SURGICAL SKILL ASSESSMENT USING MOTION TEXTURE ANALYSIS

A Thesis
Presented to
The Academic Faculty

by

Yachna Sharma

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2014

SURGICAL SKILL ASSESSMENT USING MOTION TEXTURE ANALYSIS

Approved by:

Professor Irfan Essa, Advisor
College of Computing
Georgia Institute of Technology

Professor Christopher F. Barnes
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Mark A. Clements
Co-Advisor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Thomas Plötz
Culture lab, School of Computing Science, Newcastle University
United Kingdom

Professor David Anderson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Eric L. Sarin
Division of Cardiothoracic Surgery, Department of Surgery
Emory University School of Medicine

Professor Anthony Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 04-04-2014

To my parents

ACKNOWLEDGEMENTS

I would like to thank several people whose support and guidance have helped me reach this milestone. I am highly indebted to my advisor Professor Irfan Essa and co-advisor Professor Mark A. Clements for providing exceptional support and guidance, giving me the freedom for creative research pursuit, and for providing excellent insights and advice. I would also like to thank Dr. Thomas Plötz, Dr. Eric L. Sarin, and Dr. Karen Liu for their invaluable support and providing access to data and resources required for this research. I would also like to thank the rest of my committee members Professor David Anderson, Professor Anthony Yezzi, and Professor Christopher F. Barnes for their excellent insights and advice.

I am also grateful to several ECE and CS faculty members for their support and encouragement. I am especially grateful to Professor Bonnie H. Ferri, Professor James M. Rehg, Professor John Copeland, Dr. Daniela Staiculescu, and Dr. May D. Wang for their support and encouragement. I would also like to thank Ms. Nina White, Ms. Alicia Richhart, Ms. Tasha Torrence, Ms. Jacqueline D Trappier, and Ms. Marilou Mycko for their time and support.

Many thanks to my I-team and CPL friends who were always supportive and encouraging: Syed Hussain Raza, Vinay Kumar Bettadapura, Edison Thomaz, Ahmad Humayun, Alireza Fathi, Unaiza Ahsan, Daniel Castro, and Andrew Ziegler. I am also grateful to my friends at Georgia Tech whose support helped me endure the PhD phase of my life: Qaiser M. Chaudry, Chanchala Kaddi, Robert M. Parry, Richard A. Moffitt, Chang F. Quo, Teresa H. Sanders, Jenna Fu, Adria W. Motiwalla, Shafi Motiwalla, Tushar Kumar, Shauvik R. Choudhary, Partha Chakraborty, Li Tang, Asha Sharma, and Nova Ahmed.

My gratitude towards my husband, Mayank Sharma, can be hardly expressed. I could not have made it without his support and continuing belief in all my abilities. I am especially grateful to my son Varen Sharma for being understanding and supportive when I had to miss his soccer games and other activities. I am also grateful to my siblings, Shankar Bhardwaj, Rahul Bhardwaj, Lalita Sharma, and their families for being supportive and taking care of our parents while I was pursuing my academic goals. I am thankful to my sister-in-law Richa Sharma and her spouse Deepak Sharma for being there for my parent-in-laws and supporting us in many ways.

I am profoundly thankful to my parents-in-law, Shri Chanderpal Sharma and Mrs. Saroj Sharma for their love, affection, and encouragement. I am highly grateful to my parents, Shri Ramgopal Bhardwaj and Mrs. Shakuntla Devi for their affection, encouragement and providing all the opportunities.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this Ph.D. research is to design and develop a framework for automated assessment of surgical skills. Automated assessment can help expedite the manual assessment process and provide unbiased evaluations with possible dexterity feedback.

Evaluation of surgical skills is an important aspect in training of medical students. Current practices rely on manual evaluations from faculty and residents and are time consuming. Proposed solutions in literature involve retrospective evaluations such as watching the offline videos. It requires precious time and attention of expert surgeons and may vary from one surgeon to another. With recent advancements in computer vision and machine learning techniques, the retrospective video evaluation can be best delegated to the computer algorithms.

Skill assessment is a challenging task requiring expert domain knowledge that may be difficult to translate into algorithms. To emulate this human observation process, an appropriate data collection mechanism is required to track motion of the surgeon's hand in an unrestricted manner. In addition, it is essential to identify skill defining motion dynamics and skill relevant hand locations.

This Ph.D. research aims to address the limitations of manual skill assessment by developing an automated motion analysis framework. Specifically, we propose (1) to design and implement quantitative features to capture fine motion details from surgical video data, (2) to identify and test the efficacy of a core subset of features in classifying the surgical students into different expertise levels, (3) to derive absolute skill scores using regression methods and (4) to perform dexterity analysis using motion data from different hand locations.

CHAPTER I

INTRODUCTION

Summary *The motivation and goals for this research, along with the challenges involved, lead us to the specific aims and organization of the thesis to address those specific aims.*

1.1 Surgical skill assessment

Surgical skill development, i.e., the process of gaining proficiency in procedures and techniques required for professional surgery, represents an essential part of medical training. Developing high quality surgical skills is a time-consuming process, which requires expert supervision and evaluation throughout all stages of the training procedure. This manual assessment of surgical skills poses a substantial resource problem to medical schools and teaching hospitals.

There is a desire for streamlining the skill assessment routine at least in the early stages of the medical training process. In addition to the substantial time requirements of manual evaluations, the assessment criteria used are typically domain specific and often subjective, where even domain experts do not always agree on the assessment scores. In fact, poor correlations between subjective evaluations and objective evaluations through standardized written and oral exams have been reported in the literature [6].

Structured manual grading systems, such as the Objective Structured Assessment of Technical Skills (OSATS) [37], have been proposed to alleviate the problem of subjective assessments. OSATS covers a variety of evaluation criteria: *respect for tissue* (RT), *time and motion* (TM), *instrument handling* (IH), *suture handling* (SH), *flow of*

operation (FO), *knowledge of procedure* (KP) and *overall performance* (OP). Assessments based on such objective criteria can alleviate the subjectivity problem but are still challenging, as they require time-consuming and straining manual observations and evaluations by expert surgeons.

As a response to the growing need of skill assessment for (prospective) surgeons, we propose a framework for automated assessment of OSATS criteria using video data. Such an automatic assessment system allows for direct and objective feedback on the quality of standard surgical procedures, as they have to be mastered by every medical student. By using video data, the system has minimal requirements of the infrastructure, which is of benefit for realistic and large-scale deployments. Our automatic, vision-based approach to surgical skill assessment can be used for evaluating medical students in their early training phases.

1.2 Challenges

The task of replicating an expert surgeon's evaluations is challenging due to several reasons. Some of the challenges are described below.

1. ***Diverse OSATS criteria***: First, the OSATS criteria are diverse in nature. For example, the "respect for tissue" criterion is based on the student's capability in handling the tissue without injuring it while performing the procedure. On the other hand, criteria such as "knowledge of procedure" and "time and motion" depend on the sequential aspect of the motion. Thus, it is very challenging to encode diverse OSATS criteria within a common framework.

2. ***Expert disagreement***: Secondly, expert surgeons do not agree on evaluations and the OSATS score might vary from one expert to another. Thus, it becomes difficult to define benchmarks for developing automated skill assessment system. This is further complicated by the style variations among surgeons in performing different tasks.

3. ***Capturing fine motion dynamics***: Most of the OSATS criteria are evaluated based on motion quality. An expert's motion lacks unnecessary moves and is characterized by fluid movements as compared to unnecessary moves and stiff motions of intermediate and novice surgeons [37]. Thus, there is a need to analyze the motion dynamics in detail to extract skill relevant information.

4. ***Varying camera viewpoint***: The orientation or the camera viewpoint may vary in different data acquisitions. Thus, a data representation is required that is invariant to view changes. In addition, the motion dynamics for the whole duration of the procedure should be encoded.

Due to these challenges, it is difficult to design an automated surgical skill assessment system. With the availability of cost effective cameras and memory, it is easy to collect video data in a ubiquitous manner. If the automated system can utilize the video data collected from student surgeons, then this will alleviate the need for hand tracking equipment, which, if used, might interfere with surgeon's hand motion.

1.3 Motion analysis for skill assessment

Due to the challenges mentioned above, the task of surgical skill assessment requires motion analysis to capture fine details that encode the skill involved in performing a particular procedure. Motion analysis is used for several purposes in computer vision such as activity recognition, segmentation and object tracking. Typical motion analysis techniques involve defining a motion or activity type or a gesture vocabulary. The gestures might be obtained automatically using techniques such as spectral clustering or by manually labeling the videos. Using predefined gesture vocabularies have some disadvantages, such as manual bias involved in defining the gestures and style variations rendering some gestures unused by specific skill groups. This thesis addresses some of these challenges by proposing an automated skill assessment system that does not require segmentation of motion into surgical gestures. The proposed system is

based on holistic motion analysis and attempts to capture fine motion details using frame kernel matrices.

The goal of this research is to ***develop a video based motion analysis system to improve the consistency and speed in surgical skill evaluation***. We accomplish this goal in three phases corresponding to the following three specific aims:

Specific aim 1*: To encode the skill defining motion dynamics from videos into quantitative feature descriptors.*

Specific aim 2*: To model a relative skill assessment system for classification of videos into different expertise levels.*

Specific aim 3*: To model an absolute skill assessment system for prediction of surgical skill scores and to perform dexterity analysis based on hand motion dynamics.*

1.4 Organization of the thesis

This thesis is organized as follows (Figure 1). In Chapter 2, we summarize the published works on surgical skill assessment that provided motivation and background for this work. Chapter 3 provides the background on motion analysis using frame kernel matrices and texture feature computation using different techniques. In Chapter 4, we describe the data sets used in this research. In Chapter 5, we describe our video based motion texture analysis system for classification of surgical students into different skill levels. In Chapter 6, we demonstrate the capability of motion texture analysis for OSATS skill score prediction using regression analysis. In Chapter 7, we model the surgical dexterity using motion texture features derived from the motion dynamics corresponding to dominant and non-dominant hands. Chapter 8 provides a summary of our work with possible extensions and potential future applications.

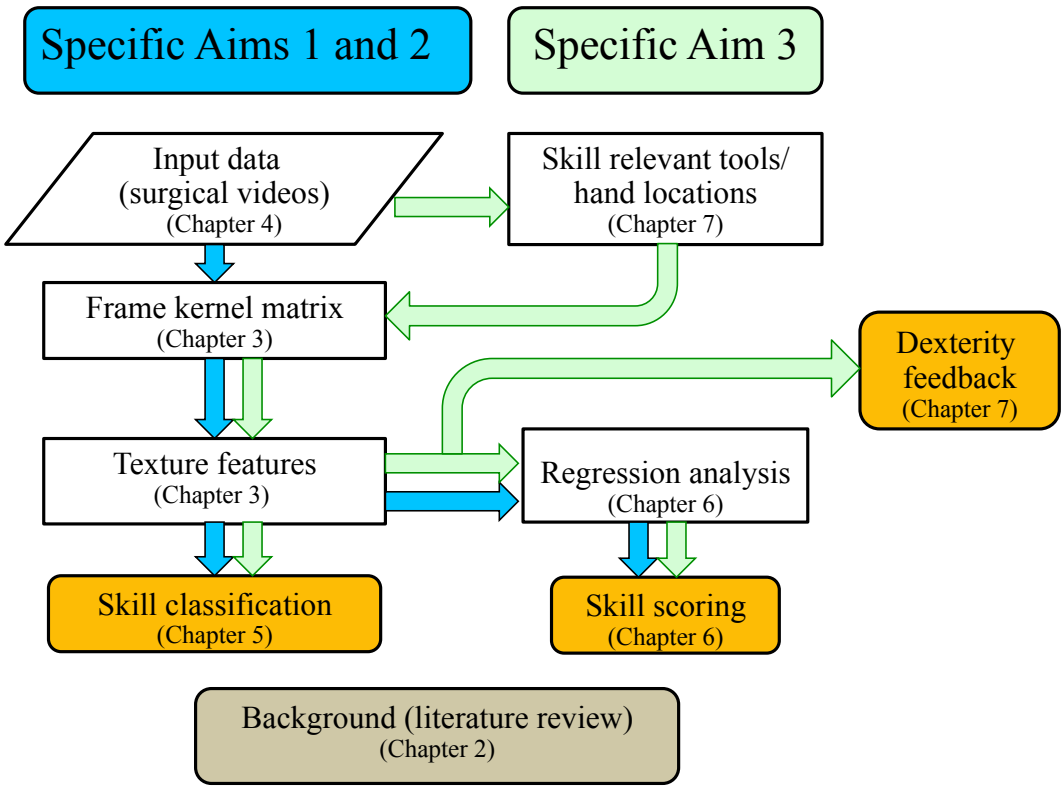


Figure 1: Flow diagram of the thesis with three main contributions: skill classification, skill scoring, and dexterity analysis.

CHAPTER II

LITERATURE REVIEW

Summary *Most of the published works pertain to recognition of manually defined surgical gestures in robotic minimally invasive surgery. Automated assessment of OSATS has not been explored much in the literature. Most works on video analysis of surgery address gesture recognition.*

2.1 State-of-the-art

There are two domains where assessment of surgical skills has been studied. The first one pertains to skill assessment of surgeons performing robotic minimally invasive surgery (RMIS). The second domain is assessment of skills in medical schools and teaching hospitals [38]. Table 1 compares these two domains. The data used in RMIS is mostly kinematic data collected from robotic arms and actuators. On the other hand, in teaching domain, skill is evaluated manually. Most of the RMIS works follow the *language of surgery* paradigm, where each surgical procedure is segmented

Table 1: Comparison of RMIS and surgical education domains.

Attributes	**Robotic Minimal Invasive Surgery (RMIS)**	**Surgical education and training**
Data	Kinematic data from robotic surgery equipment e.g. da-Vinci	Manual observations
Surgical gesture	Manually defined and procedure specific	No procedure specific vocabulary
Analysis	Gesture recognition, expertise modeling, classification	Manual scoring using methods such as Objective Standard Assessment of Technical Skills (OSATS)
Feedback on hand motion quality	Not given	Not given

into specific gestures (also known as *surgemes*). These gestures are procedure specific and they are obtained by manual demarcation of the motion trajectory. The analysis goal in RMIS is mostly surgical gesture recognition with few works on surgical skill classification. Figure 2 shows a general framework for RMIS skill assessment. However, none of the published RMIS works have reported OSATS based skill assessment. The skill assessment in medical schools and teaching hospitals is done manually. The manual scoring is time intensive and requires the expert to closely observe the student surgeons. Medical literature also recognizes the need for objective surgical skill assessment [49]. Structured grading systems such as the Objective Structured Assessment of Technical Skills (OSATS) [37] have been proposed to reduce the subjectivity. The OSATS criteria are challenging to evaluate since they require expert domain knowledge and are prone to subjective assessment. Thus, if a single human observer grades the trainee surgeon, then it may result in bias. Yu et al. [61] have suggested evaluations from residents and interns who frequently supervise the students instead of the consultant surgeons who do not have the opportunity to directly observe medical students. However, the subjectivity and time-consuming nature of these evaluations

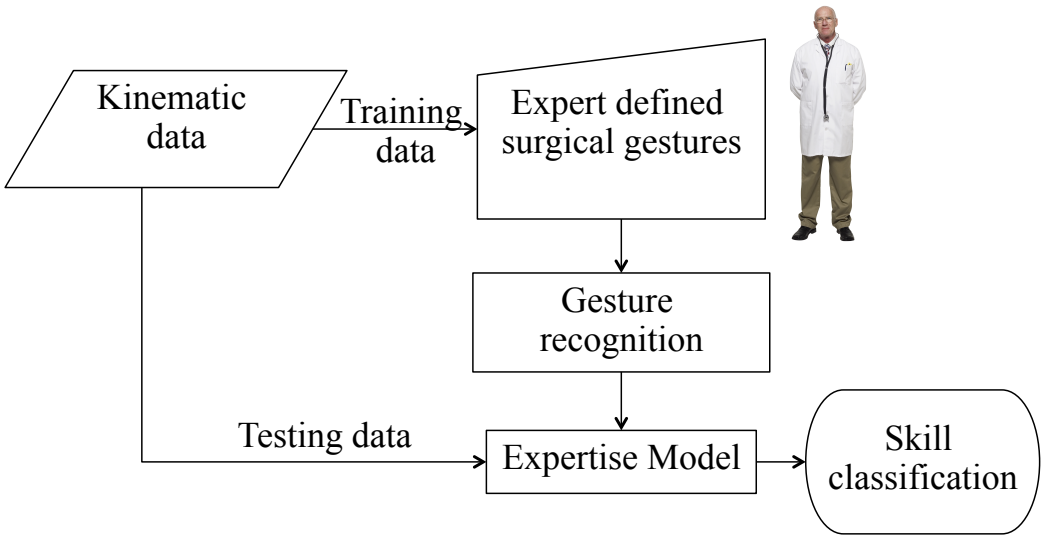


Figure 2: A general skill assessment framework used in RMIS domain.

still cannot be ruled out. Awad et al. [6] have reported poor correlation between the subjective and the objective evaluations through standardized written and oral exams. They also note that subjective evaluations may vary from faculty to faculty and for different types of procedures.

A major drawback of manual OSATS assessment are the substantial requirements on time and resources involved in getting several staff surgeons to observe the performance of trainees. Surprisingly, only very few works have addressed automated OSATS assessments for surgical teaching evaluations.

These two domains (RMIS and surgical education) can be further categorized depending upon the approaches used for time series analysis of surgical motion data. The local approaches (*e.g.* [23, 62]), model specific surgical tasks, and model the task as a sequence of manually defined surgical gestures. On the other hand, the global approaches involve the analysis of the whole motion trajectory without segmentation into surgical gestures [25, 15]. The focus of this thesis is surgical skill assessment for medical education and training using the global approach based on motion texture analysis.

The state-of-the-art in computerized surgical skill evaluation is dominated by RMIS using robots such as *da-Vinci* [35, 36, 47, 34, 46, 25]. Lin et al. [35] used kinematic data (position and velocity) from the da-Vinci manipulators to map the motion data into surgeme labels. They used Linear Discriminant Analysis (LDA) to project the motion data into a feature space where the surgeme classes were well separated.

The initial works did not address the skill assessment. Their analysis revealed motion characteristics that might separate the experts from the non-experts. For example, in [35], the authors noted that an expert surgeon's motion gestures (surgemes) are well separated in the three dimensional LDA feature space as compared to the non-experts. Reiley *et al.* [46] reported varying surgeme recognition accuracy for

expert and non-expert motion data. This occurs because the expert surgeons do not frequently use some of the surgemes. Moreover, merging of some surgemes resulted in improved recognition accuracy. These results indicate the limitation of surgeme based motion analysis in accommodating the user variations. To address this issue, additional surgemes were added in later works.

Statistical modeling approaches (such as Gaussian Mixture Models (GMM) and Hidden Markov Model (HMM)) were also used to account for the variability in the motion data. To further improve the surgeme recognition accuracy, video context cues were used to create the context-based HMM models. Lin et al. [34] used eight context cues for a four-throw suturing task. These cues were based on the interaction of the four main objects in the RMIS working space: left tool, right tool, needle with suture and the tissue. Each frame was annotated for the context cues such as "left tool touching needle", "right tool touching tissue" *etc.* With context cues, the surgeme recognition accuracy improved despite the varying expertise level of the users. However, this approach still required an expert surgeon to select the number of surgemes and to annotate the videos with the context cues.

Reiley and Hager [46] developed twenty-four skill models (three expertise models: novice, intermediate and expert for each of the eight surgemes). They calculated the probability of a test sequence against each of the three HMMs trained for the three expertise levels. For a given test sequence, the most likely skill model that produced the sequence was designated as the skill level of the test sequence. Their results show that all surgemes are not equally discriminating of the skill. Higher recognition rates were observed for those surgemes where the experts performed more efficiently than the novices. In addition, the experts do not use some of the surgemes. Reiley *et al.* [48] have summarized the works on surgical skill evaluation. Most of the works in their review also pertain to the robotic surgery based on the surgemes.

Surgical skills may also be assessed by the surgeon's capability in handling the

instruments. For example, Trejos et al. [56] designed a sensor equipped instrument for Minimally Invasive Surgery (MIS). They measured the instrument's position trajectory using electromagnetic sensors along with the forces and the torques acting at the instrument's tip. They also observed that the position trajectories of experts were clear and distinct as compared to the non-experts (similar findings as in [35]). Their instrument was specifically designed for the MIS procedures and they did not report the classification of expert versus non-expert surgeons. They used wired electromagnetic sensors, which may interfere with surgeon's hand motion. These sensors can also suffer from magnetic distortions in the presence of metals within their working volume.

King et al. [27] designed a sensor glove for the laparoscopic MIS procedure. The battery operated sensor glove consisted of seven accelerometers and one fiber-optic bend sensor. The accelerometers were attached to a battery by wires. However, this work also involved segmentation of the surgical task into elementary gestures corresponding to a specific procedure in the laparoscopic surgery. Five gestures were used to determine the sensor position on the glove. The clustering results showed reasonable separation between the experts and the novices; however, quantitative classification results were not reported. Saggio et al. [50] used a wired hand motion glove to obtain surgical motion, which was then replayed as an avatar on a screen. The trainees used this system to evaluate themselves by superimposing their motion on the expert avatars.

Recent works such as [63, 40] have reported skill assessment based on video analysis. Both these works have focused on the laparoscopic surgery. In [63], the authors used colored based tracking to track the left and the right hand. The hand trajectories were then segmented using Self Similarity Matrix (SSM) followed by tracking the tool points and the objects in each segment. They noted that the motion trajectories of experts resulted in fewer segments as compared to the novices. They used

average velocity and motion jitter histograms as the two distinguishing features for the expert and the novice groups. However, their approach is specifically designed for the laparoscopic surgery. Another work (also on robotic surgery) [40], used eighteen basic motion elements for a laparoscopic surgery task. These basic motion elements are specifically defined for the laparoscopic task and may not be applicable to the general surgical procedures. Recent publications [11, 55] also focus on the robotic surgery and the surgeme based techniques. However, the techniques developed for robotic surgical skill assessment may not be applicable for general surgical trainee assessment. In addition, there is not a standard set of surgemes to accommodate the surgeons with varying skill levels.

Table 2: Related works on surgical video analysis

Reference	Technique	Gesture	Analysis goal	Data
Haro (2012), Zapella (2013) [23, 62]	BoW, LDS	Yes, manual	Surgical gesture recognition	RMIS (both kinematic and video data from robotic surgery), 8 subjects
Bettadapura (2013) [8]	A-BoW	No	OSATS skill classification	General suturing task (only video data), 16 subjects
Padoy (2012) [44]	DTW, HMM	Yes, manual	Surgical phase recognition	Laparoscopic cholecystectomy (endoscopic video), 4 subjects
Lalys (2011) [30]	DTW	Yes, manual	Surgical phase recognition	Cataract surgery, 20 videos
Blum (2010) [9]	CCA, HMM	Yes, manual	Surgical phase recognition	Laparoscopic surgery, 10 videos
Lin (2009) [34]	HMM	Yes, manual	Skill classification but not on individual OSATS criteria	RMIS (both kinematic and video data from robotic surgery), 6 subjects

Abbreviations: BoW: Bag-of-words, LDS: Linear dynamical systems, DTW: Dynamic time warping, CCA: Canonical correlation analysis, HMM: Hidden Markov Model, OSATS: Objective structured assessment of technical skills, RMIS: Robotic minimally invasive surgery.

2.2 Video based surgical analysis

With advances in video data acquisition, the attention has shifted towards video based analysis in both RMIS and teaching domains. Table 2 summarizes recent works on surgical video data. Most of these classify different surgemes or surgical phases and the data from different types of surgeries are used. Haro et al. [23] and Zapella et al. [62], employed both kinematic and video data for RMIS surgery. They used linear dynamical systems (LDS) and bag-of-features (BoF) for surgical gesture (surgeme) classification in RMIS surgery.

Datta et al. [15] used the video snapshots of difficult surgical tasks. The expert surgeons evaluated these videos using the OSATS method. They also used an electromagnetic hand tracking system to detect the number of hand movements. The hand movement was detected as a change in the velocity. They defined surgical efficiency score as the ratio of OSATS "end product quality score" and the number of detected hand movements. This formulation is based on the fact that experts exhibit lower number of hand motions as compared to the novices. Their results indicate significant correlations between the overall OSATS rating and the surgical efficiency. However, they did not correlate the hand movements to individual OSATS criteria. It is important to provide the feedback on individual OSATS criteria so that the trainee can improve on those specific criteria.

BoF (Bag-of-Features), also known as Bag-of-Words (BoW), do not capture the underlying structural information, neither of causal nor of sequential type, which is inherent by the ordering of the words. In Augmenting-Bag-of-Words (A-BoW) [8], the motion is modeled as short sequences of events and the underlying temporal and structural information is automatically discovered and encoded into BoW models. With A-BoW technique, higher classification accuracy is reported for all seven OSATS criteria as compared to standard BoW technique.

In this thesis, we propose Motion Texture (MT) analysis and Sequential Motion

Texture (SMT) analysis that can be effectively used for surgical skill assessment. We also note that with appropriate feature and parameter selection, higher skill classification accuracy can be achieved with MT and SMT as compared to contemporary approaches such as BoW and A-BoW. Our results on a diverse data collected in a general surgical lab setting indicate the skill assessment potential of our framework for medical schools and teaching hospitals.

2.3 Conclusions from literature

Below, we summarize the conclusions derived from the literature to guide our research on surgical skill assessment.

1. Most published works pertain to robotic surgical skill assessment. However, the medical literature clearly describes the need for an objective skill assessment in a general surgical training.

2. A considerable portion of the published works attempts to model the motion primitives (surgemes) for the surgical tasks. Table 3 provides a summary of the surgical gestures (*surgemes*) used in the literature. However, there is no standard method to determine the number and types of surgemes that would accommodate surgeons of different expertise levels. For instance, [35, 36] use eight surgemes, whereas [46] and [55] use six and twelve surgemes, respectively, for the same suturing task. Additional surgemes (9, 10 and 11 marked with $*$ in Table 3) were added to account for the variability of new users [47]. Moreover, important motion dynamics might be missed in the surgeme approach especially at the boundary of the surgemes.

3. Robotic surgical assessment techniques pool all kinematic data without carrying out motion analysis of individual hand locations. Thus, it may not be possible to provide the dexterity feedback.

Table 3: Summary of surgical gestures (*surgemes*) used in the literature.

Gesture	Lin (2005, 2006) [35, 36]	Lin (2009)[34]	Tao (2012) [55]
0	-	-	Idle motion
1	Reach for needle (gripper open)	Reach for needle (gripper open)	Reach for needle
2	Position needle (holding needle)	Head towards suturing line (holding needle, right hand)	Position needle
3	Insert needle/push needle through tissue	Insert or push needle through tissue	Insert needle/push needle through tissue
4	Move to middle with needle (left hand)	Move to middle with needle (left hand)	Move to middle with needle (left hand)
5	Move to middle with needle (right hand)	-	Move to middle with needle (right hand)
6	Pull suture with left hand	Pull suture away from suturing line (left hand)	Pull suture with left hand
7	Pull suture with right hand	Pull suture away from suturing line (right hand)	Pull suture with right hand
8	Orient needle with two hands	-	Orienting needle with two hands
9	-	-	$*$ Right hand assisting left while pulling suture
10	-	-	$*$ Loosen more suture
11	-	-	$*$ End of trial

4. If the goal of skill assessment is to improve a trainee's performance over time, then skill assessment should be performed for the individual OSATS criteria. This is not addressed in the published literature.

5. Wired electromagnetic sensors may not be appropriate since they might interfere with the surgeon's hand motion.

In summary, the automated OSATS assessment has not been addressed in literature. It is essential to represent the motion dynamics in a view invariant manner to address the varying viewpoint in video data acquisition, to cater for style variations

and to provide assessments for diverse OSATS criteria. To address style variations and accommodate OSATS diversity, it is important to extract skill relevant motion from the motion dynamics in a holistic manner. In RMIS approaches, high gesture recognition accuracy is obtained, however, due to variations in skill and style, all gestures are not used by the surgeons [46]. Thus, it may not be feasible to predict OSATS with gesture based local approaches. In the next chapter we introduce our framework for motion texture analysis and present a proof of concept study to test the feasibility of the proposed framework.

CHAPTER III

MOTION ANALYSIS FOR SKILL ASSESSMENT

Summary *In pursuit of obtaining automatically segmented gestures, we explore the activity recognition and time series segmentation techniques. Using time series segmentation via spectral clustering and a simple hand motion toy data set, we observed that skill information might get masked even though semantically correct segments are obtained. This led us to encode motion dynamics into frame kernel matrices followed by texture analysis to reveal skill relevant information.*

In this chapter, we present our motivation and background for motion texture analysis. First, we describe the key differences between activity recognition and skill assessment and introduce our concept of skill specifically for the task of surgical skill assessment (Section 3.1). From Chapter 2, it is clear that some manual bias may be introduced in the surgeme-based approaches since a single surgeon usually defines the surgical gestures and all surgeons do not use all gestures. This motivated us to test whether automated gesture segmentation of time series data into surgical gestures would capture skill relevant information. Spectral clustering and related graph based approaches have gained widespread interest in recent years for time series segmentation and activity recognition [65, 58]. In Section 3.2, we describe the spectral clustering method and use it to demonstrate the differences between skill assessment and activity recognition. In Section 3.3, we present results from our study on a toy data set to demonstrate that time series segmentation techniques such as spectral clustering might mask the skill information. In Section 3.4, we demonstrate that texture analysis of affinity matrices (also known as frame kernel matrices, self-similarity matrices, and recurrence matrices) can be used to extract skill relevant

information.

3.1 Activity recognition versus skill assessment

Activity recognition methods utilize core technologies such as segmentation, feature extraction and tracking to classify activities to support diverse application domains (*e.g.* surveillance, entertainment *etc.*) [4]. The goal of activity recognition is to recognize or classify a video (or segment of a video) into distinct activity types depending on the application domain. Overall, computer vision based activity recognition labels video (or time series data) for *what* (activity type) has happened and *when* it happened.

The *quality* of how well an activity is done is of interest in several domains. For instance, some tasks require training over long periods of time under the guidance of expert professionals and skills are acquired and evaluated in a progressive manner. For example, in manufacturing assembly, the person with specific training performs each task. Such training requires frequent monitoring, evaluation, and intervention. In most training programs, a supervisor evaluates the trainee manually and evaluations become time consuming for several individuals and training tasks. An automated proficiency evaluation system can help alleviate time and resource requirements of manual skill assessment.

For proficiency evaluation, the activity type is known *a priori*. At the macro level, all trainees will be performing one given task (same activity). An activity recognition system would classify all instances of the given task into one activity type. Beyond that and of more practical relevance, the goal of skill evaluation is to score a task on a given scale from low proficiency to high proficiency. At the micro-level, the instances of a given activity will differ based on the expertise of the person performing the activity. Human experts can detect subtle differences within an activity type. However, for an automated assessment, a representation is required

that encodes low-level motion data into proficiency specific features.

To contextualize our work, we define *skill* as a measure of one's effectiveness in performing a given activity. Skill can be measured in absolute terms (giving a numeric grade *i.e.* skill score prediction) or in relative terms (comparing among a group of participants *i.e.* classification into different skill levels). Motion quality is embedded in fine dynamics of motion that may require fine-grained analysis. Segmentation of time series data into pre-defined motion primitives may not be sufficient for this purpose since it might miss important dynamics within and at the boundary of the segments, as we show later in Section 3.3. Automated assessment techniques thus need to analyze activity data at a substantially more fine grained level in order to unveil quality changes, which can be caused by only very subtle changes in motion patterns.

3.2 Spectral clustering

We now describe a popular graph based technique called *spectral clustering* [58]. Spectral clustering has been used for time series segmentation and activity recognition. Our motivation to use spectral clustering is two-fold. First, automated segmentation of time series data might alleviate the human bias in surgeme-based methods. Secondly, the affinity matrix used in spectral clustering encodes the motion dynamics and we demonstrate that fine texture patterns in the affinity matrix encode the skill relevant information, which might not be deciphered otherwise by segmentation techniques such as spectral clustering. We select spectral clustering since it is based on the affinity matrix, thus utilizing pairwise distance between data points and not being affected by the high dimension of the time series data. It is especially beneficial when using high dimensional RMIS data. In addition, for view invariant segmentation, spectral clustering can be used with affinity (or self-similarity) matrix computed by applying a feature mapping to time series data as we describe later.

3.2.1 Graph based clustering

For a given set of data points $x_1, x_2, ..., x_n$ and a given measure of similarity s_{ij} between pairs of data points x_i and x_j, clustering attempts to divide the data points into groups such that points in the same group are similar and points in different groups are dissimilar to each other. This data can be represented as a similarity graph $G = (V, E)$ with vertex set V and the edge set E. Each vertex v_i in this graph represents a data point x_i. Two vertices are connected if the similarity s_{ij} between corresponding data points x_i and x_j is positive and larger than a predefined threshold, and the edge is weighted by s_{ij}. The problem of clustering now becomes a graph-partitioning problem. That is, we want to find a partition of the graph such that the edges between different groups have very low weights and the edges within a group have high weights.

