Automated Objective Surgical Skill Assessment and Visualization in the Operating Room Using Unstructured Tool Motion for Improved Surgical Training

by

PIYUSH PODDAR

A thesis submitted to Johns Hopkins University in conformity with the requirements for the degree of Master of Science in Engineering

Baltimore, Maryland

May, 2014

Abstract

Past attempts at surgical skill assessment using tool motion in the operating room have focused on highly-structured surgical tasks such as suturing. These methods considered only generic descriptive metrics such as the operating time and the number of movements made, which are of limited instructional value. In this thesis, we develop and evaluate an automated method of surgical skill assessment of flap elevation in nasal septoplasty in the operating room. The obstructed field of view and highly unstructured nature of septoplasty hinders trainees from efficiently learning how to effectively perform the procedure. Thus, we also present the development of a real-time visualization system that allows trainees and instructors to better observe tool motion with respect to patient anatomy during the operation. In this work, we propose a descriptive structure of septoplasty that consists of the following two activity types: (1) the brushing activity directed away from the septum plane that characterizes the consistency of the surgeon's wrist motion and (2) the activity along the septal plane that characterizes the surgeon's coverage pattern. We computed features related to these activity types that allow classification of a surgeon's level of training with an average accuracy of about 72%. Further, as opposed to previously-measured generic motion metrics, the presented features provide surgeons with personalized, actionable feedback regarding their tool motion.

ADVISOR:

Gregory D. Hager, PhD

Professor and Chair of Computer Science at Johns Hopkins University and the Deputy Director of the NSF Engineering Research Center for Computer Integrated Surgical Systems and Technology

ADDITIONAL READERS:

Sanjeev Khudanpur, PhD

Associate Professor in the Department of Electrical and Computer Engineering and the Department of Computer Science at Johns Hopkins University

Masaru Ishii, MD PhD

Associate Professor of Otolaryngology-Head and Neck Surgery and Associate Professor of Neurological Surgery at the Johns Hopkins Hospital

Acknowledgements

The presented work was, by all accounts, a group effort. My contributions to the project could not have been done without the generous support, patience, and help of a number of individuals. First, I would like to extend a thank you to Narges Ahmidi who has been working with me every step of the way. Narges balanced giving me the freedom to try new things with teaching me the foundations built by the prior work on the project. Her persistence, personality, and patience have made my time working on this project an absolute pleasure. I would also like to thank my PI, Professor Dr. Gregory Hager, for offering the high-level guidance and support needed to ensure the project stayed on-track and true to the goals of the overarching project. Dr. Hager's hands-off approach to mentorship allowed me to experiment with a number of ideas and enabled me to begin to learn what it takes to become a successful researcher.

I wanted to thank Dr. Swaroop Vedula for his arduous and tireless work in data collection. His patience and meticulousness when collecting live-patient data made my job many times easier. In addition, his insight and probing questions always forced me to go beyond the surface when answering my research questions.

I would also like to thank Dr. Lisa Ishii and Dr. Masaru Ishii for their clinical guidance and insight. Working with engineers, they always took the time to bring in the perspective of the patient and the trainee into discussions and were incredibly supportive of our on-going efforts. I am especially grateful for their support in bridging the gap between engineering and medicine.

Without their efforts, it is clear that this project could not go forward as quickly as it is today.

Thanks also to Anand Malpani for helping my transition into a new lab and for always being responsive and helpful when I faced trouble. Last, but not least, I would like to thank Dr. Sanjeev Khudanpur for always providing creative and insightful ways of solving difficulties I was facing. Dr. Khudanpur seemed to care deeply about how my work on the project was developing and his encouragement was very meaningful to me.

Beyond those individuals, I would like to thank the entire Language of Surgery and Computational Interaction and Robotics Laboratory groups for their ongoing support. Our weekly discussions have introduced me to a new world of technology that I would not have access to otherwise.

On a more personal note, I'd like to thank my friends and family and, in particular, my parents Jugal Babulal Poddar and Hema Poddar. They have been my role-models and a source of seemingly-infinite support. I could not have asked for anything more.

Contents

List of Tables

List of Figures

1 Prior Work

1.1 Surgical Training and Qualitative Assessment

Surgical residency programs in the United States involve 60 months of training in an accredited program. The cases that residents participate in are logged and residents must log at least 750 major operative cases to graduate with at least 150 cases occurring in the chief resident year.

In practice, most surgical skills are taught via the Halstedian apprenticeship model of “see one, do one, teach one”. Under this model, to gain proficiency, residents should repeatedly and intensively work with treating surgical patients under the supervision of a skilled and experienced surgical teacher [1,2,3].

Feedback and assessment of these procedures currently require the time and energy of expert surgeons, and are highly variable and subjective [4,5]. Further, the development of new and innovative surgical techniques along with laws limiting a resident’s work week force surgeons-in-training to learn and practice more surgical procedures and associated techniques with less available time. Thus, active efforts are being made towards developing assessment tools that are objective and/or automated. Further, the Accreditation Council for Graduate Medical Eduation (ACGME) has demanded that the assessment of resident skills be objective, but has not offered any guidance as to how to do so.

Using outcome-based metrics, such as complication rates, morbidity rates, and mortality rates are of limited usefulness. Variability across patients and the complexity of cases could strongly skew these metrics. These metrics may also require a large number of cases and/or long post-operative analysis before realizing any significant differences between surgeons. Further, these metrics only tell the surgeon if the surgery was successful or not and are not useful in guiding the surgeon-in-training on how to better improve future surgeries. Thus, assessment of the process by which the surgery was performed is preferred.

Existing methods for objective and/or automated surgical skill evaluation can be classified in one of three ways: 1) Structured Grading 2) Descriptive Statistics and 3) Language Models [6].

1.2 Structured Grading

Structured human grading aims to improve upon the standard approach of an experienced surgeon overseeing a surgeon-in-training and providing unstructured verbal feedback. In an effort to make evaluation standardized, standard checklists are provided to a group of expert surgeons as they oversee surgeons-in-training performing simulated procedures on inanimate bench top models. These scales try to distill out the components that compose an effective surgical performance that are common across difference types of procedures. Structured grading has been implemented in different ways, some of which include [4]:

1. Objective Structured Assessment of Technical Skills (OSATS)
2. Objective Structure Clinical Examinations
3. Global Rating System (GRS)
4. Outcome Measures
5. Error Score Card Analysis

There also exist procedure-specific scales that aim to provide enhanced feedback and improve training by allowing trainees to concentrate their efforts on a subset of a task. Despite increased standardization, structured grading methods suffer from a number of drawbacks. These drawbacks include subjectivity across human observers, difficulty in broadly categorizing the subject, and the limited prior work on the relationship between technical skill and patient outcomes. Additional drawbacks include the need for high-fidelity models and work stations and the need for expert surgeons to conduct the evaluation.

1.3 Descriptive Statistics

Surgical skill evaluation using descriptive statistics from tool motion aim to eliminate many of the drawbacks inherent with structured human grading. The underlying assumption behind descriptive statistics is that an increasingly skilled surgeon-in-training will exhibit observable changes in their tool motion. Most prior work focuses on movement efficiency, i.e. a more skillful surgeon will be more efficient in their movements. Prior work in this area uses data collection systems to track surgical motion. This motion is then analyzed, usually in an automated and objective manner. These data collection systems include optical, mechanical, or electromagnetic tracking system. Captured motion data often includes positions and velocities and some include measures of force.

Difficulties with tracking tool motion include designing the tracking system to be non-obtrusive so as to not interfere or significantly alter the surgery itself. The tracking system and associated components must also satisfy the sterilization requirements of the hospitals. The variability in the types of surgery is an additional challenge. Certain tracking systems and certain metrics will be more practical for certain surgery types and it is unlikely that a single method of surgical skill evaluation using descriptive statistics will be entirely applicable across all surgeries.

Most surgeries, however, can be broadly classified into three types: 1) Traditional Minimally Invasive Surgeries 2) Robotically-Assisted Minimally Invasive Surgeries and 3) Open Surgeries. Prior work using descriptive statistics has focused on traditional and robotic-assisted minimally invasive surgeries. For non-robotic surgeries, one of the most commonly used systems is the Imperial College Surgical Assessment Device (ICSAD) with ROVIMAS motion-tracking software. The ICSAD uses an electromagnetic (EM) device to track the three-dimensional position of the surgeon's hand for the duration of the procedure. Other systems include [7,8,9]:

- ProMIS Augmented Reality Simulator
- Hiroshima University Endoscopic Surgical Assessment Device
- Advanced Dundee Endoscopic Psychomotor Tester.
- Computer-Enhanced Laparoscopic Training System
- Blue-DRAGON
- Patriot
- Simendo
- Xitact
- ITP
- TrEndo

The results from these tracking systems show correlation with global rating scales. Table 1 [4] provides a comparison of some of the advantages and drawbacks of different types of surgical skill assessment.

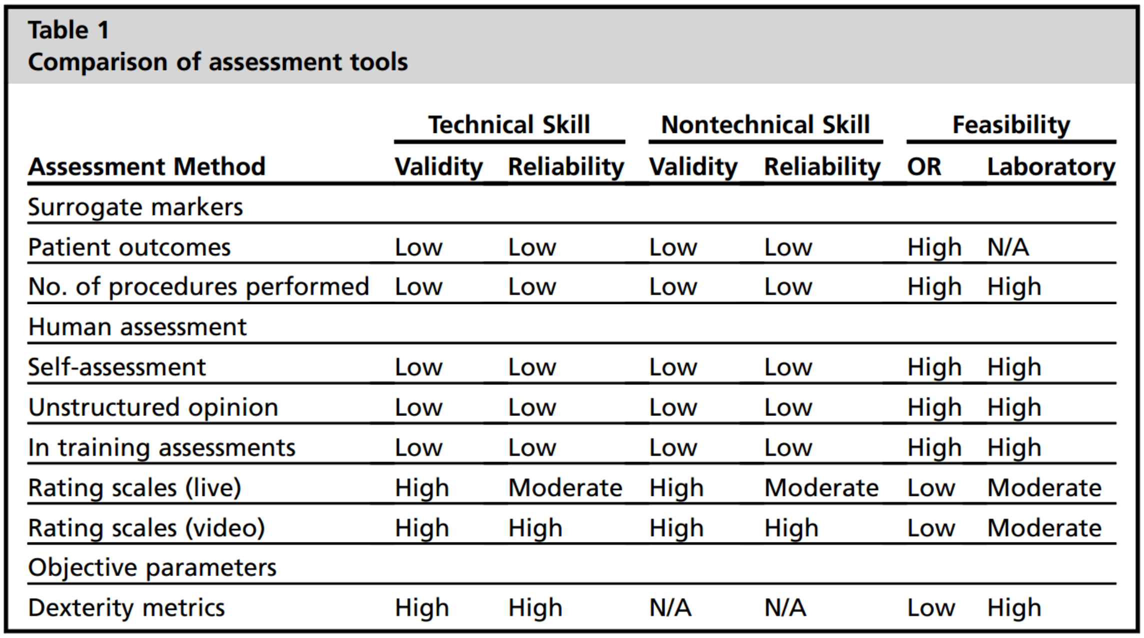


Table 1
Comparison of assessment tools

	Technical Skill		Nontechnical Skill		Feasibility	
Assessment Method	**Validity**	**Reliability**	**Validity**	**Reliability**	**OR**	**Laboratory**
Surrogate markers						
Patient outcomes	Low	Low	Low	Low	High	N/A
No. of procedures performed	Low	Low	Low	Low	High	High
Human assessment						
Self-assessment	Low	Low	Low	Low	High	High
Unstructured opinion	Low	Low	Low	Low	High	High
In training assessments	Low	Low	Low	Low	High	High
Rating scales (live)	High	Moderate	High	Moderate	Low	Moderate
Rating scales (video)	High	High	High	High	Low	Moderate
Objective parameters						
Dexterity metrics	High	High	N/A	N/A	Low	High

Figure 1: Comparison of Assessment Tools [4]

Robotic-assisted minimally invasive surgeries have helped increase the surgeon's dexterity by offering them full 6 Degree of Freedom manipulation at the surgical site. The most popular robotic system is Intuitive Surgical's da Vinci surgical system. Here, the surgeon resides in a control console and the surgeon's hand movements are mapped to robotic motions. Evaluation of surgical skill is more promising in the area of robotic surgeries because of access to rich, surgical motion data. For example, the da Vinci system provides access to position, velocity, orientation and joint and gripper angles as well as high-resolution stereoscopic video. Thus, no obtrusive hardware is needed to gather the motion data and descriptive statistics can be easily obtained from robotic systems.

Drawbacks of previously explored descriptive statistics include only considering very generic metrics such as time taken, number of hand movements, and path length. Further, these metrics vary greatly depending on the complexity of the surgical task and depend on completion of the task, making them unsuitable for real-time skill assessment.

1.4 Language Models

Language models differ from descriptive statistics by modeling motion in time. Language models view the surgery as a hierarchy and divide the overall task into subtasks and may allow for an understanding of intent or the quality of the surgical motion. Similar to a book, a complete surgery can be broken down in a hierarchy of repeated sub-parts. In a book, this could be chapters, pages, paragraphs, words, phonemes. In a surgery, a similar hierarchy could be the entire procedure, maneuvers, and surgical gestures. Work using language models [8-17] include modeling expert and novice surgeons using Hidden Markov Models, Conditional Random Fields, Descriptive Curve Coding, and other machine learning algorithms. Many of these methods, however, require manual labels from video analysis limiting their useful for real-time feedback. Analysis using language models can occur at multiple levels, 1) Procedure Level, 2) Maneuver Level or 3) Gesture Level. These models, however, tend to be more complex than descriptive statistics and, while past results have shown good performance in skill classification, the instructive value to surgeons-in-training has yet to be demonstrated.

2 Clinical Background

2.1 Problem and Epidemiology

Nasal obstruction is common with over 33% of adults experiencing some form of nasal airway obstruction. An estimated 25% of these adults seek surgical treatment. A significantly deformed or deviated nasal septum can result in cosmetic deformity and dysfunction and is the most common cause of nasal obstruction (See Figure 2). Thus, septoplasty, the surgical correction of a deviated septum, is the 3rd most frequently performed head and neck surgery in the US [18].

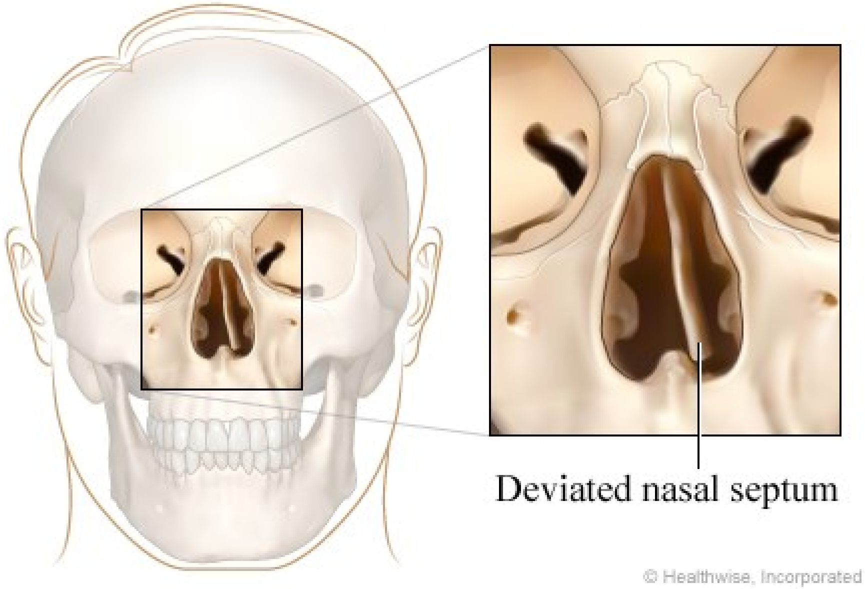


Figure 2: Diagram of a Deviated Nasal Septum [19]

2.2 Relevant Anatomy

The nose can be divided into two components: 1) The lateral wall and 2) The septum. The septum has a bony, cartilaginous component and a membranous component. The membranous component is the region of soft tissue between the cartilaginous septum and the collumnella, the skin in front of the nose. The bony portion of the septum has a midline structure made of the perpendicular plane of the ethmoid superiorly and vomer inferiorly. The cartilaginous portion of the septum is located under the nasal bones and is anterior to the bony septum. The nasal bones form a roof-like structure for the nose and are small relative to the cartilage. Thus, the lateral cartilages are the main factor in determining the shape of the nose [20,21],

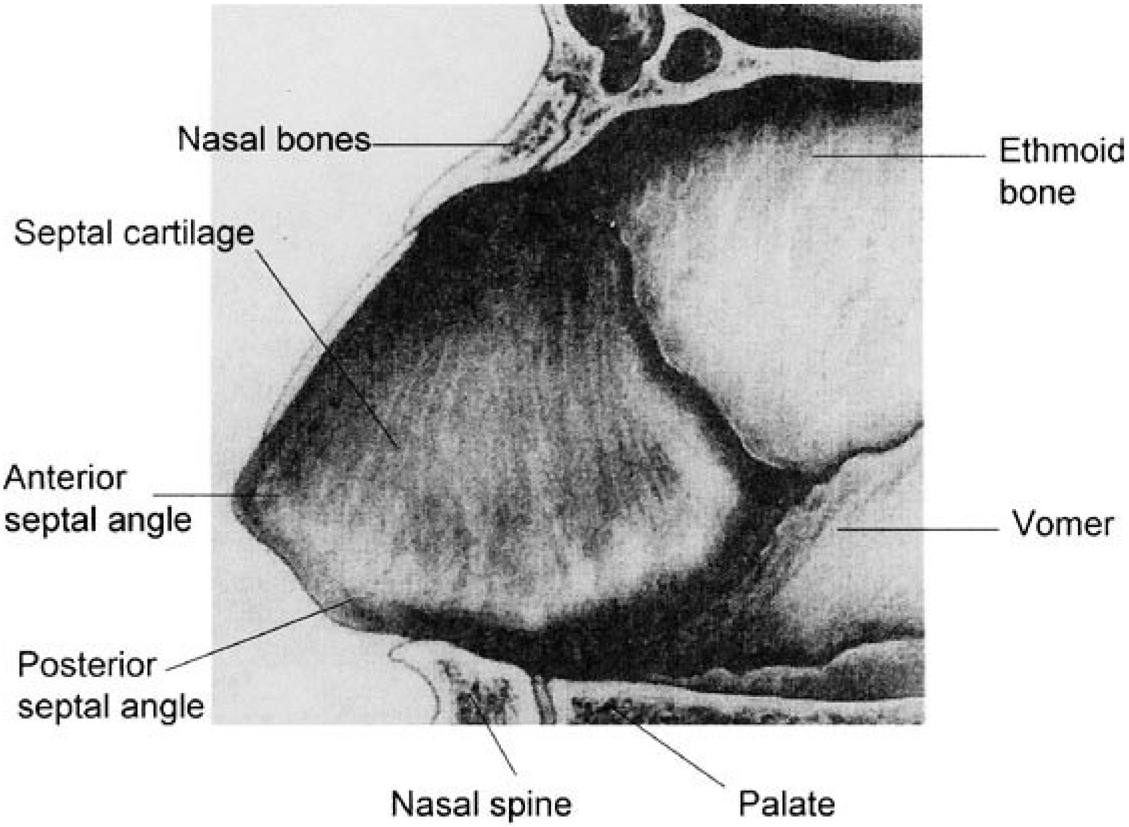


Figure 3: Anatomy of Nasal Septum [23]

2.3 Etiology

The cause of septal deviation can be traumatic or developmental. The outer portion of the nose grows at a different rate than the septum which, in some patients, can cause the septum to buckle. Traumatic impact, in childhood or in adult life, can fracture the septal cartilage with multiple fracture lines and further damage nasal bones. The site of greatest deviation is generally the junction between the bony and cartilaginous regions of the septum. Traumatic impact during childhood can result in severe problems in nasal obstruction when the patient is older as septal deviation increases with growth.

Septal cartilage may also be damaged in the neonatal period or during the birthing process. Microfractures during intrauterine life can weaken one side of the cartilage resulting in asymmetrical bending of the cartilage during growth [18].

2.4 Pathophysiology

The nasal septum serves as the main source of dorsal structural support for the nose—maintaining the position of the columnella and nasal tip. It separates the nasal passages and also acts as an effective shock absorber for the frontal fossa. When the force applied to the septal cartilage, however, exceeds its biomedical stress point, a fracture in the cartilage can occur.

Normal septum cartilage is straight with each side of the cartilage maintaining a balanced internal tension. Trauma generally results in asymmetric damage to one side of the cartilage causing overgrowth on the weakened side. Younger patients are more susceptible to septal deviation from trauma as insignificant trauma as a child can cause microfractures that impact the growth pattern of the septal cartilage [18,20].

2.5 Presentation and Indications

Septal disease presents with an obstructive sensation of not being able to move air through the nasal passages, or a sensation of increased nasal airway resistance. This can cause sinusitis, rhinitis, a stuffy nose or sleep apnea. Septal disease is treated by septoplasty. Inspection of the nasal cavity can be used to determine the presence or extent of septal deformity. Rhinoscopy can be helpful in better diagnosing the location, type, and severity of the septal deviation. Septoplasty is indicated when there is visible septal deformity with no other cause for nasal obstruction and access to the region behind the deviation is blocked i.e during sinus surgery, tumor surgery, or due to bleeding. [18, 20, 21].

2.6 Treatment

2.6.1 Intraoperative Procedure

During septoplasty, the patient is anesthetized either generally or locally. In both cases, the septum is devascularized to assist surgery.

Septoplasty proceeds as four main steps: incision, flap elevation, excision, and suture/packing. Incision is used to gain access to the anterior and posterior deviations. Incision is best made contralateral to the side with the maximum deviation, though some surgeon’s prefer always making the incision on the same side.

Flap elevation is performed to elevate the periochondrium, periostium, and mucosa from the cartilage such that it remains intact. To elevate the mucoperiochonrial flap, the surgeon must first take care in finding the avascular subperichondrial plane. Once the proper plane is found, a Cottle elevator tool is used to proceed with the flap elevation. (See Figure 4) The Cottle elevator tool is gently curved, but contains a sharp edge. The surgeon must be careful to avoid perforating or tearing the delicate mucoperichondrium. The perichondrium is attached to the underlying cartilage by fibrous attachments dispersed across the flap. The surgeon uses the Cottle elevator to break these fibrous attachments. A more blunt tool, like the Freer elevator can be used later in the procedure and a nasal speculum can be used to improve visualization of the dissection.

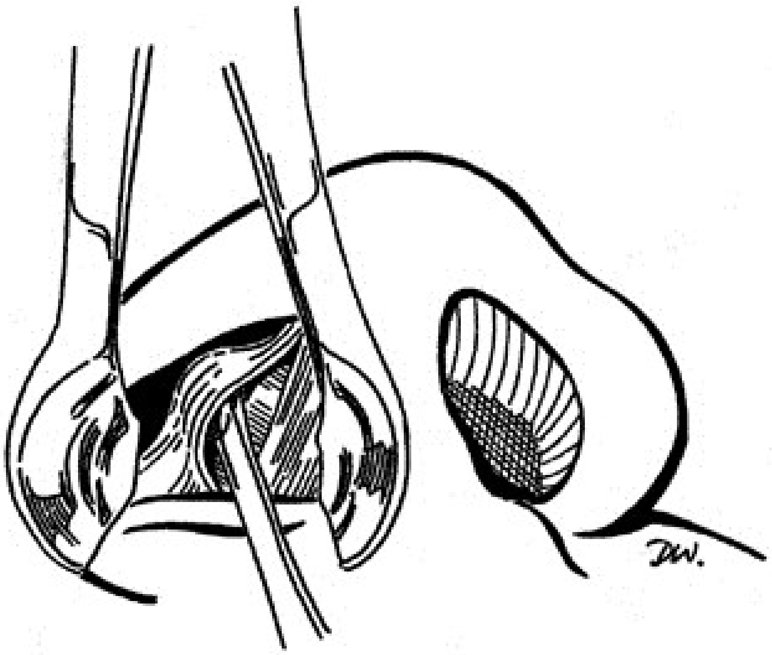


Figure 4: Elevation of the mucoperichondrial flap with a Cottle elevator [18].

When the ipsilateral flap is completely elevated, the Cottle elevator is used to incise the cartilage in the septum. The contralateral flap is then elevated. Again, care must be taken to avoid incision of the contralateral mucosa or contralateral flap elevation due to the potential of septal perforation.

When both mucoperichondrial flaps are elevated, the deviated septal cartilage is accessible and the deflection in the cartilage is either repaired in place or removed and repaired outside of the nose and replaced.

The bony deviations are corrected and the cartilaginous septum is replaced between the septal flaps such that it is aligned without deviation. An anchoring suture is sometimes required for stability. The incisions are closed using a suture [18, 20, 21].

2.6.2 Risks and Complications

Potential complications of septoplasty include epistaxis, nasal crusting, septal hematoma, sinus infection, perforation of the septum, toxic shock syndrome, cerebrospinal fluid leak, worsened nasal airway breathing, and the potential need for a revision procedure. Estimates of the success rates of septoplasty are around 70% [18].

3 Data Collection

Tracking of Cottle elevator tool motion during nasal septoplasty has never been documented. As such, a custom system was designed to easily, reliably, and unobtrusively capture high resolution tool motion data for the duration of flap elevation during nasal septoplasty. The following section describes the data recording apparatuses, set-up, and protocols that were developed for the purposes of this project.

3.1 Apparatuses

Our integrated data collection system is composed of the following three pieces of commercial equipment:

1) NDI Aurora Electromagnetic (EM) Tracker (Figure 5):

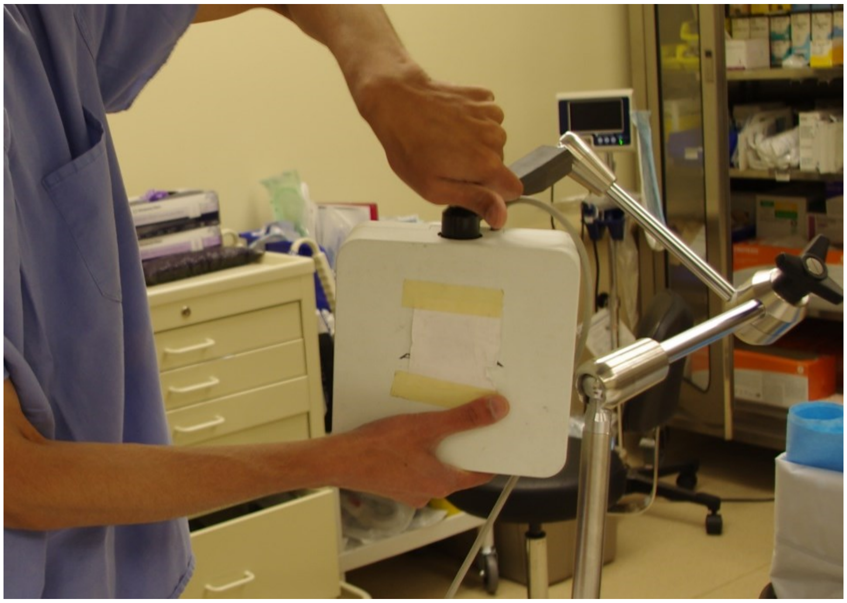


Figure 5: An NDI Aurora EM Field Generator with supporting arm in the Operating Room

Electromagnetic tracking systems can track the 3D position of objects with embedded sensor coils. These objects, when located within controlled and varying magnetic fields induce voltages in the sensor coils. By measuring these voltages, it is possible to compute the orientation and distance of the sensor object relative to the magnetic field. The advantages of using an electromagnetic system over alternatives such as optical tracking include tracking without line-of-sight constraints making it suitable for surgical applications and small sensor sizes.

An NDI Aurora electromagnetic tracker [23] was used to collect position data from the Cottle elevator. This tracker is composed of a Field Generator, Sensors, Sensor Interface Units, and a System Control Unit. Large ferromagnetic objects can results in electromagnetic noise reducing the accuracy of the readings.

The Field Generator emits a varying electromagnetic field that is low in intensity. The varying electromagnetic generated by the Field Generator induces small currents in the Sensors. Analysis of the characteristics of these currents yields information regarding the distance and orientation of the Sensor relative to the Field Generator. The Sensor Interface Units amplify and convert the analog electrical signals to a digital signal and allow for increased distance between the System Control Unit and the sensors while also minimizing the noise in the data. The System Control Unit gathers the digitized Sensor data and computes the position and orientation of all connected sensors and acts as the direct interface to the hosting computer.

The system captures six degree of freedom data from each attached sensor with an accuracy of 0.48 mm and 0.88 mm 95% confidence intervals (in an environment free of electromagnetic disturbances). It is powered by 100-240 VAC at 50/60 Hz and can take independent measurements at a rate up to 40 Hz.

2) Microsoft Kinect® Systems [24]

The Microsoft Kinect is a video recording and motion sensing input device. The device contains an RGB camera and a depth sensor that provides full-body 3D motion capture. The depth sensor operates using an infrared laser projector and a monochrome CMOS sensor. The use of the Kinect system in this data collection system was primarily for its capability as a RGB video recording camera. The RGB video stream can be captured at up to 30 Hz and at up to a resolution of 1280x1024. The practical range of use of the depth sensor is between 1.2 and 3.5 meters. The device is powered via USB.

3) Laptop Computer

A laptop computer with the Kinect and Aurora software and drivers installed is used to run the custom-built data collection software (See Figure 6).

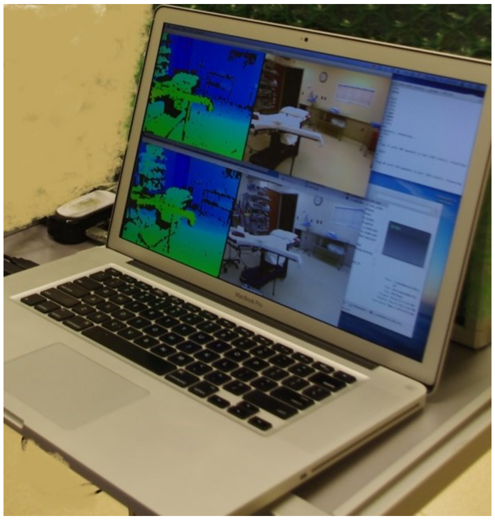


Figure 6: Data Recording Laptop showing Depth and RGB video streams

3.2 Data Collection Set-Up

The integrated data-collection system is composed of multiple parts, some using the commercially available equipment previously described and some using custom-built tools. To track tool motion, one EM sensor is affixed to a custom-designed fixture that is attached to the Cottle Elevator. A second EM sensor is attached to the sterile towel wrapped around the patient's head. This reference sensor is referred to as 'head sensor'. These sensors are connected by wire to their corresponding Sensor Interface Units located on a data collection cart. The Sensor Interface Units are connected to the System Control Unit also located on the data collection cart. The System Control Unit is connected directly to the laptop computer which is also located on the data collection cart. The Field Generator is affixed to a hinged metal arm that is attached to the patient's bed near the patients head, and is also connected to the System Control Unit (See Figure 5). Two separate Microsoft Kinect systems are rigidly fixed to a tripod that is placed at the bedside at a height such that the operation site is visible. The Kinect systems are powered via USB from the laptop.

