Automated segmentation of surgical motion for performance analysis and feedback

Yun Zhou[1], Ioanna Ioannou[2], Sudanthi Wijewickrema[2] James Bailey[1], Gregor Kennedy[3], and Stephen O'Leary[2]

[1] Department of Computing and Information Systems, University of Melbourne
[2] Department of Otolaryngology, University of Melbourne
[3] Centre for the Study of Higher Education, University of Melbourne

Abstract. Advances in technology have motivated the increasing use of virtual reality simulation-based training systems in surgical education, as well as the use of motion capture systems to record surgical performance. These systems have the ability to collect large volumes of trajectory data. The capability to analyse motion data in a meaningful manner is valuable in characterising and evaluating the quality of surgical technique, and in facilitating the development of intelligent self-guided training systems with automated performance feedback. To this end, we propose an automatic trajectory segmentation technique, which divides surgical tool trajectories into their component movements according to spatio-temporal features. We evaluate this technique on two different temporal bone surgery tasks requiring the use of distinct surgical techniques and show that the proposed approach achieves higher accuracy compared to an existing method.

Keywords: motion analysis, surgical simulation

1 Introduction

Interest in the analysis of surgical motion has flourished in recent years with the development and use of an ever expanding range of motion capture technologies. Virtual reality (VR) simulators are becoming an increasingly important component of surgical training programs, and the use of motorised haptic devices in these simulators readily enables the capture of surgical motion. Efforts have also been made to record surgical motion in the operating theatre, as increasingly accurate and unobstructive sensors become available.

Consequently, the analysis of surgical motion has become an active field of research. In the surgical domains that involve drilling - such as temporal bone surgery, orthopedic surgery and dental surgery - good technique is encompassed in the way a surgeon utilises their drill to remove tissue. Expert surgeons often describe good technique by delineating the desired characteristics of drilling strokes. For example, in a mastoidectomy procedure the use of long strokes parallel to sensitive anatomical structures is considered good technique.

The aim of this paper is to develop an automated method of segmenting drilling trajectories into a sequence of clinically meaningful component motions, which we refer to as strokes. The characteristics of these strokes (e.g. distance, shape, applied force) can be analysed to quantify the differences between expert and trainee surgeons, evaluate the quality of their surgical technique, and even provide automated feedback during training. Trajectory segmentation using spatio-temporal features has been studied widely in other application areas, such as handwriting recognition [1, 8] and geographic information systems [2].

In this work, we propose a classification approach based on spatio-temporal features to automatically segment surgical drilling trajectories. We begin with a description of the dataset used to train and evaluate the classifiers, followed by the steps of the proposed approach and the chosen set of spatio-temporal features. We proceed to define the baseline method and our classification-based method. Finally we define the evaluation metrics used to measure the quality of the segmentation, followed by the results of the evaluation.

2 Method

We begin with a formal definition of trajectory and stroke. The trajectory of a moving tool is defined as a sequence of pairs, $\tau = [(p_1, t_1), (p_2, t_2), ..., (p_n, t_n)]$, where p_i is a three-dimensional vector representing the position observed at time t_i, $i \in [1, n]$ and n is the number of data points in the trajectory. We denote a stroke from time t_i to time t_j as a sequence of points in τ: $s = \tau[t_i, t_j]$.

Experiment data: The trajectory data used to build segmentation models in this experiment was collected on a VR temporal bone surgery simulator. The data consisted of 16 expert and 10 trainee performances conducted by 7 experts and 6 trainees. Each performance included the full preparation of the temporal bone for cochlear implantation. We focussed our investigation on two stages of this procedure that require very different drilling technique, namely mastoidectomy and posterior tympanotomy. Mastoidectomy is the initial stage and typically requires long sweeping strokes with a large burr, while posterior tympanotomy is carried out in a very tight space and requires short, often more circular strokes, with a small burr.

For each performance, we randomly selected a 10 second sub-trajectory from each stage and labelled it manually (example shown in figure 1). Table 1 summarizes the statistics of our dataset. The ratio of turning points to normal points for the two stages was approximately $1:10$ and $1:8$. We randomly split the 26 labelled performances into three sets: a training set of 18, a validation set of 4, and a test set of 4. The training set was used to train the model, the validation set was used for parameter optimisation, and the test set was used to validate segmentation performance.

