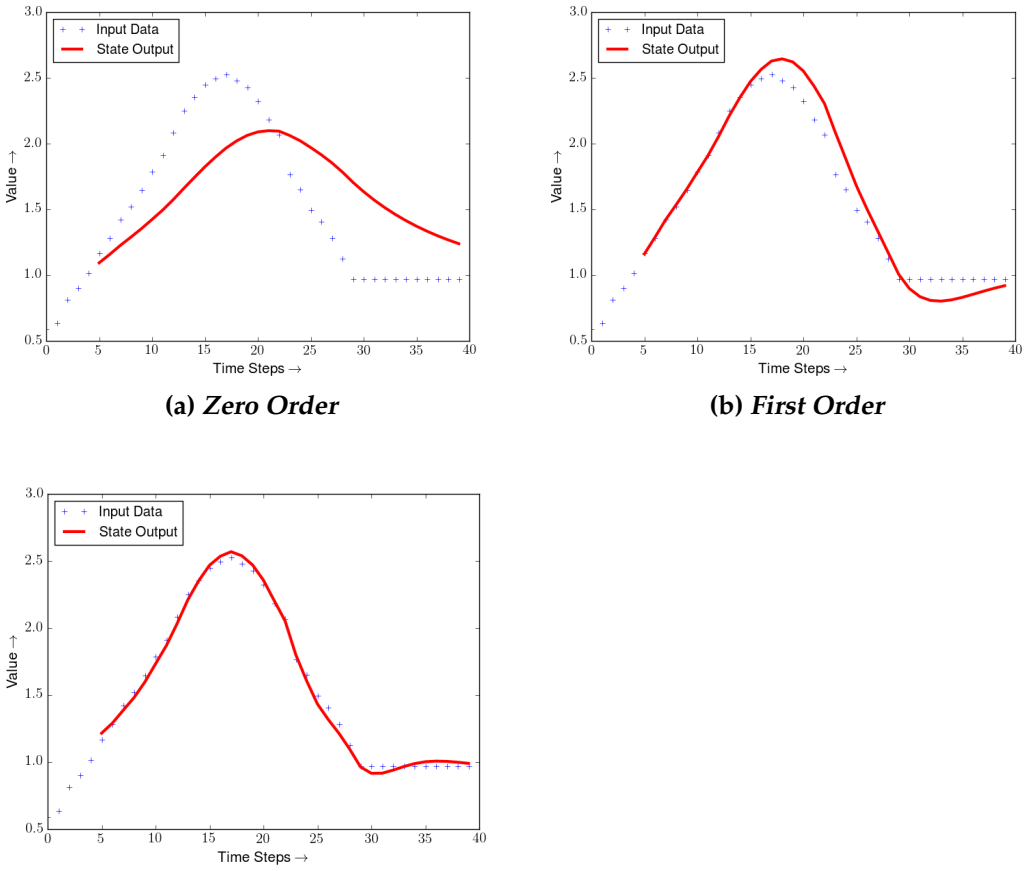


Figure 4.4: Estimation of configuration parameters for a 2D landmark in revolute motion centered at point $(2, 2)$ *with radius* 1. *Gaussian noise of* 0.01 *variance in both X and Y directions. Joint estimation yields a revolute motion centered at* $(2.18, 2, 21)$ *with a radius of* 0.83, *while separate configuration estimation yields a revolute joint centered at* $(2.05, 1.88)$ *with a radius of* 1.10

the function G. Figure 4.5 shows the motion parameter for different orders. It can be observed that the higher order motion model follows the trajectory much better than lower order motion models.

4.8.3 Articulation Estimation

To test the articulation estimation framework, we simulated an environment with one static, prismatic and revolute points each. We used a minimum of 7 samples to estimate configuration and initialize motion parameters. Figure 4.6 shows the results for the articulation estimation. Given sufficient observation, all the articulation models are estimated correctly. However, static articulation



(a) *Zero Order* **(b) *First Order***

(c) *Second Order*

Figure 4.5: Comparison of EKF filtering based state estimation for various orders of a motion parameter. For displaying purposes, we only show the zeroth order derivative state from all the different motion models.

takes the longest time to be correctly estimated. This is because of the difficulty in separating static landmark from a revolute landmark with 0 radius and 0 velocity and a prismatic landmark with 0 velocity.

4.8.4 Dynamic World SLAM

We simulated a map with point features that are either static, prismatic or revolute. A robot with limited field of view simulated readings from a laser scanner which were then used to simultaneously localize the robot as well as to map the environment.

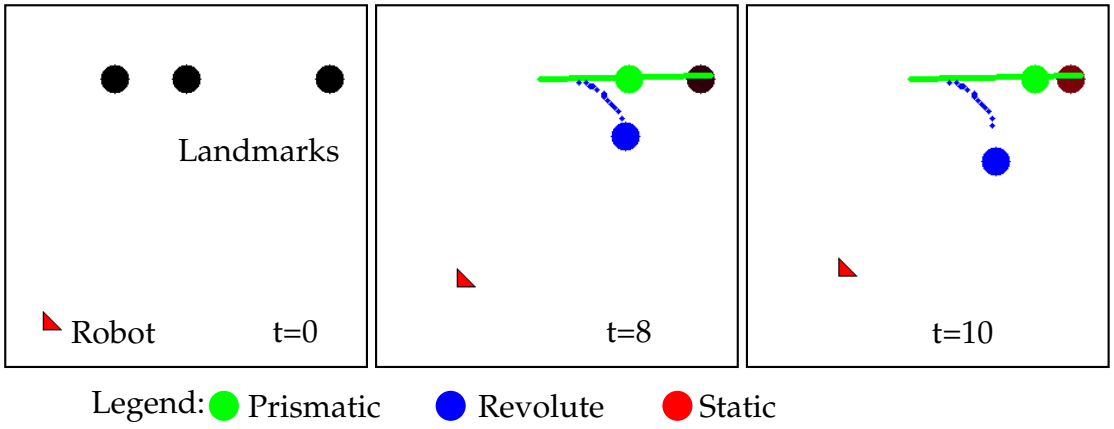


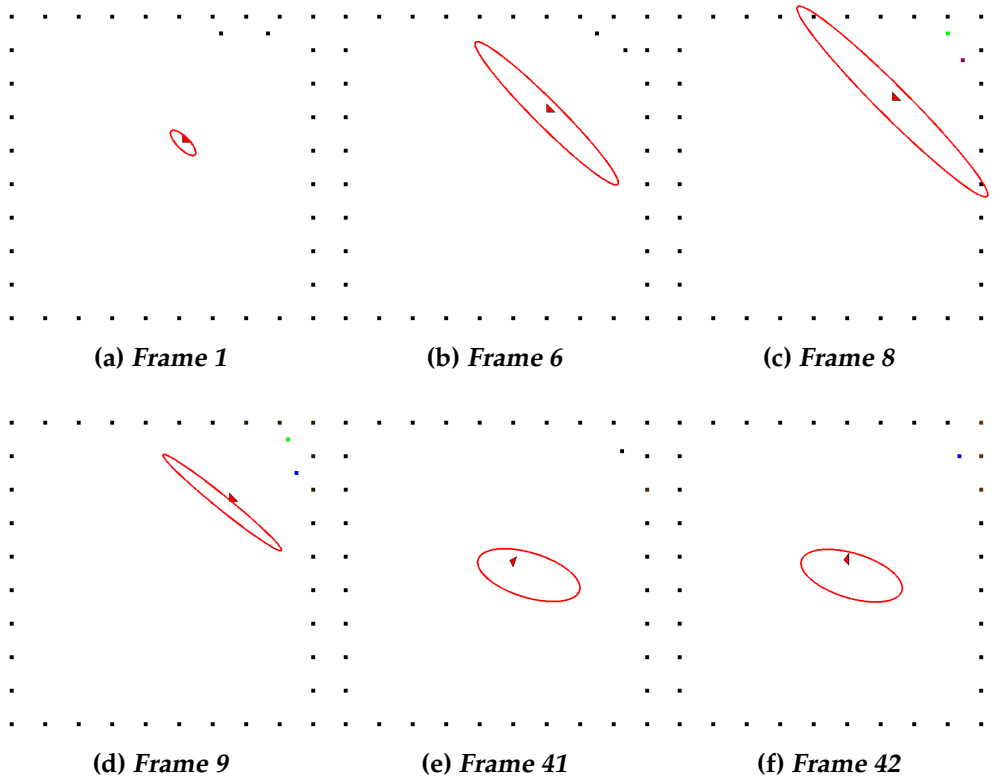
Figure 4.6: Frames at different time intervals of our simulation. Color of a landmark at a particular frame is the weighted sum of colors assigned to each motion model. The weights used are the probability of the landmark following that particular motion model and estimated by our algorithm. We also show the predicted trajectory of a landmark according to the estimated motion model.

4.8.4.1 Qualitative Analysis

We used a total of 42 landmarks in the scene with 1 revolute, 1 prismatic and 40 static landmarks. Both, the revolute and prismatic landmarks were correctly identified while our algorithm could not identify a total of 2 static landmarks with a selection threshold of $\tau = 0.75$. With a relaxed selection threshold of $\tau = 0.5$, our algorithm correctly identified all the static landmarks with more than 10 samples. Figure 4.7 shows the summary of results from the articulated EKF algorithm. Our algorithm needs a minimum of 7 samples. Hence, until frame 7, the covariance of the robot state keeps increasing. On the 8^{th} frame, our estimation algorithm correctly identifies two landmarks, and as a result, the covariance of the robot state decreases.

4.8.4.2 Quantitative Analysis

We compared the proposed Articulated EKF SLAM algorithm against the standard EKF SLAM algorithm. Standard EKF SLAM algorithm includes the landmark position in the state in contrast to our algorithm where the SLAM state includes the motion parameters. As a result, we are only comparing the resulting localization estimates of the robot. Notably, only two landmarks are non-



(a) *Frame 1* (b) *Frame 6* (c) *Frame 8*

(d) *Frame 9* (e) *Frame 41* (f) *Frame 42*

Figure 4.7: Demonstration of Articulated EKF algorithm at various time steps. At each time step, we plot the robot's true state with a triangle, and the estimation of the robot's mean and covariance SLAM states is shown by an ellipse.

Algorithm	Avg. Translation Error	Avg. Rotation Error
Articulated EKF SLAM	1.592	0.076
EKF SLAM	3.652	0.110

Table 4.2: Comparison of localization error for two different SLAM algorithms

static landmarks which violate the assumption of standard SLAM algorithms. Table 4.2 summarizes the average localization error metrics in both, position and orientation increments [91]. It is evident from the results, our algorithm significantly improves on both error metrics, even with just two landmarks in motion.

4.9 Conclusion

In this Chapter, we presented a fundamental approach to articulated structure estimation which is essential for pose estimation. Instead of assuming that detection and tracking methods as presented in Chapter 2 will directly yield structure, we use the motion of objects in the scene to estimate articulated structure. We demonstrated that our approach outperforms the traditional SLAM algorithms by integration of structure estimation into the SLAM state.

