Provision of Automated Step-by-Step Procedural Guidance in Virtual Reality Surgery Simulation

Sudanthi Wijewickrema*
Department of Surgery (Otolaryngology),
The University of Melbourne, Australia

Yun Zhou
Department of Surgery (Otolaryngology),
The University of Melbourne, Australia

James Bailey
Department of Computing and Information Systems,
The University of Melbourne, Australia

Gregor Kennedy
Center for the Study of Higher Education,
The University of Melbourne, Australia

Stephen O'Leary
Department of Surgery (Otolaryngology),
The University of Melbourne, Australia

Abstract

One of the roadblocks to the wide-spread use of virtual reality simulation as a surgical training platform is the need for expert supervision during training to ensure proper skill acquisition. To fully utilize the capacity of virtual reality in surgical training, it is imperative that the guidance process is automated. In this paper, we discuss a method of providing one aspect of performance guidance: advice on the steps of a surgery or procedural guidance. We manually segment the surgical trajectory of an expert surgeon into steps and present them one at a time to guide trainees through a surgical procedure. We show, using a randomized controlled trial, that this form of guidance is effective in moving trainee behavior towards an expert ideal.

To support practice variation and different surgical styles adopted by experts, separate guidance templates have to be generated. To enable this, we introduce a method of automatically segmenting a surgical trajectory into steps. We propose a pre-processing step that uses domain knowledge specific to our application to reduce the solution space. We show how this can be incorporated into existing trajectory segmentation methods, as well as a greedy approach that we propose. We compare this segmentation method to existing techniques and show that it is accurate and efficient.

Keywords: Virtual Reality; Surgery Simulation; Automated Guidance

Concepts: •**Applied computing** → **Interactive learning environments; Computer-managed instruction;** *Health care information systems;*

1 Introduction

Virtual reality (VR) training environments are gaining popularity as training tools for skills development in a number of professions. They are particularly useful in fields such as surgery, where training resources are limited, participant numbers are high, and failure could be catastrophic. In simulation based surgical training, performance feedback has been identified as an integral component [Stefanidis 2010]. However, the lack of automated feedback has thus far necessitated the need for expert supervision during training, impeding the more wide-spread use of VR simulation as a self-directed platform for surgical training.

Most existing automated feedback systems provide summative feedback at the end of a procedure [Mackel et al. 2006; Sewell et al. 2008]. Although this is a valid form of feedback that promotes skill acquisition, it does not fully emulate the advice expert surgeons provide. Some recent works have addressed the issue of providing real-time guidance on technical performance to overcome this drawback [Rhienmora et al. 2011; Wijewickrema et al. 2015]. However, motor (technical) skills are just one aspect of the skills that have to be mastered to achieve expertise in surgery. Thus, procedural guidance (where to drill and when) should also be provided.

'Path following', where visual cues guide the trainee on the steps pertaining to a surgical procedure, has been identified as a way of providing procedural guidance. Visual guidance of this form was presented to the trainee in a virtual laparoscopic simulator using a spline that indicated the path to follow [Passmore et al. 2001]. Botden et al. provided a similar form of guidance using visual cues such as arrows to train a suturing task [Botden et al. 2009]. Rhienmora et al. presented the trainee with a ghost drill they had to follow in a dental surgery simulation [Rhienmora et al. 2011]. Although this is more appropriate for open surgery, it does not allow the trainees to drill at their own pace. Step-by-step guidance is an alternative presentation of procedural guidance that overcomes this limitation.

One way to provide step-by-step guidance of this form is to segment procedures performed by expert surgeons and present it to the trainee one step at a time. However, different surgeons have different styles of surgery, often reflected by how they handle the drill and the sequence in which they perform non-critical steps of a procedure. Therefore, it is important that trainees are given the opportunity to learn operations based on a 'template' of the style of their preferred expert. Furthermore, as practice variability is an important aspect of gaining expertise [Stefanidis 2010], guidance should be provided on how to perform a procedure on different specimens. However, due to anatomical variations of specimens, a procedure performed on one specimen cannot be used as a template for another. Thus, it is critical that the trajectory of an expert procedure is segmented automatically to develop guidance templates.