For an undirected weighted graph with vertex set $V = v_1, v_2, ..., v_n$, assume that each edge between two vertices v_i and v_j carries a non-negative weight $w_{ij} \geq 0$. If $w_{ij} = 0$, it implies that the vertices v_i and v_j are not connected by an edge. Since G is an undirected graph, $w_{ij} = w_{ji}$. The *degree* of a vertex $v_i \in V$ is defined as $d_i = \sum_{j=1}^{n} w_{ij}$. The degree matrix D is defined as the diagonal matrix with the degrees $d_1, d_2, ..., d_n$ on the diagonal. Similarity graphs model the neighborhood relationships between the data points. Below, we describe commonly used similarity graphs.

1. *ϵ-neighborhood graph*: All points whose pairwise distances are smaller than ϵ are connected. As the distances between all connected points are approximately of the same scale (at most ϵ), weighting the edges would not incorporate more information about the data to the graph. These graphs are usually considered as an unweighted graph.

2. *k-nearest neighbor graph*: Vertex v_i is connected to vertex v_j if v_i is among the k nearest neighbors of v_j or if v_j is among the k-nearest neighbors of v_i.

3. *Fully connected graph*: All points are connected with each other with positive similarity. Since the graph should represent the local neighborhood relationships, this construction is only useful if the similarity function itself models local neighborhoods. For example, the parameter σ in the Gaussian similarity function, $s(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$, defines the neighborhood size and plays the same role as ϵ in the ϵ-neighborhood graph.

In kernel theory [52], self-similarity matrix is computed as Gram matrix after applying a feature mapping to time series data. For a d-dimensional time series $\mathbf{X} \in \mathbb{R}^{d \times n}$ of length n, the Gram matrix is defined as

$$G_{ij} =< x_i, x_j > . \tag{1}$$

If a feature mapping ϕ is applied to the data, then the resulting matrix is termed as the kernel matrix $\mathbf{K}$. Each entry in matrix $\mathbf{K}$, κ_{ij}, is given by $\kappa_{ij} =< \phi(x_i), \phi(x_j) >= \phi(x_i)^T \phi(x_j)$ and it defines the similarity between the two frames x_i and x_j using a kernel function. For a given feature map ϕ, the normalized kernel corresponds to a feature map given by

$$\mathbf{X} \mapsto \phi(\mathbf{X}) \mapsto \frac{\phi(\mathbf{X})}{\|\phi(\mathbf{X})\|}. \tag{2}$$

A normalized Gaussian kernel function is given by $\kappa_{ij} = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$ where σ is the standard deviation. The parameter σ controls the flexibility of the kernel. Small values of σ tend to make the kernel matrix close to an identity matrix. Large values of σ result in a constant kernel matrix. A frame kernel matrix defines the similarity between two frames in a time series using a kernel function [52].

It is important to note that the orientation or the camera viewpoint may vary in different data acquisitions. Thus, a data representation is required that is invariant to view changes, and that encodes motion dynamics for the whole time-series. Frame kernel matrices provide a suitable representation to encode skill relevant motion dynamics because mapping of data points to the kernel feature space ensures that the

Algorithm 1 - Spectral clustering [42]

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$ and k number of clusters to construct
Step 1: Construct a fully connected similarity graph using Gaussian kernel function as described in Subsection 4.2.1. Let W be its weighted adjacency matrix.
Step 2: Compute the normalized Laplacian L as $L = D^{-1/2}SD^{-1/2}$.
Step 3: Compute the first k eigenvectors $u_1, u_2, ...u_k$ of L
Step 4: Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, u_2, ..., u_k$ as columns.
Step 5: Form a matrix $T \in \mathbb{R}^{n \times k}$ from U by normalizing the rows to norm 1 that is set $t_{ij} = u_{ij}/(\sum_k {u_{ik}}^2)^{1/2}$.
Step 6: For $i = 1, 2, ..., n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the ith row of T.
Step 7: Cluster the points $(y_i)_{i=1,2,...,n}$ with the k-means algorithm into clusters $C_1, C_2, ...C_k$.
Output: Clusters $A_1, A_2, ..., A_k$ with $A_i = \{j | y_j \in C_i\}$.

motion dynamics depend only on the relative locations of the data points with respect to each other and not on the global origin.

Several segmentation based approaches are based on the frame kernel matrix such as the spectral clustering, aligned clustering analysis and hierarchical clustering analysis [65, 26]. These techniques exploit the block-diagonal characteristic of the frame kernel matrix [17]. A highly block diagonal frame kernel matrix indicates the activities with sharp transitions. After obtaining the fully connected graph representation, we apply the spectral clustering method proposed by Ng, Jordan, and Weiss [42] (Algorithm 1).

3.3 Time series segmentation using spectral clustering

For initial testing of the frame kernel matrix and its effect on spectral clustering, we collected Motion Capture (MOCAP) data using three optical markers (one on the left hand and two on the right hand as shown in Figure 3) from a subject performing five activities.

The activities involved simple hand motions such as sewing, slicing bread, chopping onions, mixing batter, and making dough. The right-handed subject performed the activities in a predefined order with little or no motion from the left hand except for making the dough. The motion trajectories collected from the right hand optical markers mainly dictate the skill involved in these activities. The x, y, and z

Figure 3: Optical markers (white spheres) used for collecting motion capture data fastened to gloves worn by the subject while performing the activities.

coordinates of the right hand MOCAP trajectory are shown in Figure 4. The textured block diagonal matrix results in five clusters using spectral clustering (Figure 4, bottom row).

The x, y, and z coordinates of the left hand MOCAP trajectory are shown in Figure 5. Since only right hand is used most of the time, the left hand trajectory remains mostly static except for the last activity (making dough), which requires using both the hands. The frame kernel matrices for the right and the left hand markers are shown in Figure 6. Note the homogeneous block diagonal pattern for the left hand marker and the textured pattern for the right hand marker. Although, left hand does not move much, the subject may re-position the left hand slightly while switching from one activity to another. This results in a block diagonal matrix and subsequent analysis with spectral clustering results in the segmentation as shown in Figure 5 (bottom row).

The segmentation results for the left and the right hand are similar despite the clear

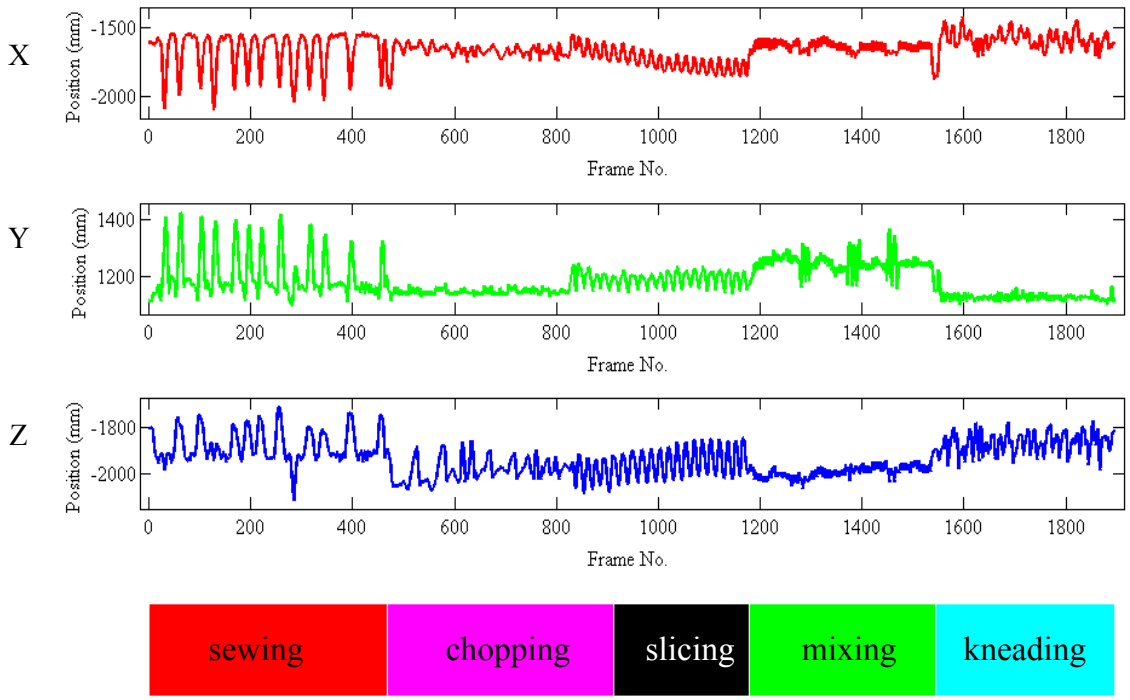


Figure 4: Top three rows: x, y, and z coordinates of the right hand optical marker. Bottom row: Segmentation results using spectral clustering.

differences in the motion dynamics (Figure 6). Thus, automated segmentation might yield motion primitives (or gestures/surgemes) that are not indicative of the skill. Manual selection of motion primitives may be time consuming besides being biased and subjective. For these reasons, we propose our technique based on holistic time series analysis using motion texture technique.

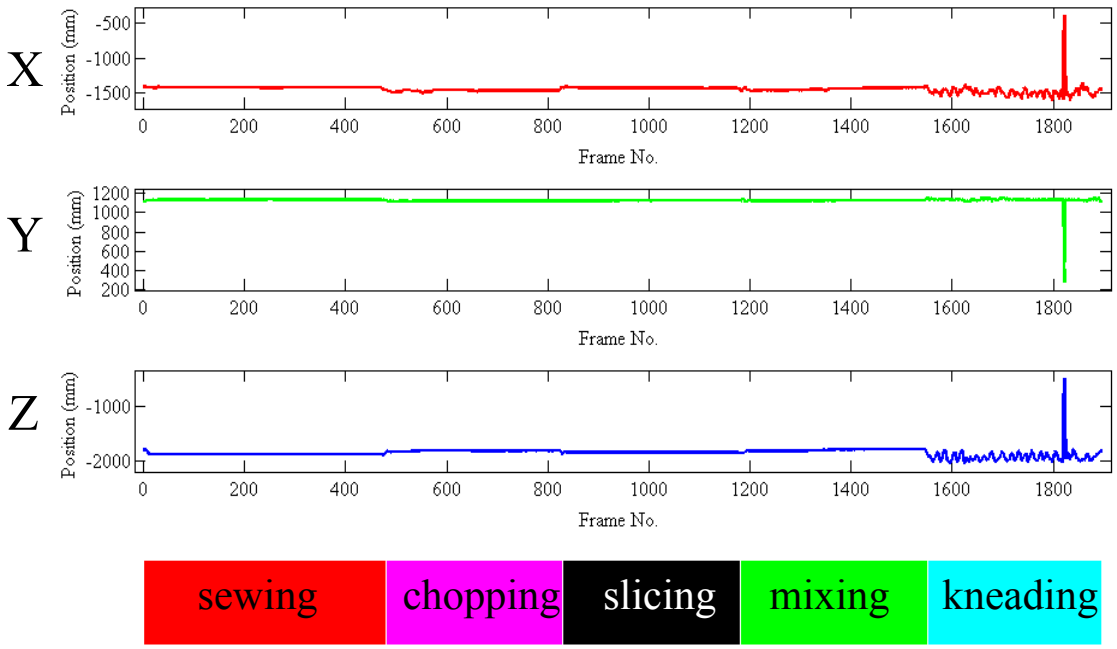


Figure 5: Top three rows: x, y, and z coordinates of the left hand optical marker. Bottom row: Segmentation results using spectral clustering.

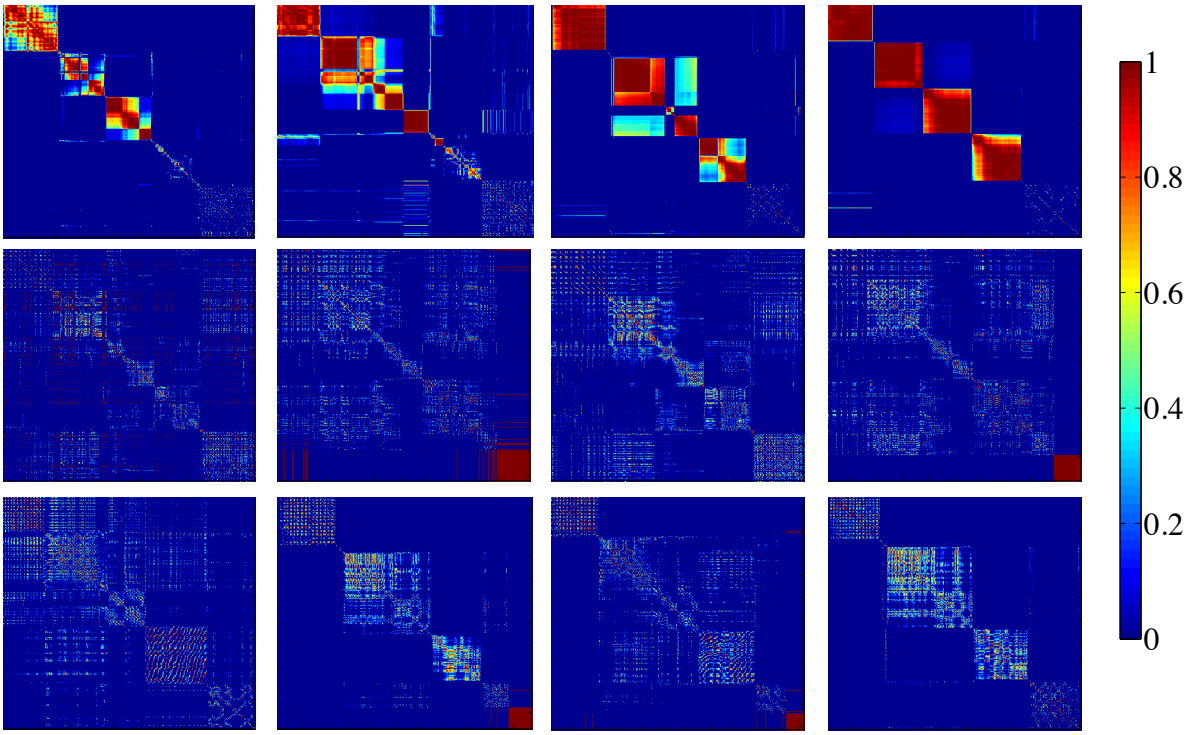


Figure 6: Sample frame kernel matrices for the left hand marker (top row), right hand marker without practice (middle row), and right hand marker with practice (bottom row).

3.4 Motion texture analysis for skill assessment

The frame kernel matrices obtained by using Gaussian kernel function have a dynamic range of [0-1]. In addition, as seen in Figure 6, the distinct textural patterns in the frame kernel matrices seem to vary corresponding to the skill relevant motions. These observations led us to texture analysis of frame kernel matrices. A $N \times N$ frame kernel matrix is equivalent to a $N \times N$ normalized gray scale image with [0-1] dynamic range, where N is the length of the time series data. In this Section, we describe two texture analysis methods that can be used to obtain skill relevant information from frame kernel matrices.

3.4.1 Gray Level Co-occurrence Matrix (GLCM)

If the spatial domain of the frame kernel matrix $\mathbf{K}$ is considered as an intensity image I, then, texture features can be extracted from this image to encode the motion dynamics. This technique allows analysis of the motion information in the frame kernel matrix by fine texture analysis. Textural statistics can be derived from the frame kernel matrix using techniques such as the Gray-Level Co-occurrence Matrix (GLCM). GLCM based texture analysis has been used widely in different domains [22, 54, 12]. GLCM is obtained by calculating how often a pixel with intensity level i occurs in a specific spatial relationship to a pixel with intensity level j. Let (x, y) represent the spatial domain of the kernel matrix $\mathbf{K}$ and I be its intensity domain. Then, the spatial domain of two pixels a and b is given by (x_1, y_1) and (x_2, y_2) respectively. Let the intensity of pixels be $I(x_1, y_1) = i$ and $I(x_2, y_2) = j$. If the pixel pair satisfies the relation $(x_2, y_2) = (x_1, y_1) + (d\cos\alpha, d\sin\alpha)$, for an offset d and direction α, then, it is termed as the pixel pair with spatial offset d and direction α.

The co-occurrence probability between two gray levels is given by,

$$p_{d,\alpha}(i,j) = \frac{P_{d,\alpha}(i,j)}{\sum_{n=0}^{N_g-1}\sum_{n=0}^{N_g-1} P_{d,\alpha}(i,j)}, \tag{3}$$

where $P_{d,\alpha}(i,j)$ is the number of occurrences of gray level i and j that are separated by an offset d in the direction α, and where N_g is the quantized number of gray levels. A sample image with four intensity levels is shown in Figure 7 (left). The corresponding 4×4 GLCM matrix using an offset of one in the horizontal direction, is shown in Figure 7 (right). The GLCM matrix represents the spatial distribution of the gray levels in the image. For instance, if the GLCM diagonal elements are large, then the image consists of contiguous regions with coarse texture (or less motion dynamics for a frame kernel matrix image). On the other hand, smaller diagonal entries correspond to the fine motion dynamics.

The frame kernel matrix encodes the fine motion dynamics as textured patterns. If the frame kernel matrix is treated as an intensity image, then the motion dynamics are exhibited as textured patterns in the image domain. This enables quantification of the motion dynamics using texture analysis.

Texture features are computed from the GLCM matrix to encode different textural characteristics (Table 4). For example, correlation measures the gray level linear

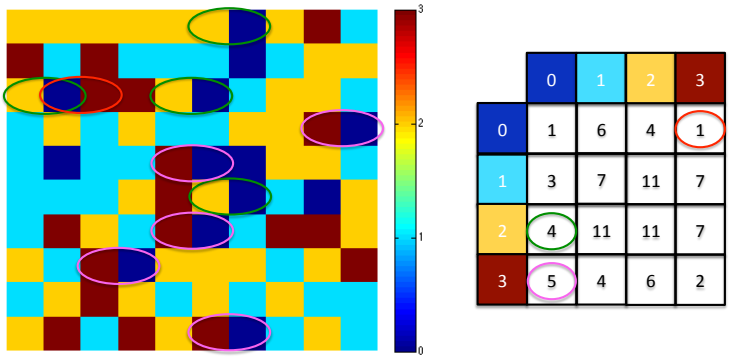


Figure 7: Left: A sample 10×10 image with four intensity levels. Right: GLCM with an offset of one in the horizontal direction with the highlighted ellipses illustrating the horizontal spatial relationship in the image.

dependency between two pixels at a specified position relative to each other. Contrast measures the local intensity variations while cluster shade and cluster prominence measures the uniformity and proximity. Energy or angular second moment measures the homogeneity in the image while dissimilarity is a measure of the total variation present in the image. Sum of squares variance assigns high weights to the elements that differ from the mean value of the normalized GLCM matrix. We use eight gray levels and compute the 8×8 GLCMs for eight directions ($\alpha = 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°$) with an offset of $d = 1$. We take the mean GLCM over eight directions and normalize the mean GLCM matrix. The mean and standard deviations for the rows and the columns of the normalized mean GLCM matrix are given by,

$$\mu_x = \sum_i \sum_j i p(i,j), \mu_y = \sum_i \sum_j j p(i,j); \tag{4}$$

and

$$\sigma_x = \sum_i \sum_j (i - {\mu_x}^2) p(i,j), \sigma_y = \sum_i \sum_j (j - {\mu_y}^2) p(i,j). \tag{5}$$

For features f_{11} to f_{16} (Table 4), we use the following notations. If $p_x(i)$ and $p_y(i)$ represent the ith entry in the marginal probability matrix obtained by summing the rows and columns of $p(i,j)$ respectively; that is,

$$p_x(i) = \sum_{j=0}^{N_g-1} p(i,j); p_y(i) = \sum_{i=0}^{N_g-1} p(i,j), \tag{6}$$

then,

$$p_{x+y}(k) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p(i,j); i + j = k, \tag{7}$$

for $k = 0, 1, ..., 2(N_g - 1)$, and

$$p_{x-y}(k) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p(i,j); |i - j| = k, \tag{8}$$

Table 4: Texture features derived from the gray level co-occurrence matrix (GLCM).

No.	Name	Formulation
f_1	Autocorrelation [54]	$\sum_i \sum_j (i,j)p(i,j)$
f_2	Contrast [22, 54]	$\sum_{n=0}^{N_g-1} n^2 \{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \vert i-j \vert = n \}$
f_3	Correlation [22, 54]	$\frac{\sum_i \sum_j (i,j)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$
f_4	Cluster prominence [54]	$\sum_i \sum_j (i+j-\mu_x-\mu_y)^4 p(i,j)$
f_5	Cluster shade [54]	$\sum_i \sum_j (i+j-\mu_x-\mu_y)^3 p(i,j)$
f_6	Dissimilarity [54]	$\sum_i \sum_j \vert i-j \vert \cdot p(i,j)$
f_7	Energy [22, 54]	$\sum_i \sum_j p(i,j)^2$
f_8	Entropy [54]	$-\sum_i \sum_j p(i,j) \log(p(i,j))$
f_9	Homogeneity [54]	$\sum_i \sum_j \frac{1}{1+i-j^2} p(i,j)$
f_{10}	Maximum probability [54]	$\max_{i,j} p(i,j)$
f_{11}	Sum of squares variance [22]	$\sum_i \sum_j (i-\mu)^2 p(i,j)$
f_{12}	Sum average [22]	$\sum_{i=2}^{2N_g} i p_{x+y}(i)$
f_{13}	Sum variance [22]	$\sum_{i=2}^{2N_g} (i-f_{14})^2 p_{x+y}(i)$
f_{14}	Sum entropy [22]	$-\sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\}$
f_{15}	Difference variance [16]	$-\sum_{i=0}^{N_g-1} i^2 p_{x-y}(i)$
f_{16}	Difference entropy [22]	$-\sum_{i=0}^{N_g-1} p_{x-y}(i) \log\{p_{x-y}(i)\}$
f_{17}	Information measure of correlation 1 [22]	$\frac{HXY-HXY1}{\max\{HX,HY\}}$
f_{18}	Information measure of correlation 2 [22]	$(1-\exp\lceil -2.0(HXY2-HXY)\rceil)^2$
f_{19}	Inverse difference normalized [13]	$\sum_i \sum_j \frac{p(i,j)}{1+\vert i-j \vert^2/{N_g}^2}$
f_{20}	Inverse difference moment normalized [13]	$\sum_i \sum_j \frac{p(i,j)}{1+(i-j)^2/{N_g}^2}$

for $k = 0, 1, ..., (N_g - 1)$. Features f_{17} and f_{18} were proposed in [22]. For f_{17}, HX and HY are the entropies of p_x and p_y respectively and HXY is given by,

$$HXY = -\sum_i \sum_j p(i,j) log(p(i,j)) \tag{9}$$

The terms $HXY1$ and $HXY2$ in Table 4, are given by,

$$HXY1 = -\sum_i \sum_j p(i,j) log\{p_x(i)p_y(j)\} \tag{10}$$

and

$$HXY2 = -\sum_i \sum_j p_x(i)p_y(j) log\{p_x(i)p_y(j)\} \tag{11}$$

respectively.

3.4.2 Local Binary Pattern (LBP)

Besides GLCM, we can also use the local binary patterns [20]. LBPs are extracted by comparing each image pixel with its neighborhood and the neighborhood is defined in a circularly symmetric manner. It can be expressed by,

$$LBP(N,R) = \sum \sum_{i=0}^{N-1} u(g_i - g_c)2^i, \tag{12}$$

where N is the number of neighboring samples and R is the radius of neighborhood, g_i is the intensity of neighboring pixel i $(i = 0, 1, ..., N-1)$, g_c is the intensity of center pixel and $u(x)$ is a step function with $u(x)$=1 if x=0 and $u(x)$=0 otherwise.

Image representation by LBP based methods could increase the robustness against illumination variation, however, the capability of encoding image configuration and pixel wise relationships might be weakened since LBPs quantize gray-level differences into two binary levels. To overcome this limitation, Guo et al. [20] proposed the Local Configuration Pattern (LCP), which models the local microscopic configuration with respect to each pattern. In this method, optimal weights are estimated for neighboring pixels to linearly reconstruct the intensity of central pixel. This can be expressed as,

$$E(a_0, a_1, ...a_{N-1}) = |g_c - \sum_{i=0}^{N-1} a_i g_i| \tag{13}$$

Here g_c and g_i denote intensity values of the center pixel and neighboring pixels of a particular pattern type respectively. The coefficients a_i $(i = 0, ..., N-1)$ are the weighing parameters associated with g_i and $E(a_0, a_1, ...a_{N-1})$ is the reconstruction

error regarding model parameters a_i $(i = 0, ..., N-1)$. Optimal parameters are determined by least squares estimation to minimize the reconstruction error.

Suppose the occurrence of a particular pattern type L is n_L for an image I, *i.e.* there are n_L pixels in I with the pattern type L. The intensities of these N_L pixels can be denoted as $c_{L,i}$ $(i = 0, 1, ...n_L - 1)$. Organizing these intensities into a vector, we get,

$$C_L = \begin{bmatrix} c_{L,0} \\ c_{L,1} \\ . \\ . \\ . \\ c_{L,n_L-1} \end{bmatrix} \tag{14}$$

The intensities of the neighboring pixels $v_{i,0}, v_{i,1}, ..., v_{i,N-1}$ $(i = 0, 1, ..., n_L - 1)$ can be organized as

$$V_L \begin{bmatrix} v_{0,0} & v_{0,1} & . & . & . & v_{0,N-1} \\ v_{1,0} & v_{1,1} & . & . & . & v_{1,N-1} \\ & & & . & & \\ & & & . & & \\ & & & . & & \\ v_{N_L-1,0} & v_{n_L-1,1} & . & . & . & v_{n_L-1,N-1} \end{bmatrix} \tag{15}$$

In order to minimize the reconstruction error, the unknown parameters a_i $(i = 0, 1, ..., N-1)$ are constructed as a column vector:

$$A_L = \begin{bmatrix} a_0 \\ a_1 \\ . \\ . \\ . \\ a_P - 1 \end{bmatrix} \tag{16}$$

When the system is over determined, optimal parameter vector A_L is determined by:

$$({V_L}^T V_L)^{-1} {V_L}^T C_L \tag{17}$$

To produce rotation invariant features, 1D Fourier transform is applied to the estimated parameter vector A_L and the transformed vector is given by:

$$H_L(k) = \sum_{i=0}^{P-1} A_L(i) e^{-j2\pi ki/P} \tag{18}$$

where $H_L(k)$ is the kth element of H_L and $A_L(i)$ is the ith element of A_L. The magnitude part of vector H_L is taken as the microscopic configuration feature given by:

$$|H_L| = [|H_L(0)|; |H_L(1)|; ...; |H_L(P-1)|] \tag{19}$$

The norm $|H_L|$ encodes the image configuration and pixel wise interaction relationship of each specific pattern and it is combined together with pattern occurrences of local binary patterns to obtain a complementary feature for both the discrimination of microscopic configuration and local structures.

The final LBP-LC feature is thus given by

$$LCP = [[|H_0|; O_0]; [|H_1|; O_1]; ...; [|H_{q-1}|; O_{q-1}]] \tag{20}$$

where $|H_i|$ is calculated by Equation 19 with respect to the ith pattern of interest, O_i is the occurrence of the ith local pattern (i.e., the LBP) of interest and q is the total

number of patterns of interest. Multi-scale analysis can be achieved by combining LCPs with different radii and neighboring samples.

3.5 Skill assessment

To test the efficacy of motion texture analysis in capturing the fine motion details, MOCAP data was collected from two subjects performing six trials of simple hand motions as described in Section 3.3. The sequential motion types were sewing, chopping onions, slicing bread, mixing batter and making dough. Three optical markers (one on the left hand and two on the right hand) were used for this purpose. This resulted in thirty-six trajectories of the MOCAP data. Frame kernel matrix was computed for each MOCAP trajectory and twenty textural features (Table 4) were obtained. A separate cluster was observed for the left hand markers by applying simple k-means clustering on two textural features. Clustering results are shown in Figure 8. It is clear that the textural features for the left and the right hand markers are different thus validating the hypothesis that texture analysis based on the frame kernel matrix is a feasible approach for isolating the skill relevant information.

3.6 Summary

It is clear from the results presented in this Chapter that frame kernel matrix provides a suitable representation to encode the motion dynamics and is especially effective for encoding skill relevant information due to following reasons:

1. Mapping of data points to the kernel feature space ensures that the motion dynamics depend only on the relative locations of the data points with respect to each other and are not dependent on the global origin.

2. It is known that the expert motions are more organized, distinct and uncluttered as compared to the non experts [35, 5]. Thus, expert motions are expected to yield well-organized patterns in the frame kernel matrix.

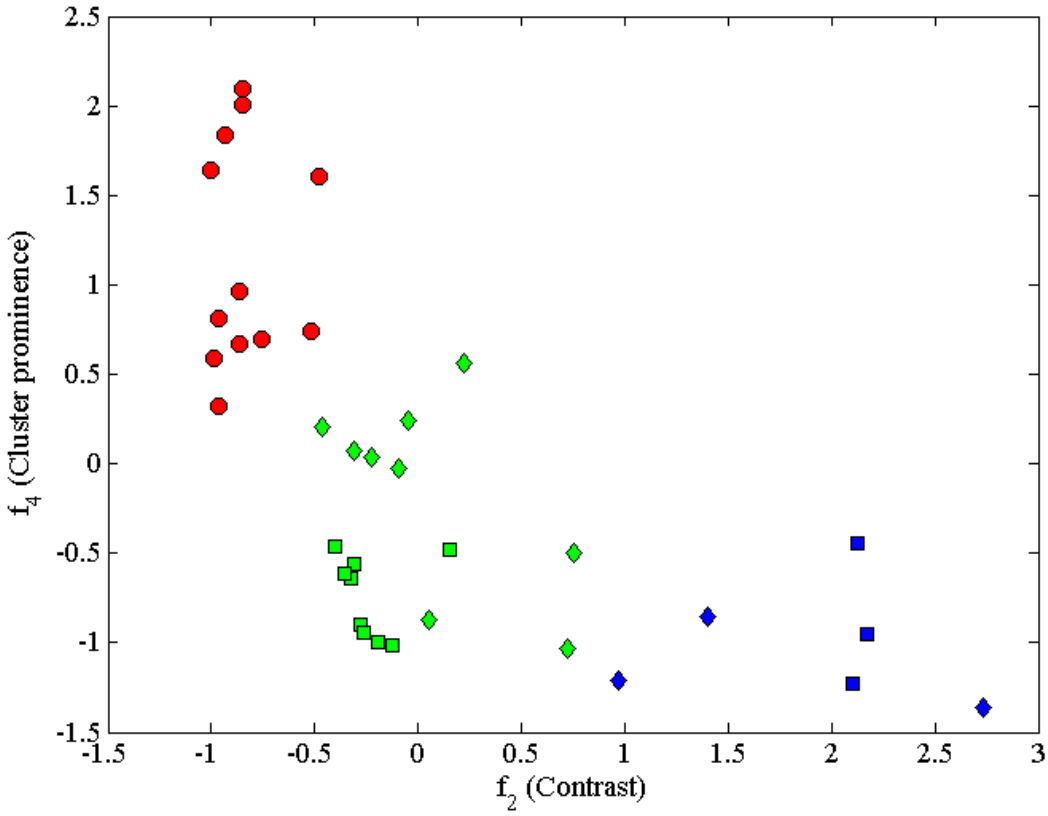


Figure 8: *k-means* clustering using two frame kernel texture features. Colors (red, green, and blue) represent the *k-means* cluster membership. Circles represent the left hand marker; squares and diamonds represent the two right hand markers respectively.

Frame kernel matrix based segmentation methods such as the spectral clustering, aligned clustering analysis and hierarchical clustering analysis [65, 26] exploit the block-diagonal characteristic of the frame kernel matrix [17]. A highly block diagonal frame kernel matrix indicates the activities with sharp activity transitions. This may not work well for general activities with smooth transitions since the motion element boundaries may be fuzzy and hard to detect. User style variations might also contribute to the block diagonal frame kernel matrix masking the actual skill encoded in the fine motion dynamics. As shown in this Chapter, texture analysis of frame kernel matrix has the potential of encoding skill relevant information.

We have used the MOCAP data to demonstrate the motion texture analysis for two reasons. First, we wanted clean data captured in controlled settings (5 activities performed in sequence for almost equal duration) with only three markers so that we can clearly understand the dynamics from the left and right hand motions. Secondly, MOCAP acquisition directly provides the X, Y, and Z trajectory of the markers and no object detection or tracking is required. This speeds up and simplifies the analysis and helps understand the key concepts.

Besides MOCAP, the motion dynamics may also be learned indirectly using the video features by tracking moving objects in the scene. The video data may not be very clean (as MOCAP) due to possible noisy motions, occlusions and detection errors. MOCAP, on the other hand requires an expensive setup with multiple cameras and software to reconstruct the 3D motion trajectory, and optical markers. Thus, there is trade-off in using MOCAP data versus video data in terms of motion accuracy (2D versus 3D), ease of acquisition and portability. With video data, it is important to note that the orientation or the camera viewpoint may vary in different data acquisitions. However, with kernel mapping, the motion dynamics depend only on the relative locations of the data points with respect to each other and are not dependent on the global origin. In next chapter, we describe the video data sets used for surgical skill assessment.