Chapter 5

Representation in Language

Humans often use language as a mechanism to convey their ideas. Hence, for robots to operate in the same environment, it is essential that robots and humans can have conversations about the physical world. In this chapter, we explore two major questions, i) What kind of language output can we generate from our articulation detection, tracking and pose estimation? ii) Can language inform vision?

First we briefly describe the type of output we can generate from detection and tracking and then from the pose evaluation. After that we devote the rest of the chapter to discuss how language can inform vision and vice-versa by exploiting the compositional nature of language. We believe that the later part of this chapter on language and vision informing each-other will form a crucial glue to link our articulation estimation, detection and tracking to human language output in order to fulfill the grand problem laid out in this thesis.

5.1 Language Output

The ability to automatically describe the content in images and videos has been widely explored by both natural language processing and computer vision communities. Recently, the availability of large datasets containing thousands of captioning examples such as MS COCO [95], Flickr30k [171] etc. have accelerated the pace of quantifiable research in image captioning. Human language

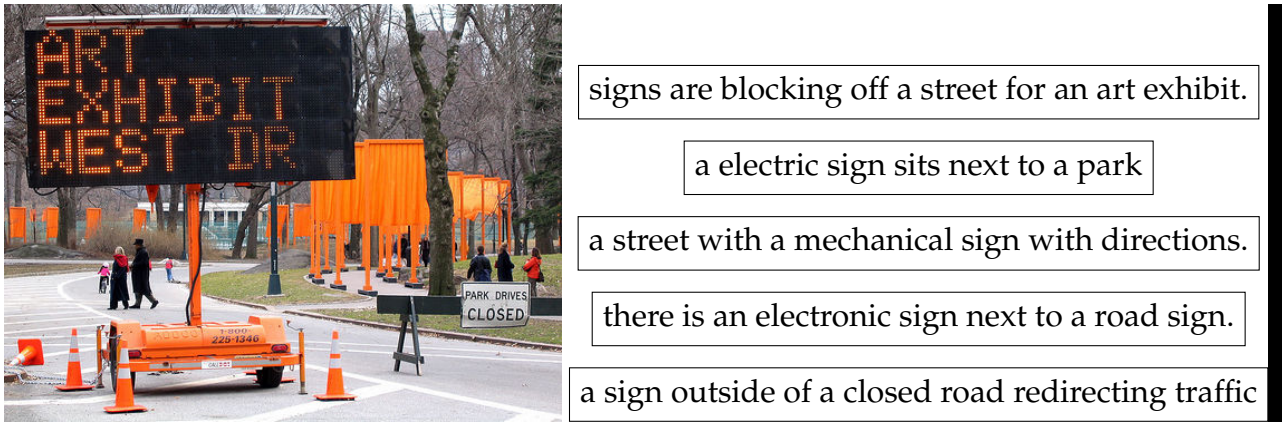


Figure 5.1: An example image from MS COCO with its associated captions

has inherent ambiguities and as a result it is hard to automatically evaluate performance metrics on usefulness of a caption. Consider an example image from MS COCO dataset as shown in Figure 5.1 with its associated human captions. Depending on the aspects of image a human focuses on, the generated captions demonstrate significant variability. A commonly used automatic performance metric is BLEU [111] which measures a modified precision score between the machine generated and human reference caption. However, a more comprehensive and accurate test of language generation is based on the Turing Test [131] which requires tremendous resources to be expended.

The research in image captioning has focussed on using Convolution Neural Networks (CNN) (2D for Images [163] and 3D for videos)based methodologies to identify semantic objects along with their attributes and relationships. Hao et. al [40] detect nouns, verbs and adjectives using CNN from an image and subsequently use an language model to generate human readable sentences. Vinyals et. al [145] use a CNN to generate features from the entire image and then use a language model to generate captions. These methods focus on the language output as well as the fine-grained content detection from the image. In contrast, to focus on the work performed in this thesis, we only generate semantically meaningful words associated with articulated objects.

5.1.1 Detection and Tracking

Similar in spirit to the recent work in generating video description using subject-verb-object triplets [78], we generate subject-verb-adjective descriptions. Using a single object track, we generate semantic descriptions of the activity in the scene using finite-state grammar [26]. Subject part of the sentence is directly based on the object class being detected. In our current setup, we only generate two types of verbs, "motion" and "interaction". The "motion" verbs are chosen from "entry", "exit", "move", "merge" and "split". For single object tracks, we generate an adjective based on temporal movement directions, 'Left','Downwards','Right' and 'upwards'. More adjectives such as big/small, color etc. can be chosen by learning various thresholds on the visual properties of the objects [12].

Figure 5.2: Tracked Person with Generated Sentence 'Person moves downwards'

Figure 5.3: Tracked Person and Cart with Generated Sentence 'Person interacts with cart.'

Figure 5.2 and 5.3 show the sample output generated from single object and multiple object tracking respectively. The interaction between multiple objects

is determined based on the state space approach for tracking as proposed in Chapter 2.

5.1.2 Pose Attributes

Attributes of articulated objects also form an essential part of description of an articulated object. Various attributes based on size, shape, texture etc. can be proposed. In this work, we restrict ourselves to the attributes based on pose of the object. Specifically, we experimentally evaluate the pose attribute "open/close" for a surgical tool which is used to provide semantic feedback to an operating surgeon.

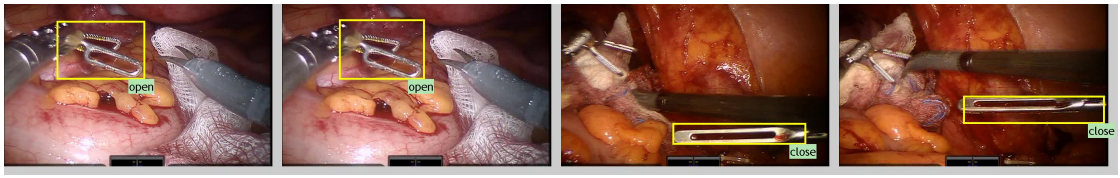


Figure 5.4: Generated results of Open/Close attribute on surgical videos

Figure 5.4 shows the output from the pose related attribute prediction for a surgical tool. For more details, we refer the reader to our paper on surgical attributes [86]. For the rest of this chapter, we discuss the information exchange and compositionality of language that can be exploited for computer vision.

5.2 Can Language inform Vision?

Consider a robot that can manipulate small building-blocks in a tabletop workspace, as in Figure 5.5. Task this robot with following vocal human utterances that guide the construction of non-trivial building-block structures, such as *place an orange rectangle on top of the blue tower to the right of the green tower*. This experimental setup, although contrived, is non-trivial even for state-of-the-art frameworks. The robot must be able to interpret the spoken language (audio perception); segment individual structures, *orange rectangle*, (visual perception); it must be able to reason about collections of structures, *blue tower*, (physical modeling); it must be able to relate such collections, *to the right of*, (linguistics); and

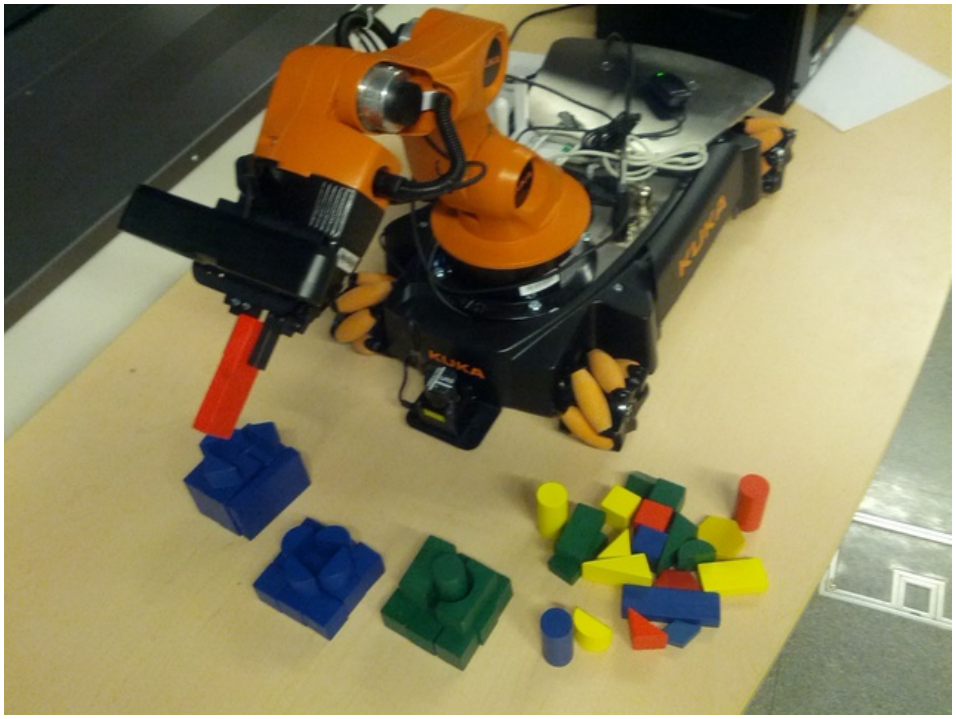


Figure 5.5: Our overarching goal is to improve human-robot/robot-robot interaction across sensing modalities while aiming at generalization ability. Multi-modal compositional models are important for effective interactions.

it must be able to execute the action, *place*, (manipulation). These challenging points are underscored by the frequency of table-top manipulation as the experimental paradigm in many recent papers in our community, e.g., [93, 102].

To achieve success in this experimental setup, the robot would need to satisfy Jackendoff's Cognitive Constraint [61], or at least a robotic interpretation thereof. Namely, there must exist a certain representation that relates percepts to language and language to percepts because otherwise the robotic system would neither be able to understand its visual-linguistic percepts nor execute its tasks. This and similar cognitive-semantic theories led to symbol-grounding [147] and language-grounding [128, 24].

Most existing work in symbol- and language-grounding has emphasized tying visual evidence to known language [103, 14], learning language models in the context of navigation [24] and manipulation tasks [73, 137] for a fixed set of perceptual phenomena, and even language generation from images and video [30, 11, 77]. Considering a pre-existing language or fixed set of percepts

limits the generality of these prior works. Recently, one method has enabled joint perceptual-lingual adaptation to novel input [102], but it does not explicitly capture the compositional nature of language hence leading to scalability challenges.

Indeed, the majority of works in language grounding do not exploit the compositional nature of language despite the potential in doing so [172]. One major limitation of non-compositional representation is the resulting overwhelming learning problem. Take, the example of *orange rectangle* and *green tower* from earlier. The adjectives *orange* and *green* are invariant to the objects *rectangle* and *tower*. Compositional representations exploit this invariance whereas non-compositional ones have combinatorial growth in the size of the learning problem.