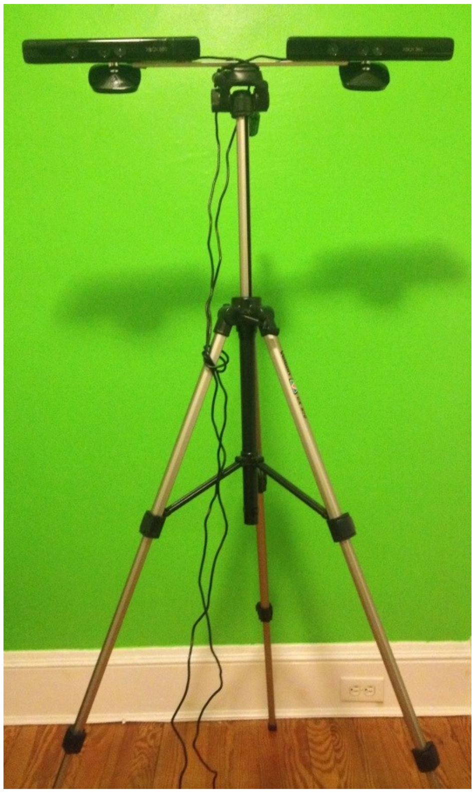


Figure 7: Kinect Tripod System

The fixture attached to the Cottle elevator was designed to allow for adequate sterilization of the tool as the tool would be in direct contact with the patient. The fixture needed to not impede the surgeon's motion in any manner. Further, it needed to be designed to minimize any interference between the sensor and other surrounding equipment. Also, as surgeons use both sides of the Cottle elevator, the sensor needed to be positioned to give reliable information regarding both tool sides. The resulting Cottle elevator design is a Y-shaped mount attached towards the middle of the cottle (See Figure 8). The sensor is designed to provide readings with acceptable accuracy for up to 20 sterilization cycles.

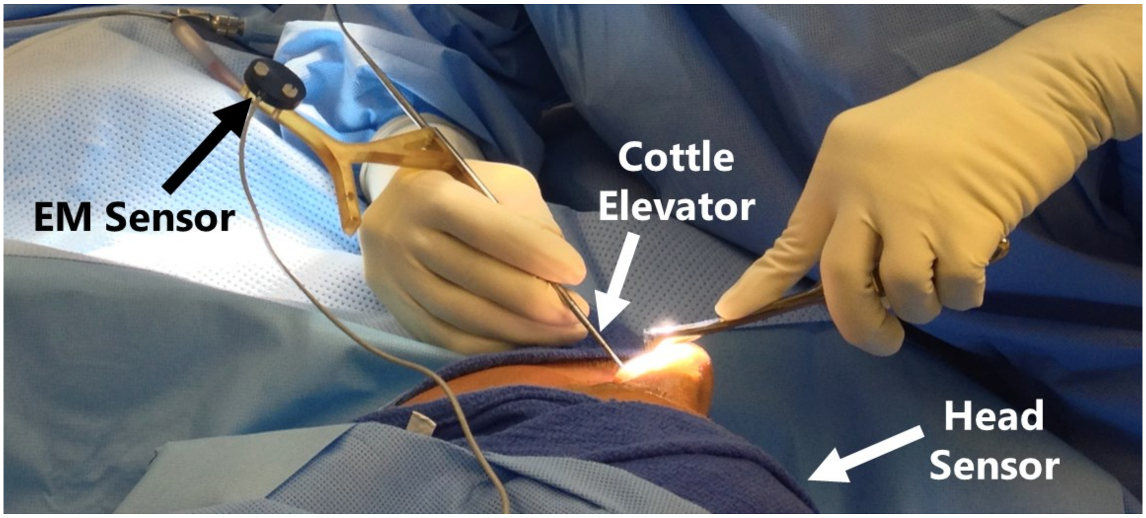


Figure 8: Cottle elevator with affixed electromagnetic (EM) sensor used by surgeon to elevate mucosal flap in a septoplasty procedure. The head sensor (not visible) affixed to forehead region for the most recent third of cases to track head motion.

The overall data recording set-up can be seen in Figure 9.

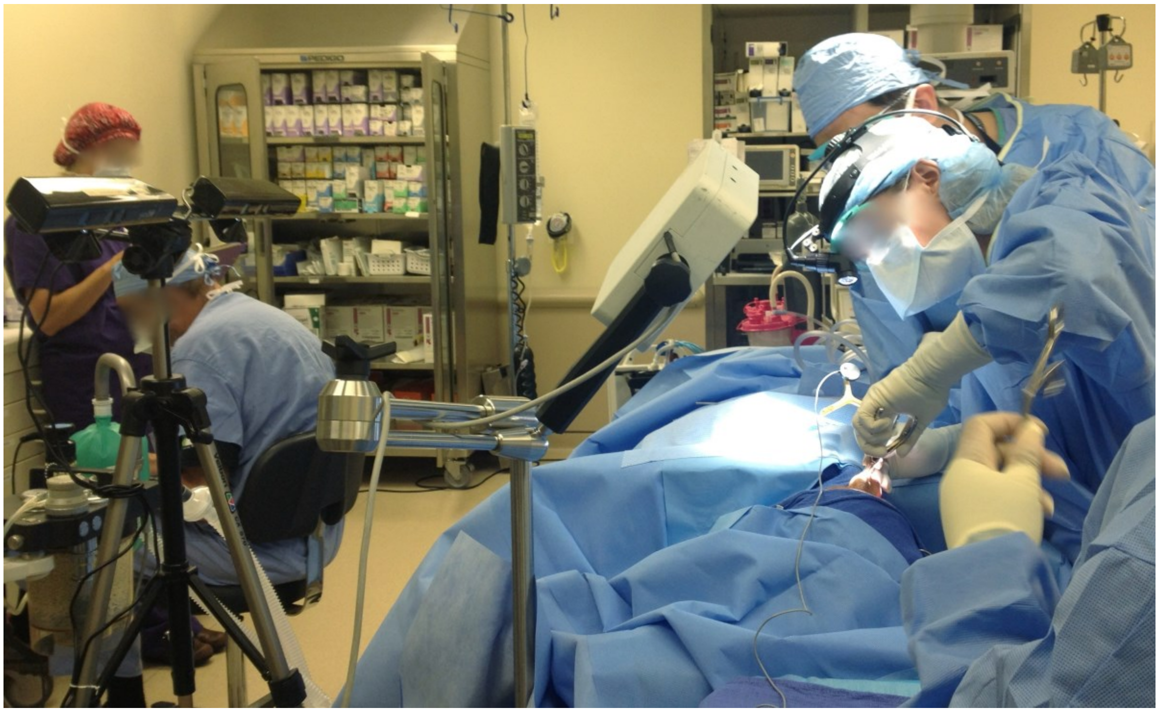


Figure 9: Data Recording Set-Up

3.3 Data Collection Protocol

The data collection protocol is designed to be operated by a single individual and to minimize interference with the procedure itself. The protocol itself was approved by the Institutional Review Board of the Johns Hopkins Hospital (Application ID: NA_00045730). Useable data has been collected from operating rooms at four different facilities across different Johns Hopkins medical institutions with attending surgeons, surgical fellows, and resident surgeons. Thus, the hardware and software components of the integrated data collection system have proven to be reliable, robust, and repeatable with a single data collector.

Before the procedure, the sensor-affixed Cottle elevator is sterilized using the hospital's standard sterilization protocol. Before the patient is brought into the operating room, all the equipment related to the data collection is brought into the operating room using the data collection cart. The Kinect cameras are set-up near the patient's bedside such that it can record the site of the surgery and the EM tracker components are set-up and connected to the interfacing laptop.

The software and hardware are tested to ensure all components are functioning properly. Once the patient is brought into the operating room and is anesthetized, the hinged arm is attached to the side of the patient's bed and positioned to be located near the patient's head. The Field Generator is then attached to the hinged arm such that it is positioned approximately 6 inches from the patient's head to ensure Cottle movements in the nasal region are within the detectable volume of the EM tracker.

The patient's pre-operative treatment is unchanged. The patient is treated as would normally be done in a nasal septoplasty case using the instruments and techniques the operating surgeons are accustomed to. Before the surgery beings, the head sensor is attached to the sterile towel wrapped around the patient's head prior to data collection. Data recording begins with the data collector pushing a single button and then proceeds without additional intervention. Once the operator is ready to begin the operation, the data collector instructs the surgeon to move the Cottle tip around the perimeter of the nose to register the tool tip with the position of the patient's nose.

3.4 Recorded Data

Kinematic data is recorded for both the Cottle sensor and the head sensor. This data includes frame number, time stamps, frame validity, 3D position and orientation data for each sensor. Data was recorded at 20 Hz.

Recorded video included depth maps and RGB images from both Kinect systems. Video data was recorded at approximately 15 Hz at a resolution of 480x640. The data recording software was designed to generate a global time-stamp such that the kinematic data and the video data could be synchronized and key events found in the video data could be accurately be traced back to the kinematic data and vice-versa.

In addition to the kinematic and recorded video data, meta data is also recorded by the data collector. This includes information such as the serial number of the surgery, the date of the surgery performed, the order of that case relative to other cases performed that day, the automatically-generated folder name, the initials of the Attending Surgeon, whether or not a trainee was present, the initials of the trainee if one was present, the year of residency of the trainee if present, the number of time the Cottle elevator in use was sterilized, whether a trainee performed part of the procedure, whether the Attending Surgeon performed a part of the procedure, the location at which the surgery was occurring, whether or not pivot calibration was performed before the given surgery, the data of the most recent pivot calibration, the detected error in the pivot calibration for both the spoon side and flat side of the Cottle elevator, the ID of the sensor, whether there was some event during the surgery that should suggest that the case be removed from analysis with an accompanying reason, whether the surgery was a revision surgery, whether the Attending Surgeon started the surgery, and other additional notes that may be of importance during analysis of the data.

To find the location of both tool tips of the Cottle elevator, the pivot calibration values are also stored. This includes a 3-dimensional vector indicating the offset between the sensor origin and the tool tip along with the date of the pivot calibration. Pivot calibration values from the same EM sensor are checked for consistency.

For surgeries occurring since July 8th 2013, subjective yet quantitative assessments of surgical skill were provided by the attending surgeons. The form for the Attending Surgeon to fill out following the surgery is shown in Figure 10 and Figure 11 below:

Septoplasty skill study – Attending form

Purpose of this form: This form should be used to record the attending surgeon's impression of how well surgeries in the septoplasty study went as well as their assessment of skill of trainees who may have participated in the case.
Who should complete this form: The attending surgeon who performed the septoplasty and/or supervised the trainee should respond to questions on this form.

1. For the portion of the septoplasty you performed, what is your assessment of how well the surgery went? □

   Instructions for Question 1: *Respond on a scale of 1 to 100, where 1 represents a catastrophic surgical failure and 100 represents a perfectly executed surgery.*

2. Did a trainee perform part or whole of septoplasty in this case?

( )$_1$ Yes. *Answer questions 3 to 5.*

( )$_2$ No. *Skip all remaining questions.*

3. For the portion of the septoplasty the trainee performed, what is your assessment of how well the surgery went? □

   Instructions for Question 1: *Respond on a scale of 1 to 100, where 1 represents a catastrophic surgical failure and 100 represents a perfectly executed surgery.*

4. In your opinion, what is the level of skill with which the trainee performed the septoplasty (either all or a portion of the procedure)?

( )$_1$ Novice

( )$_2$ Advanced beginner

( )$_3$ Competent

( )$_4$ Proficient

( )$_5$ Master

Instructions for Question 3

Novice	*Operator displays poor understanding of fundamentals; challenging time to complete surgery*
Advanced beginner	*Operator performs better than a novice but is transitioning to competence*
Competent	*Operator's performance is goal-directed and well planned*

Proficient *Operator performs with skill commensurate with that of a resident ready to graduate from the program*

Master *Operator performs with mastery of the procedure, at the level of an attending*

5. The trainee's surgical skill in this case was commensurate with that expected for an operator at the following level of training:

( )$_1$ PGY 1

( )$_2$ PGY 2

( )$_3$ PGY 3

( )$_4$ PGY 4

( )$_5$ PGY 5

( )$_6$ PGY 6

( )$_7$ PGY 7

( )$_8$ Fellow

( )$_9$ Attending

End of form

A similar form was distributed to trainees to subjectively evaluate their own performance following a procedure. The form is reproduced below:

Figure 10: Quantitative Assessment Form given to Attending Surgeons

Septoplasty skill study – Trainee form

Purpose of this form: This form should be used to record the trainee surgeon's impression of how well surgeries in the septoplasty study went as well as their self-assessment of skill with which they performed the surgery.
Who should complete this form: The trainee surgeon who performed the septoplasty, either in part or full, should respond to questions in this form.

1. For the portion of the septoplasty you performed, what is your assessment of how well the surgery went? □

   Instructions for Question 1: *Respond on a scale of 1 to 100, where 1 represents a catastrophic surgical failure and 100 represents a perfectly executed surgery.*

2. In your opinion, what is the level of skill with which you performed the septoplasty (either all or a portion of the procedure)?

( )$_1$ Novice

( )$_2$ Advanced beginner

( )$_3$ Competent

( )$_4$ Proficient

( )$_5$ Master

Instructions for Question 2

Novice *Operator displays poor understanding of fundamentals; challenging time to complete surgery*

Advanced beginner *Operator performs better than a novice but is transitioning to competence*

Competent *Operator's performance is goal-directed and well planned*

Proficient *Operator performs with skill commensurate with that of a resident ready to graduate from the program*

Master *Operator performs with mastery of the procedure at the level of an attending*

3. Your surgical skill in this case was commensurate with that expected for an operator at the following level of training:

( )$_1$ PGY 1 ( )$_5$ PGY 5

(   )$_2$ PGY 2

(   )$_3$ PGY 3

(   )$_4$ PGY 4

(   )$_6$ PGY 6

(   )$_7$ PGY 7

(   )$_8$ Fellow

End of form

Figure 11: Quantitative Assessment Form given to Resident and Fellow Surgeons

3.5 Raw Data File Structure

The raw data, both kinematic and video, are stored in a folder with a unique serial ID. This raw data is copied to a private server to which only relevant research members have access. The main folder contains two subdirectories, 1) Aurora and 2) kinects. The Aurora subfolder contains two files, StateDataCollection-componentNDISerial-State Table-DATE.csv and StateDataCollection-componentNDISerial-State Table-DATE.desc. The former contains raw kinematic data in a CSV file. This file contains the frame number, time stamps, validity, position, and rotational matrix of the tool at each frame. The descriptive file (.DESC) file contains meta information regarding the file name, the date, the file type, and the description of the columns of the CSV file. The kinects subdirectory contains two subdirectories, 1) out1 and 2) out2. These subdirectories contain the video data from each of the 2 kinect cameras. Within each of these folders are image files that are named with their global time-stamp i.e. d-1375713360.591123-134859566.pgm. These time stamps are used to synchronize video playback with the kinematic visualization. Each folder contains both RGB (.ppm) images and Depth images (.pgm) for the duration of each procedure. The first letter of the file name also indicates the image type with ‘r’ indicating RGB and ‘d’ indicating a depth image.

3.6 Pivot Calibration

Once every 2-4 septoplasty cases, pivot calibration is performed on the sensor-affixed Cottle elevator. The pivot calibration allows for the accurate detection of the position and orientation of the tool tip using only the position and orientation of the sensor. Since surgeon’s can use both ends of the Cottle elevator, pivot calibration is performed for both sides of the Cottle elevator and is recorded for post-processing. Pivot calibration is performed by rotating the tip of interest around a small indentation such that the position of the tool tip does not change and only the tool orientation changes. This is done for approximately 20 seconds while pivot calibration software made by NDI is collecting and computing the pivot values. The software computes a root mean square error and the pivot calibration process is repeated until the root mean square error is below 2 mm. Because of the gradual degradation of the sensor with repeated sterilizations, if the calibration process cannot yield a RMS less than 2 mm, the sensor is replaced with a new one.

3.7 Data Management

All collected data is stored on two separate hard-drives that are not connected to the Internet. One copy of the data is used for data analysis and the other copy of the data is used only as a backup. Access to the data is limited to those directly involved in the study. Patient information is anonymized in both the meta-data and in the videos. Data quality is assured via the RMS error in the pivot calibration as previously mentioned and also by checking and testing the equipment preoperatively.

One unanticipated problem that needed to be addressed was the Surgeon's headlight. Because of the increase in noise in the EM readings in response to ferromagnetic materials, data was collected showing an increase in the noise of the sensor readings when the surgeon's headlight was close to the EM field. Thus, a new headlight was introduced that uses a fiber optic cable to allow distance between the light source and the light projection.

3.8 Data Composition

The data collection process is an on-going procedure. Data collection started on 8/27/2012 and the first usable set of data was collected on 11/1/2012. Table 1 displays the number of cases and anonymized identity performed as of 1/28/2014.

Attending Surgeons and Surgical Fellows	
Anonymized Code	**Number of Usable Cases**
ATT-A	37
ATT-B	3
ATT-C	3
F-A (Fellow)	2
F-B (Fellow)	6
Surgical Residents	
R-A-09	6
R-B-09	1
R-C-09	3
R-D-08	1
R-E-09	1
R-F-09	3
R-G-06	4
R-H-07	1
R-I-10	0

Table 1: Number of recorded and usable cases performed by a given operator

The data in the forthcoming analysis is limited to the first 48 usable Septoplasty cases that were recorded. Of these, 28 surgeries were performed entirely by an Attending Surgeon or entirely by a Surgical Fellow. 6 surgeries were performed entirely by a Surgical Resident. The remaining 14 surgeries were performed by multiple operators. All data has been collected from one of three surgical locations.

A number of recorded cases were discarded for the purposes of the analysis. Reasons for discarding cases were varied. Examples of reasons for discarding included recording malfunction. In a few cases, the resulting kinematic raw data file did not capture the tool motion, likely due to a malfunction in the recording software. In some cases, there were major obstructions in the video stream by a nurse or by external equipment. These obstructions made accurate annotation of the videos impossible such that the nose perimeter could not be demarcated, the switching of tool side was obscured, or Cottle use could not be detected. An additional few cases were discarded because the performed procedure deviated from the typical septoplasty case. These cases included revision septoplasty cases which are significantly different than traditional septoplasty cases and also rhinoplasty cases.

4 Data Pre-Processing

Once the raw data is collected, the data is pre-processed so that analysis of the data can be automated and readily accessible and understandable. The data is pre-processed in the following ways:

1) Data Anonymization
2) Video Generation
3) Video-Kinematic Data Alignment
4) Video Annotation
5) Pivot Calibration
6) Data Compilation

4.1 Data Anonymization

For the privacy of the surgeons, trainees, and patients and to comply with our IRB-approved protocol, all data was anonymized. To be able to track an individual's progress and to be able to distinguish between experience levels, an anonymization scheme that maintains this information was necessary. Attending Surgeons were given the label of 'ATT-X' where 'ATT' denotes the operator's status as an Attending Surgeon and 'X' would be an uppercase letter unique to a specific Attending Surgeon. Resident Surgeons were given the label of 'R-X-##' where 'R' denotes the operators status as a Resident Surgeon, 'X' would be an uppercase letter unique to a specific Resident Surgeon, and '##' denotes the last two digits of the year when that operator was a first year Resident (i.e. R-A-09 for a 5th year Resident if operating in 2014). Surgical Fellows were given the label of 'F-X' where 'F' denotes the operator's status as a Surgical Fellow and 'X' would be an uppercase letter unique to a specific Surgical Fellow. To anonymize the location of the surgery, different locations were given the label of 'Loc-X' where 'Loc' indicates that the label refers to a surgical location and where 'X' is an uppercase letter unique to a specific surgical location.

Videos were anonymized by applying a Gaussian blur to the top third of the generated videos which was sufficient to conceal the identity of operators. The patient's face was sufficiently covered during the procedure so as to not allow for identification of the patient.

4.2 Video Generation

For each surgery, a RGB video was generated. The raw data stored the Kinect data as .PPM files and .PGM files. These image frames were concatenated together to form standard .AVI files playable by commonly available media players. To increase processing speed and reduce file size, the resolution of the video was scaled to 1/4th resolution and the top third of the screen had a Gaussian Blur overlay to anonymize the operator as previously mentioned. The global time-stamp of the video frame was also printed on the bottom left corner of the video. Only RGB videos were produced as the subsequent analysis did not require use of the depth information provided by the Kinect and a video was generated for both the left and the right Kinects. A screenshot of a generated video is seen in Figure 12. The anonymized videos were stored in the ‘kinects’ subdirectory of the relevant surgery.

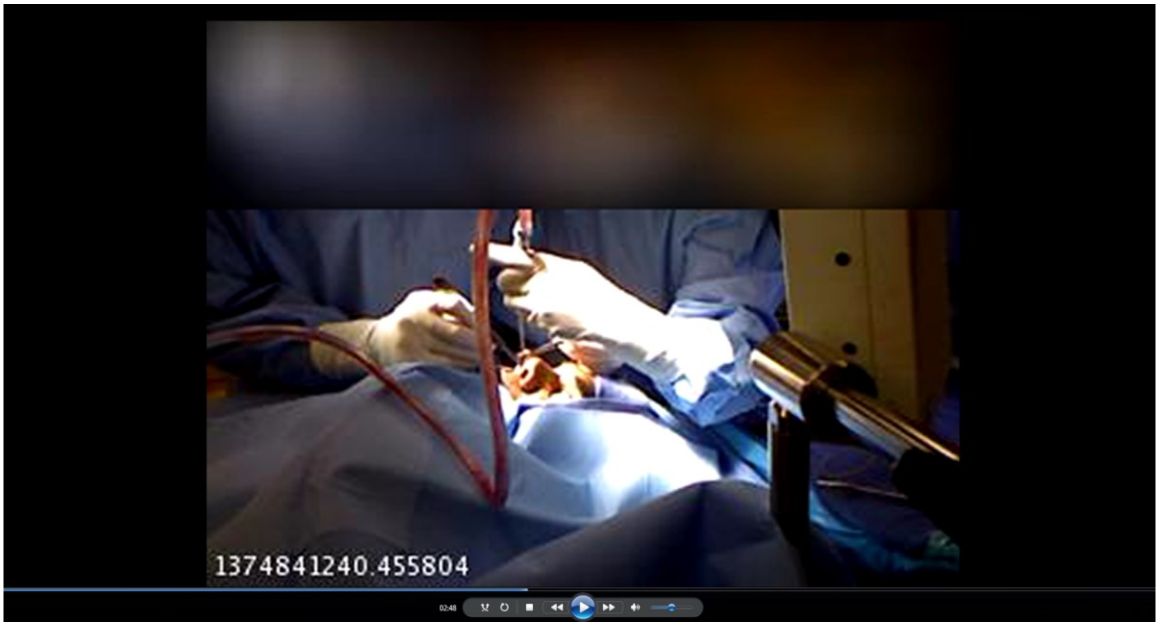


Figure 12: Screenshot of Generated RGB video with time-stamp and anonymizing Gaussian Blur

4.3 Video-Kinematic Data Alignment

To find key points of the surgery in the kinematic data, the video data and the kinematic data needed to be aligned. Since, both the video and kinematic data had a global time-stamp, alignment was straightforward. Using these alignments synchronized video playback and kinematic visualization could be generated as seen in Figure 13. These synchronized segments were used to verify the successful capture of video and kinematic data and to give context to what is being seen in the kinematic visualization.

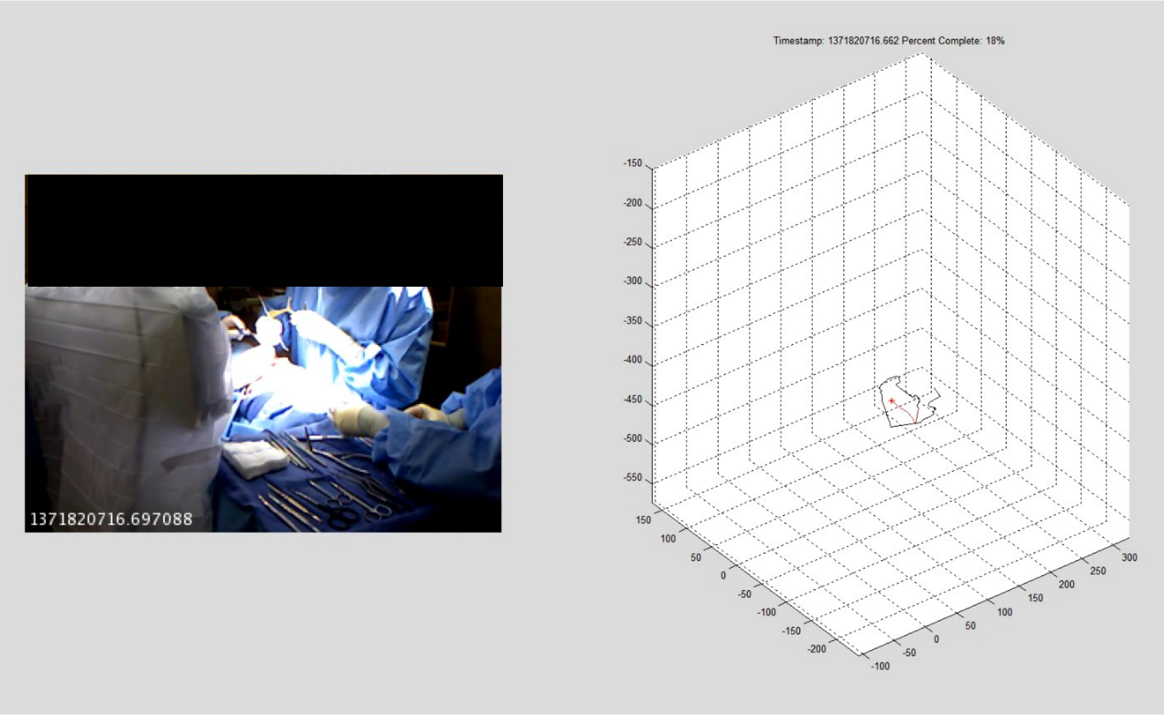

[redacted]

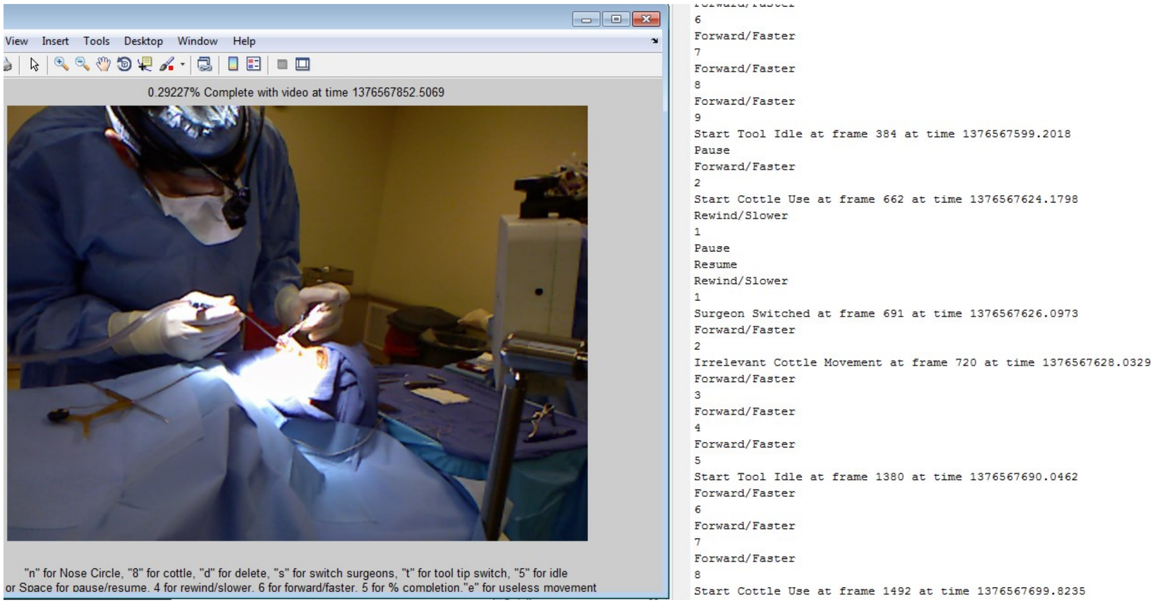
Figure 13: Synchronized Video and Kinematic Visualization Tool

4.4 Video Annotation

To incorporate critical surgical events in the surgery into the analysis, the data needed to be annotated. A number of relevant events, easily detectable through the video, are necessary for meaningful data analysis. The different events and actions that were labelled included time-points when a surgeon initiated a different part of the procedure, the surgeon switched between the different ends of the Cottle elevator, the operating surgeon switched with another surgeon, or the surgeon began use or disuse of the Cottle elevator.

To facilitate annotation using these labels, an easy-to-use and intuitive labeling tool was developed in MATLAB. The labeling could be done entirely via keystrokes and the resulting transcription of annotations was automatically saved in a machine-readable way. The recorded data included the start video frame for each event. A screenshot of the labelling tool can be seen in Figure 14.

To reduce the time required to annotate videos, the playback could be controlled using the number keypad with '6' initiating fast forward by increasing the number of frames skipped per transition by 1, '4' initiating a slowing down or rewind by reducing number of frames skipped by 1 and 'Space' or '0' initiating a pause in the video playback. Accidental or incorrect annotations could be deleted and all labels could be done using only keystrokes. Key-mappings are displayed to the user at the bottom of the playback screen. To reduce the required input from the annotator, only the beginning of each action was labeled and was considered to persist until the beginning of the next action.