Segmentation of trajectories is essentially a problem of unsupervised clustering of sequential data. Some of the existing literature on unsupervised clustering are application-oriented [Yan et al. 2011; Yoon and Shahabi 2008]. These methods typically use do-

*e-mail:swijewickrem@unimelb.edu.au

VRST '16,, November 02-04, 2016, Garching bei München, Germany
ISBN: 978-1-4503-4491-3/16/11
DOI: http://dx.doi.org/10.1145/2993369.2993397

main knowledge to obtain optimal results, and as such are difficult to apply to other application domains. General segmentation methods, which do not utilize domain-specific knowledge, usually optimize a criterion function to generate segments of a trajectory [Anagnostopoulos et al. 2006; Leiva and Vidal 2013]. These algorithms are generally applicable to any domain, and can more easily be extended to incorporate domain-specific knowledge if required.

Here, we propose a presentation technique for procedural guidance that highlights areas to be drilled one at a time. We show through a user trial that this form of procedural guidance can be successfully used in training a surgical procedure. We further introduce an automatic trajectory segmentation method that removes the need for manual identification of steps. This work was based on a haptic enabled VR temporal bone (ear) surgery simulator (see figure 1).

Figure 1: *The VR temporal bone surgery simulator*

2 Presentation of Procedural Guidance

2.1 Methods

The procedure under consideration for the provision of procedural guidance was a simple temporal bone surgery: cortical mastoidectomy. A surgical trajectory of an expert surgeon performing a cortical mastoidectomy was recorded by the simulator and manually segmented into logical steps. A set of points was considered to be a segment if 1) the points are clustered in the same region of the temporal bone, 2) the direction of movement of the trajectory is similar, 3) there are no large gaps between two consecutive drilled areas, and 4) the size of the cluster is appropriate for the area being drilled. The segmentation was performed manually and the resulting segments were approved by two expert surgeons. Algorithms were developed so that each step is highlighted on the temporal bone to indicate the area to be drilled and the next step is only shown once 75% or more of the current step has been drilled (see figure 2).

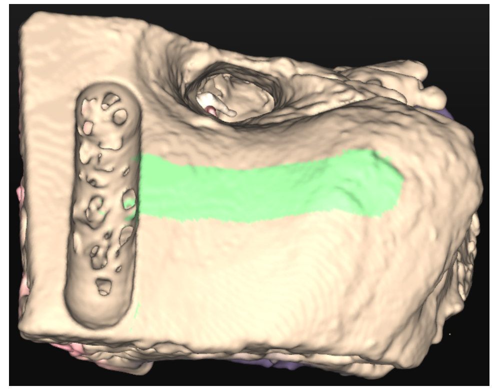

Figure 2: *Presentation of procedural guidance*

2.2 Experimental Results

To evaluate the effectiveness of this form of procedural guidance, we conducted a randomized controlled trial of 20 medical students performing cortical mastoidectomy on a VR simulator. Participants were first shown a video tutorial on how to perform the procedure and randomized into one of two groups: intervention and control. The intervention group received step-by-step procedural guidance (along with feedback on technical aspects of the procedure) during surgery while the control group did not. Both groups performed the operation twice. In the first run, the intervention group received procedural guidance throughout the procedure. In the second run, they were asked to turn the guidance on when they required it. The dissections were recorded by the simulator for all procedures.

The quality of the dissections were determined by a blinded expert surgeon in a post-experiment analysis using the validated Welling scale [Butler and Wiet 2007]. The quality of dissection is a discrete score in the range $[0, 35]$, with 35 being the best score. The scores between the two groups were analyzed for each run. An increase in both the mean and median differences in dissection quality was observed in the intervention group when compared to the control group. The differences between groups were determined to be significant through Kruskal-Wallis tests. A confidence interval of 95% was considered when testing for significance (see table 1).