In order to further test our segmentation models, we also manually labelled a random 10 second sub-trajectory from each stage of six cadaveric temporal bones. The details of the data collection procedure for cadaveric temporal bones can be found in [7].

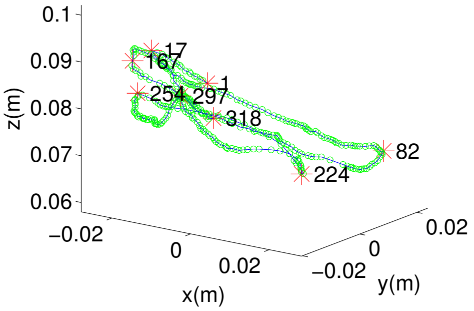


Fig. 1: Example of a manually segmented sub-trajectory from a mastoidectomy procedure containing 7 strokes. Units are in metres. Red crosses mark turning points and green dots mark normal points.

Table 1: Statistics of labelled trajectory dataset

dataset	**task**	**# turning points**	**# normal points**	**# strokes**
Simulator	mastoidectomy	786	7501	764
	posterior tympanotomy	936	7661	911
Cadaveric	mastoidectomy	7	211	6
	posterior tympanotomy	29	248	28

Proposed method: Figure 2 provides an overview of the four steps that comprise the proposed automatic trajectory segmentation method. As a first step, noise and irrelevant points (such as non-drilling points) are filtered out of the trajectory, since these points do not reflect the true spatio-temporal characteristics of drilling technique. For the purposes of this paper, points separated by a Euclidean distance of less than 0.25 mm from their neighbours were considered noise, as they are more likely to be generated by limitations in the position sensing apparatus rather than intentional human motion.

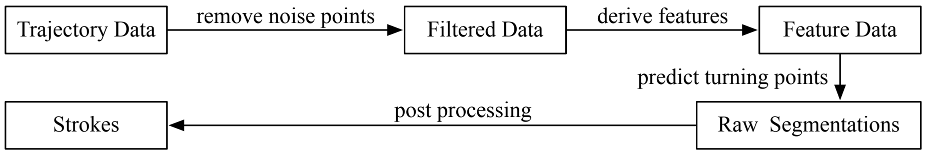


Fig. 2: Overview of trajectory segmentation steps

The second step is to derive the spatio-temporal features of each point. The features of point p_i are denoted as $\phi(p_i)$. These features are derived from the current point p_i and two other points p_{i-k} and p_{i+k}, which are k points before and after p_i. We choose k equal to 3 in our experiment. Spatio-temporal features are expected to be fairly uniform during a stroke, but change significantly at its end points, when there is a change in direction. The features below were chosen for their ability to capture a variety of different strokes encountered in drilling, such as sharp turns and smooth, circular turns. We illustrate each feature using examples only in the X and Y axes for convenience, but all features were in fact derived using all three axes for our experiments.

- Speed [4]: the velocity (in x, y, z directions) and speed magnitude at point p_t. Turning points typically feature lower speeds than normal points.

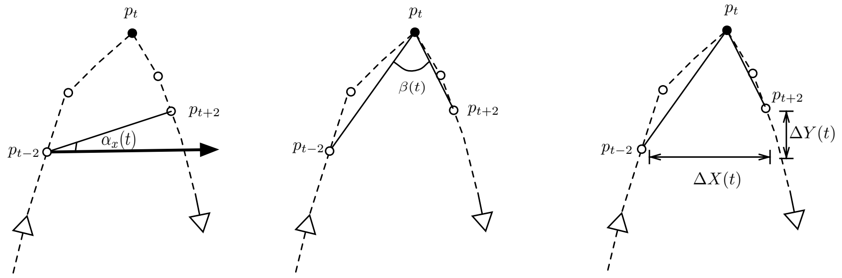
- Direction: Figure 3a illustrates the angle $\alpha_x(t)$ between the line segment $\overline{p_{t-k}p_{t+k}}$ and the x-axis. Angle $\alpha_x(t)$ is smaller for turning points compared to normal points. The sine and cosine values of $\alpha_x(t)$ are derived as two features to capture the direction of p_t with respect to the x-axis. The sine and cosine values of $\alpha_y(t)$ and $\alpha_z(t)$ are also calculated with respect to the y-axis and z-axis respectively, using the same approach.

