Pattern-Based Real-Time Feedback for a Temporal Bone Simulator

Yun Zhou*
Department of Computing and Information Systems
University of Melbourne

James Bailey†
Department of Computing and Information Systems
University of Melbourne

Ioanna Ioannou‡
Department of Otolaryngology
University of Melbourne

Sudanthi Wijewickrema§
Department of Otolaryngology
University of Melbourne

Stephen O'Leary¶
Department of Otolaryngology
University of Melbourne

Gregor Kennedy‖
Centre for the Study of Higher Education
University of Melbourne

Abstract

Delivering automated real-time performance feedback in simulated surgical environments is an important and challenging task. We propose a framework based on patterns to evaluate surgical performance and provide feedback during simulated ear (temporal bone) surgery in a 3D virtual environment. Temporal bone surgery is composed of a number of stages with distinct aims and surgical techniques. To provide context-appropriate feedback we must be able to identify each stage, recognise when feedback is to be provided, and determine the nature of that feedback. To achieve these aims, we train pattern-based models using data recorded by a temporal bone simulator. We create one model to predict the current stage of the procedure and separate stage-specific models to provide human-friendly feedback within each stage. We use 27 temporal bone simulation runs conducted by 7 expert ear surgeons and 6 trainees to train and evaluate our models. The results of our evaluation show that the proposed system identifies the stage of the procedure correctly and provides constructive feedback to assist surgical trainees in improving their technique.

CR Categories: I.2.1 [Artificial Intelligence]: Applications and Expert Systems—Medicine and science; H.5.2 [Information Interfaces And Presentation]: User Interfaces—Evaluation/methodology

Keywords: real-time feedback, surgical simulation, emerging pattern

1 Introduction

Surgical trainees in the discipline of Otolaryngology dedicate years of training to master the surgical skills required to safely perform temporal bone surgery. Traditionally, they refine their psychomotor skills by practising on plastic bones or cadavers under the supervision of expert surgeons. Experts guide trainees through surgical procedures while providing feedback on the quality of their performance. However, there are limitations to this approach, which include a shortage of cadaver bones, limited availability of expert supervision, and the subjective manner of surgical skill assessment. Due to these challenges, computer-based virtual reality (VR) platforms have recently attracted much attention in the field of surgical education [Agus et al. 2003; Bryan et al. 2001; Kerwin et al. 2009]. The introduction of new techniques such as 3D illusion, haptic feedback and augmented reality have significantly improved the realism of surgical simulators. Such simulators have the potential to provide a cost-effective platform, which allows trainees to practice many surgical cases of varying difficulty, and provides the flexibility of practising repeatedly at their own convenience.

1.1 Surgical Performance Feedback

Performance evaluation plays a critical and essential role in the development of surgical expertise through deliberate practice [Ericsson 2004]. In the traditional apprentice model, expert feedback is usually divided into two categories: immediate and summative. An expert surgeon provides immediate feedback on trainees' performance and guides them through the surgical procedure. Summative feedback is delivered at the end of the procedure by evaluating the end result (e.g. the drilled specimen) and grading their technical skill based on a scale such as the Welling scale [Butler and Wiet 2007]. However, this training approach requires considerable time commitment on behalf of expert surgeons, which is often difficult to arrange. Moreover, grading is subjective and may be influenced by human bias, while the level of detail to which the assessment is carried out is limited. If simulated surgical environments could emulate the role of expert surgeons in training by providing automated performance feedback, the problems of traditional surgical training could be mitigated considerably.

In the past few years, many efforts have been made to improve various aspects of temporal bone surgery simulation. In terms of delivering performance feedback to trainees, recent work has focused on scoring the outcome of surgical tasks (i.e. summative feedback) [Sewell et al. 2008; Kerwin et al. 2012]. Summative feedback typically evaluates the end-product of a surgical task and ignores the rich information provided by real-time performance attributes, such as motion records. Furthermore, since summative feedback is delivered at the end of each task, it does not allow any opportunity to identify and address mistakes as they occur. The real-time data generated by surgical simulators should be mined for knowledge that can be used to help trainees improve their surgical technique as well as their overall performance. There is little work focusing on this aspect in the area of temporal bone surgical simulation.

*e-mail:yuzhou@student.unimelb.edu.au
†e-mail:baileyj@unimelb.edu.au
‡e-mail:ioannoui@unimelb.edu.au
§e-mail:sudanthi.wijewickrema@unimelb.edu.au
¶e-mail:sjoleary@unimelb.edu.au
‖e-mail:gek@unimelb.edu.au

1.2 Challenges of Feedback in Surgical Simulation

The following issues have to be considered when designing and developing a real-time feedback system for a temporal bone surgery simulator (though these issues exist in many other kinds of open surgery simulation):

- The data stream has to be analysed as it is generated and feedback has to be delivered within a short time frame.
- Surgical assessment scales which are used to deliver human expert feedback (such as the Welling Scale) lack clear quantitative definition, thereby making them difficult to translate to values that can be automatically measured by computers.
- Motion-level human classification of surgical technique for training data is often unavailable, resulting in data-driven models being inaccurate. That is, due to lack of class labels at the motion level, we have to make the naive assumption that all drilling movements made by experts are of "expert quality" and all movements made by trainees are suboptimal. However, since this assumption does not generally hold in real life, making such an assumption would adversely affect any model training process and lead to lower accuracy rates.

1.3 Research Contributions

To address the above challenges in the context of temporal bone surgery simulation, we propose a pattern-driven approach to providing real-time feedback focusing on surgical technique. First, we label each stage (representing different sub tasks of the procedure) for each surgical run in our training data set with the assistance of a human expert. Then, we use low level data (such as drill position, velocity, burr size, zoom level etc.) at each time interval of these segmented runs to train an Emerging Patterns (EP) classifier [Dong et al. 1999] to predict the current surgical stage. Second, we aggregate the above low level data into surgical strokes using an online k-cos method inspired by [Hall et al. 2008] and calculate high level metrics for each stroke (such as stroke duration, stroke speed, distance to anatomical structures etc.). These stroke metrics are used to train a separate Emerging Patterns classifier for each stage of the procedure, to capture the differences in technique between expert and trainee groups and propose real-time feedback to improve performance. In using this pattern driven approach, we avoid making the assumption of "polarising" the quality of drill strokes based on expertise that was discussed in section 1.2, and only use the overall classification of each surgical run as a soft label to mine discriminative patterns. Different patterns are used for different stages since surgical technique varies between sub tasks of the surgical procedure.

