Machine learning based surgical trajectory segmentation for robotic surgery

By

Zhili, Yuan

MSc Robotics Dissertation

Department of Engineering Mathematics
UNIVERSITY OF BRISTOL
&
Department of Engineering Design and Mathematics
UNIVERSITY OF THE WEST OF ENGLAND

A MSc dissertation submitted to the University of Bristol and the University of the West of England in accordance with the requirements of the degree of MASTER OF SCIENCE IN ROBOTICS in the Faculty of Engineering.

September 12, 2022

Declaration of own work

I declare that the work in this MSc dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Zhili Yuan

September 12, 2022

Acknowledgement

Words cannot explain how grateful I am to my supervisor Dandan Zhang, for her tremendous patience and feedback. And I certainly could not have gone on this path without the help of her sharing knowledge and skills. The spirit of her diligent exploration and study will be the role model I have been studying in the future. I owe my gratitude to my parents as well. They provide solid support and unselfish love for me.

Abstract

Robot-assisted minimally invasive surgery is on the rise. A large number of videos and kinematic recordings of surgical operations have been collected from professional surgeons using surgical robots, containing visual and motion data. In surgical workflow analysis, segmentation of trajectory into multiple meaningful gestures is a vital and provisional step to facilitate learning from demonstrations, competence assessment, and other robot-assist surgical tasks. In this work, an unsupervised segmentation algorithm with a transformer-based network is combined to achieve automatic task segmentation utilizing both video and kinematics data. Specifically, surgical tasks are first segmented by combining the spatio-temporal and variance features of kinematics data. Then using visual features extracted from a pretrained 'Resnet' backbone followed by the encoding architecture of the Transformer network, a hierarchical structure is built combing both kinematics and visual features. The suggested approach was assessed using data from the publicly accessible JIGSAWS database for the suturing job. An average F1 score of 0.634 for the segmentation metric and an accuracy of 0.813 for the recognition metric were achieved.

Number of words in the dissertation: 8067 words.

Contents

List of Tables

1 Introduction

Rapidly development in the robotic-assisted system and machine learning algorithms leads to substantial growth in the field of robot-assisted surgery. Robotic surgery aims to lessen the workload of a surgeon and perform a more precise operation under probably a more safe environment. Examples of surgical robots involves in the field of neurosurgery (e.g. NeuroMate), biopsy (e.g. PAKY-RCM) [1], orthopaedic surgery (e.g. RoBoDoc [2] and Arthrobot [3]), minimally invasive surgery etc, [4]. Different efforts have been made into robotic-assisted surgery, from hardware implementation and integration to software programming. An open platform like the da Vinci Research Tool Kit (dVRK) integrated into the simulation system enables researchers to collect useful data and test novel algorithms such as control strategies, path planning and gesture recognition tasks to meet medical needs.

Surgical robots are widely used to support minimally invasive surgery. Traditionally, the patient always gets a large scar due to the large incision during the procedure, but the advance in surgical robot nowadays helps reduce the scars. Other benefits brought by surgical robots are shorter hospitalization, reduced blood loss and transfusions that result in reduced risk of infection. Those are realized by current explosion technologies of machine learning including deep learning algorithms developed for surgical robots, that could enhance dexterity, have greater precision and reduce substantial harm by providing safe surgical care etc,

Segmenting a task into gestures can provide detailed feedback for example when and where unexpected cuts, minor injuries, and reasons for resulting long procedure times happened, not just an overall 'successor 'failure' feedback. More widely, the gesture recognition and assessment result could also be used as an adaptation law for future collaborative work, in which the robot can be switched between a leader or a follower to make the trainee focus more on the certain gestures he/she had errors on [5]. Thus, finding the accurate segment point is important.

Recent development in the transformers network shows enormous potential for sequential modelling in Natural Language Processing (NLP), and can be used in surgical gesture analysis. Trans-

formers have the capacity to establish temporal links between the present and past frames utilising self-attention and achieve the best recognition results among other deep learning methods [6] [7]. However, nowadays, the proposed transformer-based technique can not identify the start and stop of each gesture, i.e. realize temporal segmentation. Thus, this paper presents an unsupervised algorithm with the transformer-based network to find the segmentation points by leveraging visual and kinematics data features.

Motivated by the benefits stated above, the aim and objectives of this paper are listed below:

1. **Aims:** By decomposing complicated surgical tasks into multiple gestures accurately, a robot-assist surgical system could automatically label the surgical motion data according to the activities being performed by a surgeon.

2. **Objectives:** To successfully segment surgical tasks and recognize surgical gestures, the following objectives will be achieved:

    - With an emphasis on current data-driven techniques, this paper covers the state-of-the-art in methods for automatically recognising fine-grained gestures in robotic surgery and identifies outstanding problems and potential future research avenues. Based on their advantages and drawbacks, this project will investigate new methods or integrate the advantages of different methods.
    - Unsupervised methods are selected for segmentation tasks to identify exact segmentation points to get rid of dataset labelling.
    - Spatial and variance characters are used to extract and observed pre-segmentation points for kinematics data.
    - Transformer-based neural network is used to extract visual features from surgical video data to enhance the segmentation result. Then, the pre-segment points are identified based on Dirichlet Process Gaussian Mixture Model (DP-GMM) clustering over all visual feature data.
    - In order to analyse both outcomes in a meaningful way, different criteria based on recall and precision are used for evaluating segmentation and classification results.