- Bow [6]: The bow of p_t is represented by the cosine of the angle $\beta(t)$ between the line segments $\overline{p_{t-k}p_t}$ and $\overline{p_t p_{t+k}}$, as shown in figure 3b. The cosine value of $\beta(t)$ is derived using vector inner product. The bow of a sharp turning point is usually larger than that of a normal point.

- Curvature: This value is defined as the ratio between the angle $\beta(t)$ and the sum of the lengths of its line segments: $\frac{\beta(t)}{|\overrightarrow{p_{t-k}p_t}|+|\overrightarrow{p_t p_{t+k}}|}$. This feature considers both the angle between the two line segments as well as their length to enable the capture of circular turning points, which usually have similar bow to normal points, but lower curvature.

- Vicinity aspect [9]: This feature captures the incremental change in position between points p_{t-k} and p_{t+k} over two axes, as shown in Figure 3c. Vicinity aspect is the ratio of the change in position across the x-axis and y-axis, defined as $VA(X,Y) = \frac{\Delta X(t) - \Delta Y(t)}{\Delta X(t) + \Delta Y(t)}$. $VA(X,Z)$ and $VA(Y,Z)$ are calculated using the same method. Vicinity aspect remains fairly constant for normal points, while it increases or decreases for turning points, depending on symmetry and turn direction.

The third step of our approach uses the above features as input to derive a trajectory segmentation model. Most previous frameworks carry out this task using a variety of predefined thresholds for spatio-temporal criteria [2, 6]. They pick a start time t_i and examine the subsequent data points until they find the longest sub-trajectory $\tau[t_i, t_j]$ that satisfies the predefined criteria. Then the end point of the stroke is regarded as the start point of the next stroke and the above process is repeated to the end of the trajectory. However, it is difficult to derive pre-defined criteria that will detect all types of strokes encountered during surgery. For example, in temporal bone surgery, surgeons tend to start with long fast strokes to efficiently remove bone that is far from sensitive anatomical structures, but switch to a more cautious technique as they approach structures



(a) Direction (b) Bow & curvature (c) Vicinity aspect

Fig. 3: Extracted features for a trajectory using $k = 2$

such as the facial nerve. Predefined criteria are unlikely to be equally effective at detecting both types of stroke. Our approach treats trajectory segmentation as a supervised learning problem, whereby a functional mapping is derived from the value of the features $\phi(p_i)$ to two labels: a turning point and a normal point. This process is described in detail later in this section.

Once points are labelled, stroke start and end points are derived automatically from the labels. Since there are far more normal points than turning points in a typical trajectory, it is very unlikely that a set of consecutive points are all turning points. Hence, the fourth step of the process smooths stroke prediction by examining consecutive turning points, identifying the point with the maximum probability of being a turning point, and changing the other consecutive turning points to normal points. The proposed classification approach was compared to an existing method; both are described below.

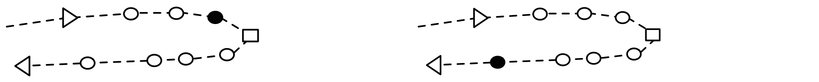
Baseline: Previous work used bow to detect turning points based on the knowledge that the bow of points inside a stroke is typically different to that of turning points [10]. However, the threshold of bow that denotes a turning point is unpredictable, due to the great variety of stroke techniques encountered during bone drilling. This implementation assumed that bow follows normal distribution, so a threshold $ST = \mu + i \times \sigma$ for each trajectory is derived, where $i \in 1, 2$. The model computes the bow of a point p_i and compares it to ST. If the value is larger than ST, it is classified as a turning point, otherwise the point p_i is regarded as a normal point. If a list of consecutive turning points is encountered, the point with the minimum speed is chosen as the turning point.