In this work, we exploit the compositional nature of language for representing bimodal visual-audial percepts describing tabletop scenes similar to those in our example (Figure 5.5). However, we do not directly learn the compositional structure of these percepts—attempts at doing so have met with limited success in the literature, given the challenge of the structure inference problem [43, 116]. Instead, we ground the bimodal representation in a language-based compositional model. We fix a two-part structure wherein groupings of visual features are mapped to audio segments. The specific mapping is not hand-tuned. Instead, it is automatically learned from the data, and all elements of the compositional model are learned jointly. This two-part compositional structure can take the form of adjective-noun, e.g., *orange rectangle*, or even noun-adjective; the method is agnostic to the specific form of the structure. The structure is induced by the data itself (the spoken language).

The specific representation we use is a sparse representation as it allows interpretability because the signal is represented by few bases while minimizing a goodness of fit measure. There is increasing physiological evidence that humans use sparse coding in representation of various sensory inputs [13, 94, 109]. The need for sparse coding is supported by the hypothesis of using least energy in neuron's excitation to represent input sensory data. Furthermore, evidence suggests the multi-modal sensory data is projected together on a common basis [74], like we do in our compositional model.

We have implemented our compositional sparse model learning for bimodal percepts in a tabletop experiment setting with real data. We observe a strong ability to learn the model from fully observed examples, i.e., the training set consists of all classes to be tested. More significantly, we observe a similarly strong ability to generalize to unseen but partially observed examples, i.e., the testing set contains classes for whom only partial features are seen in the training set. For example, we train on *blue square* and *red triangle* and we test on *blue triangle* and *red square*. Furthermore, this generalization ability is not observed in the state-of-the-art joint baseline model we compare against.

5.3 Model

We describe our modeling approaches in this section. First, we begin by introducing the basic bimodal paired sparse model, which learns a sparse basis jointly over specific visual and audial modalities. This paired sparse model is not new [167, 153, 148]; it is the fabric of our compositional sparse model, which we describe second. The novel compositional sparse model jointly learns a mapping between certain feature/segment subsets, which then comprise the compositional parts to our model, and the paired sparse model for each of them.

5.3.1 Paired Sparse Model

Paired sparse modeling is driven by two findings from neurobiology [13, 94, 109, 74]: i) sparsity in representations and ii) various modality inputs are directly related. We hence use paired dictionary learning in which individual sensory data is represented by a sparse basis and the resulting representation shares coefficients across those bases. We are inspired by the success of paired dictionary learning in visualizing images from features [148], cross-style image synthesis, image super-resolution [153, 167] and beyond.

We adapt paired dictionary learning to our problem by learning over-complete dictionaries for sparse bases in both the visual and audial domain while using the same coefficients across domain-bases. Following similar notation to [148], let x_i, y_i represent visual and audio features for the ith sample. The features rep-

resent a useful encoding of visual and audio data in vector form (Section 5.4). The audio and visual features are related by the function mapping, $x_i = \phi(y_i)$. We seek to estimate forward (ϕ) and inverse (ϕ^{-1}) mappings while representing the audio and visual features with over-complete dictionaries (bases) U and V, respectively, coupled by a common sparse coefficient vector α:

$$x_i = U\alpha_i \quad \text{and} \quad y_i = V\alpha_i \ . \tag{5.1}$$

Sparsity in the coefficient vector is enforced by an l_1 metric [139] as $||\alpha||_1 \leq \lambda$. This ensures that only few bases are actively used for representing a particular input. For a given training dataset of size N, the over-complete dictionaries U and V, and the sparse coefficient vectors $\{\alpha\}_i$ are jointly estimated by minimizing the l_2 norm of the reconstruction error in both bases:

$$\begin{aligned} \arg\min_{U,V,\alpha} \sum_{i=1}^{N} \left(\|x_i - U\alpha_i\|_2 + \|y_i - V\alpha_i\|_2 \right) \\ \text{s.t.} \quad \|\alpha_i\|_1 \leq \lambda \ \forall i, \|U\|_2 \leq 1, \|V\|_2 \leq 1 \ . \end{aligned} \tag{5.2}$$

Note that the bases of the over-complete dictionaries are further constrained to belong to a convex set such that individual bases have l_2 norm less than or equal to unity.

The inverse mapping ϕ^{-1} for a novel sample is found by first projecting y on the learned dictionary V and then using the obtained coefficients α^* to compute $x = U\alpha^*$. The process of finding these coefficients involves the following optimization problem:

$$\alpha^* = \arg\min_{\alpha} |V\alpha - y|_2^2 \text{ s.t } \|\alpha\|_1 \leq \lambda \ . \tag{5.3}$$

Similarly one can obtain the forward mapping by first projecting x on learned dictionary U to obtain α^* and then using the learned dictionary V to obtain y. Thus, the estimation of forward and inverse mapping gives one the ability to go from audial features to visual features and vice versa. We use the open source sparse coding package SPAMS [99] for solving all the sparse optimization problems.

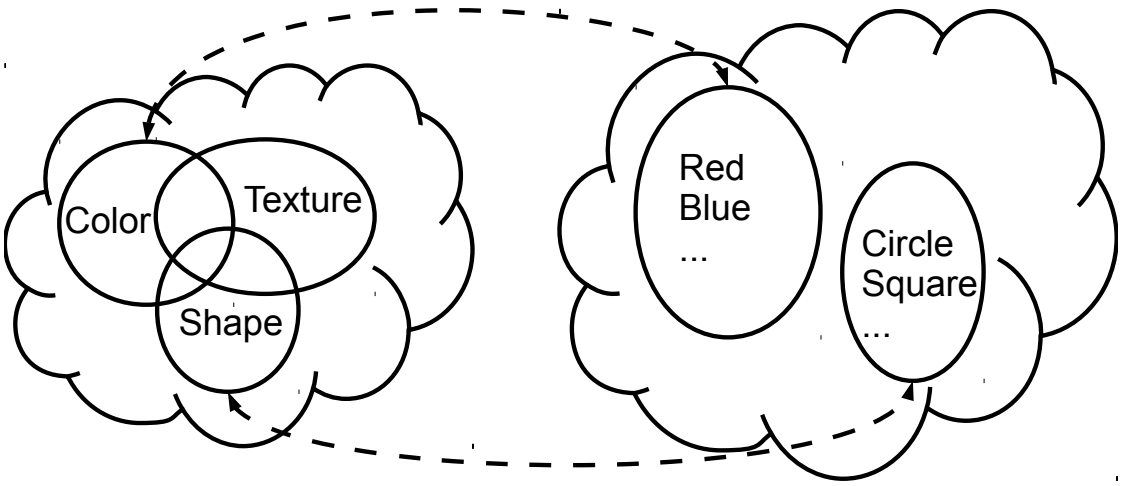


Figure 5.6: Mapping the physical concepts from visual domain such as color, texture and shape to the spoken language domain

5.3.2 Compositional Sparse Model

The paired model can link the two domains, but it can not exploit the compositionality inherently present in the language. Consider, again, the utterance *red square*. The part *red* describes the color of the object and the part *square* describes the shape of the object. The two parts are captured by distinctive and co-invariant visual features. We can hence explicitly map individual percepts between domains. Figure 5.6 illustrates the kind of mappings we expect to obtain between physically grounded concepts from the visual and audial (spoken human language) domain.

Consider n concepts, e.g., shape, from visual domain $\mathcal{V}$, which are linked to n concepts from the audial domain $\mathcal{A}$. This linking can be linearly represented by a matrix H, such that $\{\mathcal{V}_i\} = H\{\mathcal{A}_j\}$ for all $\{i, j\} = \{1, 2, ...n\}$. Here, we assume that one visual concept is linked to one and only one audial concept, implying the following two constraints on the matrix: 1) $\sum_j H(i,j) = \sum_i H(i,j) = 1$ and 2) each entry of the matrix can only be 1 or 0. Hence H is a permutation matrix.

The linking matrix H can be time-varying due to nature of spoken language: *red rectangle* and *rectangle red* mean the same thing for a human but meaning different things for representation H. In this work, we assume the audial domain has lingual structure, i.e, it has the same ordering of concepts in spoken

language. The visual feature a natural consistency induced by the ordering of the features.

The mapping between the ith audial and visual example is represented by

$$\mathbf{x_i} = \Phi \star H\mathbf{y_i} \ , \tag{5.4}$$

where $\mathbf{x_i} = [x_i^1, x_i^2, ..., x_i^n]$, $\mathbf{y_i} = [y_i^1, y_i^2, ..., y_i^n]$ with superscript denoting the feature representation from a particular visual or audial concept, $\star$ denotes the element-wise product operator. Φ is tensor of operators $\{\mathcal{L}_{p,q} : 1 \leq \{p,q\} \leq n\}$ that maps audial features (x_i^q) from q^{th} concept to visual features (y_i^p) of p^{th} visual concept with paired dictionary learning, as described in the previous section.

We jointly solve the following optimization problem:

$$\begin{aligned} \arg \min_{U^k, V^k, \alpha^k} & \sum_{i=1}^{N} \sum_{k=1}^{n} (\|x_i^k - U^k \alpha_i^k\|_2 + \|y_i^k - V^k \alpha_i^k\|_2) \\ \text{s.t.} \quad & \|\alpha_i^k\|_1 \leq \lambda^k \ \forall \{i,k\}, \|U^k\|_2 \leq 1, \|V^k\|_2 \leq 1. \end{aligned} \tag{5.5}$$

The inverse mapping $\mathbf{y} \mapsto \mathbf{x}$ is obtained by first projecting $\mathbf{y}$ on the learned basis

$$\alpha^* = \arg \min_{\alpha^k} \sum_{k=1}^{n} |V^k \alpha^k - y^k|_2^2 \text{ s.t } ||\alpha^k||_1 \leq \lambda \tag{5.6}$$

and then using the linking matrix $H(\cdot)$. Unlike with a single paired dictionary, there is an additional optimization problem required for forward and inverse mapping to estimate H after estimating the Φ tensor from the learned bases,

$$\begin{aligned} \arg \min_{H \in \mathcal{H}} & \sum_{i=1}^{N} \|\underline{x_i} - \Phi \star H\underline{y_i}\|_2 \\ \text{s.t.} \quad & \mathcal{H} : H(i,j) = \{0,1\}, \sum_j H(i,j) = \sum_i H(i,j) = 1 \ , \end{aligned} \tag{5.7}$$

where $\mathcal{H}$ is the space of all permutation matrix.

Observe that the optimization problems in Eqs. 5.5 and 5.6 become complex as Φ and H are to be simultaneously estimated involving n^2 sparse mappings

and n parameters of the permutation matrix. However, the constraints imposed on the linking matrix H ensure that only n mappings are used. Hence, we proceed in a sequential manner, first estimating the matrix H and then only learning the n sparse mappings that are required. Notice also that when $n = 1$, compositional sparse learning reduces to paired sparse learning.

We estimate the matrix H based on the intuition that distance in visual and in audial feature representations of the same physically-grounded element should co-vary. Correlation coefficients can not be directly estimated because the visual and audio features belong to different vector spaces. Instead, we estimate H based on clustering separately in each domain and then linking clusters across domains using the Hungarian algorithm [81] and V-measure [126] for cluster similarity.

