Interface Issues in Robot Scrub Nurse Design

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Amer Agovic

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Prof. Nikolaos Papanikolopoulos, Prof. Ahmed Tewfik

December, 2011

Acknowledgements

On a personal level, the most important accomplishment would have to be the realization that reaching the full potential can only happen with the help of others. I spent much of the time in graduate school in a state of emotional conflict. For the longest time I tried to avoid working for anything or anyone but myself. In hindsight, I realize that painful events outside my control shaped this attitude which makes things only worse. It is within this context that I came to appreciate the professors and students mentioned here. Only with their help was I able to come this far.

First and foremost I would like to thank my advisors Prof. Nikolaos Papanikolopoulos and Prof. Ahmed Tewfik. They both demonstrated great deal of patience and allowed me to make up my mind even if it took a little longer. Prof. Tewfik was there during the time when I was still looking for a problem. In one session we came across medical robotics and that changed everything. Prof. Nikos, later, provided the environment to further grow. Even after two years of fruitless publication attempts he had faith in me until one day I finally published my first work. One of the most exciting points came during our application for RSN funding. The manner in which they approached the problem formulation and the presentation showed me that I was in the company of truly great people. From that moment on I tried to imitate what I saw and it has been the most rewarding experience of all.

Prof. Samuel Levine and Prof. James Holte deserve a lot of the credit for my selection of the robot scrub nurse (RSN) as a problem area. I am truly grateful to know such inspiring individuals. As my teaching supervisor Prof. Holte took time to find out what I was doing and together we had numerous brainstorming sessions. His way of looking at problems epitomizes the creative and out-of-box thinking. Tapping on his experience I would eventually meet with Prof. Levine and learn about problems in the

operating room. It is no exaggeration when I say that Dr. Levine has transformed my view of research and my place in it. I had the whimsical tendency to go off tangent but after meetings with Prof. Levine I would get a sense on how to focus and organize intellectual effort. I am convinced that without his help and advice none my work would be noteworthy.

Other professors which enabled and supported me over the years include Prof. Peter Olver as a committee member, Prof. Maria Gini as my UROP supervisor, Prof. Jaijeet Roychowdhury as my honors advisor. In addition, I learned a great deal from my teaching supervisors Prof. Robbins, Prof. Wang and Prof. Higman. For their support in researching scrub nurse robots I am very grateful to the people at the Institute for Engineering in Medicine. Their foresight and support allowed me to purse a truly novel research.

Besides the professors a number of students provided essential help and feedback. My brother Amrudin Agovic took time from his own research to proof read mine. He had me revise my first successful conference paper for six hours straight. It did not sit well with me then but I sure appreciate it now. Alex Kossett helped me design the RSN gripper. If I was arrogant enough to wonder why you would need mechanical engineers by the time Alex designed the gripper that notion morphed into humility and appreciation for delegating work. Other students who provided valuable experience and assistance include Robert Martin and Joe Levine. Thank you guys.

Dedication

This achievement would be meaningless without mentioning those people whose life served as my compass. I dedicate this work to my entire family and especially to my parents.They faced adversity and smiled at it. They had less education but more wisdom than I have. Above everything they were decent human beings. I would like to especially mention my uncle Tosum. My fascination with electricity started when he suggested a career as a refrigerator repairman ;-)

Abstract

The idea of a robot providing assistance in the medical field is very promising. Due to the myriad of unresolved issues a full therapist robot is still out of our reach. In this study we approach the problem in an application area that is sufficiently constrained to allow us a certain degree of practical success. Concretely we present a system description of a robotic scrub nurse (RSN) for microsurgery. We identify robot interfacing as a crucial problem to be solved. In that regard we examine how spoken language could be utilized, how vision could guide robot motion, and finally we examined human-robot interaction at the haptic level. For the haptic interface we have designed and evaluated a shape conforming grasp mechanism. Such an approach is uncommon in the robotic industry where per-task adapter-style exchangeable grasping mechanisms are mounted on the robot for each run. While industrial robots deal with few instruments and must lift heavy weights, our problem domain contains hundreds of instruments and prohibits any type of crushing or cutting movements. Our results provide an integration study of the various components for assistant robots. Furthermore we contribute novel insights in the design of grasp planning methods, visual tracking methods and natural language dialogue systems.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The field of medical robotics can be split into two segments depending on the interaction between operator and machine. The first segment includes robots that enable the operator. These robots stand between the operator and the task. One very prominent example is the DaVinci robot [1]. The other segment includes robots that assist the operator. Such assistant robots adhere to the philosophy that the operator knows best. The assumption here is that the operator performs the primary task almost perfectly. In that case introducing an intermediary would only break the flow. One example would be the hand-eye coordination surgeons develop. By researching human-robot interface (HRI) problems for Robot Scrub Nurse (RSN) our work can be placed in the segment of assistant robots. Interfacing for enabling robots simplifies to user interface (UI) design. The interfacing problem for assistant robots is more challenging. Because the robot interacts directly with human operators the workspace is highly unstructured. A robot assistant has to demonstrate that it is safe for humans and that it reliably replicates desired activity. Using industrial grade technology and methods is usually not a good option. Industrial settings are more controlled environments. Generally speaking our work attempts to relax the constraints on the environment needed for robot deployment. Ultimately the environment should allow interacting robot applications in human workspaces.

Our study of interfacing issues is grounded in a real world application. From that perspective this research tends to emphasize empirical and engineering methodologies. Our application of choice is an assistant robot which would serve in the operating

room and automate responsibilities currently assumed by a scrub nurse technicians. Our effort to design and develop a RSN system provides us with an abundant pool of problems. Above all other problems we have identified the human robot interface as the most important. Therefore we give an overview of a Robot Scrub Nurse system but then dedicate greater detail to three interface modalities in particular. They are haptic interface or grasp planning, visual interfacing to guide the robot and spoken command interface to issue commands and query state of the robot. The overall problem setting is best illustrated on hand of the illustration in Figure 1.1.

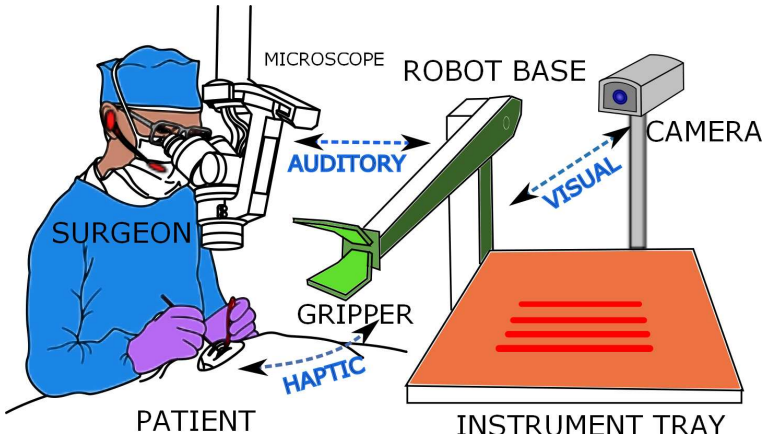

Figure 1.1: Problem setting and interfaces

What makes a robotic scrub nurse an attractive application is that we can still make certain simplifying assumptions (Figure 1.1). The ability to constrain the problem is essential. The activities a scrub nurse performs are repetitive, costly, and prone to errors. In microsurgery for Otolaryngology, which is the operating procedure that we have examined, the surgeon makes use of several hundred delicate and costly instruments. The surgeon using an operating microscope does not want to look away to get an instrument. It takes a surgeon time to get his/her mind oriented to the surgical field. Looking away delays the surgeon, allows fluids to fill the area of surgical interest and sometimes requires moving tissue to get an appropriate look at the internal structures. The surgeon-nurse interaction begins with the surgeon requesting an instrument by spoken word. The nurse uses his/her experience and training to identify it, picks it up without dropping it, moves it without hitting obstacles and places it on the stretched-out

palm of the surgeon by means of visual and haptic (touch) feedback. The nurse makes sure that the instrument does not cut the physician and does not violate sterility. This movement must occur in a timely manner. Sources of delay could come from difficulty in identifying the correct instrument, dropping it, or losing sterility by other means. A good experienced nurse will handle the instruments with care and will be intimately familiar with the procedure so that she can pro-actively deliver the next instrument, even before it is requested. The scrub nurse as the manager of instruments also makes sure that no instruments are left inside the patient after the end of the procedure. Due to the repetitive nature of the job, scrub nurses usually rotate every 30-60 minutes. During the change-over the new nurse must be quickly briefed and the surgeon needs to wait. In general scrub nurses have developed handbooks on how to behave to minimize risks such as forgetting instruments or violating sterility.

Clearly a robot must replicate all of the above steps. In fact, there are a number of issues which characterize the overall field:

- A scrub nurse is expensive: The average compensation per year is $44,000 [2].
- There is a shortage: In 2008 a combined 91,500 were employed in US hospitals across all specialties [2].
- Monotonous: Memorizing hundreds of almost-identical instruments is not very motivating but forgetting one inside the patient is catastrophic.
- Procedural complexity: Regulative guidelines require that a scrub nurse be rotated periodically and per case more than one nurse is needed (per shift, per specialty, per hospital).
- Point of delay: Dropping an instrument leads to immediate delays and overall cost increase.
- Point of delay: A nurse will delay an operation if unfamiliar with instruments (sheer number), procedure (specialty type), or present situation (after rotation).

Given the set of aforementioned issues we note that a small hospital which does not staff multiple scrub nurses is unable to perform an operation such as ear surgery. On the other hand it is estimated that there are about 1 million PE tubes inserted per

year in the US [3], so the demand for ear surgery is clearly increasing. A robot could potentially alleviate all of these issues by being easier to maintain, quicker to update with procedural changes, faster at responding to a surgeon, more accurate at managing instruments, and allowing more efficient and fulfilling use of human resources.

Within this application setting we show how different technologies are integrated into a whole system. We present our contributions in the research of haptics or grasp planning, visual servoing and natural language processing. While all of these fields are more or less mature none of them has been considered carefully for assistant robots like the RSN. To research haptic interfacing we designed and studied a novel shape conforming grasp mechanism. Utilizing an inflatable membrane our gripper is able to generate a soft grasp which is essential for human interaction. Handing of instruments to surgeons and taking them back involves domain specific knowledge of what could be called haptic communication protocols. The exchange of instruments must be repeatable, safe for humans and safe for the material. Besides using touch to actuate instrument exchange we looked at the problem of localization. Localizing the instruments on the instrument tray is not as challenging as locating the hand of the surgeon. Complicating matter is the possibility of obstructions such as microscopes used by the surgeons and various hoses attached to instruments. Of all the sensing modalities the computer vision offers the most versatility. However computer vision is still not robust enough for arbitrary tasks. In response we studied how the environment of the task can be engineered. We looked at various geometric tag patterns and their fidelity in being tracked accurately and robustly. Finally we examine the spoken command interface. In order for the robot assistant to be easily accepted into an environment which is well established and rigid procedurally the introducing technology must be as unobtrusive as possible. A speech recognition and synthesis system that understands natural human language would be optimal in that regard. Our work adapted an existing solution and measured its effectiveness in speech recognition tasks. To that end we made use of context free grammars to constrain syntax and improve accuracy. As the final step of the spoken command problem a dialogue management system is presented.

In the rest of this document we first survey the existing literature. After that we describe the application area for which the RSN is needed and present an overview of various RSN components/problems. Next the interfacing modalities are explored

further. Within the haptics section we go into more detail in two areas. First we present our gripper design and high level haptic signaling. Then we cover more abstract problem of grasp planning. These results are followed by our work on vision and speech. The final chapter summarizes and discusses our experience in designing, developing and deploying the system.

Chapter 2

Background

Our review of existing work is split in several parts. First, a short review of robotics in surgery is given. A lot of available work concerns enabling robots and discusses their performance in facilitating surgical procedures. We place emphasis on robot assistants and review work in that direction.

Next section explores factors affecting adoption of surgical robots. It is helpful to understand what drives and restrains research in medical robotics. Understanding why robots are so expensive, why only a few companies dominate or how hospitals purchase medical robots should affect our research into robot interfaces. In that spirit we examine economic and social conditions which determine the acceptance rate of this new technology.

In the remaining three section we cover more technical work that supports our research on interfaces. We examine research in natural language processing with an emphasis on dialog management. After that we cover work on grasp planning and finally methods needed for computer vision.

2.1 Robots in Surgery

The field of surgical robotics is divided into two groups. The first type of robots consists of devices which stand directly between the surgeon and the patient. For example in [1] the robot DaVinci has two components. A console, on which an operator sits, and a possibly remotely controlled instrument stand which is working on the patient. Such

systems make tele-operation possible and can amplify the ability of the operator(i.e., tremor reduction).

The drawback of such systems and sometimes the reason for their rejection by the surgeons is that they break hand-eye coordination. The other type of robot, in which category the RSN falls, is called an assistive robot because it stands on the side and provides aid. Penelope is the first commercial scrub nurse robot that was recently developed [4]. This robot and the accompanying patent address almost all features that any similar robot needs to have, such as a spoken language command interface, path planning to deliver instruments, etc. One aspect that was not considered in this work is the grasping mechanism. Instead the authors use a simplistic gripper which only allows deployment in surgeries where the surgeon can actively look for the instrument and grasp it.

Development of a trauma care assistant is detailed in [5]. The aim was to develop a scrub robot to assist a tele-operating robot like DaVinci in environments where highly skilled support staff is unavailable. The most prominent challenge is the interface between the nurse robot, the instruments, and the target robot. Unlike our approach the route in which they attempt to solve the problem is more mainstream in that they try to expand on methods already being used in industry. The end result is a flexible instrument adapter which is amenable to calibrations and allows for compensation of errors.

In [6] an example of task knowledge modeling is presented. The authors, while working towards a scrub nurse robot, have correctly identified that a prerequisite is the ability to observe and model the behavior of human scrub nurses. To that end a timed automata model is proposed which is trained in unsupervised fashion. This work is further expanded in [7] where the timed automata is then used to analyze recorded patterns and to draw some specific recommendations about the ultimate design needs. An important aspect that this work raises is that of human adaptivity. This concept has to be incorporated into the robot design and is a focal point for our work.

Another example of the task modeling problem is the work in [8]. Here the authors look at a well behaved type of surgery (Laparoscopy) and devise a system which allows for logging of instrument usage. The analysis of such data is not only used to model scrub nurse behavior patterns but also to analyze and understand their nature more clearly.

This work raises the interesting issue of record keeping which would be necessary as part of a feedback system of any scrub nurse replacement.

When we think of assistive robots an alternative viewpoint would be the direction taken by the authors in [9]. The aim of the authors has been to create robots that care or give comfort. The idea of a robot acting as a therapist is powerful but we believe that before it becomes reality interface related issues, such as those examined here, will need to be addressed.

2.2 Acceptance of Robots in Surgery

The proliferation of robots in surgery is not a straightforward matter. Clearly robots eliminate need for human labor and that is bound to face resistance. Furthermore, robots are expensive and there are only a few dominant manufacturers. It is useful to understand these circumstances. Our look at factors affecting robot acceptance was prompted by two developments during the research. During our first visit to the operating room we failed to mention to the staff the purpose of our visit, which was to observe them for purposes of automating away some of their responsibilities. Naturally on our second visit and after they learned more about it their attitudes changed. At another time closer to the end of the research the office of technology commercialization (OTC) and a local robotics company expressed interest in our work. However shortly thereafter they declined to pursue the idea further. Why this happened has motivated our review in this section.

2.2.1 Economic Conditions

According to analysis by Frost & Sullivan [10] the market in surgical robots is divided into three tiers. Tier I is dominated by one or few major players. For example on RSN side Robotic Systems & Technologies Inc. holds a very broad patent. On the enabling robots side Intuitive Surgical Inc. is the dominant player. Tier II is formed by developers of specialized tools catering to products marketed by Tier I players. Finally Tier III holds external and existing robot companies that are trying to diversify into medical devices. The market as a whole is showing increasing revenues while capital investment is going down. It indicates that the market is not saturated but entry by

new companies is made difficult. Possible product categories include sale of systems, system accessories and servicing.

As reported by the same analysts in [10, 11] and [12] major market obstacles include:

- High cost to market
- Regulatory and Political Factor
- Standards and Procedures Regulations
- Funding support from government
- Resistance from traditionalist surgeons

Private investment is harder to obtain because it requires large upfront capital without immediate results. In such case funding by government agencies would be beneficial but it varies from country to country with Europe placing more importance on this market segment. In [13], challenges of rehabilitation robots is discussed. The authors postulate that the field is considered orphaned because high capital investment inhibits faster development. As a way to address cost the authors suggest a collaborative framework model that would utilize the internet as a cost saving measure. On top of these issues the market introduces novel risks/issues such as equal access to care or robotic failure.

The technological drivers and benefits which provide incentive to manufacture and adopt this new technology include[10, 11, 12]:

- Better image visualization
- Shorter surgical time
- Enhanced surgery results
- Improved surgeon ergonomics
- New pathways like telesurgery
- Less pain
- Fewer complications

- Cost reduction
- Better cosmetic results
- Improved accuracy and efficiency
- Workforce efficiency (fewer staff needed)

The competitive factors of products determine their penetration of the market. Following attributes were identified as significant[10]:

- Patient safety
- Access to product
- Quality of material
- Ease of use
- Cost
- Uniqueness
- Strong customer support
- Familiar software usage

The market intelligence presented here played a role in our case. Both the OTC and the local company declined to pursue the idea any further because they could not identify a way to distinguish themselves from Tier I players and remain profitable.

2.2.2 Social Conditions

Our experience with the nursing staff was interesting enough to make us wonder about the social implications of a robot assistant. We deemed it important to investigate which factors would make the RSN more acceptable and looked at previous research to gain a better perspective.

Before robots in medicine there was time when robots in industry were not as common. Work in [14] is a bit older but might reveal how robots became so ubiquitous

in industry. The favorable adoption factors listed for industrial robots include: robot champion, experience with automation, existing automation, bad working conditions, careful assessment of economic and alternative performance. A robot champion is a successful first implementation that shows promise. Interestingly this study finds that labor response was not as bad or crucial as expected. At the same time the authors claim that robots do not always remove monotonous tasks but might introduce other repetitive actions. The robotization affects the structure of work from people-centered to machine-centered. Finally an impediment can be the lack of trained staff to work with a robot. Some problems with robotization include: difficulty to run the system, selection of robot model, long development periods, increased managerial effort, organizational resistance, need for extensive training, retaining trained personnel.

The effect of culture on the adoption process is examined in [15]. The study compares differences between United States and Japan and try to identify attributes which affect the adoption process. Some of the cultural factors that are influential include individuality, risk avoidance, social status, time perspective, social competition. Each of these factors affect the adoption rate differently between cultures. The attributes appear in clusters within a culture.

Frequently technical and monetary factors in adoption of robots is considered. In [16] the authors look at social psychology and try to derive models that could explain the adoption rate. By applying insights from psychology on how humans perceive technology and robot interaction it might be possible to design more acceptable interfaces and robots even for household use. They derive some guidelines that address how technology is perceived and how this perceptions are formed. The factors that affect acceptance include safety, accessibility, practical benefit, fun, social pressure, status gains, expectation of social intelligence, experience, media, social network. All of these factors should be considered in interface and robot design and be mindful of perception.

From a more technical perspective it is interesting to note research that reports on the use of robots in surgery.

An early look at robotics in surgery is given by [17]. The authors raise key issues that are still relevant today. They categorize the interaction modes between surgeon and robot along the level of cooperation, from fully autonomous robot operation to fully guided. The level of interaction is determined by the active constraints set on the robot.

Another interaction mode mention is that of teleoperation and telemonitoring. In their treatment of technical issues they also discuss problems in clinical implementation and acceptance. They see the safety as the most critical issue. Beyond safety other issues mentioned are cost and outcome.

In [18] the authors examine how novel technology affects communication and the performance in the operating room. The impact of the study sheds new light on how effective technology enhances performance and how well it is accepted by users. Introduction of technology changes the responsibilities each role assumes. It appears that the scripted communication pattern performs better than either automated or no-rule patterns. The success of communication in great deal depends on the mental state shared between the nurse and surgeon. One possible explanation is that homogeneous communication allows participants to detect anomalies in speech better. However while technology might offer performance improvements it frequently increases the number of steps. In this study the task decomposition shows that the complexity significantly increases and also that the number of modalities involved increases to compensate for limits imposed by new technology.

In [19] a method for evaluation of robot safety is presented. This work is useful in that it models risks and hazards of a robot in its interaction with the environment. Following the principles laid out in their hazard identification and safety insurance control policy one obtains a plan not only to avoid adverse consequences arising from an issue but also for debugging and ensuring quality of service. Something we might revisit to evaluate our system.

Work in [20, 21] examines recent experience with robots in laparoscopic procedures. In about 10% of cases robotic procedure had to be converted to plain one because they could not be finished with robots. In one case robot failed in others minor bleeding could not be stopped. Learning curve for robotic procedures was 10 sessions or more. Main drawback seems to be inflexible instruments and lack of arms. The benefit to patient is not yet fully evaluated. Besides long learning curve some of the pitfalls are unstable camera, limited motion, two dimensional imaging, poor ergonomics.

2.3 Natural Language Processing

One of the early works on comprehensive natural language processing systems can be found in [22]. Here a dialog system is described which is used to direct trains. It is based on available speech recognition software and addresses problems in post-processing of recognized speech, parsing, discourse management, and verbal reasoning. Another interesting study of dialog managers is presented in [23]. The system is called "CommandTalk" and the application area in this case is a front end to a battle-field simulator. Of particular interest is their use of finite state machines (FSMs) to model discourse.

Our work is more similar to recent works such as [24, 25]. In both of these cases the environment is a robot assistant in the household. Within these settings an attempt is made to fuse various interface modalities into one decision process. The goal of fusing multiple modalities is to help in disambiguating the state and user intent. As in our case the dialog manager is based on a finite state machine (FSM). In contrast we only considered speech even though visual and haptic events are available to the RSN.

An alternative approach to dialog managers can be found in [26]. In this study the dialog manager is based on Partially Observed Markov Decision Processes. It shows how such structure can handle noise and ambiguous cases and how it behaves as conditions degrade. It is motivated by the fact that the intention of the user is only partially observable, so it models the intention of the user rather than the state of the machine. Other interesting dialog managers can be found in [27, 28, 29] and [30].

As part of the high-level dialog management we need to look at the parsing of recognized speech and modeling of task related knowledge. Our setting asks for an integration of these elements and reasoning under uncertainty. We find some attempts that use statistical methods to couple various levels of the recognition process in [31, 32]. In the first paper use of context free grammars is advocated and a lattice based parsing method is presented. The second paper describes an entire robot programming framework. It also makes use of lattice processing but the framework (including low level recognition) is based on the Dynamic Bayesian networks (DBNs). A good resource for a solid treatment of statistical based language processing can be found in [33]. It presents methods for the probabilistic handling of parsing, semantic analysis and discourse.