This thesis is organized as follows. The next section listed some of the state-of-art methods used for surgical gesture segmentation and recognition. Section 3 introduces the theory of the proposed method. The experiments set up including parameters specifications are in Section 4. Section 5 presents the results of all sub-experiments including critical points and pre-segmentation points found with visual and kinematics data separately. The last section gives a summary of the whole paper and discusses future works.

2 Literature Review

Within all research fields, surgical gesture segmentation and recognition is a fundamental and provision step to enable surgical robots to learn complicated movements, realise automatic and ensure dexterity. The segmentation and recognition results of each gesture could be further used in online applications, where context awareness is essential. [8]. Nowadays, gestures recognition are used in many complex surgery procedure such as laparoscopy[9], cardiac surgery[10] and soft tissue surgery[11]. This context-aware assistance improves the safety and quality during the surgery procedure.

As in [12] [13], a surgical workflow could be decomposed in a hierarchical manner, as shown in Figure 2.1. Thus, segment gesture is well suited for task optimization.

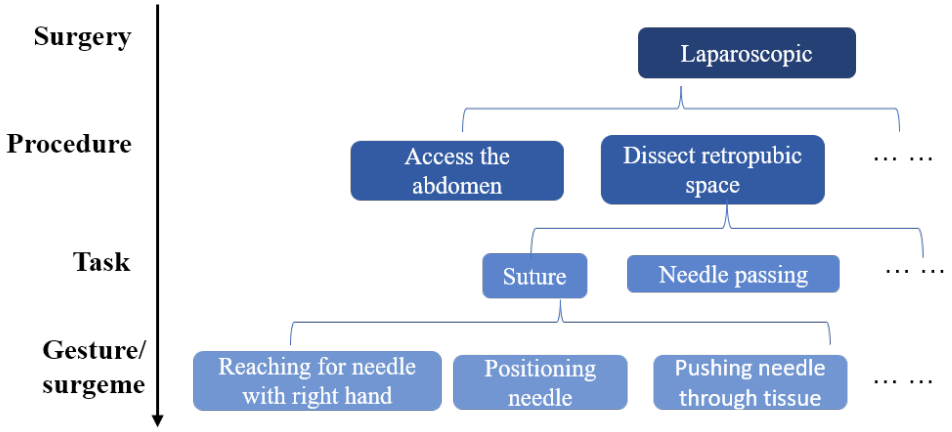

Figure 2.1: Surgery hierarchical decomposition (adopt from [14])

Many gesture segmentation has been applied with relative success and most of the methods could be divided into feature extraction and classification parts. The following paragraph reviews some of the state-of-the-art methods for gesture segmentation and recognition.

2.1 Features extraction methods

The main idea of a segment is to locate the starting point and endpoint according to [15]. However, this frequently varies from signer to signer and is frequently unclear when used in surgical manoeuvres. because gestures used during surgery are dynamic, flow smoothly, and include both local and global motion. Both temporal and spatial distortion define them. In this scenario, segment points are chosen based on useful properties such as motion, position, velocity, and acceleration in video and sensor kinematics. These traits can be extracted using several techniques, some of which are listed below:

1. Based on multiple transformations from kinematic data:

    - Descriptive Curve Coding (DCC) or Gaussian models combined with Linear Discriminant Analysis (LDA) could extract features of different gestures recognised based on the temporal model.
    - Transition states can be noticed by analysis of kinematics data. Three researchers use different methods to spot transactions. Transition Movement Models(TMM) can identify transition states between adjacent sign[16]. [17] used a variation of the Skip-Chain Conditional Random Fields (SC-CRF) which has longer memory to enable capturing transitions between gestures. Thresholding is another most common aspect for capturing transitions: Lee and Kim [18] built a threshold model to reject a sequence of input signals as a gesture if they go beyond a certain threshold likelihood. However, this model can not detect gestures in real-time as it requires the whole input sequence to make a decision.
    - [19] introduced Shared Discriminative Sparse Dictionary Learning (SDSDL) which aims to build a common dictionary for all gestures that combine all the features and the parameters of a multi-class linear support vector machine (SVM) in an unsupervised manner. And by breaking down continuous motion into sparse linear combinations of atomic motions, a shared motion dictionary is built as shown in [20]. This dictionary contains only kinematics signals and can be used to identify and segment other similar motions.
    - Bhuyan et al. use velocity and acceleration information as input to a fuzzy logic-based control scheme to decompose continuous gestures.