CHAPTER IV

VIDEO DATA FOR SURGICAL SKILL ASSESSMENT

Summary *To perform surgical motion analysis, we used two video data sets acquired in different settings and scored by different experts. The characteristics of the two data sets dictate the analytic approaches used for skill assessment and dexterity analysis in the forthcoming chapters.*

Motion trajectories can be obtained from different hand locations using the MOCAP technique. However, MOCAP data acquisition requires multiple cameras to detect an optical marker from different viewpoints. Data from these cameras is then used to reconstruct the three-dimensional motion trajectory of each optical marker. Alternatively, video data from a single camera can be collected faster and in a ubiquitous manner. Current evaluation paradigms in medical schools require faculty surgeons to evaluate the trainee surgeons either in-person or by watching their videos retrospectively. This poses a substantial time and resource problem for medical schools and teaching hospitals. We perceive that our system will help alleviate time and resource requirements by providing automated skill evaluation using video data.

In this chapter, we describe two video data sets used in our work. We use these two data sets for the following reasons. First, we anticipate that video data may be collected in different settings. By using two data sets collected in different settings *i.e.* different camera, frame rate, background, tasks *etc.*, we test our technique of extracting skill relevant information via motion texture analysis. This provides validation to our framework. Secondly, we want to test different modalities such as depth and acceleration data that might provide better discrimination between experts and non-experts. Finally, for dexterity analysis, we also compute motion features for

both right and left hands individually.

The first data set consists of video data collected using a standard high-resolution video camera and three-dimensional acceleration data collected using wireless Axivity sensors from sixteen participants with varying degree of expertise. We will call this data set "***Newcastle data***" in this and subsequent chapters. We use videos from Newcastle data for surgical skill classification and prediction based on OSATS criteria. The second data set is collected using the Creative* interactive gesture camera developer kit [2]. The Creative* interactive gesture camera is a small, lightweight, USB-powered camera optimized for close-range interactivity. It is designed for ease of setup and portability and includes a High Definition (HD) webcam, depth sensor and built-in dual-array microphones for capturing and recognizing voice, gestures and images. We will refer to the depth and video data collected from Creative* camera as "***GT-Emory Data-set***". We use this data set to analyze the individual hand motions and to perform dexterity analysis of surgical movements.

Due to moral and ethical issues involved in the use of live animals, it is becoming difficult to justify the use of animals if alternative methods and materials are available [37]. Thus, bench models are frequently used for teaching and testing technical skills since they are lower in cost, have high portability, reuse the materials, and readily available. For both Newcastle and GT-Emory data sets, we used bench models enabling ubiquitous data collection.

4.1 Newcastle data

We recruited 16 participants (medical students) for our case study. Previous suturing expertise and background of the participants varied. Every participant performed suturing activities involving tasks such as stitching, knot tying, *etc.* thereby using a needle-holder, forceps and the tissue suture pads. These training sessions were recorded using a standard video camera (50fps, 1280x720 pixels), which was mounted

Table 5: Number of samples for different expertise levels

	RT	TM	IH	SH	FO	KP	OP
Novice	2	9	8	10	3	8	6
Intermediate	14	15	16	15	16	9	17
Expert	15	7	7	6	12	14	8

on a tripod. Fifteen participants performed two sessions of a suturing task. Each session was recorded in a separate video. An expert surgeon also performed three sessions giving a total of thirty-three videos. The average duration of the videos is 18 minutes. The expert surgeon based on the OSATS scoring scheme provided ground truth assessment. We group the participants into three categories according to their expertise: low (OSATS score ≤ 2), intermediate ($2 <$ OSATS score ≤ 3.5) and high ($3.5 <$ OSATS score ≤ 5) expertise levels to train our models with sufficient samples per class. Table 5 shows the number of videos used in our study corresponding to three expertise levels for each OSATS criteria.

Figure 9 shows the sample frames from Newcastle data. As compared to contemporary works [62, 23], this data set is acquired in natural settings with varying camera viewpoint, clothing and background objects. The participants performed surgical tasks in a lab setting with people moving in the background. Figure 10 shows sample close-up images of running suturing task performed by a novice, intermediate, and an expert surgeon.

The acceleration data is collected using a tiny wireless 3-axis accelerometer [1]. Three-dimensional acceleration data is collected at 50Hz (or every 20 milliseconds). Three accelerometers are used – one on the dominant hand wrist, second on the needle-holder, and a third on the forceps. Figure 11 shows the X, Y, and Z dimensions of acceleration data collected from a novice, intermediate, and an expert surgeon.

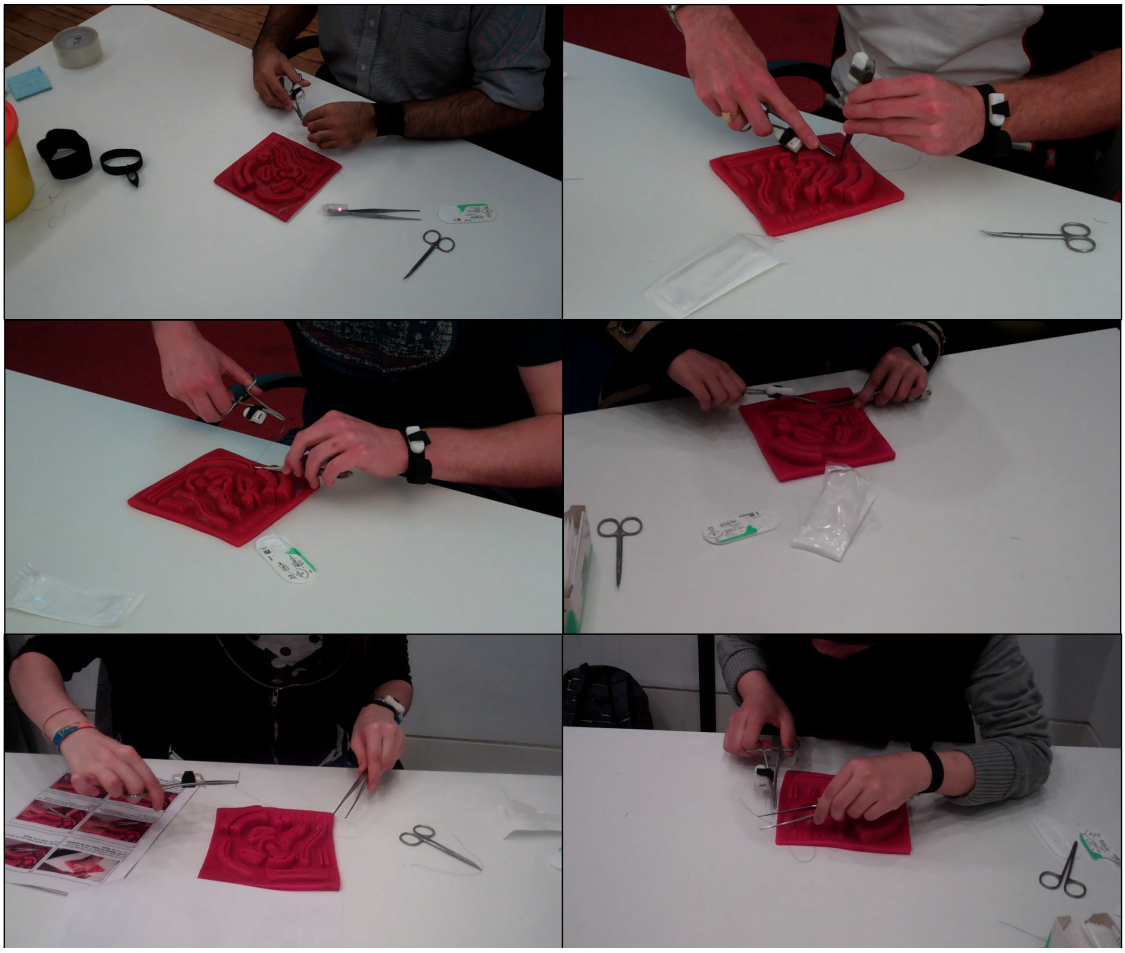


Figure 9: Sample frames from Newcastle data set.

Figure 10: Samples of a running suturing task performed by a novice (left), intermediate (center), and an expert (right) surgeon.

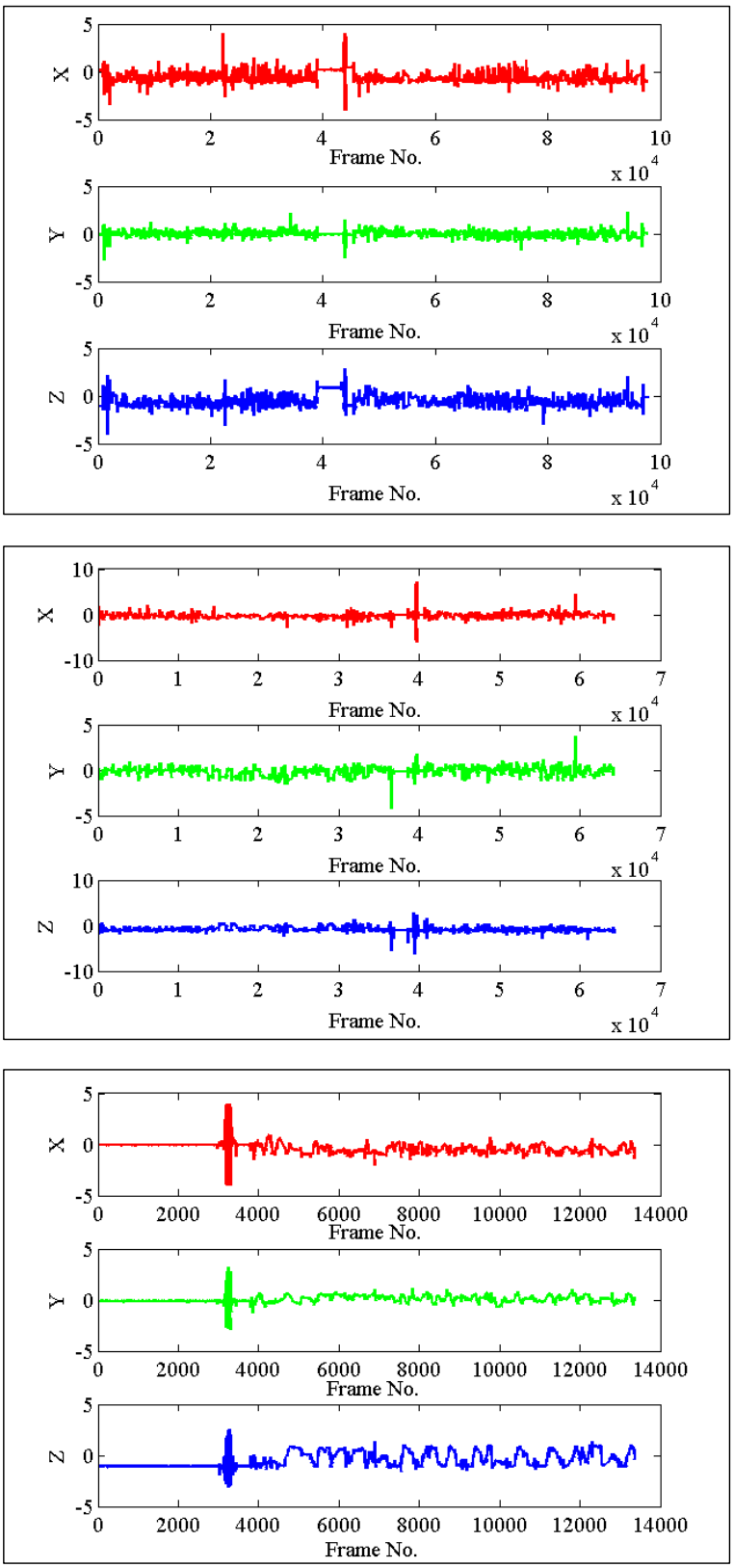


Figure 11: Sample X, Y, and Z dimensions of acceleration for running suturing task performed by a novice(top), intermediate (middle), and an expert (bottom) surgeon.

4.1.1 Key characteristics and challenges in Newcastle data set

It is clear from Figure 9 that Newcastle data was acquired from different camera viewpoints. Also, from Figure 10, it is interesting to note that a novice surgeon performed fewer stitches (Figure 10 (left)) as compared to the intermediate, and expert surgeons. The amount of task performed (and thus the expertise levels) in Newcastle data is highly variable. Also, note that the intermediate and expert stitches are comparable visually, however, the motion characteristics during task performance define the skill level.

Figure 11 demonstrates some important characteristics of Newcastle data set. Note that the acceleration data for a novice surgeon (Figure 11, top) shows movement in almost all the frames. In addition, novice surgeons generally take more time (more frames) to accomplish the task. In next Chapter, we will show how frame kernel matrices computed from video data can represent these characteristics.

4.2 GT-Emory Data-set

We recruited eighteen participants to collect data for dexterity analysis using Creative Intel Perceptual camera [2]. We used the camera, with the Intel Perceptual Computing SDK Beta 2013. We used the Perceptual Computing SDK and the Creative camera since it allows collection of both depth and video data simultaneously. We will use the depth dynamics at hand locations for dexterity analysis (Chapter 7). The camera is mounted on a tripod and the participant performs the surgical task wearing colored finger-less gloves. We use the colored gloves to track the left and right hand locations using OpenCV blob detection library (available at http://code.google.com/p/cvblob/). Since some subjects in our study were left-handed, we used the green glove for the dominant hand and red glove for the non-dominant hand. In addition, we refer to the dominant hand as right hand and non-dominant hand as the left hand. Both RGB and depth data were acquired at 30

frames per second. The maximum spatial resolution obtainable from Creative camera is 640×480 pixels for RGB frames and 320×240 pixels for the depth frames. Also, the depth and RGB cameras are located at a spatial offset. To obtain the depth values at corresponding RGB frame hand locations, we align the depth and RGB frames. We used the Intel Perceptual Computing Software Development Kit (SDK) for RGB and depth alignment. We map depth coordinates to color coordinates using the PXCProjection interface and the function "*MapDepthToColorCoordinates*". The resulting depth frames (640×480) are written into a video file using open source computer vision library (OpenCV).

We collected two instances for two tasks (suturing and knot tying) from each participant. For suturing, we collected 4000 frames and for knot tying, we acquired 1000 frames per task instance. Figure 12 shows sample frames from RGB data collected from the Creative camera. Note the varying acquisition conditions such as illumination, background, standing and sitting positions of the participants. Figure 13 shows the aligned depth frames corresponding to the RGB frames in Figure 12, and Figure 14 shows the depth masks overlaid on the RGB frames.

We also collected the acceleration data using Axivity sensors. We acquired three-dimensional acceleration data at 50Hz (or every 20 milliseconds) using two accelerometers. For suturing task, one accelerometer was attached to the dominant hand wrist and the second one to the needle-holder. For knot tying, one accelerometer was attached to each of the left and right hand wrists. Figure 15 shows the X, Y, and Z dimensions of acceleration data and the corresponding video file displayed in ELAN software [53, 3]. We used ELAN to align the acceleration data with the video frames. At the start of each instance, each participant was asked to rapidly shake the hands/instruments with the accelerometers to get the synchronization waveform that is used to align the acceleration data with the video using the ELAN software.

Figure 12: Sample RGB frames from GT-Emory data. Note the changing camera viewpoint, illumination, suturing pads.

4.2.1 Key characteristics and challenges in GT-Emory data set

We acquired GT-Emory data set for two reasons. First, to test our techniques on data acquired in different settings and scored by different expert surgeons. Secondly, we

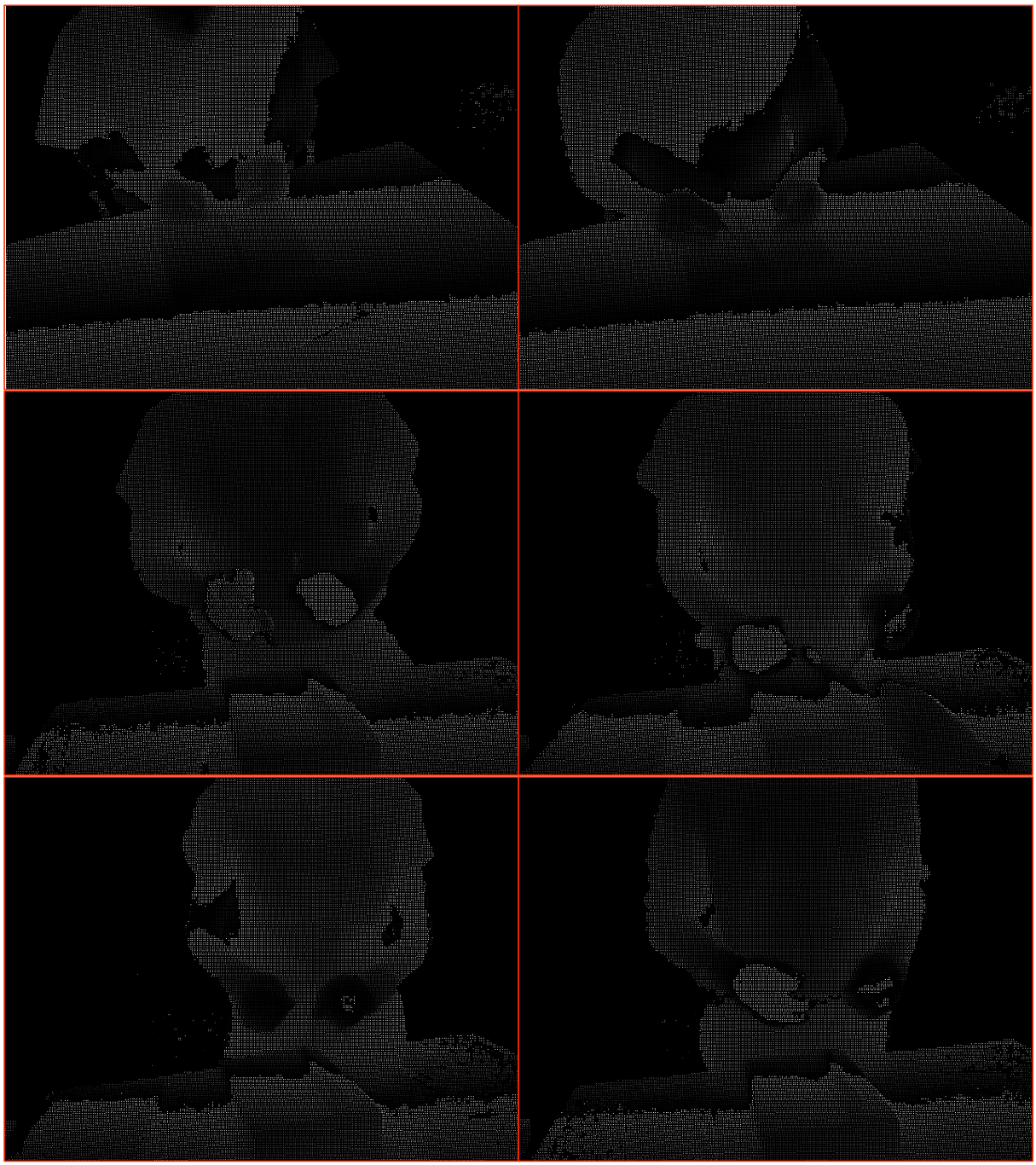


Figure 13: Aligned depth frames corresponding to RGB frames.

also need to isolate individual hand motions for dexterity analysis. In this data-set, we use the colored gloves to isolate left and right hand motions, which will be used for dexterity analysis (Chapter 7). With motion texture analysis (Chapter 3), we observed that skill information depends on fine textural details in the frame kernel

Figure 14: Depth masks overlaid on RGB frames.

matrices. Different types of motion data (STIPs, geometric (blob) features, acceleration *etc.* might capture different level of motion granularity. To test our technique with different types of motion data, we use aligned depth images and extract the depth information from the left and the right hand. We compare depth features along with

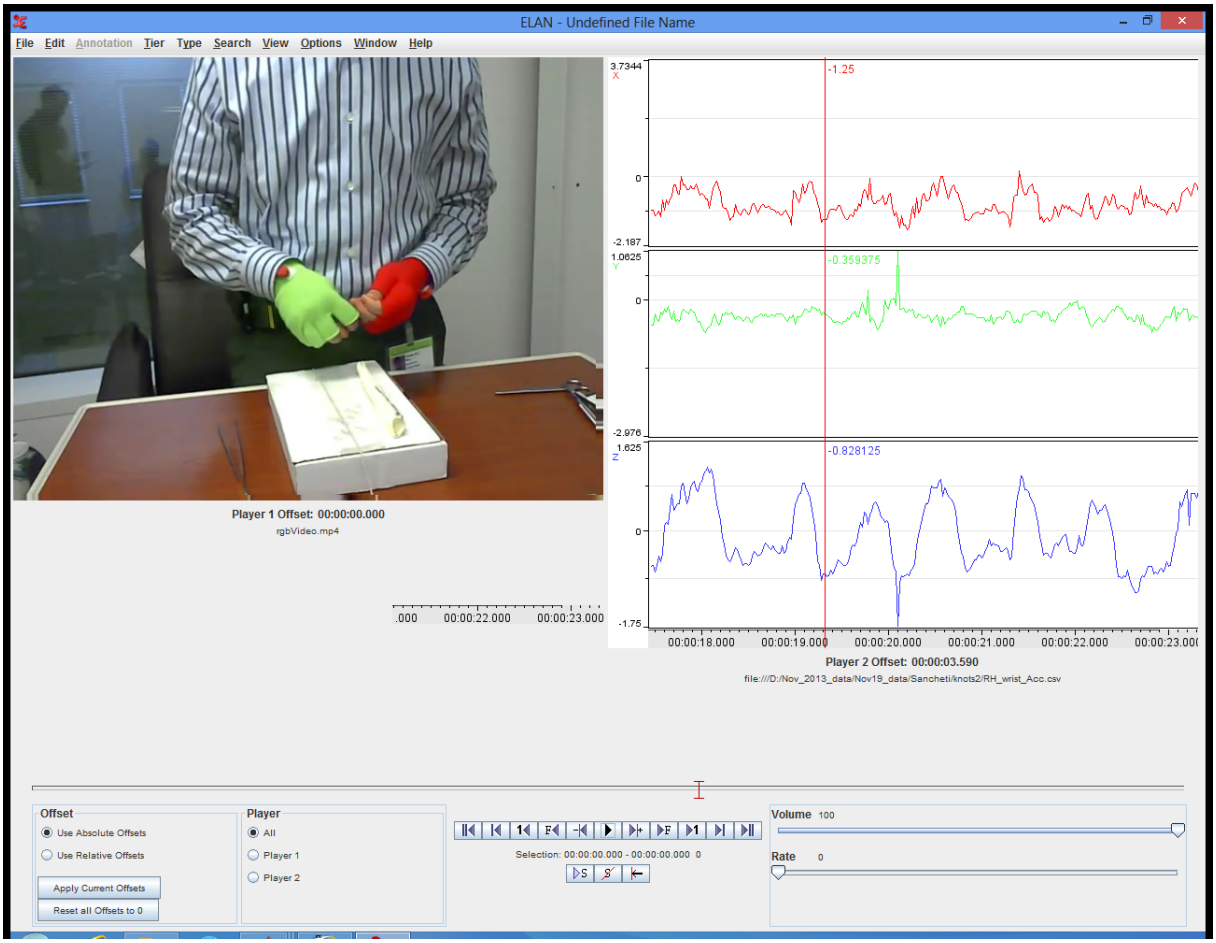


Figure 15: Screen shot of a knot tying video from GT-Emory data along with X, Y, and Z acceleration data displayed in ELAN software used for synchronization of video and acceleration data.

STIPs, blob and acceleration data to test the relative significance of different motion features in skill assessment. Besides comparing different types of motion features, we also use this data set for dexterity analysis.

CHAPTER V

SKILL CLASSIFICATION USING MOTION TEXTURE ANALYSIS

Summary *Motion texture (MT) analysis is used for relative skill assessment (classification). Sequential motion information is incorporated resulting in sequential motion texture (SMT) analysis. Comparison with state-of-the-art methods shows better performance of MT and SMT for different OSATS criteria.*

In this chapter, we extend our motion texture analysis technique for video based skill assessment. We envision an automated skill assessment application as follows. As part of their medical training, students will practice standard surgical procedure such as suturing. They use standard surgical instruments and practice the procedure on simulation equipment. A camera installation records these training sessions and our automated procedure assesses the quality of the suturing activities according to the OSATS criteria.

We perceive skill evaluation in two ways. For some applications or procedures, it might be sufficient to just classify the trainees into different skill groups. We define this process as *relative skill assessment* as participants are categorized into different skill groups relative to each other. However, in some applications, calibrated skill scoring may be required. We define the process of calibrated skill scoring as *absolute skill assessment*. In this chapter, we demonstrate the effectiveness of our motion texture analysis approach for relative skill assessment. In Chapter 6, we will present our analysis for absolute skill assessment using motion texture analysis.

In Section 5.1, we describe our methodology to encode motion data from videos

into frame kernel matrices. We observe that video motion dynamics appear as textured patterns in the frame kernel matrices as we demonstrated earlier for MOCAP data in Chapter 3. In Section 5.2, we include temporal information in our analysis to represent the sequential nature of surgical tasks resulting in *Sequential Motion Texture* (SMT) analysis. In order to obtain motion texture features, we compute GLCM texture features (Section 5.3) followed by feature selection to extract skill relevant features (Section 5.4). We use the selected features for skill classification using both simple motion texture features (no temporal information) and sequential motion texture features (with temporal information) as shown in our experimental evaluation (Section 5.5). We also analyze the effect of different parameters on the performance of our system and compare our methodology to two techniques used in activity recognition: Bag-of-Words (BoW) and Augmented Bag-of-Words (A-BoW).

Figure 16 gives an overview of the proposed procedure. The input to the system is a video recording of someone performing suturing procedure and the output is an automated skill assessment according to the seven OSATS criteria. In the following, we will discuss the technical details of the developed framework.

5.1 Frame kernel matrices from videos

In Chapter 4, we observed that the textural characteristics of frame kernel matrices can be used to extract skill relevant information. We used MOCAP data where the X, Y, and Z motion trajectories were used and the three dimensional data were encoded into $N \times N$ frame kernel matrices, where N is the number of frames. To obtain similar motion characteristics from video data, we need to extract the motion features from the videos. For MOCAP, we used the optical markers to acquire motion trajectories from specific hand locations. To achieve similar effect for videos, we need to cluster the motion features into distinct groups belonging to specific moving objects in the videos. We used three dimensional X, Y, and Z trajectory data in MOCAP. For

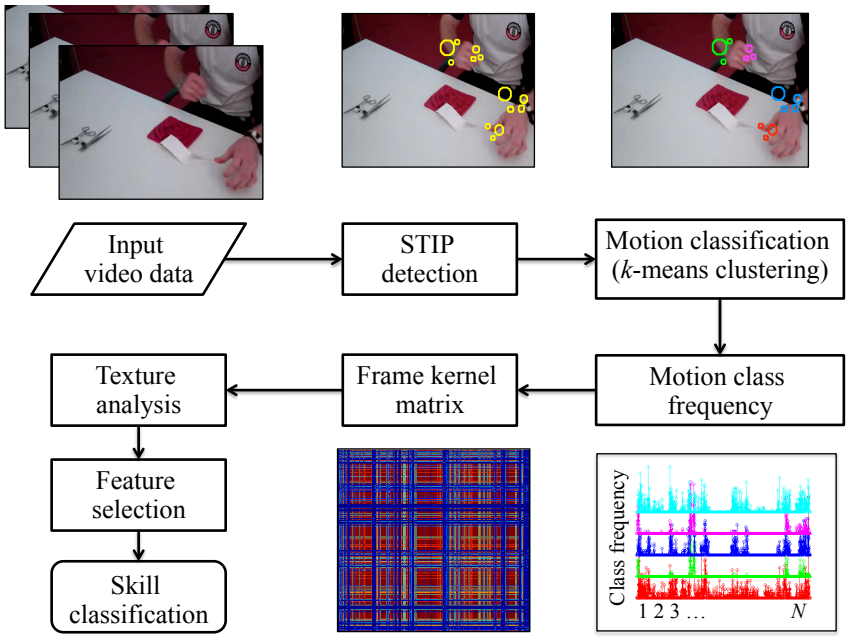


Figure 16: Motion texture analysis framework for OSATS skill classification.

videos, we need to condense the motion feature data to $N \times k$ dimensions, where N is the number of frames in the video and k is an integer.

5.1.1 Motion features

Different types of motion features have been proposed in literature and are used for various purposes such as for activity recognition [59]. For example, the spatiotemporal version of the Harris corner detector [24] proposed by Laptev [31], known as the Spatio-temporal Interest Point (STIP) detector, has been shown to work well in action classification [51].

Laptev [31] proposed an extension of the Harris corner detector with the modified Harris corner function as

$$H = det(\mu) - ktrace^3(\mu), \tag{21}$$

where

$$\mu = g(.;\sigma^2,\tau^2) * \begin{pmatrix} {L_x}^2 & L_x & L_xL_t \\ L_xL_y & {L_y}^2 & L_yL_t \\ L_xL_t & L_yL_t & {L_t}^2 \end{pmatrix} \tag{22}$$

and $g(.;\sigma^2,\tau^2)$ is a 3D Gaussian smoothing kernel with a spatial scale σ and a temporal scale τ. $L_{x,y,z}$ are the gradient functions along the x, y, and z directions. We use Laptev's STIP implementation[1], with default parameters and sparse feature detection mode and compute the STIPs with different values of σ^2 and τ^2 from the sets $\{4, 8, 16, 32, 64, 128\}$ and $\{2, 4\}$ respectively, with k set to be 0.005.

Histogram of Oriented Gradients (HOG) and Histogram of Optical Flow (HOF) are computed on a three-dimensional video patch in the neighborhood of each detected STIP. The patch is partitioned into a grid with 3×3 spatio-temporal blocks. Then, 4-bin HOG descriptors and 5-bin HOF descriptors are obtained for all the blocks. The 72-element HOG and the 90-element HOF descriptors are concatenated to get a 162-element HOG-HOF descriptor as described in [59]. We down-sampled the videos by half in both spatial and temporal dimensions to reduce the STIP computation time. Figure 17(left column) shows sample frames with detected STIPs.

5.1.2 Learning motion classes

For each video, we need to summarize the motion features into a $N \times k$ matrix, where N is the number of frames and k is an integer depending on the number of moving entities in the video. To obtain motion features for distinct moving entities in the videos, we classify the STIPs into motion classes as follows. First, we collect all the detected STIPs and their corresponding HOG-HOF descriptors from two videos of an expert surgeon. The expert motions are more distinct and uncluttered as compared to non-experts. Thus, clusters obtained from expert videos can be used to obtain motion components for different moving entities in the videos. We cluster these expert STIPs into k distinct clusters by applying k-means clustering to their HoG-HoF descriptors.

Each cluster of points thus obtained, represents a distribution for a particular motion class in the data. We assign the STIPs from remaining videos to each of the

[1] http://www.di.ens.fr/%7Elaptev/download.html#stip

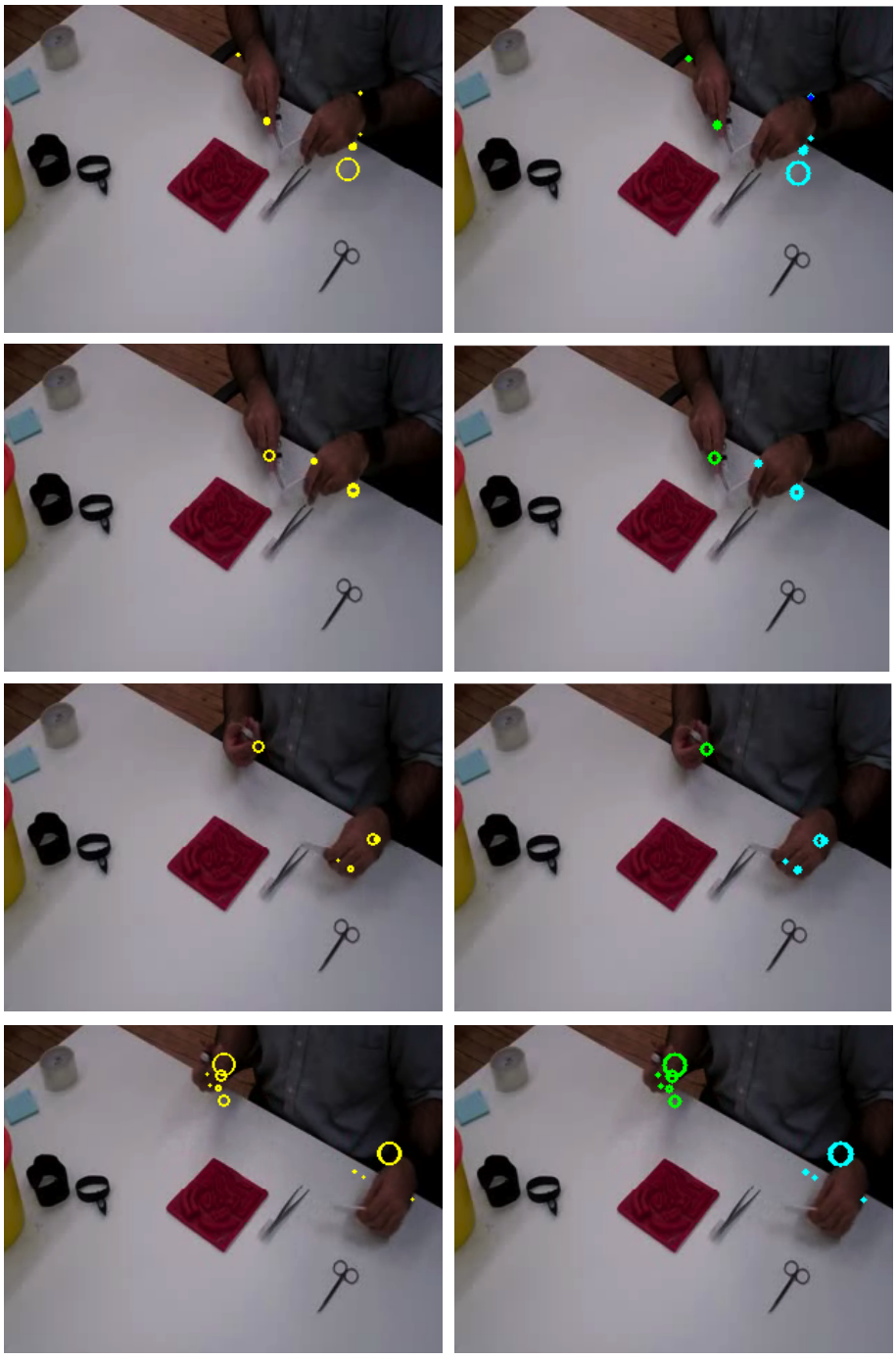


Figure 17: Left column: Detected STIPS in different frames represent the moving objects in the scene, Right column: STIPs classified into distinct motion classes.