Classification-based segmentation: Instead of using bow alone, supervised learning uses several features to perform turning point prediction. A point's label is not only dependent upon its own feature values, but those of its neighbouring points as well. Hence, we concatenate the features $\phi(p_i)$ of point p_i with those of the l nearest neighbour points. We formally define this operation as $concatenate(\phi(p_i), l) = [\phi(p_{i-l}), ..., \phi(p_i), ..., \phi(p_{i+l})]$. The length of concatenated features is $(2 \times l + 1) \times \#\phi(p_i)$, where $\#\phi(p_i)$ is the number of features for point p_i. l is usually a small number, since the concatenation operation increases feature size significantly. For the purposes of this work, we used $l = 3$ and $\#\phi(p_i) = 18$, which resulted in 126 features after concatenation.

The next task was to choose an appropriate classifier. Upon examination of the dataset (table 1), it was evident that the turning point class is a minor class (having a smaller number of instances than the other class), therefore the choice of classifier had to take into account the imbalance of the dataset. Since most tree-based classifiers are biased towards the major classes (which have a larger number of instances), we tried only Nearest Neighbour(NN), Linear Discriminant Analysis (LDA) and Naive Bayes(NB) classifiers. Preliminary results showed that LDA achieved acceptable turning point prediction accuracy, while other classifiers tended to ignore the turning point class. In addition, we experimented with kernel-based discriminant analysis (KDA) [3], which performed slightly better than LDA. However, parameter optimisation for KDA is far more time consuming than LDA. Therefore, we chose LDA as our classifier.

LDA estimates the prior probability of each class (i.e. $P(turning) = \frac{\#turning}{\#total}$) based on its frequency in the dataset. For an imbalanced dataset, this estimation skews the prediction towards the majority class. Since missing a turning point is a major error in trajectory segmentation, we treated this prior probability as a parameter and varied it to maximise the recall of turning point class prediction. However, high recall may be a result of more false positive predictions. Usually, a point is regarded as a turning point if the posterior probability of a point belonging to the turning class is larger than 50%. Since the prior probability affects the posterior probability, we treated the threshold of posterior probability as another parameter. These two parameters were tuned to achieve optimal balance between high recall and low false positive predictions, therefore providing the most accurate classification.

3 Experiment results



(a) Minor error: predicted turning point is 1 point away from ground truth

(b) Major error: predicted turning point is 4 points away from ground truth

Fig. 4: Example of two classifications representing a minor error and a major error. Filled black dots represent predicted turning points, white squares represent ground truth turning points and empty white dots represent normal points. Precision and recall are zero in both cases.

Evaluation measures: Precision and recall are often used as measures of classification performance on imbalanced data sets [5]. The precision and recall of turning points was computed as $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$ where TP, FP, and FN represent the number of true positives, false positives, and false negatives respectively. However, these measures do not completely capture our goal of segmenting the trajectory such that the detected strokes are as similar to the ground truth as possible. Figure 4 illustrates the reason. Both cases have the same recall and precision, but figure 4a is obviously a better result.

To address this limitation, we define a new performance measure to capture the matching percentage between a ground truth stroke s_i and a classified stroke s_j as $match(s_i, s_j) = \frac{|\cap(s_i,s_j)|}{max(|s_i|,|s_j|)}$, where $|s_*|$ denotes the number of points in each stroke and $\cap$ denotes the overlapping part of the two strokes. For each classified stroke, we used the Kuhn-Munkres algorithm to find the best corresponding ground truth stroke, such that the average match rate for each surgical performance is maximized.

Segmentation results: Table 2 shows the performance achieved by LDA models compared to the baseline. The match rate of the LDA approach was significantly better than the baseline for both surgical tasks. For simulation data, LDA achieved an improvement of 17.2% for mastoidectomy and 10.7% for posterior tympanotomy. The improvement in match rate was greater in mastoidectomy than posterior tympanotomy, but LDA achieved dramatically better precision and recall in the latter, while these measures remained similar in mastoidectomy. For cadaveric data, LDA achieved an improvement of 62.25% for mastoidectomy and 16.58% for posterior tympanotomy in match rate. LDA also achieved dramatically better precision in mastoidectomy.