- Kinematics data along with time tells about trajectory, thus Krishnan et al. [21] use a Dirichlet Process by assuming a categorical distribution before inferring the number of possible clusters in the trajectory. They deploy a hierarchical Gaussian Mixture Model (GMM) to cluster over kinematics and find the possible transitions. The clustering result of raw kinematics data is then input to another GMM model to detect any transitions in between the ones already found. Finally, they used a cluster model with respect to time, to only include those transitions which are observed consistently across multiple demonstrations to decrease spurious transitions.
- Considering multi-dimensional characteristics of motion trajectory, Self-Similarity Matrix (SSM) has been proposed for segmentation and recognition. [22][23]

2. Feature extraction for vision data

- Conditional Random Fields (CRFs) could take nearby and/or similar locations into account while processing an image. Then the sequence of gestures is obtained by minimizing the energy of the CRF.
- Image can provide not only surgical procedures but also surgical instruments. Lalys et al. [24] use color histogram intersection and Scale Invariant Feature Transform/Speeded-Up Robust Features (SIFT/SURF) to detect features based on the texture of objects and use AdaBoost classifier to detect specific instruments in a sub-window.
- Global Bag-of-Spatio-Temporal features (BoSTF) and LDSs used in [25][26] have higher accuracy for gesture recognition due to the powerful ability of those two models on obtaining discriminative features from video data.
- [27] argued that if combined features such as surgical tools, and relevant objects in the operation with kinematics features, better discrimination results will be achieved.
- Deep learning methods: There are also deep learning methods to extract features from kinematics and video data which will be introduced in the following paragraph.

2.2 Segmentation and recognition

After the features are extracted, they go through a classifier either to match with predefined gestures in the gesture vocabulary or grouped by their similarities. All supervised, semi-supervised and unsupervised learning algorithms could be used in this research, owning pros and cons.

2.2.1 supervised learning

Traditional machine learning methods

Obtaining temporal information for a sequence of motion can be compared to speech recognition in natural language[8]. The joint probability distribution over characteristics and gesture labels could be represented by a generative graphic model like Hidden Markov Models (HMM) or Linear Dynamical Systems (LSD). While Conditional Random Fields (CRF) is a discriminate model that gives the conditional distribution of the gesture label.

The first time HMMs were used to classify gestures in conjunction with other data transformation techniques, such as zero-mean and LDA, was in the paper [28]. When compared to regular HMM, modified HMM models like Factor-Analyzed HMM(FA-HMM)[29] and Sparse HMM [25] perform better in surgery recognition. However, states in HMMs are discrete and the transition probability distribution is predetermined to generate the observed sequence, failing to capture correlations between different states in continuous motion[8]. Thus, Switched LDS(S-LDS) is introduced to observe adjacent features. Unlike HMMs, LDS have hidden states that are continuous and dynamic. It also assumes that all the states are linear and the noise term follows a normal distribution. It is shown that S-LDS outperformed FA-HMM significantly[29].

Another supervised algorithm named Conditional Random Fields(CRF) is used to recognize gestures in a video considering the dependencies of neighbouring states. In [27], it inputs spatio-temporal visual features into a Markov/semi-Markov CRF (MsM-CRF) and enhance the performance than HMMs [30].

Deep learning methods

A deep neural network (DNN), in addition to conventional machine learning techniques, demonstrates strong capabilities when extracting features without preprocessing raw input. Recurrent Neural Network (RNN) and their variations are widely utilised today since Convolutional Neural Network cannot handle a task that has variable sequence lengths. A generic RNN contains hidden stage h_t and a recurrence function block at each step to provide output y_t according to history stage h_{t-1} and input x_t.

RNN, however, has a vanishing gradient issue. Because the gradients no longer affect the gradients at the initial time step as time passes, this results in an incredibly sluggish update time when there is a lengthy sequence of inputs. To achieve even longer memory to tackle longer temporal signals, Long Short-Term Memory(LSTM) was proposed. LSTM can remove or increase the information of "cell status" through the "gate" structure to realize the retention of important content and the removal of unimportant content. This is achieved through three special gates: i,f, and o which correspond to “input”, “forget”, and “output” gates.

Research [31] compares four RNN architectures including simple RNNs, LSTM, gated recurrent units(GRU), and mixed history RNNs. GRU synthesize the forget gate and the input gate into a single update door and mix the cell state and hidden state. Results show that LSTMs and GRUs are less sensitive to hyperparameters, which makes them more robust and achieves better recognition results. Because RNNs assume that various scale is equally important for each sample, and are unable to exploit multi-scale temporal dependencies, Multi-scale RNN(MS-RNN) is proposed in [32] utilizing hierarchical convolutions with wavelets. Inspired by Scattering Convolution Networks(ScatNet) which use a cascade of convolution with wavelets to enable translation invariance, the author combined ScatNet with LSTM to model multi-scale temporal features. It achieves high accuracy compared with LSTM and GRU when recognising signals with different lengths and is also less likely to overfit.

Hybrid DNN was first used in [33] for video action segmentation which extracts local motion features by CNN, and then uses an LSTM to extract long-range temporal action features. It shows better segmentation results than traditional machine learning methods such as HMMs and CRF and CNNs such as temporal CNNs and spatio-temporal CNNs.

In recent years, another method using Reinforcement learning (RL) set a new benchmark in surgical gesture segmentation and recognition task. In [34], the model choose step size and its corresponding gesture label to move forward at each timestamp while training. They use TCN to extract and concatenate current and future feature vectors and assigned them along with each state. And they use a multilayer perceptron (MLP) to model the policy. Though this model can achieve competitive results, it still outputs uncertain predictions when faced with rare gestures. Therefore in [35], the RL model is followed by a tree search model which determined the best action to follow.

In order to interpret language and model sequences, transformer-based neural networks provide a successful self-attention mechanism. However, they have not yet been thoroughly examined in terms of their applicability to the JIGSAW dataset. For the dynamic hand gesture recognition challenge, [6] suggest a transformer-based architecture. And [7] introduced a transformer-based architecture that accurately predicts surgical phases of laparoscopic cholecystectomy videos.