In summary, this paper makes the following contributions:

- To the best of our knowledge, this is the first formal study in the use of Emerging Patterns to detect surgical stage and provide feedback in a simulation environment.
- The proposed algorithms are designed to satisfy the time constraints imposed by a real-time system and to maximise usability by providing human understandable feedback.
- Experimental evaluation shows that the proposed Emerging Patterns methods recognise surgical stages with a high level of accuracy and generate useful feedback on surgical technique.

The rest of the paper is organised as follows. Section 2 discusses related work. Section 3 describes our temporal bone simulator. In section 4, we provide a brief background on Emerging Patterns. Section 5 provides an overview of the proposed feedback system, while sections 6 and 7 explain the stage prediction and feedback models in detail. Section 8 reports the results of our algorithm evaluation, and section 9 concludes the paper.

2 Related work

In recent years, the analysis of surgical work-flow has been gaining attention as it has become evident that an immense amount of information can be obtained during a surgical procedure. Such information can be used to deliver context-aware evaluation of performance and provide surgical technique feedback in real-time. However, most existing work on automated performance evaluation for temporal bone surgery simulators is limited to the assessment of surgical outcomes [Sewell et al. 2008; Kerwin et al. 2012].

Sewell et al. [2008] used coloured dots to indicate whether the correct bone region was drilled in an attempt to provide guidance towards completion of a procedure. They also provided information such as force and speed of the surgical drill in an evaluation console. Quantitative information presented in this manner assumes that the user possesses a "reference of correctness" to be able to usefully interpret it. Providing comparative information using the raw metrics is more useful, but could be difficult for simulation users to monitor and react to while performing the procedure. It also does not suggest to the user exactly what action to undertake to improve their surgical technique.

In previous work, we have used random forest models to predict surgical expertise and generate meaningful automated real-time feedback in a temporal bone surgery simulator [Zhou et al. 2013]. While this approach showed promise, the use of random forest models necessitated the assumption that all data provided by experts contained optimal surgical techniques while all data provided by trainees was sub-optimal. This assumption is generally not true in the real world.

Other developments in this area can be observed in the field of minimally invasive surgery [Haro et al. 2012; Rosen et al. 2001; Stylopoulos et al. 2004; Forestier et al. 2012]. Haro et al. [2012] used multiple kernel learning of Support Vector Machines to identify surgical gestures from intra-operative videos. Rosen at al. [2001] used force/torque from endoscopic instruments to train two Hidden Markov Models (HMM) representing different surgical skill levels and used likelihood to predict the expertise of new motions. Forestier et al. [2012] used Dynamic Time Warping (DTW) to classify the surgical process of expert and trainee groups. Although these studies provided online evaluation, an evaluation score alone provides trainees with no specific suggestions for improvement. Furthermore, classification models in minimally invasive surgery are based on the use of a set of tools, and identifiable "gestures" associated with these tools. On the other hand, open surgery such as temporal bone surgery often utilises a small set of instruments such as surgical drills and suction devices and there are many ways to achieve a correct outcome. As such, it is difficult to identify specific gestures that represent good surgical technique. Therefore, it is not practical to extend the same models and algorithms used in minimally invasive surgery to generate automated feedback for temporal bone surgery.

Rhienmora et al. [2011] proposed a "follow me" approach to dental surgery (which is also a type of open surgery) by providing a ghost drill during the simulation to guide the trainee. However, the pace of an expert is often faster than that of a trainee who is not as familiar with the procedure. Reconciling this difference by synchronising the expert run with the current run is not an easy task, and was not addressed by the researchers. Rhienmora et al. [2011] also assessed surgical skill in the context of a crown preparation

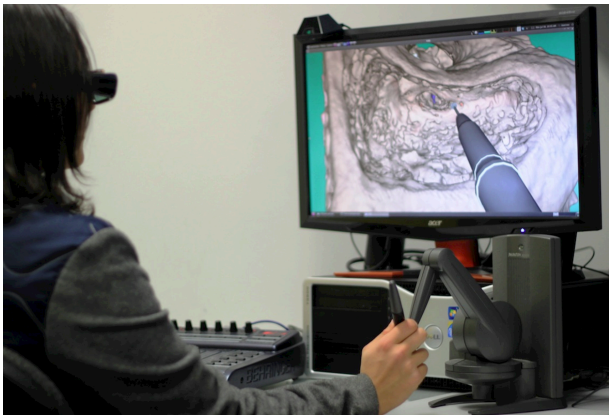

Figure 1: *Using the temporal bone simulator*

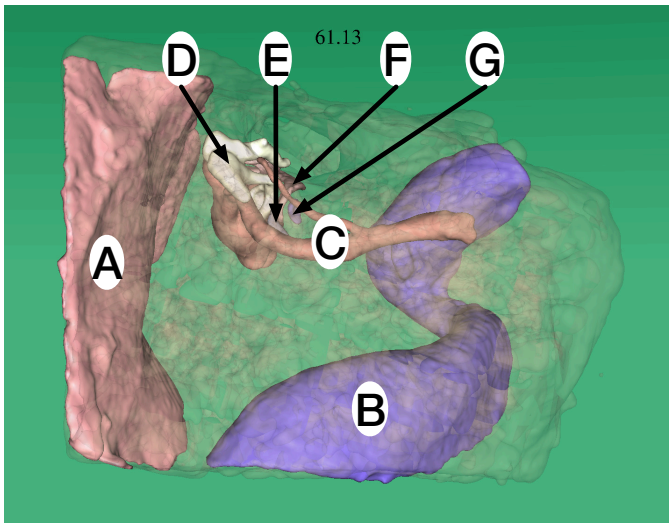

Figure 2: *A transparent temporal bone showing seven anatomical structures. A: dura, B: sigmoid sinus, C: facial nerve, D: ossicles, E: stapedius tendon F: basilar membrane, G: round window*

procedure and delivered feedback on drill force, position and orientation by comparing them to the average value of an expert group. This simple approach to real-time feedback cannot be extended to other types of surgery where it may be hard to find a set of standard force, position and orientation values across experts, as is the case with temporal bone surgery.

In view of the limitations of the existing methods discussed above and the requirements stated in section 1, we propose the use of Emerging Patterns (EP) [Dong and Li 1999] to develop a real-time feedback system for surgical simulators. EPs have been applied in other fields such as body sensor networks, to recognise the activity of individual users in given environments (e.g. at home) [Wang et al. 2012].