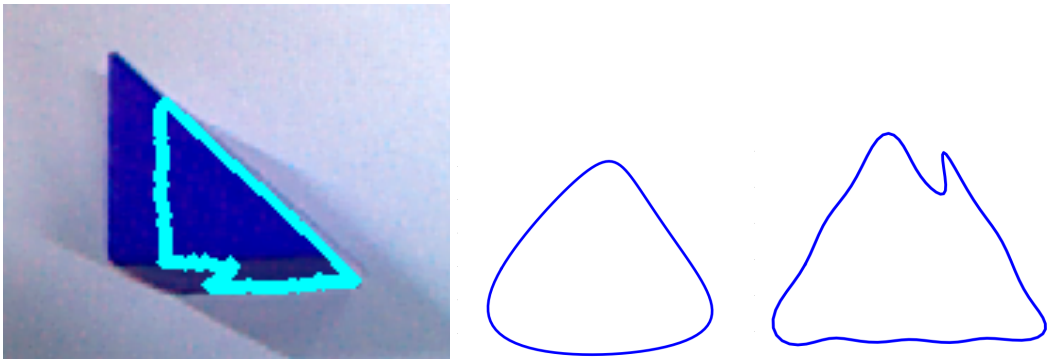
5.4 Features

In this work, we restrict our study to the concepts of color and shape, without loss of generality. We extract color and shape features from the visual domain and segment the audio into two parts (ideally, words) representing individual concepts.

5.4.1 Visual

Similar to [102], we first segment the image (in HSV space). Since the shapes used in current work consist of basic colors, they are relatively easily segmented from the background using saturation. To represent color, we describe each segment by its mean RGB values. To represent the shape of each segment we opt for a global shape descriptor based on Fourier analysis of closed contours [80].

Fourier features represent a closed contour by decomposing the contours over spectral frequency. Lower frequencies capture the mean of shape while higher frequencies account for subtle variations in the closed contours. The visual system of humans is found to have capabilities to form two- and three-dimensional percepts using only one-dimensional contour information [37].



(a) ***Segmented Image*** **(b)** ***2 Harmonics*** **(c)** ***10 Harmonics***

Figure 5.7: Fourier Representation of a triangular shape with 2 *and* 10 *fourier harmonics*

We extract contours of the segmented/foreground object and use chain codes [47] to simplify analytical extraction of elliptic Fourier features [80]. After removing the constant Fourier component, we introduce rotational invariance by rotating of Fourier elliptical loci with respect to the major axes of the first harmonic [80]. Figure 5.7 shows the Fourier feature representation of a contour of a segmented triangular shape. It can be seen that the shape is represented as a triangle even with 2 harmonics given the imperfect segmentation. Note, the representation is invariant to position, rotation and scale and hence the figure shows the triangle in a standardized coordinate frame.

5.4.2 Audio

We use Mel Frequency Cepstral Coefficients (MFCC) [96] which are widely used in audio literature to represent audio signals. MFCC features are obtained by dividing the audio signal into small temporal frames and extracting cepstral features for each frame. This feature models important human perception characteristics by ignoring phase information and modeling frequency on a "Mel" scale [96]. Since the audio files are of different time lengths, we extract the top 20 frames with maximum spectral energy.

	a_1	a_2
v_1	0.1	1
v_2	0.4	0.1

Table 5.1: V-measure distance matrix between the feature representation. v_1 and v_2 represent RGB and Fourier descriptor features, respectively, a_1 and a_2 represent the feature extracted from first and second audio segment.

5.5 Experiments and Results

We perform rigorous qualitative and quantitative evaluation to test generalization and reproduction abilities of the paired sparse and compositional sparse models. Quantitative performance is estimated to assess reproduction ability of the algorithm by performing 3-fold cross-validation. Qualitative performance is evaluated to infer the generalization capabilities of the proposed compositional sparse model and compare its performance with non-compositional paired sparse model. For the purpose of presenting results, we only consider mapping from audio to visual in order to depict results. However, with the model both audio to visual and visual to audio representations can be derived.

We extract 260 dimensional audio features from selected 20 audio frames, 20 fourier harmonics, 3 dimensional color feature and fix $\lambda = 0.15$ for all of the experiments.

Table 5.1 shows the evaluation of linking matrix H based on the ground-truth data. RGB features and shape features are denoted by v_1 and v_2 respectively. Audio feature a_1 represent features from utterance of shape and a_2 represents features from utterance of color. This matrix gives a very simple alignment of $v_1 \mapsto a_2$ and $v_2 \mapsto a_1$ which will be used in compositional model.

5.5.1 Dataset

We acquired a new dataset of shapes and colors with 156 different examples (Table 5.2) of images showing a shape captured from a camera in various rotation and translation on the tabletop. We generated machine audio that describes the color and shape of the capture image (e.g., *red rectangle*) with random speeds. We also produced segmented audio by generating machine audio separately for

Shape\Color	Blue	Green	Red	Yellow	Total
Circle	6	6	2	6	20
HalfCircle	6	4	4	4	18
Rectangle	6	6	6	2	20
Rhombus	10	0	0	0	10
Square	10	10	10	10	40
Triangle	8	6	8	6	28
Trapezium	0	0	10	0	10
Hexagon	0	0	0	10	10
Total	46	32	40	38	

Table 5.2: Shape and Color Exemplars in the dataset

color and shape of the referred image to be used with the compositional model.

5.5.2 Visualization

To generate a visualization (audial-to-visual generation), we use inverse mapping ϕ^{-1} and $(H \star \Phi)^{-1}$ to generate visual features from audial features. The generated visual feature consists of Fourier features and mean RGB intensity values. Since Fourier features are rotation and translation invariant, a close representation of original image can not be generated. For visualizing results, we reconstruct the contour using Fourier features and fill the contour with predicted RGB values.

5.5.3 Reproduction Evaluation

For reproduction, we seek to evaluate the performance of a robot for a theoretical command, *pick a 'red rectangle' from a box full of all the shapes*, which is a subset of the broader picture described in the Introduction. We perform a 3-fold cross-validation study to assess this retrieval performance by dividing the dataset into 3 parts, using 2 parts for training and remaining part for testing (and then permuting the sets). We test retrieval performance for different concepts (color and shape) separately for paired sparse learning and compositional sparse learning. A color or shape is determined to be correctly understood by the robot if the said color or shape is present in top k retrieved examples. Re-

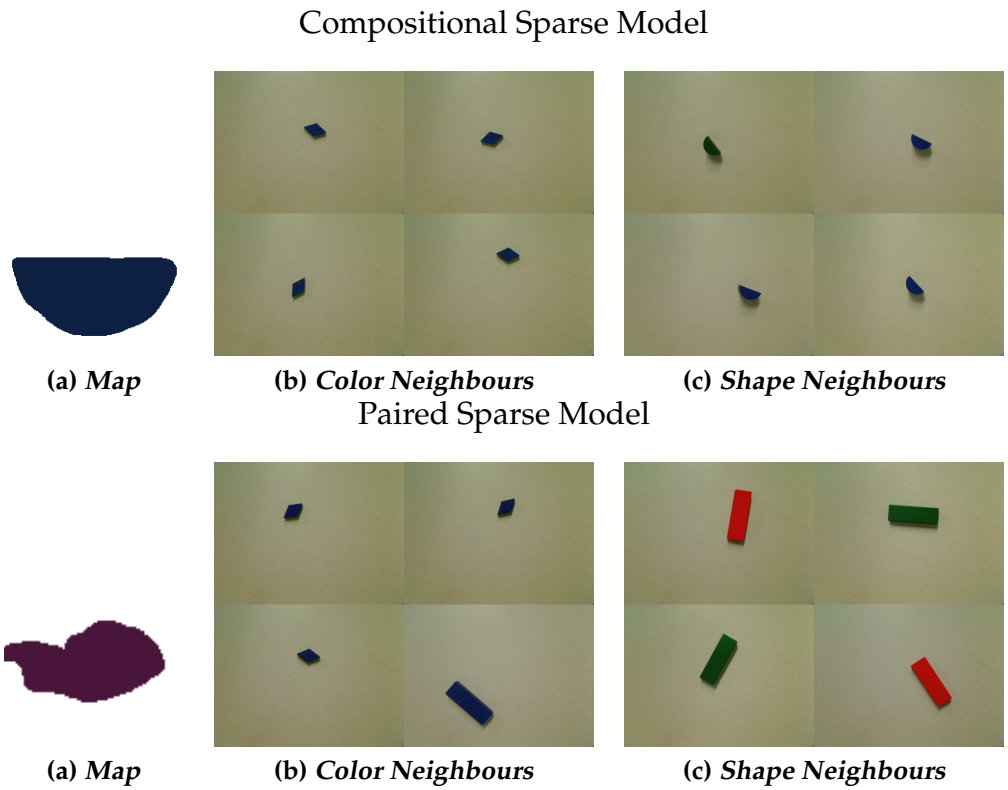


Figure 5.8: For the audial utterance blue halfcircle*, (a) generated image by mapping from audial to visual domain. (b), (c) retrieval of top 4 color and shape neighbors by both models. For both the models, a feature representation of audial utterance is mapped to visual representation which includes color and shape representation. This visual representation is then used to find the nearest neighour in the entire dataset to generate the color and shape neighbors in (b) and (c) respectively.*

trieval is performed by first extracting the audial feature from the audial stream, using the trained linking matrix to extract visual features and then picking the closest object from all the training examples.

The closeness of a visual object to generated visual feature is measured by a distance metric in the visual feature space. We compare the feature vectors to extract k nearest neighbors using an l_1 distance, in the appropriate concept feature subspace. For evaluation, we set the parameter k to be 5, which means that if there is a match in the top 5 nearest neighbors, the trial is deemed to be successful. Figure 5.8 shows the reproduction performance for an audial utterance *blue halfcircle*. It is observed that while the compositional model gets both the color and shape correct, the paired model fails in reproducing the correct

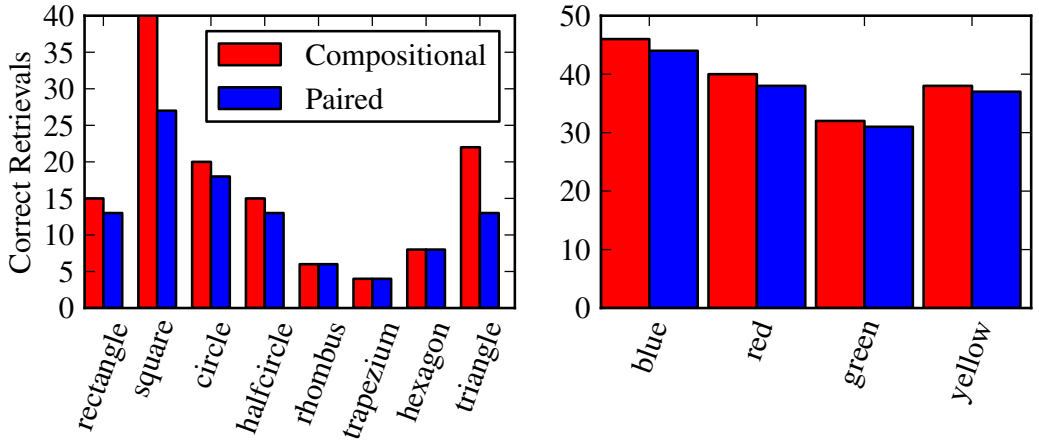


Figure 5.9: Comparison of correct retrievals by two different algorithms compositional and non-compositional. Left image shows the retrieval of shape features, while right shows that of color.

shape.