2.2.2 Semi-supervised learning

With only a small number of labelled data, semi-supervised learning methods are not complex as unsupervised methods and have usually better performance than unsupervised learning [8]. In [36], they use a Stacked Denoising Autoencoder (SDAE), an unsupervised learning model, to extract features in a decoder-encoder framework. The labelled data is compared with the unlabeled data by DTW by computing the distance between frames. Three distance metrics: Euclidean distance, Cosine Distance, and Kullback-Leibler(KL) dist are used to compare the performance. To further improve robustness to noise, and the alignment process, they implement a voting technique voted by a small set of reference labelled data.

RNN could also be used in semi-supervised methods that are trained to learn a full joint distribution over kinematic data. In [37], an RNN-based generative model is developed and the feature it learns goes into an LSTM for recognition.

Combined with transfer learning as present in [38], the features of motion trajectory are learned by Self-Similarity Matrix(SSM) and then learn segmentation policy to segment trajectory. SSM preserves multiple DoF data and is robust to noise. The segmentation rule is to find the optimal peak and filter out the noise and unwanted ones.

2.2.3 Unsupervised learning

Supervised learning suffered from the impractical requirement of a voluminous of labelled data, while unsupervised learning gets rid of this. In [21], Transition State Clustering (TSC) is applied to the segmentation task. It identified a set of the transition state of the whole trajectory, and fit the Gaussian Mixture Model whose number of regions is determined by the Dirichlet process prior (DP) to kinematic data points with sensory features along the time axis. The transition state is identified when the current state has a different most likely mixture component than the previous state.

In addition to using GMM to segment kinematic data, [39] uses CNN to extract visual features to improve performance. As this has a problem with over-segmentations, [40] add a promoting procedure after TSC that merges the clusters using four techniques: Principal Component Analysis (PCA), Mutual Infomation (MI), Data Average (DA) and Dynamic Time Warping (DTW). The paper also uses a Stacking Convolutional Auto-encoder (SCAE) that was proposed in [41] to speed up feature extractions from video.der step of SCAE sample the high-dimensional input to a low-dimensional feature map, while the decode step does the inverse to reconstruct.

Temporal Convolutional Networks (TCN) is further extended to Dense Convolutional Encoder-Decoder Netowork (DCED-Net) as an unsupervised learning algorithm [42]. Motivated by the fact that traditional unsupervised methods lost too much image information [43], the authors construct a DCED-Net that has a dense layer, a transition layer and an up-sampling layer that could effectively extract more information from a video. After the initial segmentation results come out, they were further selected by a majority vote strategy to remove spurious segment points and then merged by an iterative algorithm.

There are also more complex generative models improved to an unsupervised learning algorithm that outputs joint probability distribution of latent gestures. In [44], PRISM is present for a system that may not be in a stochastic latent dynamical structure. It is a hierarchical Bayesian model that performs degrades significantly in complex actions.

Though using visual data will improve the performance, it is computational expensive to leverage video data, thus in [45], they group the data samples based on feature similarities and temporal

constraints. This is realized by a bottom-up strategy to segment temporal sequence and a soft boundary approach to model the gradual transition between each segment. In [46], the segments are merged together based on several rules and constraints they defined.

Instead of clustering similar data or distinguishing the parameters of the clustering model, observing certain characteristics of specific tasks can be used as segmentation criteria. [47] explore both spatial and temporal properties of a given trajectory, and argued that kinematics data points are clustered when they are in a transition state while smooth intervals represent an in-motion state. In another word, when there exists a point with zero velocity and acceleration, it has a high probability of being in a transition state. Results show that this method is robust to noise, however, the threshold value is different to adjust.

3 Research Methodology

3.1 Database description

One of the well-known surgical activity datasets for human motion modelling is the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [48], which data collected from the da Vinci Surgical System containing kinematic and video data along with manual annotation with gestures name and skill levels. There are three surgical tasks, including suturing, needle-passing and knot-tying. This thesis chose suturing task to test the performance of the proposed method. The definitions of the surgical gesture for suturing task are extracted from paper [48] shown in Table 3.1. The gesture label is not continuous because suturing task does not use all the gestures defined in the whole dataset. And to remain the original label leaving the convenience for testing the proposed algorithm in another surgical task, such as needle passing. This thesis remains the same label and did not change the sequence. There are a total of 10 types of surgical gestures, each demonstration contains all or part of the gestures in a different sequence. Thus, for different demonstration file, the number of segmentation points are different, and for the recognition problem, this can be formulated as a 10-class classification problem.

3.2 Segmentation

The main idea of a segment is to locate the starting point and endpoint [15]. Gestures in surgery are dynamic and continue with smooth transitions containing both local and global motions, thus kinematic data is important and useful for noticing transition states. As for visual features, deep learning neural networks especially CNNs bring a huge breakthrough in image classification and demonstrate a strong ability to transfer learning skills such as feature extraction. What's more, Transformer is now a state-of-art neural network architecture that solves sequence transduction problems and has higher accuracy than RNN and LSTM in dealing with long-range context depen-

Task	Gesture label	Gesture description
Suturing	G1	Reaching for needle with right hand
	G2	Positioning needle
	G3	Pushing needle through tissue
	G4	Transferring needle from left to right
	G5	Moving to center with needle in grip
	G6	Pulling suture with left hand
	G8	Orienting needle
	G9	Using right hand to help tighten suture
	G10	Loosening more suture
	G11	Dropping suture at the end and moving to end points

Table 3.1: Definitions for Surgical Gestures [48]

dencies. Thus, a modified Transformer with one of the CNNs backbone is used for visual features extraction.