A detailed description of Emerging Patterns can be found in section 4. This pattern-based idea was inspired by [Sewell et al. 2005], who detected dangerous behaviours in temporal bone surgery using visibility checking, that is, checking whether a user is drilling parts of the bone that he/she cannot see directly. Our feedback model extends this idea by using a pattern-based approach in two ways. First, we identify patterns that represent low quality surgical technique such as dangerous behaviours or inefficient drilling. Second, we use these patterns to generate constructive feedback to assist users in improving their technique.

3 Virtual Reality Simulation Environment

Our simulation environment uses a 3D monitor and stereo shutter glasses to display a 3D virtual temporal bone model constructed from segmented micro-CT data of a human temporal bone. The bone and key anatomical structures were segmented by an expert surgeon using a semi-automatic segmentation method. Trainees interact with the temporal bone model using a pen-like haptic device which represents a surgical drill in the VR environment. The haptic device provides force feedback to the user and can be used to remove material from the virtual bone.

Figure 1 shows the simulator in-use and Figure 2 shows a screen shot highlighting the underlying anatomical structures present in the simulated temporal bone.

This high fidelity VR temporal bone simulator can be used to perform any type of temporal bone drilling task. For this study, the surgical procedure we considered consisted of a cortical mastoidectomy (removal of bone surrounding the important anatomical structures described in Figure 2, A-D), followed by posterior tympanotomy (removal of most of the bone overlying the facial nerve and the small chorda tympani nerve that lies in front of it, and then drilling away the bone between these two structures) and cochleostomy (drilling a small hole into the cochlea as it is viewed through the posterior tympanotomy). This procedure is part of cochlear implantation surgery. We collected 27 temporal bone simulation runs of this procedure, performed by 7 expert ear surgeons and 6 trainees. Each participant performed 1 to 3 runs which provided 16 expert runs and 11 trainee runs.

The simulator recorded two kinds of measures: outcome measures and technique measures. Outcome measures consisted of a time series of drilled voxel positions. Technique measures were recorded at the graphics frame rate of the simulator (which was approximately 30 Hz), and included kinetic metrics, simulator settings and proximity to anatomical structures.

4 Emerging Patterns

An "Emerging Pattern" is a set of items (each item representing a pair of attribute and value) that has a significantly higher support (i.e. frequency) in one class than that in other classes [Dong and Li 1999]. A comprehensive treatment can be found in [Dong and Bailey 2013] book. We refer to these patterns as the *EPs of class* c_i. Emerging Patterns were proposed to capture multi-attribute contrasts between data classes or trends over time. Formally, let $D = \{d_1, d_2, ..., d_n\}$ be a dataset consisting of a list of instances. Each row in a dataset is defined as one instance and each column in a dataset is defined as one attribute. The value of an attribute is required to be discrete. The class attribute is a special column in a dataset. Let $C = \{c_1, c_2, ..., c_k\}$ be the set of class values that the data in D falls into. Each attribute-value pair of an instance is called an item and an itemset is a set of items in a dataset. Let I be the set of all items in D, d be an instance within D and $|*|$ be the number of members in the collection. The support of an itemset E which is a subset of I is defined as:

$$sup_D(E) = \frac{|\{d \in D | E \subseteq d\}|}{|D|} \quad (1)$$

The growth rate GR (or differentiation power) of an itemset E is

defined as the ratio between the support of E in one class and the support of E in other classes:

$$GR_D(E) = \begin{cases} 0 & \text{if } sup_{D_c}(E) = 0 \\ \infty & \text{if } sup_{D-D_c}(E) = 0 \\ \frac{sup_{D_c}(E)}{sup_{D-D_c}(E)} & \text{otherwise} \end{cases} \quad (2)$$

where D_c is the subset of data in D belonging to class c and $D - D_c$ is the subset of data in D not belonging to class c. An itemset E is considered to be an Emerging Pattern (EP) in a given class dataset D_c when $sup_{D_c}(E) \geq mSup$ and $GR_{D_c}(E) \geq \rho$, where $mSup$ and ρ are predefined thresholds for support and growth rate respectively. By defining a higher support threshold $mSup$, we can guarantee that the mined EP is minimally affected by noise in the data.

Although each EP is a good indicator of its representative class, it only covers a small portion of the data. Therefore, an EP score function can be defined to aggregate the evidence contained in a set of EPs representing a class.

$$S(d, c) = \sum_{E \subseteq d, E \in EP_c} \frac{GR_D(E)}{GR_D(E) + 1} \times sup_{D_c}(E) \quad (3)$$

where, S is the score, d is a data instance, E is an EP, EP_c is the set of EPs of class c, $GR_D(E)$ is the growth rate of E in the dataset D, $sup_{D_c}(E)$ is the support for E in class c. $S(d, c)$ considers all the EPs of a class c_i to decide whether d should be in class c_i.

Table 1: *Example data for stage prediction*

Force	Speed	Stage
$(0, 0.2]$	$(4, 6]$	1
$(0, 0.2]$	$(4, 6]$	1
$(0.2, 0.5]$	$(1, 4]$	1
$(0, 0.2]$	$(4, 6]$	2
$(0.2, 0.5]$	$(6, 8]$	2

To illustrate these concepts we provide a toy example of a dataset for stage prediction, shown in Table 1. The dataset has three attributes (force, speed and stage). Stage is the class attribute. Force and speed are discretised from numerical values. The dataset has five instances, three of which belong to stage 1 and two of which belong to stage 2.

Now let us consider an itemset consisting of two items: 'force in $(0, 0.2]$; speed in $(4, 6]$'. The support of this itemset in the subset of instances belonging to stage 1 is $\frac{2}{3}$. The support of it in the stage 2 subset is $\frac{1}{2}$. It is an EP of stage 1 with a growth rate $\frac{2}{3} \div \frac{1}{2} = \frac{4}{3}$, for any $1 < \rho \leq \frac{4}{3}$. If we set $mSup = \frac{1}{3}$ and $\rho = \frac{4}{3}$, then stage 1 has totally five EPs as illustrated in Table 2.

Table 2: *EP list in stage 1*

EP itemset	sup_1	GR_1
Force in $(0, 0.2]$	$2/3$	$4/3$
Speed in $(4, 6]$	$2/3$	$4/3$
Speed in $(1, 4]$	$1/3$	∞
Force in $(0, 0.2]$; Speed in $(4, 6]$	$2/3$	$4/3$
Force in $(0.2, 0.5]$; Speed in $(1, 4]$	$1/3$	∞

Suppose we have an unseen instance '$(0, 0.2]$;$(1, 4]$', given these five EPs, it only contains EP 1) and 3). Therefore, its score for stage 1 is $S(d, 1) = \frac{1.33}{1.33+1} \times 0.66 + \frac{\infty}{\infty+1} \times 0.33 \approx 0.71$.