Table 1: *Percentage increase in the dissection score of the intervention group when compared to the control group.*

Run	Mean Difference	Median Difference	Significance	Usage
1	72.09%	70.27%	$p < 0.001$	100.00%
2	50.00%	62.86%	$p = 0.002$	60.40%

3 Automatic Trajectory Segmentation

3.1 Methods

The algorithm discussed here comprises two steps. The first step incorporates domain-specific information to pre-process the surgical trajectory and reduce the solution space. The second step is a general trajectory segmentation algorithm that uses a greedy approach. These two algorithms can be used together or with other similar algorithms.

3.1.1 Pre-Processing using Domain Knowledge

Sampling: To lighten the computational burden of segmentation, the probability of consecutive data points that are close together being detected as decision points (in the next step) should be reduced. We overcome this by defining a sampling rate $s = \left\lceil \frac{n_T}{n_r} \right\rceil$, where, n_T is the number of points in the complete trajectory, and n_r is the recommended number of points for a given procedure. n_r is determined empirically as the number of points in the trajectory of an expert surgeon familiar with the simulator.

Identifying decision points: The methods used here are designed to mimic the reasoning used in manual segmentation of surgical trajectories (see section 2.1). First, if the distance between two points is larger than a threshold, the first of the two points is selected as a decision point. The distance threshold t_d is calculated as the p_d^{th} percentile of all the distance differences for a given trajectory.

Since trajectory segments should be in the same general direction, points at which there is a sharp change in direction should be considered as a decision point. To detect changes in direction, we use a k-metric based (k-cos) algorithm [Hall et al. 2008] with an angle threshold of t_{theta} and $k = 1..3$. In a surgical trajectory, these turning points are also typically associated with lower speed. To accommodate this, and to remove outliers detected by the turning point detection method, we only select those turning points at which the speed is less than a given threshold t_s, selected as the p_s^{th} percentile of speeds between all pairs of consecutive points in a trajectory.

These sets of points detected to preserve distance and direction in a cluster are combined to form the complete set of decision points. The use of percentiles ensures that the thresholds adapt to the characteristics of each trajectory to capture the nuances of different surgical styles. The percentile values p_d and p_s are chosen empirically.

Merging small segments: Even after the sampling, some decision points that are close together may be detected in the previous step. To avoid this, all segments smaller than a given number of points n_{small} are considered for merging. For these segments, the closest neighbor of its end points is considered. If the closest neighbor is at a distance less than a given inclusion threshold t_D, the two segments are combined. To ensure that t_D is adaptable to different styles of surgery, it is calculated as the p_D^{th} percentile of all distances between two consecutive points in the trajectory under consideration. This condition is used to avoid merging segments that are too far apart and thus should be separate regardless of size.

3.1.2 Segmentation Algorithm

The proposed trajectory segmentation algorithm uses a greedy approach to minimize the intra-cluster distance at each step. By considering only the decision points as possible points at which a split can occur, this algorithm not only decreases the computation time, but also reduces the probability of points that clearly cannot be split points being detected as local minima.

The algorithm uses pre-defined minimum and maximum segment sizes (s_{min} and s_{max} respectively) as constraints to perform trajectory segmentations. We recursively split the trajectory into two segments at a time. At each level, we find the possible decision points that would form a valid split. For a valid split to occur, the first segment resulting from the current split should have a number of points within the range $[s_{min}, s_{max}]$. The second segment should be at least s_{min} in size (as it could either be one segment or split into multiple segments in the subsequent steps).

For all possible split points D_{pos}, we calculate the sum of quadratic error (SQE) $J = \sum_{j=1}^{c} \sum_{x \in C_j} ||x - \mu_j||^2$, where, c is the number of clusters and μ_j is the mean of the j^{th} cluster C_j [Leiva and Vidal 2013]. Now we find the set of all possible split points D_{split} with the least values of SQE. The possible splits are calculated as those that are closest to the value of the minimum SQE. The 'closeness' threshold is defined as a fraction (i_p) of the difference between the minimum and maximum SQE values. Thus, the points D_{split} are the splits that minimize the SQE. For all these points, the first segment caused by the split is removed from the trajectory, and the above process is continued recursively until no splits can be found.