For declaration, define those features initially identified from certain feature extraction methods as 'change points' or 'critical points'. It is noticed that change points for different profiles may not occur simultaneously, take a knot typing task, for example, orienting a needle will only cause changes in the rotation profile, while the translation profile may not change. Thus, a hierarchical structure is introduced. Multiple and meaningful features identified from kinematic data are clustered with the first layer of 'Density-Based Spatial Clustering of Applications with Noise (DBSCAN)', and all visual features are clustered with DP-GMM initially. Those output data will be named 'pre-segmentation points' or 'potential segmentation points' in the following report. Finally, the second layer of DBSCAN is used for all pre-segmentation points to generate 'estimated segmentation points'. An overview of the whole segmentation method is present in Algorithm 1.

Where:

1. cp^{ori}: Change points frame number extracted from spatial-temporal characters
2. cp^{var}: Change points frame number extracted from variance characters
3. CP_{kine}: All change points for kinematics data
4. f_{vis}: Features extracted from neural network

5. CP_{vis}: All change points for visual data
6. esp : Estimated segmentation points

Algorithm 1: Segmentation algorithm overview

Input : Raw kinematics and visual data

Output: Estimated segmentation points

for *all kinematics data* **do**
    Data pre-process;
    cp^{ori} = Bi-manual translation and rotation distance profile ;
    cp^{var} = Translation and rotation variance profile
CP_{kine} = $[cp^{ori}, cp^{var}] \leftarrow$ DBSCAN ;
for *All visual data* **do**
    Dta pre-process;
    f_{vis} = All visual data $\leftarrow$ deep learning network;
    CP_{vis}: = $f_{vis} \leftarrow$ DP-GMM
$esp = [Cp_{kine}, CP_{vis}] \leftarrow$ DBSCAN

3.2.1 Kinematic feature extraction for identifying transitions

Inspired by [47] and [49], which utilize the inherent characteristics of human movements and argued that kinematics data points are clustered when they are in a transition state, while smooth intervals represent an in-motion state. This project first finds out the transitions on the density of trajectories in space by calculating transition and rotation distance. Then, a variance-based method followed [47] is also implemented to make the result more robust to noise.

Data Preprocess

As the original kinematics data include noise, the Kalman filter and Savitzky–Golay filter are used before feature extraction.

1. Kalmen Filter

Kalman filter has been proved to be an efficient recursive filter that could filter out the noise and increase robustness in the surgical dataset as present in [47], [50] and [51]. It smooths trajectory by predicting the coordinate position and speed of the object from a set of previously observed object positions and speed.

The Kalman filter model assumes the true state $\mathbf{X}_k$ at time k is calculated by the following equation from the true state at time t-1: $\mathbf{X}_k = \mathbf{A}_k\mathbf{x}_{k-1} + B\mathbf{U}_k + \mathbf{W}_k$. Thus, the observed state $\mathbf{Z}_k$ can be calculated from: $\mathbf{Z}_k = \mathbf{H}_k\mathbf{X}_k + \mathbf{v}_k$.

In the above equation, $\mathbf{H}_k$ is the observation model, which maps the true state space into the observed space. It contains a process noise $\mathbf{w}_k$ which is assumed to be drawn from a zero mean multivariate normal distribution, $\mathscr{N}$, with covariance, $\mathbf{Q}_k$: $\mathbf{w}_k \sim \mathscr{N}(0, \mathbf{Q}_k)$. And $\mathbf{v}_k$ is the observation noise, also named as measurement noise, which is assumed to be zero mean Gaussian white noise with covariance $\mathscr{R}_k$: $\mathbf{v}_k \sim \mathscr{N}(0, \mathbf{R}_k)$

In this experience, the observed true state are 6 dimension which can be written as: $\mathbf{x}_k = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$ where the first three parameters are the coordinates, and the last three parameters are the translation speed. The process noise $\mathbf{W}_k$ and observation noise $\mathbf{v}_k$ should be determined and will be discussed in Seciton 4.

2. Savitzky–Golay filter

Savitzky–Golay filter was initially proposed by Savitzky and Golay in 1964 and published in Analytical Chemistry magazine, which is based on the polynomial smoothing algorithm by the methods of linear least squares, also known as convolution flat smoothing. Different degrees of smoothness can be achieved by selecting a different length of sub-set of the original data.

Assume the selected window has length $2m$, then the input data will consist of a set of points $X = [x_{-m}, x_{-m+1}, ..., x_{m-1}, x_m]$. After that, use a k^t polynomial to fit the data points in the

window, then the output data after fitting is shown as equation 3.1:

$$\begin{bmatrix} y_{-m} \\ y_{-m+1} \\ \dots \\ y_{m-1} \\ y_m \end{bmatrix} = \begin{bmatrix} 1 & x_{-m} & \dots & x_{-m}^k \\ 1 & x_{-m+1} & \dots & x_{-m+1}^k \\ \dots & \dots & \dots & \\ 1 & x_{m-1} & \dots & x_{m-1}^k \\ 1 & x_m & \dots & x_m^k \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_{k-1} \\ a_k \end{bmatrix} + \begin{bmatrix} b_{-m} \\ b_{-m+1} \\ \dots \\ b_{m-1} \\ b_m \end{bmatrix} \tag{3.1}$$

Where a and b are parameters to be solved by linear least squares.