5 Feedback System Overview

Figure 3 provides an overview of the proposed real-time feedback system. This system operates in two steps.

1. Surgical stage prediction: Surgical technique varies between different stages (sub-tasks) of the procedure, and therefore, it is important to predict the current stage before providing feedback.

2. Feedback construction: Different feedback should be provided for different stages of the procedure and therefore, separate models should be trained for each stage.

To predict the stage, we build an EP-based stage model [Dong et al. 1999]. To train this model, we first segment the collected surgical runs into stages with the assistance of a human expert. Then, the stage model is trained offline by mining a set of EPs from low-level data (such as drill position, velocity, burr size, zoom level etc.) and their corresponding stage label, using the algorithm proposed by [Li et al. 2007]. During a simulation procedure, stage predictions are made in real-time by reporting the stage with the highest EP score computed from the mined EPs.

After predicting the current stage of the procedure, feedback should be generated and delivered at appropriate times to improve surgical skills. Since the granularity of low-level data is too small to contain enough information about surgical technique, we adopt an approach based on k-cos [Hall et al. 2008] to aggregate a list of low-level data into a series of surgical strokes and obtain high-level metrics for each stroke, such as stroke distance, speed and force. EPs require nominal data, so we use the entropy-based discretisation method [Fayyad and Irani 1993] to discretise numeric attributes into a number of disjoint intervals. This discretised vector stream is used to discover EPs that can be used to provide feedback.

To train the EP feedback models for each stage, we use the expertise label of each run as a soft label. We mine a set of EPs to represent these two groups offline. During a procedure, a sliding window is used to store the discretised stroke data. Once this buffer is full, we choose the EP representing a low quality surgical technique that appears the most in this window. If the recurrence of this EP is above a predefined threshold, we regard it as a potential fault that needs to be rectified. Then, we iteratively change each item in this pattern and evaluate each change using its feedback score (defined in section 7.2). Finally we choose the change with highest feedback score as the proposed suggestion to the user.

6 Stage Prediction

As mentioned above, to successfully provide feedback on surgical technique, we divide the surgical procedure into stages. A stage is defined as a time interval where a certain sub-task is performed within which time the surgical technique is relatively uniform. This ensures that the different techniques used during different stages of the procedure can be treated separately.

This section describes how we build a stage prediction model from the low-level training data and how this model is used to recognise the current surgical stage of an ongoing simulation run.

6.1 Stage Definition and Low-Level Metrics

After consultation with expert ear surgeons, we have divided the surgical procedure into five stages, as shown in Table 3. To detect the stage, a low-level data stream is collected from the simulator at the graphics frame rate of approximately 30 Hz. This data stream is comprised of the attributes listed in Table 4.

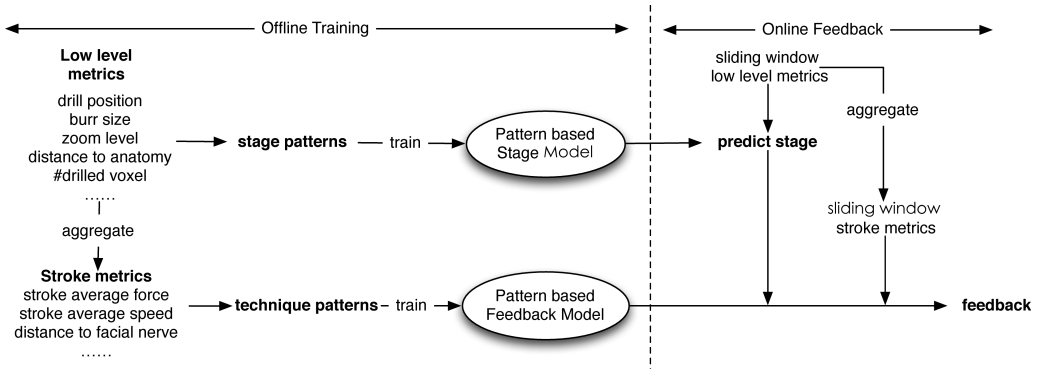

Figure 3: *Overview of EP-based real-time feedback system*

Table 3: *Definition of stages in the surgical procedure*

Stage ID	Stage purpose
1	Exposing the dura and sigmoid sinus
2	Exposing the incus
3	Identifying the facial nerve
4	Drilling through the facial recess
5	Making the cochleostomy

Table 4: *37 low level metrics collected from temporal bone simulator*

Time stamp (seconds)
Tool position, orientation and force metrics
Current force applied by drill tool (X,Y,Z)
Current position of drill tool (X,Y,Z)
Current orientation of drill tool (X,Y,Z, Angle)
Simulator settings
Current burr spinning speed
Radius of the current burr
Current zoom Level
Anatomical structure metrics
Current specimen position (X,Y,Z)
Current specimen orientation (X,Y,Z, Angle)
Number of drilled voxels from each segment (bone and anatomical structures) of the specimen
Distance of the drill tip from the closest point of each anatomical structure

6.2 EP-Based Stage Prediction

To carry out stage prediction, we first train an EP-based classifier offline. To this end, the low-level dataset comprised of 37 continuous metrics is discretised using the entropy-based discretisation method [Fayyad and Irani 1993]. From these discrete vectors, we mine a set of EPs for each stage from using the algorithm discussed in [Li et al. 2007]. Next, we derive the base score for each class (i.e. stage in this case). The training process is illustrated in Algorithm 1.

Since there may be imbalance in the spread of EPs across the classes, the raw score would be biased towards the classes occurring more frequently in the dataset. To overcome this issue, a normalised score NS is defined as the ratio of the raw score and a base score BS [Dong et al. 1999]. The base score is defined as the median of the raw EP scores for each class. If the median is zero, the smallest non-zero value is used as the base score. $NS(d,c) = \frac{S(d,c)}{BS(c)}$, where S is the EP score, d is an instance of the dataset, c is the class. This normalised score can be used to classify a new instance d_n. The class c_i with the highest normalised score $NS(d_n, c_i)$ is chosen as the predicted class for d_n.

Input: D : dataset;
Stages S=$\{s_1$,s_2,...,$s_5\}$;
Discretise dataset D;
Mine EP set E_i for each class with $mSup = 1\%$ and $\rho = 3$;
Compute EP score for all instances in D for corresponding stage;
For each stage s_k, use median as base score;
if $basescore[k]$ == *0* **then**
  $basescore[k]$ = smallest non-zero EP score for stage s_k;
end

Algorithm 1: Training phase of stage prediction model

In the prediction phase, instead of predicting the stage for each individual data instance, we use a sliding-window to ensure smoother results. For each low-level instance in the sliding-window, the normalised EP score is calculated. The stage that has the largest normalised EP score is assigned as the stage for that instance. The stage that occurs the most within the sliding-window is set as the predicted stage. The sliding-window is then moved to collect the next set of instances from the low-level data stream. The prediction stage is described in Algorithm 2.