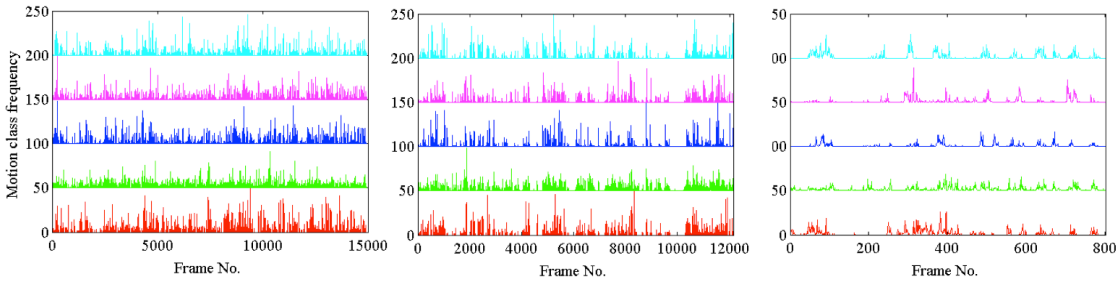


Figure 18: Motion class frequencies for a novice (left), an intermediate (center) and an expert (right) surgeon. The five classes are plotted at an offset of 50 (on y axis) for clarity. Note that the novice motions are more frequent and exist in almost all frames for all motion classes as compared to fewer motions for intermediate and expert surgeons. The plots correspond to a single suturing and knot tying task and demonstrate that experts use fewer motions than novices as reported in [15].

learnt motion distribution based on minimum Mahalanobis distance of a given STIP point from the cluster distribution. Figure 17(right column) shows sample frames with the detected STIPs classified into three distinct motion classes.

5.1.3 Computing frame kernel matrices

To obtain the frame kernel matrices, we further process each video to compute class frequency counts for each of the k classes at each frame. We represent these counts in a $k \times N$ matrix $\mathbf{X}$, where N is the number of frames in the video. Each element in $\mathbf{X}$, $x(p, q)$, represents the number of STIP points in qth frame and belonging to the pth cluster. Figure 18 shows sample class frequency counts for three subjects with different skill levels for $k = 5$ motion classes. The time frequency matrix $\mathbf{X}$ is used to obtain the $N \times N$ frame kernel matrix $\mathbf{K}$ given by $\mathbf{K} = \phi(\mathbf{X})^T \phi(\mathbf{X})$. Each entry in $\mathbf{K}$, $\kappa_{ij} = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$, defines similarity between two frames x_i and x_j using a kernel function $\phi(x_i)^T \phi(x_j)$. The parameter σ (the standard deviation) controls the flexibility of the kernel. Small values of σ tend to make the kernel matrix close to an identity matrix. Large values result in a constant kernel matrix. In general, σ is selected empirically to avoid these extremes. We select σ empirically and set it to be the average distance from twenty percent of the closest neighbors as described in [65]

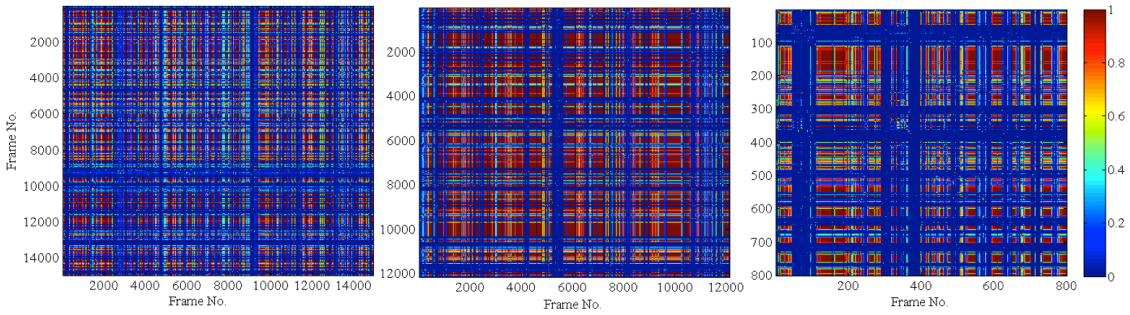


Figure 19: Frame kernel matrix corresponding to motion class frequency in Figure 18 for a novice (left), an intermediate (center) and an expert (right) surgeon.

to obtain textured frame kernel matrices. Figure 19 shows the frame kernel matrices corresponding to the motion class time series in Figure 18.

In summary, the STIPs represent the moving entities in the videos. After learning k motion classes from expert STIPs, the motion dynamics in a given video are represented by the $k \times N$ time frequency matrix $\mathbf{X}$, where N is the number of frames in the given video. The $N \times N$ frame kernel matrix thus encodes the motion dynamics of the person who performed the surgical task in the video. The pixel intensity transitions in the frame kernel matrix (Figure 19) correspond to motion dynamics (Figure 18), and vary according to the skill level of the surgeon.

5.2 Sequential motion texture (SMT) analysis

The texture features derived from the frame kernel matrix capture the overall motion quality without temporal information. However, surgery is a procedural task performed in a sequential manner with one step followed by another. We introduce this information by dividing the k-dimensional time frequency matrix $\mathbf{X}$ into equally sized temporal windows such that each window contains equal proportion of the STIPs corresponding to largest motion class in a given video. For example, if the largest motion class has 1000 STIPs in the whole video, then the time series can be divided into $W = 10$ equally sized windows with approximately 100 points in each bin. Using equal sized bins, we intend to group the motion energy into equivalent segments

that could replicate the repetitive and procedural behavior of surgical motion. We summarize the sequential motion texture technique in Algorithm 2.

For each time window, we calculate the frame kernel matrix and concatenate the 20 GLCM texture features to obtain a $20 \times W$ feature vector for each video, where W is the number of windows. We evaluate our framework with and without time windowing to study the effect of including sequential information. We refer to the time windowing setup as the *Sequential Motion Texture* (SMT) analysis and the one without time windowing as simply the *Motion Texture* (MT) analysis. Figure 20 shows the motion classes grouped into time windows for SMT and Figure 21 shows the frame kernel matrices for each of the ten time windows.

5.3 Feature extraction

We now have a matrix based representation of our videos *i.e.* STIP based time series representation in terms of motion class frequency at each frame encoded into a frame kernel matrix. This frame kernel matrix is now analyzed in order to infer about the underlying skill of the surgeon who performed the recorded procedure. For relative

Algorithm 2 - Sequential motion texture features

Require: Surgical videos in set V

Step 1: $\forall v \in V$, compute STIPs (spatio-temporal interest points) and 162-element HoG-HoF (histogram of oriented gradients-histogram of optical flow) descriptors [59].

Step 2: Cluster STIPs from two experts by applying k-means (k=5) to HoG-HoF features. We select k=5 since we expect approximately five moving entities in the videos: surgeon's two hands and the three instruments (forceps, needle-holder, and scissors).

Step 3: Assign STIPs for remaining videos to the k clusters learnt in step 2 using minimum Mahalanobis distance.

Step 4: Compute motion class counts, $\mathbf{X}$, for each of the k clusters. Each entry in the $N \times k$ motion class count matrix $\mathbf{X}$, $x(n, q)$ represents the number of STIPs belonging to the n^{th} frame and the q^{th} cluster, where N is the number of frames in the video.

Step 5: Compute time windows (bins) as follows:

Find the motion cluster corresponding to maximum class membership *i.e.*

$k_{max} = \arg\max_k \mathbf{X}$

Compute W time windows such that motion class counts for the k_{max} cluster are equally distributed into W windows *i.e.*

Step 6: Compute W frame kernel matrices, $\mathbf{K}_w$, where $w = 1, 2, ..., W$

Step 7: Compute 20 GLCM features for each of the W frame kernel matrices and concatenate them to obtain a $20 \times W$-element *Sequential Motion Texture* (SMT) feature vector.

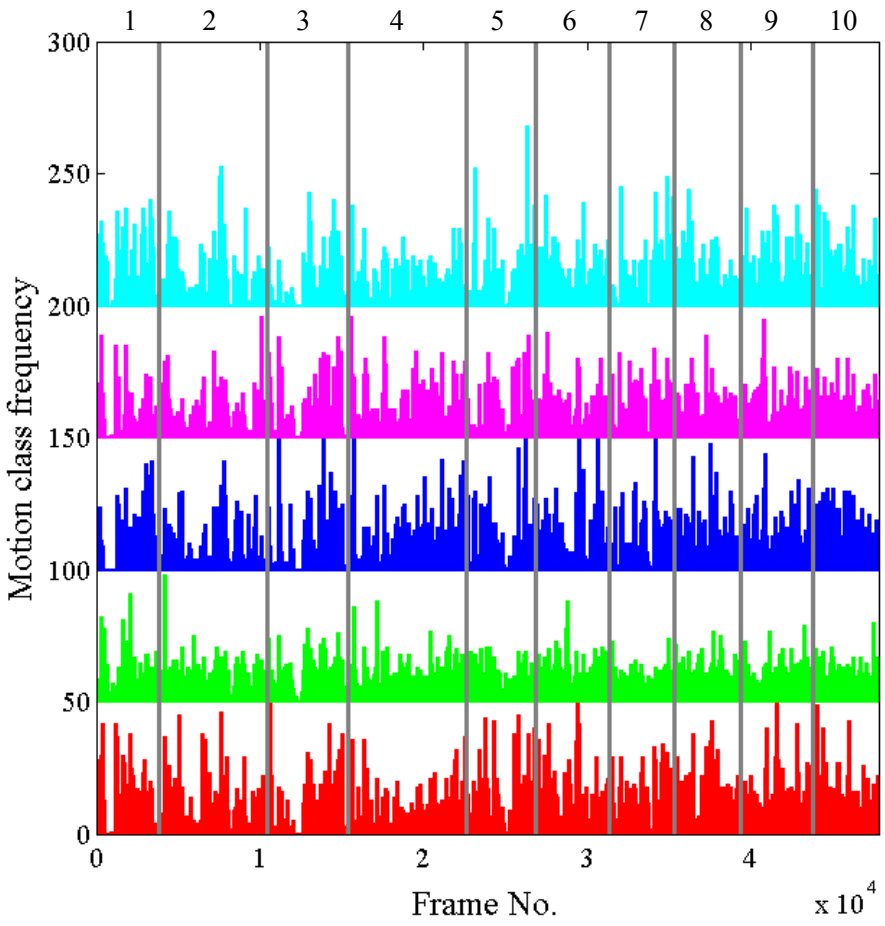


Figure 20: Motion class frequencies for SMT with W=10 time windows. The five classes are plotted at an offset of 50 (on y axis) for clarity. Note that the time windows are of varying duration depending on the motion counts.

skill assessment, we translate this into a classification task (seven classifiers, one for each OSATS criteria and the three classes – novice, intermediate and experts).

As usual for classification tasks, we first extract feature vectors from the video (encoded as frame kernel matrix). We extract the texture patterns in the frame kernel matrix using Gray Level Co-occurrence Matrix (GLCM). As explained in Chapter 3 earlier, GLCM encodes the spatial relation of different intensity levels in an image and the texture statistics from GLCM have been used as features for image classification tasks [22, 54, 13, 16]. However, we use GLCMs here to derive feature vectors from frame kernel matrices, which in turn, encode motion dynamics in our surgical videos.

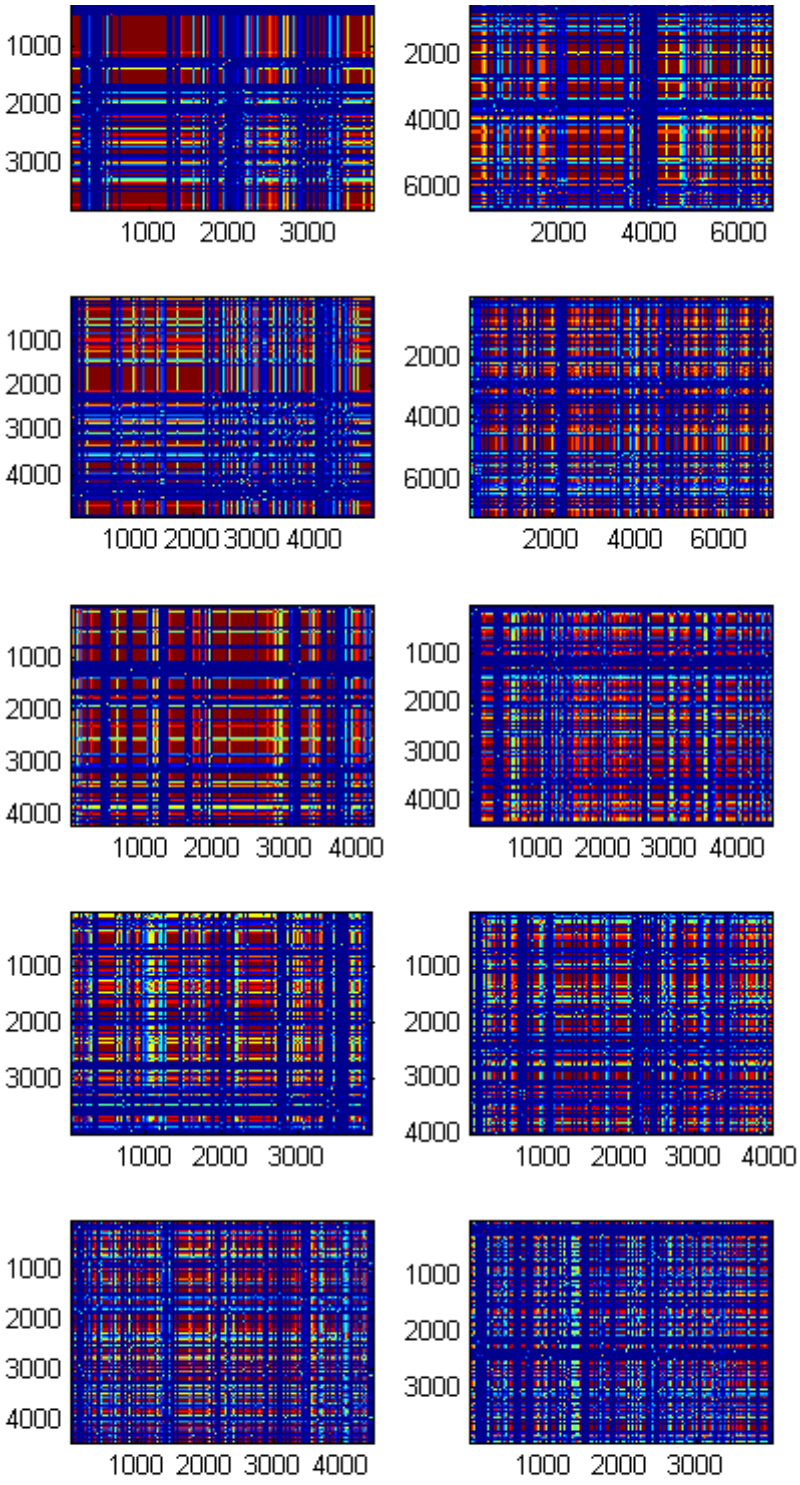


Figure 21: Kernel matrices for $W = 10$ time windows corresponding to motion classes in Figure 20

We employ $N_g \times N_g$ dimensional Gray Level Co-occurrence Matrices (GLCM), calculated for N_g gray levels and eight directions ($0° - 360°$ in steps of $45°$) at a spatial offset of 1 pixel. Averaging (and normalizing) over the GLCM provides the final representation, which is used to compute twenty standard texture features as proposed in literature. These features are [22, 54, 13, 16]:

1. Autocorrelation (f_1)
2. Contrast (f_2)
3. Correlation (f_3)
4. Cluster prominence (f_4)
5. Cluster shade (f_5)
6. Dissimilarity (f_6)
7. Energy (f_7)
8. Entropy (f_8)
9. Homogeneity (f_9)
10. Maximum probability (f_{10})
11. Sum of squares variance (f_{11})
12. Sum average (f_{12})
13. Sum variance (f_{13})
14. Sum entropy (f_{14})
15. Difference variance (f_{15})
16. Difference entropy (f_{16})

17. Information measure of correlation 1 (f_{17})

18. Information measure of correlation 2 (f_{18})

19. Inverse difference normalized (f_{19})

20. Inverse difference moment normalized (f_{20})

GLCM features encode various texture properties. For instance, correlation measures the grey level linear dependency between the pixels at a specified position relative to each other. Contrast measures the local intensity variations while cluster shade and cluster prominence measures the uniformity and proximity in a perceptual manner. Energy or angular second moment is a measure of homogeneity in the image while dissimilarity measures the total variation present in the image. Homogeneity (or inverse difference moment) measures the image homogeneity and takes larger values for smaller gray tone differences in pixel pairs. These features have been used for classification of images based on texture. However, we use them here to encode video motion dynamics, which in turn, are represented as textured patterns in frame kernel matrices.

5.4 Feature selection

Some of the GLCM texture statistics are highly correlated with one another and may be redundant [14]. Also, some features might be noisy and irrelevant for the skill classification task. In addition, the MT texture analysis yields a 20-element feature vector while SMT has $(20 \times W)$-element feature vector. To derive skill relevant features and to compensate for the effect of more features (over-fitting) in SMT as compared to MT, we perform feature selection.

Feature selection is the process of selecting a subset of relevant features. Feature data may contain redundant or irrelevant features. Redundant features are those that

provide no more information than the currently selected features. Irrelevant features provide no useful information in any context.

A feature selection algorithm can be viewed as the combination of a search technique for new feature subsets and evaluation criteria, which scores the different feature subsets. The simplest algorithm is to test each possible subset of features finding the one, which minimizes the error rate. There are two standard approaches for feature selection – filter methods and wrapper methods. Wrapper methods utilize the learning machine of interest as a black box to score subsets of variable according to their predictive power. Filters based methods select subsets of variables as a pre-processing step and are independent of the chosen predictor.

For small feature sets such as our twenty GLCM features, wrapper methods can be used for feature selection. Since our feature size is small, we use the simplest greedy search algorithm Sequential Forward Feature Selection (SFFS) [45] to select a subset of relevant features for each OSATS criteria. SFFS starts from the empty set and sequentially adds the feature that results in the highest objective function when combined with the features that have already been selected. We use a Nearest-Neighbor (NN) classifier with cosine distance metric as a wrapper function for SFFS and select the feature subset with minimum classification error in leave-one-out cross-validation (LOOCV). The maximum size of selected feature subset was limited to 20 to allow comparable number of features for both MT and SMT approaches.

5.5 Experimental Evaluation

We use the Newcastle data set to demonstrate the efficacy of our motion texture analysis technique for relative skill assessment. We group the participants into three categories according to their expertise: low (OSATS score ≤ 2), intermediate ($2 <$ OSATS score ≤ 3.5) and high ($3.5 <$ OSATS score ≤ 5) expertise levels to train our models with sufficient samples per class. Table 6 shows the number of videos used

Table 6: Number of samples for different expertise levels

	RT	TM	IH	SH	FO	KP	OP
Novice	2	9	8	10	3	8	6
Intermediate	14	15	16	15	16	9	17
Expert	15	7	7	6	12	14	8

in our study corresponding to three expertise levels for each OSATS criteria. We used our MT and SMT frameworks to classify all participants into three pre-defined expertise groups (low, intermediate, and high), based on the surgical OSATS criteria. We use our framework to train seven classifiers corresponding to each OSATS criteria.

5.5.1 Generalization across different users

To test the generalization across different users, we performed experiments with two setups. In first setup, a single video was left out, *i.e. leave one sample out (LOSO)*, for testing while training was done on the remaining videos. Since, we have two videos from each subject (except for one subject), we also use a setup in which all the videos from a single user are left out for training, *i.e. leave one user out (LOUO)* while training was done on the remaining videos. In LOUO setup, the training data does not contain any video from the test subject and the left out test videos present unseen data to the classifier. Thus, the classification accuracy of seven OSATS classifiers in LOUO set-up indicates their generalization capability to classify previously unseen data.

5.5.2 Effect of different parameters

We also test the effect of different parameters on the classification accuracy. The number of gray levels used to evaluate GLCM (N_g) represents the level of granularity to encode motion dynamics. With fewer gray levels, only limited number of motion transitions can be encoded. This may be sufficient to represent simple activities, however, a complex activity such as surgery might require fine-grained analysis. To test this, we varied the number of gray levels from 2^3-2^8 keeping other parameters

constant ($k = 5$ motion classes and $W = 10$ time windows for SMT). We also computed the classification accuracy with varying number of STIP motion clusters (k) from [2-10] with constant $N_g = 8$ gray levels and $W = 10$ time windows (for SMT). For SMT, we also test the effect of varying the number of time windows W from [2-16].

5.5.3 Comparison with standard activity recognition methods

We also compared our methods with the state-of the-art *Bag-of-Words* (BoW) models (built directly from the HoG-HoF descriptors), that are typically used for video-based action recognition [59] and have also been used for surgical gesture recognition [62, 23].

The bag-of-words model was originally developed for document representation. The basic idea is to first define a codebook that contains a set of code words, which are then used to represent a document as a histogram of the code words, where each entry is the count of a code word occurring in the document. Although the order information of words is ignored, the bag-of-words model still captures the document information effectively because of the significance of frequency information of code words in documents [57].

Recently, the bag-of-words model is extended to analyze images and videos in computer vision [43]. BoW techniques also represent the state-of-the-art for video-based activity recognition with applications to realistic and diverse settings [33]. Local patches extracted from images or videos are treated as words and the codebook is constructed by clustering all the local patches in the training data. To apply BoW model to images, we can treat an image as a document. However, we need to define the "words" in the images. To achieve this, standard BoW methods usually includes following three steps: Feature detection (computer vision), feature description and code book generation.

For action recognition, the BoW model is typically constructed using visual codebooks derived from local spatio-temporal features [60]. More recently BoW based

Algorithm 3 - Temporal and structural modeling using BoW

Require: Surgical videos in set V

Step 1: $\forall v \in V$, compute STIPs and 162-element HoG-HoF descriptors [32].

Step 2: Build bag-of-visual-events through k-means clustering, with $k = 50$.

Step 3: $\forall v \in V$, create *event sequences*, $E_v =< e_1, e_2, \ldots, e_n >$, by assigning each STIP to its nearest cluster.

Step 4: $\forall E_v$, augment E_v with *temporal events*, $\tau_{i,j}$ (where $\tau_{i,j}$ is the time elapsed between the end of event e_i and the start of event e_j, where $j > i$), such that $E_v =< e_1, \tau_{1,2}, e_2, \ldots, e_i, \tau_{i,j}, e_j, \ldots >$.

Step 5: Divide the total time into N bins and quantize the temporal events, such that $\forall E_v$, we have $E_v =< e_1, \psi(\tau_{1,2}), e_2, \ldots, e_i, \psi(\tau_{i,j}), e_j, \ldots >$, where ψ is function that maps each temporal event to its respective bin.

Step 6: Capture *local structure* and causality information by extracting n-grams from event vectors, E_v, using a moving window of size n [21, 8]. With $n = 3$, $\forall E_v$, we have $E_v =<$ "$e_1\psi(\tau_{1,2})e_2$","$e_2\psi(\tau_{2,3})e_3$",...,"$e_i\psi(\tau_{i,j})e_j$",... $>$.

Step 7: Capture *global patterns*: Create a sub-space of regular-expressions and generate R regular-expressions by randomly sampling that sub-space:

for $1 \leq i \leq R$ **do**

Generate a new random regular-expressions r_i [8]

Add r_i to E_v

end for

Step 8: Using event vectors E_v, train and test using the Vector Space Model (VSM) framework: (1) Re-weight each word in E_v using the TF (term frequency) and IDF (inverse document frequency) metrics and (2) Classify using k-NN with the cosine similarity distance metric.

approaches have focused on recognizing human activities in more realistic and diverse settings [33]. However, when activities are represented as bags of words, the underlying structural (causal and sequential) information provided by the ordering of the words is typically lost. To address this problem, recent approaches have included temporal information into BoW models. For example, *n-grams* have been used to represent activities in terms of their local event sub-sequences [21]. While this preserves local structural information, adding absolute and relative temporal information results in more powerful and expressive BoW representations as shown by Bettadapura *et al.* [8] in the Augmented BoW (A-BoW) model. We compare our MT and SMT approaches with BoW and A-BoW methods. Algorithm 3 (steps 1-2) describes the standard BoW models, which are then augmented with temporal and structural information (Algorithm 3, steps 3-8).

Table 7: Percentage of correctly classified videos using all features.

OSATS	MT (LOSO)	MT (LOUO)	SMT (LOSO)	SMT (LOUO)
RT	77.4% (24/31)	74.1% (23/31)	70.9% (22/31)	74.1% (23/31)
TM	58.0% (18/31)	61.2% (19/31)	80.6% (25/31)	80.6% (25/31)
IH	61.2% (19/31)	51.6% (16/31)	70.9% (22/31)	70.9% (22/31)
SH	58.0% (18/31)	54.8% (17/31)	61.2% (19/31)	61.2% (19/31)
FO	58.0% (18/31)	58.0% (18/31)	64.5% (20/31)	64.5% (20/31)
KP	54.8% (17/31)	54.8% (17/31)	70.9% (22/31)	70.9% (22/31)
OP	54.8% (17/31)	58.0% (18/31)	77.4% (24/31)	77.42% (24/31)

5.6 Results

We present the results using MT and SMT techniques as percentage of correctly classified videos using seven classifiers trained for each OSATS criteria. All results are compared against the ground truth provided by expert surgeon.

Table 7 shows the results using all features for classification in LOSO and LOUO setups. Note that with MT, higher classification accuracy is obtained for qualitative criteria such as "respect for tissue" (RT). With SMT, higher classification accuracy is obtained for all OSATS (except RT) as compared to MT approach. Table 8 (columns 2-5) shows the results using selected features. With feature selection, skill relevant features are extracted resulting in higher classification accuracy with a smaller subset of discriminating features. Also, note that there is not much difference in the classification accuracies between LOSO and LOUO setups. This indicates the generalization capability of the classifiers on unseen data.

Table 8: Percentage of correctly classified videos with selected features.

OSATS	MT (LOSO)	MT (LOUO)	SMT (LOSO)	SMT (LOUO)	BoW (LOSO)	A-BoW (LOSO)
RT	83.8% (26/31)	83.8% (26/31)	100% (31/31)	100% (31/31)	66.6% (42/63)	73.0% (46/63)
TM	80.6% (25/31)	83.8% (26/31)	100% (31/31)	100% (31/31)	50.7% (32/63)	74.6% (47/63)
IH	70.9% (22/31)	70.9% (22/31)	100% (31/31)	100% (31/31)	50.7% (32/63)	68.2% (43/63)
SH	74.1% (23/31)	70.9% (22/31)	96.7% (30/31)	93.5% (29/31)	69.8% (44/63)	73.0% (46/63)
FO	70.9% (22/31)	77.4% (24/31)	100% (31/31)	100% (31/31)	49.2% (31/63)	66.6% (42/63)
KP	61.2% (19/31)	58.0% (18/31)	100% (31/31)	96.7% (30/31)	60.3% (38/63)	80.9% (51/63)
OP	74.1% (23/31)	77.4% (24/31)	100% (31/31)	100% (31/31)	52.3% (33/63)	71.4% (45/63)

5.6.1 Effect of varying number of time windows W in SMT

Figure 22 shows the effect of varying the number of time windows for SMT with constant number of gray levels ($N_g = 8$) and motion classes ($k = 5$) in LOUO setup. With only two windows, lower classification rates are observed for all OSATS criteria. As the number of windows is increased, the performance improves. Multi-modal trend is observed for some OSATS criteria. For instance, OP performance peaks at 6 windows and then again at 9 windows. Similarly, IH performance peaks are at 5 windows and at 10 windows. This may be caused by possible periodicity in the time series due to repetitive nature of the suturing task.

5.6.2 MT Vs. SMT

Figure 23, 24, and 25 show the confusion matrices with MT and SMT techniques corresponding to classification accuracy in Table 8 (columns 2 and 4 corresponding to LOSO setup). SMT performs better than MT for all OSATS. For RT, we have only two novice videos (Table 6, column 2) which are classified correctly with SMT even though only one sample is available for training. For most of the criteria, majority

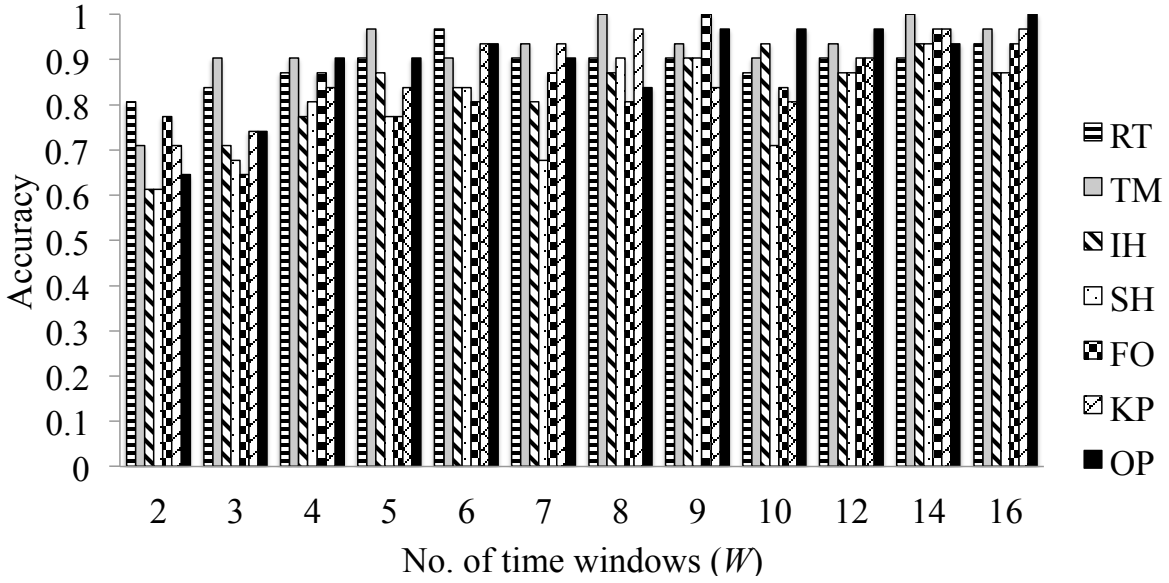


Figure 22: Effect of varying the number of time windows in the SMT approach

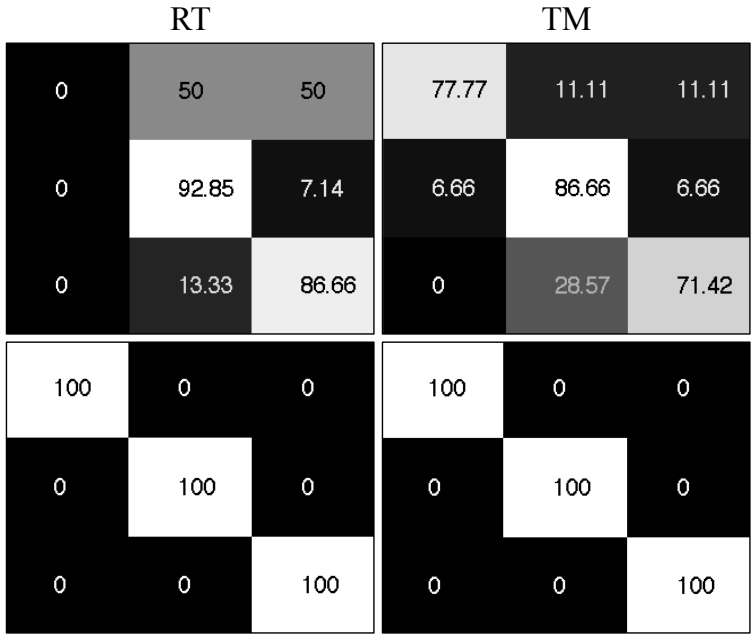


Figure 23: Confusion matrices for RT and TM OSATS criteria corresponding to classification accuracy in Table 8 with MT (top row) and SMT (bottom row).