0.29227% Complete with video at time 1376567852.5069

"n" for Nose Circle, "8" for cottle, "d" for delete, "s" for switch surgeons, "t" for tool tip switch, "5" for idle
or Space for pause/resume. 4 for rewind/slower. 6 for forward/faster. 5 for % completion."e" for useless movement

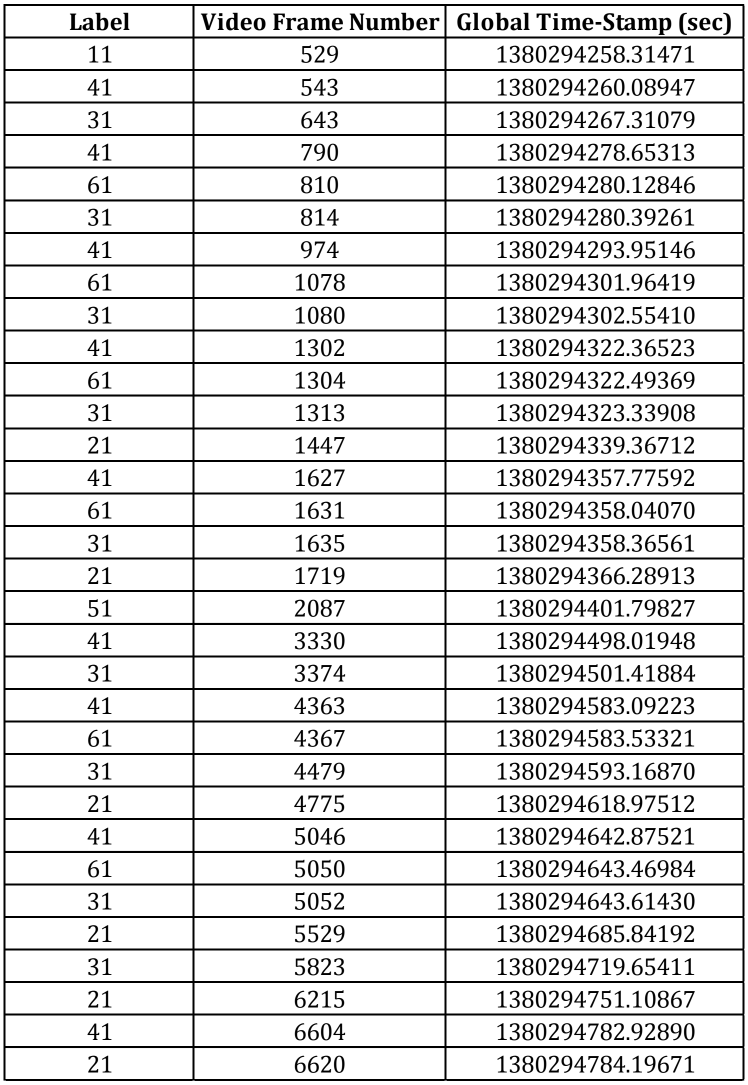
```
Forward/Faster
6
Forward/Faster
7
Forward/Faster
8
Forward/Faster
9
Start Tool Idle at frame 384 at time 1376567599.2018
Pause
Forward/Faster
2
Start Cottle Use at frame 662 at time 1376567624.1798
Rewind/Slower
1
Pause
Resume
Rewind/Slower
1
Surgeon Switched at frame 691 at time 1376567626.0973
Forward/Faster
2
Irrelevant Cottle Movement at frame 720 at time 1376567628.0329
Forward/Faster
3
Forward/Faster
4
Forward/Faster
5
Start Tool Idle at frame 1380 at time 1376567690.0462
Forward/Faster
6
Forward/Faster
7
Forward/Faster
8
Start Cottle Use at frame 1492 at time 1376567699.8235
```

Figure 14: Screenshot of Video Annotation Tool

A sample of the resulting data is shown in Figure 15. Each label was mapped to a number with ‘11’ representing the start of the nose perimeter circling, ‘21’ representing the start of the tool being idle, ‘31’ representing the start of Cottle elevator use, ‘41’ representing the start of irrelevant cottle movement, ‘51’ representing the switching of surgeons, and ‘61’ representing the time at which the tool tip is switched.

To label each video, the annotator loads the video to be labelled by changing the parent MATLAB directory to the ‘Video’ subdirectory of the surgery to be annotated. The annotator then runs labelVideos.m and the video labeling tool begins streaming the recorded video and the annotator can control and annotate the video with progress being displayed in the MATLAB console.

To maximize consistency of annotations, a set of annotation guidelines were used. Since the nose perimeter circling is critical for detection of the septal plane, the annotated beginning of the circling must be within a few frames of the actual circling and the next action must begin immediately after circling is completed. The subsequent annotations do not require such precise annotations. The 'Cottle in Use' action (31) begins when the annotator can see the surgeon moving the Cottle towards the surgical site. The 'Cottle Idle' action (21) begins once the Cottle is rested on the patient's bed or elsewhere where there is no Cottle use. The 'Irrelevant Cottle Use' action (41) occurs whenever the Cottle is not actively being used at the surgical site for surgical purposes and when it is not at rest, among other reasons, this can include transport, transferring the Cottle to another surgeon or nurse, or simply holding the Cottle. The 'Switching of Surgeons' action (51) begins at any time during the transition when one surgeon steps away from the patient and another surgeon assumes the operating position. The annotation only dictates the change of the operating surgeon. The 'Switching of Tool Side' (61) begins when the surgeon switches from using the Spoon-side of the Cottle elevator to using the Flat-side of the Cottle elevator or vice-versa. This action is usually done following a 'Irrelevant Cottle Use' action. It is assumed that the procedure begins using the spoon-side of the Cottle, so if the operator begins the procedure using the flat-side, a label of 61 should be assigned before Cottle use begins.



Label	Video Frame Number	Global Time-Stamp (sec)
11	529	1380294258.31471
41	543	1380294260.08947
31	643	1380294267.31079
41	790	1380294278.65313
61	810	1380294280.12846
31	814	1380294280.39261
41	974	1380294293.95146
61	1078	1380294301.96419
31	1080	1380294302.55410
41	1302	1380294322.36523
61	1304	1380294322.49369
31	1313	1380294323.33908
21	1447	1380294339.36712
41	1627	1380294357.77592
61	1631	1380294358.04070
31	1635	1380294358.36561
21	1719	1380294366.28913
51	2087	1380294401.79827
41	3330	1380294498.01948
31	3374	1380294501.41884
41	4363	1380294583.09223
61	4367	1380294583.53321
31	4479	1380294593.16870
21	4775	1380294618.97512
41	5046	1380294642.87521
61	5050	1380294643.46984
31	5052	1380294643.61430
21	5529	1380294685.84192
31	5823	1380294719.65411
21	6215	1380294751.10867
41	6604	1380294782.92890
21	6620	1380294784.19671

Figure 15: Sample Raw Annotation Data

4.5 Pivot Calibration

The data stored in the raw .CSV files are with reference to the origin of the sensor rather than the tool tip. To find the position and orientation of the tool tips, the results of the pivot calibrations are applied. For each surgery, the pivot values for both the spoon side and the flat side of the Cottle elevator are copied to two separate text files named pivot.txt for the flat side and spoonPivot.txt for the spoon side. These files are stored in the Aurora subdirectory of the relevant surgery. To apply these pivot values, load_EM_data.m is run with the root directory of the relevant surgery as an input. This function applies the pivot values to each frame of the raw data using the following formula:

$$T_{spoon}(t) = R(t) \times p_{spoon} + s(t) \tag{1}$$

$$T_{flat}(t) = R(t) \times p_{flat} + s(t) \tag{2}$$

Where $T_{spoon}(t)$ is the 3x1 tool tip position vector at the spoon side tip and $T_{flat}(t)$ is the 3x1 tool tip position vector at the flat side tip, $R(t)$ is the 3x3 rotation matrix referring to the orientation of the raw sensor, p_{spoon} is the 3x1 pivot vector for the spoon side and p_{flat} is the 3x1 pivot vector for the flat side, and $s(t)$ is the raw 3x1 position vector of the origin of the sensor at time t.

4.6 Data Compilation

To better organize the relevant data from each surgery, data from each surgery was compiled into a MATLAB struct with multiple fields. This is done by running the load_EM_data.m function with the root directory of the relevant surgery as an input. The relevant fields of the struct include the timestamp offset of each frame, the position and rotation of the sensor, the positions of the flat and spoon side tips of the Cottle elevator, the t_0 or the global time stamp of when the procedure started, and the head sensor position and rotation if a head sensor was present.

5 Data Post-Processing

Once preliminary processing data was completed, the context of the surgery needed to be considered for meaningful statistics to be computed and further post-processing of the data was required.

5.1 Annotation Processing

5.1.1 Kinematic Frame Alignment

To find meaningful events of the septoplasty procedure in the kinematic data, the manual annotations were used. As the annotations were made using the RGB video data, the frames at which the annotations were made did not correspond to the kinematic data frames. As such, alignment of the annotations was needed. Since a global time-stamp was also recorded, the kinematic frame corresponding to each manual annotation was the first kinematic frame that occurred at a global time stamp that followed the global time stamp of the annotation.

In addition, each annotated action was assumed to last until the beginning of the following annotation and thus the end frame for each annotation was also computed.

5.1.2 Operating Surgeon Inference

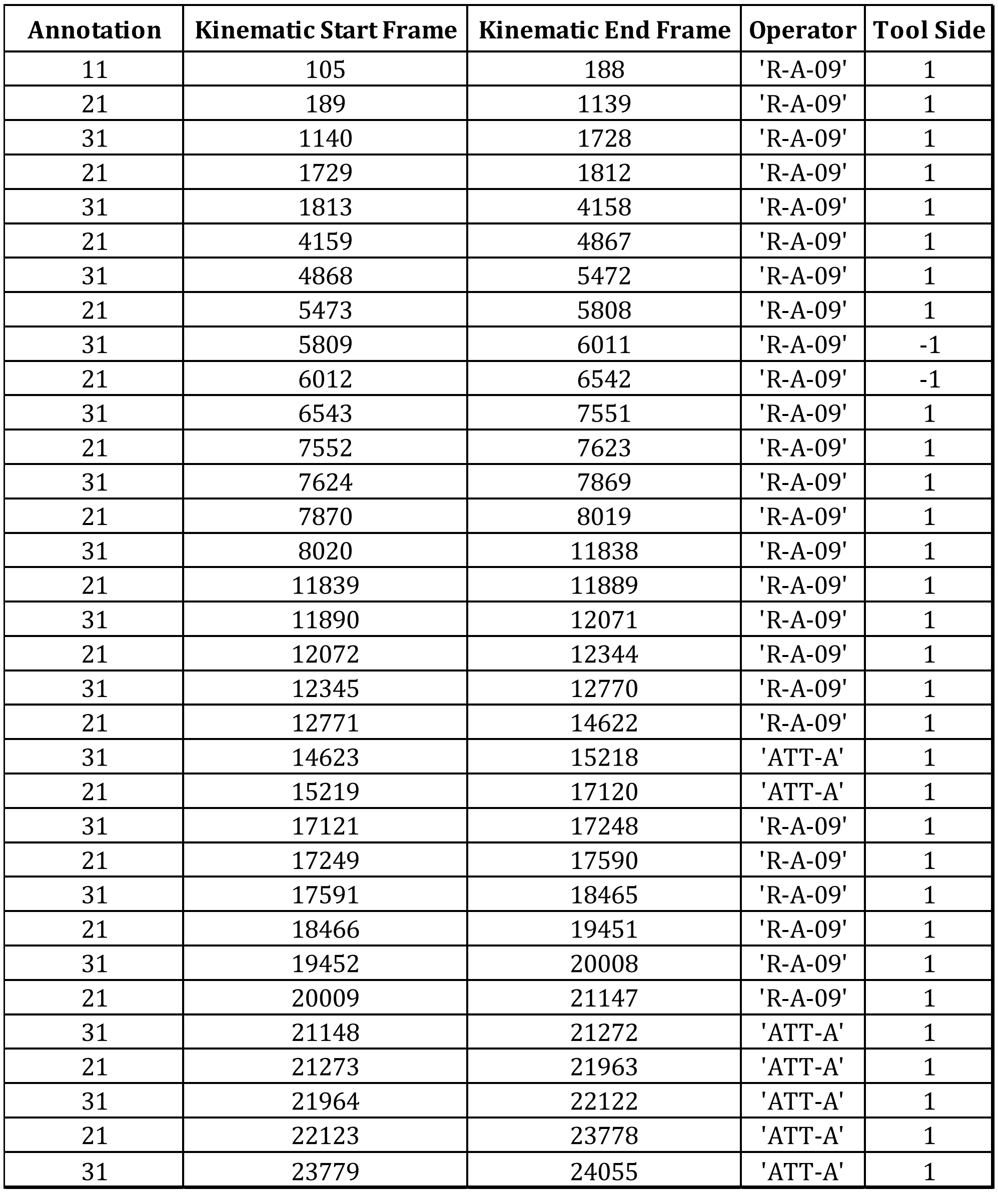
To ease the annotation process for the manual labeler, only the occurrence of a change in the operator was annotated, rather than the anonymized identifier of the operator. Thus, from the raw annotation, the operator at each kinematic frame had to be inferred. Using information from the metadata collected during each surgery, the surgeon who began each surgery could be determined. It was assumed that the surgeon who started the surgery continued until an annotation of ‘51’ was detected at which point it was assumed the other surgeon who participated in the surgery was the operating surgeon. The anonymized identities of each surgeon were also included in the meta-file.

5.1.3 Tool Side Inference

To ease the annotation process for the manual labeler, only the occurrence of a change in the Cottle side in-use was annotated (either Spoon-side to Flat-side or Flat-side to Spoon-side both labelled as ‘61’), rather than each tool side having its own identifier. Thus, from the raw annotation, the tool-side-in-use at each kinematic frame had to be inferred. The procedure is primary done using the spoon-side of the Cottle elevator and thus it was assumed the starting tool side in-use was the spoon-side. It was assumed that the tool side that was being used at the start of the surgery continued until an annotation of ‘61’ was detected at which point it was assumed the other tool-side was used.

5.1.4 Post-Processed Annotations

Post-processing of the labels transformed the data as shown previously in Figure 15 to a more informative set of annotations shown below in Figure 16. The post-processed annotations are located in the ‘Transcriptions’ subdirectory in the main septoplasty drive. Each annotation was renamed to reflect the serial number of the surgery seen in the meta-file.



Annotation	Kinematic Start Frame	Kinematic End Frame	Operator	Tool Side
11	105	188	'R-A-09'	1
21	189	1139	'R-A-09'	1
31	1140	1728	'R-A-09'	1
21	1729	1812	'R-A-09'	1
31	1813	4158	'R-A-09'	1
21	4159	4867	'R-A-09'	1
31	4868	5472	'R-A-09'	1
21	5473	5808	'R-A-09'	1
31	5809	6011	'R-A-09'	-1
21	6012	6542	'R-A-09'	-1
31	6543	7551	'R-A-09'	1
21	7552	7623	'R-A-09'	1
31	7624	7869	'R-A-09'	1
21	7870	8019	'R-A-09'	1
31	8020	11838	'R-A-09'	1
21	11839	11889	'R-A-09'	1
31	11890	12071	'R-A-09'	1
21	12072	12344	'R-A-09'	1
31	12345	12770	'R-A-09'	1
21	12771	14622	'R-A-09'	1
31	14623	15218	'ATT-A'	1
21	15219	17120	'ATT-A'	1
31	17121	17248	'R-A-09'	1
21	17249	17590	'R-A-09'	1
31	17591	18465	'R-A-09'	1
21	18466	19451	'R-A-09'	1
31	19452	20008	'R-A-09'	1
21	20009	21147	'R-A-09'	1
31	21148	21272	'ATT-A'	1
21	21273	21963	'ATT-A'	1
31	21964	22122	'ATT-A'	1
21	22123	23778	'ATT-A'	1
31	23779	24055	'ATT-A'	1

Figure 16: Post-Processing of Surgery Annotations

5.2 Initial Plane Estimation

The outcome of septoplasty depends on how the surgeon interacts with the septum. Using only tool motion and entirely ignoring the relative location of the septum discards likely useful information. To enrich the context of the tool motion, the septum was estimated to be a plane and the location of the septum was estimated using tool motion data.

Estimation of the position of the septal plane required information regarding the orientation of the head and the location of the nose. To estimate the septum plane, we took advantage of the similarity across patient with regards to relative proportions of body parts. We examine the anatomy of the nose for the average human adult in Figure 17.

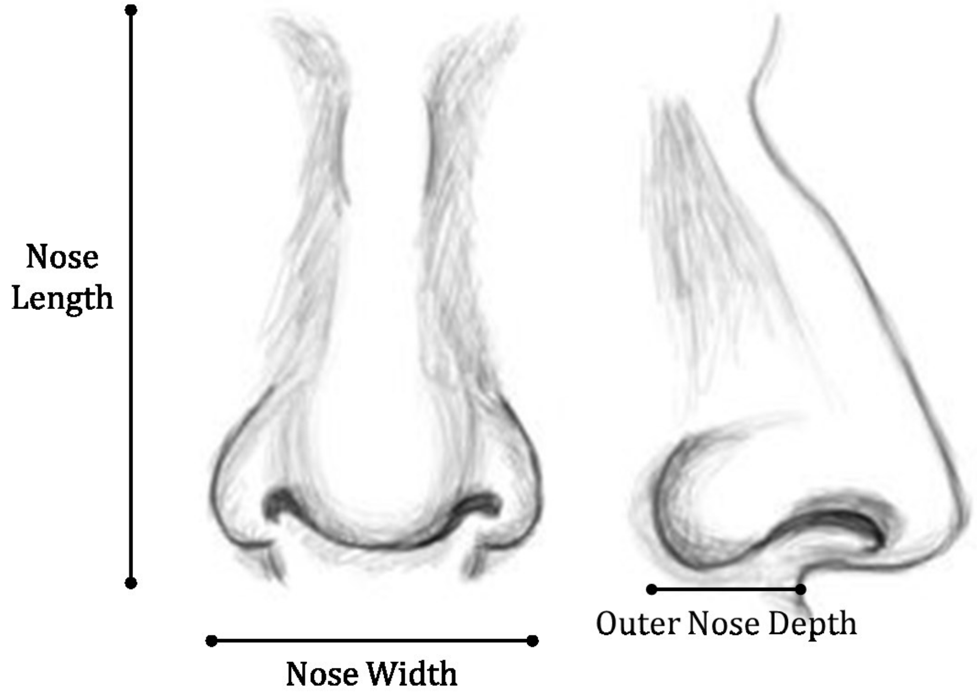


Figure 17: Human Nose Proportions [25]

We consistency see that the nose length is greater than the nose width which is greater than the outer nose depth. Thus, this information can be used to predict the location and orientation of the septum plane. The septum plane is oriented parallel to the nose length and the outer nose depth and is perpendicular to the nose width. During the surgery, the surgeon is instructed to use the Cottle tool to make an outline around the perimeter of the nose and thus the nose length, nose width and outer nose depth can be determined from this data. In the post-processed annotations, the beginning and end of the nose perimeter circling is also marked. Using the principal components of the tool tip data during the nose circling, an estimate of the orientation of the septum plane can be found.

5.2.1 Principal Component Analysis

Principal component analysis is an orthogonal linear transformation of the data to a new coordinate system such that projection of the data onto the choice of first principal component or first coordinate that would yield the greatest variance, the projection of the data onto the second principal component would yield the second greatest variance, etc. Further, each of the principal components are orthogonal to each other by construction [26].

More formally, we look at statistical PCA for three-dimensional data. Expressed mathematically, we look at x, where x is a multivariate random variable in 3-dimensions. The i^{th} principal component of x is y_i where

$$y_i = u_i^T x \text{ for } i = 1 \; to \; 3 \qquad \textbf{(3)}$$

Such that the variance of y_i is maximized subject to the constraints that each principal component is uncorrelated with preceding components and:

$$u_i^T u_i = 1 \text{ and } Var(y_1) \geq Var(y_2) \geq Var(y_3) > 0. \qquad \textbf{(4)}$$

Principal components can be computed efficiently using the singular value decomposition of the covariance matrix of the data.

5.2.2 PCA for Initial Septal Plane Estimation

Taking the three-dimensional position of the tool tip during the nose perimeter circling would yield data where the highest variance would occur along the vector parallel to the nose length. The second highest variance perpendicular to that vector would be parallel to the nose width. The third highest variance perpendicular to both nose length and nose width would be parallel to the outer nose depth. Thus, applying principal component analysis to the nose perimeter circling data can yield meaningful principal components from which the septal plane orientation can be estimated. More precisely, the septal plane can be estimated to be parallel to the plane formed by the first and third principal components of the nose perimeter circling data. The normal vector that also defines the plane would be the cross product of the first and third principal components i.e.

$$\vec{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = p_1 \times p_3 \tag{5}$$

To define the position of the plane, we estimate the center of the nose and ground the plane to that position. To estimate the center, we find the mean of each coordinate of every data point collected during the nose perimeter circling phase i.e. given N points during the nose perimeter circling phase, the center $[x_{center}; y_{center}: z_{center}]$ is defined as:

$$x_{center} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (6)$$

$$y_{center} = \frac{1}{N}\sum_{i=1}^{N} y_i \quad (7)$$

$$z_{center} = \frac{1}{N}\sum_{i=1}^{N} z_i \quad (8)$$

Thus, we can get the scalar equation of the plane of the form $n_1x + n_2y + n_3z = d$ by computing d as:

$$d = n_1x_{center} + n_2y_{center} + n_3z_{center} \quad (9)$$

As seen in Figure 18 and Figure 19, the visualization of the plane estimation closely matches the expected location and orientation of the septal plane:

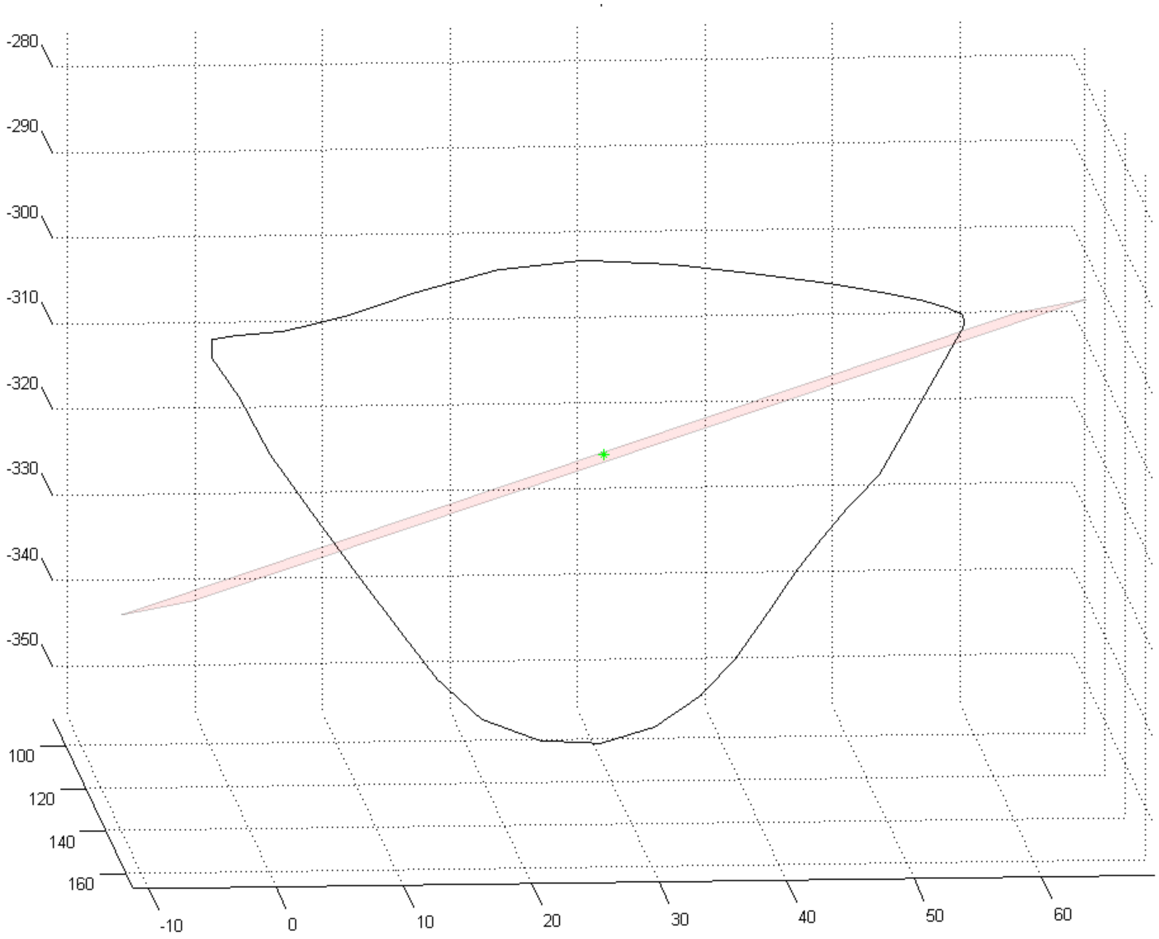


Figure 18: Visualization of Initial Septal plane estimation. Septal plane should in translucent pink and nose perimeter outline shown in black.

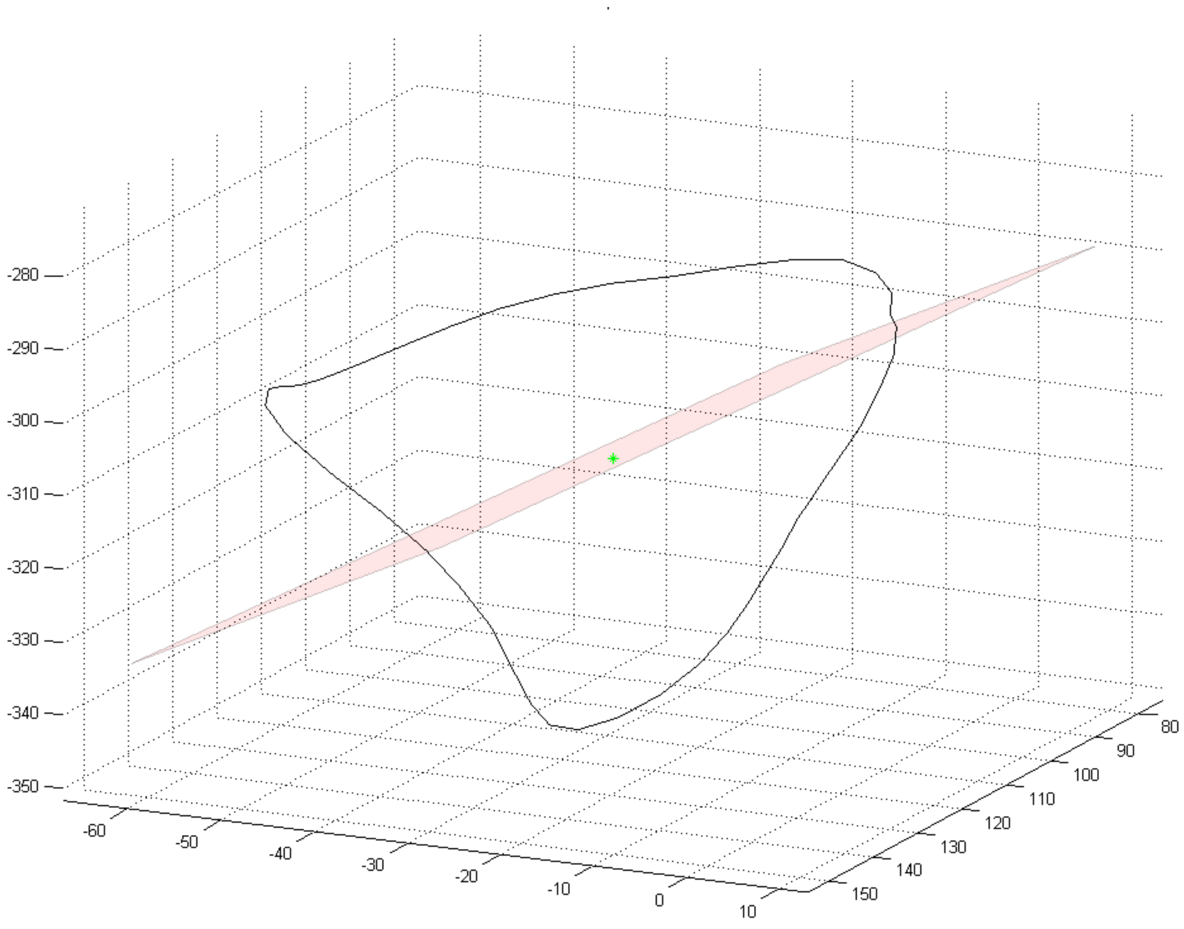


Figure 19: Visualization of Initial Septal plane estimation of a different surgery. Septal plane should in translucent pink and nose perimeter outline shown in black.

5.3 Dynamic Plane Estimation

Early analysis of the data used the initial plane position and orientation for the duration of the procedure. Visualization of the tool motion relative to the septal plane revealed, however, that using only the initial plane estimate was ineffective. See Figure 20. Analysis of the video revealed significant head motion throughout the surgery.

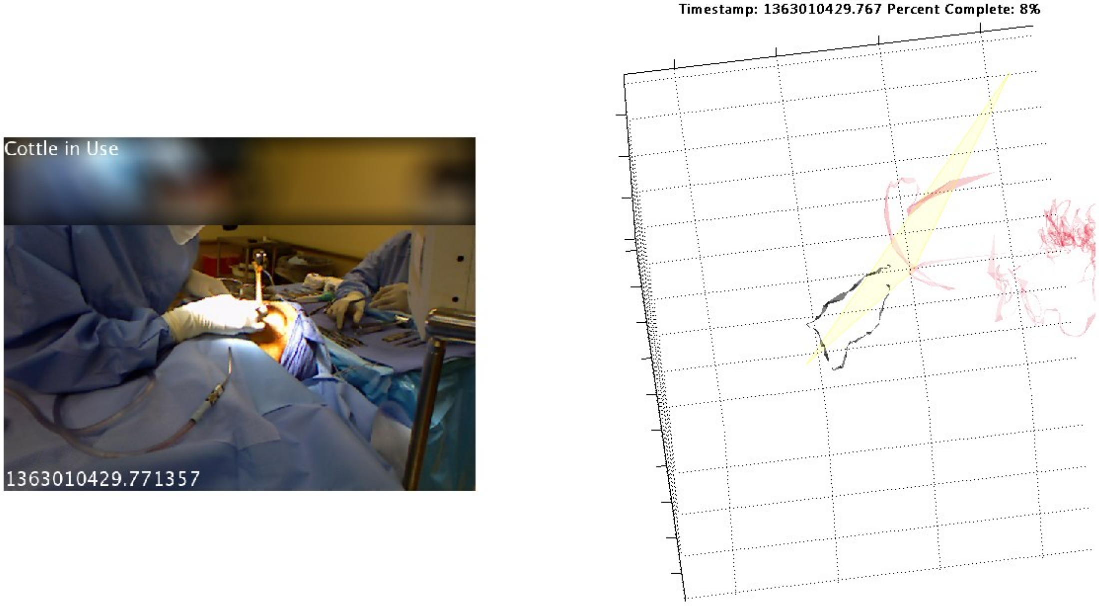


Figure 20: (Right) Tool Trajectory (translucent red) strays from the initial septum plane estimate and nose perimeter outline.