If no possible split points are available, it could mean one of two things. If the size of the trajectory at this level is within the given size range $[s_{min}, s_{max}]$, then the sequence of prior splits that led to the current level is valid. If not, this sequence of splits is not valid. As the lower limit of the range has already been checked when calculating the possible splits, only the upper limit has to be checked at this point. Thus, if the number of trajectory points $n \leq s_{max}$, the split sequence is accepted. If not, it is rejected.

The valid split sequences thus calculated are then tested globally (for the complete trajectory) to find the one with the best segmentation/clustering quality. The accuracy of a clustering was determined using the Davies-Bouldin (DB) index [Davies and Bouldin 1979]. The DB index is an internal evaluation scheme that is defined as a function of the ratio of the intra-cluster distance, to the inter-cluster distance. As such, a lower value of the DB index indicates a better clustering.

3.2 Experimental Results

Eight consulting ENT surgeons performed 17 procedures of cochlear implant surgery on one specimen in the VR temporal bone surgery simulator. Their surgical trajectories were saved by the simulator.

Parameter values that drive the pre-processing step were selected empirically, and through consultation with expert surgeons. The recommended number of trajectory points was selected as $n_r = 8000$. The k-cos angle threshold was set to $t_\theta = \frac{8\pi}{9}$. Speed and distance percentiles for the detection of decision points were selected to be $p_s = 5\%$ and $p_d = 95\%$ respectively. A distance percentile of $p_D = 99.9\%$ was used to avoid combining segments that are too far apart. The size of segments to be merged was set to $n_{small} = 5$. The minimum and maximum segment sizes were defined as $s_{min} = 80$ and $s_{max} = 1300$, based on manual segmentations. The experiments were conducted on a 2.5GHz computer with an 8-core processor and 16GB of RAM running Windows 7 Professional (64 bit). The algorithms were implemented and run on Matlab 2016a.

The proposed algorithm was compared with two existing segmentation algorithms: mutual nearest neighbour [Gowda and Krishna 1978] and warped k-means [Leiva and Vidal 2013]. The original algorithms that use all trajectory points (MNN-All and WKM-All), along with modified versions constrained to split segments only at decision points (MNN-Dec and WKM-Dec), were compared. The mean (and median) DB indices and processing time of the 5 algorithms on the 17 surgical procedures were calculated. Out of the algorithms tested, the proposed method performed best in terms of clustering/segmentation quality. With respect to processing time, it was only better than the original warped k-means algorithm (see table 2).

Table 2: *Clustering quality and processing times of the different methods. The best values are shown in bold.*

Method	DB Index		Processing Time (s)	
	Mean	Median	Mean	Median
Proposed	**2.09**	**1.94**	33.53	12.71
MNN-All	2.71	2.75	4.01	2.54
MNN-Dec	2.53	2.43	**0.30**	**0.30**
WKM-All	2.44	2.43	483.28	273.05
WKM-Dec	2.29	2.04	6.27	5.93

The results show that incorporating the pre-processing method to detect decision points into existing algorithms not only reduced their processing time, but also increased their accuracy. It indicates that the pre-processing algorithm was successful in detecting the most likely points for splitting the trajectory.

As trajectory segmentation can be performed off-line to create guidance templates, accuracy (or segmentation/clustering quality) is the more important factor in our application. Thus, we conducted a further analysis to compare the differences in segmentation quality. Kruskal-Wallis tests showed that the differences in the DB indices were significant at a 95% level of confidence (see table 3).

Table 3: *Decrease in the DB index of the proposed method when compared to each of the other methods*

Method	Mean Difference	Median Difference	Significance
MNN-All	25.96%	34.72%	$p < 0.001$
MNN-Dec	19.33%	22.32%	$p = 0.024$
WKM-All	15.79%	22.59%	$p = 0.001$
WKM-Dec	9.29%	5.17%	$p = 0.008$

4 Conclusions

In this paper we introduced a method of presenting automated step-by-step procedural guidance to trainees performing surgery on virtual reality simulation platforms. We showed using a randomized controlled trial of medical students performing a temporal bone surgical procedure, that this form of guidance is effective in improving trainee behavior.