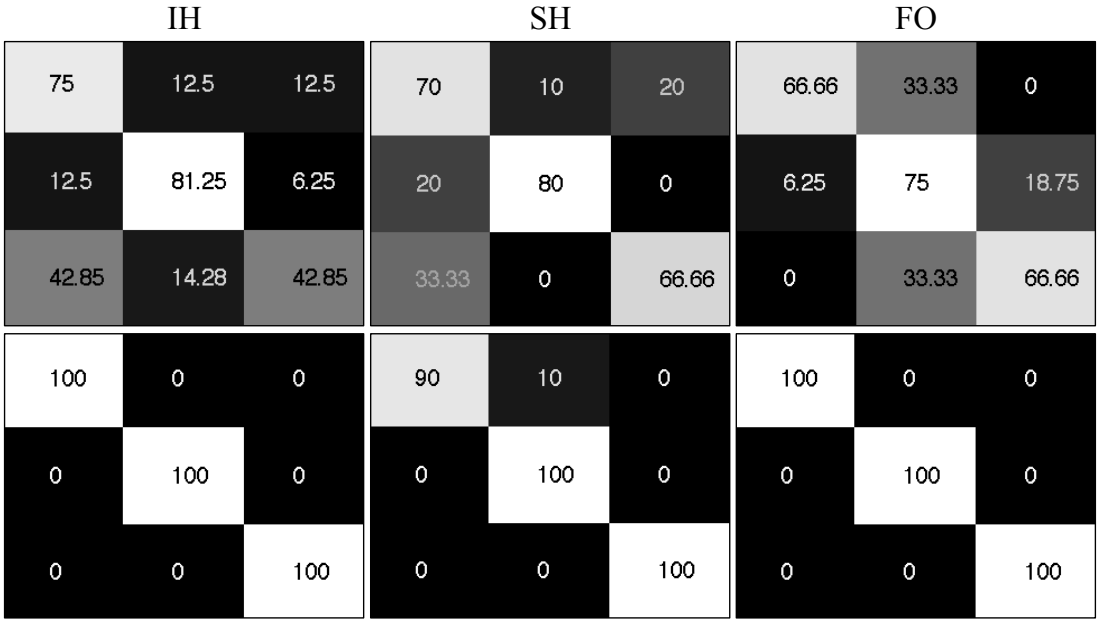


Figure 24: Confusion matrices for IH, SH, and FO OSATS criteria corresponding to classification accuracy in Table 8 with MT (top row) and SMT (bottom row).

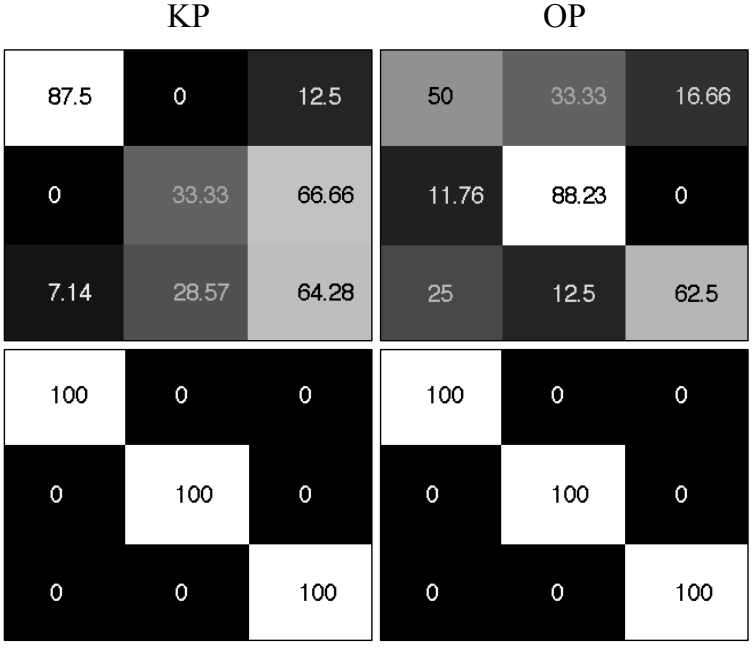


Figure 25: Confusion matrices for KP and OP OSATS criteria corresponding to classification accuracy in Table 8 with MT (top row) and SMT (bottom row).

of the videos come from intermediate subjects. Despite very few samples to train for novice and expert classes, SMT is able to discriminate the expertise levels. Figure 24 shows the confusion matrices for KP and OP criteria. Note that majority of the subjects were knowledgeable about the procedure (14 experts for KP in Table 6), while only 8 were rated as experts for the OP criteria. Thus, it is very important to provide assessments on individual OSATS criteria.

5.6.3 Comparison with BoW and A-BoW

We compare our MT and SMT approaches with BoW and A-BoW in LOSO setup. Table 8 (columns 6 and 7) shows the results. A-BoW captures the temporal and sequential motion aspects and performs better than standard BoW. Our MT technique captures the qualitative motion aspects and higher classification accuracy of 83.8% (an increase of 10% from A-BoW) is achieved for qualitative OSATS criteria such as RT. However, for sequential OSATS such as KP, A-BoW performs better (around 20% better than both MT and standard BoW). For TM, our MT approach performs slightly well with 80.6% correctly classified videos (an increase of 6% over A-BoW) possibly due to finer analysis of motion dynamics. For other OSATS, both A-BoW and MT show comparable performance but better than standard BoW. SMT, with both qualitative and sequential motion aspects performs better than MT, BoW, and A-Bow techniques. Note that BoW and A-BoW works [8] have used sixty-three videos since they used both the long range and close up videos for each participant. We used only long-range videos to ensure that the moving entities (hands, instruments etc.) exist in most of the frames. By using only thirty-one videos, we have less training data as compared to [8]. In addition, we also wanted to test our method in a LOUO set-up to test the generalization of classifier on unseen data.

5.6.4 Effect of varying gray levels (N_g) in the GLCM computation

Figure 26 shows the effect of varying the number of gray levels. MT captures the qualitative aspects (RT) and in general, there is no appreciable increase in accuracy with varying number of gray levels. With SMT, higher accuracy is achieved for all OSATS criteria as compared to MT and slight increase in accuracy is observed for IH, FO and KP with 256, 32 and 16 gray levels respectively (Figure 26, bottom).

5.7 Summary

The results presented in this chapter clearly demonstrate that MT and SMT approaches are suitable for assessment of surgical skills and perform better than BoW and A-BoW. BoW approach is useful for classification of human activities in general, that is, it may help in predicting *what* is being done in the video and sufficient literature exists to support its efficacy. For example, RMIS works on gesture recognition [62], and Haro et al. [23] reported good results for surgical gesture recognition using BoW since the goal is to classify *what (or which)* gesture is the test sample, however, for skill assessment, it is essential to assess the motion quality, *i.e.*, how *competent* the subject is in performing a given activity. The framework presented in this work successfully achieved automated OSATS assessment of surgical competency using video data.

Given the very encouraging assessment results of our case study, we believe that automatic surgical skill assessment has the potential to have a positive impact to real-world training settings in medical schools and teaching hospitals. In next chapter, we extend our framework for OSATS skill score prediction (absolute skill assessment) using regression analysis.

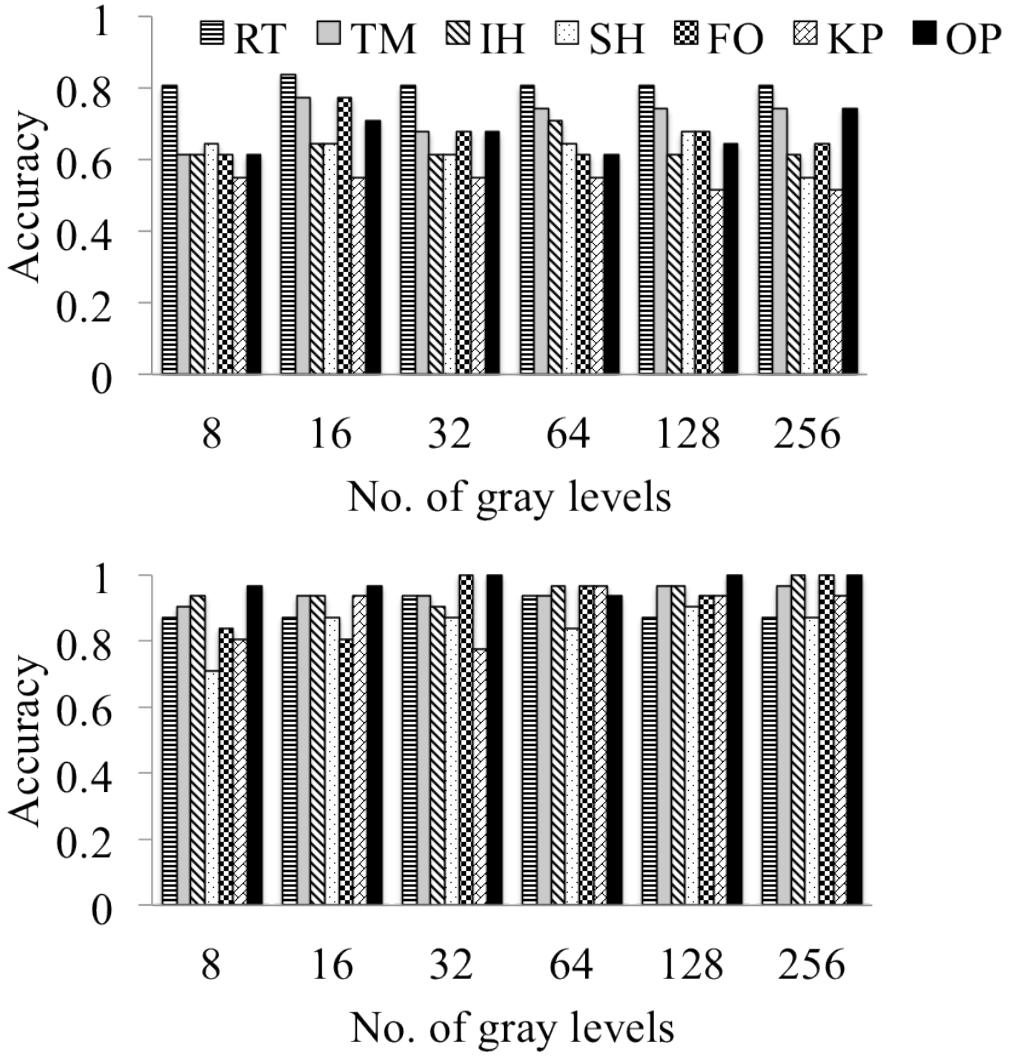


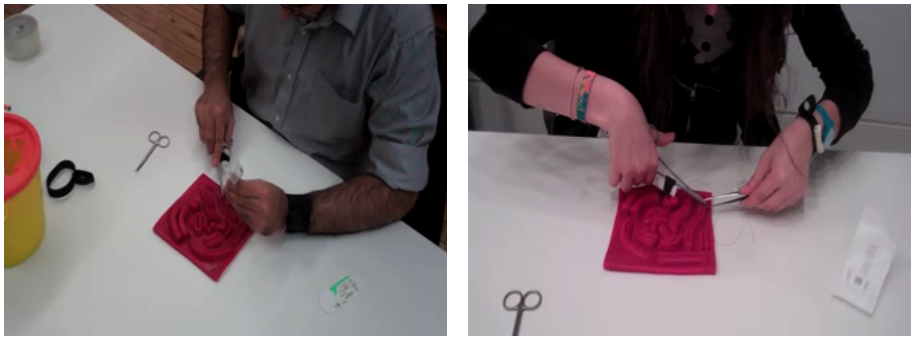
Figure 26: Top: Classification accuracy for various OSATS criteria with varying number of gray levels using MT approach; Bottom: Top: Same as top but using SMT approach.

CHAPTER VI

OSATS SKILL PREDICTION

Summary *Motion texture (MT) features are used for absolute skill assessment (prediction). Linear discriminant analysis is used to reduce the feature dimensions followed by regression analysis for skill score prediction. Statistically significant correlation is achieved between true and predicted scores.*

In Chapter 5, we demonstrated the skill classification using motion texture features. In this chapter, we go beyond simple skill categorization and predict the actual skill scores. We demonstrate that a linear regression function can be learnt in the reduced dimensional motion texture feature space using training data that can be used to predict the OSATS scores for test data. We achieve statistically significant correlation (p-value <0.01) between the ground-truth (given by domain experts) and the skill scores predicted by our framework. Figure 27 illustrates the outcome of proficiency evaluation approach for an exemplary surgical skill assessment task.



Ground truth: 5, Predicted: 4.73 Ground truth: 1.5 Predicted: 1.65

Figure 27: Proficiency evaluation based on motion texture analysis for an exemplary surgical skill assessment task. Ground truth quality judgments (from domain experts) are automatically replicated with high precision for expert (left) and novice surgeons (right).

Figure 28 shows the flow diagram for skill score prediction via motion texture analysis. Part 1 and 2 (in Figure 28 involve low-level motion feature extraction followed by computing the frame kernel matrices as explained in Chapter 5. Part 3 (Figure 28) shows the flow diagram for skill score prediction. We accomplish this by first extracting a lower dimensional feature subspace followed by learning a linear regression function in the reduced dimensional feature space. Next, we explain the technical details of score prediction framework.

6.1 Feature dimensionality reduction

After obtaining textural features using the GLCM or LBP-LC methods, we create a linear regression model using the training data. GLCM and LBP-LC features encode the fine motion details embedded in the frame kernel matrix. To cope with the curse-of-dimensionality, we use dimensionality reduction. In many applications, it is useful to achieve a broad categorization into coarse skill levels. Since our goal is to predict the skill score, we use Linear Discriminant Analysis (LDA) [7] to find a linear projection from the feature space that maximizes the separation of coarse skill levels obtained by grouping the skill scores into C categories. For example, if the ground truth scores range from 1–5, then three ($C = 3$) coarse skill levels could be: low (score ≤ 2), intermediate ($2 <$ score ≤ 3.5) and high ($3.5 <$ score ≤ 5).

6.1.1 Linear discriminant analysis

We use LDA to project the data into a subspace that can discriminate the participants into coarse skill groups based on the seven OSATS criteria. The coarse skill levels are linearly separable in the LDA subspace. This subspace with coarse skill separation can be used to learn the linear regression function for precise skill prediction. Other dimension reduction techniques such as principal component analysis (PCA) project data along the direction of maximum variance, which may not be discriminating

between different skill groups. LDA projects all the data points into lower dimensional subspace, which maximizes the between-class separation while minimizing their within-class variability. The dimensionality of the transformed space computed by the LDA is one less than the number of classes in the problem.

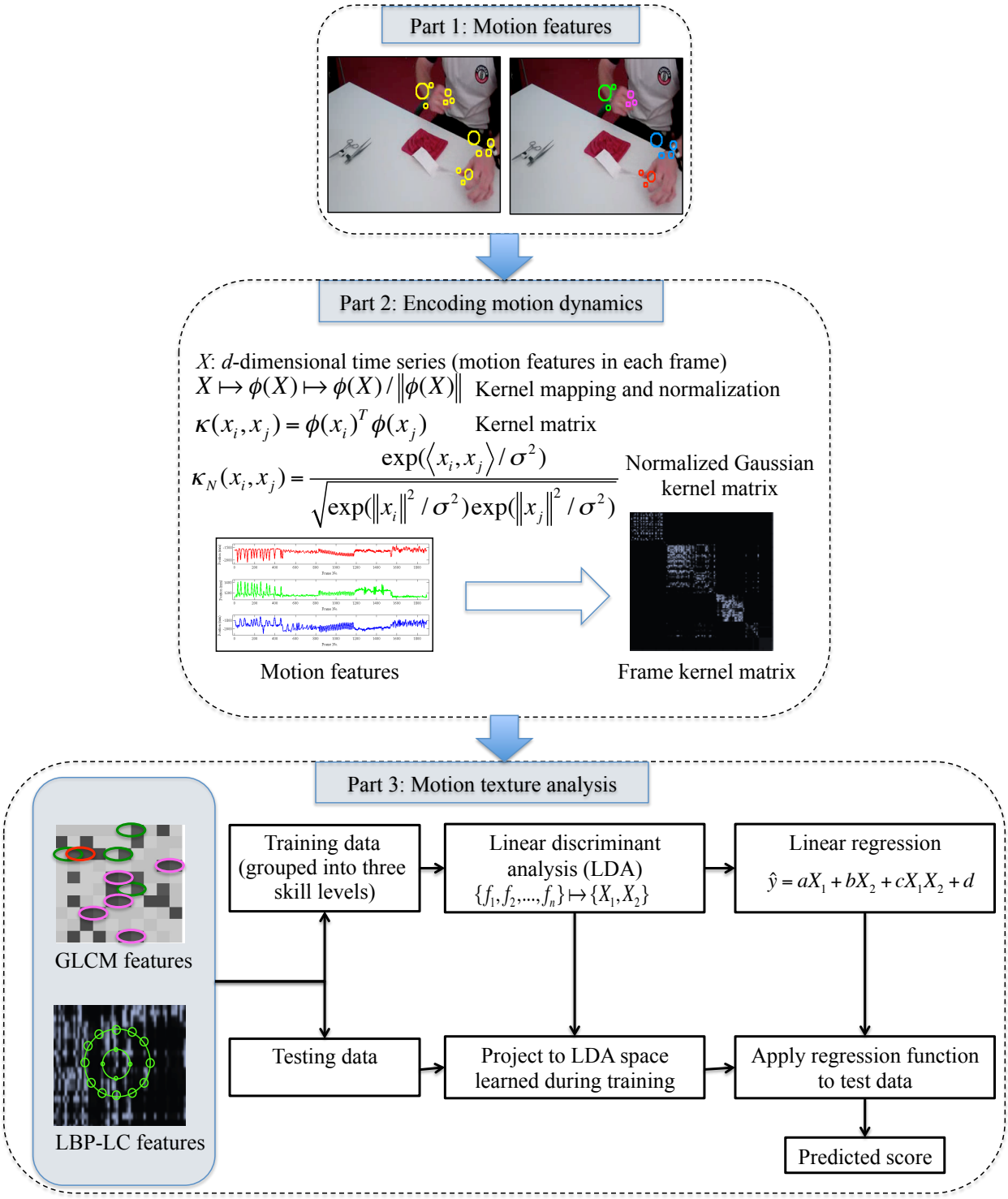


Figure 28: Motion texture analysis framework for skill score prediction.

The first step in the LDA is finding two scatter matrices referred to as the "between class" and "within class" scatter matrices. If we have C different classes or sample groups, then each sample group π_i has a class mean $\bar{x_i}$ given by,

$$\bar{x_i} = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{i,j}, \tag{23}$$

where there are N_i data points in class π_i. We can also define a sample group covariance matrix given by,

$$\Sigma_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x_i})(x_{i,j} - \bar{x_i})^T, \tag{24}$$

The global mean for the whole data set is given by,

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{C} N_i \bar{x_i} = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{N_i} x_{i,j}, \tag{25}$$

The *between-class* scatter matrix is defined as,

$$S_b = \sum_{i=1}^{C} (\bar{x_i} - \bar{x})(\bar{x_i} - \bar{x})^T \tag{26}$$

The *within class* matrix is defined as follows:

$$S_w = \sum_{i=1}^{C} (N_i - 1)\Sigma_i = \sum_{i=1}^{C} \sum_{j=1}^{N_i} (\bar{x}_{i,j} - \bar{x_i})(\bar{x}_{i,j} - \bar{x_i})^T \tag{27}$$

The main objective of LDA is to find a projection matrix ϕ_{LDA} that maximizes the ratio of determinants of S_b and S_w. Mathematically,

$$\phi_{LDA} = \arg\max_{\phi} \frac{|\phi^T S_b \phi|}{|\phi^T S_w \phi|} \tag{28}$$

The ratio given by above equation is known as Fisher Criterion and it attempts to maximize the variance of the class means and minimize the variance of the individual classes. The projection matrix ϕ_{LDA} can be obtained by solving the generalized eigenvalue problem [64],

$$S_b \phi_{LDA} = S_w \phi_{LDA} \Lambda, \tag{29}$$

or

$$S_b\phi_{LDA} - S_w\phi_{LDA}\Lambda = 0 \tag{30}$$

Multiplying by inverse of S_w, we get

$$S_w{}^{-1}S_b\phi_{LDA} - S_w{}^{-1}S_w\phi_{LDA}\Lambda = 0 \tag{31}$$

that is,

$$S_w{}^{-1}S_b\phi_{LDA} - \phi_{LDA}\Lambda = 0 \tag{32}$$

$$S_w{}^{-1}S_b\phi_{LDA} = \phi_{LDA}\Lambda \tag{33}$$

Thus, the Fisher criterion is maximized when the projection matrix ϕ_{LDA} is composed of the eigenvectors of $S_w{}^{-1}S_b$. There will be at most $C-1$ eigenvectors with non-zero corresponding eigenvalues since there are at most C points to estimate S_b. Once, the projection is found, all data points can be projected to the new axis system.

Note that we could have used LDA in Chapter 5 for dimensionality reduction. However, our primary goal in Chapter 5 was to test the motion texture features without projecting them into a class discriminatory feature subspace such as the one obtained by LDA. We do this to compare with other methods such as BoW and A-BoW. In addition, we also wanted to test the effect of including sequential information on classification accuracy.

Using broadly categorized training data, we use LDA to map the n-dimensional motion texture features to $C-1$ dimensions ($\{f_1, f_2, ..., f_n\} \mapsto \{X_1, X_2, ...X_{C-1}\}$). This gives us a skill discriminating $(C-1)$-dimensional feature subspace that we use to predict the skill score of a test sample.

6.2 Linear regression

A general linear regression model to represent the relationship between a continuous response y and a continuous or categorical predictor x can be represented as

$$y = \beta_1 f_1(x) + \beta_2 f_2(x) + ... + \beta_L f_L(x) + \epsilon \tag{34}$$

The response y is modeled as a linear combination of functions of the predictor, plus a random error ϵ. The expressions $f_j(x)(j = 1, ..., L)$ are the terms of the model and the $j(j = 1, ..., L)$ are the coefficients. Given n independent observations $(x_1, y_1), ..., (x_n, y_n)$ of the predictor x and the response y, the linear regression model becomes an $n \times L$ system of equations:

$$\begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_n \end{bmatrix} = \begin{bmatrix} f_1(x_1) & ... & f_L(x_1) \\ . & . & . \\ . & . & . \\ . & . & . \\ f_1(x_n) & ... & f_L(x_n) \end{bmatrix} \begin{bmatrix} \beta_1 \\ . \\ . \\ . \\ \beta_L \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ . \\ . \\ . \\ \epsilon_n \end{bmatrix} \tag{35}$$

Ignoring the unknown error ϵ, above equation can be written as $\mathbf{y} = \mathbf{X}\beta$ and can be solved by least square estimation. The best possible estimate of β, $\hat{\beta}$ is defined as the quantity that minimizes the L_2 norm of the error:

$$\|y - \mathbf{X}\hat{\beta}\| = \min_{\beta} \|y - \mathbf{X}\beta\| \tag{36}$$

To compute the least squares estimate, we calculate the partial derivatives of $\|\mathbf{y} - \mathbf{X}\beta\|^2$ by β and equate it to zero giving the following equation:

$$(-2)\mathbf{X^T}(\mathbf{y} - \mathbf{X}\hat{\beta}) = 0 \tag{37}$$

The above equation can also be written as $\mathbf{X^T X}\hat{\beta} = \mathbf{X^T y}$ giving $\hat{\beta} = (\mathbf{X^T X})^{-1}\mathbf{X^T y}$. Once $\hat{\beta}$ is computed, the model can be evaluated at the predictor data to compute the predicted response $\hat{y_t}$ as $\hat{y_t} = X_t\hat{\beta}$ or

$$\hat{y_t} = X_t(\mathbf{X^T X})^{-1}\mathbf{X^T y} \tag{38}$$

The predictor variable can be used in different forms in the linear regression. For instance, polynomial terms such as $f_1(x)^2$ (for curvature) and product terms such as $f_1(x)f_2(x)x_2$ (for interaction) may be used. The following equation gives a linear

regression equation with interaction between two predictor variables:

$$y = \beta_1 f_1(x) + \beta_2 f_2(x) + \beta_3 f_1(x) f_2(x) + \epsilon \tag{39}$$

We obtain a linear regression model (with interaction) using the $C-1$ dimensions in the reduced LDA feature space. For $C = 3$ coarse categories, the linear regression function with interaction is given by,

$$\hat{y} = aX_1 + bX_2 + cX_1X_2 + d, \tag{40}$$

To predict the skill score, the n-dimensional test feature data (from GLCM or LBP-LC) is first projected to the LDA space learnt during training giving $C-1$ dimensional feature vector. The reduced test features $\{X_{1t}, X_{2t}, ...X_{(C-1)t}\}$ are then used to predict the skill score $\hat{y_t}$ of the test sample using the regression function obtained during the training.

6.3 Experimental evaluation

To evaluate the efficacy of our framework for skill score prediction, we use the Normalized Root Mean Square Error (NRMSE), given by, $\sqrt{\frac{\sum(y_n - \hat{y_n})^2}{\sum(y_n)^2}}$ where y_n is the ground truth skill score and $\hat{y_n}$ is the predicted skill score of the n-th sample. We also compute the Pearson correlation coefficient R and the corresponding p value between the true and predicted scores to test whether the true and predicted scores are correlated in a statistically significant manner.

Surgical skill score prediction: First, we process the surgery video data to obtain motion features as explained in Chapter 5 (Section 5.1.1). Then, we use the motion features to compute frame kernel matrices as described in Chapter 5 (Section 5.1.3). After computing frame kernel matrices, we extract GLCM and LBP-LC features at different granularity, i.e., by varying number of gray levels N_g for GLCM ($2^3 - 2^8$) and computing LBP-LC features at different neighborhood sizes (8, 10, 12) and radii (2, 4, 8).

We test our framework for predicting OSATS scores in a Leave-One-Out Cross Validation (LOOCV) scheme. To reduce the feature dimensions using LDA, we group the participants in training data into three coarse skill levels for each of the OSATS criteria: low (OSATS score ≤ 2), intermediate ($2 <$ OSATS score ≤ 3.5) and high ($3.5 <$ OSATS score ≤ 5). The test feature data (from GLCM or LBP-LC) is first projected to the LDA space learnt during training. The reduced test features are then used to predict the score using the regression function obtained during the training as explained in Section 7.2. Figure 29 and Figure 30 show single instance prediction for different OSATS criteria in LOOCV scheme along with broadly categorized training data. The encircled sample in Figure 29 and 30 is the test sample in the LOOCV scheme and shows the true and predicted OSATS score in the LDA space. The remaining instances in each plot are the training samples plotted in the LDA space. The color map is used to show the predicted OSATS value at each combination of LDA components. Green diamonds represent the experts ($3.5 <$ OSATS score ≤ 5), cyan squares represent the intermediate ($2 <$ OSATS score ≤ 3.5) and red circles (OSATS score ≤ 2) represents the novices in the LDA space.

Table 9 shows the NRMSE and correlation coefficient R between the ground truth and the predicted OSATS criteria using LBP-LC at different texture granularity. For instance, multi-scale LBP-LC features computed at neighborhood sizes 12, 10 and 8 with radii 2, 4 and, 8, can be represented as $N_{12}(r_2)N_{10}(r_4)N_8(r_8)$. Multi-scale LBP-LC features with different radii and neighborhood sizes resulted in statistically significant correlation (p-value <0.01) between the true and predicted scores. In Table 9 and 10, "**" refers to p value < 0.01, "*" refers to p value < 0.05, $\mathrm{N}_i(\mathrm{r}_j)$ represents the LPB-LC feature evaluated for neighborhood size i around the radius j. For example, multi-scale LBP-LC features computed at neighborhood sizes 12, 10 and 8 with radii 2, 4 and, 8, can be represented as $N_{12}(r_2)N_{10}(r_4)N_8(r_8)$. Multi-scale LBP-LC features with different radii and neighborhood sizes resulted in statistically

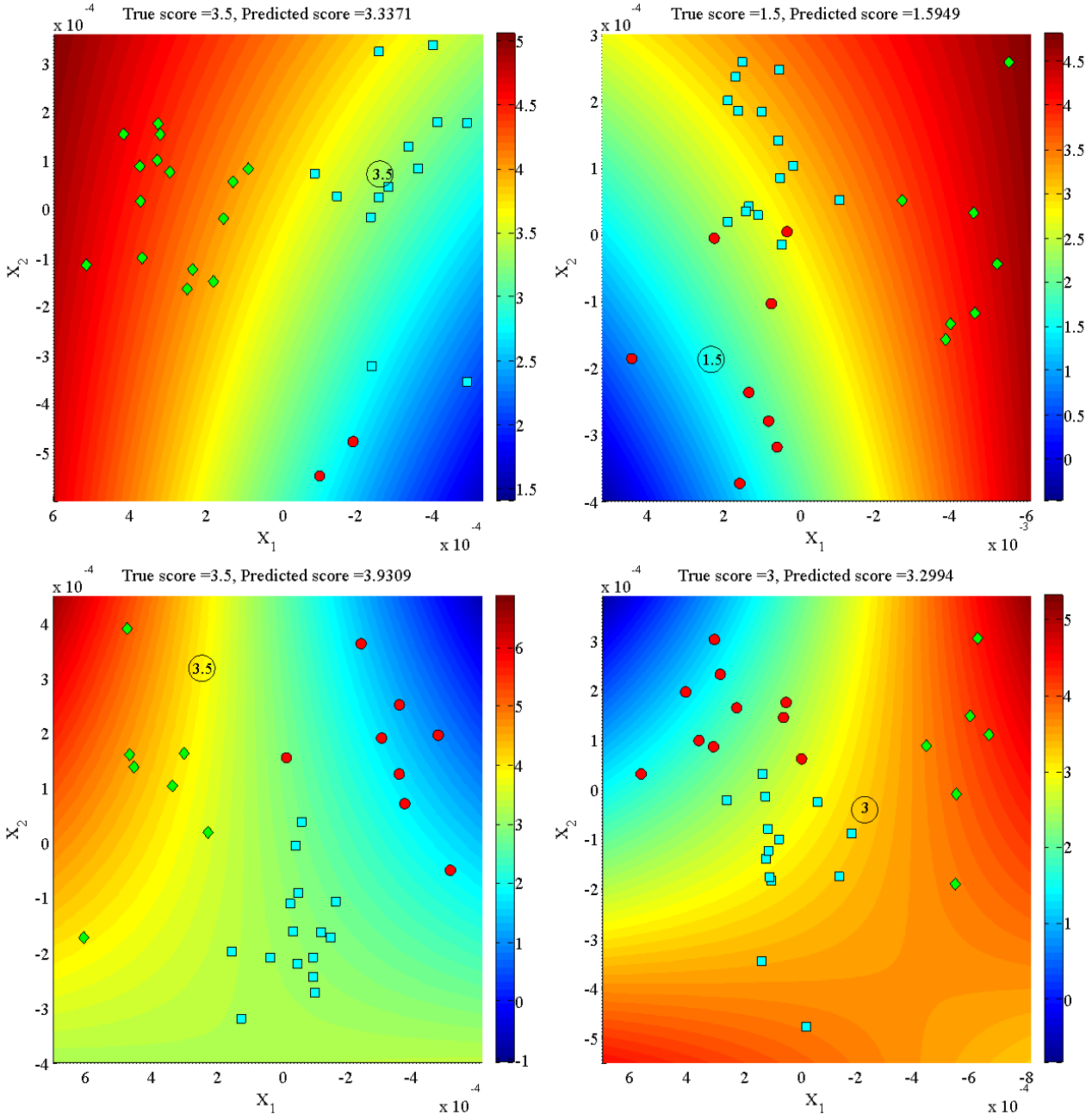


Figure 29: Single instance prediction for OSATS criteria in LOOCV scheme. Top left: respect for tissue, Top right: time and motion, Bottom left: instrument handling, bottom right: suture handling. Note the separation of experts (green diamonds), intermediates (blue squares) and novices (red circles) in the LDA feature space. X_1 and X_2 are the two dimensions in the reduced LDA space. The color map shows the predicted OSATS score using linear regression function at each combination of X_1, X_2.