7 EP-Based Real-Time Feedback

Different stages of the surgical procedure require different surgical techniques, therefore the feedback provided should be customised for each stage. For example, a short stroke might be inefficient in stage 1, but may be common behaviour in stage 4, and a high force magnitude in stage 1 may be an efficient way to open up the field of view, but the same stroke may be dangerous in later stages due to proximity to sensitive anatomical structures. Therefore, we train a separate model for each stage, which is used to generate feedback once the stage is predicted using the stage prediction model discussed in section 6.

The goal of developing these feedback models is to emulate the role of an expert trainer by providing human-friendly suggestions to improve the surgical technique of trainees using the simulator. For example, an expert trainer would suggest that the trainee should increase/decrease parameters such as burr size or stroke length. They would also ask the trainees to drill parallel to anatomical structures and open up the bone for a better field of view. While the former is relatively easy to automate, the latter is more difficult. In this paper, the models we develop only provide feedback on one

Input: Window of low-level metrics W = w_1,w_2,...,w_l;
Stages S = $\{s_1,s_2,...,s_5\}$
Output: Predicted Stage for the window W

```
for each w_i ∈ W do
    Discretise w_i;
    for each s_j in S do
        Compute S(w_i, s_j) as the raw score of w_i in s_j;
        Calculate normalised score NS(w_i, s_j) by dividing the
        raw score by the base score BS(s_j);
    end
    if sum of NS for all stages are zero then
        #No EPs found for w_i;
        #Assign the previous stage prediction #since stage do not
        change frequently;
        stage[w_i]=stage[w_{i-1}];
    else
        Assign the stage with largest normalised score to
        stage[w_i];
    end
end
return mode of stage
```

Algorithm 2: Prediction phase using pattern-based stage model

Table 5: *Stroke metrics aggregated from the low level metrics*

Motion-based metrics
Stroke duration
Stroke length
Average stroke speed
Average acceleration
Average force
Stroke straightness
Average centroid distance
No. of bone voxels removed by stroke
Average force near 7 anatomical structures
Average speed near 7 anatomical structures
Proximity metrics
Average distance to the facial nerve
Average distance to the ossicles
Average distance to the basilar membrane
Average distance to the dura
Average distance to the sigmoid sinus
Average distance to the stapedius tendon
Average distance to the round window

stroke attribute at a time.

7.1 Stroke Metrics

To meaningfully analyse and classify surgical technique, we divide drill trajectories into surgical "strokes". A stroke is a collection of low-level data points segmented using an adapted k-cos algorithm based on [Hall et al. 2008]. Only the drill motions that result in material removal are considered to be part of a stroke. Once a stroke is detected, the low-level data is aggregated into a set of more meaningful metrics to be used in the feedback models. Stroke metrics include aggregated motion-based metrics and proximity data as shown in Table 5.

To calculate the average force and speed near each anatomical structure, we define a proximity threshold of 5mm. The metrics are then calculated as the average force and speed of the stroke during the time when the drill tip is within less than 5mm of each anatomical structure. In total, we derive 29 stroke-level metrics that are used to describe stroke technique.

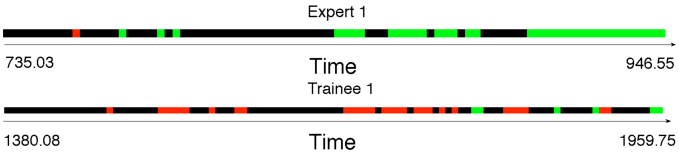

Figure 5: *Example of a surgical performance model in stage 3 of an expert and a trainee. Black indicates common techniques; red indicates low quality techniques; green indicates high quality techniques.*

7.2 Stroke Quality Model

To build a traditional data-driven feedback model [Witten and Frank 2005], an expert surgeon would have to go through all the strokes in each simulation run and label them as 'Expert' or 'Trainee'. If we use the label of the entire simulation run to train the model, we assume that all strokes performed by an expert are expert strokes and vice versa, which results in inaccurate models. For example, force and speed are two common physical measures that model stroke technique. Figure 4 shows the distribution of these two measures for expert and trainee groups for stage 3. It can be observed that the histograms of these two measures for the two groups are very similar.

An alternative that allows the modelling of different motions in surgery is clustering [Witten and Frank 2005]. However, clustering suffers from two drawbacks: 1) It is difficult to guide clustering algorithms to find clusters that differentiate stroke quality using domain knowledge. The clusters that are formed may have different meanings (e.g. each cluster may represent strokes of similar shape instead of stroke technique). 2) Clustering also ignores the label of the simulation run, which should be used in some way to discover the techniques that distinguish experts and trainees.

We illustrate a more suitable model to evaluate surgical performance in Figure 5, which shows the predicted surgical performance of an expert and trainee simulation run in stage 3 using an EP-based algorithm. Black indicates common techniques, red indicates low quality techniques, and green indicates high quality techniques.

The strategy is to use the label of each simulation run as a soft label when we find EPs in our dataset. These patterns are divided into two categories: 1) EPs of high quality strokes are those EPs that appear often in expert surgeries but rarely in trainee surgeries. We call these *expert EPs*; 2) EPs of poor quality strokes are those that appear frequently in trainee surgeries but not so frequently in expert surgeries. We call these *trainee EPs*.

These two sets of EPs allow us to deliver online feedback using the sliding-window approach. This online feedback approach consists of two steps: 1) We examine each stroke in the sliding window and identify the most popular EP. 2) If the most popular EP is a trainee EP and it occurs more frequently than a predefined percentage (50% of strokes in our case), we conclude that the surgical technique quality is poor. In such a case we want to provide feedback to help the trainee improve their technique.