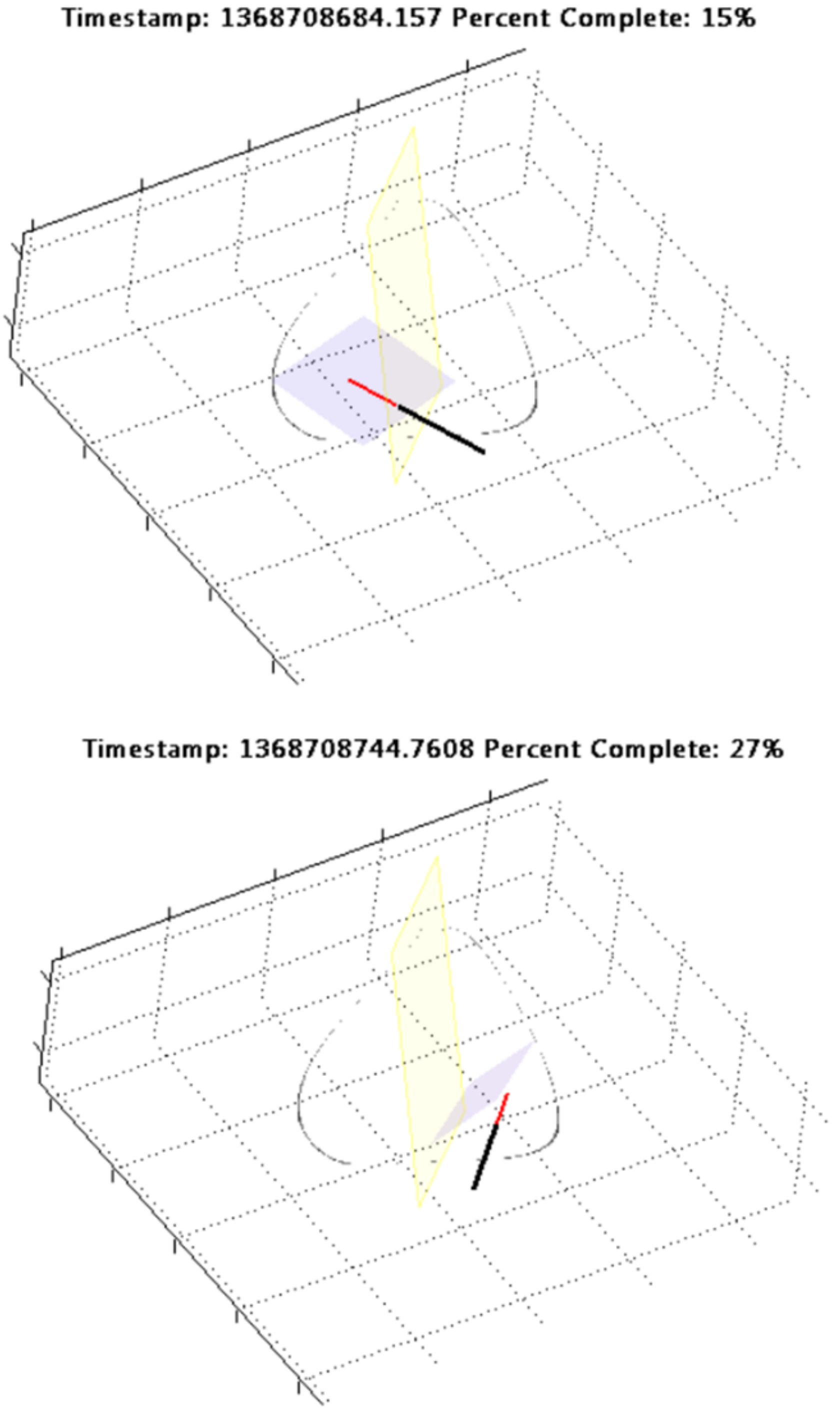
The force of the Cottle elevator and other tools on the patient caused the head to variably rotate around the neck throughout the procedure. As such, using a single estimate for the location of the septal plane yielded inaccurate results regarding the tool motion relative to the septum.

To compensate for the on-going head motion present throughout the procedure using only tool motion, techniques to dynamically estimate the septal plane location were developed and tested:

5.3.1 Sliding Principal Component Analysis

Our initial understanding of the septoplasty procedure led us to believe that the tool motion while the Cottle elevator was in use was along the septal plane. Further, we assumed that the tool movement occurred at a time-scale shorter than that of the head movement. Thus, using principal component analysis on a local window of tool motion data, we can dynamically estimate the location of the plane. More specifically, assuming tool motion was dominantly along the plane of the septum—the plane formed by the first and second principal components of tool motion that occur in a ~5 second window when the Cottle elevator was in use would yield an estimate of the septal plane at that time. By computing this plane for a sliding ~5 second window, a new estimate of the septal plane could be found at each time step.

The result, however, of dynamically estimating the septal plane using a sliding PCA approach yielded a highly unstable, non-representative septum. See Figure 21.



Timestamp: 1368708684.157 Percent Complete: 15%

Timestamp: 1368708744.7608 Percent Complete: 27%

Figure 21: Instability of Sliding PCA plane estimation. Yellow translucent plane shows initial septal plane estimation. Purple translucent plane shows septal plane estimation from Sliding PCA. Top and bottom images refer to different time points in the same surgery.

Analysis as to why the estimated plane was inaccurate yielded two insights. First, tool motion was *not* predominantly along the septum plane. In fact, tool motion was often directed away from the septal plane. Second, we saw that the head movements and tool movements occurred at a similar time scale. The force of the tool was the cause of the head motion and thus there was significant head motion each time the tool pushed against or pulled away from the septal plane.

5.3.2 Extrema PCA Approach

The sliding PCA approach yielded the insight that tool motion was often directed away from the septal plane and then often returned to the septal plane similar to a stroking motion. Thus, if principal component analysis was applied only to points where the tool was on the septal plane, the resulting septal plane would be more accurately positioned.

Using the assumption that the tool motion is predominantly away from the septal plane followed by a return to the septal plane, the local minima of the distance from the septal plane would yield points that should lie on or near the septal plane. Once a set of points these points were found, principal component analysis could yield a dynamic estimate of the septal plane orientation and position.

When detecting local extrema, noise and jitter in the sensor readings cause many false-positives. Thus, the data was smoothing using a moving-average filter with a window of four data points. Since the exact location of the septal plane was unknown and changing with time, computing the distance with respect to the current estimate of the plane would yield unreliable results. Thus the distance to the origin of the sensor was computed instead. More precisely, the Euclidean distance of points to the arbitrary origin was computed using the equation below:

$$d(t) = \sqrt{x(t)^2 + y(t)^2 + z(t)^2} \qquad \textbf{(10)}$$

The minima of these points were computed and assumed to be on or near the septal plane. For each set of three minima found, PCA was applied to these points with dynamic estimate of the septal plane being the plane resulting from the first and second principal components. The procedure continued, with the estimate of the septal plane being updated with each additional detection of a local minimum.

The results of this approach showed similarly unstable and inaccurate septum estimations. See Figure 22.

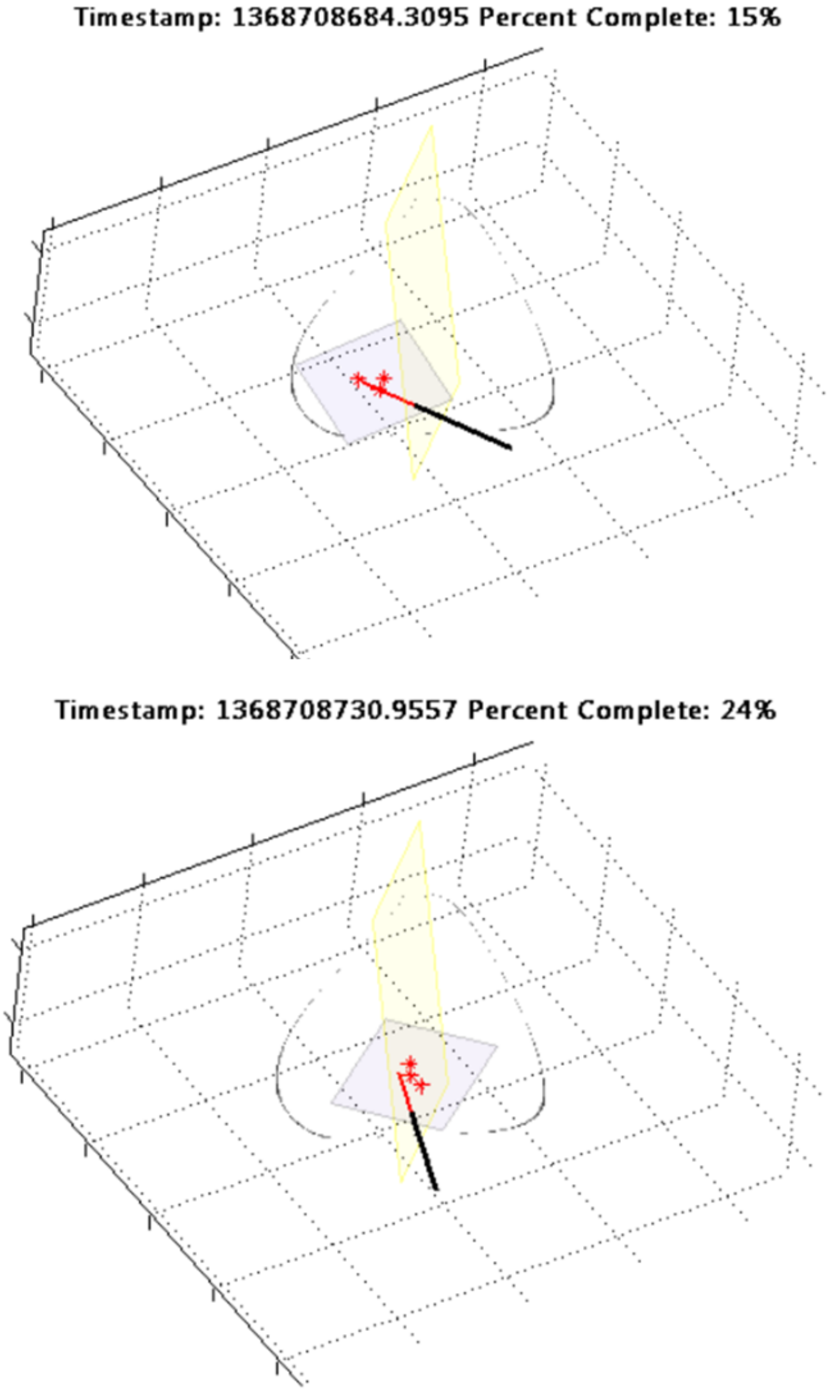

Timestamp: 1368708684.3095 Percent Complete: 15%
Timestamp: 1368708730.9557 Percent Complete: 24%

Figure 22: Instability of minima plane estimation. Yellow translucent plane shows initial septal plane estimation. Purple translucent plane shows septal plane estimation from minima PCA approach. The red dots show the detected minima. Top and bottom images refer to different time points in the same surgery.

Analysis of this approach shows that even though extrema were reliably detected, they were often clustered around a single point. Thus, small variations in a single point would cause the resulting principal components to be misaligned with the true septal plane. Further, the method was sensitive to outliers.

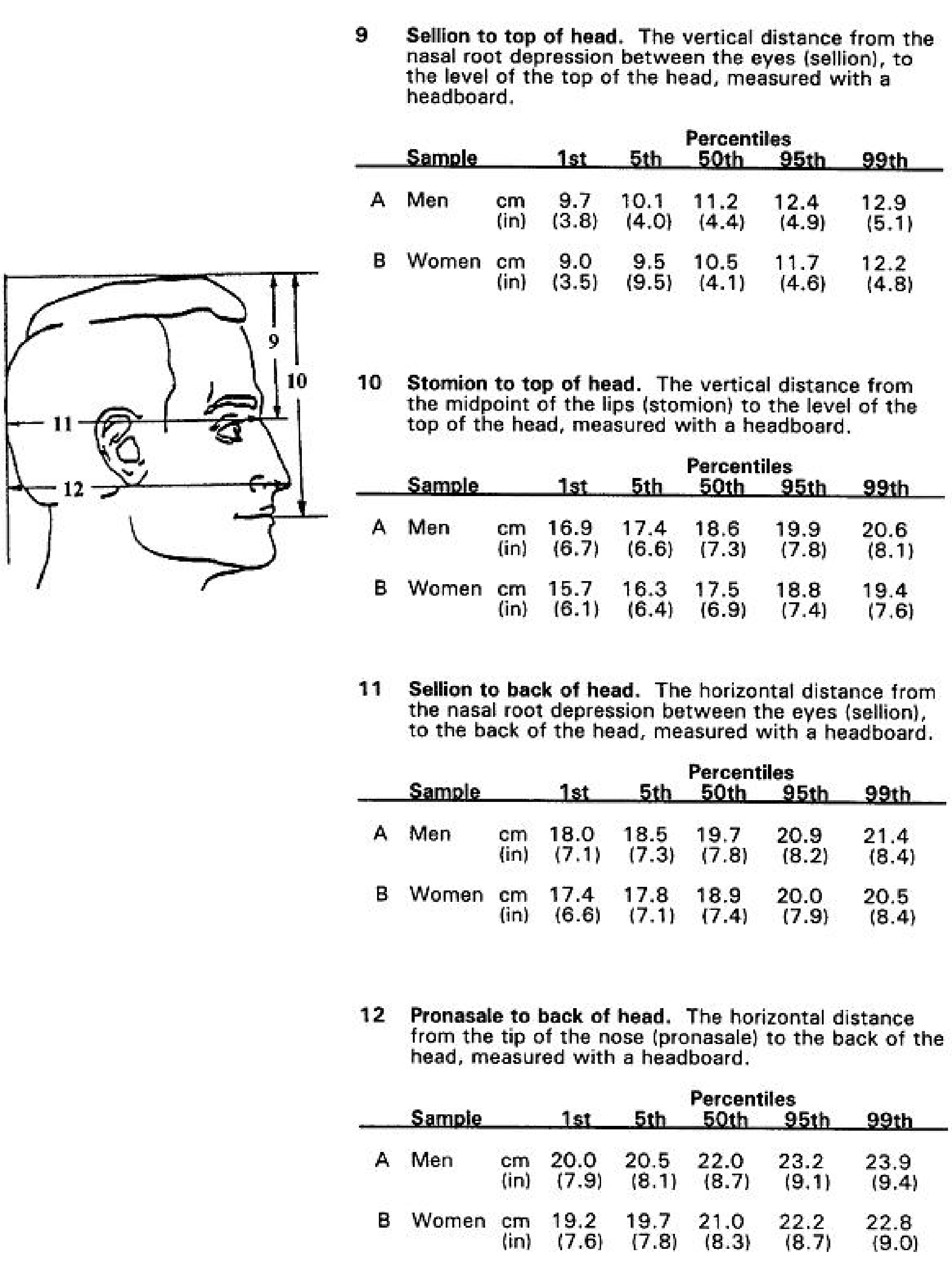
5.3.3 Extrema Approach with Constrained Rotation

While the plane generated using the previous approach was often not representative of the actual septal plane, the detected extrema did seem to appear near the region of activity indicating that the extrema did lie on the septal plane. Realizing that the head movement was predominantly manifest as a rotation around the neck of the patient, an approach that used the optimal rotation of the initial plane was attempted.

For this approach, a number of assumptions were made. First, the most significant movements of the septal plane were caused by a rotation about the axis of the patient's neck. Second, we assume that the distance between a patient's septal plane and the axis of rotation of their neck is roughly constant. Third, we assume that the initial septal plane, which is determined when there is negligible movement of the septal plane, is correct and that all following plane estimating are rotations of the initial septal plane estimate.

The initial steps to the algorithm followed the previous approach. The position data for when the Cottle was in use was smoothed using a moving average filter and the Euclidean distance to the arbitrary origin was computed. From this, local minima were detected.

The next steps aim to find the optimal rotation of the initial septal plane to match the observed Cottle activity. First, we needed an estimate of the location of the axis of rotation of the neck. Using head anthropometry measurements as seen in Figure 23, an estimated distance from the nose perimeter center to the axis of rotation was about 10 cm in the direction parallel to the third principal component of the nose perimeter circling data.



9 **Sellion to top of head.** The vertical distance from the nasal root depression between the eyes (sellion), to the level of the top of the head, measured with a headboard.

	Sample		Percentiles 1st	5th	50th	95th	99th
A	Men	cm	9.7	10.1	11.2	12.4	12.9
		(in)	(3.8)	(4.0)	(4.4)	(4.9)	(5.1)
B	Women	cm	9.0	9.5	10.5	11.7	12.2
		(in)	(3.5)	(9.5)	(4.1)	(4.6)	(4.8)

10 **Stomion to top of head.** The vertical distance from the midpoint of the lips (stomion) to the level of the top of the head, measured with a headboard.

	Sample		Percentiles 1st	5th	50th	95th	99th
A	Men	cm	16.9	17.4	18.6	19.9	20.6
		(in)	(6.7)	(6.6)	(7.3)	(7.8)	(8.1)
B	Women	cm	15.7	16.3	17.5	18.8	19.4
		(in)	(6.1)	(6.4)	(6.9)	(7.4)	(7.6)

11 **Sellion to back of head.** The horizontal distance from the nasal root depression between the eyes (sellion), to the back of the head, measured with a headboard.

	Sample		Percentiles 1st	5th	50th	95th	99th
A	Men	cm	18.0	18.5	19.7	20.9	21.4
		(in)	(7.1)	(7.3)	(7.8)	(8.2)	(8.4)
B	Women	cm	17.4	17.8	18.9	20.0	20.5
		(in)	(6.6)	(7.1)	(7.4)	(7.9)	(8.4)

12 **Pronasale to back of head.** The horizontal distance from the tip of the nose (pronasale) to the back of the head, measured with a headboard.

	Sample		Percentiles 1st	5th	50th	95th	99th
A	Men	cm	20.0	20.5	22.0	23.2	23.9
		(in)	(7.9)	(8.1)	(8.7)	(9.1)	(9.4)
B	Women	cm	19.2	19.7	21.0	22.2	22.8
		(in)	(7.6)	(7.8)	(8.3)	(8.7)	(9.0)

Figure 23: Average Dimensions of a Human Head [27]

Once the axis of rotation is estimated, the optimal angle of rotation around that axis is found. This is computed using the MATLAB single-variable bounded nonlinear function minimization (fminbnd.m). The approach tested out all angles of $-90°$ to $+90°$ and found the angle at which the sum of the square errors between the detected extrema and the rotated plane was minimized.

To compute this, the given angle-axis combination was converted to matrix form. This can be done using the following formula:

$$R = \begin{bmatrix} \cos\theta + u_x^2\left(1-\cos\theta\right) & u_x u_y\left(1-\cos\theta\right) - u_z\sin\theta & u_x u_z\left(1-\cos\theta\right) + u_y\sin\theta \\ u_y u_x\left(1-\cos\theta\right) + u_z\sin\theta & \cos\theta + u_y^2\left(1-\cos\theta\right) & u_y u_z\left(1-\cos\theta\right) - u_x\sin\theta \\ u_z u_x\left(1-\cos\theta\right) - u_y\sin\theta & u_z u_y\left(1-\cos\theta\right) + u_x\sin\theta & \cos\theta + u_z^2\left(1-\cos\theta\right) \end{bmatrix},$$

Figure 24: Equation converting angle-axis formulation of rotation to a Rotation Matrix [28]

where θ is the angle to rotate by and $u = \left[u_x; u_y; u_z\right]$ is the unit vector aligned with the axis of rotation. Once the rotation matrix is computed, the initial estimate of the septum plane is rotated. This is done by rotating the normal of the initial plane as well as the center point contained in the initial plane:

$$n_{rotated} = R \cdot n_{original} \tag{11}$$

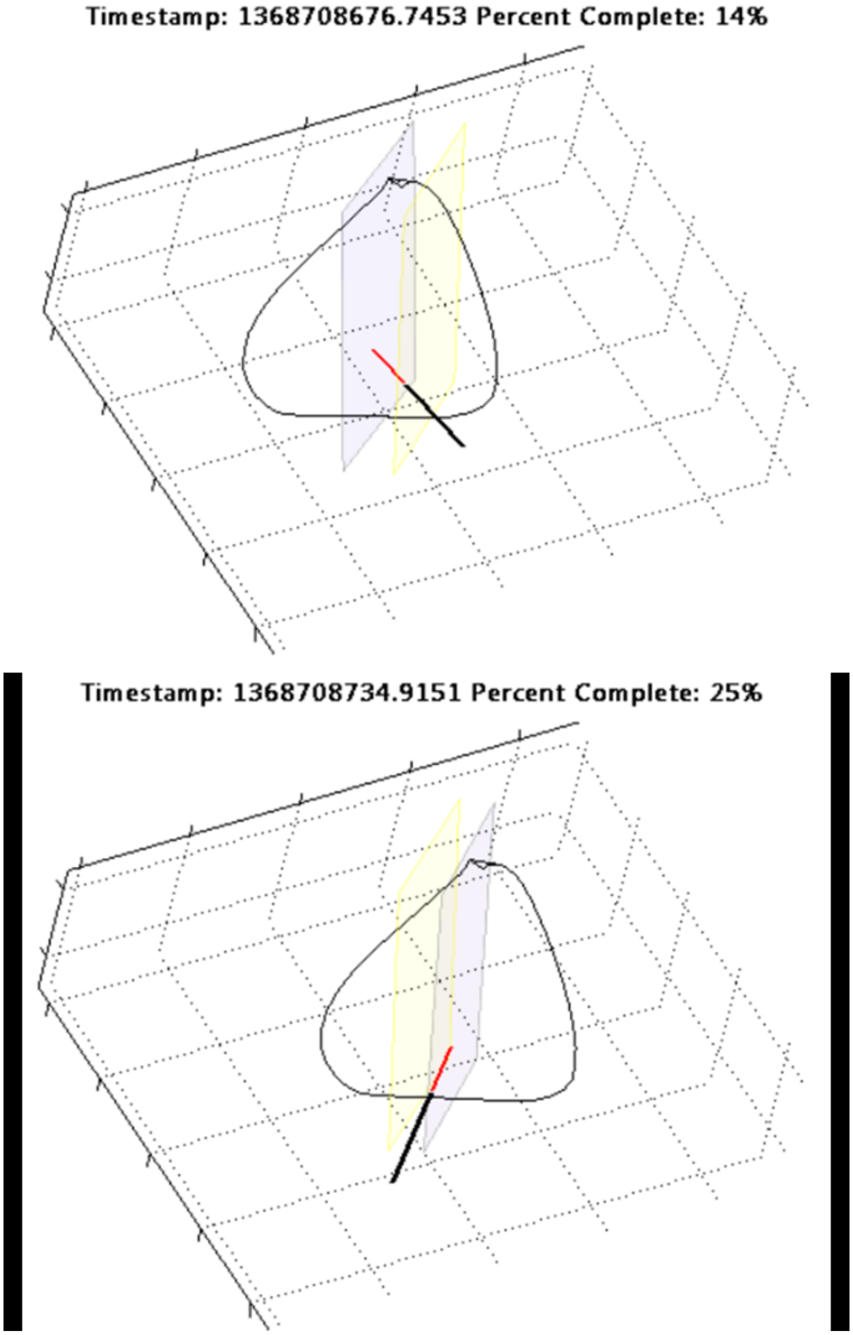
$$center_{rotated} = R \cdot center_{original} \tag{12}$$

Then, sum of square errors are computed using the formula for the distance from a point to a plane.

$$error = \sum_{i=1}^{N}\left(\frac{\left(n_{rotated}\cdot p_i - n_{rotated}\cdot center_{rotated}\right)}{\left|\left|n_{rotated}\right|\right|}\right)^2 \tag{13}$$

Where N is the number of extrema for which an optimal rotated plane is to be found. This process is repeated for each detection of a local minimum.

The result of the algorithm showed a more stable and seemingly accurate representation of the septum plane. See Figure 25.



Timestamp: 1368708676.7453 Percent Complete: 14%

Timestamp: 1368708734.9151 Percent Complete: 25%

Figure 25: Stability of Constrained Extrema approach. Yellow translucent plane shows initial septal plane estimation. Purple translucent plane shows septal plane estimation from constrained extrema approach. The top and bottom images refer to different time points in the same surgery.

The accuracy of this plane estimation was further confirmed when the addition of a reference sensor was used. The estimated plane closely tracked the location and orientation of transformed initial plane when the measured head movements were compensated for. To test the validity of the plane estimation, the estimated plane was compared to the plane generated after compensation using the reference sensor. The estimated plane was on average 3.4 degrees and 7 mm off the real plane introducing ~6% error to the feature vector.

5.4 Reference Sensor Compensation

The challenge of dynamic plane estimation was discovered while data collection was ongoing. Thus, to make dynamic plane estimation more reliable for the data collected in the future and to test the estimation methods, a second EM sensor was affixed to the towel located on the patient's forehead for surgeries occurring on or after September 9th, 2013. The data from this reference sensor was included in the same raw data CSV file in each surgery-specific subdirectory.

Using this data, much more reliable dynamic plane estimation could be obtained. This sensor was fully independent of tool motion and thus the plane location could be determined even when the tool was not in use. To incorporate the reference sensor information, first the initial reference sensor position, $p_{0,head}$, and orientation, $R_{0,head}$, was captured at the time when the nose perimeter circling was occurring. The normal vector, n, of the initial septal plane estimate was transformed to be from the world coordinate frame to be with reference to the reference sensor, i.e.

$$n_0^{ref} = R_{0,head}^{-1} \cdot \left(n_0^{world} - p_{0,head}\right) \tag{14}$$

The same was done for the estimated center of the nose, c:

$$c_0^{ref} = R_{0,head}^{-1} \cdot \left(c_0^{world} - p_{0,head}\right) \tag{15}$$

Then, for each subsequent frame, the transformed initial plane in reference coordinates was again transformed back to the world coordinates i.e.:

$$n^{world}(t) = R_{head}(t) \cdot n_0^{ref} + p_0(t) \tag{16}$$

$$c^{world}(t) = R_{head}(t) \cdot c_0^{ref} + p_0(t) \tag{17}$$

In this way, each data frame has an associated septal plane that reflected the most recent reference sensor information.

5.5 Automatic Stroke Detection

Observation of the tool motion showed that a stroking motion dominated much of the surgical activity. The motion of trying to elevate the mucosa from the underlying cartilage was repeated throughout the surgery in different regions of the septal plane. Thus, if information regarding the strokes could be extracted, the relevant information regarding how the surgery was performed could be maintained while the overall complexity of the data could be greatly reduced.

The strokes were too small, too frequent, and often obstructed and thus could not be manually annotated in an accurate or efficient manner. Observation of these strokes, however, showed the potential for their automatic detection from the kinematic data. The beginning of a stroke seemed to be characterized by the occurrence of a local minimum in the distance from the active tool tip to the septal plane during Cottle motion. Similarly, the end of a stroke seemed to be characterized by the occurrence of a local maximum in the distance from the active tool tip to the septal plane during Cottle motion.

To encode these observations algorithmically, an approach similar to the extrema approach for dynamic septal plane estimation was used. First, since an estimation of the location of the initial septal plane was known, the distance of the active tool tip to the septal plane could be calculated. Even though an estimate of the dynamic location of the plane could be determined, much of the head movement occurred between strokes as that is when the tool was in contact with the physical nose. Thus, computing extrema of the distance to a moving plane would be unreliable and thus only the initial plane was used. Once the vector of the distance from the active tool tip to the initial plane was determined, the vector was smoothed using a moving average filter to reduce false extrema caused by noise or jitter in the sensor readings. Next, the extrema of this vector were found. Next, the extrema had to be filtered to find only those involving strokes. This was done by taking each consecutive set of extrema, defined as the first (start of the stroke) and second (end of the stroke) and imposing a few conditions on the pair of extrema:

1) Both extrema must have occurred while the Cottle was active.
2) The distance of the first extrema to the initial plane must be less than the distance of the second extrema to the initial plane
3) The duration of the stroke must be at least 0.15 seconds

4) The duration of the stroke must be at most 1.1 seconds
5) The Euclidean distance between the tool tip at the start and end of the stroke must be at least 1.2 mm
6) The Euclidean distance between the tool tip at the start and end of the stroke must be at most 20 mm
7) The Euclidean distance from the beginning of the stroke to the initial nose center must be less than 350 mm
8) The Euclidean distance from the end of the stroke to the initial nose center must be less than 350 mm
9) The distance of the beginning of a stroke to the beginning of the previous stroke must be less than 250 mm

The parameters in these conditions were determined using a combination of intuition and trial and error and were validated by observing the visualization of detected strokes. Most of the imposed conditions were to reduce the number of 'false positive' strokes. This was done by limiting the duration of a stroke, the length of a stroke, the set of the possible locations of a stroke, and enforcing that consecutive strokes are not too distant.

The result of the stroke detection algorithm was successful with a majority of strokes being accurately detected and most non-strokes being ignored. Since a stroke itself has no explicit definition with regards to the surgery, it is impossible to define a notion of stroke detection accuracy. Despite the imposed conditions, some strokes were not detected and some movements were falsely attributed to be a stroke. A representative image of a detected stroke can be seen in Figure 26.

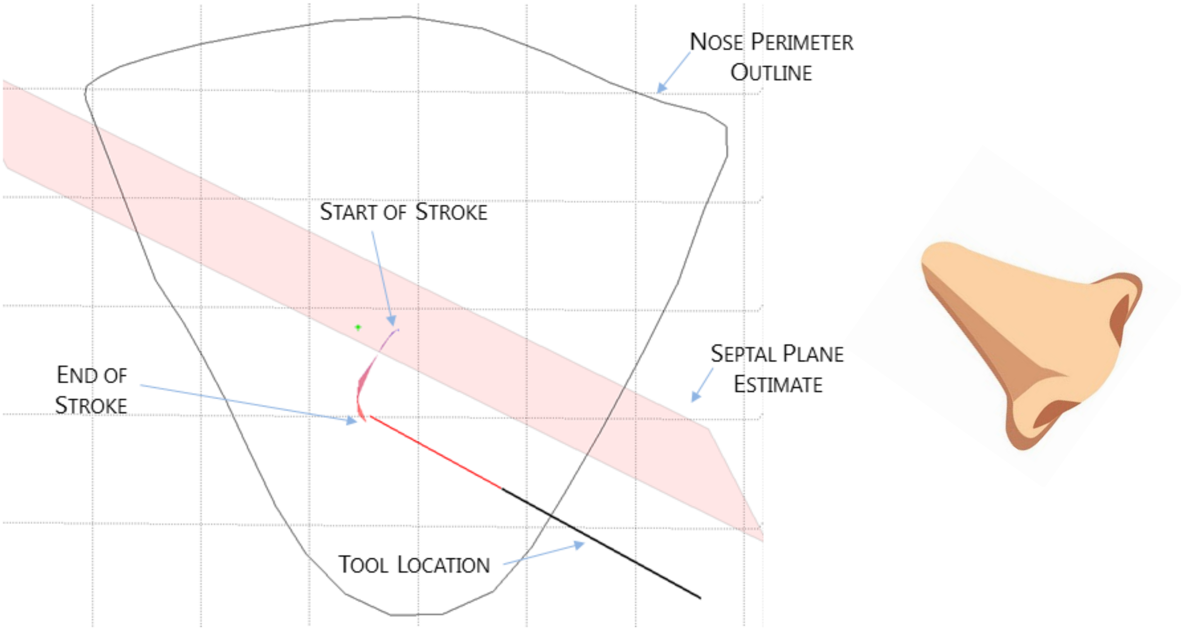


Figure 26: (Left) Visualization of a detected stroke (Right) Illustration of Orientation of Nose

6 Visualization of Tool Motion

The small surgical site of septoplasty and the visual obstruction caused by the arrangement of the operating surgeon and tool during surgery makes observation of the tool motion impossible. Furthermore, the complexity of 3D motion cannot be easily understood via the noisy raw data. To better understand the tool motion and the surgery itself, to evaluate the effectiveness of algorithms, and to offer trainees an improved medium from which to learn, multiple visualization tools were developed to see how the surgeon’s tool interacts with the patient. These tools include a projection of the tool motion onto the septal plane and an offline 3D visualization of tool motion and past tool trajectory. Further, to help trainees while they are observing surgeries, the

3D visualization was modified to be operable in real-time during the actual surgery.