In this experience, the Savitzky–Golay filter is applied after the normalized euclidean translations distance and rotation distance which will be introduced in the following paragraph.

Spatial-temporal based

Translation and rotation distance between time t and t-1 is calculated for both left and right hands, respectively. For the translation component, the euclidean distance of the end-effector at time t and time t-1 is calculated as shown in equation 3.2:

$$D_{trans} = \sqrt{(x(t)-x(t-1))^2 + (y(t)-y(t-1))^2 + (z(t)-z(t-1))^2} \tag{3.2}$$

As for the rotational component, the distance is determined by equation 3.3, where qua is the quaternion converted from the 3*3 rotation matrix. In total, there are four distance profiles for two hands which form a bi-manual method.

$$D_{rot} = arccos(2 * (qua(t).qua(t-1))^2 - 1) \tag{3.3}$$

The four distance characteristics should be normalized to find the peaks which represent the potential segmentation points. Euclidean distance is one of the most well-known and effective methods in data mining. As segmentation points are points that separate two gestures, they likely happen when there is a large distance between two trajectory points. Thus, certain peaks and corners should be found. There are two methods that could be used to find peaks and corners. The first one is to specify a threshold height and prominence, while the second one is to use continuous wavelet transformation to identify relative peaks and corners. Two methods are compared in Section 5.

Variance based

Only calculating translation and rotation distances will still cause over-segmentation due to less robustness. This is because the raw trajectories still possess noise even after filtering and will lead to misidentified. Thus, to lessen noise effects and frame dependency, fresh frames are created after the variance-based approaches suggested by [47]. Translation and rotation effects are added to the original frame separately, thus, new trajectories are generated on new frames.

Denote the original frame as B, translation matrix as T(t), rotation matrix s R(t), then for each point p(t) = [x(t), y(t),z(t)] in frame B, the new trajectory point in the new translation frame could be written as $p^t(t) = p(t) + T(t)$, and the other new trajectories in the new rotation frame could be to written as $p^r(t) = p(t) * R(t)$. Assume there are total $2N$ new frames generated equally from random translation and rotation matrix, then the variance profile can be calculated based on all new frames at each time step, as illustrated in equation 3.4:

$$\begin{aligned} Var^j_{trans}(t) &= Var[{}^n p^t_x(t)] + Var[{}^n p^t_y(t)] + Var[{}^n p^t_z(t)] \\ Var^j_{rot}(t) &= Var[{}^n p^r_x(t)] + Var[{}^n p^r_y(t)] + Var[{}^n p^r_z(t)] \end{aligned} \tag{3.4}$$

where $n = 1, 2, 3....N$

Variance profile keeps the character of kinematics data similar to the distance profile. As segmentation points of the trajectory are likely clustered, and points in a continuous motion are likely sparse. These characteristics are frame invariant, in other words, will keep when to transform into a new frame. Thus, calculating the variances among all new frames will help examine the greater changes in angles. Points with higher variance indicate that the angle or speed change is large, thus points are likely within a continuous motion rather than in a change period. The same two methods will be used to find peaks and corners and the one with better results will be selected.

3.2.2 Merging kinematics changing points with DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clusters points that have high sufficient density. It does not need to specify the number of clusters in advance and only requires 2 hyperparameters. In addition, it can identify and handle abnormal values, which helps to eliminate misidentified change points.

In this experience, two layers of DBSCAN are used to find segmentation points. The first layer is used for clustering all kinematics critical points, and the second layer combined the output of kinematics critical points with the visual critical points as input and finds the final estimated segmentation points. To realize this algorithm, two parameters should be determined. The minimum number of points *minp* should appear in one cluster and the maximum distance *eps* between two points. When the number of points within the radius $eps/2$ of the neighboring domain is more than *minp*, it forms a dense region. The point at which the number of samples within its adjacent domain R is greater than *minp* is called the core point. The point that does not belong to the core point but is in the neighborhood of a certain core point is called the boundary point. If a point is neither a core point nor a boundary point, then it is a noise point.

3.2.3 Visual Feature Extraction for Identifying Transitions

This research employs the same network design for gesture recognition and extracts characteristics before the classification layer as [6], which gives a strategy for categorising dynamic hand gestures based on transformer architecture. The structure of feature extracting network network is shown in Figure3.1 modified from [6].