Figure 5.9 compares the quantitative retrieval performance for the compositional and paired models. It is observed that the paired model forms a good baseline for evaluating the compositional model, which always achieves equivalent or better performance. The reason for good performance of the paired model can be attributed to the presence of similar examples in the training data.

5.5.4 Generalization Evaluation

We test compositional sparse and paired sparse models with respect to their generalization capabilities on novel samples. Here, we test generalization across color and shape. Generalization is evaluated by generating images of a particular color and shape whose training examples have been removed from the dataset. For a good generalization performance, the model must generate implicit meaning of utterances such as *green* and *triangle*.

Figure 5.10 shows the pictorial results from various audio utterances from compositional sparse and paired sparse models. For the audio utterance *blue circle*, both models get the right color but compositional model achieves better shape generation which is the case for utterance *green rectangle* as well. For the audial utterance *red square* compositional model achieves both shape and color

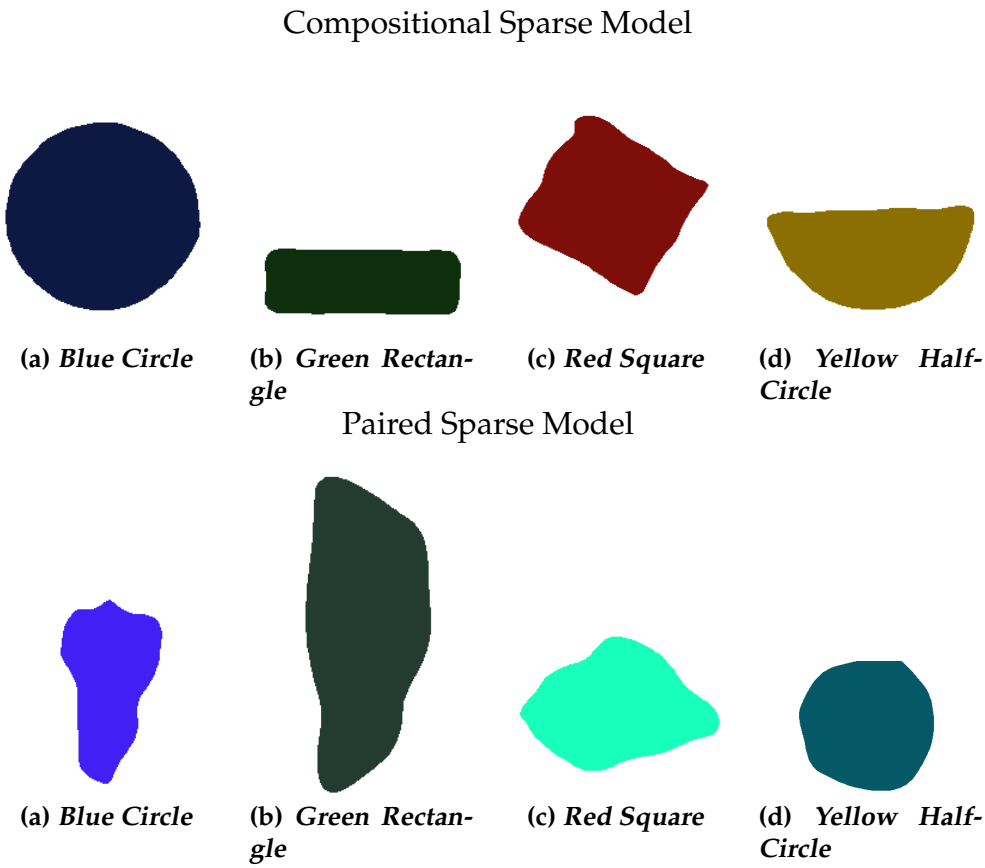


Figure 5.10: Generalization performance result depiction for audial utterances (a) blue circle *(b)* green rectangle *(c)* red square *(d)* yellow halfcirle

while the paired model is not able to represent color. From these examples, it is clear that compositional model can handle generalization both across shape and color much better compared to the paired model. The paired sparse model—as reflected in these results—is incompetent for this task because it does not distinguish between individual percepts.

For qualitative evaluation of generalization capabilities, we use evaluation by human subjects. For this, we generate two sets of images, one from the compositional model and one from the paired model. Each set of these images is then presented to human subjects through a web-based user interface, and the humans are asked to "Describe the color and shape of the above image" while being presented with the color and shape options along with "None of these" from the training data. Note that in this experiment the human subject is not

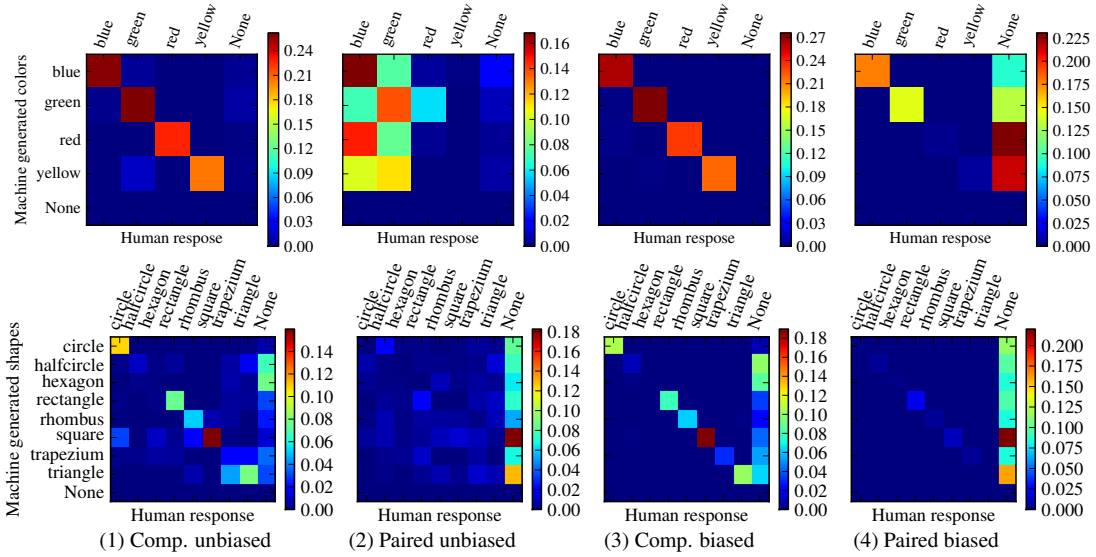


Figure 5.11: Confusion matrices for generalization experiments evaluated by human subjects. Rows are for different features: colors and shapes. Columns from left to right are four different experiments (1) Images generated by compositional model are evaluated by humans with unbiased *questions like "Describe the color and shape of this image" from fixed set of choices (2) Paired model with* unbiased *questions. (3) Compositional model with biased questions like "Is the shape of generated image same as the given example image?" 4) Paired model with biased questions.*

shown any samples from the training data. Hence, we call these experiment *compositional unbiased* and *non-compositional unbiased* depending on the generating model.

In another set of experiments we *bias* the human subject by showing them an example image of the color and shape for which the image has been generated. The subject is expected to answer in "Yes" or "No" to the question: "Is the color (shape) of the above image same as the example image?" Whenever the subject says "Yes", we take the response as the expected color/shape; for "No" we assume "None of these" option.

Figure 5.11 shows the human qualitative performance metrics for this test. It is observed that color generalizes almost perfectly using our proposed compositional sparse model while the paired sparse model gives poor performance in both biased and unbiased human evaluation. On the generalization of shape, the compositional model again achieves much better performance over the baseline paired model in both biased and unbiased experiments. Using the inter-

nal semantics of humans, it is observed that *halfcircle* is frequently represented as *rectangle* or *trapezium* for the compositional model. It is likely because of the shape feature with invariance whose closed contour representation is not enough to distinguish perceptually similar shapes. Furthermore, *triangle* is often mistaken as *trapezium* which can be explained by a similar starting sound. It is seen that biased results give better performance denoting improved assessment after recalibration of human semantics to current experimental shapes.

5.6 Conclusion

We demonstrated the language generation capabilities from articulated object detection, tracking and pose estimation methods as described in Chapter 2 and Chapter 3. We also presented a novel model representing bimodal percepts that exploits the compositional structure of language. Our compositional sparse learning approach jointly learns the over-complete dictionaries, sparse bases, and cross-modal linking matrix. In contrast to prior work in bimodal modeling which is primarily discriminative in nature, e.g., [128, 125], our compositional sparse learning approach is generative and hence transparent. We demonstrate the effectiveness of sparsity and compositionality by both qualitative and quantitative evaluations on a dataset of different shapes of multiple colored objects.

Our results show that indeed it is possible to drive computer vision using the language and vice-versa by learning relationships across the modalities. In future, we will extend the compositionality by removing the constraint of one-to-one mapping which will be important for percepts grounded in more than one basic feature representation of the input. Furthermore, it is now possible to extend the shape and color problem to shape, color and motion problem which paves the way for solving the grand problem outlined in this thesis. A simple extension of "red triangle" example to shape, color and motion problem is "move red triangle" which when integrated with articulated structure estimation can potentially solve "Open the door" problem.

The experiments to drive vision using language also demonstrate the limitation of our current language model. The assumption of one-to-one mapping between concepts is often violated depending on the context. For example, a

"big cube" might be an appropriate description for a "blue cube" or "big blue cube" depending on the context. A simple extension of our current framework needs to incorporate spatial context to define "big". Furthermore, to integrate verbs , such as "play", the temporal context needs to be taken into account. A context dependent "H" matrix might extend to incorporate such extensions but it will tremendously increase the complexity of our model and hence the inference.

Chapter 6

Conclusion

In the introductory chapter, we identified two major shortcomings of contemporary research in computer vision: i) lack of temporal models, and ii) missing articulated structure estimation. We demonstrated the effective usage of temporal modeling for both object tracking and pose tracking. In the case of object tracking, temporal models enabled prediction of the location of objects solely using past observation. For the task of pose estimation, incorporation of a temporal model showed improved predictions by filtering the pose estimation resulting from static observations.

For articulated structure estimation specifically for novel objects with unknown structure, we presented a framework to estimate articulation solely from visual observations. The estimation of articulated structure along with an explicit temporal model significantly improved robot localization in an environment with moving articulated objects. Another important contribution outline in previous chapters was in the exploration of connections between human language and computer vision for effective human-robot communication. We demonstrated the utility of the compositional nature of natural language to enable effective language grounding via learning a mapping between semantic elements in vision and language.

6.1 Overview of Results

6.1.1 Tracking

We presented a tracking algorithm, Product of Tracking Experts (PoTE) which probabilistically combines output from various individual(Experts) trackers. This is in contrast to the strategy of designing a best tracker based on a combination of a subset of motion and appearance features. The PoTE method does not require individual trackers to generate a probabilistic output, and, consequently, allows for effective usage of various probabilistic and non-probabilistic methods, such as detection output, background subtraction, and point based trackers. As demonstrated by experiments on surgical tool tracking, PoTE significantly improves tracking results over a baseline tracker.