In order to generate feedback, we iterate through all discrete attribute-value pairs in the most popular EP and regard each pair as a potential candidate for feedback. Out of these possibilities, we need to choose the pair that would produce the largest improvement

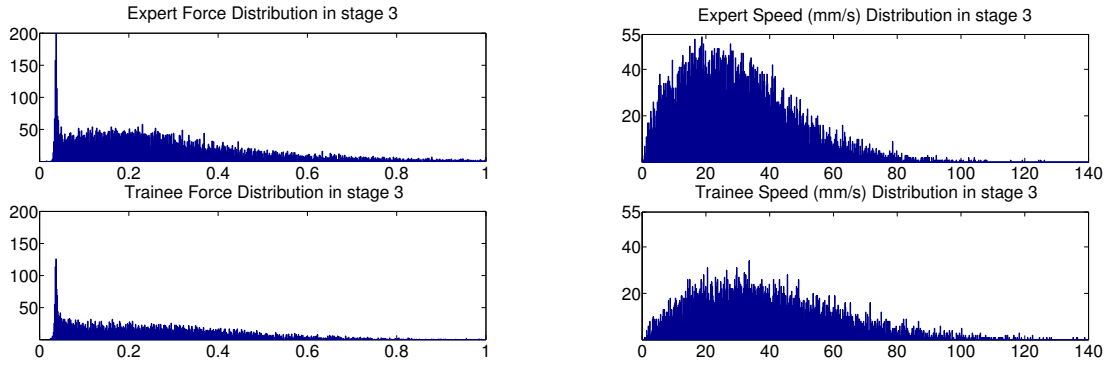

Figure 4: *Histograms of force and speed in stage 3 for expert and trainee groups*

Table 6: *Example of mined EPs*

Mined EP	$[Sup_{pos}, Sup_{neg}]$
EP_1: force in (0.01, 0.09], speed in (2,5]	[12%,1%]
EP_2: force in (1.0, 4.0], speed in (1,2]	[1%,4%]
EP_3: force in (0.01, 0.09], speed in (1,2]	[1%,3%]

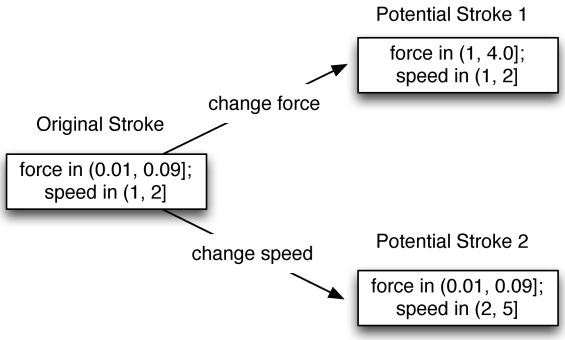

Figure 6: *Example of potential feedback*

in stroke technique.

Let us consider an example where we mined one expert EP (EP_1) and 2 trainee EPs (EP_2 and EP_3) from our training dataset as shown in Table 6. The first column is the EP and the second column is the support for the expert and trainee classes respectively. Suppose inside a window of ten strokes, we find that all the strokes contain EP_3. If we assume that speed and force have two different value intervals, then there are 2 possible changes we can suggest: changing the force to $(1.0, 4.0]$ or changing the speed to $(2, 5]$ (see Figure 6). The former change would result in EP_2 being present in all modified strokes. The latter would result in EP_1 being the most popular EP in the modified strokes. Since EP_1 is an expert EP with a high support, this change gets a higher feedback score. Therefore, the optimal feedback to provide is "increase speed to (2,5]". Changing the force does not achieve expert stroke technique as it results in trainee EPs. Now let us consider a real trainee EP that appears frequently in our training dataset for stage 3: **stroke length in '(11.47, inf)', stroke straightness in '(0.38, inf)', force in '(0.17, 0.67]'**

From a surgical point of view, this means that when long strokes are used by trainees, they are more likely to apply more force. Applying too much force is dangerous in stage 3, as it may damage the facial nerve and cause facial paralysis. Therefore, the ideal feedback should be a suggestion to decrease force at this stage. To determine the best possible change to suggest, we generate a window of synthetic strokes from the original sliding window of stroke data, by changing the value of one attribute in all strokes to the value of each potential change. Then, we compute a feedback score (FS) for each change, as the increase in the likelihood of expert stroke technique caused by that change.

$$FS(s, s', EP) = \frac{\sum_{s'=1}^{l} S(s', EP(exp))}{\sum_{s'=1}^{l} S(s', EP)} - \frac{\sum_{s=1}^{l} S(s, EP(exp))}{\sum_{s=1}^{l} S(s, EP)} \quad (4)$$

where s is a stroke in the original window, s' is a stroke modified from s according to one of the possible changes, l is the number of strokes in the window, $EP(exp)$ is an expert EP, and S is the score defined in equation (3).

We select the change that has the highest FS value as the feedback suggestion. The process for selecting the optimal feedback is illustrated in Algorithm 3. Since attributes are all discretized, there is a finite candidate value set for each attribute. Our idea is to iterate through this finite set to select the optimal feedback. Note that when we consider all possible changes in attribute values, we ignore attributes such as "force near structure" and "speed near structure", as they can only be changed when near a structure, and even then it is often not practical. We call these "near structure" attributes in Algorithm 3.

8 Evaluation and Results

In this section, we present the results of experiments conducted to evaluate our system. We first illustrate the performance of the stage prediction model. Then we show the effectiveness of using EPs to deliver feedback.

8.1 Stage Prediction Performance

Examination of simulator performance videos revealed that stage 2 is often not distinct from stage 1 or stage 3. For example, a user is drilling bone near the incus (stage 2) and then goes back to removing bone near the dura (stage 1). Our stage labelling assumed sequential progression through the stages and did not account for non-sequential stage progression. As a result of this problem and the fact that stage 2 is a relatively short stage, few EPs were found

Input: A window of strokes S = $\{s_1,s_2,...,s_l\}$;
A list Expert EPs EEP = $\{EP_{e1},EP_{e2},EP_{e3},...,EP_{en}\}$;
A list Trainee EPs TEP = $\{EP_{t1},EP_{t2},EP_{t3},...,EP_{tm}\}$;
Threshold t;
Output: Proposed Feedback fb

```
countTEP= count each EP in TEP for S;
if max(countTEP) > t then
    maxFbScore = −∞;
    fb = null;
    mEP = most popular Trainee EP;
    for each item in mEP do
        if not near structure attribute then
            for each possible change of this attribute do
                value = change;
                fbCandidate = <attribute, value>;
                S′= modify all strokes in S according to
                fbCandidate ;
                score = FS(S, S′, EEP ∪ TEP);
                if score > maxFbScore then
                    maxFbScore = score;
                    fb = fbCandidate;
                end
            end
        end
    end
end
return fb;
```

Algorithm 3: EP-based feedback algorithm

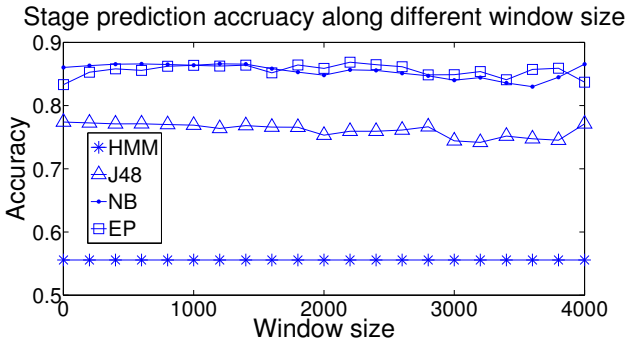

Figure 7: *Stage prediction of different approaches*

for stage 2. Therefore, stage 2 has been omitted from the stage prediction evaluation results.