On the parsing side one big issue to address is management or speech repairs.

Namely, how should the parser behave if the speaker switches sentences in mid-air. In this regard work in [34] and [35] discusses strategies to detect when such events occur. Frequently the switch is made to better articulate the same contextual meaning. However we cannot rule out the possibility that the context needs to be switched. With context we understand here a description of the robot's state in the environment. As shown in [36, 37] we can effectively use the context to improve the accuracy and robustness of the discourse manager. For example if the robot is moving from the instrument tray towards the surgeon some sentences will not make sense and that should be encoded in some way.

Very interesting work is presented in [38]. Here the authors present a system for the management of hazards in an aircraft. It uses Bayesian networks to reason about the environment and sort, schedule, and issue advisories and alerts to the pilot. It is interesting because some aspects of the chaotic nature in an aircraft resemble that of an operating room. Two aspects in this work translate to our work. First is reasoning about the state of the environment under uncertainty. The other is the management of hazards. The interaction of the robot nurse with the surgeon must be primarily seen as a hazard minimization process.

Available literature covering all of these topics is extensive. Our goal here was to combine some of these ideas into a real-time system. Clearly our problem setting alters the importance of various functions with hazard minimization being at the top.

2.4 Grasp Planning

For an overview of grasp planning we refer the reader to [39] and [40]. While grasp planning overlaps with other fields such as haptics one can define it as the search for the best way to grasp an object. This problem is complicated by the geometry of the object, modeling of the grasp contact, number of contacts, nature of external forces, and the parameterization of the gripper. The earliest attempts were based on the idea of "form closure" which tries to determine when an object becomes immovable. Form closure is mostly determined by the geometry but it requires more contacts than are absolutely necessary. If one considers friction as part of the contact model we can find grasps with fewer contacts. Such grasps are based on force closure.

Work done in [41] and [42] is arguably the basis for many efforts in this area. It starts by considering point contacts with friction. The friction cone is approximated by a pyramid. The overal impact of external wrenches or forces is combined into a convex hull which contains the origin if we have achieved force closure. The distance of the origin to the border of such a convex hull is usually used as the quality metric. Another example of analytic grasp synthesis can be found in [43]. The authors formulate a quality metric based on the Q-distance which is differentiable and therefore allows simple gradient descent to be used in the search process. An attempt to overcome non-uniformity of the geometry is presented in [44]. Here a grasp quality measure is derived which approximates the grasp wrench space via spherical shapes that account for the worst-case disturbances. Another approach to grasp quality functions is shown in [45]. However the assumption here is that the geometry is smooth and numerically well behaved. For a more realistic contact model soft contacts have to be considered. The initial attempt at soft contacts addressed the issue of sliding [46]. Soft contact modeling demands the application of the more elaborate Hertzian analytic model [39, 47]. According to this model moments and forces at a contact are coupled. The friction cone of the Coulomb model is replaced with a friction limit surface. This appears to be the primary reason why soft contact modeling has not found wider usage. More recently in [48] an approach was presented which treats limit surfaces in a similar fashion as friction cones in [41]. The limit surface is approximated by a convex polyhedron.

Even if the contacts are understood and a quality metric is formulated, there still remains the issue of gripper parameterization. The problem is one of correspondence and the question is how we position, orient and articulate the gripper in order to maximize the grasp quality. One interesting approach has been proposed in [49, 50]. The idea is to find the gripper parameters if one knows the contact points on the target which yield a stable grasp. Both sources settle on an optimization scheme to find the best parameterization. The dilemma here is whether contacts generate a gripper configuration or if it is the other way around. As a purely analytic method, the need for optimization means that real-time application is limited. In this paper we assume that the gripper configuration determines contacts.

The impact of geometry is considered by the authors in [51]. The shape is assumed concave and convex decomposition is used to divide and conquer the problem. Along

similar lines use of shape primitives has been proposed [52]. In fact the authors pursue a more heuristic methodology. Besides using shape primitives for geometry representation the search for grasps occurs over a predetermined set of gripper preshapes. The grasp quality evaluation happens on-line by means of a grasping simulator. Building on the heuristic method of using simple geometric preshapes the authors in [53] propose a more generic means of object representation. They use hierarchical decomposition trees in terms of quadrics. One disadvantage of the approach is the tree decomposition which is a clustering problem coupled with the actual fitting of superquadrics. Quadrics were chosen over other decompositions because they encode normals naturally. Similarly in [54] authors present work on a prosthetics system to enable human teleoperation of robotic grasping. The system achieves real-time performance by reducing the dimensionality of the gripper DOF space. While this simplifies the search, it yields suboptimal results which requires an on-line validation step to fine-tune the grasp. Instead of looking for the optimal grasp analytically, the aim of the authors was to reduce the search space while still preserving most of the good grasps.

Having considered geometry, contact modeling, grasp quality and gripper parameterization the remaining problem is reachability. The search for a good grasp will inevitably yield multiple possible grasps [39]. In [55] the authors examine a scoring method as a way to rank the identified grasps. In particular this approach looks at how nearby clutter might affect a grasp planning algorithm. The method is based on force closure.

Alignment of $3D$ objects is a research field in its own right. The field is mainly driven by media content retrieval problems and registration of medical data [56]. The methods can be roughly split into optimization-based approaches where object geometry is directly used [57], and methods which use a variety of local and global features usually invariant to certain transformations. An interesting idea is the use of spherical harmonics [58]. The problem with the majority of these methods is that they try to solve the alignment problem between whole objects. Partial matching is evidently more difficult.

Shape descriptors have found wide usage in computer vision where they have lead to decent results in classification [59]. An early example of shape driven grasp planning is presented in [60]. Here we observe the use of antipodal grasps. The underlying basis

is found in force closure and as such this can be considered part of the more general approaches complicated by visual sensing.

2.5 Visual Servoing

The problem of visual tracking which is at the core of this paper is quite mature. We have made every effort to review previous work and to make use of it as much as we could. In the following paragraphs we only mention a limited number of sources which played an instrumental role in our effort. Among the integration studies work in [61] is similar to ours as it describes a vision driven grasp planning robot. Further, we only give sparse mention regarding the choice of features used. Two popular feature types that exhibit some invariance are SURF and SIFT [62, 63]. Other invariant features exist but we have not included them [64, 65] or we did not have much success in using them [66, 67, 68]. Finally, we should note that all of the model fitting methods described for depth recovery or point alignment can be embedded inside an optimization routine such as RANSAC [67, 69] to combat outliers.

2.5.1 Depth Recovery

Given correspondence, recovering depth is possible even for a single triangle. The simplest problem of depth recovery for triangles has been known since 1841 [70]. Usually, the solution to a three point depth recovery yields a fourth order polynomial and hence four possible solutions. In [71] the authors use the same idea. It is derived from photogrammetry and is based on the laws of cosines. The end-result is an eight-order polynomial which can be considered as fourth order if constraints are taken into account. For better insight into the theoretical foundation of this result, please see [72]. The ambiguity arising in the three point case is modest because often only two roots are real. Furthermore, the area of the triangle can be used as an additional constraint to obtain a unique result. For the more general cases of four or more points in correspondence it is possible to recover depth uniquely. Notable examples can be found in [73, 74]. The work in [74] describes the popular POSIT algorithm and is iterative in nature. More recent work that performs complete depth recovery and alignment is presented in [75]. It is not iterative, requires at least four points and runs in time O(n).

2.5.2 Alignment of 3D Points

After depth recovery alignment of 3D point clouds is used to recover the pose of the model. A very concise review of four major methods is presented in [76]. Two basic methods we decided to use are [77, 78]. The method in [77] is based on singular value decomposition. The correlation matrix is decomposed into eigenvectors which are then used to recover rotation. An eigenvalue approach is taken by [78]. Here the rotation is obtained by recovering the eigenvector of the largest eigenvalue of a quaternion matrix. In [79] we used an alternative method based on cross products to determine the rotation matrix. Our method requires only two correspondences and does not depend on complex matrix decomposition. In all cases the rotation matrix is determined and then used to recover translation.

2.5.3 Selection of Good Features and Clustering

In computer vision the term feature selection appears to be ambiguous. In this paper we have picked the feature type. We do not attempt to construct a feature. Rather, with feature selection we mean attributing to a feature a quality to perform as expected. We want to know if a feature is amenable for correspondence matching. Earlier work on this subject can be found in [80, 81, 82]. In [80] most of the effort is spent on enforcing temporal and spatial coherence. Still determination when a feature is lost is based on how far it wanders from an initial position. Important for this task is also how we measure similarity between features. As shown in [83] the estimation of feature distributions and its modeling can have a significant impact on the matching results. Some efforts [84] have bundled this task together with feature extraction in a classifier framework. For correspondence matching a classifier or quantization framework appears to be a natural setting. The features need to be either rejected or accepted and if accepted then assigned to one of a limited number of code words. Other interesting routines that look at correspondence are based on nearest neighbor classification and can be found in [85]. For an extensive review of clustering methods the reader is referred to [86, 87]. Among the various clustering methods one elegant approach is based on mixture models [88]. For correspondence, however, we need unambiguous cluster assignments. For clear cut cluster formation the agglomerative and graph theoretic clustering methods

appear a better fit. An optimization idea using graph matching is proposed in [89].

2.5.4 Structure from Motion

Arguably one of the greatest contributions in computer vision from the 90s involves methods concerning structure from motion. In [67] an entire chapter is dedicated to the various aspects of this problem. A thorough treatment of the subject and a toolbox is presented in [90]. The appeal of this method is its completeness and the treatment of missing points. It is based on matrix factorization. Another useful toolbox is provided in [91]. Besides implementing more classic algorithms like the Tomasi-Kanade method [91], a more recent method from [92] is also available. One issue with any of these methods is how they deal with missing points. The expectation is that a fixed number of points is available for a set of frames. A realistic problem is when a fixed number of points is available for every frame but their composition is not constant but interlocking over the set of frames.

Chapter 3

Overview

The goal of our project is to develop a robot capable of automating scrub nurse responsibilities. A scrub nurse performs a number of tasks but for our purposes we limited the scope to the task of instrument management and delivery to the primary surgeon. The overall diagram (Figure 3.1) of our proposed system follows similar proposals elsewhere [4]. Some components such as a robot platform, visual servoing, motion planning, speech recognition are necessary requirements.

Our first step was always to consider off-the-shelf solutions. In cases where we use existing work we explain how it was integrated and refer the reader to proper sources for more detail. We frame the role of each interface but then spend most of the time on the haptic interface. For the haptic interface we designed a novel shape-conforming gripper. Our gripper is simple in its mechanical design and uses inflatable membranes as a way to generate shape conforming grasp surfaces. Such an approach to grasping mechanisms is uncommon in mainstream industry. A more natural approach is to design specialized grippers for each task. Since we have to manage potentially hundreds of instruments that could cause problems, the grasping mechanism needs to be accommodating. In this regard we present novel work. As will be shown such an adaptive and generic gripper allows for the design of locally autonomous responses to ensure that the robot does not crush a human hand or the instrument. The inflatable chamber allows for sensing when the instrument has been removed from the gripper.

Before we go into the details of the essential components, we first provide a description of the problem. In particular we describe the environment and the instruments

used.

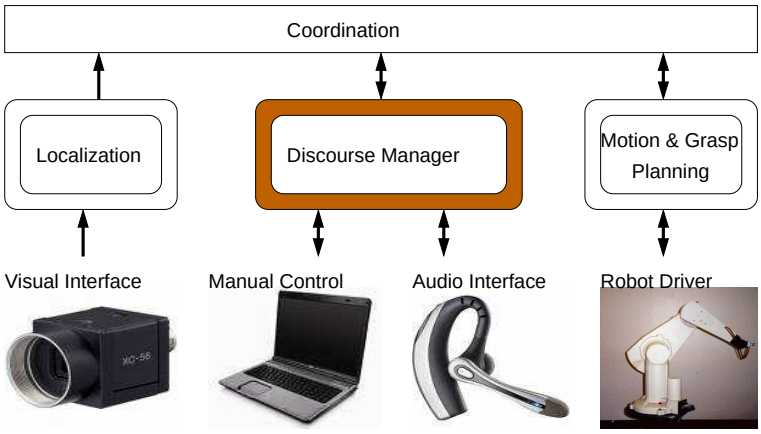

Figure 3.1: Main system components

3.1 The Environment: Microsurgery in Otolaryngology

Otolaryngology is the medical field that treats problems of head, neck, and throat. For example one common procedure is the ear surgery. The surgical procedures are characterized by the delicate nature of the tissue. A typical setup is depicted in Figure 3.2. As can be observed the surgeon works behind a stereo microscope. Hand-eye coordination is impacted by that fact already. Any change in the field of view breaks the hand-eye coordinate even more. As a consequence the surgeon will rely on the scrub nurse technician to hand him/her the tool instead of actively looking for it. A robot scrub nurse like Penelope would not be suitable here. This is because an essential function of the robot is to actively seek out the surgeon's hand instead of expecting the surgeon to find the robot. The surgeon will also expect the nurse to take the instrument from his hand when finished. This nurse-surgeon handshake occurs almost entirely on a haptic level (by touch). The number of instruments used in this type of surgery goes into hundreds. While the surgeon is intimately familiar with his field of expertise the scrub nurse might not be.

In addition to the surgeon and the scrub nurse, other staff involved in the surgery include an anesthesiologist, a circulating nurse, and any number of additional instructor or student surgeons. During the operation a circulating nurse which is not sterile can

enter and leave the room as needed. The anesthesiologist is stationary and of little concern. The surgeon is frequently accompanied by a team of students or colleagues. At times the primary surgeon sitting behind the microscope might address either the scrub nurse, the circulating nurse, or one of his colleagues. Due to union and safety regulations the scrub nurse must rotate every 30 minutes. We observed that during one particular procedure the scrub nurse was utilized by the surgeon for 16 minutes in one hour. This number might not be representative of every procedure but nevertheless it raises the question of scrub nurse downtime.

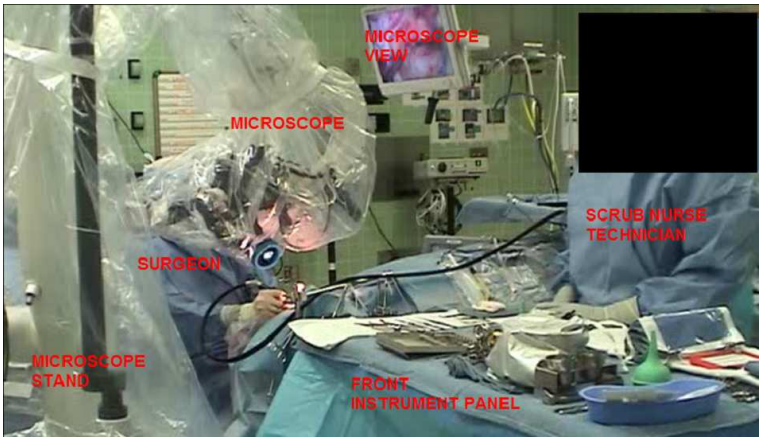

Figure 3.2: Microsurgery operation

3.2 Instruments

The instruments in microsurgery for otolaryngology are in some ways are similar to instruments used by dentists. An example instrument tray is shown in Figure 3.3. They are usually made of metallic materials but some tools such as pumps could be plastic or rubber based. Most tools are discrete objects except power tools which might attach to cords or hoses. For attached instruments the nurse must take care of the instrument and make sure that the hose is not in the way. The working tips of some instruments such as knifes are so small that for the untrained eye, it is difficult to distinguish between them. Their shape is mostly planar and they weigh few ounces at most. Instrument trays will vary from procedure to procedure. They are designed

for human operations. More streamlined instrument servers could be devised to serve a robot. The exact position of each instrument can be recorded ahead of time for automation purposes. The identification of instruments is amenable to labeling. We have used infrared reflective labels for better visual tracking. Additional magnetic and radio-frequency tags could be used for localization in the 3D space and orientation detection inside the grasping mechanism.

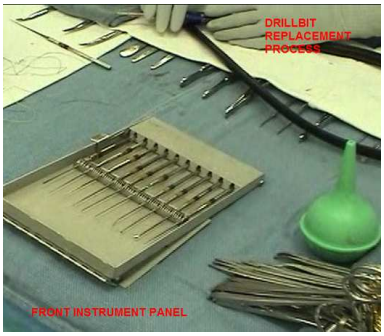

Figure 3.3: Instrument panel

3.3 Robot Platform

Our robotic platform consists of a PUMA 560 robot controlled by a Linux real-time software platform. The software framework which was chosen is named Robot Operating System (ROS [93]). It models the overall system from Figure 3.1. It is a modular environment whereby each component is serviced by a processing node that runs in a separate thread of execution. The synchronization occurs via message passing. The toughest constraint on the platform is imposed by the motion control node. This node needs to send commands to the PUMA's control box at constant intervals, otherwise it locks up.

3.4 Motion Planning

Our motion planning is based on potential fields followed by non-regular cell decomposition of the workspace. We first construct a potential field over the 3D work space. Then a skeleton topology of points furthest from all obstacles is retrieved (white area in

Figure 3.4). Such topology represents all free paths in the environment. Using dynamic programming over the skeleton we can find the shortest path to our goal and then construct a control law that goes along with the path. In Figure 3.4 we can see how this approach works for the planar case. The robot would travel on a trajectory that follows the red path which is extracted from the connectivity skeleton. Most of the work for motion planning has been described in the following references [94, 95][96, 97][98, 99]. Our contribution was to combine the three concepts above and use them. In contrast Penelope, the commercial product, uses a method based on potential fields alone.

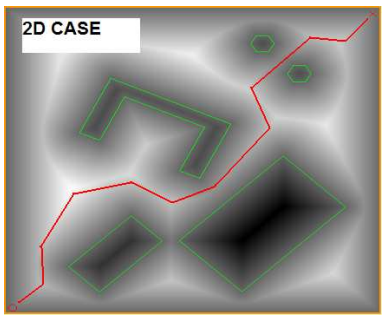

Figure 3.4: Motion planning using potential fields

3.5 Human - Robot Interface (HRI)

Without doubt designing a robot assistant like the robot scrub nurse presents a domain rich in a variety of interesting research problems. As we realized from the start, our effort needed a focus area. This specialization was a necessary trade off between how complete the final design would become and how much we could further any particular research area and contribute something novel. In light of this trade off we understood that the final design would only be sufficient as a proof-of-concept. Given limited time, man power and funding it was far more important to select a key problem that would advance future studies and designs of assistant robots like RSN.

One such key problem is how the robot and the human operator interact. A successful human robot interface (HRI) is essential for at least two reason. First the robot must execute its functions in a manner that is safe for the human operator. Failure to achieve acceptable system integration levels is arguably the main reason why robots

have not been deployed in human workspaces with the same success that they enjoy in more controlled industrial settings. Beyond immediate safety concerns the HRI is important because successful introduction of new technologies depends on the technology being as non intrusive as possible.

Our analysis of video and audio footage from real surgical procedures showed that instrument hand-off involves a delicate sense of touch. One example in Figure 3.5 shows the handing over of a knife.

Figure 3.5: Nurse-surgeon handshake

Since human nurses appear to make heavy use of touch sensing at close proximity during hand-off, we concluded that a mechanized replacement would do well to approximate this condition. It is from this perspective that we picked up the problem of haptic communication and grasp planning. Coincidentally to support the haptic interface we needed to examine two additional modalities. These secondary interface modalities include visual servoing and spoken language interpretation. Visual servoing represents a flexible and least intrusive method to sense the grasping process and the environment. On the other hand natural language processing exposes RSN functionality in the most convenient manner possible by forcing the technology to adapt to human communication instead of constraining the human operator. While all three interface modalities have a rich body of previous work their integration is still far from trivial and their functionality for purposes of robot assistants is still not sufficient.

This study presents our research on the three interface modalities and how they integrate together to facilitate a safe yet easily deployable robot assistant to automate

the responsibilities of a scrub nurse technician. We first go into more detail on the the three modalities and present novel methods to overcome obstacles to their better integration in robot assistants. Then, in the last chapter, we give a critical analysis of our implementation and try to place it within current and future research in this area.

Chapter 4

Haptic Interface

4.1 Introduction

Having described the responsibility of a scrub nurse and identified key components we have a blueprint of technical requirements a scrub nurse robot must replicate. To model the behavior of the robot we utilize a finite state automata, as it is commonly done [7, 8]. One example of such task modeling is given in Figure 4.1.

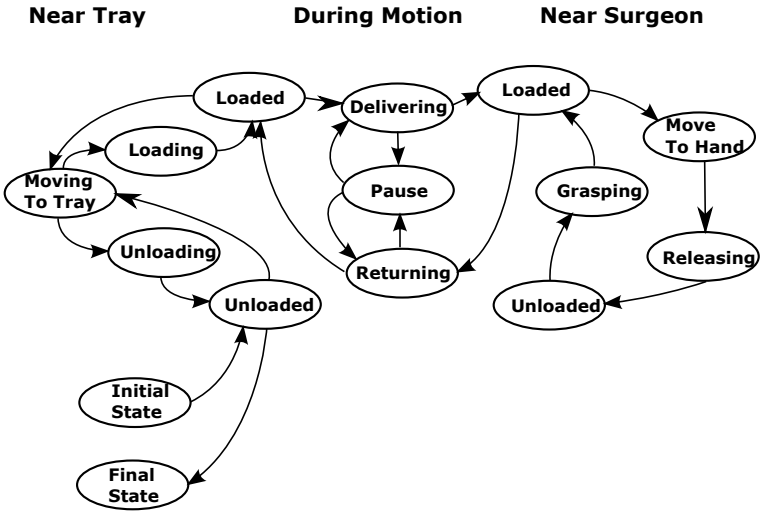

Figure 4.1: Behavior model

This type of model can act as a context at the planning level during visual servoing,

dialog management and during grasping. Some of the events which affect the behavioral model include the following:

- Near Tray
    1. Robot waits for new cycle
    2. Robot moves to load instrument from tray
- During Motion
    1. Robot executes obstacle free motion from tray to surgeon vicinity
    2. Robot moves to stand-by location
- Near Surgeon
    1. Robot executes proximity move and detects placement of instrument
    2. Robot executes a fine proximity move to provide instrument
    3. Surgeon requests an instrument
    4. Surgeon offers instrument back
    5. Robot detects, anticipates and locates.

We next present our own gripper design which was designed to provide soft touch grasping. We first explain its design and then show how it facilitates human-machine communication at the haptic level.