To address the problem of interactions in multiple object tracking, we proposed a state-space based interaction model. This model effectively increases uncertainty in tracking outputs of interacting individual objects and reduces that uncertainty given sufficient observations. This framework was tested on a publicly available person tracking dataset and our algorithm achieved state-of-the art results. The trajectory interpolation and entry/exit estimation was found to be robust for the tracking challenge presented by the PETS dataset. To summarize, we proposed a combination of trackers to track individual object and a state-space based interaction model for multiple object tracking.

6.1.2 Pose Estimation

For pose estimation, we presented a learning based approach which maps a visual representation of the object directly to pose in real-time. Such methods are known to fail in generalizing beyond the learned data in contrast to the generative methods based on geometrical and appearance model. To reduce the degradation of pose outputs on novel inputs, we incorporated an explicit process model to enable pose tracking. The Kalman filter based tracking method benefited from the choice of our Gaussian Process based regression which generates variance measure in addition to the mean estimate. We tested our algorithm on a customized box trainer setup which enabled us to collect a vast

amount of ground-truth pose data in addition to streaming visual data. Our pose estimation algorithm generates a robust pose estimate by first predicting pose using Gaussian Process based regression and then filtering the resulting pose estimates using a process model.

6.1.3 Articulation Estimation

Pose estimation relies on the assumption of a known articulated structure (the way joints are connected in an articulated body) which may not be readily available for novel objects. We presented an online articulation estimation method which can estimate this structure given sufficient number of visual observations. The online nature of the proposed algorithm enables the estimation process to converge to the true model without any restrictive assumption on the type of articulated structure. In addition to the online nature, we explicitly separated the motion parameters (e.g. opening angle of a door) from configuration parameters (e.g. motion axis of a door). This separation generated better results compared to the combined model for configuration parameters estimation.

We developed a framework to incorporate temporal models on the motion parameters which allows one to predict the state of the articulated body in future (e.g. location of a door). We integrated the proposed temporal modeling with articulated structure estimation into an off-the-shelf Extended Kalman Filter(EKF) based Simultaneous Localization and Mapping(SLAM) algorithm. The resulting Articulated-EKF-SLAM approach generated lower translation and rotation errors in robot localization compared to the traditional EKF-SLAM algorithm in an environment with articulated motion.

6.1.4 Vision and Language

For language generation, we used the semantic output from object tracking and pose estimation. Object tracks are used to generate spatial location (e.g. to the left of) and temporal movement (e.g. entered, moved to the right). In order to generate natural language output, we used the subject-verb-adjective triplet. Pose information was used to further describe fine-grained details of a rigid body, specifically the opening and closing of a surgical tool. We experimen-

tally evaluated our open/close attribute generation performance on a surgical dataset.

On the language understanding front, we enabled deep integration of vision and language by using compositional nature of language. We evaluated the proposed methodology on two semantic aspects: i) shape, and ii) color. Due to the nature of our algorithm, our robot could generate visual output of "green square" by simply looking at "green rectangle" and "red square". This augments the generalization capabilities of language-grounding methodologies.

6.2 Applications

6.2.1 Surveillance

Detection and tracking have been vastly adopted for use in visual search and automated surveillance. Visual search catalogs a feature representation of objects which can then be used to localize an entity. Based on our detection and tracking output, we have developed a demo application to query vast amounts of video data to search for a specific person and additionally generate a human readable summary of events in a video by integrating language models with tracking output. This can potentially enable text search and pattern mining while avoiding the time-consuming, and labor-intensive task of manually looking for events in surveillance feeds.

6.2.2 Surgical Semantic Feedback

Increasingly, surgical procedures are being performed using minimally invasive surgery (MIS) techniques which rely on the endoscopic camera to provide a rich real-time sensing channel from surgical site to the surgeon console. MIS techniques have significantly reduced the bleeding, and as a result, the recovery time, compared to traditional laparoscopic surgeries. However, the physical separation between the patient and surgeon engenders its own set of problems due to lack of force feedback and reliance on software, hardware and network elements. This has resulted in limited somatosensory feedback available to the

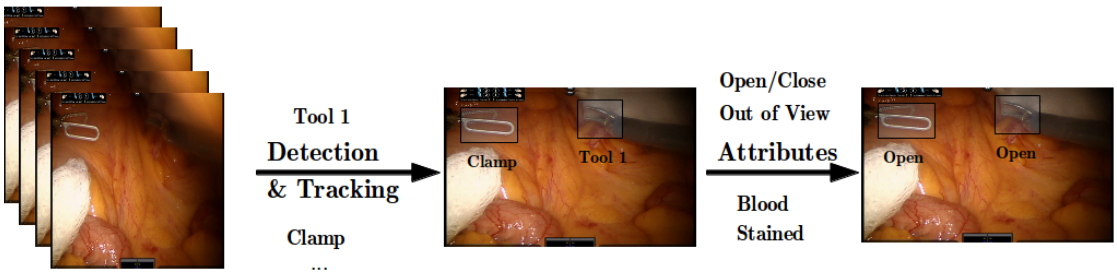


Figure 6.1: Surgical semantic feedback from tracking and attribute generation. Two different surgical instruments "Tool 1" and "Clamp" are detected and tracked in images. This tracked bounding box is used in an attribute generation framework, such as open/close, which can be used as feedback to the surgeon.

surgeon in addition to the safety challenges arising due to the intermediary interfaces. To compensate for the loss of this feedback and enable additional safety measures, we propose using the articulation motion techniques as proposed in this thesis. Figure 6.1 shows the type of feedback that can be made available to surgeon based on detection, tracking, and attribute generation. This information when coupled with critical surgical landmark (e.g. blood vessels, heart) detection can be used to guide the surgeon in avoiding errors during surgery [127].

6.2.3 Motion Planning in Dynamic Environments

Motion planning algorithms generate control inputs for a robot in moving from one point to another point in the environment, given robot's process model and models of the environment (specifically obstacles and free space). The most popular way of solving a motion planning problem is to sample way-points in the environment and then find a feasible path between those way-points while avoiding obstacles. However, often the dynamic nature of environment itself is not taken into consideration. Consider a hypothetical scenario where a robot needs to attempt going inside a garage whose door is closing. Integration of our online articulation estimation approach with the traditional motion plan-

ning approach can potentially generate aggressive trajectories to maneuver in a dynamic environment.

6.2.4 Minimum Time Scene Understanding

Computer vision, as applied in contemporary research, analyzes passively sampled data while ignoring the process of data acquisition. Studying the process of capturing this data or "Active Vision", seeks to identify the data to be captured in the next time instant that would directly lead to an improvement in performance. Using active vision, we can endow the robot with an ability to identify the optimal path which enables it to explore an unknown environment in minimum time while maximizing the amount of information acquired from its observations.The importance of active perception is also substantiated by the way human vision has evolved. Humans have significant "active" abilities in their vision such as the ability to control the light, focus, position and orientation of the head. Human perception puts a strong emphasis on "what the mind wants to see" [22] along with understanding what the mind actually "sees".

A complicated yet rewarding insight for improving computer vision performance lies in incorporating this vital feedback mechanism of human perception with the current passive data acquisition methodologies. Our articulation estimation framework can enable this feedback mechanism when incorporated into a scene understanding framework. For example, a drawer when observed along the prismatic axis is quite ambiguous compared to when observed perpendicular to the motion. The probabilistic interpretation of articulated objects $P(X(t))$ as developed in Chapter 4 directly provides information about uncertainty in position estimates. This information can be incorporated into a scene understanding framework that seeks to minimize errors in position estimates in its representation of the physical environment.

6.2.5 Language Grounding

As outlined in the motivation of this thesis, we need to enable robots to be able to communicate about the physical environment with humans in natural language. The more conventional machine learning based approaches for human-

robot communication require the investment of a significant amount of time and effort to generate curated data to resolve ambiguities in human language. Fortunately, however, human language is often based on context which can be understood by applying some of the approaches as proposed in this thesis. For example, consider the example presented in the introductory chapter of "Open the door", our detection, tracking, pose, articulation estimation enable a robot to understand the context of the scene. This context in conjunction with the compositional nature of the language as presented in Chapter 5 enables a robot to understand the physical environment while reducing the amount of required curated ground truth data.

6.3 Future Work

6.3.1 Theoretical Development

Our experiments in tracking demonstrated the need for complex interaction models. Social interaction models especially those for crowded scenes [124] need to be explored for incorporation into our tracking framework. For pose tracking, we considered simple motion continuity as process models for incorporation into a Kalman filter. However, tracking results can be significantly improved by the knowledge of the action being performed, such as suturing in case of surgical tool pose estimation. These actions have an underlying pattern which can be characterized by a low-order probabilistic dynamic model [152].

Articulation estimation as presented in this thesis only estimated articulated objects with one moving link. However, we need to consider extensions to multiply connected articulated bodies, such as a robot arm. Another fundamental extension, as outlined in conclusion of Chapter 5, is to use language as a glue for semantic understanding, specifically articulated motion. This will not only enable communication between robots and humans about the physical world but will provide the robots with a tool to predict/understand the action as described by human language.

6.3.2 Experimentation

Experimentally, we need to test our Articulated-EKF-SLAM algorithm on a real indoor environment. This would require geometrical representation of objects, such as doors, chairs etc. This representation will then need to be localized in an incoming visual data stream to evaluate the transformation of an object between time frames which can then be used to estimate articulated structure (see section 4.3.2 for details). Furthermore, the articulated structure estimation framework needs to be tested for incorporation into smoothing based SLAM algorithms which use future observation apart from past observations to estimate robot and environment state.

Appendix A

Additional Derivations

A.1 Upper Bound on Point Tracker Error

We prove an analytical bound on tracking drift using point feature based tracking as referred in the original manuscript. In this proof we refer to equation numbers in original manuscript and follow the same notation. 1. Proof that Drift error by our KLT algorithm is bounded.
Considering drift error only in x coordinates, we get

$$x_B(t+1) = \sum_{j=1}^{J} w_j \left[x_f(t+1,j) + x_B(t) - x_f(t,j) \right]$$

$$x_B(t+1) - x_B(t) = \sum_{j=1}^{J} w_j \left[x_f(t+1,j) - x_f(t,j) \right]$$

because $\sum_{j=1}^{J} w_j x_B(t) = x_B(t)$. True movement in x coordinate of bounding box is $x_B(t+1) - x_B(t)$. Assuming that object is rigid and moving planar to camera, ideally each feature should move by the same amount. Let each feature j has an error of η_j.

$$x_f(t+1,j) - x_f(t,j) = x_B(t+1) - x_B(t) + \eta_j$$

$$x_B(t+1) - x_B(t) = \sum_{j=1}^{J} w_j \left[x_B(t+1) - x_B(t) + \eta_j \right]$$
$$= x_B(t+1) - x_B(t) + \sum_{j=1}^{J} w_j \eta_j$$

Hence total drift error is $\sum_{j=1}^{J} w_j \eta_j$. Let maximum residual error in KLT feature matching optimization be e_{max} and minimum error be e_{min}. Further. let us denote maximum drift error by η_{max}. Upper bound on the drift error is given by

$$w_j = \frac{e_j}{\sum_{j=1}^{J} e_j} \leq \frac{e_j}{J e_{min}} \leq \frac{e_{max}}{J e_{min}} \text{ (Obtained by normalization)}$$

Substituting this result in total drift error, we get

$$\sum_{j=1}^{J} w_j \eta_j \leq \sum_{j=1}^{J} \frac{e_{max}}{J e_{min}} \eta_j \leq \frac{e_{max}}{J e_{min}} \sum_{j=1}^{J} \eta_j \leq \frac{e_{max}}{J e_{min}} J \eta_{max} \leq \frac{e_{max} \eta_{max}}{e_{min}}$$

Please note that this bound is under the assumption of brigthness consistency (KLT) and rigid body motion.