We also proposed an algorithm for automatically segmenting a surgical trajectory to develop guidance templates using which procedural guidance can be provided. This is important because different guidance templates are required to reflect different surgical styles and to support practice variation. This process consists of two stages: pre-processing and trajectory segmentation, which can be used together or in combination with other relevant methods.

Experimental results showed that incorporating the pre-processing algorithm into trajectory segmentation methods not only reduced their processing time, but improved their segmentation quality as well. This is due to the reduction in the solution space to the most likely points at which the trajectory should be segmented, reducing the probability of local minima being detected as split points. Results further showed that the proposed trajectory segmentation algorithm performed well with respect to segmentation quality. This algorithm has the added advantage of being able to detect segments within a given size range. Although the processing time was higher than most methods it was compared with, it was still low (about 33 seconds on average). In our application, this is acceptable as trajectory segmentation can be performed off-line, and as such processing time is a secondary concern to the segmentation quality.

Note that the user study discussed here only tested for the effectiveness of the guidance technique in changing user behavior and did not test for retention of knowledge. In future work, further studies will be conducted to investigate how skills learned through automated guidance is retained. Expert data will be collected on different specimens and the methods discussed will be applied to develop guidance templates for those as well. The segmentations thus generated will be qualitatively evaluated by expert surgeons.

Acknowledgements

This work was funded by an Australian Research Council Linkage grant in collaboration with Cochlear Ltd.

References

ANAGNOSTOPOULOS, A., VLACHOS, M., HADJIELEFTHERIOU, M., KEOGH, E., AND YU, P. S. 2006. Global distance-based segmentation of trajectories. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 34–43.

BOTDEN, S. M., DE HINGH, I. H., AND JAKIMOWICZ, J. J. 2009. Meaningful assessment method for laparoscopic suturing training in augmented reality. *Surgical endoscopy 23*, 10, 2221–2228.

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.

DAVIES, D. L., AND BOULDIN, D. W. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, 2, 224–227.

GOWDA, K. C., AND KRISHNA, G. 1978. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition 10*, 2, 105–112.

HALL, R., RATHOD, H., MAIORCA, M., IOANNOU, I., KAZMIERCZAK, E., OLEARY, S., AND HARRIS, P. 2008. Towards haptic performance analysis using k-metrics. In *Haptic and Audio Interaction Design*. Springer, 50–59.

LEIVA, L. A., AND VIDAL, E. 2013. Warped k-means: An algorithm to cluster sequentially-distributed data. *Information Sciences 237*, 196–210.

MACKEL, T., ROSEN, J., AND PUGH, C. 2006. Data mining of the e-pelvis simulator database: a quest for a generalized algorithm for objectively assessing medical skill. *Studies in Health Technology and Informatics 119*, 355–360.

PASSMORE, P. J., NIELSEN, C. F., COSH, W., AND DARZI, A. 2001. Effects of viewing and orientation on path following in a medical teleoperation environment. In *Virtual Reality, 2001. Proceedings. IEEE*, 209–215.

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

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

STEFANIDIS, D. 2010. Optimal acquisition and assessment of proficiency on simulators in surgery. *Surgical Clinics of North America 90*, 3, 475–489.

WIJEWICKREMA, S., PIROMCHAI, P., ZHOU, Y., IOANNOU, I., BAILEY, J., KENNEDY, G., AND OLEARY, S. 2015. Developing effective automated feedback in temporal bone surgery simulation. *Otolaryngology–Head and Neck Surgery*, 0194599815570880.

YAN, Z., GIATRAKOS, N., KATSIKAROS, V., PELEKIS, N., AND THEODORIDIS, Y. 2011. Setrastream: semantic-aware trajectory construction over streaming movement data. In *International Symposium on Spatial and Temporal Databases*, Springer, 367–385.

YOON, H., AND SHAHABI, C. 2008. Robust time-referenced segmentation of moving object trajectories. In *2008 Eighth IEEE International Conference on Data Mining*, IEEE, 1121–1126.