4.1.1 Key Contributions

Our main contribution is the mechanical design of a simple two finger gripper. It uses an inflatable membrane to overcome several challenges. On of these challenges is how to detect haptic events. We introduce here algorithms for haptic signal processing.

4.2 Gripper Design

We started with the simplest possible gripper which is a two finger gripper. However most two finger grippers were designed for industrial settings with fingers being made

from hard materials such as plastic or metal. Most grasp planning algorithms take this detail into consideration and derive grasp strategies based on Coulomb's law of friction or (hard) point contacts.

A gripper design using hard contacts can be visualized in analogy as the task of using chopsticks to pick up rigid bodies (Figure 4.2).

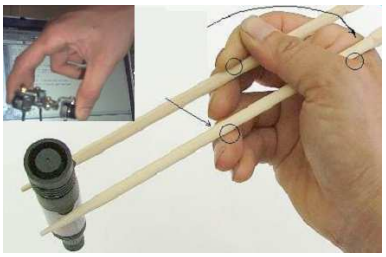

Figure 4.2: Point contact vs surface contact

At any point the two finger hard contact gripper will form two point contacts with the object. Using two point contacts a static force equilibrium can be achieved yielding a stable grasp. However the object is still able to rotate around the axis that connects the two contact points. Compare this to a grasp generated by a human hand (left upper corner of Figure 4.2). Even though two fingers are used, the hand generates a surface contact. The end result is that the grasp stops the object from moving and rotating.

During our research we identified this property of the human hand as essential to our design consideration. What appears necessary is a grasping mechanism that generates soft, shape conforming and adaptive surface contacts. Such surface contact is not as easy to deal with because we have to employ a Hertzian contact model but it addresses many of the above mentioned needs. In previous work [79] we showed how the traditional grasp planning approach can be augmented to address this problem.

To mimic the human hand and produce soft shape conforming grasps we designed a simple two jaw gripper shown in Figure 4.3.

The mechanism has two sections. A back section is used to mount all the electronic and other equipment. On the front section we have the actuators and sensors. The loading plate separates the two sections. At the same time the loading plate also acts as a sterility barrier and a conduit for sensors. For example the central slit is used for an up-close camera sensor. The jaws too are equipped with sensors.

Figure 4.3: Gripper assembly (right) and its computer model (left)

The lower jaw is immovable and has tactile sensors on the outside to be used during initial proximity movements. The inside of the lower jaw contains a sensor pad which is used to determine the impression an instrument makes and from that the exact orientation.

The upper jaw has slits for up-close cameras and other sensors which are to be located behind the loading plate. Here we note a potential advantage a robot has over a human scrub nurse. We designed the system with two cameras in mind. One far-out camera for overall rough localization of the hand and a close-up camera for fine movements. In effect a robot features an eye which is installed inside the palm of the hand, a property which a human nurse does not have. On the inside of the upper jaw we have mounted inflatable membranes. These membranes are what generates a soft, shape conforming grasp. At the same time pressure monitors create a feedback loop which is used to control the membrane pressure.

Please notice that as we generate an appropriate grasp we also have means of adjusting the grasp and responding to changes. Two changes in particular are very interesting. One is the change due to disturbances during movement. By detecting onset of slippage we can ensure that an instrument is never dropped. Second type of change is related to haptic communication events. Namely we can now talk in terms of events such as human subject tugging on the robotic hand because it wants to take the instrument. A very interesting case is detecting if a robot is grasping the wrong thing such as going for the surgeon's finger instead of the instrument.

4.3 Haptic Communication Signaling

Having examined the mechanical aspect of our gripper assembly we turn to the signal processing side. Our gripper electronic controller uses an AVR ATmega32 chip. Being of RISC architecture and running at 8MHz provides plenty of computing power right at the gripper. Unfortunately the link between the gripper and our software platform relies an USART-USB bridge which only allows 56K bps. This imbalance in speed and reaction time led us to consider an autonomous reflex behavior. In particular giving the gripper on-site autonomy for certain transient tasks so that they do not have to be micromanaged. The software platform that is further away has to tend to a number of other event sources. Locally at the gripper controller we constructed a simple PD (proportional-derivative) controller which controls the grasping jaw and membrane pressure. The rudimentary state attributes the gripper can assume are: inactive, active, jaw closed, membrane inflated, and loaded/holding. In Figure 4.4 we can observe an overview of states and their interaction. At each state the gripper might utilize a different control law.

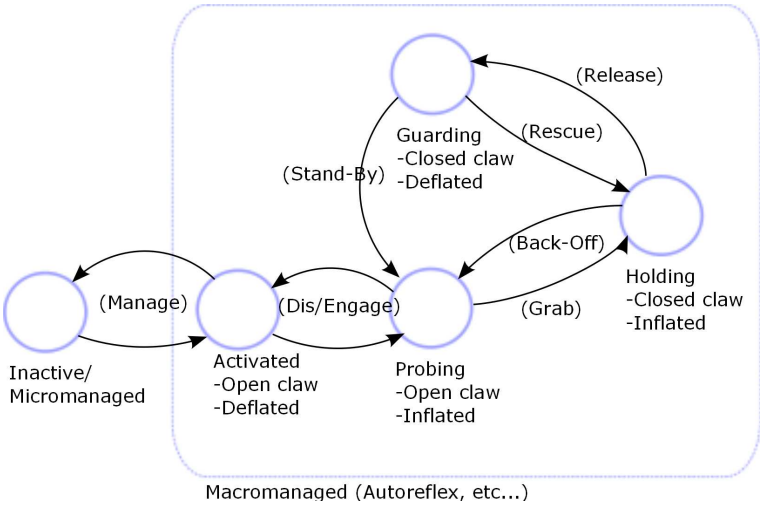

Figure 4.4: Gripper state transitions

Next we present some plots which were obtained during a number of experiments to verify how the simple controller behaves. Again, initially we first tried to use a very simple positional controller but circumstances demanded that we examine the rate of

change in membrane pressure as well. We empirically determined all cut-off parameters used by the gripper to decide if it is holding something or not. Making this more generic is part of our future plans.

In Figure 4.5 we show the membrane and gripper jaw after activation. Once activated the gripper tracks the state and our PD controller will affect local actuators such as the jaw position or membrane pressure. In active state the membrane is pressurized. As you can see from Figure 4.5 we are using about 13 psi at rest. We use a diaphragm micro pump manufactured by Parker-Hargraves that can generate at most 16 psi. That is sufficient for the weight or shape of instruments we need. Please notice a peculiar see-saw pattern in the membrane pressure. The reason for it is a defect in pneumatic plumbing that causes leaking. The leakage would cause a state transition from 'Holding' to 'Guarding' and would result in the loss of the instrument. We could have eliminated the leak but instead opted to make our design flexible enough to deal with such eventuality. For that reason you can see a special 'Rescue' event in the state transition diagram. That is also the reason why our PD controller has to consider the rate of change. Namely at rest the pressure is slowing dropping and that can not be misidentified. Once pressure drops under certain value the controller re-pressurizes. To a degree this behavior can tolerate weakening of the membrane.

From the active state the two most important functions are the ability to detect when to release an instrument and when to take an instrument. In Figure 4.6 a human subject is trying to give a knife to the gripper. Around the time of the 36th second, a spike in pressure is visible. This comes from the human and is interpreted as a haptic command to close the jaw and grab the instrument. After the jaw closes, we see a slight difference in pressure as now the knife is inside the jaw.

In Figure 4.7 we observe the opposite case. Here the gripper is holding something (the same knife again). Again a human attempt to pull it out is detected as a pressure variation and causes the jaw to open.

The next two figures demonstrate the difference or the ability to feel. In Figure 4.8 we see the entire process of detecting the signal to take a knife from the human, holding it while the membrane is slowly leaking and then at some point releasing the knife after detecting the tugging motion.

However in Figure 4.9 we see what happens when the human tries to place a finger

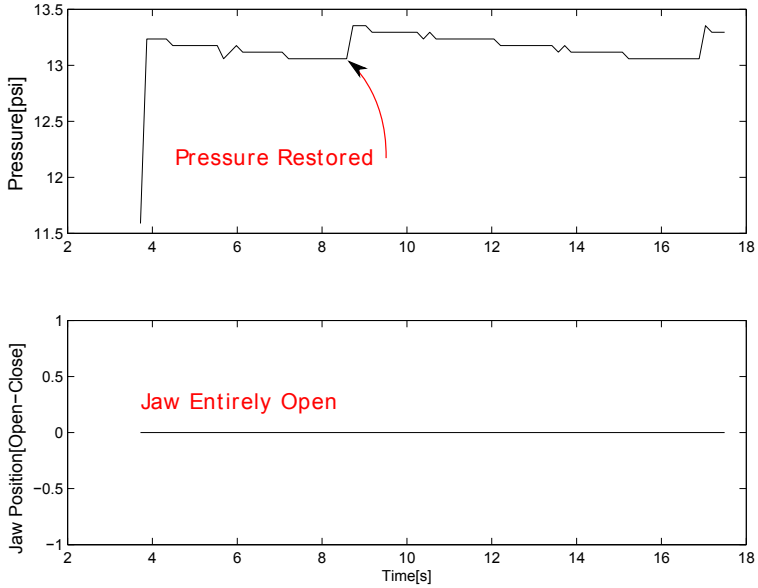

Figure 4.5: Active state: jaw open, membrane inflated and control law on

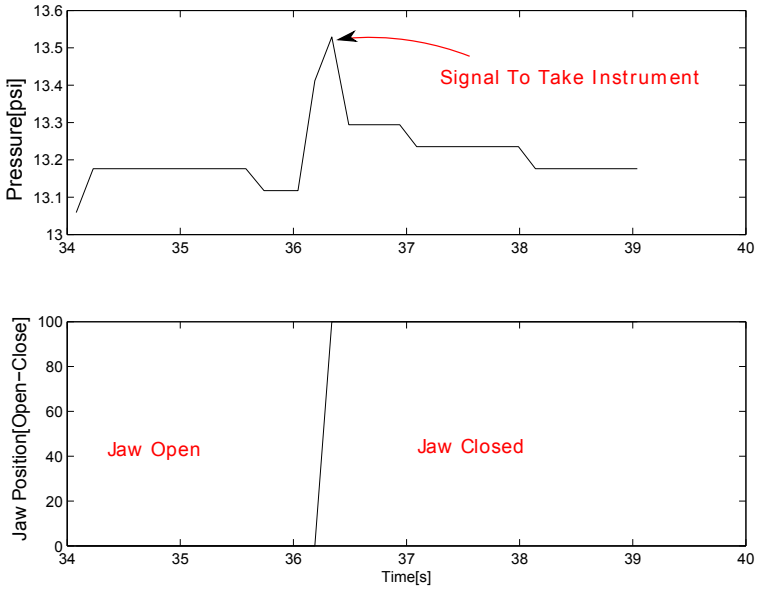

Figure 4.6: Detecting when to take the instrument

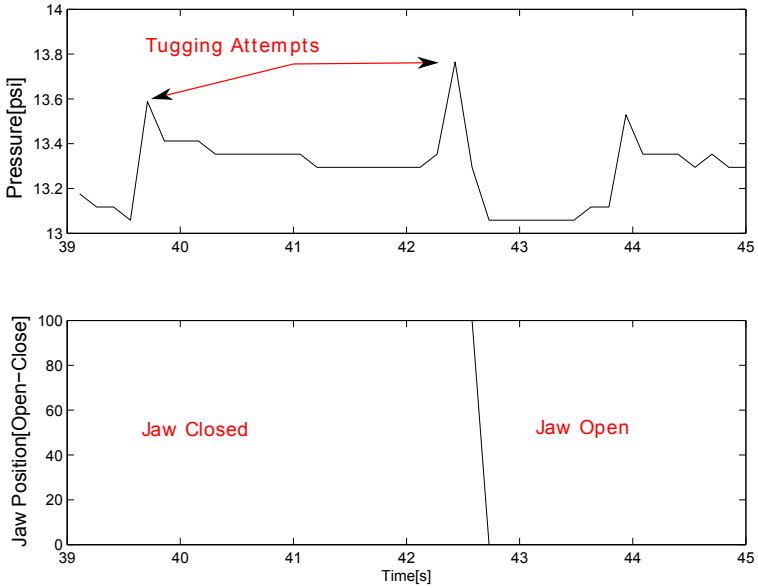

Figure 4.7: Detecting when to give the instrument

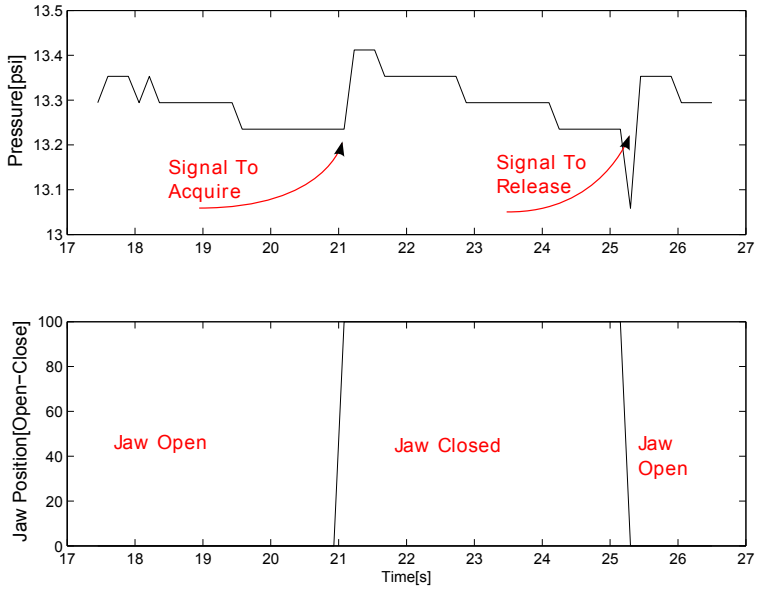

Figure 4.8: Valid instrument: grasping a knife

whose profile is unknown to the gripper controller. What we see here is an autonomous response. The gripper tries to close and hold. However it detects that it is not holding the right object so it backs out and tries again. During open house events the subjective impression visitors get is that the gripper appears to know or have intelligence. The objective reality is that the gripper responds based on its current state and analysis of membrane signals. Such behavior pattern is built-in with the use of a soft deformable contact surface.

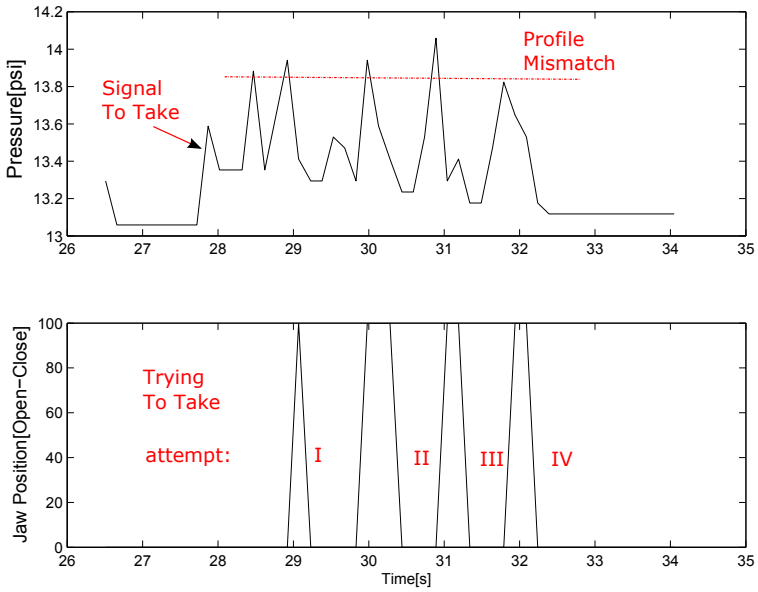

Figure 4.9: A mistake: grasping a finger

Finally in Figure 4.10 we show how the pressure is changing during transport. Here we simulate what happens when the entire robot grabs an instrument at the tray, moves over a distance and then makes an abrupt stop to deliver it to the surgeon. The variation of membrane pressure in this case is due to rocking motion and should not be interpreted as a signal to release or do something else. As you can see the gripper is holding the instrument throughout.

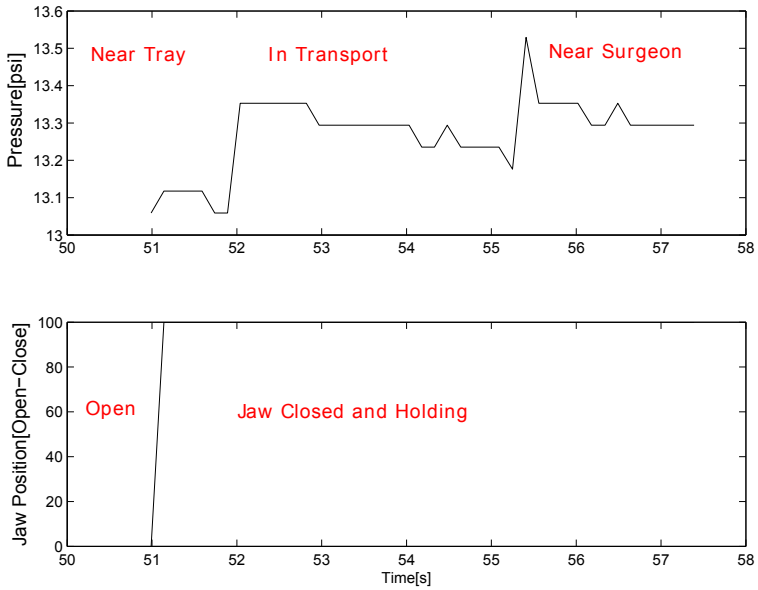

Figure 4.10: Transporting instrument: start, move and stop

4.4 Conclusions

We presented here our work on the gripper during an attempt to build a robotic replacement for a scrub nurse technician. Our grasp mechanism design is different and offers a number of advantages over mainstream techniques. At the moment the grasping mechanism is equipped with only the basic sensors and actuators. Next steps will include improving and adding a variety of sensor modalities. Chiefly among them is vision. We would like to fuse them in a control law that will also be more generic and will not have to rely on hard-coded parameters. Our consideration of the mechanical aspect has not been as thorough as it needs to be. The leak in the pneumatic plumbing illustrates the importance of material selection. In the future we would like to examine material selection for the membrane and also for gloving so that a gripper like this one could actually be made sterile.

Chapter 5

Grasp Planning

5.1 Introduction

Our interest in grasp planning is motivated by the application of robotics in surgical environments. In our case the robot is required to assist the surgeon during microsurgery. For the sake of precision and timing the surgeon needs to keep constant eye contact with the working area. Because of that the instruments are primarily handled by the scrub nurse who delivers them when requested. To automate this instrument delivery process, a delicate grasping tasks needs to be performed. Beyond the usual complications such as obstacles in the environment this particular case also demands that grasp planning be recast as planning a grasp on an instrument already being grasped by the surgeon. In effect the grasp planning that is required is one where the target surface is only partially reachable and graspable.

Motivated by this application we looked at the grasp planning problem as whole. In light of our requirements we developed a grasp planning algorithm and also a mechanical actuator. We described the actuator in more detail as part of the haptic interface. With the algorithm we have tried to stay as generic as possible and that explains why we show some results using unnecessarily more complex scenarios and gripper models.

The algorithm we present here utilizes shape alignment to predict good grasps. After establishing correspondence between points on a gripper and a graspable object (the target) our method returns one or more alignment poses. Finding correspondence is made possible by our use of pair-wise shape descriptors. They are invariant to rotation

and translation and generate a localized and redundant description amenable to determine partial or full alignments. We hypothesize that good alignments (as defined later) should lead to good grasps. The end-result is an algorithm that finds grasps based on on-line evidence as opposed to using off-line precomputed results and that executes the planning process in time, when needed. While this algorithm is helpful in grasp planning it does not represent a complete system. Another problem which we mention here but do not extensively explore is that of characterizing the space of all degrees of freedom also known as pre-shape selection. Within this space of gripper parameters lie all possible grasps. Because the gripper is non-linear and the space high-dimensional brute force enumeration is intractable. Now in the case of a Robot Scrub Nurse the gripper and the instruments are known ahead of time. That means in practice precomputing all good grasps is an option that makes sense. Nevertheless we looked into the problem and describe one possible approach that comes from machine learning community. In this approach we model the act of grasping as a Gaussian process and use a database of proven grasps to train the model so that it can predict which DOF pre-shape goes with any given object.

5.1.1 Key Contributions

Our primary contribution of this chapter is the grasping method. The grasping method uses pairwise features to establish geometric alignment. We study how the geometric alignment affects the quality of grasps. To support our method a novel object alignment method is also presented. This method is based entirely on quaternions. Its advantage over mainstream method lies in the lower complexity while maintaining same accuracy and speed. That is a suitable characteristics for embedded environments such as robots.

5.2 Problem Statement

In this section we present our problem formulation. We outline conventions and assumptions which we use throughout the paper.

Given a gripper $G(\vec{\pi})$ and a target object T we seek to find a set of optimal parameterizations $\vec{\pi}$ that yield good grasps.

The gripper "G" is modeled as a triangulated mesh (Figure 5.1) and parametrized

by $\vec{\pi}$ whose dimension equals the number of degrees of freedom in addition to the three dimensions for wrist position and to the three for wrist orientation ($6D + DOF$). Furthermore "G" is subdivided into "N" fingers with each finger having "M" links. The mesh surface of each link is partitioned into active facets and inactive facets. The active facets are commonly on the inside of the fingers and only they form contacts with the target. While not strictly necessary we also observe that the active facets usually subtend convex shapes [100] also known as grasping convex.

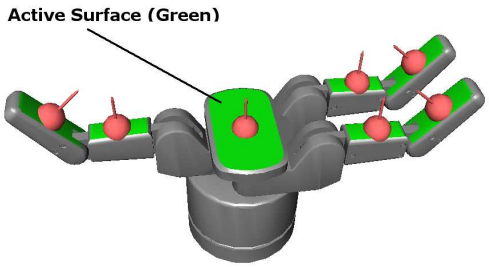

Figure 5.1: Gripper sampling

The target "T" is also modeled as a triangulated mesh. The mesh is partitioned into occluded and free facets. It is the task of the sensor to paint the facets according to this partitioning.