Bibliography

[1] Priyanshu Agarwal, Suren Kumar, Julian Ryde, Jason J Corso, and Venkat N Krovi. Estimating human dynamics on-the-fly using monocular video for pose estimation. In *Robotics: Science and Systems*, 2012.

[2] Priyanshu Agarwal, Suren Kumar, Julian Ryde, Jason J Corso, and Venkat N Krovi. An optimization based framework for human pose estimation in monocular videos. In *Advances in Visual Computing*, pages 575–586. Springer, 2012.

[3] Priyanshu Agarwal, Suren Kumar, Julian Ryde, Jason J Corso, and Venkat N Krovi. Estimating dynamics on-the-fly using monocular video for vision-based robotics. *IEEE/ASME Transactions on Mechatronics*, 19(4):1412–1423, 2014.

[4] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. June 2015.

[5] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, pages 1–8, 2008.

[6] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 623–630. IEEE, 2010.

[7] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *CVPR*, 2011.

[8] B. Babenko, M.H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632, 2011.

[9] Tim Bailey. *Mobile robot localisation and mapping in extensive outdoor environments*. PhD thesis, Citeseer, 2002.

[10] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[11] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, L. Schmidt, J. Shangguan, J. M. Siskind, J. Waggoner, S. Wang, J. Wei, Y. Yin, and Z. Zhang. Video in sentences out. In *UAI*, 2012.

[12] Andrei Barbu, Alexander Bridge, Zachary Burchill, Dan Coroian, Sven Dickinson, Sanja Fidler, Aaron Michaux, Sam Mussman, Siddharth Narayanaswamy, Dhaval Salvi, et al. Video in sentences out. *arXiv preprint arXiv:1204.2742*, 2012.

[13] Horace B Barlow. Possible principles underlying the transformation of sensory messages. *Sensory communication*, (13):217–234, 1961.

[14] K. Barnard, P. Duygulu, D. Forsyth, N. de Frietas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[15] Charreau S Bell, Gustavo A Puerto, Gian-Luca Mariottini, and Pietro Valdastri. Six dof motion estimation for teleoperated flexible endoscopes using optical flow: A comparative study. In *ICRA*. 2014.

[16] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR*, volume 1, pages 744–750, 2006.

[17] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *Journal on Image and Video Processing*, 2008.

[18] S. Birchfield. KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker. Available: http://www.ces.clemson.edu/ stb/klt/, 2007.

[19] Wolfgang Birkfellner, Franz Watzinger, Felix Wanschitz, Rolf Ewers, and Helmar Bergmann. Calibration of tracking systems in a surgical environment. *IEEE Transactions on Medical Imaging*, 17(5):737–742, 1998.

[20] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. *ECCV*, pages 168–181, 2010.

[21] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *PAMI*, 33(9):1820–1833, 2011.

[22] William J. Broad. Computer scientists stymied in their quest to match human vision, September 1984. .

[23] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[24] D. L. Chen and R. J. Mooney. Learning to interpret natural language navigation instructions from observations. In *AAAI*, 2011.

[25] Changhyun Choi and Henrik I Christensen. 3d textureless object detection and tracking: An edge-based approach. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3877–3884. IEEE, 2012.

[26] N. Chomsky. Three models for the description of language. *Information Theory, IRE Transactions on*, 2(3):113–124, 1956.

[27] Cristina García Cifuentes, Marc Sturzel, Frédéric Jurie, Gabriel J Brostow, et al. Motion models that only work sometimes. In *BMVC*, volume 2, page 5, 2012.

[28] João Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[29] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005.

[30] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*, 2013.

[31] Christophe Doignon, Florent Nageotte, Benjamin Maurin, and Alexandre Krupa. Pose estimation and feature tracking for robot assisted surgery with medical imaging. In *Unifying perspectives in computational and robot vision*, pages 79–101. Springer, 2008.

[32] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009.

[33] Norman Richard Draper, Harry Smith, and Elizabeth Pownell. *Applied regression analysis*, volume 3. Wiley New York, 1966.

[34] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.

[35] Carl Henrik Ek and Danica Kragic. The importance of structure. In *15th International Symposium on Robotics Research. Flagstaff, AZ. 28 August-1 September 2011*, 2011.

[36] Carl Henrik Ek, Philip HS Torr, and Neil D Lawrence. Gaussian process latent variable models for human pose estimation. In *Machine learning for multimodal interaction*, pages 132–143. Springer, 2008.

[37] James Elder and Steven Zucker. The effect of contour closure on the rapid discrimination of two-dimensional shapes. *Vision research*, 33(7):981–991, 1993.

[38] Felix Endres, Jeff Trinkle, and Wolfram Burgard. Learning the dynamics of doors for robotic manipulation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3543–3549. IEEE, 2013.

[39] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[40] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John Platt, et al. From captions to visual concepts and back. *arXiv preprint arXiv:1411.4952*, 2014.

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

[42] Mark Fiala. Comparing artag and artoolkit plus fiducial marker systems. In *Haptic Audio Visual Environments and their Applications, 2005. IEEE International Workshop on*, pages 6–pp. IEEE, 2005.

[43] Sanja Fidler and Ales Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*. IEEE, 2007.

[44] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *The journal of Neuroscience*, 5(7):1688–1703, 1985.

[45] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, 1983.

[46] K. Fragkiadaki and J. Shi. Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In *CVPR*, June 2011.

[47] Herbert Freeman. Computer processing of line-drawing images. *ACM Computing Surveys (CSUR)*, 6(1):57–97, 1974.

[48] Yixin Gao, S Swaroop Vedula, Carol E Reiley, Narges Ahmidi, Balakrishnan Varadarajan, Henry C Lin, Lingling Tao, Luca Zappella, Benjamın Béjar, David D Yuh, et al. Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. 2014.

[49] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[50] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[51] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013.

[52] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, volume 1, pages 260–267. IEEE, 2006.

[53] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. *Computer Vision–ECCV 2008*, pages 234–247, 2008.

[54] Steven Gray, Subhashini Chitta, Vipin Kumar, and Maxim Likhachev. A single planner for a composite task of approaching, opening and navigating through non-spring and spring-loaded doors. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3839–3846. IEEE, 2013.

[55] M. Groeger, K. Arbter, and G. Hirzinger. *Medical Robotics*, chapter Motion tracking for minimally Invasive Robotic Surgery, pages 117–48. I-Tech Education and Publishing, 2008.

[56] I. Haritaoglu, D. Harwood, and L.S. Davis. Hydra: Multiple people detection and tracking using silhouettes. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 280–285. IEEE, 1999.

[57] Karol Hausman, Scott Niekum, Sarah Osentoski, and G Sukhatme. Active articulation model estimation through interactive perception. In *submitted to) IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[58] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[59] Geoffrey E. Hinton. Products of experts. In *ICANN*, pages 1–6, 1999.

[60] Inuitive-Surgicals-Inc. `http://www.intuitivesurgical.com`, 2015.

[61] R. Jackendoff. *Semantics and Cognition*. MIT Press, 1983.

[62] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, 2013.

[63] H. Jiang, S. Fels, and J.J. Little. Optimizing multiple object tracking and best view video synthesis. *IEEE Transactions on Multimedia*, 10(6):997–1012, 2008.

[64] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Attention, Perception, & Psychophysics*, 14(2):201–211, 1973.

[65] S.-K. Jun, M.S. Narayanan, A. Eddib, S. Garimella, P. Singhal, and V. Krovi. Robotic minimally invasive surgical skill assessment based on automated video-analysis motion studies. In *IEEE International Conference on Biomedical Robotics and Biomechatronics (BioROB)*, 2012.

[66] S.-K. Jun, M.S. Narayanan, A. Eddib, S. Garimella, P. Singhal, and V. Krovi. Evaluation of robotic minimally invasive surgical skills using motion studies. *Journal of Robotic Surgery*, 7(3):1–9, 2013.

[67] Seung-kook Jun, Suren Kumar, Xiaobo Zhou, Daniel K Ramsey, and Venkat N Krovi. Automation for individualization of kinect-based quantitative progressive exercise regimen. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 243–248, 2013.

[68] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *PAMI*, (99), 2011.

[69] Dov Katz and Oliver Brock. Extracting planar kinematic models using interactive perception. In *Unifying Perspectives in Computational and Robot Vision*, pages 11–23. Springer, 2008.

[70] Dov Katz, Moslem Kazemi, J. Andrew (Drew) Bagnell, and Anthony (Tony) Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *Proceedings of IEEE International Conference on Robotics and Automation*, May 2013.

[71] Ben Kehoe, Gregory Kahn, Jeffrey Mahler, Jonathan Kim, Alex Lee, Anna Lee, Keisuke Nakagawa, Sachin Patil, W Douglas Boyd, Pieter Abbeel, et al. Autonomous multilateral debridement with the raven surgical robot. In *ICRA*, 2014.

[72] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008.

[73] Ross A Knepper, Stefanie Tellex, Adrian Li, Nicholas Roy, and Daniela Rus. Single assembly robot in search of human partner: versatile grounded language generation. In *HRI*, number 1, pages 167–168. IEEE Press, 2013.

[74] EI Knudsen and MS Brainard. Creating a unified representation of visual and auditory space in the brain. *Annual review of neuroscience*, 18(1):19–43, 1995.

[75] Jonathan Ko and Dieter Fox. Learning gp-bayesfilters via gaussian process latent variable models. *Autonomous Robots*, 30(1):3–23, 2011.

[76] L. Kratz and K. Nishino. Tracking with local spatio-temporal motion patterns in extremely crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 693–700. IEEE, 2010.

[77] N. Krishnamoorthy, G. Malkarnenkar, R. J. Mooney, K. Saenko, and S. Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *AAAI*, 2013.

[78] Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond Mooney, Kate Saenko, and Sergio Guadarrama. Generating natural-language video descriptions using text-mined knowledge. *NAACL HLT 2013*, page 10, 2013.

[79] 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.

[80] Frank P Kuhl and Charles R Giardina. Elliptic fourier features of a closed contour. *Computer graphics and image processing*, 18(3):236–258, 1982.