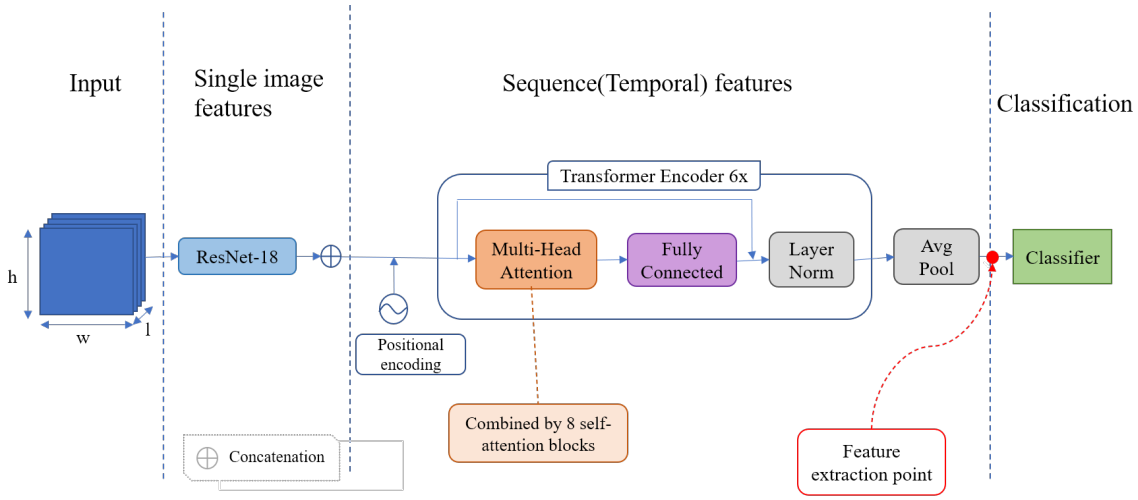

Figure 3.1: Network Architecture

For each task video in JIGSAW data, it is first converted to images at a frame rate of 30HZ with its original size of 640x480. Each input data is a sequence of l images belonging to the same gesture label. All inputs overlap with $(l-s)$ frames where s indicates the step. The step s should

be modest in order to overlap each motion and generate more training data, which will help the system learn more specifics and improve accuracy. The detailed parameters setting will present in Section 4.2.

Clustering using DP-GMM

The clustering method is inspired by Krishnan, Sanjay, et al. [21] who proposed an unsupervised transition state clustering (TSC) algorithm which is capable to spot segmentation points by identifying critical change points between two Gaussian clusters. It shows that DP-GMM has good performance in high dimensional data clustering without knowing the ground truth cluster number. Following the unsupervised TSC algorithm, [39] implement DP-GMM after deep learning network to leverage video and kinematic data.

This thesis combines the advantages of Transformer Network with DP-GMM in a hierarchical structure combing visual and original kinematics features. There are two layers of DP-GMM for visual critical points extraction. The first layer cluster across all the frames to find change points as much as possible, while the second layer further clusters those change points to find final segmentation points. As previously described, each visual feature represents a short sequence of certain gestures, and all output features are chronological. Then if there is a big difference between two adjacent features, gestures are likely to change at that point, and thus the potential segmentation points could be found.

Dirichlet process is a probability distribution which solves the problem of the pre-required class number of implementing a Gaussian mixture model. Assume there are N data $[x_1, x_2, ..., x_N]$, where each is generated from different distribution $g_1, g_2, ..., g_N$, and each distribution will have parameters $\theta_1, \theta_2, ..., \theta_N$ corresponding to themselves. Then if all g_i follows a different Gaussian distribution, then $\theta_i = \{\mu_i, \sigma_i\}$. Assuming θ_i follow a certain continuous distribution $H(\theta)$, Dirichlet process is about constructing a discrete distribution G to make $\theta_i \sim G$, where $G \sim DP(\alpha, H)$. α is called the concentration parameter and can be understood as a discrete level. A large concentration parameter distributes nearly equal weights to all samples, thus the weight is close to zero. In contrast, if $\alpha = 0$, this means using only one distribution to model all combinations.

The whole algorithm for visual feature extraction and change points founding is summarized in Algorithm 2.

Algorithm 2: Transforme + DP-GMM

Input : Images captures from video, α_1, α_2

Output: Potential segmentation points

length = l;
step = s;
for $t \leftarrow$ *All captured demonstration* **do**
    # get start frame for gesture 1,2,3...;
    $Start_{gt}$ = [start$_1$, $start_2$, $start_3$...]
    for $g \leftarrow$ *All gestures* **do**
        label = g;
        for $i \leftarrow$ *All frames in one gestures* **do**
            $d = [img_i, img_{i+1}, ... img_{i+l}]$
        $In_{ges} = [d_1, d_2, ..., d_g]$;
    $In_{demo} = [In^1_{ges}, In^2_{ges}, ..., In^t_{ges}]$;
    $f_{vis} = In_{demo} \leftarrow$ Transformer Network;
    $f_{vis} \leftarrow$ DP-GMM($comp_1$, α_1);
    $CP^{vis0} = f^i_{vis}$ if f^{i+1}_{vis} belongs to another cluster;
    $CP^{vis} = CP^{vis0} \leftarrow$ DP-GMM($comp_2$, α_2)

Where

1. $Start_{gt}$: The ground truth start frame number of a gesture.
2. $label$: The ground truth label of the gesture.
3. d: A set of l images.
4. In_{ges}: A dataset contains all sets of images in one demonstration file.
5. In_{demo}: The whole dataset contains all demonstration files for training and validation.
6. f_{vis}^{i}: Visual features, where i indicates frame number.
7. CP^{vis0}: Change points identified by the first DP-GMM.
8. CP^{vis}: Final change points identified by the second DP-GMM.