To evaluate the performance of EP-based stage prediction, we compared it to some other popular models. For this purpose, we built a sequential Hidden Markov Model (HMM) with four states to represent the surgical stages, and calculated the probability of each stage over a sliding window of input. In addition, we carried out the same tests using a Naive Bayes (NB) classifier, and decision trees (J48). We calculated the performance of each model across different sliding window sizes. We also identified the highest accuracy achieved for each approach by selecting the optimal window size. We used ten-fold cross-validation for the calculation of accuracy. We selected 90% of the 27 simulation runs to build the models and used the remaining 10% for testing.

Figure 7 shows the results of the comparison for window sizes ranging from 1 to 4000 data points. Table 7 shows the best accuracy that could be obtained for each method and the corresponding window size. The results show that EP-based stage prediction accuracy significantly outperformed decision trees and HMM models. The size of the sliding window did not significantly affect the level of accuracy in any of the methods. Although EP and NB models showed similar accuracy levels, it is hard to use a NB model to deliver feedback. Since NB assume that each stroke attribute is independent, which does not hold in real world and thus adversely affects feedback quality. Hence we consider EP to be a more suitable algorithm for this area of application, as it can predict the stage of the procedure with high accuracy and also deliver useful feedback.

Table 7: *Stage prediction with optimal window size*

Method	Optimal Window Size	Accuracy
EP	2200	86.60%
NB	1200	86.86%
J48	1	77.40%
HMM	1	55.57%

To determine the accuracy of each stage prediction, we calculated the confusion matrix as shown in Table 8. The columns show the ground truth stage labels and the rows show the predicted stages. We observe that stage prediction accuracy is high. Moreover, the majority of errors that are made are predictions for adjacent stages. For example, all the data from stage 4 were predicted to be stages 3, 4, or 5, with the majority (85.17%) being correctly classified as stage 4. These results are acceptable as there is no clear definition of the stages and some overlap between adjacent stages is unavoidable.

Table 8: *Confusion matrix for EP-based stage prediction*

Predict \ Truth	S1	S3	S4	S5
S1	89.76%	10.24%	0%	0%
S3	12.43%	81.38%	6.19%	0%
S4	0%	11.94%	85.17%	2.89%
S5	0%	0%	6.55%	93.46%

8.2 EP-Based Feedback Performance

To illustrate the importance of selecting the correct attribute to change when delivering feedback, we established a baseline by randomly choosing an attribute to change. We then carried out a comparison of the results from the random selection method with the EP-based approach. Since different support ($mSup$) and growth rate (ρ) thresholds result in different sets of EPs, we varied them in our tests in order to discover the optimal thresholds for best feedback score FS (as defined in section 7.2). For a given support, we started the growth rate from 2 and increased it until we could find no more than two EPs in each class. Figure 8 shows the results of these tests for stage 3. In all cases, we can see that the feedback score FS of the EP-based feedback selection significantly outperformed the random selection.

We also observe that the feedback quality is highly correlated to the support and growth rate thresholds. Given a certain support, the general trend of the feedback score is to increase in tandem with the growth rate. Since patterns with higher growth rate have more discriminative power, raising the growth rate threshold results in patterns with higher contrast between experts and trainees. On the other hand, a high growth rate threshold will reduce the number of patterns discovered, which could adversely affect the delivery of feedback. The support threshold also affects the feedback quality. In Figure 8, feedback score drops from 0.34 at $mSup = 0.05$ to 0.24 at $mSup = 0.15$. Based on these tests, the optimal support

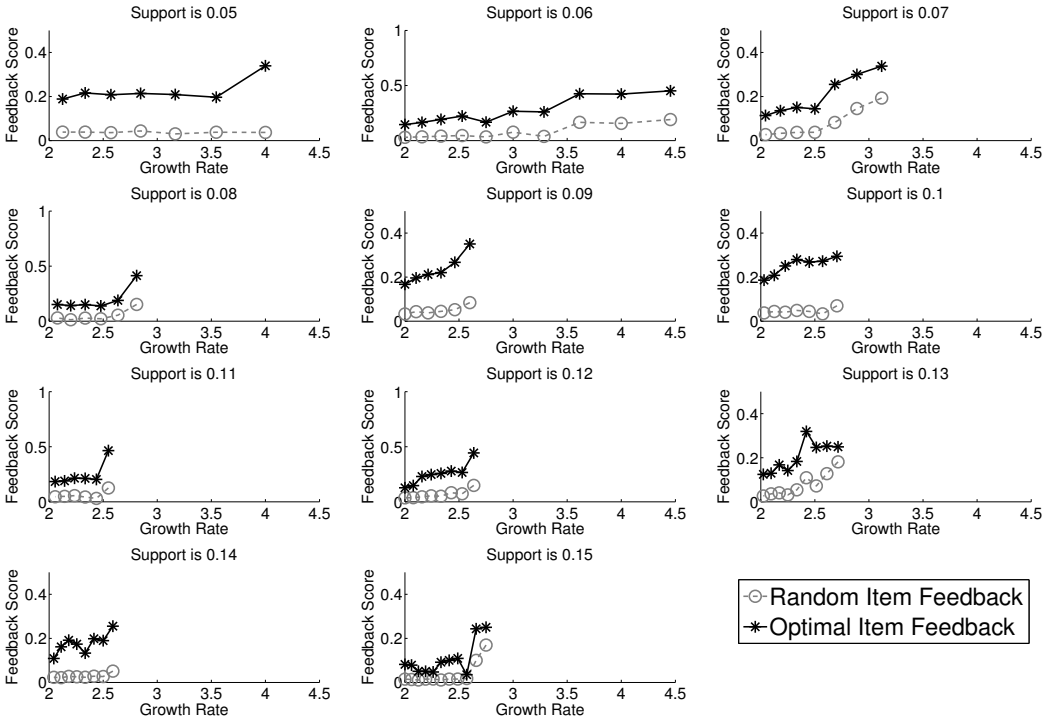

Figure 8: *Feedback scores for different supports and growth rates in stage 3*

and growth rate for stage 3 is $mSup = 0.11$ and $\rho = 2.55$ respectively and the corresponding feedback score is 0.46.