What constitutes a good grasp depends on the application. Frequently a stable grasp is a good grasp but a good grasp might be the one that is sufficiently stable while providing the surgeon the opportunity to also grasp the object in a stable manner. We define an objective function $Q(G(\vec{\pi}), T)$ that is at maximum for an optimal value of π. Given the grasp quality function our primary goal can be reformulated as

$$\hat{\vec{\pi}} \quad = \quad \arg\max_{\pi} Q(G(\vec{\pi}), T). \tag{5.1}$$

During a grasp, active facets of "G" are in contact with "T". To limit the complexity we sample a set of "P" contacts $\{p_1, \ldots, p_P\}$ evenly over the facets. The set of these P samples will be called a constellation. For each of these contacts a set of valid forces within the friction cone can be expressed as an approximation over the "R" edge vectors $\{d(p_i)_1, \ldots, d(p_i)_R\}$ of the cone:

$$f(p_i) = \sum_{j}^{R} \alpha_{ij} d_j(p_i) \tag{5.2}$$

$$1 \geq \sum_{j=1}^{R} \alpha_{ij}. \tag{5.3}$$

Here $f(p_i)$ denotes a contact force at point p_i, and α_{ij} are non-negative coefficients. Figure 5.2 illustrates this approximation step.

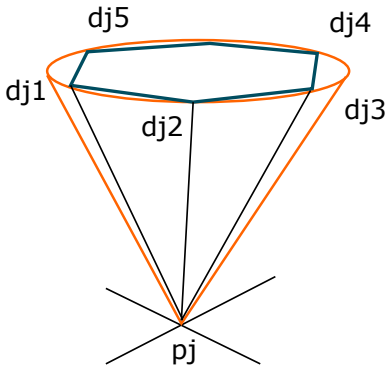

Figure 5.2: Friction cone

If the sum of all forces acting on a body is zero, that body will not move but it might rotate. We therefore consider both forces and torques. The contact wrench is defined as

$$w(p_i) = [f(p_i), p_i \times f(p_i)]^T. \tag{5.4}$$

The overall wrench acting on "T" is then

$$w_T = \sum_{i=1}^{P} \sum_{j=1}^{R} \alpha_{ij} w_{ij}. \tag{5.5}$$

where $w_{ij} = [d_j(p_i), p_i \times d_j(p_i)]^T$.

With the above derivation which follows [43] the target is in force closure if

$$0 \quad \in \quad interior(convexHull\,[w_{11}, \ldots, w_{PR}]). \tag{5.6}$$

One common quality measure is the minimal distance of the origin to the surface of the convex hull.

$$Q \quad = \quad \min_{\vec{w} \in \mathbb{W}} ||\vec{w}||. \tag{5.7}$$

We will make use of it and in addition consider only painted facets on the target as reachable. Finally in the end a ranking metric can be used to order and consolidate alignment and grasp metrics.

5.3 Proposed Method

A robotic assistant in microsurgery must be able to grasp objects held by a surgeon. All other issues aside we need a grasping method that will run in real-time and is capable of planning grasps over partially covered objects.

Our proposed method is inspired by the use of pairwise features in computer vision. We observe that the search for a good grasp necessitates an efficient identification of correspondence (which finger goes where). The same problem is also present in computer vision and object alignment. Another inspiration has been the work done with the GraspIT simulator. By studying it one realizes that their use of preshapes and heuristics to obtain gripper approach vectors is an attempt to simplify the problem of how to align the gripper to the target to obtain a stable grasp. This is where we contribute novel work. For a given DOF parameterization of the gripper we find optimal geometric alignment between the gripper and the target. By maximizing the contact surface between the two we should obtain a subset of poses leading to a good grasp. In order to match $3D$ object of different parameterization we utilize pairwise feature vectors. These features are invariant to translation and rotation and therefore simplify the pose estimation problem. Please notice that unlike in vision we do not need more invariance and so designing the features is easier.

Given an "N" finger gripper our method finds partial and full grasp alignments and works even if certain parts of the target object are occluded (i.e., surgeons hands).

Direct geometry alignment is an expensive proposition. Partial matching makes it even more difficult. Our method lifts the representation of T and G into an invariant space. The only requirement is that the facets we match be normalized to have approximately equal size. Such unit facet will ensure that we test the target face multiple times if there is room for the finger to move.

The end result of alignment is to obtain the orientation $\vec{R}$ and the position $\vec{P}$ of the gripper wrist. Since multiple results are possible, we rank the results by means of an alignment quality metric. The quality metric we choose considers average distance within the estimated pose and the number of votes or correspondences reporting it:

$$QA = \frac{VoteCount}{1 + DistanceVariance}. \tag{5.8}$$

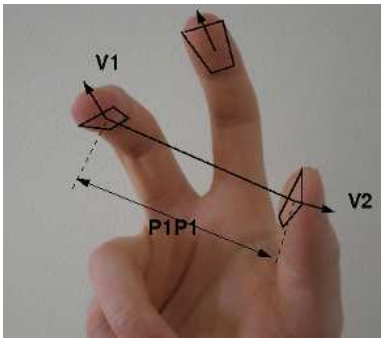

Figure 5.3: Design of features

We now describe the construction of our feature vectors. As illustrated by Figure 5.3, we consider two unit facets sampled over the active surface of the gripper. Each facet will have a position p_i and a normal v_i. From this information we compute three quantities. The distance d_{ij} between two facet positions:

$$d_{ij} = ||\vec{p}_i - \vec{p}_j||. \tag{5.9}$$

The cosine similarity ϕ_{ij} between the normals $\vec{v}_i$ and $\vec{v}_j$:

$$\phi_{ij} = \vec{v}_i \cdot \vec{v}_j. \tag{5.10}$$

As the third quantity we compute the cosine similarity between the normals and the line between the points p_i and p_j:

$$\beta_{ij} = \frac{\vec{v}_i \cdot (\vec{p}_i - \vec{p}_j)}{||\vec{v}_i||||\vec{p}_i - \vec{p}_j||}. \tag{5.11}$$

We now construct our feature vectors as:

$$e_{ij}(\pi) \quad = \quad (d_{ij}, \phi_{ij}, \beta_{ij} + \beta_{ji})^T. \tag{5.12}$$

Note we could have kept β_{ij} and β_{ji} separate which would give us a directional feature. However when they become equal we loose the directionality and have to deal with ambiguity, so we assume the ambiguity is there by design.

To find all possible combinations of "N" contacts of "G" over the geometry of "T" having "V" vertices including the partial matches we consider both "G" and "T" as point sets (centered on unit facets). Then the size of the search space is given by

$$NC \quad = \quad \sum_{i=1}^{N} \binom{V}{i}. \tag{5.13}$$

if we sample the gripper active surface once per finger (i.e., tip of the finger).

Our method obtains $\frac{V(V-1)}{2}$ pairs for a target object "T" with "V" unit facets. For a gripper "G" sampled at "P" unit facets across all fingers we obtain $\frac{P(P-1)}{2}$ pairs.

One way to do pairwise matching is all against all. A more efficient way is to build a volume hierarchy of the target using Axis Aligned Boxes (AAB). For a well balanced binary tree the size of the search space becomes:

$$C \quad = \quad \left(\ln \frac{V(V-1)}{2}\right) \times \frac{P(P-1)}{2}. \tag{5.14}$$

5.3.1 Pose Recovery

Once pairwise correspondences are found, we estimate orientation first and then position. When we say we have correspondence it also means we have two unit vectors $\vec{A}$ and $\vec{B}$ that are related by a rotation.

$$\vec{B} = Rq(\theta)\vec{A}. \tag{5.15}$$

A naive approach for orientation recovery would be to pick the three vectors a correspondence gives us (two normals and a difference) and invert the rotation matrix. We can obtain two possible rotations this way because we do not have point correspondence. However this requires matrix inversion and provides no means of dealing with singular cases (i.e., where two or more vectors are co-linear). Our approach is more efficient in

terms of computation and stability. We find six rotations from a single correspondence of which three should agree.

To explain our approach we try to visualize rotations in Figure 5.4. Any rotation can be expressed using unit quaternions or versors. A versor encodes an axis of rotation and an angle. For example in figure 5.4 one axis of rotation goes through the south and north pole $\vec{N}$. A set of rotations is depicted as a thick arc on the equator. One would think that to recover the rotation we could simply take the cross product of $\vec{A}$ and $\vec{B}$. The problem is that an entire set of of versors can rotate A into B. Any quaternion lying on the great circle that travels half way between $\vec{A}$ and $\vec{B}$ will do the job. The versor having axis $\vec{N}$ rotates our points using the smallest angle. Versor $\vec{C}$ is the other extreme and rotates by 180 degrees. A pairwise correspondence gives us three vectors. For each correspondence we can register a great circle on a spherical map and then pick pairwise intersections with the most votes. The problem is that spherical to cartesian mapping is not uniform. In addition we would need two coordinates for the axis and one for the angle requiring a $3D$ accumulator and it is well known that Hough transform suffers from artifacts when the bins are too large. As an alternative we go a step further and actually find intersections between two great circles. Accounting for directional ambiguity pairwise correspondence gives us six vector pairings. Using these vector pairings we can verify versor agreements in angle magnitude. In case of singular cases which are unconstrained we revert to using the axis with the least angle (cross product).

First we observe that the set of all versors rotating A into B can be expressed as:

$$\vec{n} = cos(t)\vec{N} + sin(t)\vec{C}. \tag{5.16}$$

The intersection of two such great circles is given by:

$$\vec{m_{ij}} = (\vec{N_i} \times \vec{A_iB_i}) \times (\vec{N_j} \times \vec{A_jB_j}). \tag{5.17}$$

In Figure 5.4 this axis is shown as the intersection between the red and black equator line and going through intersection points $X^{'}$ and $X^{''}$.

To test if two circles really intersect at that point we must also check that the angle amounts are in agreement. For a given intersection point we calculate two angles one for each A,B pair:

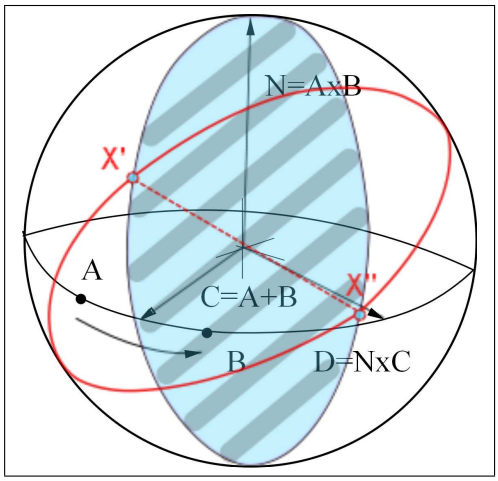

Figure 5.4: Pose recovery

$$\begin{aligned} \theta_i &= \angle(\vec{A}_i - (\vec{A}_i \cdot \vec{m}_{ij})\vec{m_{ij}}), (\vec{B}_i - (\vec{B}_i \cdot \vec{m_{ij}})\vec{m_{ij}}) \\ \cos(\theta_i) &= \frac{[\vec{m}_{ij} \times (\vec{m}_{ij} \times \vec{A}_i)] \cdot [\vec{m}_{ij} \times (\vec{m}_{ij} \times \vec{B}_i)]}{|(\vec{m}_{ij} \times \vec{A}_i)||(\vec{m}_{ij} \times \vec{B}_i)|}. \end{aligned} \tag{5.18}$$

Please notice that two great circle intersect at two antipodal locations and both must be considered as valid. This also impacts how we determine if two angles are in agreement since the same rotation results if angles and axis are negated. We can eliminate axis ambiguity by choosing only upper hemisphere and flipping the angle appropriately. After that the sign of the angle only depends on the angle between the found axis $\vec{m}_{ij}$ and the cross axis $\vec{N}_i$.

The position of wrist is obtained in a second pass from the midpoint of the correspondence (btw constellation points). Given an identified rotation $\vec{q} = (\theta_i, \vec{m_{ij}})^T$ we rotate the midpoint forward in gripper frame and then subtract it from its position in the target frame:

$$\vec{P} = \vec{P_T} - Rq(\theta)\vec{P_G}. \tag{5.19}$$

In the end we could theoretically end up with more than C possible rotations. If that happens, we most likely do not have a good alignment. If we find correspondences for x contacts, then $3[x(x-1)/2]$ rotations should agree on the same pose.

5.3.2 Practical Consideration

In Figure 6.6 we present the pseudo code for our algorithm that summarizes our method.

```
Tesselate T into unit faces
PT← ComputePairs(T)
BVT← ComputeBoundingHierarchy(PT)
Tesselate G active surface into unit faces
foreach pi in (DOFSpace)
    PG← ComputePairs(G(pi))
    foreach gPair in (PG)
        bPairs← gPair intersect BVT
        foreach bPair in (bPairs)
            q← EstimatePose(bPair, gPair)
            p← RecoverPosition(q, bPair, gPair)
            VoteFor(q, p) in Q



RankPose(Q)
PrunePenetration(Q)
ComputeGraspQuality(Q)
```

Figure 5.5: Pairwise feature alignment planner

The structure of selected pairwise feature is where a lot of the application specific engineering occurs. For example directionality can be desirable. Another desirable property is that the facets we try to align be of approximately same shape and size (compatibility). Measures such as facet area or facet aspect ratio come to mind. For this paper we preprocess both surfaces by tessellating them into approximately equal triangles which we call unit facets.

The next practical issue is the distance metric used to detect intersection of features, intersection of orientations and intersection of positions. For the recovery of position given an orientation we use the euclidean distance. The intersection of features is a bit more tricky. The feature elements could have different domains such as distances and angles. We chose to use a weighted distance metric and picked the weighting parameters by manual introspection. For orientation distance measure we opted for angle measurements. For example rotation $(\theta_i, \vec{m_{ij}})^T$ is close to $(-\theta_i, -\vec{m_{ij}})^T$ and so is $(\pi, \vec{m_{ij}})^T$and $(-\pi, \vec{m_{ij}})^T$.

We present the algorithm as a nested loop to emphasize that it is local and sequential in nature and amenable to parallelism. In practice however we converted the loops into three passes. Pass one would accumulate correspondences, pass two would accumulate

rotation agreements and pass three would recover position given rotation.

Post-processing involves actual validation of alignment against the grasping kinematics and dynamics. Besides the actual grasping simulation we also perform interpenetration testing. We picked a very simple feature to capture shape characteristics and it will not guard against all eventualities. Figure 5.6 shows an alignment that is impossible for rigid gripper and rigid body.

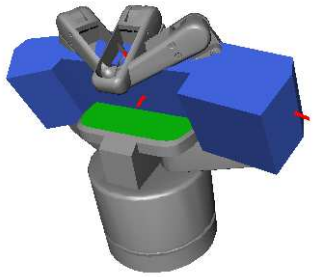

Figure 5.6: Case for penetration test

5.4 Results

The experimental validation of our method was performed in octave using OpenRAVE as the back-end server. OpenRAVE shares many of the features with GraspIT but appears to have larger audience. Besides integrating with the ROS (Robot Operating System) framework it allows scripting in Matlab/Octave, Python and C++. We performed our experiments in Octave to take advantage of computing facilities. In hindsight, after having implemented most of the numeric routines to support side-by-side comparisons, we should have used Python as it allows better structuring of code while still maintaining the benefits of a duck-typed high level interpreter language.

To evaluate our grasp planning method we looked at several characteristics. We examined how our approach to object alignment compares to already established methods. We also examined how surface sampling density affects the accuracy of the alignment process. Finally in the main experiment we looked at how geometric alignment is related to grasp quality.

For the main experiment we selected two grippers and several objects and performed

grasping tasks over the entire DOF space. For grippers we selected the model of our own gripper (Figure 4.3) and that of the Barrett hand (Figure 5.1). Our own gripper represent what we would use in practice, and that is a relatively simple 1-DOF actuator. The Barret hand is a more generic and more complicated 4-DOF actuator but still a modest model in comparison with the human hand, which requires 20 or more degrees of freedom. As for objects (Figure 5.7) we selected some that are rather generic but complex with high polygon count and others which actually show up in day-to-day operation of a Scrub Nurse robot. Here again we notice that surgical instruments for most part are simple almost two dimensional shapes. The fact that we do not need a Barret hand and most instruments in Otolaryngology are flat simply means that our practical setup for Scrub Nurse is simplified. However using this initial simplicity to narrow down the problem would be ill-advised as a case of premature optimization. We mention this as a justification for our attempt to stay generic throughout this study.

Please note that the issue of sampling arises in two contexts. In the main experiment we obtain the results by sampling evenly the DOF parameter space of the gripper. This top most process reflects the search over all gripper parametrization. Another context where sampling is mentioned is the representation of gripper and target as point clouds. In the main experiment the gripper and target are sampled randomly and uniformly. For the gripper we made sure that some samples came from each of the finger links.

At each DOF sample we pick the highest alignment geometrically speaking and then evaluate what grasp quality we get. The grasp quality evaluation is based on equation (5.7). Configurations that clearly penetrate the object were disqualified. For the rest we would perform a fine tuning step during which we slightly perturb the wrist position and DOF values (within the sampling step). During the fine-tuning the best grasp quality is retained. The majority of our computation occurred on the client side but collision checks and link transformation states were queried from the simulator.

We manually performed the target facet painting where it was feasible. For example the flask has a region which is unreachable at the top of the neck.

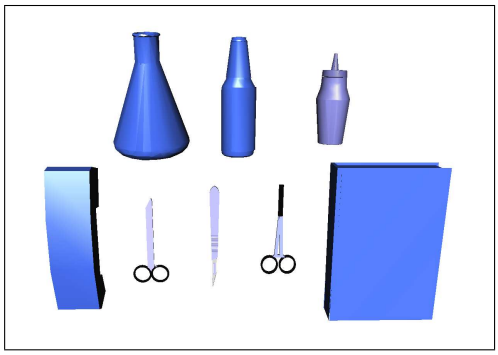

Figure 5.7: Selected targets of varying polygon count

5.4.1 Pose Alignment Validation

The grasping method we present here relies heavily on geometric alignment. In that regard we have developed an alternative to existing methods. Our goal was a simpler, faster and possibly more accurate method than what is already available. The motivation came from our limited computing resources. To ascertain how our method compares with existing solutions we considered a number of proven alternatives. In Figure 5.8 you can review the results that were collected. For comparison we selected two widely used 3D-3D alignment methods by Arun and Horn [77, 78]. We first examined how these methods performed globally when provided with a point cloud. Then we embedded them together with our method (PXT for pose intersect) in a RANSAC process (stochastic best-wins sampling). The RANSAC algorithm returned the best of ten hypotheses. We also had access to the ICP (iterative closest point) algorithm but its accuracy was worse than the others,it would frequently not identify the right point correspondences even for slightly perturbed clouds and its running time was significantly higher. It was therefore not included. All methods were subjected to increasing amount of Gaussian noise. The method labeled as "hypVoid" is there for reference. It shows the error we obtain if the hypothesis is drawn from random translation and rotation. From it we see that at around 0.5 variance the results become useless.

As we can see there is no systematic advantage of one method over the other. Both global methods perform almost identically. As expected, they yield better results, but our experiment assumes perfect correspondence and only level of noise varies. Comparative analysis of the runtime also shows that no method significantly outperforms the

others. One important point to make is that while we implemented our method entirely in Octave the other two methods make heavy use of optimized code for eigenvalue and singular value decomposition. We were tempted to use this as a sign that our method could be faster in addition to having lower complexity. However, after implementing all three methods in C++ we found that the timing did not change much. One reason is that in RANSAC setting both Arun and Horn method work with the minimal number of points and any complexity arising from use of EIG or SVD algorithms appears minimal.

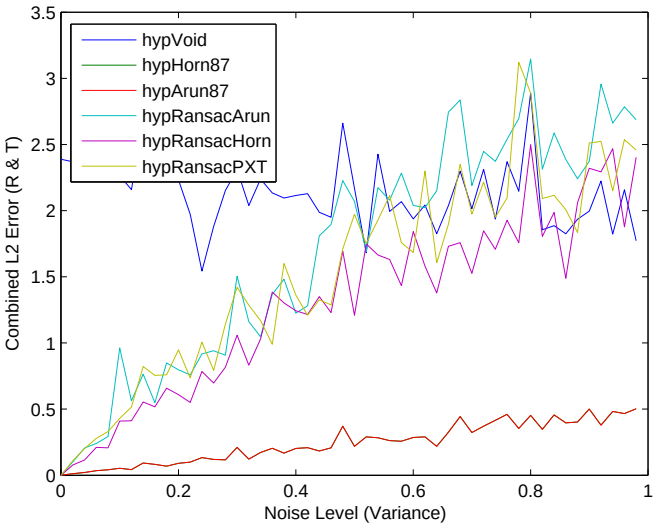

Figure 5.8: Alignment error

5.4.2 Selection Strategy of Pairwise Features for Grasp Planning

For the alignment of 3D objects direct methods require knowledge of point correspondences. Other methods like ICP (Iterated Closest Point) can work without this information but since they solve two problems simultaneously there is a drawback. We belong to the first group. To help establish correspondence we make use of pairwise features but there is a problem. For a gripper with N points we obtain $N(N-1)/2$ features. Similar number of features compute from the target and matching them becomes a problem quadratic in N. Also many of the features are redundant. Naturally, the question of selecting salient or good features imposes itself. To that end we formulated three

different feature selection strategies and compared them against the full all against all feature matching.

In figure 5.9 we measured the alignment error against the size of selected features. We picked one of the targets and created a point cloud by sampling all facets into equal size. It was then rotated and translated by a random amount to represent create an alignment problem void of outliers. The original point cloud was always sampled uniformly with increasing sample size while the rotated cloud was sampled using one of the presented strategies. The features obtained this way on the two point clouds were then used to recover correspondence and then the pose. The feature selection strategies include: Completely, Randomly, Most Distant Feature and Most Orthogonal Feature. The complete selection strategy is to use all features. The random strategy samples randomly over the entire set. The most distant feature selection picks features that have the largest radius to the closest neighbor. Finally most orthogonal pair is an attempt to select features that have most orthogonal position and normal. As you can see the results show that selecting only a subset of all pairwise features is definitely justified. However our results show that neither of our heuristic methods significantly outperforms the random case.

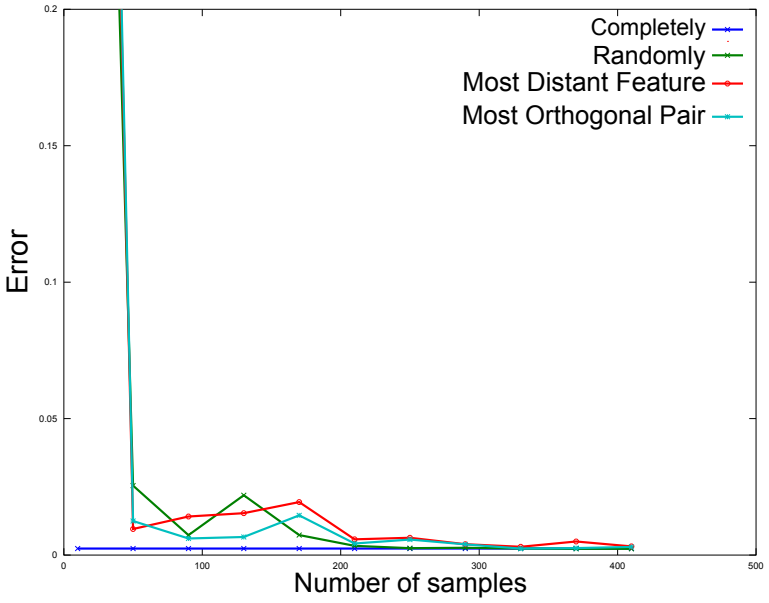

Figure 5.9: Feature selection strategies

5.4.3 Grasp Planning Results

We have already discussed algorithmic complexity and here we present two results that we feel support the premise of this paper.