6.1 Projection onto the Septal Plane

One of the hypotheses regarding differences in how expert and novice surgeons perform septoplasty was that expert surgeons cover the area of the septum in a more efficient manner than do novices. To analyze this, the tool motion of the surgeon was projected onto the estimate of the septum plane. Given the estimate of the septal plane at time t, the projection $p(t) = [p_x(t); p_y(t)]$ onto the septal plane can be given by:

$$p_x(t) = (T(t) - c) \cdot x_{axis} \tag{18}$$

$$p_y(t) = (T(t) - c) \cdot (n(t) \times x_{axis}) \tag{19}$$

where $T(t)$ is the 3D tool tip position, c is the origin of the plane, x_{axis} is the vector that determines the orientation of the x-axis, and $n(t)$ is the normal to the plane at time t. Thus, for each data frame for which the Cottle tool was active, the tool tip was projected to the estimated septal plane at the associated time frame and then plotted.

Through discussions with surgeons, we found that the most intuitive orientation of the plane was to have the x-axis aligned with the nose height. See Figure 27.

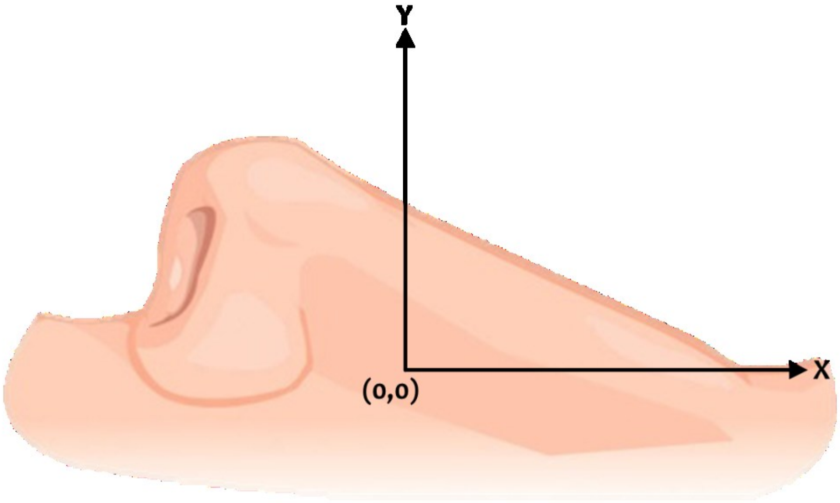


Figure 27: Orientation of Nose relative to Projection Visualization

To incorporate time information in the plot, a color gradient was used where the first projections of tool motion started as red. As the surgery progressed, the projections were shown to be increasingly blue and then, from the middle of the surgery to the end of the surgery, the projections transitioned from blue to green.

Examples of these projections plots can be seen for an expert surgeon in Figure 28 and a novice surgeon in Figure 29.

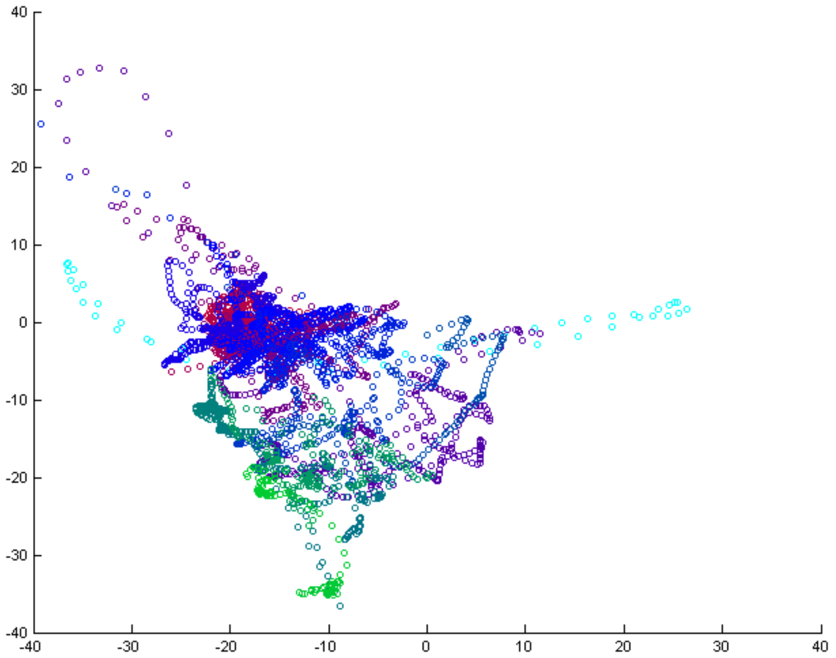


Figure 28: Projection plot of tool motion for an expert surgeon. Axes are in mm. The light blue dots mark the initial nose perimeter outline.

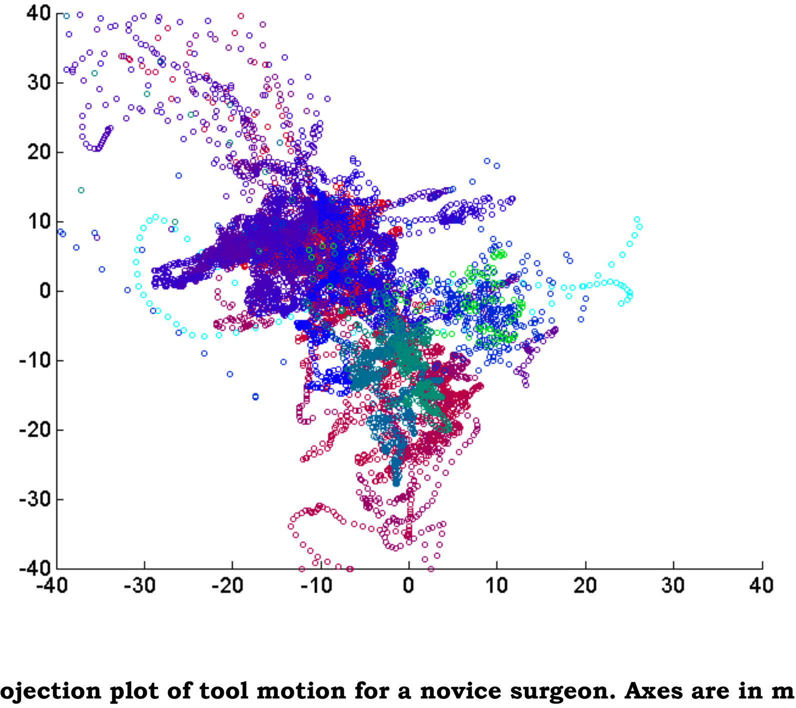


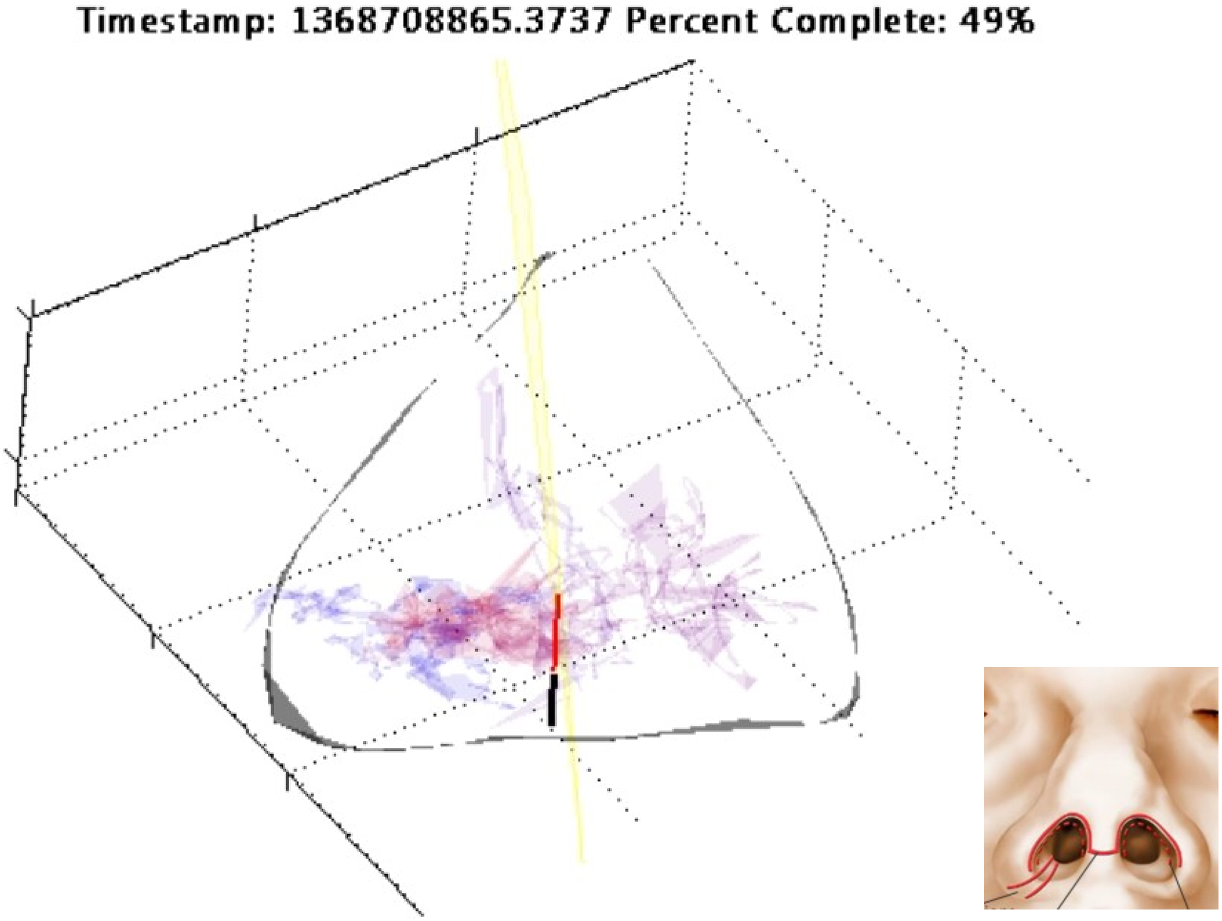
Figure 29: Projection plot of tool motion for a novice surgeon. Axes are in mm. The light blue dots mark the initial nose perimeter outline.

Cursory analysis of the projection plots shows a few trends. Experts tend to operate more quickly and thus have fewer projection points. Both experts and novices seem to focus activity in one localized area and then proceed to another area of the septal plane. While informative, a richer visualization tool is needed to gain more insight.

6.2 Offline 3D Visualization

The previous projection plots remove essential depth information and do not show how the tool was moving. To offer a richer, more informative visualization tool, a MATLAB script was developed to plot a simulated tool and the past tool trajectory with an overlay of the nose perimeter outline and estimate of the septal plane. Using this plot, observers can understand how the operator is moving the tool relative to the septum and also rotate the current 3D perspective interactively.

First, the window of the plot was limited to 40 mm from the center of the initial nose outline in all coordinate directions. The nose perimeter outline was marked in black using the fill3 command in MATLAB. To get a sense of the orientation of the Cottle elevator when in use, information regarding both sides of the tool was used. More specifically, the tool at time *t* was drawn as a 3D ray connecting the active tool tip to the inactive tool tip. The first quarter of the drawn tool that was closest to the active tool tip was drawn in red and the remaining three quarters of the tool tip closest to the inactive tip was drawn as a thicker black line. The estimate of the septal plane was drawn by extending out in all directions from the nose center along the 1st and 3rd principal components of the nose perimeter. The tool trajectory was drawn as translucent triangles connecting sets of three consecutive raw data tool tip points. The colors of these triangles were drawn as a gradient to give a notion of time. The current timestamp and percent of the surgery that has been completed by that timestamp is also displayed as the title of the plot. See Figure 30.



Timestamp: 1368708865.3737 Percent Complete: 49%

Figure 30: Example screenshot of offline 3D visualization tool. Estimated septal plane shown in yellow. (Bottom-Right) Illustration of Orientation of Nose. The translucent red, blue, and purple triangles indicate the past tool trajectories made by the tool-tip (shown as a black-red line).

6.3 Real-Time 3D Visualization

To overall goal of this project is to enhance the training process for trainee surgeons. The visual obstruction during the observation of live surgeries hinders the ability for the trainee to see how attending surgeons manipulate the tool with respect to the relevant anatomy. Thus, a real-time visualization system in the operating room could be valuable to trainees. Further, the visualization system could perform real-time analysis of the surgery as it happens and offer feedback, assessment, or general summarizing statistics.

Implementing a real-time data visualization system comes with a number of challenges. The system must communicate with the data collection software. For this reason, the system must be highly stable or maximize independence from the data collection so that a failure in the visualization system does not affect the data collection. Second, since the procedure involves real patients, the system must be non-obtrusive and it must limit the external involvement required by the surgeon. The system must fit in with the existing flow of the procedure. Third, the visualization must not distract the operators from performing the procedure. Fourth, the visualization must be fast and reliable. The tool shown in the visualization must track closely with the actual tool so that it is useful.

The real-time data system was built by modifying the existing offline 3D visualization system. The User Datagram Protocol (UDP) was used to transmit the raw data to the visualization system. UDP allows applications to send messages, or datagrams, to others hosts without requiring prior communication and no handshaking dialogue. The use of UDP minimizes the interdependence between the data-collection software and the visualization system and thus allows data collection to continue even if the visualization system crashes. UDP does not guarantee the delivery or order of communication, but since the purpose is for real-time visualization, an occasional lost packet would not cause a noticeable defect and a delayed, out-of-sequence packet would be of little value anyway once a more recent packet has been received. Operation of the visualization system is in the form of a Graphical User Interface (GUI) to ease data collection for the data collector.

The set-up of the real-time visualization required the entering of the tool IDs in the meta file of the data-recording program and recording of the pivot values from the pivot calibration of the tool in use in the MATLAB script.

To improve visualization, the plot was made with reference to the nose to ensure that the visualization of the nose outline did not move with head movement. Since the reference sensor was independent of the tool sensor and because of the nature of UDP, each reference sensor data frame was not necessarily accompanied by a tool data frame. Thus, the most recent reading of the reference sensor at the time of computation was used. To transform the raw tool tip data, first the pivot information was applied using:

$$T_{spoon}^{world}(t) = R_{tool}(t) \times p_{spoon} + s_{tool}(t) \qquad (20)$$

$$T_{flat}^{world}(t) = R_{tool}(t) \times p_{flat} + s_{tool}(t) \qquad (21)$$

where $T_{spoon}^{world}(t)$ is the 3x1 tool tip position vector at the spoon side tip in the world frame and $T_{flat}^{world}(t)$ is the 3x1 tool tip position vector at the flat side tip in the world frame, $R_{tool}(t)$ is the 3x3 rotation matrix referring to the orientation of the raw tool sensor, p_{spoon} is the 3x1 pivot vector for the spoon side and p_{flat} is the 3x1 pivot vector for the flat side, and $s_{tool}(t)$ is the raw 3x1 position vector of the origin of the tool sensor at time t. To bring the tool tips into the reference frame of the reference sensor, the following equations were used:

$$T_{spoon}^{reference}(t) = R_{reference}^{-1}(t) \times \left(T_{spoon}^{world} - s_{reference}(t)\right) \qquad \textbf{(22)}$$

$$T_{flat}^{reference}(t) = R_{reference}^{-1}(t^{*}) \times \left(T_{flat}^{world} - s_{reference}(t^{*})\right) \qquad \textbf{(23)}$$

where $T_{spoon}^{reference}(t)$ is the 3x1 tool tip position vector at the spoon side tip in the reference sensor frame and $T_{flat}^{reference}(t)$ is the 3x1 tool tip position vector at the flat side tip in the reference sensor frame, $R_{reference}(t)$ is the 3x3 rotation matrix referring to the orientation of the raw reference sensor, and $s_{reference}(t)$ is the raw 3x1 position vector of the origin of the tool sensor at time t. t^{*} refers to the time at which the reference sensor reading closest to t took place. In this way, movements of the head would be seen as a movement in the tool tip.

Rather than showing the entire tool tip trajectory, only the most recent set of tool frames (set by a parameter) were shown. The tool was displayed in the same way as in the offline 3D visualization. The nose perimeter circling procedure was slightly altered. The GUI developed for the live visualization can be seen in Figure 31.

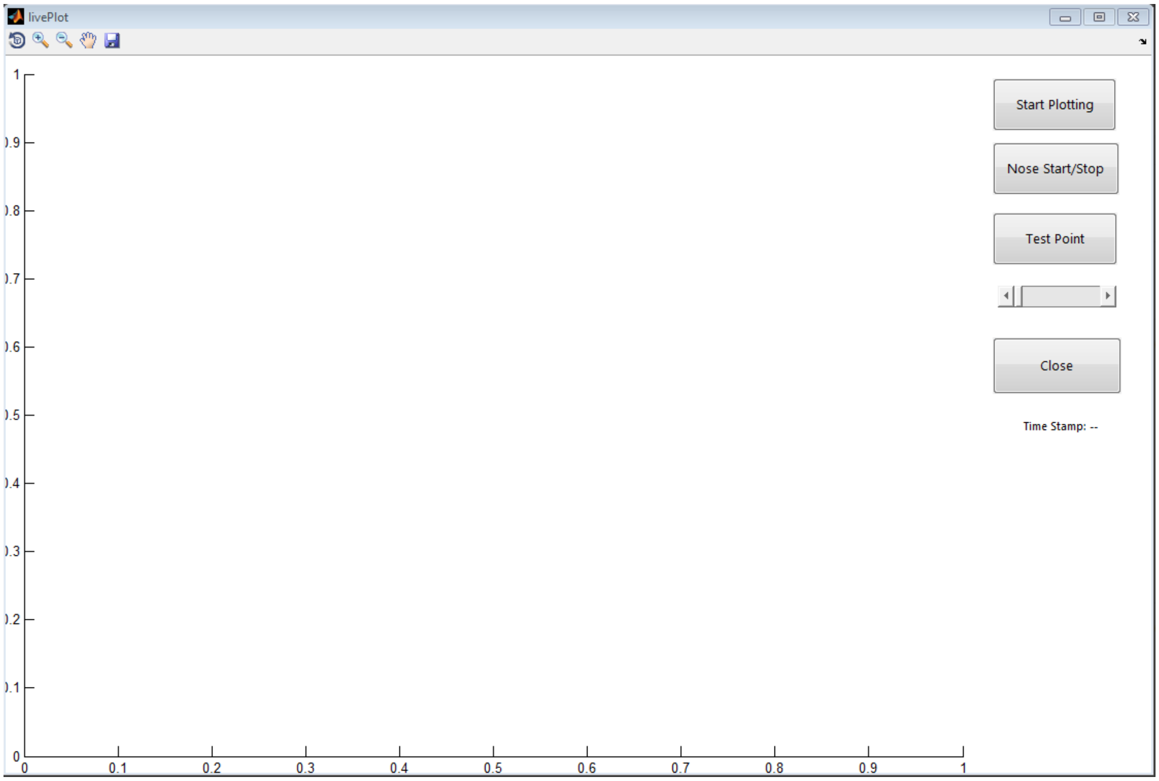


Figure 31: Screenshot of Real-Time Visualization GUI

To operate the GUI, the data collector sets-up the data recording system and clicks the 'Start Plotting' button. The tool motion and trajectory should then be displayed in a manner similar to the offline 3D visualization tool. Then, the operator is instructed to begin outlining the nose perimeter with the tool tip and the data collector clicks on the 'Nose Start/Stop' button. Once the operator has circled the nose 2-3 times, the data collector clicks the 'Nose Start/Stop' button once again. This will determine the initial estimate of the septal plane and plot the nose outline and the septal plane on the screen and center it. At this point, the operator proceeds with the surgery and can look on the screen to see the tool visualized in real-time with minimal lag and interference. The data collector can rotate the view of the scene by clicking and dragging after selecting the rotate tool in the toolbar of the GUI. Below, in Figure 32, is an image showing a test run of the 3D visualization on a real patient.

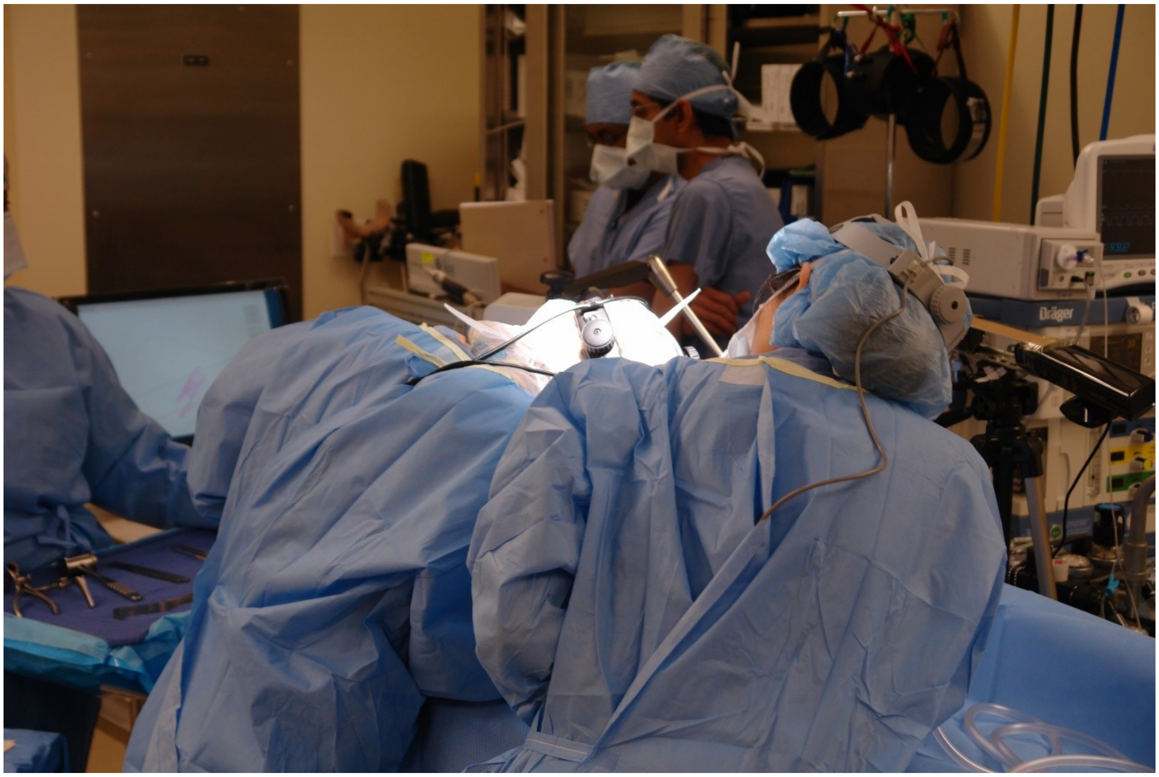


Figure 32: Picture of Real-Time Visualization System (back) in use during Nasal Septoplasty

Other simple metrics are also computed and displayed on the screen in real-time. These include elapsed time and current tool velocity.

7 Feature Computation

Once the raw data was processed to have readily accessible data regarding the active tool tip and the estimate of the current location of the plane, features could be computed. Features of interest would be those that have some ability to discriminate between a novice and expert surgeon. We define two general types of features: Stroke-Independent Features and Stroke-Based Features.

Stroke Independent Features are features that do not make direct use of the automated stroke segmentation. Stroke-based features are features derived from information contained in these automatically detected strokes.

Given the nature of live patient surgeries, features could not be dependent on the entire trial because a single surgery was often performed by multiple operators. Second, to incorporate real-time analysis of a feature, the feature would need to be able to be computed from a subset of data occurring by a given point in time. Third, the features would ideally be robust to noise and jitter in the sensors and be invariant to differences in the patient and to differences in operating style.

To handle the issue of multiple operators, a hierarchy was set-up. See Figure 33.

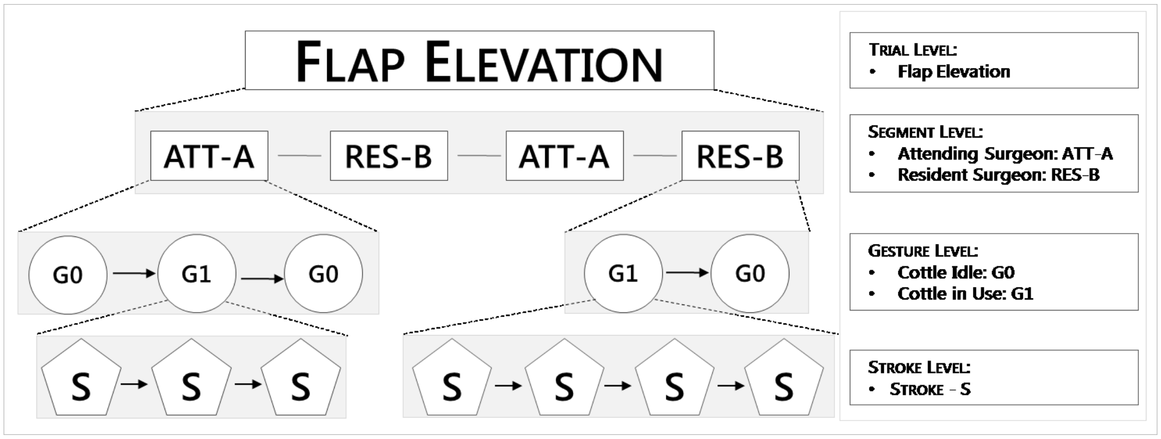
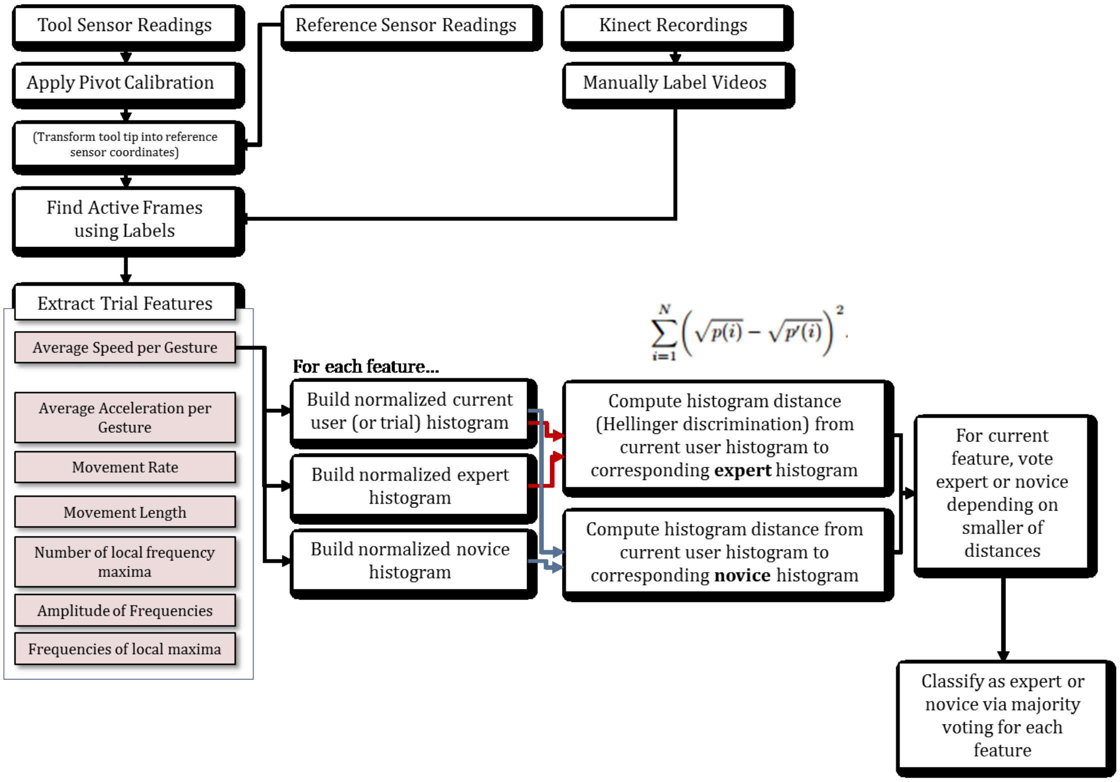


Figure 33: Diagram of Procedure Hierarchy showing the proposed structure of Nasal Septoplasty.

Each continuous instance of Cottle use was termed a gesture. Since each gesture was necessarily performed by a single operator, a feature could be computed at the gesture level and the combination of the gesture-level features could form a combined trial-level feature for a given individual.

7.1 Stroke-Independent Features

The earliest attempts at finding meaningful features involved building histograms of simple computations performed on gesture-level data. As seen in the data flow diagram in Figure 34, a histogram for each feature was built and compared to a reference novice histogram and a reference expert histogram built from training data.



Tool Sensor Readings
Reference Sensor Readings
Kinect Recordings
Apply Pivot Calibration
Manually Label Videos
(Transform tool tip into reference sensor coordinates)
Find Active Frames using Labels
Extract Trial Features
Average Speed per Gesture
Average Acceleration per Gesture
Movement Rate
Movement Length
Number of local frequency maxima
Amplitude of Frequencies
Frequencies of local maxima
For each feature...
Build normalized current user (or trial) histogram
Build normalized expert histogram
Build normalized novice histogram
$\sum_{i=1}^{N}\left(\sqrt{p(i)}-\sqrt{p'(i)}\right)^2.$
Compute histogram distance (Hellinger discrimination) from current user histogram to corresponding **expert** histogram
Compute histogram distance from current user histogram to corresponding **novice** histogram
For current feature, vote expert or novice depending on smaller of distances
Classify as expert or novice via majority voting for each feature

Figure 34: Data Flow diagram of Stroke-Independent Features

The features that were computed for each gesture were as follows: average speed, average acceleration, movement rate, mean movement length, number of local frequency maxima, the amplitude of those frequencies, and the frequencies of local maxima.

7.1.1 Average Speed and Acceleration

Average speed and acceleration was computed by smoothing each dimension of the active tool tip data using a moving average. The average speed, s_{avg}, was computed as:

$$s_{avg} = mean\big(s(t)\big);\ s(t) = \left(\frac{\|pos(t)\|_2 - \|pos(t-\Delta t)\|_2}{\Delta t}\right) \qquad \textbf{(24)}$$

Where t goes from the time of the start of a gesture to the time when that gesture ends. Average acceleration, a_{avg}, was computed as:

$$a_{avg} = \ mean\big(a(t)\big); a(t) = \frac{\|s(t)\|_2 - \|s(t-\Delta t)\|_2}{\Delta t} \qquad \textbf{(25)}$$

7.1.2 Number of Movements

Movements were defined and served as a pre-cursor to stroke detection. Movements were defined as the activity that occurs between subsequent extrema in the position data. They were computed by first using a moving average filter to smooth each dimension of the position data. Then, for each gesture, the movement rate, mr, was defined as:

$$mr = \frac{n_{movements}}{t_{end} - t_{start}} \qquad \textbf{(26)}$$

Where $n_{movements}$ is the number of movements occurring in a gesture, t_{end} was the time when the gesture ended, and t_{start} was the time when the gesture began.

7.1.3 Mean Distance

The mean distance, md, covered by each movement was computed as:

$$md = mean(d(i)); d(i) = \left\| p\left(t_{i,start}\right) - p\left(t_{i,end}\right) \right\| \qquad (27)$$

For all movements, $i,$ within a gesture where $p(t)$ is the tool tip position at time t, $t_{i,start}$ is the time of the start of the i^{th} movement and $t_{i,end}$ is the time at the end of the i^{th} movement. An example of a trial histogram for an expert and novice can be seen in Figure 35.