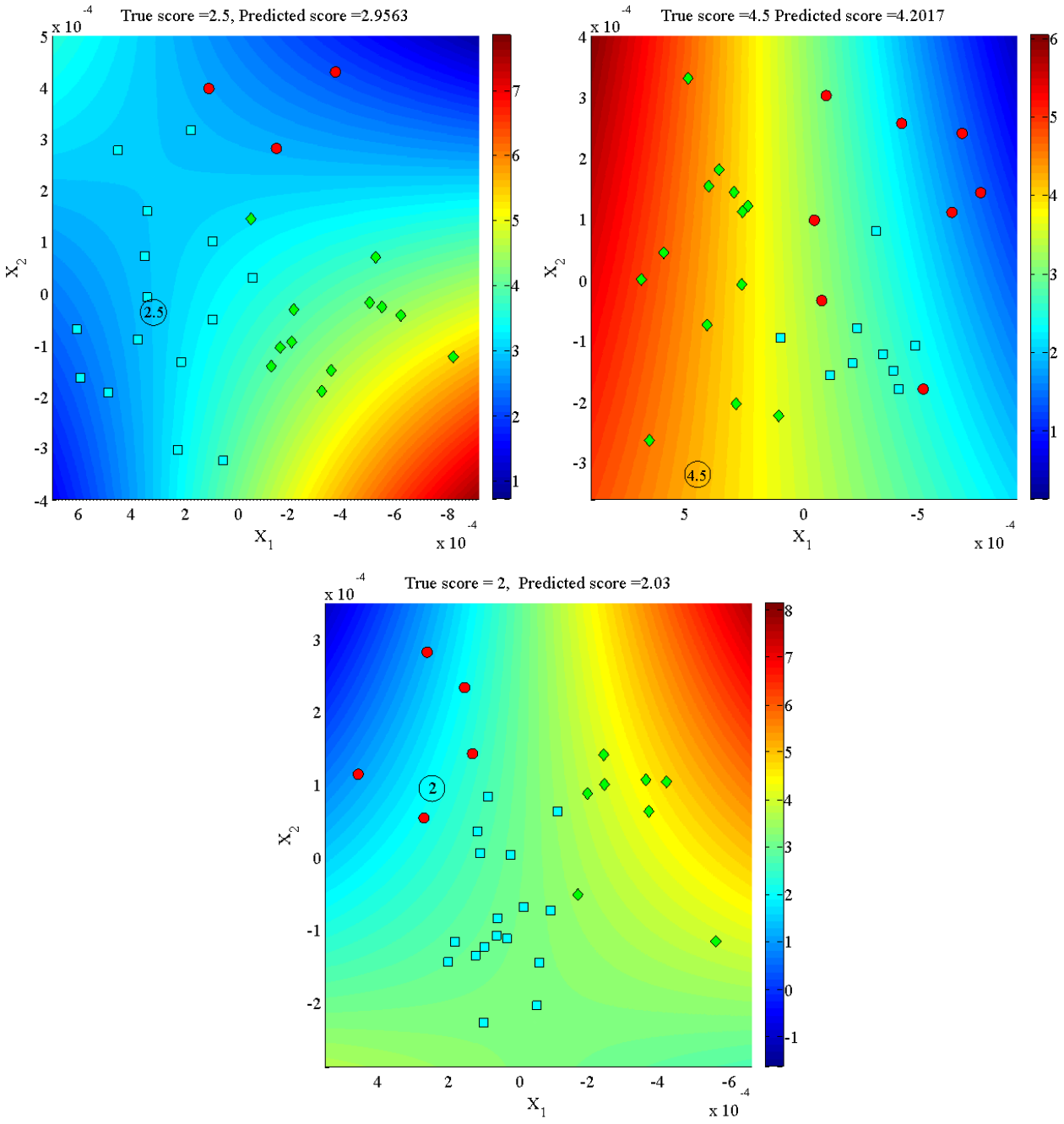


Figure 30: Single instance prediction for OSATS criteria in LOOCV scheme. Top left: flow of operation, Top right: knowledge of procedure, Bottom: overall performance.

significant correlation (p-value <0.01) between the true and predicted scores.

We also achieved statistically significant correlation with GLCM features (Table 10) for several OSATS criteria, however; overall better performance was achieved

Table 9: OSATS prediction (LBP-LC features)

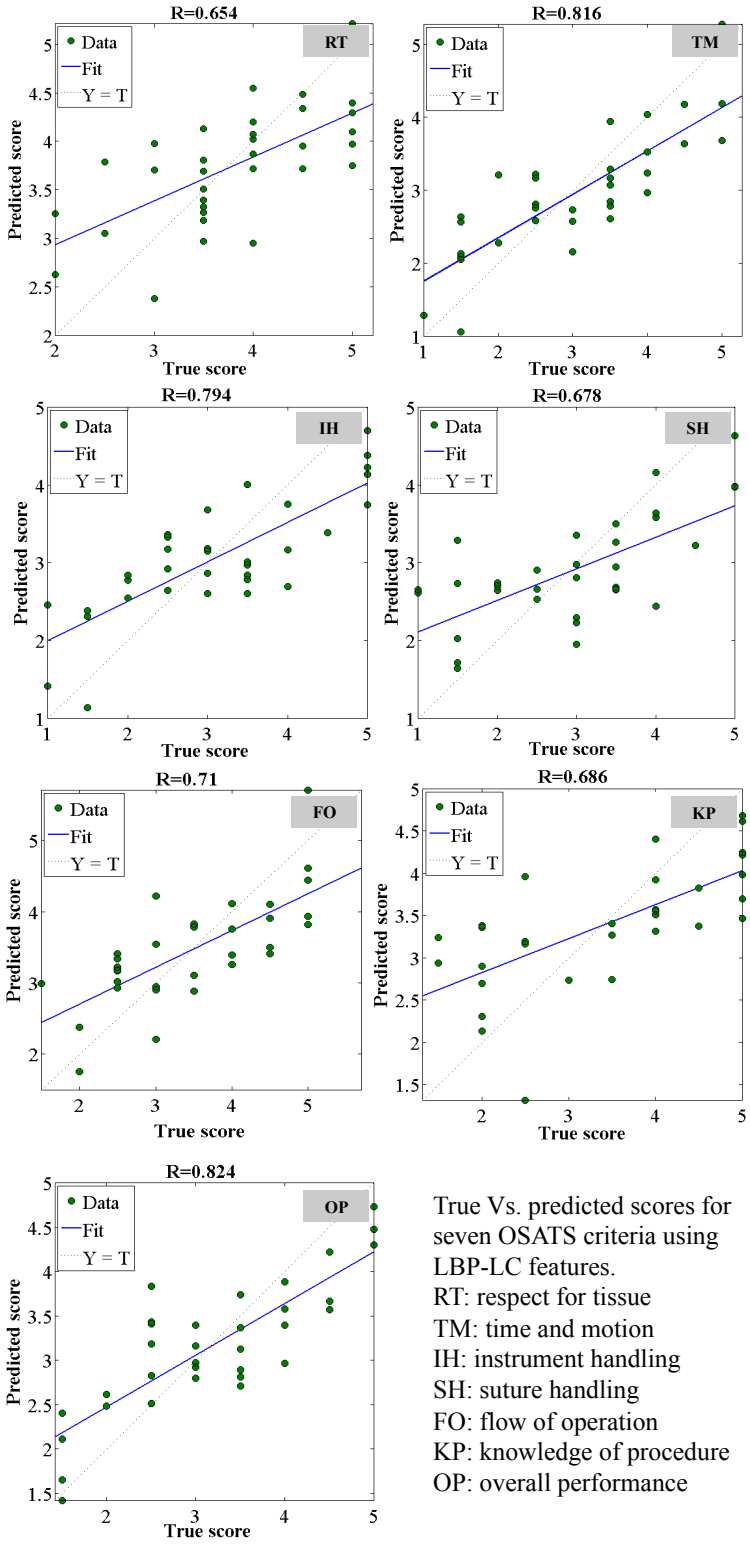
Criteria	Texture feature	NRMSE	R
Respect for Tissue	$\{N_{12}(r_2)\ N_{10}(r_4)\ N_8(r_8)\}$	0.16	0.65^{**}
Time and Motion	$\{N_8(r_2)\ N_8(r_4)\ N_8(r_8)\}$	0.20	0.81^{**}
Instrument Handling	$\{N_8(r_2)\ N_8(r_4)\ N_8(r_8)\}$	0.22	0.79^{**}
Suture Handling	$\{N_{12}(r_2)\ N_{10}(r_4)\ N_8(r_8)\}$	0.26	0.67^{**}
Flow of Operation	$\{N_{12}(r_2)\ N_{10}(r_4)\ N_8(r_8)\}$	0.19	0.71^{**}
Knowledge of Procedure	$\{N_8(r_2)\ N_8(r_4)\ N_8(r_8)\}$	0.24	0.68^{**}
Overall Performance	$\{N_8(r_2)\ N_8(r_4)\ N_8(r_8)\ N_{10}(r_2)\}$	0.17	0.82^{**}

with LBP-LC features. In Table 10, N_g refers to the number of gray levels used to compute the GLCM. Figure 31 shows the true versus predicted scores for seven OSATS criteria using LBP-LC features. With our technique, we are able to predict skill scores for all seven OSATS criteria for a diverse group of participants using two different texture analysis methods (GLCM and LBP-LC). This demonstrates our general concept of skill encoding via texture analysis of motion data encoded into frame kernel matrices.

6.4 Summary

The results in this chapter clearly indicate that motion texture analysis can be used for both relative and absolute skill assessment. Using simple feature dimension reduction techniques such as LDA, and linear regression, we can predict the skill scores effectively for different OSATS criteria.

So far, we have discussed skill assessment using motion data from the whole video (holistic time series analysis) and using all the moving entities (all STIPs). However, for dexterity analysis, we need to isolate motion data from left and right hands. In addition, we can use other data modalities such as depth and acceleration data. In next chapter, we use GT-Emory data set to accomplish dexterity analysis and to compare different feature types for relative skill assessment.



True Vs. predicted scores for seven OSATS criteria using LBP-LC features.
RT: respect for tissue
TM: time and motion
IH: instrument handling
SH: suture handling
FO: flow of operation
KP: knowledge of procedure
OP: overall performance

Figure 31: Surgery data: True vs. predicted scores for seven OSATS criteria.

Table 10: OSATS prediction (GLCM features)

Criteria	Texture feature	NRMSE	R
Respect for Tissue	$N_g = 64$	0.26	0.45^{**}
Time and Motion	$N_g = 128$	0.34	0.56^{**}
Instrument Handling	$N_g = 8$	0.30	0.56^{**}
Suture Handling	$N_g = 128$	0.39	0.43^{*}
Flow of Operation	$N_g = 128$	0.36	0.33
Knowledge of Procedure	$N_g = 128$	0.49	0.45^{**}
Overall Performance	$N_g = 64$	0.31	0.52^{**}

CHAPTER VII

HAND MOTION AND DEXTERITY ANALYSIS

Summary *Motion features are extracted from right and left hand along with corresponding depth and acceleration data. Comparative analysis of different features and hand locations is performed via MT and SMT techniques. Dexterity analysis of hand motion data helps provide expertise rating for different time segments.*

Relative and absolute skill assessments provide the overall competency for different OSATS criteria. However, in medical training, an important issue is providing dexterity feedback to the trainees. Dexterity feedback may be given in several different forms. For example, an expert faculty could provide dexterity feedback by observing the hand movements of the trainees. They can also tell the trainee if they are using the instruments properly or not. The hand (particularly the wrist) movements are very important in surgery. For example, smooth rotation of the wrist as the needle passes through tissues, is considered an important skill [18]. In addition, proper usage and handling of the instruments is required to cause minimal tissue damage.

In this chapter, we analyze the motion data collected from specific instruments and hand locations. We use different types of data (RGB videos, depth videos, and acceleration data) to provide dexterity analysis. Motion features from different hand locations are computed to study the surgical dexterity. In Section 7.1, we describe these features in detail. In Section 7.2, we describe our technique for dexterity analysis using SMT and hand location information. In Section 7.3, we present our results to compare the skill classification using different features. We also present our results on dexterity analysis for different time windows in the surgical videos.

7.1 Hand motion features

First, we test our MT and SMT techniques using different types of features from different hand locations. In previous chapters, we presented results using Newcastle data set with motion features such as STIPs along with GLCM and LBP-LC features. In this chapter, we use GT-Emory data set with both RGB and depth videos. Using colored gloves, we are able to track the hand locations in each frame and can use standard blob features to compute the frame kernel matrices. We also use the depth histograms for left and right hands to compute depth based frame kernel matrices. Besides our previous method using STIPs, we also isolate the STIPs at right and left hand locations. Table 11 gives a summary of the features used for comparative analysis. We refer to the dominant hand as right hand (RH) and non-dominant hand as left hand (LH).

7.1.1 Spatio-temporal interest points from right and left hand

We use color based blob tracking to extract the masks for left and right hand pixel locations. We used *cvblob* [41] library for this purpose. Figure 32 shows a sample frame with corresponding left and right hand masks obtained using blob tracking. Note that color thresholding is used at each frame and factors such as varying illumination may result in noisy mask images. We apply morphological operations (image dilation and hole filling) to obtain clean masks. In addition, note that we used finger-less gloves so that the surgeons could perform the tasks without any obstruction. However, the masks obtained do not include the finger region and image dilation helps alleviate this issue by growing the mask boundaries beyond the glove region.

To extract the STIPs belonging to right and left hands, we use the mask boundary pixel coordinates to define a polygon. The points inside the polygon are then detected using standard point-in-polygon tests. We use MATLAB's inbuilt function *inpolygon* for this purpose.

Table 11: Summary of features used for dexterity analysis

Feature	Description	Usage (purpose)
STIPs	Spatio-temporal interest points as described in Chapter 5	Baseline for comparison with other motion features (*e.g.* STIPs extracted from left and right hand (LH and RH) locations
STIPs (RH, LH, both)	Same as above except that STIPs are extracted from left and right hand locations using binary masks obtained via color thresholding	To test whether eliminating noisy (or irrelevant motion features) will improve skill assessment.
Depth (RH, LH, Both)	Normalized 10-bin depth histograms using depth data from left and right hands	We use depth since fine motion dynamics might be captured with depth variations at hand locations.
Blob (LH, RH, Both)	Blob features (*e.g.* eccentricity, circularity, perimeter, area etc.) extracted from binary masks obtained via color thresholding	To compare the standard blob features with depth features described above
Acceleration : Suturing (RH, NH, Both)	Three dimensional acceleration data collected from dominant hand (RH) wrist and needle-holder (NH)	To test the efficacy of using 3D acceleration data from dominant hand wrist and needle-holder for suturing skill assessment
Acceleration: Knot tying (LH, RH, Both)	Same as above but for knot tying task using accelerometers on left and right hands	To test the efficacy of using 3D acceleration data from dominant hand wrist and needle-holder for knot tying skill assessment

Figure 33 shows sample frames with STIPs belonging to LH (left or non-dominant hand) and RH (right or dominant hand) locations. For each frame, we count the number of STIPs belonging to left and right hands to obtain the frame kernel matrices. For each video, we compute three types of frame kernel matrices – from left hand STIPs, right hand STIPs, and using STIPs from both hands. Note that, we do not learn STIP motion classes when using the left and right hand STIPs. When we used all STIP points (as in Chapter 5), we needed to learn the motion classes to approximately cluster the motion information into different moving entities. However,

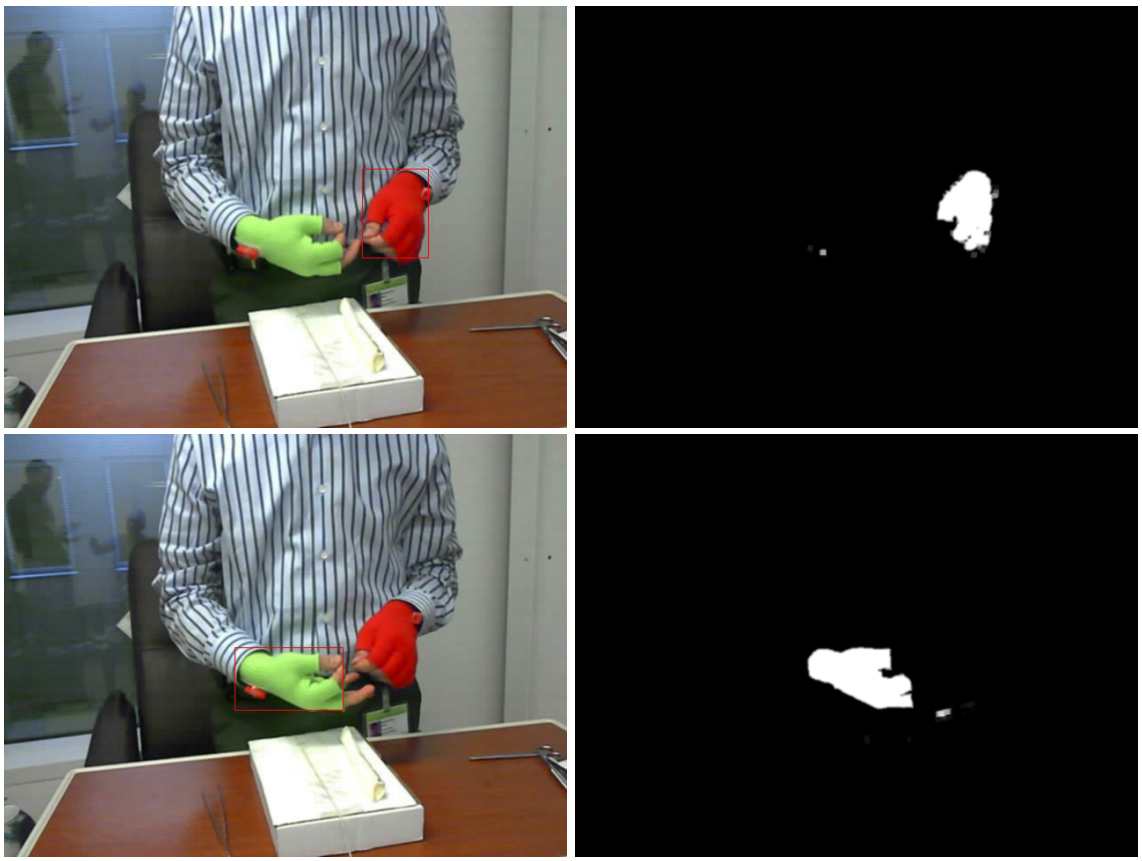


Figure 32: Left and right hand tracking using colored gloves. Top row (left): A sample frame with bounding box for red glove (left hand). Top row (right): Binary mask for the right hand region. Bottom row: Same as top row for right hand.

when we use hand location based STIPs, the motion classes are inherently defined into two (left and right hand).

In GT-Emory data set, we collected data for same time duration and allowed the subjects to keep doing the task repetitively until the acquisition time is over. In such scenario, an expert may be able to accomplish more instances of the task as compared to the novice. These differences get encoded into frame kernel matrices. For example, Figure 34 and 35 show the sample frame kernel matrices for expert and novice surgeons computed using left and right hand STIP features. The relatively fine texture (smaller rectangular patterns) of expert frame kernel matrices as compared to novices (larger rectangular patterns) describes this difference.

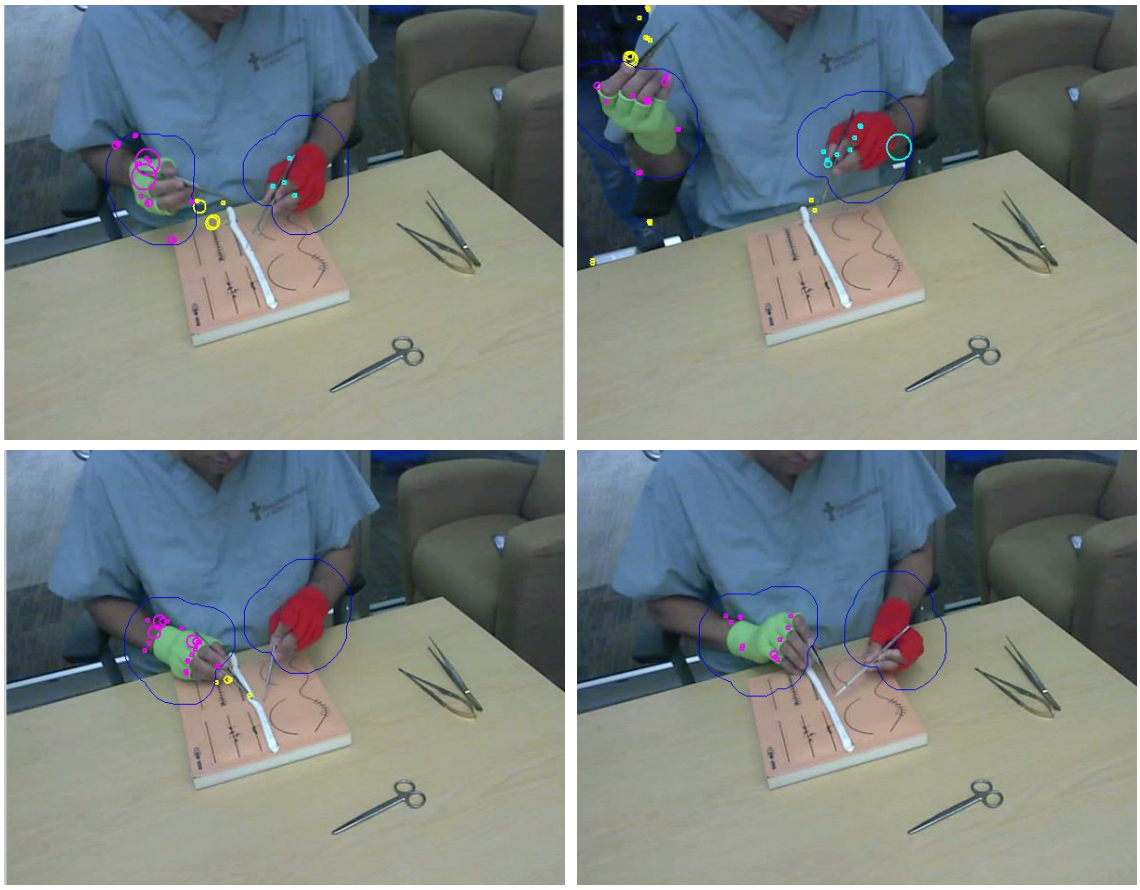


Figure 33: Top row: sample frames with RH (magenta) and LH (cyan) STIPs. Note the irrelevant motion (*e.g.* top right frame has a moving person in the top left region of the frame. Bottom row shows the distinct localization of RH STIPs close to fingers and wrist region.

7.1.2 Blob Features

We used *cvblob* [41] library to extract masks for left and right hands. We compute following standard blob features using MATLAB's *regionprops* function to obtain a 11-element descriptor – (x, y) coordinates of the centroid, area, orientation, perimeter, convex area, solidity, eccentricity, major axis length, minor axis length, and equivalent diameter. Figure 36 and 37 show the sample frame kernel matrices for expert and novice surgeons computed using blob features. Due to varying illumination and background in the videos, blob detection might result in noisy masks resulting in the frame kernel matrices that look almost similar for experts and novices.

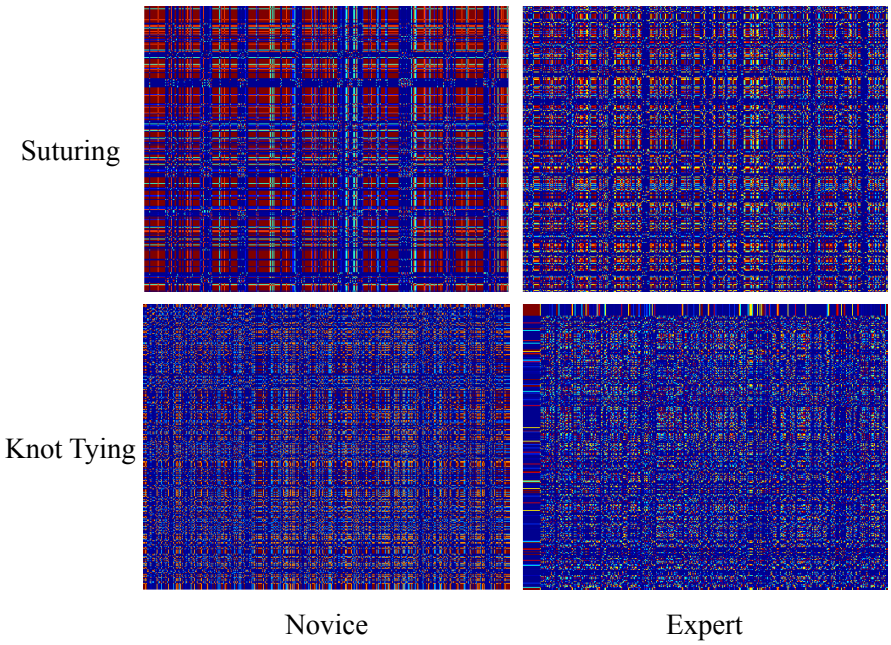


Figure 34: Sample frame kernel matrices computed using left hand STIPs.

7.1.3 Depth features

Using the right and left hand masks, we compute the depth features at hand locations. We use the RGB-aligned depth frames and compute 10-bin normalized (between 0-1) histograms using the non-zero depth values in the mask regions. Figure 38 shows a sample aligned depth frame and corresponding depth histograms at hand locations. Figure 39 and 40 show the sample frame kernel matrices for expert and novice surgeons computed using depth features. Since depth values are used from right and left hand masks obtained using blob detection, the noise in the mask is propagated to the depth features. However, we use the histogram obtained using depth values at the mask locations. This gives more information on motion dynamics as compared to the blob features. Thus, we see clearer patterns in the frame kernel matrices obtained using depth features as compared to the blob features.

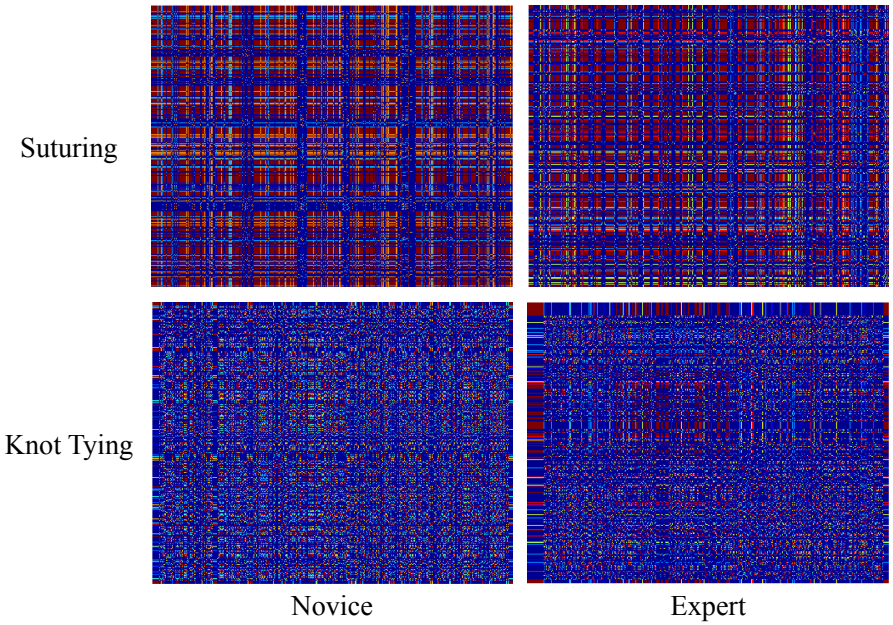


Figure 35: Sample frame kernel matrices computed using right hand STIPs.

7.1.4 Acceleration Features

We used two accelerometers to obtain three-dimensional acceleration data. For the suturing task, one accelerometer is used on the wrist of the subject and another is mounted close to the base of the needle-holder (NH). We align acceleration data with video data using ELAN software [10]. Acceleration data is acquired at 50Hz and video data at 30 frames per second. Thus, for 4000 video frames in a video, we have $4000 \times (50/30)$ samples for acceleration data. Figure 41 and 42 show the sample frame kernel matrices for expert and novice surgeons computed using acceleration data. Acceleration data is not affected by artifacts related to image and video based features such as varying illumination, occlusions *etc.* On the other hand, the vision features, especially the STIPs and depth features, capture motion information from multiple locations. Thus, there is a trade-off between the precision and coverage when we compare the motion information extracted via STIPs and acceleration data.

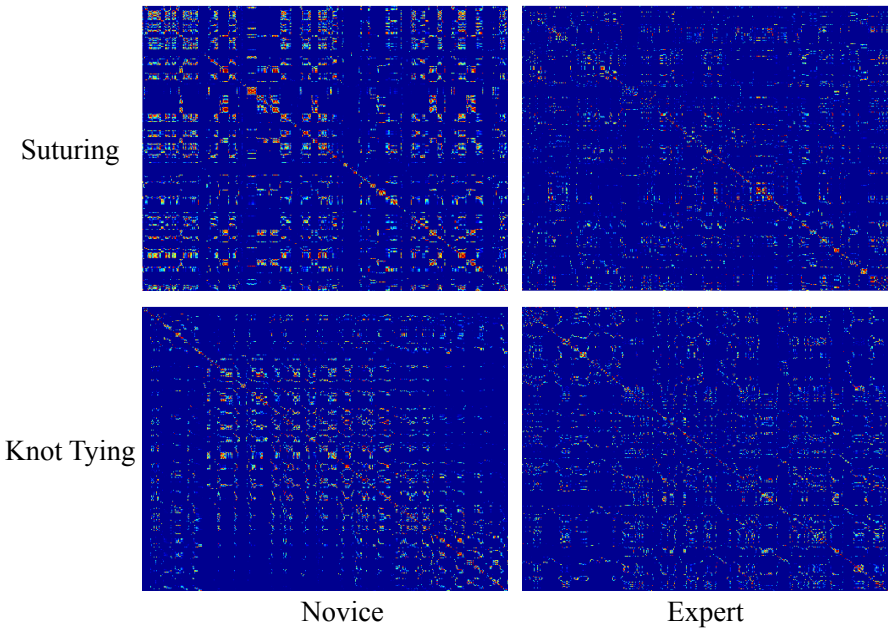


Figure 36: Sample frame kernel matrices computed using left hand blob features.

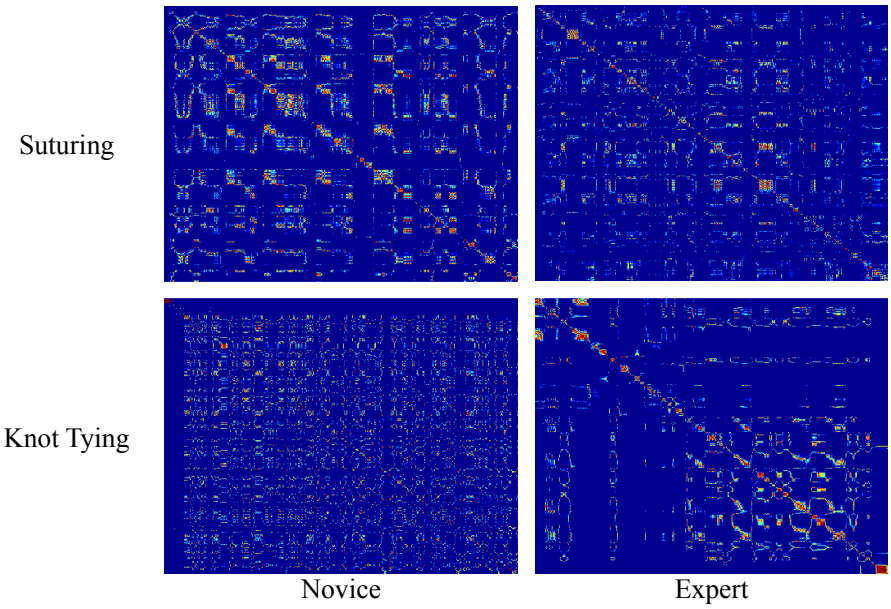


Figure 37: Sample frame kernel matrices computed using right hand blob features.

7.2 Dexterity analysis

We can use the SMT technique and hand location information to provide dexterity feedback. In our analysis so far, we have classified or predicted skill values for the

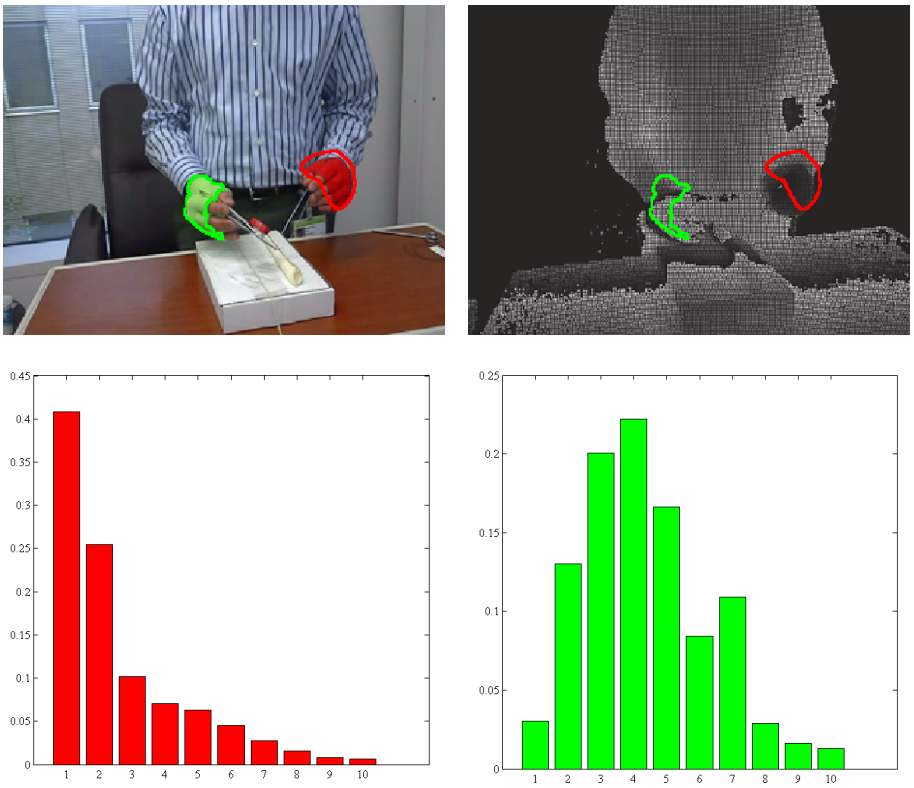


Figure 38: Sample depth histograms computed using the left (red) and right (green) hand depth values.