First in Figure 5.10 we show a projection of the gripper and target constellation in feature space. We have plotted side by side constellations of a number of grasp configurations (case 2-case 6) and contrast that with the constellation of the target (case 1). We used Multi Dimensional Scaling (MDS) to visualize the data as it is commonly done in machine learning. The intent of this experiment is to demonstrate that the gripper constellation has a degree of separability from the target constellation and that this degree depends to some extent on the parameterization. Simply put we have proposed here a method that quickly finds gripper to object alignment but we still have the problem of sampling the DOF space and searching it. As Figure 5.10 demonstrates there is separability, the search process can thus be optimized. If we can't reduce complexity then at least we can ensure that the least amount of time is spent at each point.

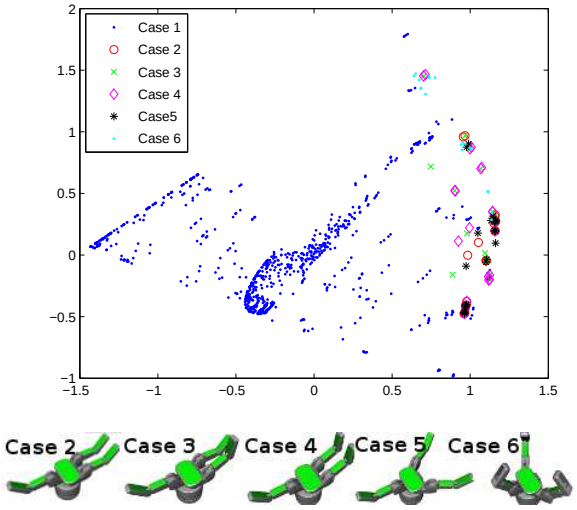

Figure 5.10: MDS Projection of gripper and target in feature space

The second result follows in Figure 5.11.

It establishes by empirical means the relationship between gripper-target alignment and the resulting grasp quality. The reasoning here is that if there exists proportionality

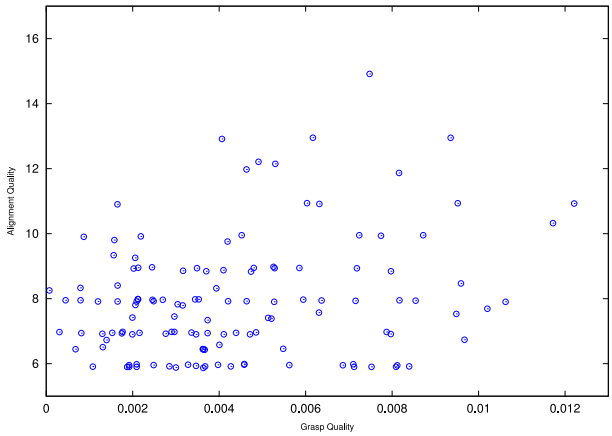

Figure 5.11: Quality of alignment vs quality of grasp

between the two, we might be able to cast part of one problem in terms of the other as we have attempted to do here. While this transitivity might not have any benefits, it does provide one extra tool in pursuing the grasping problem. As can be seen from this figure that relationship is not as clear cut as we initially assumed. Instead of clear proportional relation we get a more measured behavior. It appears that a good grasp need not be optimally aligned but at the same time if we have a good alignment our chances of obtaining a good grasp are better. The results seem to suggest that a good alignment could indeed be used a heuristic in the search for a good grasp.

Lastly, Figures 5.12-5.14 present a qualitative analysis of our work. They show the top grasps found by the method described here on the three sample targets. As we can see each grasp indeed agrees with intuitive expectations of a good grasp.

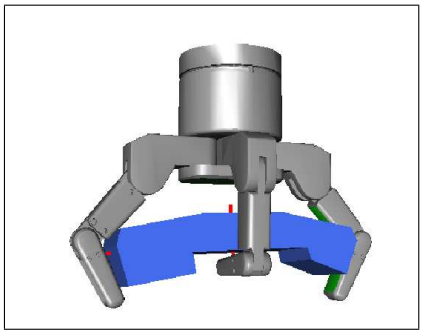

Figure 5.12: An example of alignment grasps

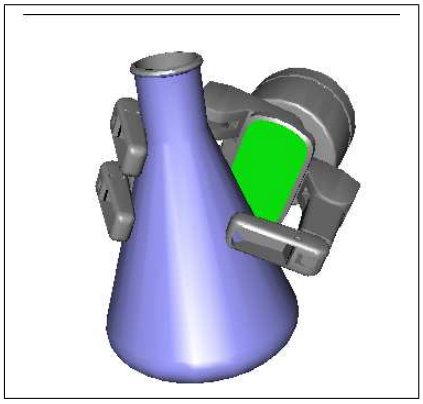

Figure 5.13: An example of alignment grasps

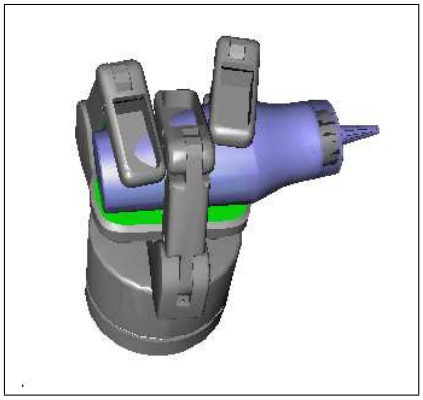

Figure 5.14: An example of alignment grasps

5.5 Conclusion

We have presented here a method that addresses a sub-problem of the grasp planning problem. The sub-problem arises from the high dimensional search space in which a grasp planner must operate to find the approach direction and position of the gripper wrist relative to a target. Our initial assumption was that good grasps require good alignment. While we have not tried to reduce the search space spanned by the degrees of freedom (DOF) we have explored the separability of a gripper and a target in the feature space. The idea is that absent any means of reducing the DOF space we could construct a classifier that might quickly eliminate certain regions.

This research was performed within the specific context of a human robot interface (HRI) needed in medical robotics. We believe that the majority of grippers used today including the Barrett Hand utilized here are rigid tools and as such more suitable for industrial tasks. Even if soft rubber fingers are attached the active surface is used passively and is blind. Robots capable of human interaction should feature soft, deformable active surfaces. For one the softness prevents damage to the human subject (hand). Additionally the soft surface provides for embedding of sensor matrix needed for the increased dexterity requirements. It also generates shape conforming surface grasps. In this light even "overgrasps" such as Figure 5.6 can be considered as mildly valid.

Future work will involve direct field studies in the operating room during which we intend to record how the surgeon interacts with an assistant at the haptic level (since his eyes are on the microscope). Besides collecting evidence for a touch based communication protocol this field study will also examine how the instrument is obscured (partial coverage) during the interaction. The results here and from such field studies will allow us to complete a novel shape conforming gripper for HRI being developed in our lab.

Chapter 6

Visual tracking

6.1 Introduction

In robotic applications vision is not the only way to sense the environment. Some of the other modalities include laser, radio-frequency (RF) tags, tactile sensing or sonar. Each modality performs differently in terms of accuracy, range or level of intrusion. A visual tracking system offers high versatility and accuracy. Furthermore, it is not intrusive to humans. For best adaptation rates a novel technology needs to introduce minimum on interference. For example one of the biggest obstacles of enabling robots like Da Vinci [1] is that they stand between the surgeon and the patient. In this position the robot undermines the surgeon's hand-eye coordination, arguably their most important skill. The visual system we have in mind would track the hands of the surgeon and that information would be used for accurate and safe instrument delivery by the robotic nurse.

This chapter is a study of how well existing methods perform and how they can be reliably integrated. We hope that these findings will be helpful in advancing future efforts to implement vision in assistive robotic settings.

We examine both on-line and off-line tasks that are needed. The on-line tasks are signal processing operations that occur during robot use. Their purpose is to track an object. We assume that we have a suitably placed and calibrated camera, a feature detector that extracts essential statistics from each camera frame and a 3D model of the hands. Operating over the extracted features the first step is to identify correspondences

between image points and points on our model. After that we perform depth recovery to obtain 3D points in the camera frame. The final step is to align these points with the model frame. The recovered pose consisting of rotation and translation is then available to the robot motion planning to calculate a trajectory for instrument delivery.

To achieve robust on-line tracking we must select good features to track. This selection process takes several steps and is done off-line. How we accomplish these steps frequently impacts the quality of the on-line process. A naive approach would be to manually collect parameters such as the intrinsic camera parameters and measurements of the model and then to assume that the same feature can be found from one frame to the next by using their proximity. We go a step further and examine how these tasks could be automated. We pose the task of feature selection as a clustering problem. We assume the number of clusters to be equal to the number of points being tracked. Each cluster is constrained to contain no more than one feature from every frame. Our goal is to retrieve clusters containing features that are in correspondence to the same point on the 3D model. The features are clustered across frames using their proximity in feature space. We call this task feature selection even though in computer vision the term is sometimes used to describe feature extraction. In our case we are given a feature detector such as SIFT [63] or SURF [62]. Our clustering in effect provides a tracking for each point simultaneously. Based on the estimate of the within-cluster feature variation we choose the most reliable points to track. We next investigate structure from motion methods and how they recover 3D shape from 2D observations with known correspondence. These methods make it possible to extract model description driven by available observations. In the best case we could skip manual measurements on the real object. A positive side effect of the projective reconstruction is that it also returns intrinsic camera parameters.

After outlining and testing the essential tasks above we then proceed to evaluate the overall performance for a number of different tags that could be attached to the back side of the surgeon's hand. The results can then inform our ultimate selection of markers used for the operating room.

6.1.1 Key Contributions

The main result of this chapter is a vision system with a novel clustering method to select good features. The selection process is also supported by specially designed marker tags. Having demonstrated good tracking performance, the features obtained by our main method are then used to reconstruct 3D models in the training phase. The method to accomplish this task was specially designed for cases of missing correspondences. That is to say, in our case a matrix of 2D point correspondences will have missing values that need to be recovered in tandem with the 3D reconstruction.

6.2 Problem Statement

In the field is Otolaryngology a common procedure is the operation of the inner ear. The surgery is performed under a microscope. The robot would replace the scrub nurse technician. Its primary function would be to manage instruments laid out on the table and deliver them from and to the surgeon. The instrument delivery is initiated by the surgeon but driven by the scrub nurse. The scrub nurse needs to find the stretched out hand and place or take an instrument. That is the reason why a simple 2D vision system is not sufficient [4].

We need a vision system that can find and track the pose of the surgeon's hands. The operating room is a chaotic environment and occlusions are possible, especially coming from the microscope stand. We will assume that sufficient processing power is available for a high fidelity stationary camera. Other possibilities would be to mount the camera on the robot's hand, use multiple cameras or a moving camera. The human hand presents a challenge for tracking. It is a complex quasi-rigid body consisting of multiple joints that can go in and out of occlusion. Parts of the hand are actively used by the surgeon and others could be engineered to hold special tags. The entire glove could be marked with patterns but during operation blood will cover up parts of it. For simplicity we decided to attach special geometric markers on the flat back side of the hand and consider that for tracking. Using flat segments could be extended to the back-side of fingers and even the palm or the wrist. Ultimately, this tiling approach can be generalized into a full deformable body treatment. Besides simplifying the approach, another advantage we might have is the sterility. The surgeon and the scrub nurse are

sterile. The other staff or objects might not be. In that regard an operating room is a controlled environment. Only sterile objects might come in-between the surgeon and the instruments. That should cut down on the sources of occlusion.

While not the primary goal, the vision system should be extensible to allow for other functions such as obstacle detection and instrument identification.

6.3 Proposed Method

In order to handle instruments a robot assistant must reliably identify and localize the instruments. The task of identification can be addressed by special RF tags, magnetic strips or coloring. We can precompute ahead of time instrument location on the tray. This leaves, as the only big problem, the localization of the instrument when it is held by the human subject.

The robot must be able to estimate the orientation and position of the instrument so that an appropriate approach vector can be computed. Both of these tasks occur when the gripper is in close proximity to the instrument. Because of that we can utilize a variety of sensors such as tactile, magnetic, or pressure sensors. Before the proximity sensors can be used there still remains the issue of moving the robot closer. This problem requires the robot to localize the human hand and the tray from afar. Some sensor modalities commonly used in industrial settings such as lasers are not suitable because of human presence. For that reason the most ideal sensing modality would be vision.

Our vision system follows well established patterns from available literature. In Figure 6.1 we present the main vision pipeline. We engineer the environment to the extent that we attach special tags imprinted with geometric patterns to the back of surgeon's hand. Of course the assumption is that in the first step we need to localize the hands which need to receive the instrument. Given an image captured from a camera we extract suitable features. Initially we considered projective invariant features. One such invariant is the cross ratio [67, 101, 102, 103, 104]. We have spent considerable time pursuing this idea without much success. In the end we reverted to more proven features like SIFT and SURF. Any other feature could be substituted easily. A good feature would be highly discriminative and as robust or invariant to changes in viewing

conditions as possible. In the next step we use the discriminative property of features to find correspondence between a 3D point on a target model and the observed image evidence. Using the so obtained correspondence and geometric information from the model we can then recover 3D depth of each registered point. The methods for depth recovery have been studied extensively for the past 20 years and some of them have been known in photogrammetry for the past century. In our system we use three points of known correspondence to recover depths. After depth is recovered the last step is 3D alignment of point clouds. This step returns the final deliverable which is the translation and rotation of the target model.

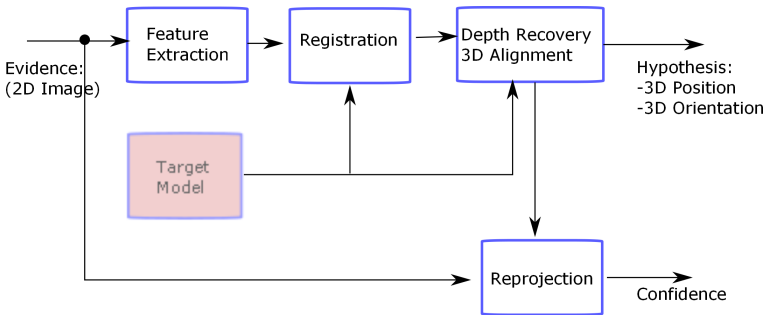

Figure 6.1: Computer vision pipeline

The main pipeline might have some issues with robustness but given the right conditions it is accurate and all methods that are needed are available and have a rich body of literature to justify their use. Where we encountered the real problem is in target modeling. The problem is that we need to identify high-quality or salient features of the model. Frequently the image operations that produce features like SIFT or SURF are analytically intractable which means we can not predict what part of the model will carry features that are robust and can be tracked. So acquiring the right model with the best feature became the real issue to resolve. In Figure 6.2 we illustrate our existing answer to that problem.

We film the target, in our case surgeon's hands, and collect a stack of video frames. From each frame we extract any number of features of some type. Then a minimization loop is entered. In this loop we try to cluster features across different views that might correspond to the same point on the model. Having established correspondence between 2D points over a number of views we can use projective reconstruction methods to

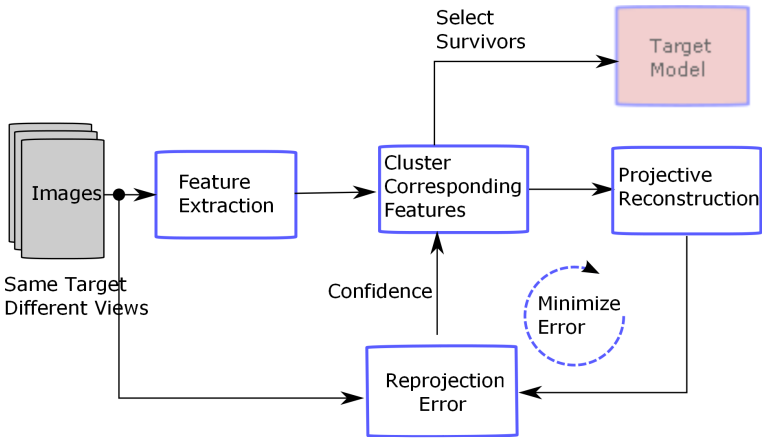

Figure 6.2: Computer vision model acquisition

estimate position and orientation of each view. There are two issues with the loop. Some features are not very robust and might disappear. In addition some features might lack the discriminative power and get assigned to the wrong point. Both of these issues lead to errors in projective reconstruction. Hence the need to iteratively eliminate certain features and reconstruct point correspondences. Once the reprojection error has dropped sufficiently we are left with a number of surviving feature clusters. Each cluster represents a point on the 3D target model each feature in the cluster represents a different view. This surviving clusters are finally selected as our target model. Later in the actual vision pipeline we use nearest neighbor search to match a new feature with a feature from the target model. After that we query the cluster id to identify which 3D point is represented. That 2D-3D correspondence is then what is needed for the final stage of the vision pipeline as explained above.

6.3.1 Assumptions

As explained in the problem statement we assume a stationary camera of sufficient fidelity. The camera model is the ubiquitous pin-hole camera:

$$p_{img} = \Pi(cam, P_T). \tag{6.1}$$

It projects 3D points P_T from the camera frame onto an image plane. The projection mapping Π is non-linear in general. The linear parameters are the focal point and image

center f_x, f_y, c_x, c_y. Additional parameters are the radial and tangential distortions k_1, k_2, k_3, p_1, p_2. We assume the hand is represented by one or more planar patches that yield sufficient number of features and are embedded on a rigid body. The 3D model is represented by a cloud of points where the frame is aligned with the wrist of the human hand.

$$P_{model} = \{P_1 ... P_O ... P_N\}. \tag{6.2}$$

If both hands are allowed in the image, then they will use different tag patterns. The patterns on the tags are singular in appearance from anything else in the scene.

6.3.2 Features

We decided to use two popular features called SIFT and SURF. Both are represented by an image position,scale,orientation and a 128-dimensional feature descriptor used in our distance computations.

$$F_{x,y,\sigma} = \{x, y, \sigma, dir, \vec{f}\}. \tag{6.3}$$

These features are obtained by computing statistics such as histograms in the vicinity of a key point. We chose them because they appear to satisfy most of the requirements for a good feature. The features should ideally latch onto a characteristic of the hand and change little under different viewing conditions. The lighting conditions in an operating room are quite controlled and the projective effects are the most prominent variation. A feature should also be local to avoid drifting when parts of it are occluded. Finally, a feature should be unique and distinguishable to enable easy establishment of correspondence. Please note a very important observation about features. Since we want to track singular patterns, we expect our features to be singular. If a feature arises based on a repeating pattern, it will show up multiple times in the frame. Its radius to the closest neighbor will therefore approach zero. This observation about the minimum inter-frame radius of a feature is exploited later on. Namely, we use this radius in testing for overlap between two features or between a feature and a cluster.

6.3.3 Depth Recovery

Assuming we have 2D-3D point correspondence, the next step is depth recovery. Without correspondence this stage suffers from combinatorial issues. By shifting responsibility for correspondence to the features, we make this step much easier at the expense of the feature. For depth recovery we primarily use the three point method for its simplicity [71]. The three point method is based on the law of cosines. As illustrated in figure 6.3, given three lines $x\vec{a}, y\vec{b}, z\vec{c}$ emanating from camera origin O and going through 3D points A, B, C with side lengths d_1, d_2, d_3 the actual depth of the 3D triangle is recovered by solving three equations in three unknowns:

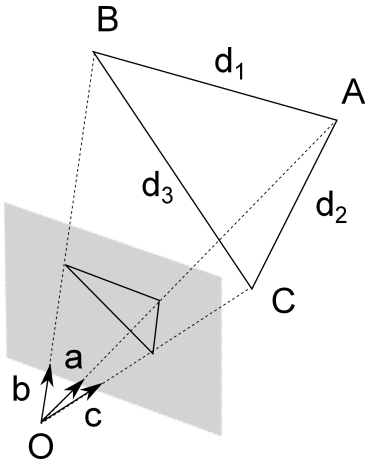

Figure 6.3: Recovering depth from three points

$$d_1^2 = (x\vec{a} - y\vec{b})\dot{(}x\vec{a} - y\vec{b}). \tag{6.4}$$

$$d_2^2 = (x\vec{a} - z\vec{c})\dot{(}x\vec{a} - z\vec{c}). \tag{6.5}$$

$$d_3^2 = (y\vec{b} - z\vec{c})\dot{(}y\vec{b} - z\vec{c}). \tag{6.6}$$

There are multiple solutions to this system. We pick the solution for which the covariance of the triangle sides is at the maximum.

$$\max Cov([A-B, A-C], [x\vec{a} - y\vec{b}, x\vec{a} - z\vec{c}]). \tag{6.7}$$

6.3.4 Pose Alignment

Once we have recovered depth in camera frame, we proceed to align the points to our model frame. The alignment process yields a rotation and a translation that explains the pose of the camera or the object when the image was taken. The pose alignment problem recovers the rotation R and translation T which transforms a cloud of original points (model frame) to a cloud of transformed points (camera frame) as follows:

$$P_T = \mathbf{R}P_O + \mathbf{T} = [R, T]\, P_O. \tag{6.8}$$

In [79] we introduced a method that recovers the rotation (from which translation follows) given only two correspondences i, j. Recalling that all possible rotations from A_i to B_i are of the form:

$$\vec{n} = cos(t)\vec{N} + sin(t)\vec{AB}. \tag{6.9}$$

The intersection of two great circles over the unit sphere of rotations is then given by:

$$\vec{m_{ij}} = (\vec{N_i} \times \vec{A_iB_i}) \times (\vec{N_j} \times \vec{A_jB_j}). \tag{6.10}$$

If the angles for both correspondences i, j are close or equal we have recovered pose by intersection.

$$\theta_i = \angle(\vec{A_i} - (\vec{A_i} \cdot \vec{m}_{ij})\vec{m_{ij}}), (\vec{B_i} - (\vec{B_i} \cdot \vec{m}_{ij})\vec{m_{ij}}). \tag{6.11}$$

6.3.5 Selection of Good Features

Before we can proceed with tracking, we need a means of establishing correspondence between image frames. Frequently in literature this step is overlooked and commonly the features are tracked as long as they are close spatially to each other. We pose the problem of finding good features to track as a clustering problem, whereby the number of clusters is assumed equal to the number of points tracked. Each cluster is assumed to contain no more than one feature from each frame. The resulting clusters provide us with a means of establishing correspondences. The advantage of this approach is twofold. First, it does not assume that every feature is necessarily present in every frame. Secondly, by analyzing the clusters one can obtain better insight as to how invariant and how discriminative the used features truly are.