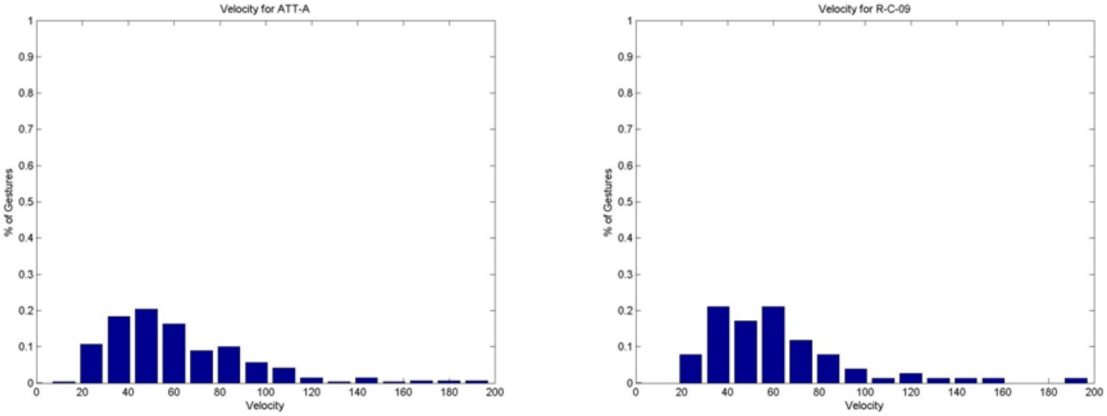


Figure 35: Expert Histogram (Left) and Novice Histogram (Right) of Speed.

7.1.4 Frequency-Domain Metrics

Frequency based metrics were based on the observation that movement occurs in an oscillatory fashion away from the septum. Thus, the raw data was smoothed in each dimension using a moving average filter. The raw data was projected to the 2nd principal component of points contained within a gesture. MATLAB's implementation of Fast Fourier Transform was applied to the projected data. The results of these plots showed substantial peaks at low frequencies (<1.5 Hz), but very small, dense peaks above this frequency. Thus, for the subsequent analyses, only frequencies less than 1.5 Hz were computed. The frequencies at which local maxima occurred, f_{max}, the amplitude of local maxima, A_{max}, and the number of local maxima, n_{max} were computed as metrics. Figure 36 shows an example of a histogram of A_{max} for an expert and a novice trial.

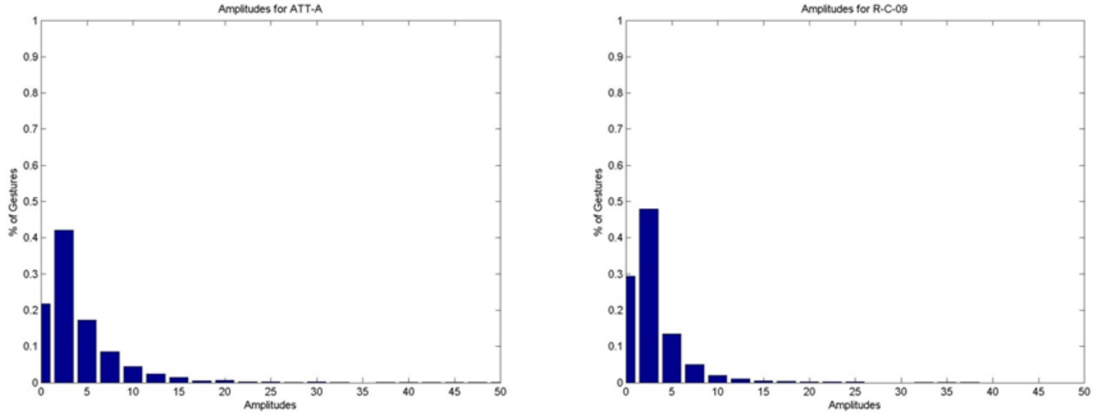


Figure 36: Expert Histogram (Left) and Novice Histogram (Right) of Amplitude of Local Maxima in the Frequency Domain.

7.1.5 Histogram Comparison

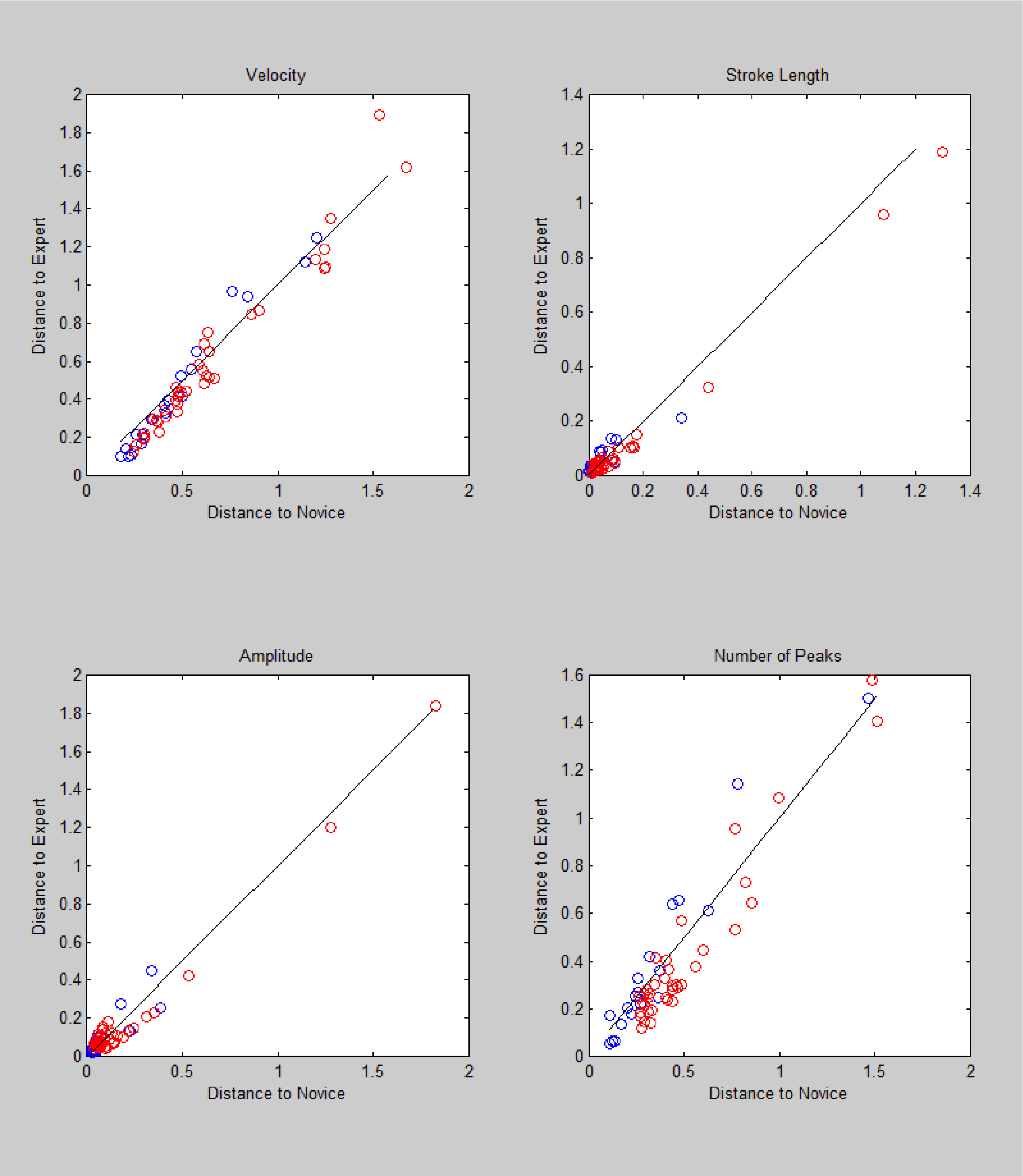
For every test trial by a user, a normalized histogram was built for each feature. In addition, a reference expert normalized histogram was built using all other expert trials and a reference novice normalized histogram was built using all other novice trials. For a given feature, the number of bins and the bin size of the histogram were kept constant. To classify whether the test trial was an expert trial or a novice trial, a distance metric comparing the histograms was constructed. One standard histogram distance metric is known as the Hellinger Distance [29]. For a histogram p and a second histogram p', the Hellinger Distance, d, is computed as:

$$d = \sum_{i=1}^{N} \left(\sqrt{p(i)} - \sqrt{p'(i)}\right)^2 \qquad (28)$$

Where i refers to the index of the i^{th} bin of the histogram. The larger the d, the greater the distance between the two compared histograms. Thus, for each feature, the distance of the test histogram to the reference expert histogram and the reference novice histogram was computed. Whichever distance yielded a smaller distance determined the “vote” of that feature. To combine all the “votes” of each feature into a single classification output, the majority decision of the participating voters was used.

The resulting histograms, as seen in part by the previous Figure 35 and Figure 36 did not show any clear, immediate difference among novice and expert operators. None of the feature histograms had a salient, distinguishing characteristic unique to an experience class. Thus, no meaningful information could be extracted and be relayed to trainees and their teachers regarding what makes an expert and novice surgeon. While visible differences could not be observed, the histogram comparison and classification could still be performed.

Figure 37 shows the comparison of distances of trials to the reference novice and expert histograms for each feature.


Velocity
Distance to Expert
Distance to Novice
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
0.5
1
1.5
2
Stroke Length
Distance to Expert
Distance to Novice
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
Amplitude
Distance to Expert
Distance to Novice
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
0.5
1
1.5
2
Number of Peaks
Distance to Expert
Distance to Novice
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
0
0.5
1
1.5
2

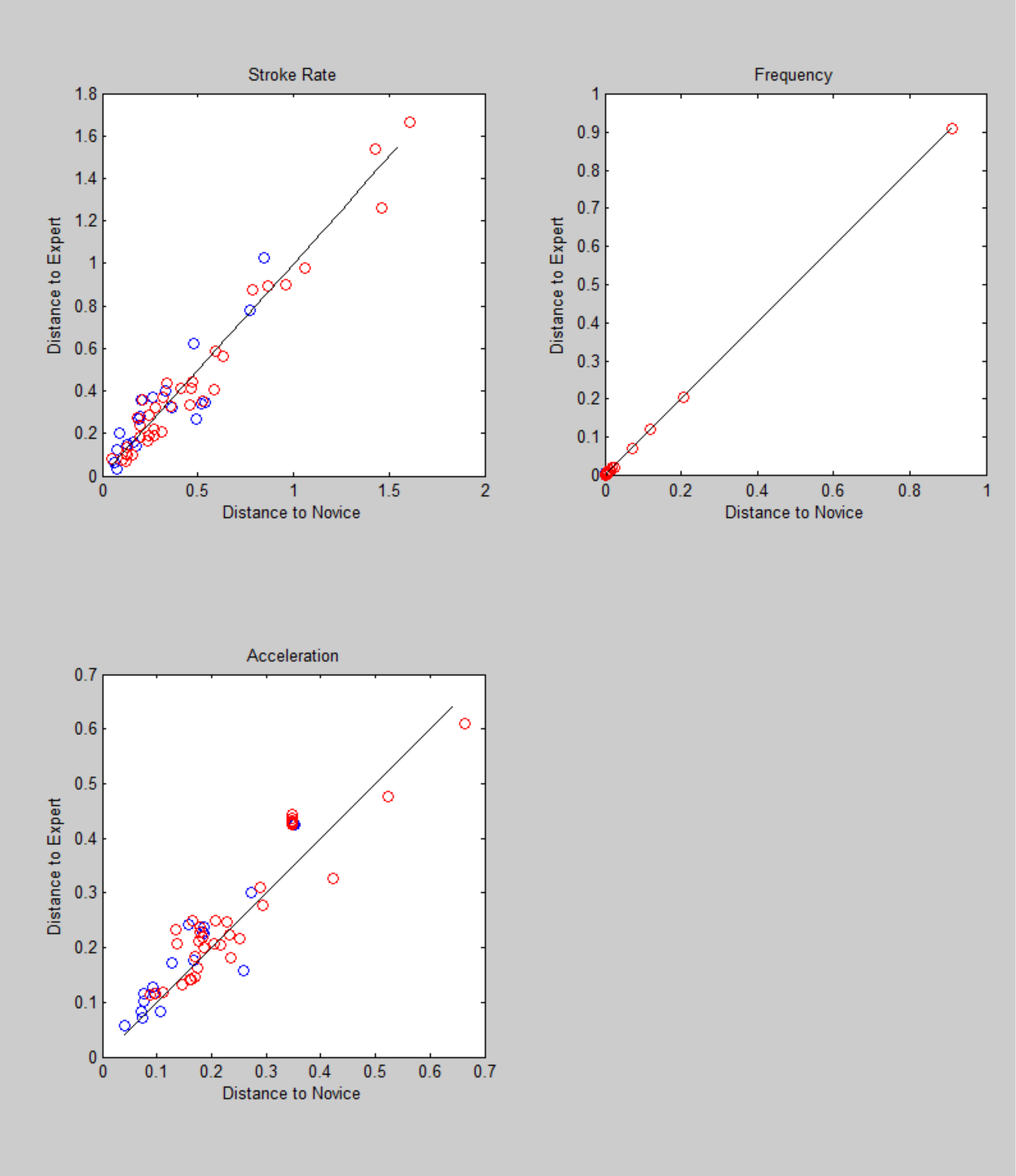
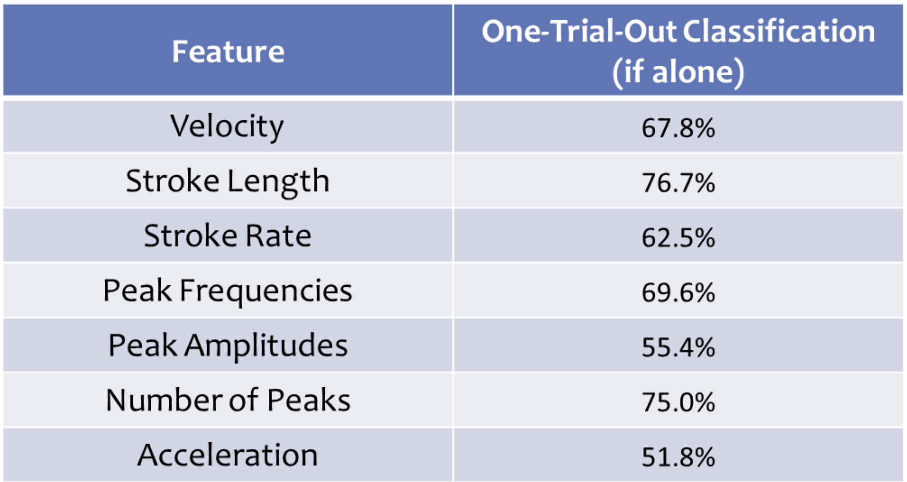


Figure 37: Comparison of distance of each trial to the reference novice histogram and to reference expert histogram for each feature. Red dots indicate true expert trials and blue dots indicate true novice trials.

Red dots in the figures above indicate true expert trials and blue dots indicate true novice trials. Ideally, red dots would be concentrated on the bottom-right of the graph and blue dots would be concentrated on the top-left of the graph. As we see, there is minimal separation in the distances. The feature that seems most correctly separated is in the number of frequency peaks, n_{max} and movement distance, *md*. In the leave-one-trial-out setup, we used data from one trial (at a time) as the test data and data from the remaining trials for training. In the leave-one-user-out setup, we used all trials performed by one surgeon as the test data and data from the remaining surgeons for training. This is reflected in the leave-one-trial-out classification results as seen in Table 2.

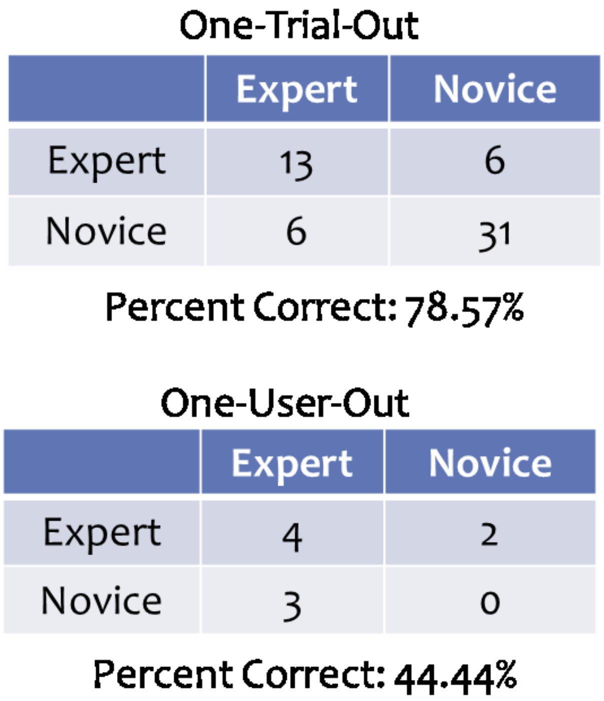
Table 2: Leave-One-Trial-Out Classification Rate if using only a single stroke independent feature



Feature	One-Trial-Out Classification (if alone)
Velocity	67.8%
Stroke Length	76.7%
Stroke Rate	62.5%
Peak Frequencies	69.6%
Peak Amplitudes	55.4%
Number of Peaks	75.0%
Acceleration	51.8%

Features were combined using a majority voting method. A classifier was built for each feature and the output from each classifier served as a single vote. Whichever classification received the most votes determined the overall classification. When combining features in this manner, the resulting confusion matrix is shown in Table 3.

Table 3: Leave-One-Trial-Out (top) and Leave-One-User-Out (below) confusion matrix using combined Stroke Independent Features



One-Trial-Out

	Expert	Novice
Expert	13	6
Novice	6	31

Percent Correct: 78.57%

One-User-Out

	Expert	Novice
Expert	4	2
Novice	3	0

Percent Correct: 44.44%

While the trial-out performance was sufficiently high, the one-user-out classification was very poor, yielding a worse-than-chance performance. This indicates that these metrics do not generalize across users effectively. While individual users may differ in these metrics in a distinguishable way, those differences are not maintained across different experience levels. Because of these poor results, more user-independent features needed to be computed.

7.2 Stroke-Based Features

Flap elevation in Septoplasty is inherently unstructured. There are no well-defined actions to take or a defined order of steps. The procedure is largely free-flowing and the operator operates under only general guidelines. Other surgical tasks, such as suturing tasks or laparoscopic cholecystectomy, however, are highly structured with a well-defined set of actions to be done in a specified order. Previously-developed language-model methods often made use of this structure in classification models. The unstructured nature of septoplasty, however, limited the power of the previously developed methods.

While inherently unstructured, the repetitive nature of the brushing motion of the tool away from the septal plane offered the opportunity for the introduction of a descriptive structure that we define as a stroke. To reduce the complexity of the three-dimensional motion data, the continuous motion data could be quantized into strokes. Because of the prevalence of strokes throughout the whole procedure, we hypothesize that the factors that separate a novice surgeon from an expert surgeon would manifest itself in how strokes are performed. Once the strokes have been computed, more meaningful and varied features can be computed from the strokes.

7.2.1 Type I and Type II Activity

The activity of septoplasty can be simplified into thinking about (1) brushing activity directed away from the septum plane characterizing the consistency of the surgeon's wrist motion and (2) activity along the septal plane characterizing the surgeon's coverage pattern. These two orthogonal activities both can give insight into how effectively a surgeon is completing the flap elevation task. Thus, stroke-based features were classified as either a Type I stroke-based feature if it revealed information regarding the tool as it brushed away from the septal plane or classified as a Type II stroke-based feature if it revealed information about how the surgeon covered the area of the septum. Figure 38 shows the difference between Type I and Type II activity.

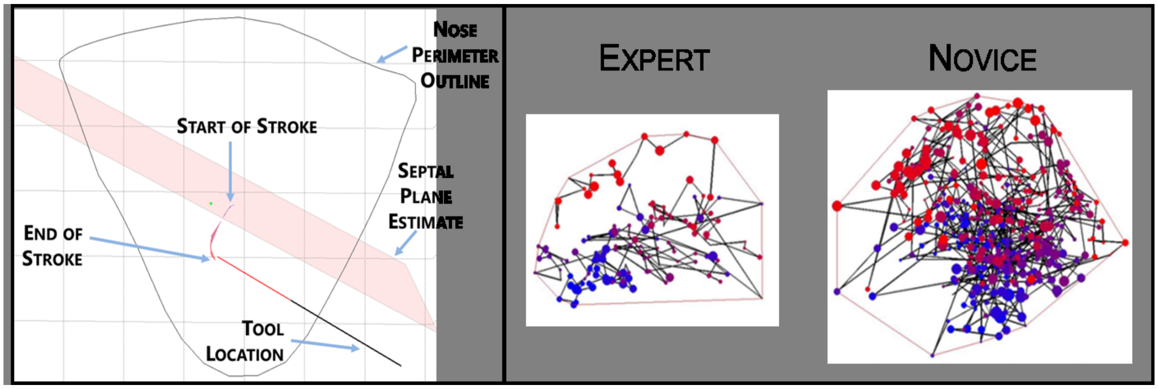


Figure 38: (Left) Type I Activity: 3D Visualization of a detected stroke brushing away from septal plane, (Right) Type II Activity: Expert and novice 2D search graphs on the septal plane. Color (blue to red) indicates progression of time. The vertex size is proportional the length of a stroke. The red outline marks the convex hull representing the area of septum covered by the surgeon.

7.2.2 Type I Activity Stroke-Based Features

Type I Activity stroke-based features include stroke distances, stroke trajectory lengths, stroke curvature, stroke durations, and stroke gaps, and regularity in stroke curvature and stroke gaps.

7.2.2.1 Stroke Distance

Stroke distance, sd, is a reformulation of the previously described Movement Distance. The difference is that strokes are more stringently filtered than movements were. Further, only the orthogonal distance to the septal plane is considered. For this feature, the distance of the active tool tip to the septal plane estimate at the beginning and end of a stroke is computed and the difference is computed. The distance, $d(t)$, of a point at time t, $p(t)$, to a plane defined by $ax + by + cz + f = 0$ where $n = [a; b; c]$ is given by:

$$d(t) = \frac{p(t) \cdot n + f}{\|n\|} \tag{29}$$

Thus stroke distance of a stroke that starts at t_{start} and ends at t_{end} is given by:

$$sd = d(t_{end}) - d(t_{start}) \tag{30}$$

7.2.2.2 Stroke Trajectory

Stroke trajectory is a measure of the arc length of the stroke. Visualization of a stroke showed that the strokes tend to be curved and thus the path length might be a better indicator than stroke distance. The stroke trajectory, st, was computed as the sum of the Euclidean distances between subsequent active tool tip positions, $p(t)$, that occurred during a stroke that started at t_{start} and ended at t_{end} i.e.

$$st = \sum_{t=t_{start}}^{t_{end-1}} \|p(t+1) - p(t)\| \tag{31}$$

7.2.2.3 Stroke Curvature

Stroke curvature, c, as seen in Figure 39 is the ratio between stroke trajectory length and the straight-line distance between the start and end of the stroke.

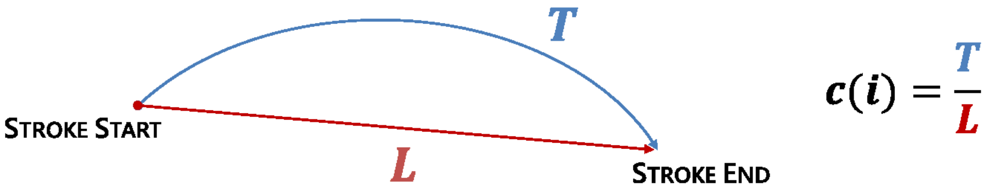


Figure 39: Simplified depiction of Curvature calculation

7.2.2.4 Stroke Duration

Stroke duration, Δt_{stroke} is simply the difference between the start time, t_{start} and end time, t_{end} of a stroke i.e.

$$\Delta t_{stroke} = t_{end} - t_{start} \tag{32}$$

Other time based metrics were computed such as stroke gap, Δt_{gap}:

$$\Delta t_{gap}(i) = t_{start}(i) - t_{end}(i-1) \tag{33}$$

And stroke start time differences, $\Delta t_{stroke\ starts}$:

$$\Delta t_{stroke\ starts}(i) = t_{start}(i) - t_{start}(i-1) \tag{34}$$

In addition to these base metrics, consistency across these metrics was also used.

7.2.2.5 Stroke Curvature Consistency

Since different segments of septoplasty require different stroke types, measures of local variance rather than the standard global variance were computed. Stroke curvature consistency (SCC) measures the local consistency of stroke curvatures across strokes. A consistent motion will yield strokes with similar curvatures and thus we expect experts to demonstrate a lower SCC than novice surgeons. We computed the curvature of the ith stroke, c(i), to be the ratio of the stroke trajectory length to the Euclidean distance between the start and end points of the stroke. Using the vector of all curvatures $\mathbf{C}$, we then measured the local consistency of curvatures by computing the median squared distance between C and its smoothed representation, $\hat{\mathbf{C}}$:

$$SCC = median\left((\boldsymbol{C} - \widehat{\boldsymbol{C}})^2\right),\ \ \widehat{\boldsymbol{C}} = median\ filter(\boldsymbol{C}) \qquad \textbf{(35)}$$

7.2.2.6 Stroke Duration Consistency

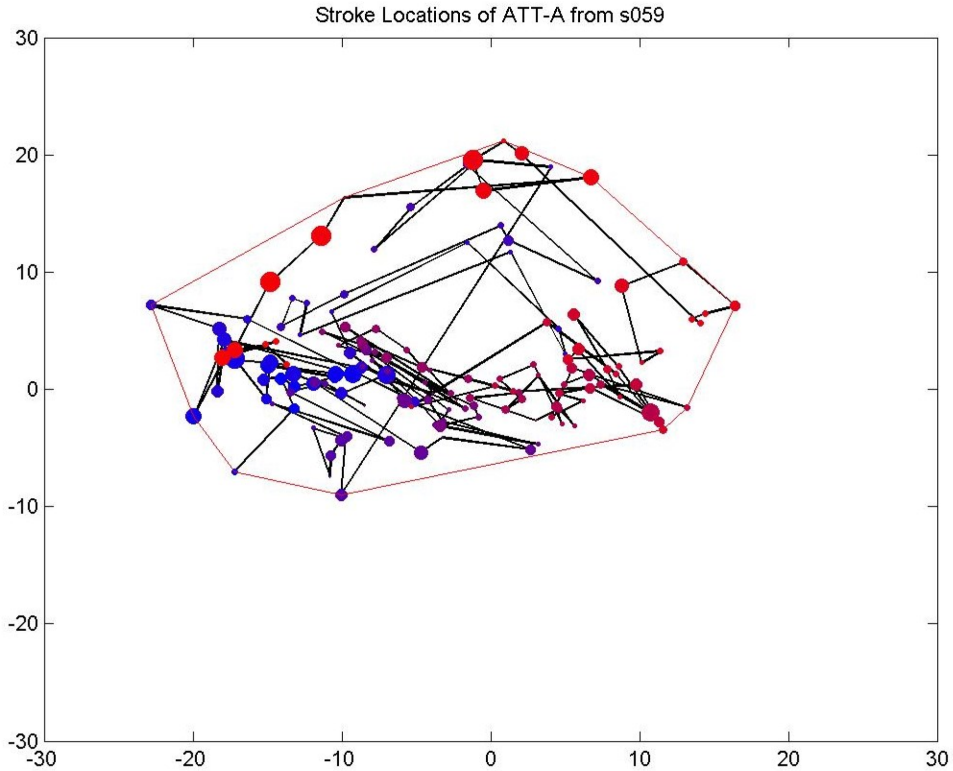
Similarly, stroke duration consistency (SDC) measures the local consistency in stroke duration:

$$SDC = median((\boldsymbol{\Delta}\mathbf{t}_{\mathbf{stroke}} - \boldsymbol{\Delta}\hat{\boldsymbol{t}}_{\boldsymbol{stroke}})^2);\ \ \boldsymbol{\Delta}\hat{\boldsymbol{t}}_{\boldsymbol{stroke}} = median\ filter(\boldsymbol{\Delta}\mathbf{t}_{\mathbf{stroke}}) \qquad \textbf{(36)}$$

where $\boldsymbol{\Delta t_{stroke}}$ is the vector of all stroke durations and $\boldsymbol{\Delta\hat{t}_{stroke}}$ is the vector of all stroke durations after application of a median-filter. Locally consistent stroke durations, hypothesized to be practiced by experts, will result in a low SDC whereas locally variable stroke durations, seen in novices, will result in a high SDC.

7.2.3 Type II Activity Stroke-Based Features

Type II Activity stroke-based features reflect how the operator covers the area of the septum. To help visualize the coverage pattern of operators, we built a visualization of the tool path that we call a search graph. Examples of a novice and expert search graph can be seen in Figure 40. The search graph is built by projecting the active tool tip position that occurs at the beginning of a stroke to the estimate of the septal plane at that time. The search graph is color coded to represent time. In other words, the color of the dot transitions from blue to red as the procedure progresses. The size of the dots are proportional to the distance of the stroke. The red line surrounding the graph represents the convex hull that correlates with the area of the septum that the surgeon had covered.



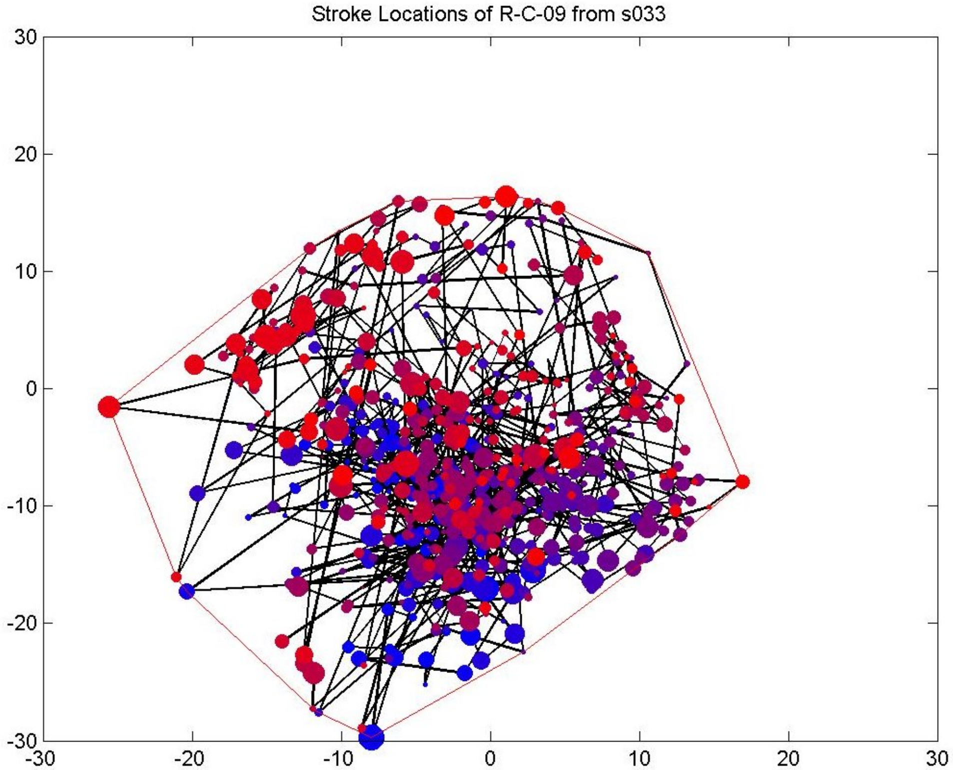


Figure 40: Search Graph of Expert (Top) and Novice (Bottom). Expert and novice 2D search graphs on the septal plane. Color (blue to red) indicates progression of time. The vertex size is proportional the length of a stroke. The red outline marks the convex hull representing the area of septum covered by the surgeon.