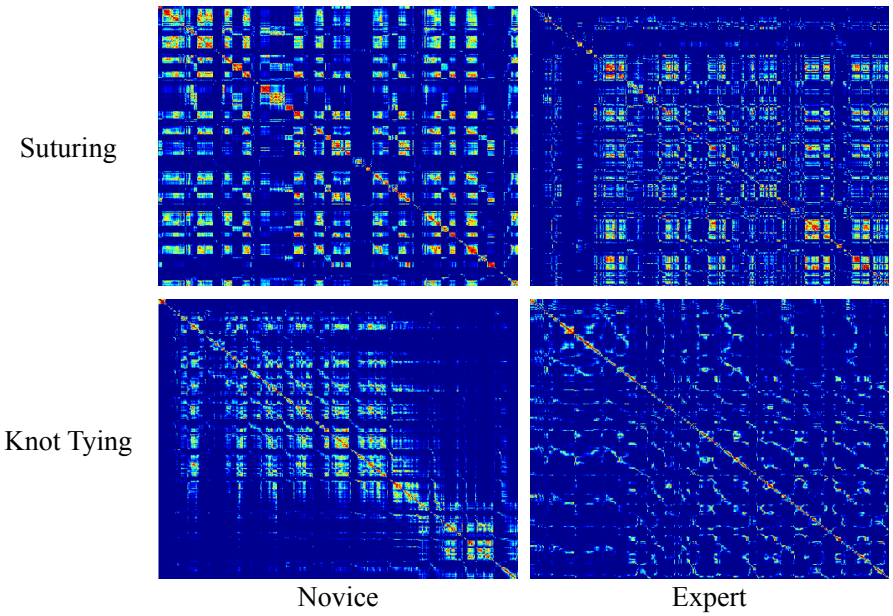


Figure 39: Sample frame kernel matrices computed using left hand depth features.

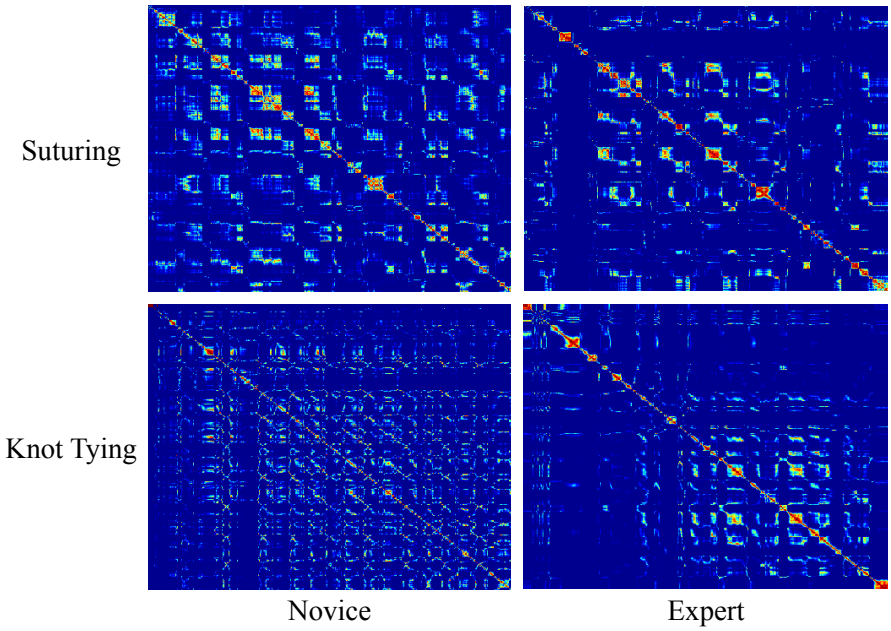


Figure 40: Sample frame kernel matrices computed using right hand depth features.

whole video data. However, it will be more informative for the trainees to get feedback on individual hand movements and in different time slots (or surgical phases) so they could improve themselves on those particular phases and also work on the dexterity of specific hand motions.

Since manual segmentation of time series data might involve human bias and surgical gesture vocabularies may not be sufficient due to non-standard gesture definitions, we adopt a data-driven approach. We use the SMT windows as time slots and compute motion texture features for each time window. In a LOOCV scheme, the training videos are processed to compute features from all time windows from all videos. For example, with $W = 10$ windows and twenty videos in the training data, we obtain features for 200 time windows. Similarly, a test video is processed to obtain 10 time windows. Frame kernel matrices are computed for all the videos. For training data windows, the ground truth is skill label of the parent video. We predict the skill score of the test video's time windows using a nearest neighbor classifier with cosine distance metric.

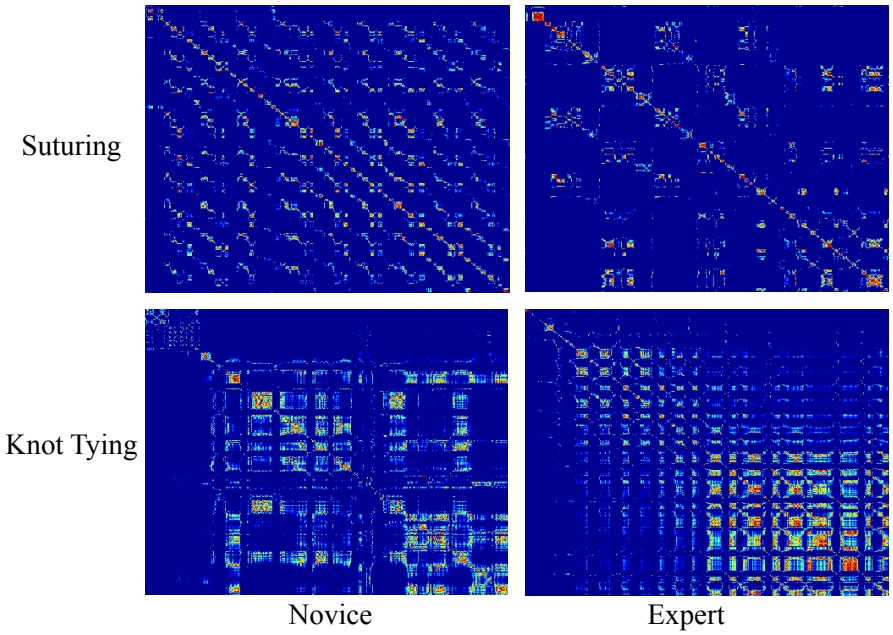


Figure 41: Sample frame kernel matrices computed using left hand acceleration features.

We assess the accuracy of predicted time window labels by computing the mode of the predicted labels and comparing it with the ground truth label of the parent video. If the mode is equal to the ground truth label of the parent video, it implies that for most of the time windows, the predicted label is same as the overall skill label of the whole video. However, using time windows, the trainee can go back and review their performance and skill levels during specific time windows. We report the dexterity based classification accuracy as the percentage of videos for which the mode of the predicted skill labels is same as the overall skill label. Figure 43 shows the flow diagram for dexterity analysis.

7.3 Experimental evaluation

7.3.1 Feature analysis

We performed several experiments to assess the efficacy of different data modalities and feature types. We use GT-Emory data-set and classify trainee surgeons into three skill levels – novice, intermediate, and expert based on overall assessment (ground

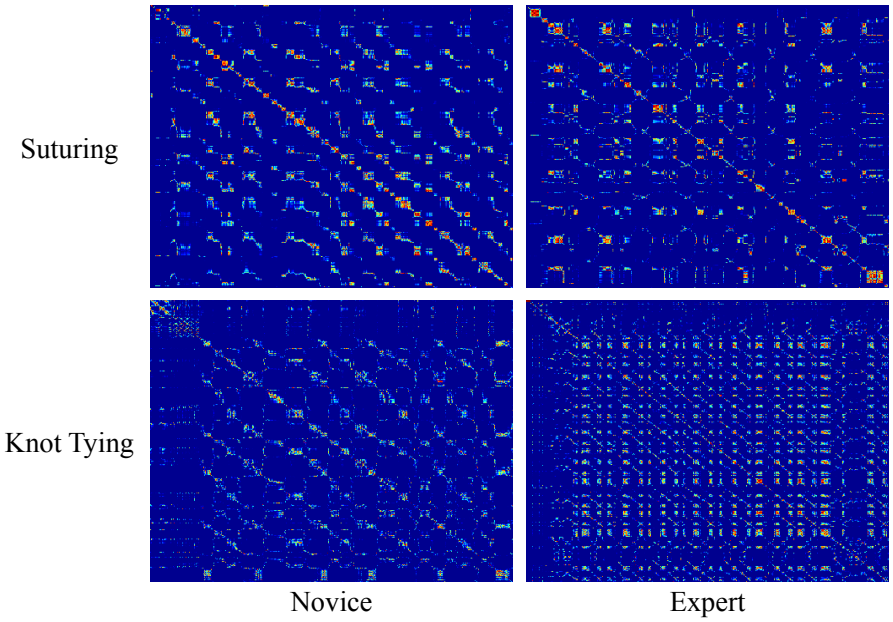


Figure 42: Sample frame kernel matrices computed using right hand acceleration features.

truth) provided by a senior faculty surgeon. We test both MT and SMT techniques and select features using sequential forward feature selection (SFFS) in a LOOCV scheme as explained in Chapter 5. We perform parameter selection using grid search to select number of motion classes (k), GLCM gray levels (N_g), and number of time windows (W for SMT). We also test the significance of features extracted from different hand locations. To test the generalization across different users, we test both the LOSO (leave one sample out) and LOUO (leave one user out) set-up.

7.3.1.1 STIP motion features

Table 12 shows the results for our STIP based technique using all STIPs and STIPs extracted from right and left hand locations. For the suturing task, LH STIPs seem to provide better performance in both LOSO and LOUO set-ups as compared to STIPs from all the locations and right hand. In general, extracting STIPs from left and right hand seems to provide better performance as compared to STIPs from all locations. Also, combining both left and right hand STIPs doesn't improve the performance

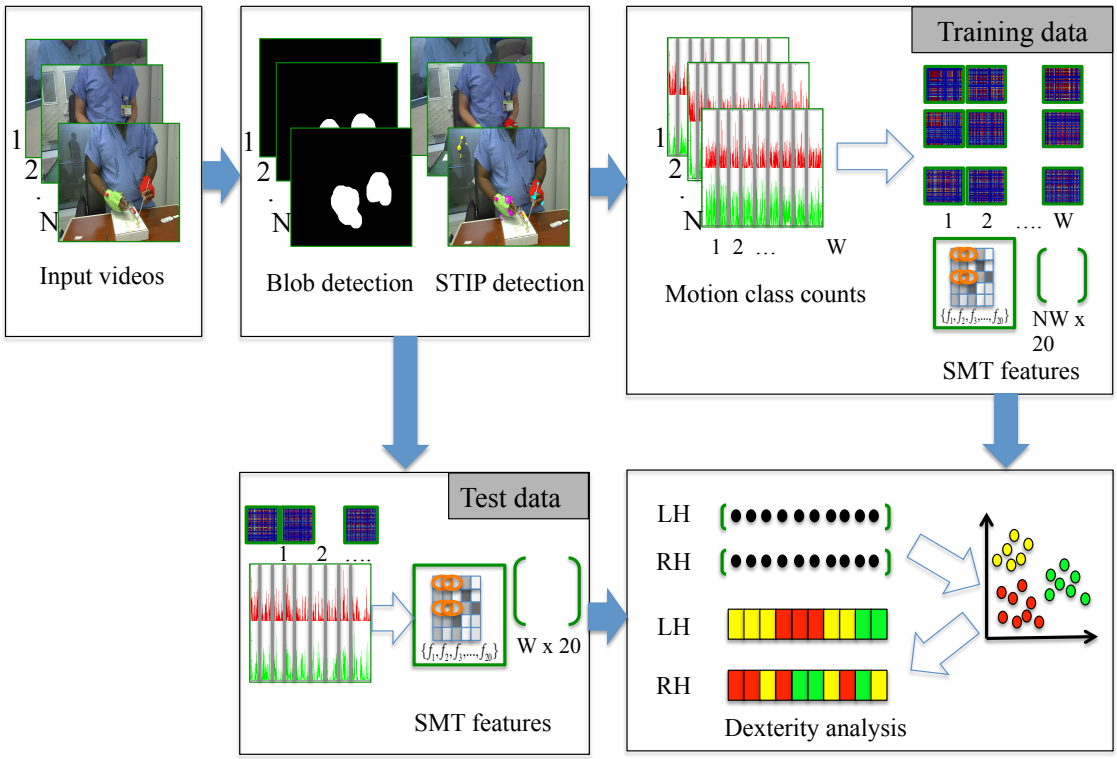


Figure 43: Flow diagram for dexterity analysis.

as compared to the left and right hand STIPs when used individually. For the knot tying task, right hand STIPs provide better performance as compared to all STIPs and left hand STIPs in both LOSO and LOUO set-ups. For suturing one video was corrupt and we used two expert videos to learn motion classes for both suturing and knot tying resulting in thirty-three videos for the suturing task and thirty-four videos for the knot tying task analyses.

Using sequential information in SMT analysis, performance improvement is observed for all cases for both suturing and knot tying tasks as shown in Table 13.

Table 12: Percentage of correctly classified videos – MT with STIP features.

Task	All STIPs	LH STIPs	RH STIPs	RH–LH STIPs
Suturing (LOSO)	87.87% (29/33)	93.93% (31/33)	90.90% (30/33)	84.84% (28/33)
Suturing (LOUO)	81.81% (27/33)	84.84% (28/33)	78.78% (26/33)	72.72% (24/33)
Knot Tying (LOSO)	85.29% (29/34)	76.47% (26/34)	85.29% (29/34)	76.47% (26/34)
Knot Tying (LOUO)	79.41% (27/34)	70.58% (24/34)	85.29% (29/34)	70.58% (24/34)

Table 13: Percentage of correctly classified videos – SMT with STIP features.

Task	All STIPs	LH STIPs	RH STIPs	RH–LH STIPs
Suturing (LOSO)	96.97% (32/33)	93.94% (31/33)	100.00% (33/33)	100.00% (33/33)
Suturing (LOUO)	87.88% (29/33)	87.88% (29/33)	84.85% (28/33)	96.97% (32/33)
Knot Tying (LOSO)	88.24% (30/34)	91.18% (31/34)	91.18% (31/34)	91.18% (31/34)
Knot Tying (LOUO)	85.29% (29/34)	82.35% (28/34)	76.47% (26/34)	79.41% (27/34)

7.3.1.2 Blob features

Table 14 and 15 shows the results using blob features. Using both left and right hand features as compared to individual hand locations, we obtain performance improvement for both MT and SMT analysis. Reasonable performance improvement is obtained using SMT as compared to MT in both LOSO and LOUO set-ups.

Table 14: Percentage of correctly classified videos – MT with blob features.

Task	LH Blob	RH Blob	RH–LH Blob
Suturing (LOSO)	75.75% (25/33)	63.63% (21/33)	78.78% (26/33)
Suturing (LOUO)	69.69% (23/33)	51.51% (17/33)	63.63% (21/33)
Knot Tying (LOSO)	70.58% (24/34)	73.52% (25/34)	82.35% (28/34)
Knot Tying (LOUO)	61.76% (21/34)	64.70% (22/34)	70.58% (24/34)

Table 15: Percentage of correctly classified videos – SMT with blob features.

Task	LH Blob	RH Blob	RH–LH Blob
Suturing (LOSO)	72.73% (24/33)	75.76% (25/33)	78.79% (26/33)
Suturing (LOUO)	69.70% (23/33)	69.70% (23/33)	75.76% (25/33)
Knot Tying (LOSO)	76.47% (26/34)	70.59% (24/34)	76.47% (26/34)
Knot Tying (LOUO)	64.71% (22/34)	64.71% (22/34)	70.59% (24/34)

Table 16: Percentage of correctly classified videos – MT with depth features.

Task	LH Depth	RH Depth	RH–LH Depth
Suturing (LOSO)	69.69% (23/33)	75.75% (25/33)	93.93% (31/33)
Suturing (LOUO)	60.60% (20/33)	60.60% (20/33)	81.81% (27/33)
Knot Tying (LOSO)	64.70% (22/34)	79.41% (27/34)	79.41% (27/34)
Knot Tying (LOUO)	58.82% (20/34)	61.76% (21/34)	58.82% (20/34)

7.3.1.3 Depth features

Table 16 and 17 show the results with right and left hand depth features. With MT analysis, better performance is achieved by including right hand depth features for both the suturing and knot tying tasks. With SMT analysis, classification accuracy improved for both the tasks and for both LOSO and LOUO set-ups. Interestingly, the performance of right hand depth features improves substantially with sequential information in SMT analysis for both suturing and knot tying tasks. Thus, depth features from the dominant hand seem to capture skill relevant information.

7.3.1.4 Acceleration features

Table 18 shows the results with MT analysis. Both right and left hand acceleration data provides reasonable performance for the knot-tying task. For the suturing task, acceleration data from right hand provides better performance as compared to the

Table 17: Percentage of correctly classified videos – SMT with depth features.

Task	LH Depth	RH Depth	RH–LH Depth
Suturing (LOSO)	75.76% (25/33)	87.88% (29/33)	81.82% (27/33)
Suturing (LOUO)	69.70% (23/33)	78.79% (26/33)	63.64% (21/33)
Knot Tying (LOSO)	67.65% (23/34)	79.41% (27/34)	70.59% (24/34)
Knot Tying (LOUO)	64.71% (22/34)	70.59% (24/34)	61.76% (21/34)

Table 18: Percentage of correctly classified videos – MT with acceleration features.

Task	NH (or LH)	Wrist (or RH)	Both
Suturing (LOSO)	60.60% (20/33)	72.72% (24/33)	69.69% (23/33)
Suturing (LOUO)	54.54% (18/33)	66.66% (22/33)	66.66% (22/33)
Knot Tying (LOSO)	82.75% (24/29)	72.41% (21/29)	79.31% (23/29)
Knot Tying (LOUO)	68.96% (20/29)	68.96% (20/29)	62.06% (18/29)

left hand and combination of right and left hand features. Table 19 shows the results with SMT approach using acceleration features. As seen previously for other feature types, SMT gives better performance especially using acceleration data from both the accelerometers. Due to technical difficulties during data acquisition, acceleration data was not acquired properly for some subjects while performing the knot tying task. The acceleration data from these subjects was not used in analysis resulting in twenty-nine samples for knot tying.

In general, better performance is observed using SMT approach as compared to MT approach. Table 20 shows comparison of best performance obtained by different feature types. For suturing, acceleration features provide better performance than blob features followed by depth and STIP features. For knot tying, blob, depth, and acceleration features provide comparable performance and STIP features perform better than other feature types. Next, we present the dexterity analysis results for

Table 19: Percentage of correctly classified videos – SMT with acceleration features.

Task	NH (or LH)	Wrist (or RH)	Both
Suturing (LOSO)	84.85% (28/33)	81.82% (27/33)	87.88% (29/33)
Suturing (LOUO)	81.82% (27/33)	78.79% (26/33)	87.88% (29/33)
Knot Tying (LOSO)	86.21% (25/29)	86.21% (25/29)	89.66% (26/29)
Knot Tying (LOUO)	79.31% (23/29)	79.31% (23/29)	82.76% (24/29)

Table 20: Comparison of performance with different features.

Task	STIPs	Blob	Depth	Acceleration
Suturing (LOSO)	100%	78.79%	93.93%	87.88%
Suturing (LOUO)	96.97%	75.76%	81.81%	87.88%
Knot Tying (LOSO)	91.18%	82.35%	79.41%	89.66%
Knot Tying (LOUO)	85.29%	70.59%	70.59%	82.76%

different hand motion data.

7.3.2 Dexterity analysis

First, we present the results on dexterity analysis, in terms of the percentage of correctly classified videos based on the classification mode of SMT based time window classification as explained in Section 7.2. These results provide validation for SMT that can be also be used to classify individual time windows in a video into different skill levels. We obtain reasonably good classification accuracy using different feature types. Table 21 and 22 show the results obtained with SMT dexterity analysis using different features from left (or non-dominant) (Table 21) and right (or dominant) (Table 22) hands. Classification accuracy obtained with left hand is lower than that obtained with right hand for all the feature types. This indicates that the dominant hand motion is a better predictor of skill for the suturing task as compared to the

non-dominant hand motion.

Figure 44 and 45 show the ground truth label and the predicted expertise level for a novice surgeon using LOUO set up, $W = 12$ time windows and different feature types. Note the visual correspondence between overall expertise and time window based expertise labeling. Since the subject is a novice, most of the frames correspond to novice level (red). In addition, note that the right hand features result in more frame correspondence with the ground truth. Thus, the dominant hand is more predictive of skill for the suturing task.

Figure 46 and 47 show the predicted expertise level for an expert surgeon using LOUO set up, $W = 12$ time windows, and different feature types. Note the visual correspondence between overall expertise and time window based expertise labeling. Since the subject is an expert, most of the frames correspond to expert level (green). In addition, note that the right hand features result in more frame correspondence with the ground truth. Note that the overall skill level (obtained by mode of window labels) corresponds to the ground truth given by the expert surgeon.

It is interesting to note the information consistency obtained with different feature types. For example, in the initial frames, the skill labels might not be consistent among different feature types since the trainees were asked to shake hands for acceleration synchronization. However, for most of the remaining frames the skill label obtained with different feature types remains the same.

The right hand motion of the novice surgeon shows interesting progression from most of the novice labeled windows to intermediate labeled windows followed by few expert-like windows in the end. This indicates expert like performance after getting practiced on initial frames.

With dexterity analysis, we provide predicted skill labels based on specific hand motion data. Thus, the trainee gets feedback on both the hands, *i.e.* "how well" they have performed with respect to the right hand and left hand. In addition, we use

Table 21: SMT dexterity analysis (left hand).

Task	STIPs	Blob	Depth	Acceleration
Suturing (LOSO)	60.61% (20/33)	66.67% (22/33)	63.64% (21/33)	63.64% (21/33)
Suturing (LOUO)	60.61% (20/33)	60.61% (20/33)	60.61% (20/33)	66.67% (22/33)

Table 22: SMT dexterity analysis (right hand).

Task	STIPs	Blob	Depth	Acceleration
Suturing (LOSO)	75.76% (25/33)	75.76% (25/33)	81.82% (27/33)	81.82% (27/33)
Suturing (LOUO)	81.82% (27/33)	81.82% (27/33)	81.82% (27/33)	81.82% (27/33)

data driven time windowing (SMT), which gives expertise labels for each time window. Time windows are obtained using equal sized binning of motion class frequency counts as explained in Chapter 5. This data driven approach avoids manual definition of time segments or gestures, while still providing expertise level for different time segments.

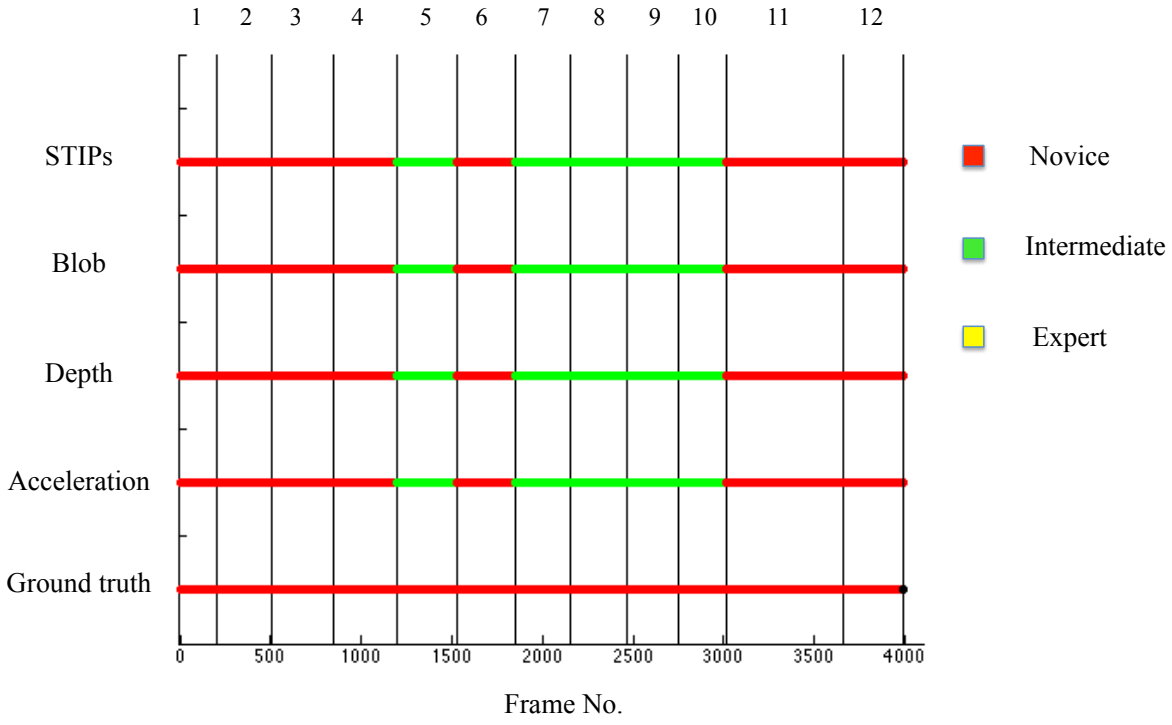


Figure 44: Dexterity analysis for a novice surgeon using left hand features and SMT technique. Note that for most of the time windows, the predicted skill is novice.

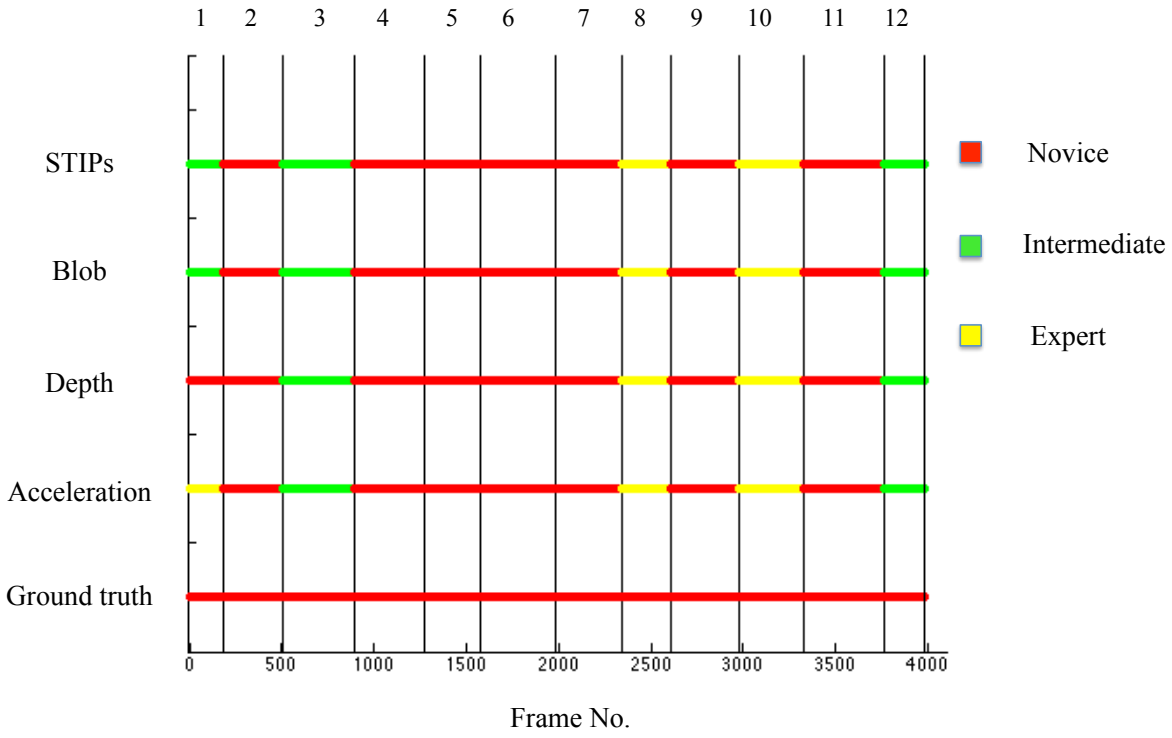


Figure 45: Dexterity analysis for a novice surgeon using right hand features and SMT technique. For most of the windows, the subject performs like a novice although some intermediate and expert like performance is observed in some windows.

7.4 Conclusion

In this chapter, we compared different feature types under different set-ups (LOSO and LOUO) and for different tasks. We used both MT and SMT techniques. In general performance improved with inclusion of sequential information as we observed for Newcastle data set. The GT-Emory data set used in this Chapter was acquired in different settings as compared to Newcastle data set. In addition, a different expert surgeon provided the ground truth. Thus, our approach can be used for data collected in different settings and our results match the expert surgeon's evaluation.

An interesting outcome of SMT approach is skill labeling of individual time windows. Using hand specific motion information and time windowing, we provide skill labels to individual time windows with significant correspondence among different feature types. Thus, for dexterity analysis, we can use the less computationally expensive

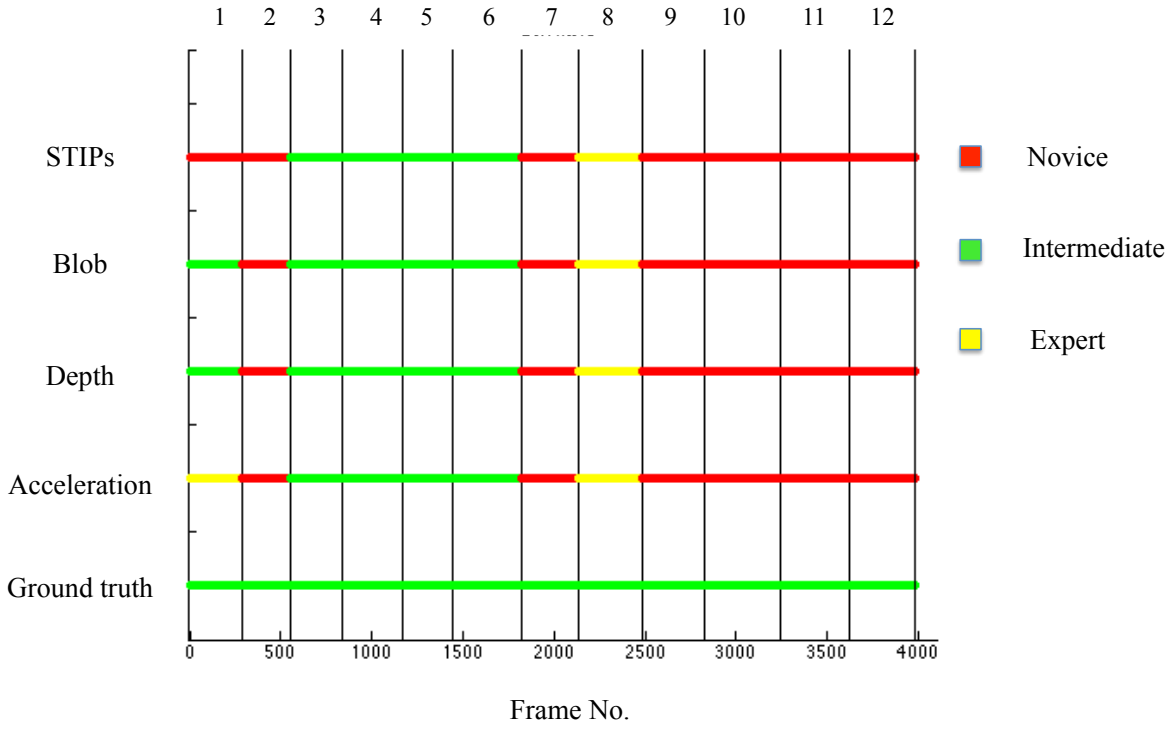


Figure 46: Dexterity analysis for an expert surgeon using left hand features and SMT technique. Left hand motion does not seem to be a good predictor of overall skill.

features such as the three dimensional acceleration data to get equivalent performance as obtained with STIPs, blob and depth features. In addition, the computation of vision feature might get difficult in real surgical scenarios. The acceleration data might be more useful in that situation.

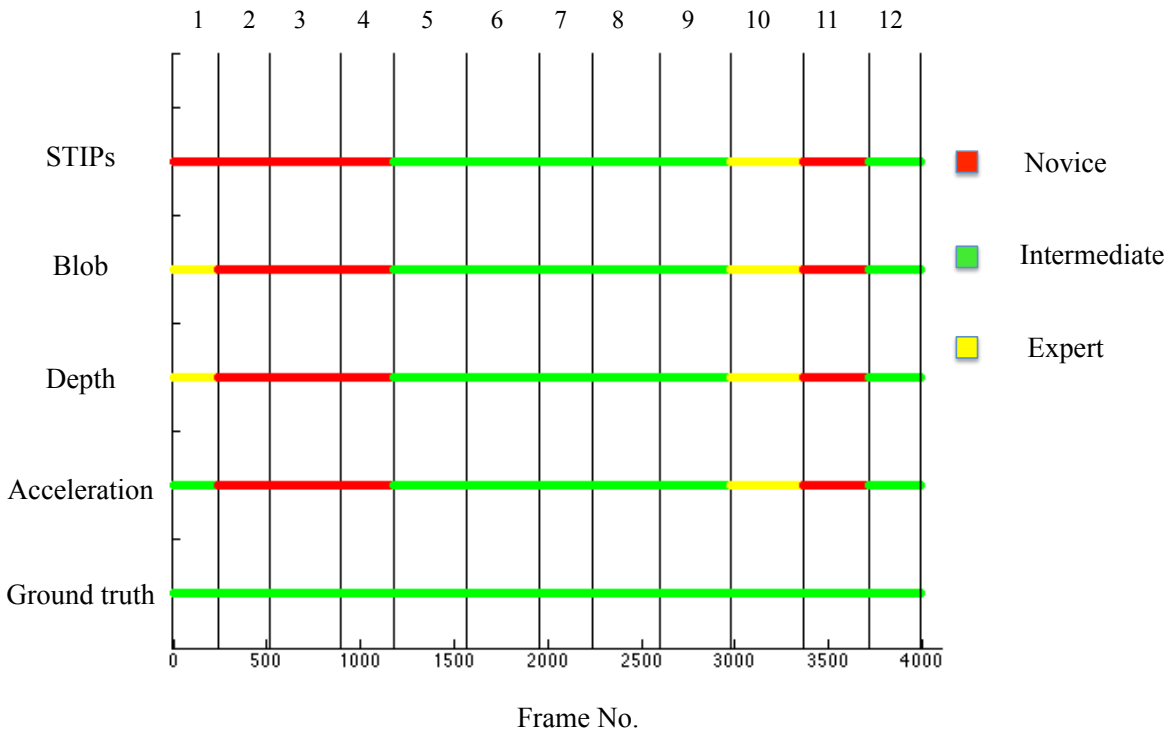


Figure 47: Dexterity analysis for an expert surgeon using right hand features and SMT technique. The subject performed like an expert for most of the time windows. Right hand motion seems to be a better predictor of overall skill as compared to left hand motion.