Methods which simply use spatial proximity typically rely on the nearest neighbor. To illustrate potential problems with such methods, we present two examples in Figure 6.4. The lower example is what happens to an observation (right frame) as we try to identify closest neighbors in the model (left frame) using nearest neighbors in feature space. Clearly we must be able to reject an observation and not try to assign all of them. The upper example shows what happens if we try to infer a distribution over the model features by means of a radius of rejection. In this case the radius of rejection is given by the distance to the closest neighbor. Granted we obtain fewer correspondences but the quality of the result is much improved. The goal in our approach is

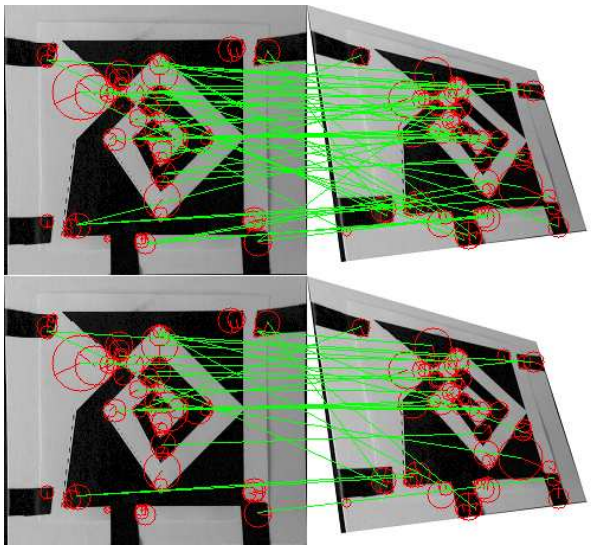

Figure 6.4: Feature correspondence (with and without radius rejection)

to cluster all features across a stack of frames, if they correspond to the same point on the model.

Our clustering scheme 6.6 starts by computing the minimum radius for each feature within a frame, which is given by the distance to the nearest neighbor. This radius can be considered as a crude estimate of the underlying cluster given only the current frame. It will be almost zero for any feature that is repeated and greater than zero for distinguishable features. We repeatedly try to grow clusters until we have assigned all points to a cluster, this is accomplished by iterating over all frames. Each cluster is grown by adding the closest feature that also overlaps the cluster. The overlap is determined by considering the the values and radii of the features respectively. A

feature overlaps the cluster if it overlaps any of the cluster points already added. We observe that for singular features there can only be one feature per frame at most. We modify the cluster to include only one feature from any given frame. Finally, we note that as a cluster grows, it might find the closest point to be already in a different cluster. In that case the point is moved to the new cluster and the old cluster is reprocessed in the next cycle.

Our clustering scheme is illustrated in Figure 6.5. As stated, we are trying to connect features across multiple frames by testing for spherical overlap. The cluster grows by adding points using the minimum distance. This process gives rise to minimum spanning trees. On the right side we see that any new point is added because it was closest to an older member of the cluster. We record for each cluster member the minimum distance that was used when determining whether or not to include it. This distance is later used to decide if a point should be moved to another cluster. If a point is moved to another cluster, any points that were added because of it will also end up in the new cluster and therefore maintain the minimum spanning property.

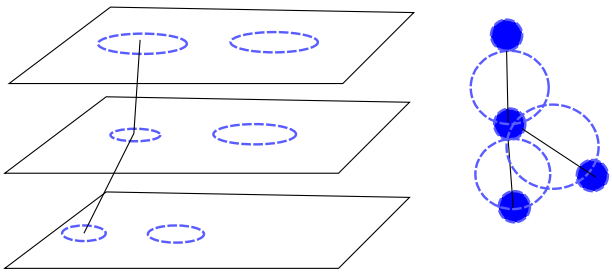

Figure 6.5: Clustering process (across frames and within cluster).

6.3.6 Projective Reconstruction

Which point on the model is tracked depends, to a great extent, on the best features selected by the clustering stage. This fact might preclude us from acquiring the model description ahead of time. Instead, the selected features will dictate what is relevant. This situation presents an opportunity to automate an error prone and manual task. At issue is the prospect of measurement error for each and every point that is recorded by hand. Projective reconstruction methods offer a solution here. The reconstruction methods also have the benefit of providing us with camera calibration parameters. At

```
points.clear()
clusters.clear()
# not all features appear in every frame
for feature in frames :
    feature.computeMinRadius(ownframe)
    feature.setCluster(null)
    points←points ∪ feature
done←False
while(done≠True):
    # seed cluster with one point or restart first open
    cluster←seedNewOrOpen()
    if cluster is Empty:    done←True
    cluster.close()
    while done≠True and cluster.isGrowing():
        next←cluster.findClosest()
        if hasOverlap(cluster,next)=False: skip
        prev←next.getCluster()
        dNew←cluster.distanceTo(next)
        dOld←prev.distanceTo(next)
        if(dNew<dOld):
            cluster.addMember(next)
            prev.removeMember(next)
            prev.open()
```

Figure 6.6: Feature clustering algorithm

the core of the reconstruction problem is the following equation:

$$p_{img}^{i,j} = \Pi(cam, [R,T]^i P_O^j). \tag{6.12}$$

where $i = 1 \ldots M$ specifies number of views and $j = 1 \ldots N$ specifies number of points in correspondence. The known variables are the observed 2D points p_{img} and possibly camera parameters. Making use of the SFM Toolbox [91] we make use of the Sturm and Triggs algorithm.

6.4 Results

We performed two sets of experiments. Baseline experiments evaluate how elements of the system behave. For that we look into model fitting under noise for a number of different algorithms. In the same experiment we also use MDS to embed in 2D a few of the feature clusters. This is done to gain insight into the cluster variation. For the baseline experiments the book video was the template (Figure 6.7). It is about 550 frames long. After that we use the tracking system and input a number of distinctive

tag patterns (Figure 6.7). For each pattern we evaluate how well the tracking performs by measuring re-projection error.

6.4.1 Experimental Setup

We started out expecting to manually acquire the model points. During that phase we manually calibrated our camera. The parameters we obtained using the MATLAB Calibration Toolbox are $cam = (397.2, 398.6, 157.3, 132.9,$
$-0.262, 0.169, 0, 0.001, -0.0005)$.

As the choice of feature types we selected SIFT and SURF. Both seem to posses the qualities we need. We did not try to tweak the parameters in any way except the SURF threshold to ensure that both detectors generate between 100 and 200 features per frame. We only select the top 100 features from each frame with the largest min-radius. The experiments were implemented in MATLAB and in C++. The C++ routines depend heavily on the OpenCV [105] and VLFeat [106] libraries.

In Figure 6.7 we show the objects used for baseline and regular evaluation. The book is used for baseline experiments. The tags were selected to have a variety of geometric patterns. The hope was that with the experiment results we would get a better understanding of what type of geometric patterns are most suitable. All video sequences are are converted to gray scale before feature extraction.

6.4.2 Depth Recovery Validation

Our first experiment evaluated how various depth recovery methods perform (Figure 6.8). For comparison we selected five different methods plus a random method (hypVoid). The random method returns a random pose and delimits the region of good results. The other methods include the EPnP method [75] run globally and within RANSAC (sampling only four points). The other three methods use the same three point depth recovery [71] followed by different alignment methods: PXT,Arun and Horn [79, 77, 78]. Our augmentation to the 3-point depth recovery has been very effective in recovering true depth. The RANSAC algorithm returned the best of ten hypotheses. The EPnP was also run in global mode because it can handle any number of points. We also had access to the POSIT algorithm but its accuracy was worse than the

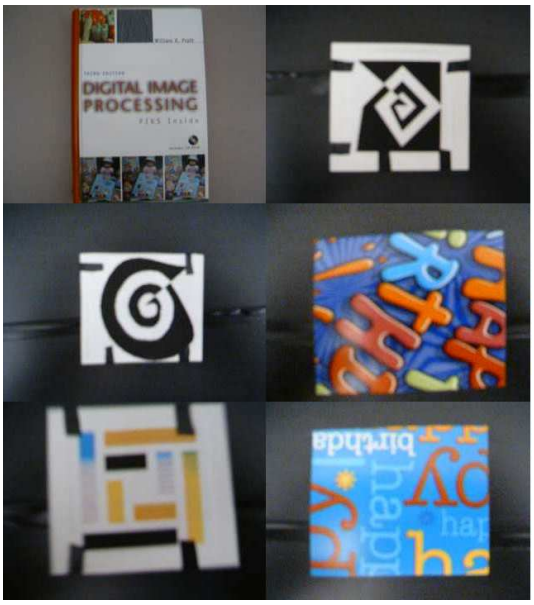

Figure 6.7: Selected objects (baseline + tags)

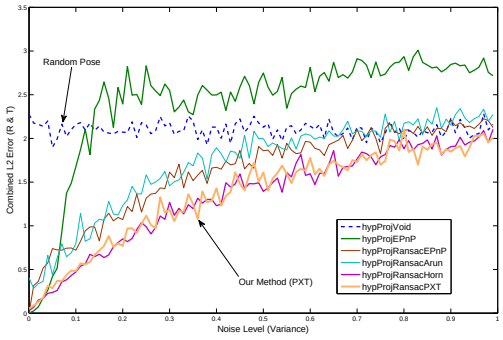

Figure 6.8: Projection error

others and it seemed to have convergence issues. It was dropped since it represents yet another global method. All methods were subjected to increasing amount of Gaussian noise. As we can see there is no systematic advantage of one method over the other but Horn's method appears more stable. The only surprising result is how sensitive global EPnP is.

6.4.3 Good Features to Track

For the clustering stage we collected exactly 100 features from each frame. For the book video sequence of some 500 frames the clustering algorithm has to process a database of 50000 features. The process is time consuming but if temporal and spatial constraints are added it finishes under an hour. The heavy computing needs at this stage are justified and we believe properly placed. The quality of selected clusters will significantly determine the accuracy of correspondence matching. For this step we have also developed a manual inspection utility that allows us to view individual clusters and to edit them(Figure 6.9). The utility provides a browser over clusters and clus-

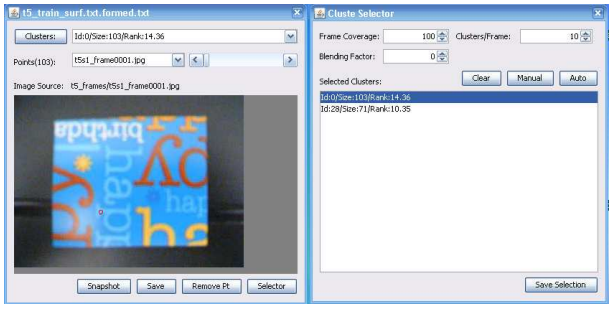

Figure 6.9: Manual inspection

ter members on the left side. On the right side it provides an interface for automated cluster selection. The goal is to select enough clusters to cover a significant number of frames with a certain ration of clusters per frame. The automated selection process is guided by the ranking or the quality of the cluster. The quality in turn is determined from two measures. The first measure is the size or number of member features. This measure reflects the frequency of a feature across frames. The second measure is the volume of the cluster and it facilitates robustness or certainty analysis. Together, these two measure can be used to rank the clusters and aid in selecting which ones will be part of the on-line model. We have selected several top clusters, ranked by frequency,

and embedded them in 2D for qualitative analysis. Figure 6.10 illustrates the degree of separability for three very likely SURF clusters. In Figure 6.11 we observe the same for SIFT features. In both cases the clusters were obtained from the book video sequence. What is apparent from the embeddings is that clusters exist

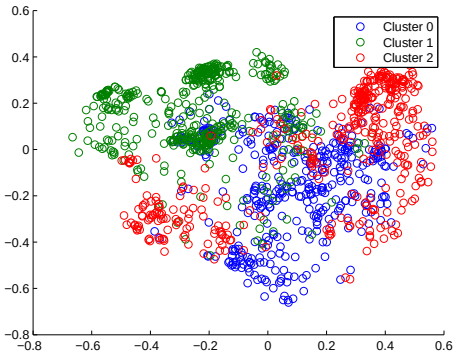

Figure 6.10: SURF feature distribution (embedded in 2D using MDS)

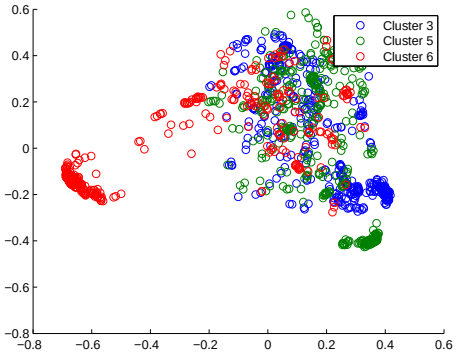

Figure 6.11: SIFT feature distribution (embedded in 2D using MDS)

but they overlap and they are not simply connected. Another observation which is not obvious but was observed is that one cluster might be tracking two actual points on the model. One frustrating observation which we made when clustering only by proximity of features and without spatio-temporal constraints is that the same cluster would track one point for about 100 frames perfectly then jump and continue tracking another point for another 100 frames, again perfectly. Since our clustering scheme grouped features by their closeness this indicates that both SIFT and SURF could be producing similar

features for fundamentally different image characteristics. With added constraints on how much a pixel is allowed to move the situation improves somewhat but outliers are still present. In Figure 6.12 we show an almost perfect case observed during our main experiment.

Figure 6.12: Well behaved cluster (4/200 frames)

6.4.4 Tracking Validation

We present here the results of our vision system. After performing clustering and selecting enough clusters to cover every frame we acquire a 3D model representation. This 3D reconstruction and model acquisition step proved tricky and the most unstable part of the system. The problem was finding clusters that overlap over a number of frames. With this information we then performed full tracking for the four sample tags in Figure 6.7. In the case of tags we collected four independent (different backgrounds) video sequences, each lasting for about 200 frames. One sequence was used to find good features and recover the 3D model shape and the others were used for validation. The tags are numbered in increasing order from top to bottom. The tracking quality is measured by the re-projection error. In Table 6.1 and 6.2 we present the results for different scenarios. For each scenario we present two errors (A/B). One is the error during projective recovery (A) and the other is the average error using recovered shape (B). With the different scenarios we tried to asses how various parameters affect the

performance. The three scenarios LOW,MED,HIGH represent number of features we used per frame. In all three cases the feature detectors were initialized to generate at least 100 features then for tracking and shape recovery we would choose the best 5,10 or 20 clusters and evaluate error rates.

Table 6.1: Tracking error

	Tag 1(A/B)	Tag 2(A/B)	Tag 3(A/B)
SIFT/LOW	0.11/14.40	0.100/10.3	0.15/15.3
SIFT/MED	0.13/10.30	0.078/8.6	0.17/14.1
SIFT/HIGH	0.098/7.60	0.081/6.1	0.14/12.4
SURF/LOW	0.14/15.3	0.11/13.5	0.16/16.3
SURF/MED	0.14/12.1	0.12/10.7	0.19/15.4
SURF/HIGH	0.12/10.2	0.13/8.9	0.17/13.3

Table 6.2: Tracking error continued

	Tag 4(A/B)	Tag 5(A/B)
SIFT/LOW	0.18/24.10	0.19/23.8
SIFT/MED	0.17/20.8	0.17/21.6
SIFT/HIGH	0.16/19.50	0.18/18.50
SURF/LOW	0.20/25.3	0.21/24.2
SURF/MED	0.21/22.3	0.19/21.7
SURF/HIGH	0.18/20.7	0.17/19.6

The error rates suggest that the first three tags (Figure 6.7) perform somewhat better. The last three tags are colored but the color is not used and their contrast might be lower. The first and second appear similar. In fact similar process was used to generate them. Overall the tracking error is a bit higher than we expected but that can be attributed to outliers which the clustering could not remove or avoid.

6.5 Conclusion

Based on the findings here we can see which parts of the vision system are more fragile than others. Due to the nature of the SIFT and SURF features we had to rely more heavily on projective reconstruction than anticipated. This part still remains a weakness because features do not show up in every frame. Also we have observed that the clusters formed with additional constraints tend to be smaller but more reliable (tracking only one point). Using a single radius might have been too simplistic.

Parts of the system that are needed for embedding have been migrated to C++ as part of a ROS (Robot Operating System) package. The biggest issue has been to maintain the accuracy as we move from double precision in MATLAB to more realistic single precision. Especially troublesome has been the three point depth recovery[71]. It seems to have suffer from numeric stability issues and the fact that it relies on root finding is also problematic.

Before the system is fully integrated into the scrub nurse robot we will need to address above issues and further refine our tag pattern design to incorporate findings made here. Finally we will need to conduct a series of experiments during real operations with our tags attached to determine how they stand up to the stress.

Chapter 7

Spoken Command Interface

7.1 Introduction

Robot assistants, like the robot scrub nurse (RSN), need to be as non-intrusive as possible. This is a challenging proposition in two ways. On one side, the purpose of the robot is to reliably and more efficiently replicate human task execution. From the other side robot assistants replace human activity which makes their acceptance rate dependent not only on feasibility but also on ease of use. Unlike industrial settings we should not assume a specially trained operator. Instead the machine must undertake a larger role in facilitating seamless interaction. The capacity to manage basic human dialogs would allow hands-free interaction between the robot and the operator and it would go a long way towards making deployment of robot assistants more successful.

Recent advances in processing speed and our understanding of speech recognition have led to recognition systems that are ready for commercial applications. One very prominent commercial application is that of call routing or automated answering systems. At first glance such a system would seem adequate for our needs as well. In both cases we are dealing with a known limited vocabulary and limited corpus of possible sentences. We could define the corpus after consulting a surgeon and use it to compute bi-gram and tri-gram word occurrence probabilities. Equipped with a language model, available systems such as CMU's Sphinx II could be deployed to recognize the speech coming from a mobile microphone. This idea sounds reasonable and can be set up very quickly.

Unfortunately, at the latest when everything is set up, one realizes that processing the incoming stream of words is problematic. Since we are considering an application as vital as assisting a surgical process, it demands special consideration. We are faced with the following concrete problem: the recognized word will not necessarily be the same as the uttered word all the time. For one, background noise will either corrupt the recognition process or generate a phantom word when none was uttered. In addition, the speaker might have an accent which can throw off the recognizer. The underlying speech recognizer is a time-series classifier. That means the accuracy of the speech model could be affected if we try to accommodate or train for additional accents. Finally even if the words are perfectly recognized, a human will frequently start a sentence and switch to a better formulation in mid-air. While such repairs are barely noticeable to a human listener, a computer system might get very confused.

Taking all of this into account we need to design a robust dialog manager which needs to analyze incoming recognized words, validate them against grammatical rules for correct speech, and also validate them against the task model for our particular application. If indeed a mistake was made by the recognizer, the sentence structure might be violated. Such mistakes will be detected by comparison with the grammar of our sentence corpus. In the rare event that the mistaken word is still part of the same type (verb, noun, etc.) the grammatical analysis might fail us. For the purposes of a scrub nurse robot this represents a catastrophic failure. It means that a plausible sentence has been detected which is wrong semantically. In that case the last line of defense would be the comparison with the possible commands and tasks that are executable from the present context or state in which the robot is. The dialog manager must be able to detect such inconsistencies and either recover (if speaker tried to correct himself or a similar sounding word is possible due to an accent) or reject the query from the user entirely. We present our work on a natural language interface for a robot scrub nurse that would provide instruments to a surgeon in otolaryngology. After describing the language processing pipeline we evaluate it for accuracy. In the rest of the paper we review available work in the area and compare it to our approach. After that we will describe our method,its innovative aspects, and its implementation/evaluation. At the end, we summarize key details of a language interface for robot assistants in life critical settings.

7.1.1 Key Contributions

Our work introduces two novelties. First we present an iterative CYK algorithm to improve accuracy by considering syntax. After that we also design a suitable dialog manager and chart various modelling issues that arise in the setting of an RSN. One of the results suggest that the dialog system should be designed to anticipate the catastrophic cases instead of trying to achieve perfect accuracy.

7.2 The Environment

For language processing we envisioned the surgeon wearing a bluetooth headset. Since the surgeon could talk to any of the staff, the microphone might pick up a variety of sounds:

- Generic background noise,
- Background conversation,
- Surgeon voice not directed to the scrub nurse, and
- Surgeon voice directed at scrub nurse.

We reasoned that undesired background speech and noise could be suppressed to some degree by a voice activity detector on the surgeon's neck. However determining if the surgeon is addressing the nurse would still need to be made by the speech interface.

7.3 Setup

As indicated our primary goal is the development of a hands-free interface for an assistive robot. The overall system is depicted in Figure 3. Concerning the interface we had to make several decisions. First on the hardware level a bluetooth headset was picked for extracting speech and delivering synthesized response. The headset is wirelessly

attached to a Pentium III computer with 384MB of RAM. The operating system is real-time Linux primarily because we have to control a PUMA robot but it also has advantages for the speech subsystem as well. Next we opted for off-the-shelf speech recognition software. The SPHINX-II/POCKETSPHINX was selected because it is open source and geared towards embedded applications.

The overall system depicted in Figure 3 is implemented as a modular package for the Robot Operating System (ROS) infrastructure [93]. Each processing component, sensor or actuator are separate modules called nodes which share the same system bus exchanging information via event dispatching or remote method invocation (services). The ROS architecture is particularly suitable for distributed computing and robotics and standardizes many interfaces to enable greater collaboration in the research community. On top of our application is a central thread responsible for message distribution and coordination between the nodes. Each node, including the discourse manager, is addressable and running in a separate thread. Even though we envision a spoken language interface as a primary form of communication, prudence dictates a fail safe mechanism. To that end the manual override is also part of the discourse manager. However it really bypasses it and acts more as a command terminal with a view into the kernel and the ability to issue direct commands.

At the platform level our goal is to generate reliably and correctly events going from the discourse manager towards other subsystems and back. In Figure 7.1 we illustrate the internals of our speech pipeline and highlight the areas of interest in this study. On the speech synthesis side we employ ready-made software called FLITE which is used unmodified. Before feeding audio data into POCKETSPHINX we perform noise suppression followed by Voice Activity Detection (VAD). Noise suppression is there to reduce systematic background noise (unlike human chatter). The voice activity detection is used to segment speech into utterances along silence points.