[81] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[82] S. Kumar, M. S. Narayanan, S. Misra, S. Garimella, P. Singhal, J. J. Corso, and V. Krovi. Video-based framework for safer and smarter computer aided surgery. In *Hamlyn Symposium on Medical Robotics (HSMR)*. 2013.

[83] S. Kumar, M. S. Narayanan, P. Singhal, J. J. Corso, and V. Krovi. Surgical tool attributes from monocular video. In *ICRA*. 2014.

[84] Suren Kumar, Vikas Dhiman, and Jason J Corso. Learning compositional sparse models of bimodal percepts. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[85] Suren Kumar, Madusudanan Sathia Narayanan, Pankaj Singhal, Jason J Corso, and Venkat Krovi. Product of tracking experts for visual tracking of surgical tools. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 480–485. IEEE, 2013.

[86] Suren Kumar, Madusudanan Sathia Narayanan, Pankaj Singhal, Jason J Corso, and Venkat Krovi. Surgical tool attributes from monocular video. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4887–4892. IEEE, 2014.

[87] Suren Kumar, Pankaj Singhal, and Venkat Krovi. Computer-vision-based decision support in surgical robotics. *IEEE Design and Test*, 32(5):89–97, Oct 2015.

[88] Suren Kumar, Javad Sovizi, and Venkat N Krovi. Motion models and gaussian process regression based observation model for pose estimation. *In Preparation*, 2015.

[89] Suren Kumar, Javad Sovizi, Madusudanan Sathia Narayanan, and Venkat Krovi. Surgical tool pose estimation from monocular endoscopic videos. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 598–603. IEEE, 2015.

[90] Suren Kumar, Javad Sovizi, M.S. Narayanan, and Venkat Krovi. Surgical tool pose estimation from monocular endoscopic videos. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 598–603, May 2015.

[91] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Michael Ruhnke, Giorgio Grisetti, Cyrill Stachniss, and Alexander Kleiner. On measuring the accuracy of slam algorithms. *Autonomous Robots*, 27(4):387–407, 2009.

[92] C.H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, pages 1217–1224, 2011.

[93] N. Kyriazis and A. Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *CVPR*, 2013.

[94] Michael S Lewicki. Efficient coding of natural sounds. *Nature neuroscience*, 5(4):356–363, 2002.

[95] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014.

[96] B. Logan. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, 2000.

[97] B.D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 3, pages 674–679, 1981.

[98] David G Luenberger and Yinyu Ye. *Linear and nonlinear programming*, volume 116. Springer Science & Business Media, 2008.

[99] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.

[100] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel. Performance measures for information extraction. In *Proc. DARPA Broadcast News Workshop*, 1999.

[101] Roberto Martin Martin and Oliver Brock. Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2494–2501. IEEE, 2014.

[102] Cynthia Matuszek, Nicholas Fitzgerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A joint model of language and perception for grounded attribute learning. In *ICML*, 2012.

[103] N. Mavridis and D. Roy. Grounded situation models for robots: Where words and percepts meet. In *IROS*, 2006.

[104] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[105] Duy Nguyen-Tuong and Jan Peters. Local gaussian process regression for real-time model-based robot control. In *IROS*, pages 380–385. IEEE, 2008.

[106] S Obdržálek, Gregorij Kurillo, Jay Han, Ted Abresch, Ruzena Bajcsy, et al. Realtime human pose detection and tracking for tele-rehabilitation in virtual reality. *Studies in health technology and informatics*, 173:320–324, 2012.

[107] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[108] K. Okuma, A. Taleghani, N. Freitas, J.J. Little, and D.G. Lowe. A boosted particle filter: Multitarget detection and tracking. *ECCV*, pages 28–39, 2004.

[109] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[110] Seymour Papert. The summer vision project. 1966.

[111] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[112] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009.

[113] Zachary Pezzementi, Sandrine Voros, and Gregory D Hager. Articulated object tracking by rendering consistent appearance parts. In *ICRA*, pages 3940–3947. IEEE, 2009.

[114] Sudeep Pillai, Matthew Walter, and Seth Teller. Learning articulated motions from visual demonstration. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[115] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208. IEEE, 2011.

[116] J. Porway, B. Yao, and S. C. Zhu. Learning compositional models for object categories from small sample sets. *Object categorization: computer and human vision perspectives*, (1), 2008.

[117] Gustavo A Puerto-Souza and Gian-Luca Mariottini. Wide-baseline dense feature matching for endoscopic images. In *Image and Video Technology*, pages 48–59. Springer, 2014.

[118] Jan Puzicha, Joachim M Buhmann, Yossi Rubner, and Carlo Tomasi. Empirical evaluation of dissimilarity measures for color and texture. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1165–1172. IEEE, 1999.

[119] D. Ramanan, D.A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *PAMI*, 29(1):65–81, 2007.

[120] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, MA, 2006.

[121] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

[122] Austin Reiter, Peter K Allen, and Tao Zhao. Articulated surgical tool detection using virtually-rendered templates. In *Computer Assisted Radiology and Surgery (CARS)*, 2012.

[123] Austin Reiter, Peter K Allen, and Tao Zhao. Marker-less articulated surgical tool detection. In *Computer Assisted Radiology and Surgery*, 2012.

[124] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2423–2430. IEEE, 2011.

[125] Stephen Roller and Sabine Schulte im Walde. A multimodal lda model integrating textual, cognitive and visual modalities. In *EMNLP*, pages 1146–1157, Seattle, WA, October 2013.

[126] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, volume 7, pages 410–420, 2007.

[127] Louis B Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pages 76–82. IEEE, 1993.

[128] D. K Roy and A. P Pentland. Learning words from sights and sounds: A computational model. *Cognitive science*, 26(1):113–146, 2002.

[129] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014.

[130] Bart Rylander, Terry Soule, and James Foster. Computational complexity, genetic programming, and implications. In *Genetic Programming*, pages 348–360. Springer, 2001.

[131] Ayse Pinar Saygin, Ilyas Cicekli, and Varol Akman. Turing test: 50 years later. In *The Turing Test*, pages 23–78. Springer, 2003.

[132] Christian D Schunn and Dieter Wallach. Evaluating goodness-of-fit in comparison of models to data. *Psychologie der Kognition: Reden and vorträge anlässlich der emeritierung von Werner Tack*, pages 115–154, 2005.

[133] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *Automatic Control, IEEE Transactions on*, 49(9):1453–1464, 2004.

[134] Jürgen Sturm. Learning kinematic models of articulated objects. In *Approaches to Probabilistic Model Learning for Mobile Manipulation Robots*, pages 65–111. Springer, 2013.

[135] Jürgen Sturm, Kurt Konolige, Cyrill Stachniss, and Wolfram Burgard. 3d pose estimation, tracking and model learning of articulated objects from dense depth video using projected texture stereo. In *RGB-D: Advanced Reasoning with Depth Cameras Workshop, RSS*, 2010.

[136] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, pages 477–526, 2011.

[137] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. *Proc. AAAI*, (1), 2011.

[138] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[139] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, (1):267–288, 1996.

[140] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[141] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1653–1660. IEEE, 2014.

[142] Shimon Ullman. *The interpretation of visual motion.* MIT Press, 1979.

[143] Alberto Vaccarella, Elena De Momi, Andinet Enquobahrie, and Giancarlo Ferrigno. Unscented kalman filter based sensor fusion for robust optical and electromagnetic tracking in surgical navigation. *IEEE Transactions on Instrumentation and Measurement*, 62(7):2067–2081, 2013.

[144] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the International Conference on Multimedia*, pages 1469–1472. ACM, 2010.

[145] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014.

[146] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[147] P. Vogt. The physical symbol grounding problem. *Cognitive Systems Research*, 3(3):429–457, 2002.

[148] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. Hog-gles: Visualizing object detection features. In *ICCV*, 2013.

[149] S. Voros and G.D. Hager. Towards real-time tool-tissue interaction detection in robotically assisted laparoscopy. In *2nd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics, 2008*, pages 562–7, 2008.

[150] 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.

[151] Chieh-Chih Wang, Charles Thorpe, and Sebastian Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 842–849. IEEE, 2003.

[152] Jack Wang, Aaron Hertzmann, and David M Blei. Gaussian process dynamical models. In *Advances in neural information processing systems*, pages 1441–1448, 2005.

[153] Shenlong Wang, Lei Zhang, Yan Liang, and Quan Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *CVPR*, pages 2216–2223. IEEE, 2012.

[154] G.-Q. Wei, K. Arbter, and G. Hirzinger. Automatic tracking of laparoscopic instruments by color coding. In *CVRMed-MRCAS'97*, volume 1205 of *Lecture Notes in Computer Science*, pages 357–66. Springer Berlin Heidelberg, 1997.

[155] G. Welch and G. Bishop. An introduction to the kalman filter. *Design*, 7(1):1–16, 2001.

[156] Greg Welch and Gary Bishop. An introduction to the kalman filter, 1995.

[157] Christopher Wottawa, Richard E Fan, Catherine E Lewis, Brett Jordan, Martin O Culjat, Warren S Grundfest, and Erik P Dutson. Laparoscopic grasper with an integrated tactile feedback system. In *ICME International Conference on Complex Medical Engineering*, pages 1–5. IEEE, 2009.

[158] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *IJCV*, 75(2):247–266, 2007.

[159] Shuang Wu, Hongjing Lu, and Alan L Yuille. Model selection and velocity estimation using novel priors for motion patterns. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1793–1800. Curran Associates, Inc., 2009.

[160] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418. IEEE, 2013.

[161] F. Xiong, O. Camps, and M. Sznaier. Dynamic context for tracking behind occlusions. *ECCV*, pages 580–593, 2012.

[162] Ran Xu, Priyanshu Agarwal, Suren Kumar, Venkat N Krovi, and Jason J Corso. Combining skeletal pose with local motion for human activity recognition. In *Articulated Motion and Deformable Objects*, pages 114–123. Springer, 2012.

[163] Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Proceedings of AAAI Conference on Artificial Intelligence*, volume 6, 2015.

[164] Bar-Shalom Yaakov, XR Li, and Kirubarajan Thiagalingam. Estimation with applications to tracking and navigation. *New York: Johh Wiley and Sons*, 245, 2001.

[165] Jingyu Yan and M. Pollefeys. Automatic kinematic chain building from feature trajectories of articulated objects. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 712–719, June 2006.

[166] J. Yang, PA Vela, Z. Shi, and J. Teizer. Probabilistic multiple people tracking through complex situations. In *11th IEEE International Workshop on PETS*, 2009.

[167] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *Image Processing, IEEE Transactions on*, 19(11):2861–2873, 2010.

[168] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, 2011.

[169] Angela Yao, Juergen Gall, Gabriele Fanelli, and Luc J Van Gool. Does human action recognition benefit from pose estimation?. In *BMVC*, volume 3, page 6, 2011.

[170] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 38(4):13, 2006.

[171] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[172] H. Yu and J. M. Siskind. Grounded language learning from videos described with sentences. In *ACL*, 2013.