Since match rate is a better indicator of accuracy, we will focus on that measure. The higher match rate observed in mastoidectomy indicates that the longer strokes with sharper turning points were classified more precisely, which is to be expected. When the change in direction is small (as in the case of the circular strokes used during posterior tympanotomy), LDA models may produce false negatives. Posterior tympanotomy also includes more short, jittery movements, and it is not always clear whether these represent genuine surgical motion or noise. In this case, the LDA classifier may produce false positives. Many strokes in the posterior tympanotomy stage do not have clearly defined turning points, which makes even manual segmentation challenging and subjective. Therefore, the difference in match rate between the two stages may be a result of genuine ambiguity.

4 Discussion and conclusion

We have presented an automated method for segmenting drilling-based surgical motion into its component drill strokes. This technique was validated on two temporal bone surgery tasks and shown to achieve acceptable accuracy despite encountering a great variety of surgical strokes. In the future, we may investigate

Table 2: Segmentation performance

		mastoidectomy			posterior tympanotomy		
dataset	**approach**	**precision**	**recall**	**match rate**	**precision**	**recall**	**match rate**
Simulator	Baseline	0.58	0.64	62.25%	0.17	0.13	57.24%
	LDA	0.66	0.62	79.44%	0.51	0.57	67.93%
Cadaveric	Baseline	0.5	0.42	20.7%	0.57	0.37	34.77%
	LDA	1	0.28	82.95%	0.33	0.34	51.35%

the use of semi-supervised learning or Hidden Markov Models to further improve classification accuracy.

The ability to accurately segment a long surgical tool trajectory into smaller motions that are surgically meaningful is highly beneficial in facilitating the analysis of surgical technique in a variety of situations, ranging from simulation-based training to the operating theatre. The detailed characteristics of good surgical technique can be objectively quantified, and this understanding can be built into intelligent surgical training and guidance systems that guide surgeons towards optimal performance.

References

1. Bengio, Y., LeCun, Y., Nohl, C.R., Burges, C.J.C.: Lerec: a nn/hmm hybrid for on-line handwriting recognition. Neural Computation 7(6), 1289–1303 (1995)
2. Buchin, M., Driemel, A., van Kreveld, M., Sacristán, V.: An algorithmic framework for segmenting trajectories based on spatio-temporal criteria. In: Advances in Geographic Information Systems. pp. 202–211. ACM (2010)
3. Cai, D., He, X., Han, J.: Speed up kernel discriminant analysis. The VLDB Journal 20(1), 21–33 (2011)
4. Forestier, G., Lalys, F., Riffaud, L., Trelhu, B., Jannin, P.: Classification of surgical processes using dynamic time warping. J Biomed Inform 45(2), 255–264 (2012)
5. Gu, Q., Zhu, L., Cai, Z.: Evaluation measures of the classification performance of imbalanced data sets. In: Computational Intelligence and Intelligent Systems, pp. 461–471. Springer (2009)
6. Hall, R., Rathod, H., Maiorca, M., Ioannou, I., Kazmierczak, E., O'Leary, S., Harris, P.: Towards haptic performance analysis using k-metrics. In: HAID. pp. 50–59 (2008)
7. Ioannou, I., Avery, A., Zhou, Y., Szudek, J., Kennedy, G., O'Leary, S.: The effect of fidelity: How expert behavior changes in a virtual reality environment. The Laryngoscope 124(9), 2144–2150 (2014)
8. Izadi, S., Haji, M., Suen, C.Y.: A new segmentation algorithm for online handwritten word recognition in persian script. In: Frontiers in Handwriting Recognition. pp. 598–603 (2008)
9. Sanna, M., Khrais, T.: Temporal Bone: A Manual for Dissection and Surgical Approaches. Thieme (2011)
10. Wijewickrema, S., Ioannou, I., Zhou, Y., Piromchai, P., Bailey, J., Kennedy, G., O'Leary, S.: A temporal bone surgery simulator with real-time feedback for surgical training. NextMed/MMVR21 196, 462 (2014)