CHAPTER VIII

CONCLUSIONS AND FUTURE WORK

This thesis explores automated skill assessment in the domain of surgical education and training. Specifically, we consider the basic skills of suturing and knot tying as they are taught to a majority of medical students and require careful manual assessments based on several different criteria. We present motion texture and sequential motion texture frameworks for both relative (classification) and absolute (prediction) skill assessment along with dexterity analysis.

The main contribution of this work is a generalized framework for skill assessment. The framework is built upon low-level motion data such as STIPs, blob features and three-dimensional acceleration data, which can be encoded into frame kernel matrices. The distinct texture patterns in frame kernel matrices correspond to the skill level. Texture analysis can be used to extract skill defining information from frame kernel matrices.

We present motion texture analysis, and sequential motion texture analysis that incorporates sequential information also. Sequential motion texture analysis also provides data driven time segments, which can be assessed individually to provide segment-based skill assessment. Using simple mechanisms to isolate hand motion features such as by using colored gloves and by using data driven sequential motion texture analysis dexterity can be assessed. Using appropriate feature selection methods, skill can be assessed either for a single task or a corpus of tasks.

8.1 Future Directions

There are several directions for future research that can branch out of this thesis. Some of the concepts can be directly extended to develop new ways to measure

surgical skill levels and others can be built upon the framework to obtain even finer measures of skill.

8.1.1 Incorporating other attributes besides motion

In this thesis, we presented skill assessment based on raw motion data. However, other attributes such as surgeon's gaze, hand-eye coordination and other biometrics such as body temperature might also be used for skill assessment. Moreover, progression of skill acquisition can be monitored over a period of time and time-based skill assessment models can be developed.

8.1.2 Real time skill assessment and feedback

In this work, we have presented framework for surgical skill assessment in a retrospective manner by analyzing the video recordings of surgical trainees. Real time system for skill assessment and dexterity feedback can be developed by training the system on more data-sets and parallel computing for real time processing. Computational load can also be reduced by careful feature selection and selecting less computationally expensive features and using several motion sensors. An interesting feedback could be providing a live comparison with expert's motion so that trainees can get real time feedback on their performance.

8.1.3 Extending skill assessment to real surgical procedures

The scope of this thesis is confined to surgical training on simulation models. However, as the trainees achieve proficiency, they start learning complicated surgical procedures that are performed in the operating room. Assessment of surgical procedures that are performed on real patients in an operating room is an important research direction. This might involve challenges such as not being able to track the hands due to occlusions by blood tissue *etc.* Wireless motion sensors such as accelerometers might be suitable for real operating room assessments. In addition, the system

may be extended to assess collaborative surgical environments where each surgeon's motion can be analyzed within the context (or phase of the surgery).

8.1.4 Video information summary

We used the sequential motion texture analysis for providing skill scores in different time windows. This technique can also be used to generate summary of video data. For example, the activity type or other meta-data can be substituted for skill levels and models can be trained using this information. New video data can be labeled for meta-data information using trained models.

8.1.5 Motion based surgical phases

Using hand motion data, we noticed specific configuration of STIPs as shown in Figure 48. The top row in Figure 48 shows two parallel bands of STIPs on the dominant hand (green glove) of the surgeon. On the other hand, both top and bottom left frames show the movement of non-dominant hand (red glove). The bottom right frame shows only the movements of dominant hand. Thus, specific arrangement of moving points can be used to define surgical phases. This will also help finer analysis of motion data by correlating the surgical phases with motion data.

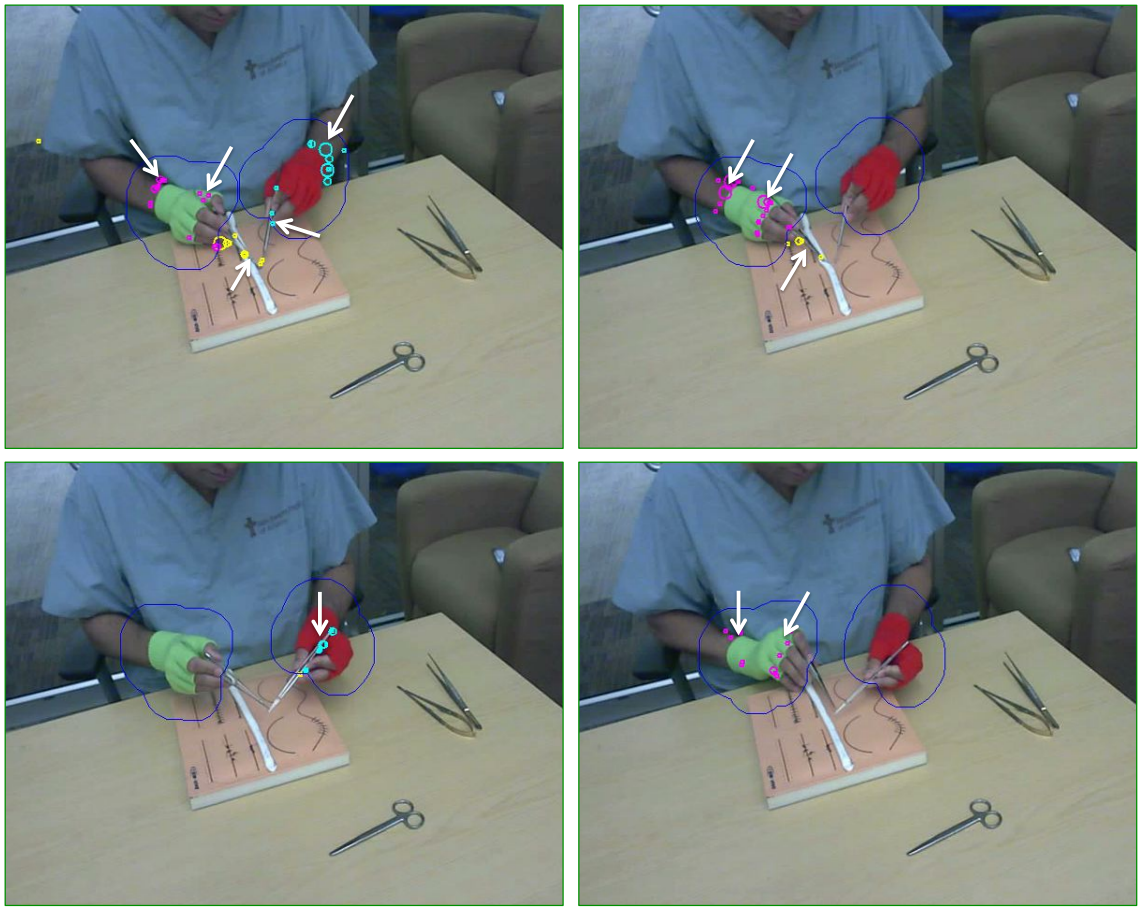


Figure 48: Sample frames showing specific arrangements of STIPs (marked by arrows) on the surgeon's hands and instruments. These specific geometric relations between STIPs can be used to define gestures without manual intervention.

APPENDIX A

BASIC INTRODUCTION TO SUTURING AND OSATS

Summary *Skill assessment of even basic surgical tasks, such as suturing, involves qualitative and sequential motion characteristics. Scoring systems such as OSATS are used in medical schools and teaching hospitals and they provide standard assessment guidelines to evaluate surgical trainees on several different criteria.*

A.1 Need for objective assessment in surgery

In surgical residency programs, learning of surgical skills is an essential part of the training process. The need for formal testing of surgical technical skills has been noticed and studied by various groups. For instance, Martin et al. [37] stated that the introduction of formal testing for specific operative skill could be used to provide constructive feedback that would be of use in resident promotion decisions and could identify deficiencies in the training program.

Advanced surgical procedures are mostly learnt in the operating room under the supervision of expert surgeons. In addition, the basic skills may be learnt in animal laboratory using anesthesia on animals. However, due to moral and ethical issues involved in the use of live animals, it is becoming difficult to justify the use of animals if alternative methods and materials are available [37]. Another option for teaching and testing technical skills is by using the bench models. As compared to patients and live animals, the bench models are lower in costs, have high portability, reuse the materials, and are readily available.

In order to seek reliable and valid methods for surgical skill evaluation, checklists and detailed global rating scales were introduced as far back as in 1971 [29]. Martin

et al. proposed the Objective Structured Assessment of Technical Skills (OSATS) criteria to evaluate surgical skills in late 1990s. The OSATS model involves direct observation of residents performing a variety of structured operative tasks. The OSATS model was developed for both live animals and bench models were used for this and two types of scoring systems were developed – an operation-specific checklist and a detailed global rating scale.

Before discussing the details of the global OSATS criteria, we briefly describe the surgical tasks analyzed in this thesis. This will also provide the background required to understand the details in the OSATS.

A.2 Surgical tasks

In this Section, we briefly provide a description of two basic surgical tasks–suturing and knot tying. Our description here is in context with OSATS and is intended to provide background to understand the work in later chapters. However, for details on specific tasks, the reader is referred to [19, 39].

A surgical suture is used to hold body tissues together after an injury or surgical incision. Figure 49 shows the instruments used for surgical suturing and Figure 50 shows the suturing needle. Suturing needle is a curved needle with main body and swage. The swage is the point where the material joins with the needles. It creates a single, continuous unit of suture and needle. There are three different types of swage. The quality of the swage is critical to the performance of the suture. A high quality needle and the strongest thread becomes futile if the needle detaches during surgery.

The first step in suturing involves mounting the needle with attached suture into a needle holder. Then, the needlepoint is pressed into the tissue and advanced along the trajectory of the needle's curve until it emerges, and pulled through. The trailing thread is then tied into a knot. Sutures should bring together the wound edges, but should not cause indenting or blanching of the skin, since the blood supply may be

impeded with increased risk of infection and scarring [28].

There are two common suture types – *interrupted* and *running*. In interrupted suture, the stitches are not connected to each other. Instead, they are placed and tied off individually. Placing and tying each stitch individually is time-consuming, but this technique keeps the wound together even if one suture fails. The knot crosses the wound perpendicularly. Interrupted sutures allow the surgeon to make adjustments as needed to properly align wound edges as the wound is sutured. Figure 51 (left) shows the interrupted sutures.

In running sutures, the surgeon uses a continuous piece of suture material and works on alternating sides of the opening to pull the edges together to promote healing. Main advantage of the running suture is that it is easy and fast, but the stitch has several disadvantages such as the tendency to let the tissues shift or ripple and erroneous approximation of wound edges compared to the interrupted stitch. Figure 51 (right) shows the running sutures. Compared with running sutures, interrupted sutures are easy to place, have greater tensile strength, and have less potential for causing wound edema and impaired cutaneous circulation.

Several factors determine the choice of running or interrupted sutures such as the wound's location, cosmetic concerns, and the thickness of the tissue or skin. In both running and interrupted sutures, two important sub tasks are *suture placement* and *knot tying*. For suture placement, the needle is allowed to penetrate the skin at a 90° angle to minimize the size of the entry wound. The curved shape of the needle is followed by circular wrist movement allowing the needle to exit perpendicular to the skin surface [19].

Here we briefly describe the commonly used square knot. First, the tip of the needle holder is rotated clockwise around the long end of the suture material for two complete turns. The tip of the needle holder is used to grasp the short end of the suture. The short end of the suture is pulled through the loops of the long end by

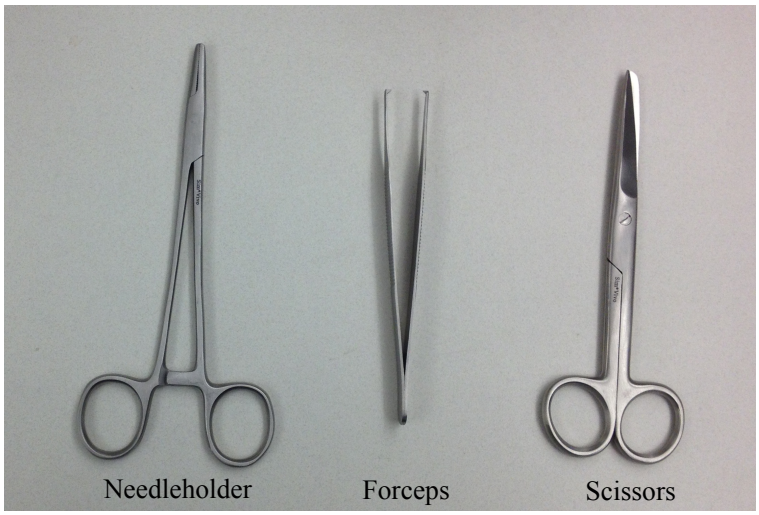


Figure 49: Instruments used in surgical suturing

crossing the hands, such that the two ends of the suture material are situated on opposite sides of the suture line. The needle holder is rotated counterclockwise once around the long end of the suture. The short end is grasped with the needle holder tip, and the short end is pulled through the loop again.

The suturing task requires clear motions performed in a well-defined manner and handling the instruments in an appropriate way to minimize tissue damage. All surgical students practice suturing skills as part of their training. To reduce subjectivity in assessing the trainee's skills, the global rating scales such as Objective Structured Assessment of Technical Skills (OSATS) are used. Next, we provide a brief description of the global OSATS rating scale.

A.3 OSATS

Table 23 shows the global OSATS rating scale [37]. The global OSATS criteria are briefly described below in the context of motion and to provide the foundation for motion texture analysis in this work.

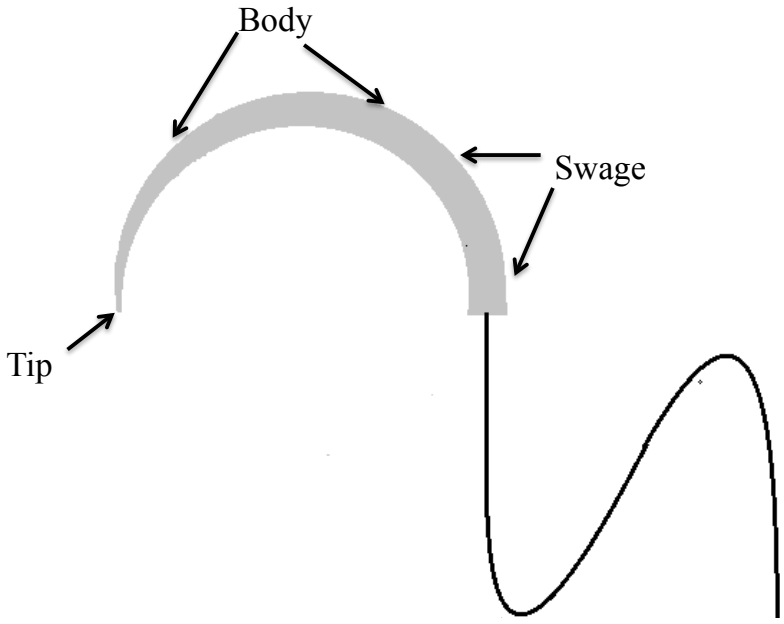


Figure 50: Suturing needle.

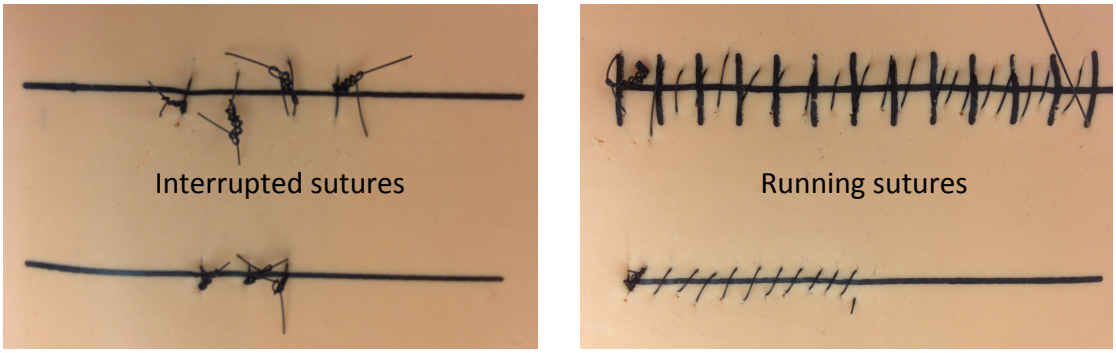


Figure 51: Interrupted and running suture

1. ***Respect for tissue (RT)***: This criterion describes the extent of damage caused to the tissue while performing the procedure. It measures how well the tissue is handled. The damage to the tissue may not be caused in a deterministic sequential manner. Thus, this criterion is mostly *qualitative* and depends more on the motion quality and less on the execution order.

2. ***Time and motion (TM)***: Time and motion relates to efficiency of time while performing the procedure. In the context of motion, this criterion depends on

Table 23: Objective structured assessment of technical skills (OSATS) scale [37].

Criteria	1	2	3	4	5
Respect for tissue	Frequently used unnecessary force on tissue or caused damage by inappropriate use of instruments		Careful handling of tissue but occasionally caused inadvertent damage		Consistently handled tissues appropriately with minimal damage
Time and motion	Many unnecessary moves		Efficient time and motion but some unnecessary moves		Clear economy of movement and maximum efficiency
Instrument handling	Repeatedly makes awkward or tentative moves with instruments through inappropriate use		Competent use of instruments but occasionally appeared stiff or awkward		Fluid movements with instruments and no stiffness or awkwardness
Suture handling	Awkward and unsure with repeated entanglement, poor knot tying and inability to maintain tension		Careful and slow with majority of knots placed correctly with appropriate tension		Excellent suture control with correct placement of knots and correct tension
Flow of operation	Frequently stopped operating and seemed unsure of next move		Demonstrated some forward planning and reasonable progression of procedure		Obviously planned operation with efficiency from one move to another
Knowledge of procedure	Insufficient knowledge, looked unsure and hesitant		Knew all important steps of operation		Demonstrated familiarity with all steps of operation
Overall performance	Very poor		Competent		Clearly superior

unnecessary moves resulting in wasted time. The unnecessary moves might occur in a sequential manner if the trainee performs these moves during specific subtasks, *e.g.* knot tying or they might happen abruptly especially for trainees in very early stages. Thus, this criterion may have both *qualitative and sequential* aspects depending upon the sub-tasks and the trainee's experience.

3. ***Instrument handling***: This criterion pertains to instrument usage. Specifically, the fluidity of motion is examined while using the instruments. It is

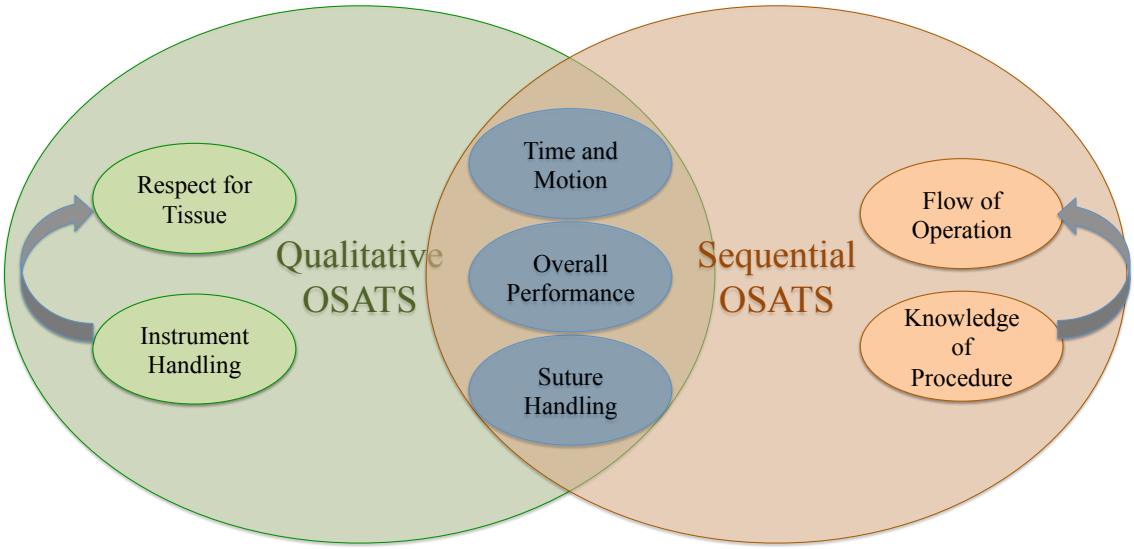


Figure 52: Qualitative and Sequential OSATS criteria.

important to note that this criterion might be related to respect for tissue since tissue damage might be caused by inappropriate usage of instruments. Since instruments are typically used during the whole procedure, this criterion is more *qualitative* but might have some sequential aspects as well.

4. ***Suture handling***: Suture handling predominantly pertains to knot tying. This criterion is mostly *sequential* since knot tying is done sequentially (*e.g.* in interrupted suturing) and within knot tying there are predefined moves.

5. ***Flow of operation***: This criterion is mostly *sequential* since the surgical moves are predefined. However, during early training, the trainees might be unsure of the procedure and might perform steps out of sequence. Note that this criterion may be related to knowledge of procedure since forward planning and flow will depend on knowledge of procedure.

6. ***Knowledge of procedure***: This criterion is *sequential* and depends on the how well the trainee knows the sequence of steps to be performed. This criterion is exemplified by the execution of the procedure and the flow of operation criterion.

7. ***Overall performance***: Overall performance depends on *both sequential and qualitative* motion aspects.

In conclusion, the OSATS criteria are very diverse and it is challenging to design an automated system to evaluate all these criteria within a common framework. Figure 52 shows the categorization of the seven OSATS criteria into sequential and qualitative aspects along with their relation to each other.

REFERENCES

[1] "Axivity 3-axis accelerometer." http://axivity.com/v2/products/WAX3/WAX3-2.1-Datasheet.pdf.

[2] "Creative* interactive gesture camera developer kit." http://click.intel.com/intelsdk/Creative_Interactive_Gesture_Camera_Developer_Kit-P2061.aspx.

[3] "The language archive: Elan software." http://tla.mpi.nl/tools/tla-tools/elan/.

[4] Aggarwal, J. and Ryoo, M. S., "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[5] Ahmidi, N., Ishii, M., Fichtinger, G., Gallia, G., and Hager, G., "An objective and automated method for assessing surgical skill in endoscopic sinus surgery using eye-tracking and tool-motion data," in *International Forum of Allergy & Rhinology*, Wiley Online Library, 2012.

[6] Awad, S., Liscum, K., Aoki, N., Awad, S., and Berger, D., "Does the subjective evaluation of medical student surgical knowledge correlate with written and oral exam performance?," *Journal of Surgical Research*, vol. 104, no. 1, pp. 36–39, 2002.

[7] Balakrishnama, S. and Ganapathiraju, A., "Linear discriminant analysis-a brief tutorial," *Institute for Signal and information Processing*, 1998.

[8] Bettadapura, V., Schindler, G., Plötz, T., and Essa, I., "Augmenting bag-of-words: Data-driven discovery of temporal and structural information for activity recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[9] Blum, T., Feussner, H., and Navab, N., "Modeling and segmentation of surgical workflow from laparoscopic video," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*, pp. 400–407, Springer, 2010.

[10] Brugman, H. and Russel, A., "Annotating multimedia/multi-modal resources with elan," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, pp. 2065–2068, Citeseer, 2004.

[11] Chatelain, P., Padoy, N., and Hager, G., "Surgical gesture modeling using switching linear dynamical systems," *Report, John Hopkins University*, 2011.

[12] Chi, P., Scott, G., and Shyu, C., "A fast protein structure retrieval system using image-based distance matrices and multidimensional index," *International Journal of Software Engineering and Knowledge Engineering*, vol. 15, no. 03, pp. 527–545, 2005.

[13] Clausi, D., "An analysis of co-occurrence texture statistics as a function of grey level quantization," *Canadian Journal of remote sensing*, vol. 28, no. 1, pp. 45–62, 2002.

[14] Cossu, R., "Segmentation by means of textural analysis," *Pixel*, vol. 1, no. 2, pp. 21–24, 1988.

[15] Datta, V., Bann, S., Mandalia, M., and Darzi, A., "The surgical efficiency score: a feasible, reliable, and valid method of skills assessment," *The American journal of surgery*, vol. 192, no. 3, pp. 372–378, 2006.

[16] Dean, C., *Quantitative Description and Automated Classification of Cellular Protein Localization Patterns in Fluorescence Microscope Images of Mammalian Cells*. PhD thesis, Carnegie Mellon University, 1999.

[17] Fischer, I. and Poland, J., "Amplifying the block matrix structure for spectral clustering," in *Proceedings of the 14th annual machine learning conference of Belgium and the Netherlands*, pp. 21–28, Citeseer, 2005.

[18] Fried, G. M. and Feldman, L. S., "Objective assessment of technical performance," *World journal of surgery*, vol. 32, no. 2, pp. 156–160, 2008.

[19] Giddings, F. D., *Surgical Knots and Suturing Techniques third edition*. Giddings Studio Publishing, 2009.

[20] Guo, Y., Zhao, G., and Pietikäinen, M., "Texture classification using a linear configuration model based descriptor.," in *BMVC*, pp. 1–10, 2011.

[21] Hamid, R., Maddi, S., Johnson, A., Bobick, A., Essa, I., and Isbell, C., "A novel sequence representation for unsupervised analysis of human activities," *Artificial Intelligence*, vol. 173, no. 14, pp. 1221–1244, 2009.

[22] Haralick, R., Shanmugam, K., and Dinstein, I., "Textural features for image classification," *Systems, Man and Cybernetics*, no. 6, pp. 610–621, 1973.

[23] Haro, B. B., Zappella, L., and Vidal, R., "Surgical gesture classification from video data," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pp. 34–41, Springer, 2012.

[24] Harris, C. and Stephens, M., "A combined corner and edge detector.," in *Alvey vision conference*, vol. 15, p. 50, Manchester, UK, 1988.

[25] Judkins, T. N., Oleynikov, D., and Stergiou, N., "Objective evaluation of expert and novice performance during robotic surgical training tasks," *Surgical endoscopy*, vol. 23, no. 3, pp. 590–597, 2009.

[26] Junejo, I., Dexter, E., Laptev, I., and Pérez, P., "View-independent action recognition from temporal self-similarities," *PAMI*, 2011.

[27] King, R., Atallah, L., Lo, B., and Yang, G., "Development of a wireless sensor glove for surgical skills assessment," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 5, pp. 673–679, 2009.

[28] Kirk, R. M., *General surgical operations*. Elsevier Health Sciences, 2006.

[29] Kopta, J. A. and others, "An approach to the evaluation of operative skills," *Surgery*, vol. 70, no. 2, pp. 297–303, 1971.

[30] Lalys, F., Riffaud, L., Bouget, D., and Jannin, P., "An application-dependent framework for the recognition of high-level surgical tasks in the or," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, pp. 331–338, Springer, 2011.

[31] Laptev, I., "On space-time interest points," *IJCV*, 2005.

[32] Laptev, I., "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[33] Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B., "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[34] Lin, H. and Hager, G., "User-independent models of manipulation using video contextual cues," in *M2CAI-Workshop*, 2009.

[35] Lin, H., Shafran, I., Murphy, T., Okamura, A., Yuh, D., and Hager, G., "Automatic detection and segmentation of robot-assisted surgical motions," *MICCAI*, 2005.

[36] Lin, H., Shafran, I., Yuh, D., and Hager, G., "Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions," *Computer Aided Surgery*, vol. 11, no. 5, pp. 220–230, 2006.

[37] Martin, J., Regehr, G., Reznick, R., MacRae, H., Murnaghan, J., Hutchison, C., and Brown, M., "Objective structured assessment of technical skill (osats) for surgical residents," *British Journal of Surgery*, vol. 84, no. 2, pp. 273–278, 1997.

[38] Moorthy, K., Munz, Y., Sarker, S. K., and Darzi, A., "Objective assessment of technical skills in surgery," *BMJ: British Medical Journal*, vol. 327, no. 7422, p. 1032, 2003.

[39] MOY, R. L., WALDMAN, B., and HEIN, D. W., "A review of sutures and suturing techniques," *The Journal of dermatologic surgery and oncology*, vol. 18, no. 9, pp. 785–795, 1992.

[40] Naikavde, Y., *Robotic Surgical Skill Assessment Based on Pattern Classification Tools*. PhD thesis, State University of New York, 2012.

[41] nán, C. C. L., "cvBlob." http://cvblob.googlecode.com.

[42] Ng, A. Y., Jordan, M. I., Weiss, Y., and others, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[43] Niebles, J. C., Wang, H., and Fei-Fei, L., "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[44] Padoy, N., Blum, T., Ahmadi, S.-A., Feussner, H., Berger, M.-O., and Navab, N., "Statistical modeling and recognition of surgical workflow," *Medical Image Analysis*, vol. 16, no. 3, pp. 632–641, 2012.

[45] Pudil, P., Novovičová, J., and Kittler, J., "Floating search methods in feature selection," *Pattern recognition letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[46] Reiley, C. and Hager, G., "Decomposition of robotic surgical tasks: an analysis of subtasks and their correlation to skill," in *MICCAI*, 2009.

[47] Reiley, C., Lin, H., Varadarajan, B., Vagvolgyi, B., Khudanpur, S., Yuh, D., and Hager, G., "Automatic recognition of surgical motions using statistical modeling for capturing variability," *Studies in health technology and informatics*, vol. 132, p. 396, 2008.

[48] Reiley, C., Lin, H., Yuh, D., and Hager, G., "Review of methods for objective surgical skill evaluation," *Surgical endoscopy*, vol. 25, no. 2, pp. 356–366, 2011.

[49] Reznick, R. and MacRae, H., "Teaching surgical skills–changes in the wind," *The New England journal of medicine*, vol. 355, no. 25, p. 2664, 2006.

[50] Saggio, G., Santosuosso, G., Cavallo, P., Pinto, C., Petrella, M., Giannini, F., Di Lorenzo, N., Lazzaro, A., Corona, A., D'Auria, F., and others, "Gesture recognition and classification for surgical skill assessment," in *MeMeA*, pp. 662–666, IEEE, 2011.

[51] Schuldt, C., Laptev, I., and Caputo, B., "Recognizing human actions: a local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3, pp. 32–36, IEEE, 2004.

[52] Shawe-Taylor, J. and Cristianini, N., *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[53] Sloetjes, H. and Wittenburg, P., "Annotation by category: Elan and iso dcr.," in *LREC*, 2008.

[54] Soh, L. and Tsatsoulis, C., "Texture analysis of sar sea ice imagery using gray level co-occurrence matrices," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 37, no. 2, pp. 780–795, 1999.

[55] Tao, L., Elhamifar, E., Khudanpur, S., Hager, G., and Vidal, R., "Sparse hidden markov models for surgical gesture classification and skill evaluation," *Information Processing in Computer-Assisted Interventions*, pp. 167–177, 2012.

[56] Trejos, A., Patel, R., Naish, M., and Schlachta, C., "Design of a sensorized instrument for skills assessment and training in minimally invasive surgery," in *Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS & EMBS International Conference on*, pp. 965–970, IEEE, 2008.

[57] Turney, P. D., Pantel, P., and others, "From frequency to meaning: Vector space models of semantics," *Journal of artificial intelligence research*, vol. 37, no. 1, pp. 141–188, 2010.

[58] Von Luxburg, U., "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[59] Wang, H., Ullah, M. M., Klaser, A., Laptev, I., Schmid, C., and others, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.

[60] Wang, H., Ullah, M. M., Klaser, A., Laptev, I., Schmid, C., and others, "Evaluation of local spatio-temporal features for action recognition," in *BMVC 2009-British Machine Vision Conference*, 2009.

[61] Yu, T., Wheeler, B., and Hill, A., "Clinical supervisor evaluations during general surgery clerkships," *Medical Teacher*, vol. 33, no. 9, pp. 479–484, 2011.

[62] Zappella, L., Béjar, B., Hager, G., and Vidal, R., "Surgical gesture classification from video and kinematic data," *Medical image analysis*, 2013.

[63] Zhang, Q. and Li, B., "Towards computational understanding of skill levels in simulation-based surgical training via automatic video analysis," *Advances in Visual Computing*, pp. 249–260, 2010.

[64] Zhao, W., Chellappa, R., and Phillips, P. J., *Subspace linear discriminant analysis for face recognition*. Citeseer, 1999.

[65] Zhou, F., De la Torre, F., and Hodgins, J., "Hierarchical aligned cluster analysis for temporal clustering of human motion," *PAMI*, 2013.