The features derived from the projection of stroke to the septum plane are the angle of movement, perimeter of the coverage region, area of the coverage region, and rate of increase in area of the coverage region.

7.2.3.1 Angle of Movement

To see the position of one stroke relative to the position of the previous stroke, the angle of movement, θ, was computed. The angle of movement was captured by:

$$\theta(i) = \cos^{-1}\left(\frac{v_1(i\,) \cdot v_2(i)}{\|v_1(i)\|\|v_2(i)\|}\right) \qquad (37)$$

$$where\ v_1(i) = p(i) - p(i-1); v_2(i) = p(i+1) - p(i)$$

7.2.3.2 Perimeter, Area, and Coverage Rate

In addition, features regarding the area of the entire search area were computed. The convex hull of a set of points X is defined as the smallest convex set that contains X. It can be visualized by imagining stretching a rubber band around the set of points X. For the computation of this metric, MATLAB's built-in convhull.m was used. Two measures on the convex hull were computed, perimeter and area. The perimeter, p, at the time of the i^{th} stroke was computed as the sum of the length of each of the n edges, e, in the convex hull:

$$p(i) = \sum_{p=1}^{n} \|e_p\| \qquad (38)$$

Coverage rate measures the rate of flap elevation and provides insight into how a surgeon covers the septum while searching for adhesions between the mucosal layer and the underlying septum. We defined area covered, $AC(i)$, as the area inside a convex hull of the search graph after completion of the i^{th} stroke. The convex hull covers the finite set of points, $S(i)$, consisting of the vertex $s_1$to vertex s_i of the search graph.

$$AC(i) = Convex\ Area\big(S(i)\big)\,; \quad S(i) = \{s_1, s_2, \ldots, s_i\}; \quad i \geq 3 \qquad (39)$$

We defined the coverage rate, CR, to be the median increase in AC with each stroke:

$$CR = median\big(\dot{\boldsymbol{AC}}\big);\ \dot{AC}(i) = AC(i) - AC(i-1)\ for\ i = 4\ to\ N \qquad (40)$$

where N is the number of vertices in the search graph. We hypothesized that the CR will be larger for experts compared with novice surgeons because experts elevate large areas of the mucosal flap with each stroke. In contrast, novice surgeons may fail to elevate the mucosal flap around adhesions and thus, they must re-elevate previously explored portions of the septum.

7.2.4 Stroke-Based Feature Results

The ability for a given metric to separate between novice and expert surgeons was determined by looking for reliable difference in plots of the metrics between the two experience groups. Based on discussions with Attending and Resident surgeons, we hypothesized how these features might differ for operators with different amounts of training and experience.

With regards to stroke distance, we hypothesized that experts would exhibit a larger average stroke distance than novices as they make fuller strokes that are directed away from the septal plane. Thus, we hypothesize experts come off the plane of the septum more than novices. Plots, illustrated by example in Figure 41, however, do not show any reliable difference between the two classes:

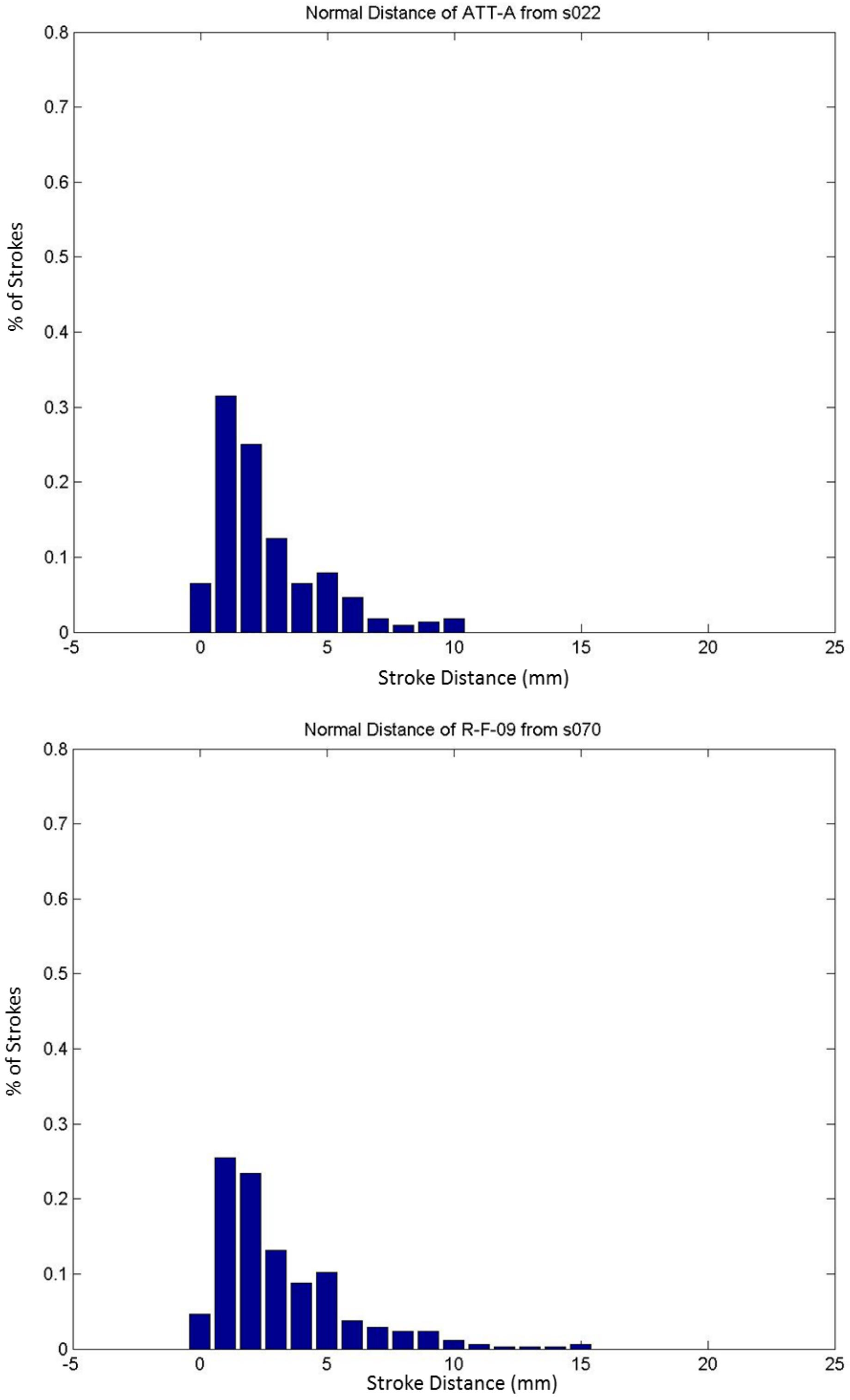


Figure 41: Normalized Histogram of Stroke Distance for Expert (Top) and Novice (Bottom)

With regards to stroke trajectory lengths, we hypothesized a similar phenomenon of experts showing longer trajectories than novices, but the plots again showed no reliable difference as seen by example in Figure 42.

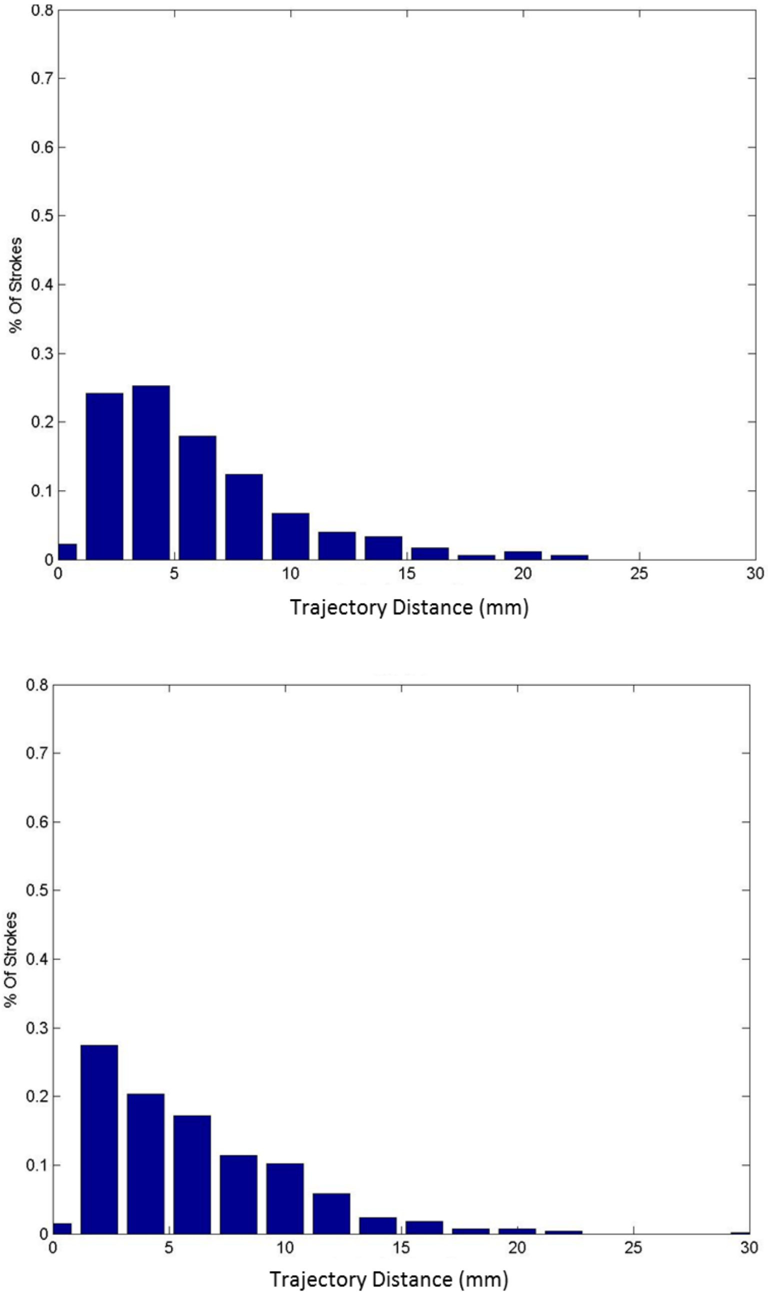


Figure 42: Normalized Histogram of Trajectory Distance for Expert (Top) and Novice (Bottom)

For stroke curvature, we expected the plot of the histograms to show lower variance for experts because we expected experts to be more regular in how they make strokes. Looking at the plots, as shown by example in Figure 43, however, the difference in variance was not immediately clear.

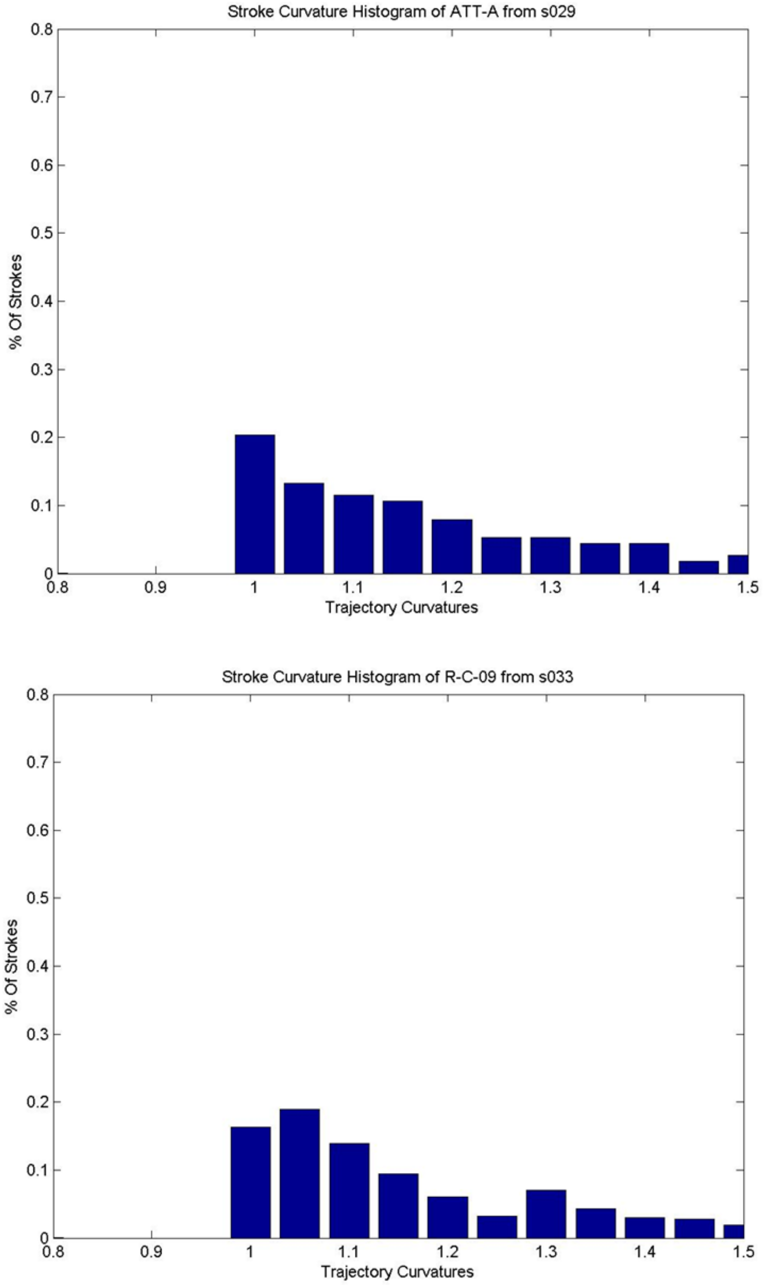


Figure 43: Normalized Histogram of Curvatures for Expert (Top) and Novice (Bottom)

To further examine the variance of stroke curvature, the local variance was examined using the previously described stroke curvature consistency metric. The result can be seen in Figure 44.

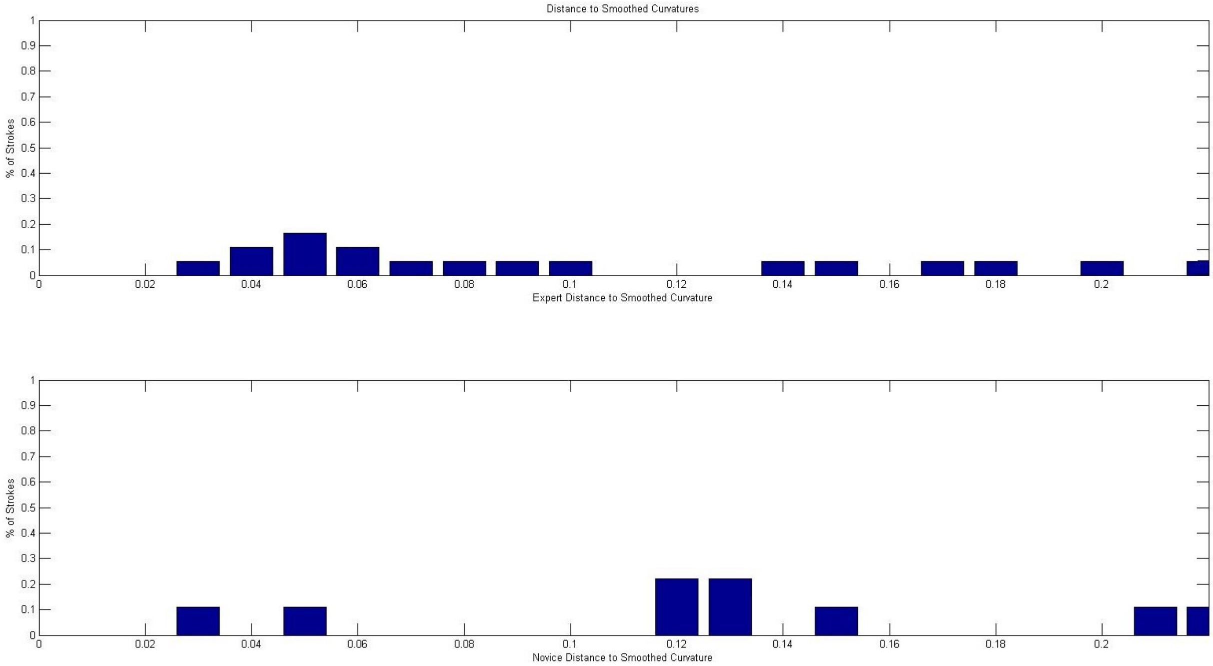


Figure 44: Normalized Histogram of Stroke Curvature Consistency across all Expert trials (Top) and Novice trials (Bottom)

We see that, in this plot, expert show a smaller distance to their smoothed curve which means that the vector of curvatures exhibited less local variance and thus that consecutive strokes of experts tended to be more consistent than those of novice. This finding was in line with our hypothesis and made stroke curvature consistency a potentially valuable metric.

Looking at stroke durations and stroke gaps, we expected to find increased regularity in experts. We hypothesized that experts perform more regular strokes and that this would be seen in low local variance in stroke durations and/or stroke gaps. Plots of these were inconclusive. The plot showing stroke duration consistency did show some differences between experts and novices, as seen by example in Figure 45.

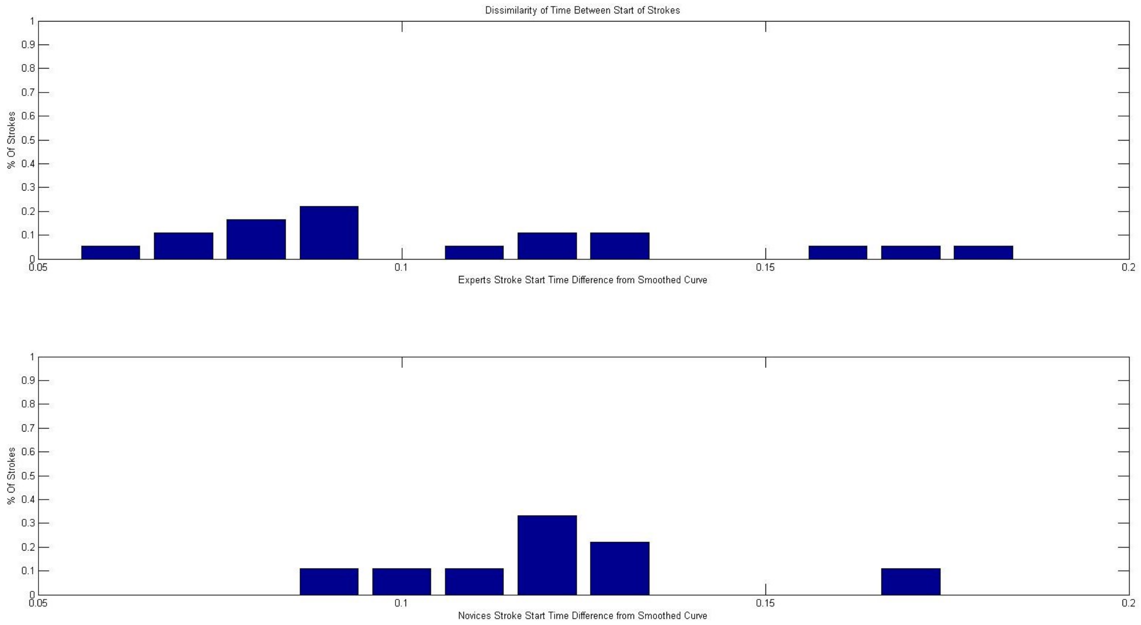


Figure 45: Normalized Histogram of Stroke Duration Consistency across all Expert trials (Top) and Novice trials (Bottom)

It seemed that experts tended to show either very low stroke duration consistency or very high stroke duration consistency and that novices were intermediate. While not conclusive, this metric showed potential for showing discrimination between operators. Stroke start differences and stroke gaps did not show reliable differences between classes of operators.

Looking at Type II Activity stroke-based features, we perform a similar analysis. We first look at angle of movement. We hypothesized that experts may exhibit broader angles as they would tend not to repeat strokes in locations they have already explored. Again, however, the histogram plots do not show reliable differences between experts and novices as illustrated by example in Figure 46.

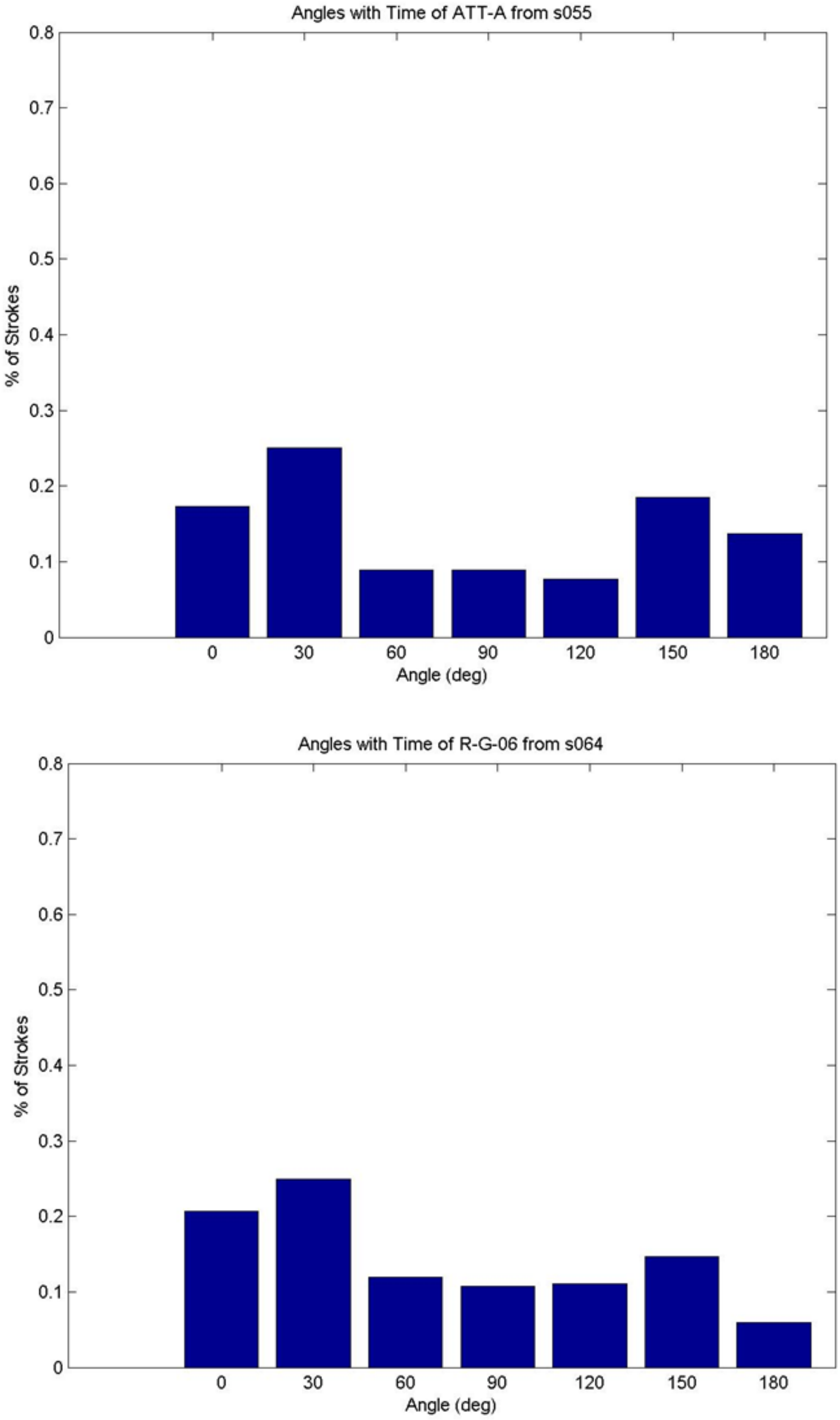


Figure 46: Normalized Histogram of Angle of Movement for Expert (Top) and Novice (Bottom)

We expected that experts are more efficient in covering the septum because they are more trained and empirical observations showed shorter procedure times for novices. We looked at the rate of increase in perimeter of the coverage region and rate of increase in the area of the coverage region. The graph of perimeter did not yield reliable differences seen in a representative example in Figure 47.

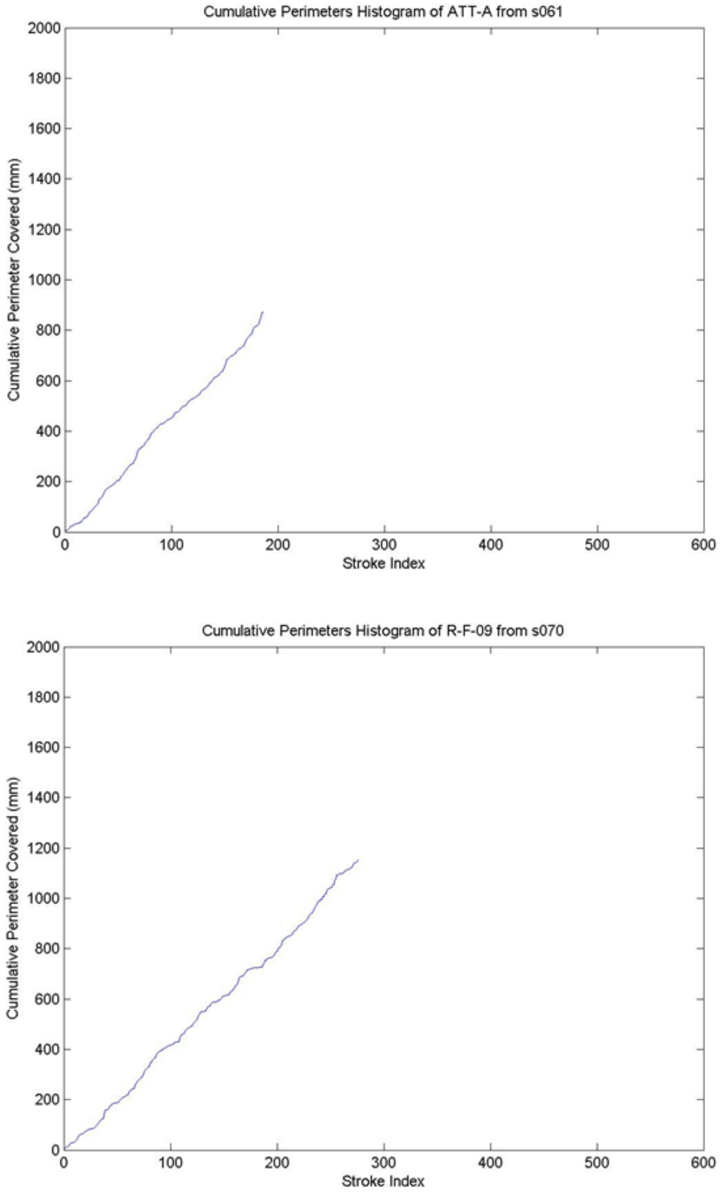


Figure 47: Plot of Cumulative Perimeter against Stroke Index for Expert (Top) and Novice (Bottom)

However, looking at the rate of increase in coverage area showed much more reliable differences as seen in Figure 48.

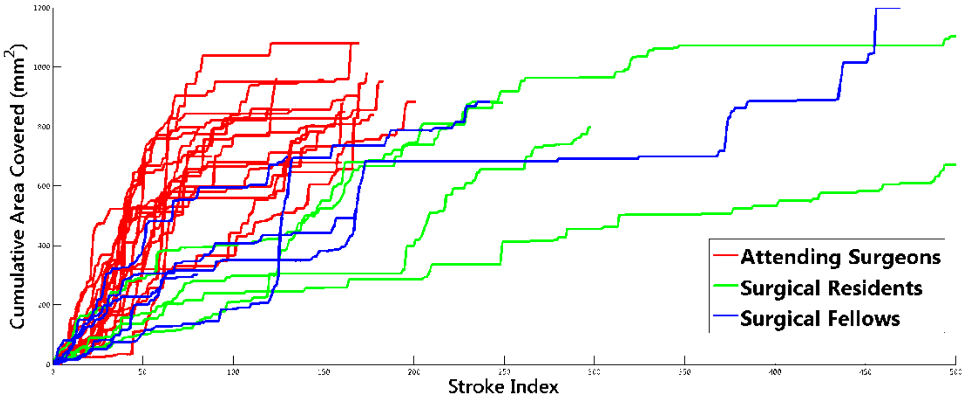


Figure 48: Cumulative areas for single-operator trials performed by attending and trainee surgeons for which at least seventy strokes were detected; CR is the median slope of each curve.

This plot shows a clear difference in the rate of area increase between novices and experts, showing that experts cover a greater area using fewer strokes.

The features that were determined to be relevant can be seen in box plots shown in Figure 49.

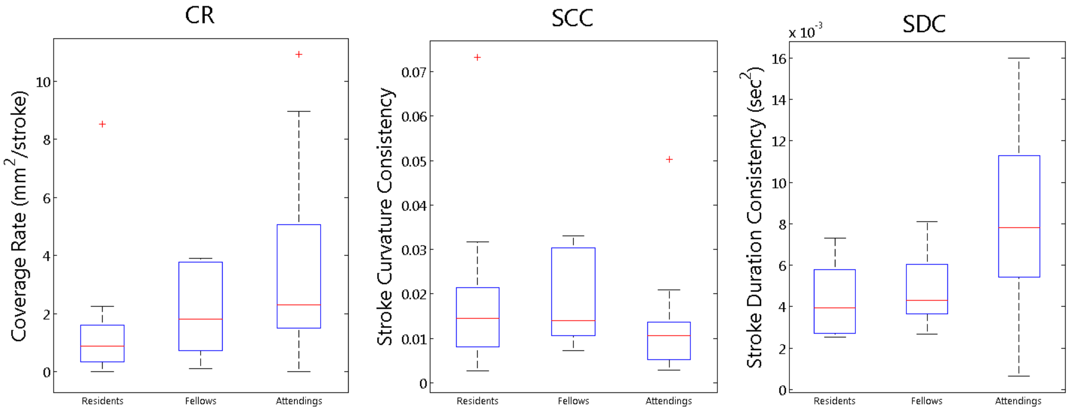


Figure 49: Box Plots of Coverage Rate (CR), Stroke Curvature Consistency (SCC), and Stroke Duration Consistency (SDC) for Residents, Fellows, and Attending Surgeons.

7.2.5 Stroke-Based Feature Classification Results

Once the features of interest were determined, they were computed for all trials and fed into a Support Vector Machine. The entire flow of data can be seen in Figure 50.