7.4 Parsing and Syntactic Analysis

The output of the speech recognizer will not always be correct. The recognizer gets easily confused by background noise and inadequate acoustic (per phoneme) or language models (per word or sentence). The result is a sequence of observed words or tokens

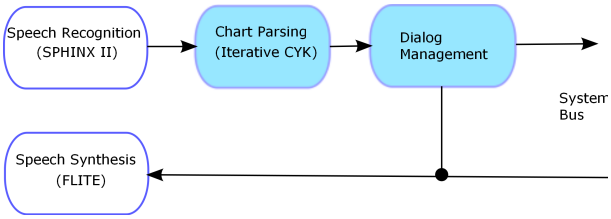

Figure 7.1: Speech pipeline

that are in our vocabulary (and hence valid) but might not be what we actually said. To counter this problem we attempt to parse the sequence into a valid sentence. Often per word errors will make the sentence syntactically unsound.

The two most widely known parsers are the Early parser and the Cocke-Younger-Kasami (CYK) algorithm. From the available literature one easily gets the impression [107],[108] that the CYK is less suitable for online or incremental parsing. On the other hand, as a bottom-up parser, CYK is clearly appealing in terms of evidence based analysis and manageability. Because of that we have implemented an online version of the CYK algorithm. Figure 7.2 depicts how the algorithm works.

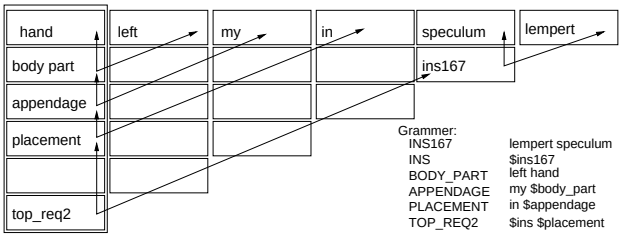

Figure 7.2: Incremental CYK parse snapshot

In essence the CYK is a dynamic algorithm which enumerates all possible word combinations to find possible parses acceptable by the given grammar. The grammar, which is a set of production rules, must be in Chomsky normal form because at any point the CYK only compares two tokens or cells. As depicted we organize the bottom up parsing by aligning the pyramid to the left. This way it becomes clear that any time a new word or token is discovered only the left most cells are affected as far as the possible parses are concerned. With this we obtain the incremental parsing scheme in Figure 7.3.

```
foreach token in (@tokens){
        forget(@right_most_cells)
        shift_right(pyramid)
        token2pyramid(row←0,column←0)
        foreach cell in (@left_most_cells){
                @pairings←all_cyk_pairs_from(cell)
                foreach pairing in (@parings){
                if(pairing in grammer) add_rule(cell)
                }
        }
}
```

Figure 7.3: Incremental CYK algorithm

In this scheme we store recognized rules in nodes which form a chain. Each cell of the parse pyramid points to nothing or to one such chain. As we are shifting to the right all we have to do is shift the pointers. Also if the pyramid is allocated large enough, we do not need to shift the pyramid but only the part being used. Each time a sentence end is encountered, the pyramid can be flushed by destroying or "forgetting" all attached nodes. Since the grammar rules can contain references to other rules plus raw word tokens the first row in the pyramid must hold the raw tokens and then any additional rules derived from them.

The ultimate goal of the parsing is to detect the top most rules which describe the structure of the sentence. If such a rule is encountered, we have a possible explanation for the given observations (tokens). Since the recognizer tends to confuse words, we can extract from our corpus of task related sentences confusion probabilities $P(word_i|token_i)$. In addition we can measure the relative frequency of occurrence for all of the rules in our grammar and use these probabilities to place weights at each cell in the tree. This would give us a probabilistic CYK parser. Such weighting could improve recognition performance for different sets of instruments in different surgeries. For now we have only made use of word confusion probabilities. In effect we assume a joint probability between the observed and the really uttered sentence to be:

$$P(S,O) = \Pi_i P(word_i|token_i) * P(O).$$

At any point in time we could estimate the most probable sentence uttered by maximizing the aforementioned joint probability over all sentences "S" that have been successfully parsed as evidenced by the recognized top level rules. For that we also need

the prior probability "$P(O)$". Alternatively we could avoid the need for the prior if we use pairwise comparison. Unfortunately as the pyramid shifts to the right, the number of tokens making up our observation "O" changes and hence our prior is changing. As a consequence we make use of a heuristic to track the most likely parse that explains the observation. First we always pick the top level rule with the highest row number in the pyramid. This way we pick sentences that explain more of the observation. Within the same cell we pick the one rule with the highest probability. Within a cell the prior "P(O)" does not change and a pair-wise comparison is valid. With the ability to track the best sentence parse we are able to consider the next step which is semantic analysis.

7.5 Meaning and Semantic Analysis

Parsing of incoming speech will check the syntax or sentence structure. As we will see that alone filters out many errors but it does not catch all. Namely since a token has a confusion probability we must consider all the alternatives and that leads to an entire forest of possible parse trees. It is possible that an alternative parse is found which only partially covers the meaning of the original uttered sentence or not at all. If the meaning is preserved partially we might be able to recover by using context to infer missing information or we might ask the user for clarification.

One particular example is a sentence such as:

```
[SURGEON]: give me senn retractor
```

With visual context (position of the left and the right hand) considered the actual event sent to the robot subsystem should be:

```
[+VIS CTX]: give me senn the retractor in my left hand
```

This example shows the purpose of a discourse or dialog manager. To infuse the manager with meaning, we must be able to model it in some way. By modelling meaning we capture various contexts or states in which the system resides. This narrows down further what the user really wants and at the same time maps the decision points at which we could ask for clarification.

Our proposed approach uses finite state automata (FSM) to model all of the task knowledge. For the robot scrub nurse we have identified a number of contexts which describe different parts. The task knowledge of the overal robot is illustrated in Figure

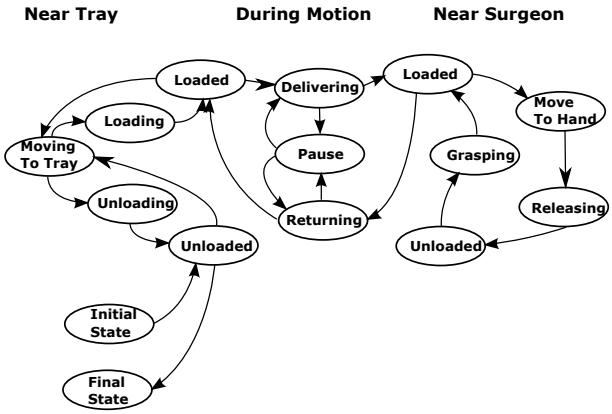

Figure 7.4: Robot task knowledge model

7.4. The robot FSM tracks the state of the robot arm. Some of the states near the surgeon and near the tray overlap with states of the grasping manipulator. We only present the states of the grasping manipulator near the surgeon here. They are shown in Figure 7.5.

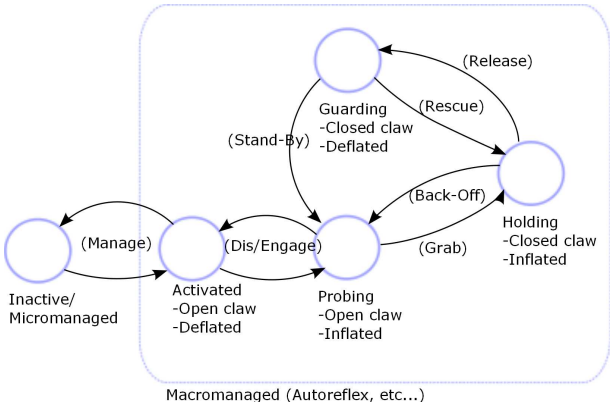

Figure 7.5: Gripper knowledge model

Our grasping manipulator has two fingers and uses an inflatable membrane to generate shape conforming grasps. This feature enables us to monitor pressure variations on the membrane and affect behaviour local to the gripper. Stated differently, the gripper controller has a limited autoreflex that causes state transitions without input from base software.

The robot and grasping manipulator are guided by visual servoing. A simplified

visual context is displayed in Figure 7.6. It only models certain useful states near the surgeon. The visual context is only one of possibly many sensor derived contexts. The sensor derived contexts are problematic because often their output is continuous (position and orientation) and discrete modelling via FSM does not represent complete knowledge.

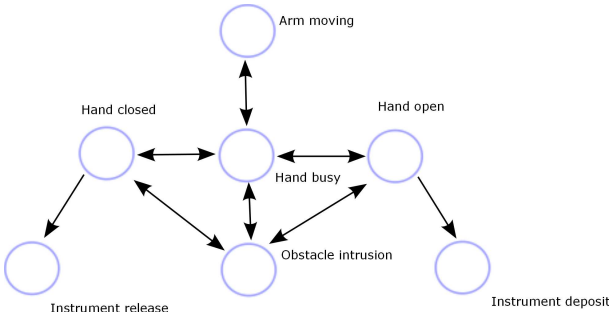

Figure 7.6: Visual context

Two contexts which are tightly related to robot scrub nurse are shown in Figure 7.7 and Figure 7.8. The workflow chart in Figure 7.7 is a crude capture of initial steps we observed during one live surgery. Clearly, the instruments and work of a RSN depends on the stage of the workflow. Furthermore the workflow is likely to change from procedure to procedure. Having a generic way of representing workflows allows us to record and recall procedures on demand. On the other hand, the execution plan in Figure 7.8 shows the structure of the actual code that is executed based on a user spoken command. Spoken language is more expressive than programming languages and one utterance could translate into one or more machine instructions. In a way, the mapping from natural language to machine code is a continuous cycle of template assembly, code compilation and debugging. We can see that a template mirrors the structure of a function. The atomic code resides in the nodes which are connected in a graph that indicates dependency on local and global variables and direction of code execution. Please note that the template code can result in orderly termination, exceptional termination or enter a limit cycle which should be interruptible. An exception created during dialog processing is also a context and is more expressive than traditional language exceptions.

A few contexts have been illustrated here and additional are possible. Each context is relatively independent which means the true state of the RSN is a tupple consisting

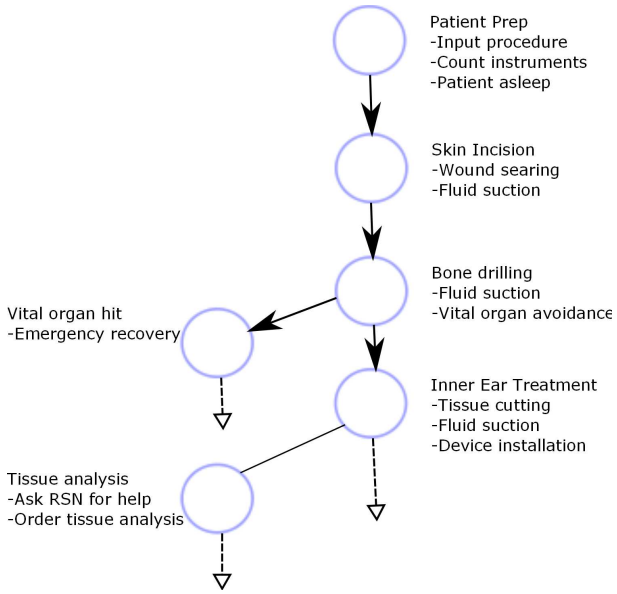

Figure 7.7: Surgical workflow

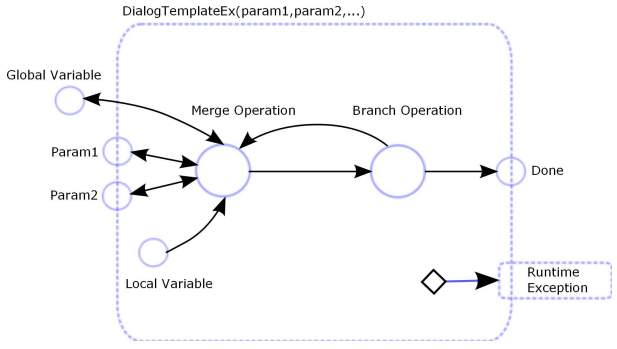

Figure 7.8: Execution plan

of all of them. Dialog manager needs to be able to integrate all of these contexts and make them available to resolve ambiguity or gather missing information. In Figure 7.9 you can see a diagram of our proposed manager.

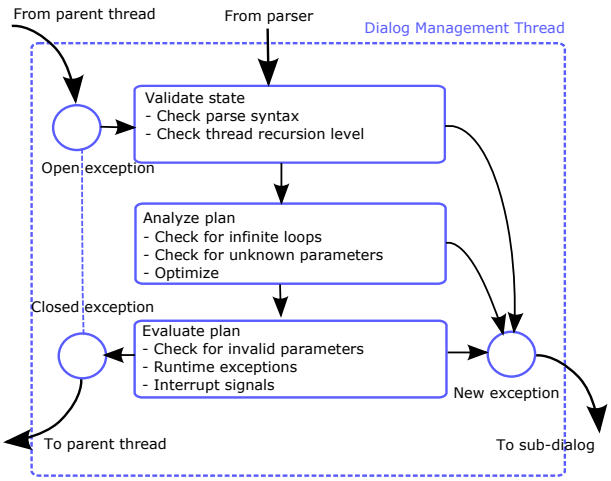

Figure 7.9: Dialog manager

Our dialog manager is recursive in nature. An exception thrown at any point during dialog processing pauses current conversation thread and leads to a new conversation thread with intent to gather additional information so that the exception can be closed and processing can resume. The processing that takes place is divided up into three stages. First we validate state and check input. Then, based on input one or more templates are recalled and linked together. Graph analysis at this point can uncover missing parameters or unresolved dependencies or infinite loops. The final stage is the execution of the code. Besides guarding against exceptions this stage must also react to abort signals. When completed, the final stage will return back to previous conversation and resume there or it will lead to a sub dialog for futher clarification.

7.6 Data

For the purposes of below evaluation and real time operation the audio data in our system is recorded at a rate of 16kHz. The input into the recognizer are chunks of audio samples no longer than 1sec. The recognizer is running in batch mode and uses a specially trained language model of word co-occurrence probabilities for our task. The

acoustic model used is "ws1_5k" directly from the CMU repository. The language model is trained from a corpus of 1064 sentences with 158 unique word tokens. The corpus discusses 94 instruments which were compiled from a catalog provided by Dr. Levine. These instruments are used for surgical procedures in Otolaryngology. The grammar for the dialogs lists about 205 distinct instruments based on that list (i.e., from lempert speculum we get speculum as well).

7.7 Evaluation

We have gone to some length to explain the details of the parsing algorithm. Of interest is in particular the resource usage. During our evaluations of the above corpus we found that the average width of the parse pyramid is 7 tokens with the worst case being 23. The average number of nodes used to construct the forest of parse trees is 14687 while the average number of nodes used to copy out top level rules into queries was 718. However the worst case numbers for the number of nodes for parse tree and query stack are 6907157 and 292585. Given that we are trying to build an embeddable system, these worst case numbers deserve further explanation. It turns out that only 1/1064 sentences needs that many nodes. The problem is that we did not limit the width of our search beam. We would start out at row 0 with the observed tokens and all possible confused words. Then continue the all against all matching until the top of the parse tree. Limiting the width to only the K most likely choices at each step is not uncommon in alternative studies. This would address the problem but might impact to some extent our reported results.

We first examined the speech recognition performance of the SPHINX-II out of the box. The hope was that if it performed well enough, then the human robot interface would be one less headache for our robotic scrub nurse. These results are presented in Table 7.1. We modified the well known edit distance (Levenshtein distance metric) function to operate on tokens instead of characters. This metric would return 0 if two sentences match exactly and if they don't we could recover an alignment between the two. The first row in the table reports the edit distance for the entire sentence. The second row is optimistic in the sense that we checked if the correct sentence was a substring of the recognized sentence. We also looked at raw word recognition rates.

First, the results were obtained by uttering a word by itself. Since the SPHINX-II recognizer uses word bi-gram and tri-gram frequencies, it is to be expected that the word error rate will go down if a word is uttered within a context of the sentence. That is confirmed by the last row.

Table 7.1: SPHINX-II/POCKETSPHINX Recognition performance

Full Sentence, Exactly	47.18%
Full Sentence, Optimistic	52.25%
Words, Isolated (Best Of Five)	59.49%
Words, Withn Sentence	86.14%

Next we examined the impact of enforcing syntactic constraints by means of context free grammar and CYK parsing. These results are shown in Table 7.2. Here again we used the edit distance function. The results for the word recognition rate are slightly better but it has a major impact on our ability to recognize the full sentence. We include in our reports the optimistic measure because one might be able to deal with the clutter around the sentence at the time of event generation but it is not significantly higher. This time we also report what kind of errors the edit distance produces. Apparently half of the time when it makes a mistake, it does so because a "phantom" word was recognized. During the recoding we could observe one particular case: If our model only contains the word "scissor" but we mistakenly utter "scissors", it will sometimes recognize "scissor six" or similar shorter "phantom" word.

Table 7.2: Role of the CYK parsing on recognition performance

Full Sentence, Exactly	77.74%
Full Sentence, Optimistic	80.18%
Words, Correct Match	89.28%
Words, Incorrect Substitute	4.30%
Words, Incorrect Delete	5.67%
Words, Incorrect Insert	0.74%

Finally we asked how many of the 237 incorrectly matched sentences were still recoverable. With recoverable we mean how many of them preserve the partial or full

meaning of the correct sentence even if there is something missing. For example we could observe that in some cases our parser would not recover everything. A sentence such as:

```
give me brown adson forcep in my left hand
```

would be parsed out as:

```
adson forcep
```

or even:

```
i need adson forcep
```

The results are provided in Table 7.3. After manually going through the sentences we could establish that 63.7% of the sentences still preserved enough meaning to allow clarification or immediate execution with later adjustments to motion. The remaining 86 sentences conveyed syntactically correct but semantically incorrect information. Firing an event to the robot from such sentence would be possible and could lead to catastrophic results. There is a chance that such sentence would not make sense within the current context but there is still a chance that it might. Therefore regardless of what the dialog manager recognizes an informative response needs to be sent back to the user. Each time an event is to be sent, the user should hear what the robot is about to do. The dialog manager should think out loudly.

Table 7.3: Role of the semantic analysis on recognition performance

Incorrect Sentences from Parsing	22.26% or 237 out of 1064
Meaning Preserved Fully	32.06%
Meaning Preserved Partially	31.64%
Meaning Lost	36.28% or 86 sentences

7.8 Conclusion

In this chapter we have presented our experience during the implementation of a Human-Robot Interface (HRI). The setting under which such an interface must operate is determined above all by the risk. The interaction of the discourse manager with the video

subsystem can function even if a valid event with incorrect meaning is sent but the same must not happen when the robot is concerned. In assistive robotic applications these mistakes will most likely lead to delays but technology tends to evolve and a mistaken command to shut down or power up could be vital even though the surgeon still has full control.

When Human Subjects approvals are obtained we can perform more comprehensive tests of the system in a real operating room. The conditions during which the test speech was recorded were such that various systematic noises were present (air conditioning, washing machine, broken freezer fan) but no background chatter. In the operating room two most obvious sources of noise will be background chatter coming from other individuals and surgeon's instructions not directed at the scrub nurse. The background chatter is minimized by the close proximity of the microphone to the surgeon and our noise suppression but there still remains the problem of surgeon's instructions. To suppress this noise our corpus would need to be augmented with negative samples or sentences which we should ignore.

Chapter 8

Summary

In this study we looked at three interface modalities for a robot scrub nurse. A robot scrub nurse is a robot assistant. As already explained a robot assistant is different from an enabling or active robot because it does not stand between the surgeon and the patient. This is an important difference with implications for risk assessment and regulatory approval. In our review of economic and social factors affecting adoption rates of medical robots we have found that prior exposure to technology makes further adoption more likely. Coupled with the fact that robot assistants do not require same regulatory approval as enabling robots our work charts a less expensive way to enter the medical robotics market. This side observation should be valuable to anyone who is considering such a move.

Several factors determine if a robot solution will be accepted in the operating room. One of them is certainly ease of use and our work contributes novel work in that area by studying interfaces. Our choice to pick three interface modalities and study them in more detail was borne out of field observations. We have contributed novel research on all three interface modalities. Our research gains most credibility when it is taken together in the context of robot scrub nurse development.

8.1 Our Contribution

In our attempt to develop a working robot scrub nurse we realized early that there would have to be a trade off between how much research and how much development would take

place. We decided that novel research had precedence over engineering. Consequentially our prototype robot scrub nurse is only a proof of concept. Still we have produced a software package that should be of significant help to any future work. Our software is compatible with the robot operating system(ROS) and is therefore available to a wide audience of researchers who contribute to ROS. Our software is important because it shows how all the components integrate together even if they are not production ready.

As explained through our observation of a real-world operation we identified three interface modalities of particular significance: grasp planning and haptics, natural language processing and visual servoing.

For grasp planning and haptics we developed a grasping method based on pairwise shape descriptors. To facilitate grasp planning we needed a fast object alignment method that would work for partial meshes as well. Our alignment method is comparable to existing methods but requires fewer points and no matrix decomposition. It is based on cross products. In conjunction with the grasp planning problem we realized that common approach using hard contact modeling was a weakness. While we have not been able to move to soft contact modeling at the mathematic level we have designed a special grasping mechanism that generates soft grasps. Using an inflatable membrane our gripper generates shape conforming contact with the target and the pressure within the membrane can be used for feedback processing. This added benefit allows us to interpret haptic signals and communicate with the user via touch.

To guide the gripper and robot arm we investigated visual tracking. The human hand is not the easiest object to track because it is deformable and composed of multiple rigid bodies. To simplify our problem we designed special tags that would be unobtrusive to the surgeon and would allow us to identify them in an image and also to recover its position and orientation. To support such model based visual tracking we also had to develop special methods to cluster good features for tracking and to extract 3D models from 2D images with missing correspondences.

Our work on speech recognition and synthesis tried to use off-the-shelf components. That was not entirely possible. We designed an iterative CYK parser to extract syntactic information while running in an embedded environment.

8.2 Robot Scrub Nurse in Action

In addition to the various theoretical and practical contributions we also created a prototype robot scrub nurse. The prototype robot represents a vehicle to integrate all of the other parts and to demonstrate proof-of-concept.

In Figure 8.1 we show the robot scrub nurse in its initial position. It consists of a PUMA 560 robot whose actuator is replaced by our own shape conforming gripper. On the side of the robot arm we have attached the micro-controller box. Both the robot arm and the micro controller box are controlled by our collection of ROS nodes. The ROS nodes are largely hosted on a powerful laptop that can handle speech, vision and other processing demands. The ROS node controlling the robot is connected via Ethernet and runs on a dedicated machine with hardware connected to the PUMA control panel.