Table 9: *Contingency table of stroke quality across two groups. Columns show stroke quality classification according to the EP model and rows show the different groups.*

	Expert Stroke	Common Stroke	Trainee Stroke
Expert Runs	29.20%	63.18%	7.62%
Trainee Runs	12.93%	73.16%	13.91%

Using the optimal values for support and growth rate, we examined stroke quality among expert and trainee groups. We used ten-fold cross-validation to evaluate stroke quality. We selected 90% of the 27 simulation runs to build the stroke quality model and used the remaining 10% for testing. Table 9 shows the results of this evaluation. We can see that common strokes are dominant in both groups, but experts are more likely to perform expert strokes (29.20% vs 12.93%) compared to trainees, and less likely to perform trainee strokes (7.62% vs 13.91%).

9 Discussion and Conclusion

We have presented a framework to automatically deliver online context-aware feedback in a temporal bone surgical simulation. We discussed two pattern-based models: 1) to predict the current stage of the surgical procedure and 2) to detect poor surgical technique and deliver feedback. Both models build a set of EPs offline from a training dataset and deliver predictions in real-time during simulator training.

Our evaluation showed that the pattern-based stage prediction model achieves a high accuracy when compared to other methods. The evaluation of the pattern-based feedback models also demonstrated that they have the potential to successfully to deliver constructive feedback on surgical technique.

The next step in this research is to test the effectiveness of these models in improving the surgical technique of trainee ear surgeons. This will be achieved by prospectively running trials with trainee surgeons using the temporal bone simulator with real-time feedback and comparing their performance with that of a control group. The feedback generated by the models will also be assessed by expert surgeons to evaluate its usefulness and meaningfulness in practice.

References

AGUS, M., GIACHETTI, A., GOBBETTI, E., ZANETTI, G., AND ZORCOLO, A. 2003. Real-time haptic and visual simulation of bone dissection. *Presence-Teleop Virt 12*, 1, 110–122.

BRYAN, J., STREDNEY, D., WIET, G., AND SESSANNA, D. 2001. Virtual temporal bone dissection: a case study. In *Proc. Conf. Visualization'01*, IEEE Computer Society, 497–500.

BUTLER, N. N., AND WIET, G. J. 2007. Reliability of the Welling scale (WS1) for rating temporal bone dissection performance. *Laryngoscope 117*, 10, 1803–1808.

DONG, G., AND BAILEY, J., Eds. 2013. *Contrast Data Mining: Concepts, Algorithms, and Applications*. CRC Press.

DONG, G., AND LI, J. 1999. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. 5th ACM*

SIGKDD Conf. Knowledge, Discovery and Data Mining, ACM, 43–52.

DONG, G., ZHANG, X., WONG, L., AND LI, J. 1999. CAEP: Classification by aggregating emerging patterns. In *Discovery Science*, Springer, 30–42.

ERICSSON, K. A. 2004. Deliberate practice and the acquisition and maintenance of expert performance in medicine and related domains. *Acad Med 79*, 10, S70–S81.

FAYYAD, U. M., AND IRANI, K. B. 1993. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In *Proc. AI'93*, 1022–1027.

FORESTIER, G., LALYS, F., RIFFAUD, L., TRELHU, B., AND JANNIN, P. 2012. Classification of surgical processes using dynamic time warping. *J Biomed Inform 45*, 2, 255–264.

HALL, R., RATHOD, H., MAIORCA, M., IOANNOU, I., KAZMIERCZAK, E., O'LEARY, S., AND HARRIS, P. 2008. Towards haptic performance analysis using k-metrics. In *Proc. HAID'08*, 50–59.

HARO, B. B., ZAPPELLA, L., AND VIDAL, R. 2012. Surgical gesture classification from video data. In *Proc. MICCAI'12*, 34–41.

KERWIN, T., SHEN, H.-W., AND STREDNEY, D. 2009. Enhancing realism of wet surfaces in temporal bone surgical simulation. *IEEE T Vis Comput Gr 15*, 5, 747–758.

KERWIN, T., WIET, G., STREDNEY, D., AND SHEN, H.-W. 2012. Automatic scoring of virtual mastoidectomies using expert examples. *IJCARS 7*, 1, 1–11.

LI, J., LIU, G., AND WONG, L. 2007. Mining statistically important equivalence classes and delta-discriminative emerging patterns. In *Proc. 13th ACM SIGKDD Conf. Knowledge, Discovery and Data Mining*, ACM, 430–439.

RHIENMORA, P., HADDAWY, P., SUEBNUKARN, S., AND DAILEY, M. N. 2011. Intelligent dental training simulator with objective skill assessment and feedback. *Artif Intell Med 52*, 2, 115–121.

ROSEN, J., HANNAFORD, B., RICHARDS, C. G., AND SINANAN, M. N. 2001. Markov modeling of minimally invasive surgery based on tool/tissue interaction and force/torque signatures for evaluating surgical skills. *IEEE T Bio-Med Eng 48*, 579–591.

SEWELL, C., MORRIS, D., BLEVINS, N., BARBAGLI, F., AND SALISBURY, K. 2005. Quantifying risky behavior in surgical simulation. *St Heal T 111*, 451–457.

SEWELL, C., MORRIS, D., BLEVINS, N., DUTTA, S., AGRAWAL, S., BARBAGLI, F., AND SALISBURY, K. 2008. Providing metrics and performance feedback in a surgical simulator. *Comput Aided Surg 13*, 2, 63–81.

STYLOPOULOS, N., COTIN, S., MAITHEL, S., OTTENSMEYER, M., JACKSON, P., BARDSLEY, R., NEUMANN, P., RATTNER, D., AND DAWSON, S. 2004. Computer-enhanced laparoscopic training system (CELTS): bridging the gap. *Surg Endosc 18*, 5, 782–789.

WANG, L., GU, T., TAO, X., AND LU, J. 2012. A hierarchical approach to real-time activity recognition in body sensor networks. *Pervasive and Mobile Computing 8*, 1, 115–130.

WITTEN, I. H., AND FRANK, E. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

ZHOU, Y., BAILEY, J., IOANNOU, I., WIJEWICKREMA, S., O'LEARY, S., AND KENNEDY, G. 2013. Constructive real time feedback for a temporal bone simulator. In *Proc. MICCAI'13*.