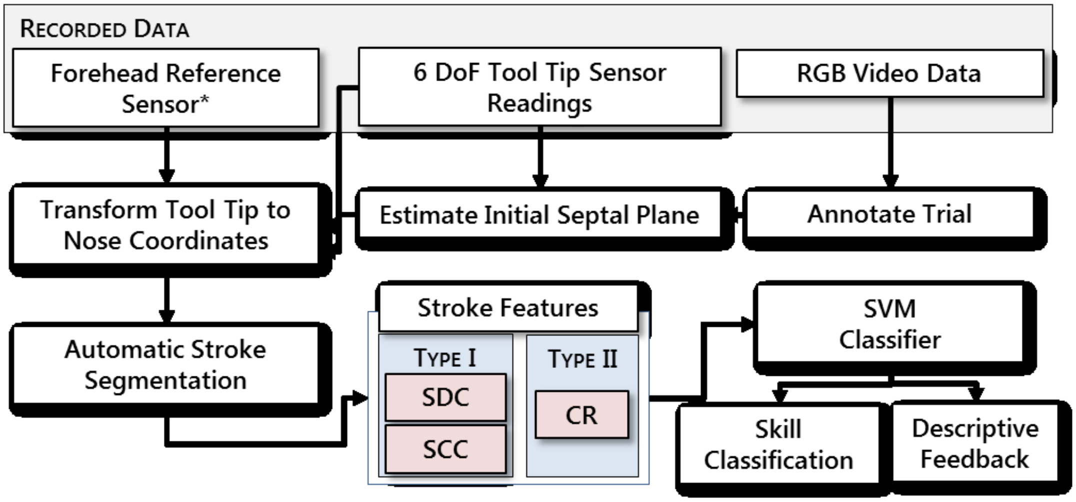

RECORDED DATA
Forehead Reference Sensor*
6 DoF Tool Tip Sensor Readings
RGB Video Data
Transform Tool Tip to Nose Coordinates
Estimate Initial Septal Plane
Annotate Trial
Automatic Stroke Segmentation
Stroke Features
TYPE I
SDC
SCC
TYPE II
CR
SVM Classifier
Skill Classification
Descriptive Feedback

Figure 50: System components and data flow for automatic skill classification in septoplasty- DoF: Degrees of Freedom, RGB: Red-Green-Blue, SCC: Stroke Curvature Consistency, SDC: Stroke Duration Consistency, CR: Coverage Rate, *For cases using a head sensor

A kernel support vector machine is a learning model that is often used for classification. It constructs a separating hyperplane between training data such that examples from separate categories are split to maximize the distance between the plane and the points of contention. Furthermore, the inputs are mapped to a higher dimensional feature space using the kernel trick. The MATLAB implementation of Support Vector Machines was used for the analysis. For our analysis, a three-dimensional vector was formed from each trial.

We trained and tested the SVM classifier under two setups: leave-one-trial-out (TO) and leave-one-user-out (UO). In the TO setup, we used data from one trial (at a time) as the test data and data from the remaining trials for training. In the UO setup, we used all trials performed by one surgeon as the test data and data from the remaining surgeons for training. Trials performed by two operators were considered to be two separate trials with all data from a single operator concatenated together. We excluded from our analysis all trials where fewer than seven strokes were performed in each sub-trial. We computed the micro-average (ratio of correctly classified samples to the total number of samples) and macro-average accuracy (average of true positive rates of each class) of our classifier as a measure of its ability to discriminate surgical skill. The results of the classification can be seen in Table 4 and Table 5:

Table 4: Classification performance (%) using leave-one-trial-out setup (chance=50%)

Only SCC		Ground Truth		**Only SDC**		Ground Truth		**Only CR**		Ground Truth		**Overall**		Ground	
		Exp	Nov			Exp	Nov			Exp	Nov	**(SCC, SDC, CR)**		Exp	Nov
Classifier	Exp	76.3	55.6	Classifier	Exp	50.0	5.6	Classifier	Exp	60.5	22.2	Classifier	Exp	55.3	11.1
Output	Nov	23.7	44.4	Output	Nov	50.0	94.4	Output	Nov	39.4	77.8	Output	Nov	44.7	88.9
Microaverage:		66.1		Microaverage:		69.6		Microaverage:		66.1		Microaverage:		69.6	
Macroaverage:		60.4		Macroaverage:		72.2		Macroaverage:		69.2		Macroaverage:		72.1	

Table 5: Classification performance (%) using leave-one-user-out setup (chance=50%)

Only SCC		Ground Truth		**Only SDC**		Ground Truth		**Only CR**		Ground Truth		**Overall**		Ground	
		Exp	Nov			Exp	Nov			Exp	Nov	**(SCC, SDC, CR)**		Exp	Nov
Classifier	Exp	89.5	55.6	Classifier	Exp	57.9	5.6	Classifier	Exp	65.8	22.2	Classifier	Exp	63.2	16.7
Output	Nov	10.5	44.4	Output	Nov	42.1	94.4	Output	Nov	34.2	77.8	Output	Nov	36.8	83.3
Microaverage:		75.0		Microaverage:		69.6		Microaverage:		69.6		Microaverage:		69.6	
Macroaverage:		67.0		Macroaverage:		76.2		Macroaverage:		71.8		Macroaverage:		73.3	

We define micro-average as the ratio between the number of correct classifications and the total number of test trials. We define a macro-average as the arithmetic mean of the percent accuracies of each ground-truth skill class. It is can be computed by taking the average of the main diagonal of each confusion matrix seen in Table 4 and Table 5. Our classifier discriminated between trials performed by expert and novice surgeons under both the TO and UO setups with an overall micro-average accuracy of 69.6% for both TO and UO and overall macro-average accuracy of 72.1% for TO and 73.3% for UO. The classification accuracy using each individual feature was similar to the accuracy obtained using all three features. Comparing these results to those from the stroke-independent features, we see a significantly higher UO average and comparable TO average even though only three features were used. Further, these features have more direct relation to how the surgeon operates the tool and insight into these features could be incorporated into training programs.

8 Conclusion

This thesis has described the development and implementation of an automated approach to the skill evaluation and visualization of an unstructured surgical task using tool motion data in the operating room for the purposes of improved surgical training. The work describes how data from the operating room during flap elevation in septoplasty is reliably collected, processed, and analyzed to produce actionable, meaningful metrics that can be used to classify the skill of the operating surgeon with an accuracy of near 72%. The developed features were based on surgeons' understanding of the procedure and are meaningful for providing feedback to trainees. For example, feedback may be focused on efficient strategies to search for adhesions between the mucosal flap and the underlying septal cartilage, or on practicing the stroke motion such that trainees learn to elevate larger areas using optimal force or to adapt to changes in patient anatomy.

Further, the work describes visualization tools, including a real-time visualization system that allows trainees and their instructors to better observe how experienced surgeons move the surgical tool with respect to patient anatomy and offers the possibility of incorporating real-time metric computation with accompanying suggestions and guidance.

Some of the limitations of the described work include a limited number of available cases and a limited number of expert operators. Further, there does not exist enough longitudinal data to conduct a rigorous examination of learning curves. Further, as it stands, the study relies on a self-proclaimed and experience-based ground truth rather than an objective measure of surgical skill. The data-collection, process, however is ongoing and most of these concerns will be alleviated when more data is available.

Extensions of this work include the development of additional metrics to improve the classification accuracy and to further improve the available actionable feedback to trainees. Future work also includes comparing the metric-based method of classification to language-based models such as Hidden Markov Models or Descriptive Curve Coding methods. Beyond that, future plans involve generalizing the data collection and analysis process so that it can scale more easily to more locations and more types of unstructured surgical tasks. This would involve automated annotation of procedures and a more robust, user-friendly data collection process. Further, it would involve developing a framework for testing new metrics that might be procedure-specific.

Regarding the real-time visualization system, the overall response by the attending surgeon and the resident surgeon regarding the system was promising. After testing the system during a live-patient septoplasty surgery, a number of potential improvements were identified. These included automatic resizing of the window, playback of the surgery, incorporating a touch screen monitor, having multiple views on the screen, incorporating the Kinect Video feed in the same window.

In addition to these improvements, further computation of metrics and the display of these metrics in real-time is also being explored. Once metrics that relate to surgical skill are identified, they can be computed in real-time and be presented to the operators. Further, the system can identify weaknesses or potential areas of improvement on the screen. General potential improvements of the visualization system include better integration to the surgical workflow and a more formal integration of the visualization system to the surgical training program.

The work to date has been submitted to the 2014 annual Medical Image of Computing and Computer Assisted Intervention conference. I would like to express my gratitude to Narges Ahmidi, Dr. Swaroop Vedula, Professor Gregory Hager, Dr. Masaru Ishii, Dr. Lisa Ishii, and Dr. Sanjeev Khudapur for their insight, time, guidance, and feedback.

The skill evaluation system and real-time visualization system presented in this thesis is part of the on-going Language of Surgery project. The project as a whole aims to create descriptive mathematical model to represent and analyze surgical training and performance. This project can have a measurable impact on the development of trainees and ultimately improve patient outcomes and human understanding of surgical skill and learning. The results can have broad implications for the evaluation and training of surgeons, measuring surgical proficiency, annotation of surgical recordings, and for other medical informatics applications. For the first time, the presented work on flap elevation in Septoplasty is the first Language of Surgery work in the realm of unstructured surgical motion tasks. Previous work focused on highly structured tasks such as in suturing and has used robotic surgical systems. The presented work shows that the underlying principles of the Language of Surgery are applicable to all types of surgical tasks and can provide valuable information and insight to all surgical settings.

Bibliography

[1] Polavarapu H, Kulaylat A, Sun S, Hamed O, 100 years of surgical education: The past, present, and future. The American College of Surgeons. 2013.

[2] Moller M, Karamichalis J, Chokshi N, Kaafarani H, Santry H, Mentoring the modern surgeon. The American College of Surgeons. 2008; 19-25.

[3] Bass BL. Fundamental changes in general surgery residency training. Am Surg. 2007 Feb;73:109-13.

[4] Sugden C, Aggarwal R. Assessment and feedback in the skills laboratory and operating room. Surg Clin North Am. 2010 Jun;90:519-33.

[5] Champagne, B.J.: Effective Teaching and Feedback Strategies in the OR and Beyond. Clinics in Colon and Rectal Surgery 26, 244-249 (2013)

[6] Reiley, C.E., Lin, H.C., Yuh, D.D., Hager, G.D.: Review of Methods for Objective Surgical Skill Evaluation. Surgical Endoscopy 25, 356-366 (2011)

[7] Mason JD, Ansell J, Warren N, Torkington J. Is motion analysis a valid tool for assessing laparoscopic skill? Surg Endosc. 2013 May;27:1468-77.

[8] Moorthy, K., Munz, Y., Sarkar S.K., Darzi, A.: Objective Assessment of Technical Skills in Surgery. BMJ 327, 1032-1037 (2003)

[9] Dosis, A., Aggarwal, R., Bello, F., Moorthy, K., Munz, Y., Gillies, D., Darzi, A.: Synchronized Video and Motion Analysis for the Assessment of Procedures in the Operating Theater. Archives of Surgery 140, 293-299 (2005)

[10] Ahmidi, N., Hager, G.D., Ishii, L., Gallia, G.L., Ishii, M.: Robotic Path Planning for Surgeon Skill Evaluation in Minimally-Invasive Sinus Surgery. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012, Part I. LNCS, vol. 7510, pp. 71–478. Springer, Heidelberg (2012)

[11] Zappella, L., Béjar, B., Hager, G., Vidal, R.: Surgical gesture classification from video and kinematic data. Medical Image Analysis (2013)

[12] Tao, L., Elhamifar, E., Khudanpur, S., Hager, G.D., Vidal, R.: Sparse Hidden Markov Models for Surgical Gesture Classification and Skill Evaluation. In: Abolmaesumi, P., Joskowicz, L., Navab, N., Jannin, P. (eds.) IPCAI 2012. LNCS, vol. 7330, pp. 167–177. Springer, Heidelberg (2012)

[13] Rosen, J., Solazzo, M., Hannaford, B., Sinanan, M.: Task decomposition of laparoscopic surgery for objective evaluation of surgical residents' learning curve using hidden Markov models. Computer Aided Surgery 7(1), 49–61 (2002)

[14] Dosis, A., Bello, F., Gillies, D., Undre, S., Aggarwal, R., Darzi, A.: Laparoscopic task recognition using hidden Markov models. Studies in Health Technology and Informatics 111, 115–122 (2005)

[15] Varadarajan, B.: Learning and inference algorithms for dynamical system models of dexterous motion. PhD thesis, Johns Hopkins University (2011)

[16] Kumar, R., Jog, A., Vagvolgyi, B., Nguyen, H., Hager, G.D., Chen, C.C.G.: Objective measures for longitudinal assessment of robotic surgery training. The Journal of Thoracic and Cardiovascular Surgery 143(3), 528–534 (2012)

[17] Martin, J.A., Regehr, G., Reznick, R., MacRae, H., Murnaghan, J., Hutchison, C., Brown, M.: Objective Structured Assessment of Technical Skill for Surgical Residents. British Journal of Surgery 84, 273–278 (1997)

[18] Watson, Deborah, MD. "Septoplasty ." Septoplasty. Ed. Arlen D. Meyers, MD, MBA. Medscape, 13 Feb. 2013. Web. 14 Apr. 2014.

[19] Husney, Adam, MD, and Donal R. Mintz, MD. "Deviated Nasal Septum." WebMD. WebMD, 7 Nov. 2011. Web. 14 Apr. 2014.

[20] Fettman, N., Sanford, T., Sindwani, R.: Surgical Management of the Deviated Septum: Techniques in Septoplasty. Otolaryngologic Clinics of North America 42, 241-253 (2009)

[21] Hwang, P.H., McLaughlin, R.B., Lanza, D.C., Kennedy, D.W.: Endoscopic Septoplasty: Indications, Technique, and Results. Otolaryngology-Head and Neck Surgery 120, 678-182 (1999)

[22] Steele, N, Thomas JR, (eds) Stucker FJ, Souza C, Kenyon GS, Lian TS, Draf W, Schick B, Chapter 1: Surgical Anatomy of the Nose, Rhinology and Facial Plastic Surgery, 2009

[23] "Aurora." NDI Medical. N.p., n.d. Web. 14 Apr. 2014. <http://www.ndigital.com/medical/products/aurora/>.

[24] "Communicate with Computers Naturally." Kinect for Windows. N.p., n.d. Web. 14 Apr. 2014. <http://www.microsoft.com/en-us/kinectforwindows/>.

[25] "How to Draw Noses." How to Draw Noses. N.p., n.d. Web. 15 Apr. 2014. <http://www.discover-how-to-draw.com/how-to-draw-noses.html>.

[26] Jolliffe, I. T. Principal Component Analysis. New York: Springer, 2002. Web.

[27] "Human Head." Wikipedia. Wikimedia Foundation, 15 Apr. 2014. Web. 15 Apr. 2014. <http://en.wikipedia.org/wiki/Human_head>.

[28] "Rotation Matrix." Wikipedia. Wikimedia Foundation, 15 Apr. 2014. Web. 15 Apr. 2014. <http://en.wikipedia.org/wiki/Rotation_matrix>.

[29] Beran, Rudolf. "Minimum Hellinger Distance Estimates for Parametric Models."The Annals of Statistics 5.3 (1977): 445-63. Print.

Appendix A: Code Samples

The following sections contains code samples that have been developed for the presented work:

- labelVideos.m – Produces a playback window that is controlled using keystrokes that produces and saves annotations of the surgery
- myCallback.m - Callback Method to livePlot.m. Called when new UDP Packet arrive. Plots incoming data for real-time tool visualization
- convertPPM_to_AVI.m - Generates RGB compressed and anonymized .avi video from .ppm frames. Also, adds timestamp to the generated video.

The software is still in development. As such, the presented code is shown in its state as of April 2013. For questions, the author can be contacted at ppoddar1@jhu.edu.

```
function labelVideos
%%Set the Current Directory to the 'kinects' folder of the trial to label.
%%The script produces a playback windows that is controlled using
%%keystrokes that produces and saves the annotations.

%%Generate RGB Video%%

clear;
%Load corresponding video frames
mkdir(fullfile('Video'));
RGBimageNames = dir(fullfile('out1','*.ppm'));
RGBimageNames = {RGBimageNames.name}';
RGBimageStrings = regexp([RGBimageNames{:}],'\d\d*[.]\d\d*','match');
RGBimageNumbers = str2double(RGBimageStrings');
[~,RGBsortedIndices] = sort(RGBimageNumbers);
RGBsortedImageNames = RGBimageNames(RGBsortedIndices);
RGBsizes = length(RGBsortedImageNames);
disp('Video starting');
ind = sort([strfind(pwd,'\') strfind(pwd,'-') strfind(pwd,' ')]);
path = pwd;
toSave = [];
close all;

%Instantiate Display Window
figure('KeyPressFcn',@dispkeyevent);
stepSize = 2; %Frames to Skip per Loop Iteration
ii = 1; %Current Frame Index

%Loop through frames of video
while (ii < RGBsizes && ii >= 1)
    img = imresize(imread(fullfile('out1',RGBsortedImageNames{ii})),1/1); %Find current frame
    stringToAdd = RGBsortedImageNames{ii};
    endIndex = strfind(stringToAdd,'-');
    stringToAdd = str2double(stringToAdd(3:endIndex(2)-1));
    hold off;
    figure(1);
    imshow(img);
    xlabel(char('"n" for Nose Circle, "8" for cottle, "d" for delete, "s" for switch surgeons,
"t" for tool tip switch, "5" for idle','0 or Space for pause/resume, ...4 for rewind/slower, 6
for forward/faster, 5 for % completion,"e" for useless movement'));
    title([num2str(ii/RGBsizes) '% Complete with video at time ' num2str(stringToAdd)]);
    pause(1/4000);
    ii = ii + stepSize;
end

%Save Labels
save(['..\..\..\Videos\FixedLabels\' path(ind(3)+1:ind(4)-1) '.mat'],'oldLabels');
disp('Video completed');

%Handle key press events
function dispkeyevent(~, event)
    % Display keypress info in GUI and return event struct if requested
    % event - Structure containing keypress identifiers
    p = event.Character;

     if (p == '5' || p == 'i')
            disp(['Start Tool Idle at frame ' num2str(ii) ' at time ' num2str(stringToAdd)]);
            toSave = [toSave;21,ii,stringToAdd];
     elseif (p == '8')
            toSave = [toSave;31,ii,stringToAdd];
            disp(['Start Cottle Use at frame ' num2str(ii) ' at time ' num2str(stringToAdd)]);
     elseif (p == 'n')
            disp(['Nose circle started at frame ' num2str(ii) ' at time ' num2str(stringToAdd)]);
            toSave = [toSave;11,ii,stringToAdd];
     elseif (p == 's')
               disp(['Surgeon Switched at frame ' num2str(ii) ' at time ' num2str(stringToAdd)]);
               toSave = [toSave;51,ii,stringToAdd];
     elseif (p == 't')
```

```
disp(['Cottle Tool Tip Switched at frame ' num2str(ii) ' at time ' num2str(stringToAdd)]);
               toSave = [toSave;61,ii,stringToAdd];
     elseif (p == 'e')
               disp(['Irrelevant Cottle Movement at frame ' num2str(ii) ' at time '
num2str(stringToAdd)]);
               toSave = [toSave;41,ii,stringToAdd];
     elseif (p == '4')
            disp('Rewind/Slower');
            stepSize = stepSize - 1;
            disp(num2str(stepSize));
     elseif (double(p) == 32 || p == '0')
            if stepSize == 0
                disp('Resume')
                stepSize = 2;
            else
                disp('Pause');
                stepSize = 0;
            end
     elseif (p == '6')
            disp('Forward/Faster');
            if (stepSize > 0)
                stepSize = stepSize+1;
            else
                stepSize = 2;
            end
            disp(num2str(stepSize));
     elseif (p == 'd')
            disp('Deleted Last Line');
            toSave = toSave(1:end-1,:);
     elseif (p == 'c')
           disp(['Side of Nose Changed ' num2str(ii) ' at time ' num2str(stringToAdd)]);
           toSave = [toSave;71,ii,stringToAdd];
     end

end

end
```

```
function y= myCallback(obj,event, udpB,hObject)
%Callback Method to livePlot.m. Called when new UDP Packet arrive. Plots
%incoming data for real-time tool visualization

global circled;  %Global variables indicating whether nose-perimeter circling
is complete
global circling;

handles = guidata(hObject);

y=0;
string = fscanf(udpB); %Read UDP Data
disp(['received: ' string]);
C = strread(string,'%s','delimiter',';'); %Parse Data

%If a full-packet is received...
if (size(C,1) == 15)

    %Parse Data
    sensor = C{1};
    time = C{2};

    sensor_xPos = str2double(C{3});
    sensor_yPos = str2double(C{4});
    sensor_zPos = str2double(C{5});
    R =
[str2double(C{6}),str2double(C{7}),str2double(C{8});str2double(C{9}),str2double
(C{10}),str2double(C{11});str2double(C{12}),str2double(C{13}),str2double(C{14})
];

    R = R';

    %If received packet refers to reference sensor data
    if ( strcmp(sensor,'Reference') == 1)

      handles.headRotation = inv(R);
      handles.headPosition = [sensor_xPos;sensor_yPos;sensor_zPos];

    %If received packet refers to Cottle elevator data
    elseif (strcmp(sensor,'Pointer') == 1)

        %Apply pivot calibration
        TipPos_tracker     = R*handles.tip_pivot     +
[sensor_xPos;sensor_yPos;sensor_zPos];
        TipPosFlat_tracker = R*handles.tip_pivotFlat +
[sensor_xPos;sensor_yPos;sensor_zPos];

        %Transform tool tip to Reference Sensor Coordinates
        TipPos     = handles.headRotation\TipPos_tracker     -
(handles.headRotation)\handles.headPosition;
        TipPosFlat = handles.headRotation\TipPosFlat_tracker -
(handles.headRotation)\handles.headPosition;

        %Add receieved packet to vector of most recent data points
        handles.tipPos     = [handles.tipPos;TipPos'];
        handles.tipPosFlat = [handles.tipPosFlat;TipPosFlat'];

        %Trim vector to contain only most recent points
        handles.tipPos = handles.tipPos(max(1,size(handles.tipPos,1)-
25):end,:);
        handles.tipPosFlat =
handles.tipPosFlat(max(1,size(handles.tipPosFlat,1)-25):end,:);

        %Plot
        pause(0.01);
        cla(handles.axes1);
        hold(handles.axes1,'on');


        toolTip = plot3(handles.axes1,
[handles.tipPos(end,1);(handles.tipPosFlat(end,1)+9*handles.tipPos(end,1))/10],
...

[handles.tipPos(end,2);(handles.tipPosFlat(end,2)+9*handles.tipPos(end,2))/10],
...

[handles.tipPos(end,3);(handles.tipPosFlat(end,3)+9*handles.tipPos(end,3))/10],
...
                                        'r','LineWidth',1.5);
        tool    = plot3(handles.axes1,
[handles.tipPosFlat(end,1);(handles.tipPosFlat(end,1)+9*handles.tipPos(end,1))/
10],...

[handles.tipPosFlat(end,2);(handles.tipPosFlat(end,2)+9*handles.tipPos(end,2))/
10],...

[handles.tipPosFlat(end,3);(handles.tipPosFlat(end,3)+9*handles.tipPos(end,3))/
10], ...
                                        'k','LineWidth',2);

        for (i = size(handles.tipPos,1):-1:max(3,size(handles.tipPos,1)-
handles.tailLength))
             hold(handles.axes1,'on');
            h=fill3(handles.tipPos(i-2:i,1),handles.tipPos(i-
2:i,2),handles.tipPos(i-2:i,3),[1 0 1], 'FaceAlpha', .1,
'Parent',handles.axes1);
            set(h,'EdgeColor','None');
        end

        %Compute current velocity for display

        curVel = 0;
        if (size(handles.tipPos,1) > 1)
            displacement = handles.tipPos(end,:) - handles.tipPos(end-1,:);
            handles.prevTime = handles.curTime;
            handles.curTime = toc;
            curVel = sum(displacement.^2,2)^.5/(handles.curTime-
handles.prevTime);
        end

        %Compute elapsed time
        elapsed = toc;
        min = floor(elapsed/60);
        sec = round(mod(elapsed,60));
        set(handles.text2,'String',{['Elapsed Time: ' num2str(min) ' min '
num2str(sec) ' s'],['Velocity: ' num2str(round(curVel)) ' mm/sec']});   %Set
metrics text

        %If nose-circle complete, then plot nose
        if ( handles.noseSet==1)

shiftNose=fill3(handles.shiftedNose(:,1),handles.shiftedNose(:,2),handles.shift
edNose(:,3),[0,0,0],'FaceAlpha',0,'Parent',handles.axes1);
            hold(handles.axes1,'on');

shiftPlane=fill3(handles.shiftedPlane(:,1),handles.shiftedPlane(:,2),handles.sh
iftedPlane(:,3),[0,1,0],'FaceAlpha',0.1,'Parent',handles.axes1);
            hold(handles.axes1,'on');
            set(shiftPlane,'EdgeAlpha',.2);
        end


        %If nose circling just completed...
        hold(handles.axes1,'on');
        if (circled == 1)
            if(handles.noseSet == 0)

                %Compute initial septal plane estimate
                pcaRes   = pca(handles.noseCircle);
                refPoint = mean(handles.noseCircle,1);
                perp  = pcaRes(:,1);
                perp2 = pcaRes(:,3);
                cP = cross(perp,perp2);
                point1 =
mean(handles.noseCircle(1:round(size(handles.noseCircle,1)/3),:),1);
                point2 =
mean(handles.noseCircle(round(size(handles.noseCircle,1)/3):round(2*size(handle
s.noseCircle,1)/3),:),1);
                point3 =
mean(handles.noseCircle(round(2*size(handles.noseCircle,1)/3):end,:),1);
                cPView = cross(point1-point2,point1-point3);


set(handles.axes1,'xticklabel',[],'yticklabel',[],'zticklabel',[]);
                set(handles.axes1,'FontName','Segoe UI','FontWeight','demi');
                handles.noseSet = 1;

                point1 = refPoint + 45*perp';
                point2 = refPoint - 45*perp';
                point3 = refPoint + 45*perp2';
                point4 = refPoint - 45*perp2';
                septumPlane = [point1;point3;point2;point4];

                %Transform nose outline to reference coordinates
                n = size(handles.noseCircle,1);
                handles.shiftedNose  = (handles.headRotation) \
(handles.noseCircle)'  - repmat( (handles.headRotation)\ handles.headPosition ,
[1, n]) ;
                handles.shiftedPlane = (handles.headRotation) \ (septumPlane)'
- repmat((handles.headRotation)\handles.headPosition , [1, 4]);

                handles.shiftedNose  = handles.shiftedNose';
                handles.shiftedPlane = handles.shiftedPlane';
                meanNose = mean(handles.shiftedNose,1);

                %Plot nose and septal plane
                axis(handles.axes1, [meanNose(1)-100 meanNose(1)+100 ,...
                meanNose(2)-100 meanNose(2)+100 ,...
                meanNose(3)-100 meanNose(3)+100 ]);
```

```
shiftNose=fill3(handles.shiftedNose(:,1),handles.shiftedNose(:,2),handles.shift
edNose(:,3),[0,0,0],'FaceAlpha',0,'Parent',handles.axes1);
                hold(handles.axes1,'on');

shiftPlane=fill3(handles.shiftedPlane(:,1),handles.shiftedPlane(:,2),handles.sh
iftedPlane(:,3),[0,1,0],'FaceAlpha',0.1,'Parent',handles.axes1);
                hold(handles.axes1,'on');
                set(shiftPlane,'EdgeAlpha',.2);
                tic;

            end
        end

        %Store nose-circle coordinates while nose is being circled
        if (circling == 1)
            handles.noseCircle     = [handles.noseCircle;
TipPos_tracker'];
            handles.noseCircleFlat = [handles.noseCircleFlat;
TipPosFlat_tracker'];
        end
    end

    %Flush input buffer if size is exceeded
    if(udpB.BytesAvailable > 1000)
          disp('Input Flushed!');
          flushinput(udpB);
    end

    %Save changes
    guidata(hObject,handles);

end

%Look for more data
readasync(udpB);
end
```

convertPPM_to_AVI.m

```
%%Generates RGB compressed and anonymized .avi video from .ppm frames.
Also
%%add timestamp to the generated video.
%%To run, set the Current Directory to the 'kinects' folder.

mkdir(fullfile('Video'));  %Make Video subfolder if it does not exist
RGBimageNames = dir(fullfile('out1','*.ppm')); %Find all .ppm files
from out1 folder (Change to 'out2' for other camera)
RGBimageNames = {RGBimageNames.name}';
RGBimageStrings = regexp([RGBimageNames{:}],'\d\d*[.]\d\d*','match');
%Find frame names
RGBimageNumbers = str2double(RGBimageStrings');
[~,RGBsortedIndices] = sort(RGBimageNumbers); %Sort frames by times
RGBsortedImageNames = RGBimageNames(RGBsortedIndices);
RGBoutputVideo = VideoWriter(fullfile('Video','RGB1.avi')); %Prepare
video
open(RGBoutputVideo); %Open Video
RGBsize = length(RGBsortedImageNames);

for ii = 1:RGBsize
    img =
imresize(imread(fullfile('out1',RGBsortedImageNames{ii})),1/2); %Load
current image and compress
    h = fspecial('disk',30);
    img(1:round(size(img,1)/3),:,1:3) =
imfilter(img(1:round(size(img,1)/3),:,1:3),h); %Applying anonymizing
blur
    stringToAdd = RGBsortedImageNames{ii}; %%Find time-stamp
    endIndex = strfind(stringToAdd,'-');
    txtInserter = vision.TextInserter(stringToAdd(3:endIndex(2)-
1),'Location',[4 226],'Color',[255 255 255],'FontSize',10); %Add time-
stamp to video
    img = step(txtInserter,img);
    writeVideo(RGBoutputVideo,img); %Add frame to video
    if(mod(ii,50) == 0)
     disp([num2str(ii/RGBsize) '% Complete with RGB1']); %Show video
progress
    end
end

close(RGBoutputVideo); %Close video
```

Vita

Piyush Poddar was born near Princeton, New Jersey. He grew up in the nearby town of Plainsboro and, in 2009, graduated as valedictorian of West Windsor-Plainsboro High School South. He received his Bachelors of Science in Biomedical Engineering from Johns Hopkins University in 2013 with a focus in Sensors and Instrumentation and received the Richard J. Johns Award for academic excellence for graduating with a 3.97 GPA. Piyush was president of the MD-A Chapter of Tau Beta Pi, the National Engineering Honor Society and co-president of Hopkins Medical Device Network.

Piyush's prior work includes being selected as an undergraduate design team leader where he led a team of 8 to develop and market a patent-pending novel patch electrode system for use during defibrillation and cardioversion. In addition, the team won first prize at the Collegiate Inventors Competition held at the US Patent and Trademark Office. He has also worked on the development of a microcontroller-based neonatal feeding monitoring system as part of Sensor Technology Fellowship at the University of Pennsylvania and also collaborated with the Applied Physics Lab to develop an intuitive haptic interface for a surgical robot intended for improved osteolysis.

Outside of his professional life, Piyush enjoys photography, singing, web development, and bhangra (A traditional Indian folk dance). Piyush aims to start his own technology start-up.

Piyush can be reached via email at ppod1991@gmail.com or via phone at (609)851-0053.