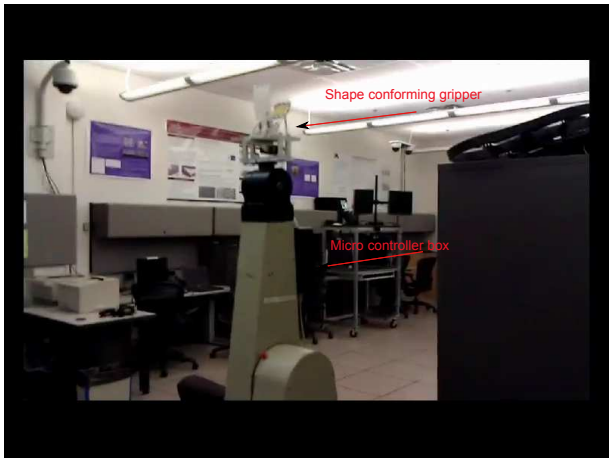

Figure 8.1: Intial arm position

As can be seen in Figure 8.2 we have assembled a mock-up operating room around the robot arm. In particular we have a wooden scaffolding that holds two tables imitating the place where the patient is located and where the instruments are located. The table in view is for instruments and lying on it is a makeshift instrument tray that can hold one instrument. The instrument tray is a simple holder that makes it easy for the robot to load and unload instruments. The other table where the patient would be located will be seen in later figures and is oriented at 90 degrees.

A sample run of the robot scrub nurse will move the robot arm in a position above

Figure 8.2: Arm above tray before loading

the tray before loading the instrument. The next step is show in in Figure 8.3. The arm together with the gripper is actuated to close around the instrument and to grasp it. This process involves heavy use of finite state machines to coordinate motion of the arm with that of the gripper. The whole task is simplified because the location of the tray and the instrument can be hard coded. As a result the movement to load the instrument can be precomputed and recalled when necessary. We should note that our main ROS node is a task scheduler that can handle various coordination event. The arm motion node works like a scripted interpreter and can therefore store and replay motion. This is a useful feature for exactly this situation.

Once the instrument is grasped it is delivered in the vicinity of the user and presented (Figure 8.4). At this first version of the prototype the robot does not attempt to track the user's hand.

The next step in Figure 8.5 involves haptic processing. Here the user attempts to take the instrument and the robot must detect that and release its hold. Using measurements from the inflatable membrane that is also what happens. This stage involves physical interaction between user and machine. After several runs the plastic material of the gripper showed signs of stress. Part of the upper jaw broke off and had to be reinforced.

Finally in Figure 8.6 the robot has accepted the instrument back and is delivering it to the tray. Here too the robot scrub nurse needs to detect the exact time when to

Figure 8.3: Loading instrument from tray

Figure 8.4: Offering instrument to user

Figure 8.5: Taking instrument from user

accept the instrument. It completes one processing cycle by delivering the instrument to the tray and unloading it. The robot scrub nurse then returns to its initial position awaiting next command.

Figure 8.6: Unloading instrument to tray

8.3 Remaining Issues

Our research has tried to focus on very specific parts of a robot scrub nurse. The most immediate topics which we did not have time to cover is the instrument server side of

the robot and the problem with obstacle avoidance. These topics are important because they would prevent our proof-of-concept from even a limited production deployment. The instrument server is a mechanism to store instruments and to load and unload them for actual use. The obstacle avoidance is an extension to the motion planning algorithm that is needed especially for robot movement between the instrument server and the surgeon's workspace.

Another limitation of our robot assistant is that it would not replace all responsibilities of a scrub nurse even if we had a production ready prototype. For example, the scrub nurse sometimes collects sample tissues from the surgeon for further testing. These secondary responsibilities were not considered at all even though we can imagine a robot assistant performing the duties of a testing lab much like the robot rovers on Mars.

In the field of grasp planning our method has produced useful results but it still has some shortcomings. Biggest of them is the search in DOF space for possible solutions. Here we are attempting to use machine learning methods to learn the mapping of DOF space since it has a highly non-linear topography. This problem manifests itself in our method being somewhat slower.

For visual tracking we have found that features have degenerate regions which creates problems in our clustering algorithm and still requires manual inspection of selected good features to track. In the next stage we also face an issue with missing points. Both of these problems cause our off-line learning stage to be still manual in some steps.

In the language processing interface we have found that perfect recognition is very difficult to achieve. Some heuristic rules have been examined but in the end this represents a risk factor.

Finally, we were unable to secure broader interest from industry or government funding agencies for this type of research. While we have a better understand for their reasons it represents a failure in some way.

8.4 Future Work

For the future we would like to overcome remaining issues of our work, including better success securing additional support and funding. Designing an instrument server that

would also be capable of testing tissue samples is a direction which might have better immediate success in the operating room. Further research in this direction would be warranted. As far as the gripper is concerned here too we could see more work done. On the mechanical side material selection is an important topic that deserves further analysis. On the signal processing side we think that more sensors would extend the capabilities of the gripper tremendously.

References

[1] G. S. Guthart and J. K. Salisbury. The intuitive telesurgery system: Overview and application. In *ICRA '00: Proceedings of the IEEE international conference on robotic automation*, pages 618–621, 2000.

[2] Bureau of Labor Statistics. Occupational outlook handbook: Surgical technologists, 2010. `http://www.bls.gov/oco/ocos106.htm`.

[3] James P Oberman and Craig S Derkay. Post-tympanostomy tube otorrhea. *American Journal of Otolaryngology*, March.

[4] A. Kochan. Scalpel please, robot: Penelope's debut in the operating room. *Industrial Robot: An International Journal*, 32(6):449–451, 2005.

[5] Mark W. Noakes, Randall F. Lind, John F. Jansen, Lonnie J. Love, Francois G. Pin, and Bradley S. Richardson. Development of a remote trauma care assist robot. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2580–2585, Oct. 2009.

[6] Juri Vain et al. Human-robot interaction learning using timed automata. *ICROS-SICE International Joint Conference*, August 2009.

[7] F. Miyawaki, K. Masamune, S. Suzuki, K. Yoshimitsu, and J. Vain. Scrub nurse robot system-intraoperative motion analysis of a scrub nurse and timed-automata-based model for surgery. *Industrial Electronics, IEEE Transactions on*, 52(5):1227–1235, Oct. 2005.

[8] F. Miyawaki et al. Development of automatic acquisition system of surgical-instrument informantion in endoscopic and laparoscopic surgey. In *Industrial*

Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on, pages 3058–3063, May 2009.

[9] Ming-Yuan Shieh, Cheng-Ming Lu, Chin-Chien Chen, Chen-Yang Chuang, and Yu-Sheng Lai. Design and implementation of an interactive nurse robot. In *SICE, 2007 Annual Conference*, pages 2121–2125, Sept. 2007.

[10] Frost & Sullivan. US Image Guided and Robot Assisted Surgery, 2008.

[11] Frost & Sullivan. Advances in Image Guided Surgery and Surgical Navigation. `www.frost.com`, 2008.

[12] Frost & Sullivan. Emerging Image Guided and Robot Assisted Surgery, 2006.

[13] Larry Leifer, George Toye, and Machiel Van der Loos. Tele-service-robot: Integrating the socio-technical framework of human service through the internet-world-wide-web. *Robotics and Autonomous Systems*, 18(1-2):117 – 126, 1996.

[14] J. Fleck. The adoption of robots in industry. *Physics in Technology*, 15, 1984.

[15] Paul A. Herbig and Fred Palumbo. The effect of culture on the adoption process: A comparison of japanese and american behavior. *Technological Forecasting and Social Change*, 46(1):71 – 101, 1994.

[16] James Young, Richard Hawkins, Ehud Sharlin, and Takeo Igarashi. Toward acceptable domestic robots: Applying insights from social psychology. *International Journal of Social Robotics*, 1:95–108, 2009.

[17] Robert D. Howe and Yoky Matsuoka. Robotics for surgery. *Annual Review of Biomedical Engineering*, 1:211–240, 1999.

[18] Jessica L. Webster and Caroline G. L. Cao. Lowering Communication Barriers in Operating Room Technology. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(4):747–758, Winter 2006, http://hfs.sagepub.com/content/48/4/747.full.pdf+html.

[19] The safety issues of medical robotics. *Reliability Engineering and System Safety*, 73(2):183 – 192, 2001.

[20] Cuccurullo D Settembre A Miranda N Amato F Pirozzi F Caiazzo P. Corcione F, Esposito C. Advantages and limits of robot-assisted laparoscopic surgery: preliminary experience. *Surgical Endoscopy*, 19:117–119, 2005.

[21] G.H. Ballantyne. Robotic surgery, telerobotic surgery, telepresence, and telementoring. *Surgical Endoscopy*, 16:1389–1402, 2002. 10.1007/s00464-001-8283-7.

[22] James F. Allen et al. A robust system for natural spoken dialogue. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 62–70, 1996.

[23] Amanda Stent, John Dowding, Jean Mark Gawron, Elizabeth Owen Bratt, and Robert Moore. The CommandTalk spoken dialogue system. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 183–190, 1999.

[24] Ioannis Toptsis Axel et al. Modality integration and dialog management for a robotic assistant. *INTERSPEECH*, pages 837–840, 2005.

[25] Hartwig Holzapfel. A dialogue manager for multimodal human-robot interaction and learning of a humanoid robot. *Industrial Robot: An International Journal*, 35:528 – 535, 2008.

[26] Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 93–100, 2000.

[27] Stanislao Lauria, Guido Bugmann, Theocharis Kyriacou, and Ewan Klein. Mobile robot programming using natural language. *Robotics and Autonomous Systems*, 38(3–4):171–181, 2002.

[28] L. Seabra Lopes and A. Teixeira. Human-robot interaction through spoken language dialogue. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2000.

[29] Donna K. Byron. Improving discourse management in trips-98. In *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech-99*, 1999.

[30] Charles Rich and Candace L. Sidner. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3-4):315–350, 1998.

[31] J. Chappelier, M. Rajman, R. Aragues, and A. Rozenknop. Lattice parsing for speech recognition. *Sixth Conference sur le Traitement Automatique duLangage Naturel*, pages 95–104, 1999.

[32] Tim Miller, Andy Exley, and William Schuler. Elements of a spoken language programming interface for robots. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 231–237, New York, NY, USA, 2007. ACM.

[33] Scott Miller, David Stallard, Robert Bobrow, and Richard Schwartz. A fully statistical approach to natural language interfaces. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 55–61, 1996.

[34] Peter A. Heeman. Modeling speech repairs and intonational phrasing to improve speech recognition. In *Automatic Speech Recognition and Understanding Workshop, Keystone Colorado*, December 1999.

[35] Eugene Charniak and Mark Johnson. Edit detection and parsing for transcribed speech. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–9, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[36] Deb Roy and Niloy Mukherjee. Towards situated speech understanding: visual context priming of language models. *Computer Speech and Language*, 19(2):227–248, April 2005.

[37] P. Gorniak and D. Roy. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470, 2003.

[38] Alan Carlin, Nathan Schurr, and Janusz Marecki. ALARMS: Alerting and reasoning management system for next generation aircraft hazards. Proceedings of Conference on Uncertainty in Artificial Intelligence, July 2010.

[39] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. pages 348–353, 2000.

[40] Maximo Alejandro Roa Garzon and Raul Suarez Feijoo. Grasp synthesis for 3d objects. *Technical University of Catalonia*, July 2006.

[41] Carlo Ferrari and John Canny. Planning optimal grapsps. *International Conference on Robotics and Automation*, May 1992.

[42] Brian Mirtich and John Canny. Easily computable optimum grasps in 2-d and 3-d. In *In IEEE International Conference on Robotics and Automation*, pages 739–747, 1994.

[43] Xiangyang Zhu and Jun Wang. Automatic grasp planning using shape primitives. *IEEE Transactions on Robotics and Automation*, 19, August 2003.

[44] Ch.Borst, M.Fischer, and G.Hirzinger. Grasp Planning: How to Choose a Suitable Task Wrench Space. *proceedings of the International Confererence on Robotics and Automation*, pages 319–325, April 2004.

[45] Guanfeng Liu, Jijie Xu, Xin Wang, and Zexiang Li. On quality functions for grasp synthesis, fixture planning, and coordinated manipulation. *IEEE Transactions on Automation Science and Engineering*, October 2004.

[46] R. D. Howe, I. Kao, and M. R. Cutkosky. The sliding of robot fingers under combined torsion and shear loading. *Proceedings of the International Confererence on Robotics and Automation*, 1:103–105, April 1988.

[47] Yanmei Li and Imin Kao. A review of modeling of soft-contact fingers and stiffness control for dextrous manipulation in robotics. *Proceedings of the International Confererence on Robotics and Automation*, 3:3055–3060, May 2001.

[48] Matei Ciocarlie, Claire Lackner, and Peter Allen. Soft finger model with adaptive contact geometry for grasping and manipulation tasks. In *WHC '07: Proceedings*

of the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pages 219–224. IEEE Computer Society, 2007.

[49] Ch. Borst, M. Fischer, and G. Hirzinger. Calculating hand configurations for precision and pinch grasps. In *In Proc. of the 2002 IEEE/RSI/GI International Conference on Intelligent Robots and Systems*, pages 1553–1559. IEEE, 2002.

[50] C. Rosales, J.M. Porta, R. Suarez, and L. Ros. Finding all valid hand configurations for a given precision grasp. *Technical University of Catalonia*, May 2008.

[51] E.Lopez Damian, D.Sidobre, and R.Alami. Grasp planning for non-convex objects. *36th International Symposium on Robotics (ISR'2005)*, December 2005.

[52] A.T.Miller, S.Knoop, H.I.Christensen, and P.K.Allen. Automatic grasp planning using shape primitives. *Proceedings of the International Confererence on Robotics and Automation*, 2:1824–1829, September 2003.

[53] Corey Goldfeder, Peter K. Allen, Claire Lackner, and Raphael Pelossof. Grasp planning via decomposition trees. April 2007.

[54] Matei T. Ciocarlie and Peter K. Allen. On-line interactive dexterous grasping. In *EuroHaptics '08: Proceedings of the 6th international conference on Haptics*, pages 104–113. Springer-Verlag, 2008.

[55] Dmitry Berenson, Rosen Diankov, Koichi Nishiwaki, Satoshi Kagami, and James Kuffner. Grasp planning in complex scenes. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids07)*, December 2007.

[56] Benjamin Bustos, Daniel Keim, Dietmar Saupe, Tobias Schreck, and Dejan Vranic. An experimental comparison of feature-based 3d retrieval methods. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 215–222. IEEE Computer Society, 2004.

[57] Marcin Novotni and Reinhard Klein. A geometric approach to 3d object comparison. In *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, page 167, Washington, DC, USA, 2001. IEEE Computer Society.

[58] Heng Huang, Li Shen, Rong Zhang, Fillia Makedon, Bruce Hettleman, and Justin Pearlman. Surface alignment of 3d spherical harmonic models: Application to cardiac mri analysis. *Medical Image Computing and Computer-Assisted Intervention*, September 2005.

[59] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

[60] M.Taylor, A.Blake, and A.Cox. Visually guided grasping in 3d. *International Confererence on Robotics and Automation*, 1:761–766, May 1994.

[61] A.T. Miller D. Kragić and P.K. Allen. Real-time tracking meets online grasp planning. In *ICRA '01: Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2460–2465, 2001.

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

[63] D.G. Lowe. Object recognition from local scale-invariant features. volume 2, pages 1150 –1157 vol.2, 1999.

[64] Motilal Agrawal, Kurt Konolige, and Morten Blas. Censure: Center surround extremas for realtime feature detection and matching. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision ECCV 2008*, volume 5305 of *Lecture Notes in Computer Science*, pages 102–115. Springer Berlin / Heidelberg, 2008.

[65] F. Chaumette. Image moments: a general and useful set of features for visual servoing. *Robotics, IEEE Transactions on*, 20(4):713 – 723, aug. 2004.

[66] Isaac Weiss. Projective invariants of shapes. In *Proceedings of the Computer Society Conference on Computer Vision*, volume 5, pages 291–297, June 1988.

[67] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[68] Vasu Parameswaran and Rama Chellappa. View invariance for human action recognition. *Int. J. Comput. Vision*, 66(1):83–101, 2006.

[69] Peter Meer, Doron Mintz, Azriel Rosenfeld, and Dong Yoon Kim. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6:59–70, 1991.

[70] R. M. Haralick, C. Lee, K. Ottenberg, and Michael Nölle. Analysis and solutions of the three point perspective pose estimation problem. Technical report, Universität Hamburg, Hamburg, Germany, Germany, 1991.

[71] S. Linnainmaa, D. Harwood, and L.S. Davis. Pose determination of a three-dimensional object using triangle pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):634–647, 1988.

[72] W.J. Wolfe, D. Mathis, C. Weber Sklair, and M. Magee. The perspective view of three points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):66–73, 1991.

[73] J.S.-C. Yuan. A general photogrammetric method for determining object positionand orientation. *IEEE Transactions on Robotics and Automation*, 5(2):129–142, 1989.

[74] Daniel DeMenthon and Larry S. Davis. Model-based object pose in 25 lines of code. In *European Conference on Computer Vision*, pages 335–343, 1992.

[75] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *Int. J. Comput. Vision*, 81(2):155–166, 2009.

[76] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Mach. Vision Appl.*, 9(5-6):272–290, 1997.

[77] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(5):698 –700, sep. 1987.

[78] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[79] Amer Agovic and Nikolaos Papanikolopoulos. Grasp planning by alignment of pairwise shape descriptorss. In *IROS*, pages 1797–1804, 2009.

[80] Jianbo Shi and C. Tomasi. Good features to track. pages 593 –600, jun. 1994.

[81] Nikolaos P. Papanikolopoulos. Selection of features and evaluation of visual measurements during robotic visual servoing tasks. *Journal of Intelligent Robotic Systems*, 13:279–304, 1995.

[82] F. Janabi-Sharifi and W.J. Wilson. Automatic selection of image features for visual servoing. *Robotics and Automation, IEEE Transactions on*, 13(6):890 –903, dec. 1997.

[83] D. Omercevic, O. Drbohlav, and A. Leonardis. High-dimensional feature matching: Employing the concept of meaningful nearest neighbors. pages 1 –8, oct. 2007.

[84] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004.

[85] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–11, 2009.

[86] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[87] Qizheng Sheng, Yves Moreau, Frank De Smet, Kathleen Marchal, and Bart De Moor. Advances in cluster analysis of microarray data, 2005.

[88] M.H.C. Law, M.A.T. Figueiredo, and A.K. Jain. Simultaneous feature selection and clustering using mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1154 –1166, sep. 2004.

[89] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 596–609. Springer-Verlag, Berlin, Heidelberg, 2008.

[90] Tomás Svoboda, Daniel Martinec, and Tomás Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence: Teleoper. Virtual Environ.*, 14(4):407–422, 2005.

[91] Vincent Rabaud. Vincent's Structure from Motion Toolbox. `http://vision.ucsd.edu/~vrabaud/toolbox/`.

[92] John Oliensis and Richard Hartley. Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2217–2233, 2007.

[93] Willow Garage. Robot operating system, 2010. `http://www.ros.org/wiki/`.

[94] S. Kambhampati and S. Davis. Multiresolution path planning for mobile robots. *IEEE Journal of Robotics and Automation*, 2:135–145, September 1986.

[95] J.Y. Hwang, S.S. Kim, J.S.and Lim, and K.H. Park. A fast path planning by path graph optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 33:121–129, January 2003.

[96] Milos Seda. Voronoi diagrams and their applications. *ASR 2001 Seminarm,Instruments and Control,Ostrava*, April 2001.

[97] Howie Choset. Incremental construction of the generalized voronoi diagram, the generalized voronoi graph, and the hierarchical generalized voronoi graph. In *Proceedings of the First CGC Workshop on Computational Geometry*, October 1997.

[98] K.E. Hoff, J. Keyser, M. Lin, D. Manocha, and T. Culver. Fast computation of generalized voronoi diagrams using graphics hardware. In *SIGGRAPH 99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 277–286, 1999.

[99] S. Garrido, L. Moreno, and D. Blanco. Voronoi diagram and fast marching applied to path planning. *Proceedings 2006 IEEE International Conference on Robotics and Automation*, pages 3049–3054, May 2006.

[100] K. Harada, K. Kaneko, and F. Kanehiro. Fast grasp planning for hand/arm systems based on convex model. *IEEE International Conference on Robotics and Automation*, May 2008.

[101] U. Uenohara and Takeo Kanade. Geometric invariants for verification in 3-d object tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 785–790, November 1996.

[102] S. J. Maybank. Relation between 3d invariants and 2d invariant. In *VSR '95: Proceedings of the IEEE Workshop on Representation of Visual Scenes*, page 53, Washington, DC, USA, 1995. IEEE Computer Society.

[103] Peter Meer, Sudhir Ramakrishna, and Reiner Lenz. Correspondence of coplanar features through p^2-invariant representations. In *Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision*, pages 473–492, London, UK, 1994. Springer-Verlag.

[104] Guo Lei. Recognition of planar objects in 3-d space from single perspectiveviews using cross ratio. *IEEE Transactions on Robotics and Automation*, 6(4):432–437, August 1990.

[105] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[106] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[107] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[108] William Schuler Tim Miller, Lane Schwartz. Incremental semantic models for continuous context-sensitive speech recognition. In *Proceedings of the Workshop on Semantic Representation of Spoken Language (SRSL'07)*, 2007.

Appendix A

Glossary and Acronyms

Care has been taken in this thesis to minimize the use of jargon and acronyms, but this cannot always be achieved. This appendix defines jargon terms in a glossary, and contains a table of acronyms and their meaning.

A.1 Glossary

- **Visual servoing** – Vision-based robot control.
- **Laparoscopy** – Minimally invasive surgery using a thin, lighted tube called laparoscope.
- **Natural language processing** – Field of signal processing/computer science trying to understand human language.
- **Form closure** – An immovable state of a physical body caused by external forces and body friction at contact.
- **Versor** – Unit quaternion.
- **Hertzian model** – Contact model that explains how pressure, friction and force interact at a surface contact between two bodies. Used to measure hardness of materials.
- **Da Vinci** – Dominant robotic system for minimally invasive surgery.

- **Otolaryngology** – Branch of medicine that specializes in treatnebt of ear, nose and troat problems.

A.2 Acronyms

Table A.1: Acronyms

Acronym	Meaning
FSA	Finite state automata (same as FSM)
FSM	Finite state machine
GUI	Graphic user interface
HRI	Human robot interface
ICP	Iterative closest point
MDS	Multidimensional scaling
NLP	Natural language processing
OTC	Office of technology commercialization
PE Tube	Pressure equalizing tube (artificial eustachian tube)
POSIT	Pose from orthography and scaling with iterations
PXT	Pose by intersection
RANSAC	Random sample consensus
RF	Radio-frequency
ROS	Robot operating system
RSN	Robotic scrub nurse
SFM	Structure from motion
SIFT	Scale invariant feature transform
SURF	Speeded up robust features
UI	User interface