3.3 Classification

Though Transformer architecture in Figure 3.1 is not able to identify the start and end point, it has high accuracy in gesture recognition. Thus, after the segmentation points are found, the segmented data will go through the transformer network and output a certain gesture label.

3.4 Parameters tuning

There are several methods for parameter tuning, for example, assuming prior knowledge, using Grid Search, or using Random Search. However, those methods suffered from inefficient, limited searching range, are expensive for calculation or can not guarantee to return the best value. While Bayesian optimization uses the information of already-searched points to guide the new search point information, which can improve the search quality and the overall search speed. Thus, Bayesian optimization is used to find the best value of hyperparameters in this project.

The main process of Bayesian optimization is to first place a Gaussian process before target function f, then find some initial points that are distributed as uniform as possible in the definition domain of the function, and get their corresponding function values. Then update the posterior probability distribution on f using all available data. After that, use an acquisition function to determine the next (or the next batch) experimental point [52]. For the acquisition function, the upper confidence band is used in this project, which balances the exploitation and exploration

tradeoff by a linear-weighted method to make the sampling. This sampling method is simple and effective.

For visual feature extraction, hyperparameters included the concentration parameter α and components number in Dirichlet Process, where the component number indicates the value that the effective components should be smaller than. The maximum distance in DBSCAN should be tuned as well. Thus, there are a total of 5 parameters to tune for visual feature extraction. For final segmentation points estimation, Bayesian optimization is used for finding the best maximum distance *eps* of DBSCAN which clustering kinematics and visual pre-segmentation points to generate the final result. Both target functions use the f1 score of the segmentation metric, which will be introduced in the following paragraph.

3.5 Evaluation metric

In this thesis, two kinds of metrics are used for evaluating segmentation results and classification results. Both of them include precision, recall, and F1 score. For segmentation performance, denote the estimated segment point as p_e, the ground truth as p_g, then the estimated point is considered as correct (True Positive)(TP) if $|p_e - p_g| < \eta$, where η is 1s in this experience. According to the definition of precision, recall, and F1 score, metrics for segmentation points are calculated using the following equations:

$$recall = \frac{TP}{No.p_g} \qquad precision = \frac{TP}{No.p_e} \qquad F1 = \frac{2 * (recall * precision)}{recall + precision} \tag{3.5}$$

Where $No.p_g$ and $No.p_e$ denote the number of total ground truth points and estimated points.

As for multi-class classification, the micro and the macro average are calculated based on the confusion matrix following equation 3.6.

$$\begin{aligned} Micro_{precision} &= \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \qquad Micro_{recall} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i} \\ Macro_{precision} &= \frac{1}{n}\sum_{i=1}^{n} P_i \qquad Macro_{precision} = \frac{1}{n}\sum_{i=1}^{n} R_i \end{aligned} \tag{3.6}$$

Where:

1. TP_i, FP_i,FN_i denotes True Positive, False Positive and False Negative for each class i.
2. P_i, N_i denotes the precision and recall for each class i.

4 Experiments setup

This section introduces the experiments conducted for building the segmentation model by specifying the input data and hyperparameters and doing cooperation experiments. In addition, as demonstrated in Section 3, two criteria for finding critical points are compared in the following paragraph as well. Kinematic data provided in JIGSAW dataset include both master and slave tooltip information, this paper use master tooltip information.

4.1 Hyperparameters in kinematics feature extraction model

4.1.1 Filters setting

Kalman filter

As described in Section 3, process noise and observation noise should be determined. A smaller process noise means the more the system trust in the prediction value, and the system is easier to converge. In other words, the closer it is to 0, the more it believes in the prediction value, while the closer it is to infinite, the more it believes in the observation value. As for measurement noise, the larger it is, the higher the filter trust in the value of the new measurement value, so the system will respond slower. In a contract, the smaller value it has, the system responds faster but easier to fluctuate.

In this experience, the trajectory information is collected from surgeons with various robotic surgical experiences, thus different levels of errors are included which could be seen as noise. Thus the covariance of measurement noise is set to 0.05 and the covariance of process noise is set to $1 * 10^{-1}$ indicating that the system trust more on the prediction value.

Savitzky–Golay filter

Proper window size and order of polynomial function should be determined first. A shorter window length makes the smooth curve closer to the original curve, while a longer window length will have a stronger smooth effect. And a higher order of the polynomial function also makes the curve closer to the original curve. In addition, a higher order will cause a high-frequency curve into a straight line.

By visualizing the raw distance trajectory, it is found that the rotation distance is more oscillated, thus, larger window size and lower order are chosen for rotation distance.

4.1.2 Critical points finding

As introduced in Section 3, critical points are found when two points have a large distance. Two methods are used to find peaks and corners. Figure 4.1 and 4.2 shows the normalized euclidean distance and ground truth segmentation points for the left hand of suturing file D004 and suturing file B002.

The blue curve is the normalized euclidean distance, and ground truth segmentation points are shown in red and purple, indicating start and end points respectively. And the figures show that the distance range for D004 and B002 are [0,0.5] and [0, 0.25] separately. This means that even after normalization, the distance profile still has a large significant difference in mean and variance, which makes it hard to find a fixed threshold for height and prominence for all demonstrations. Thus, continuous wavelet transformation is used in this experience.

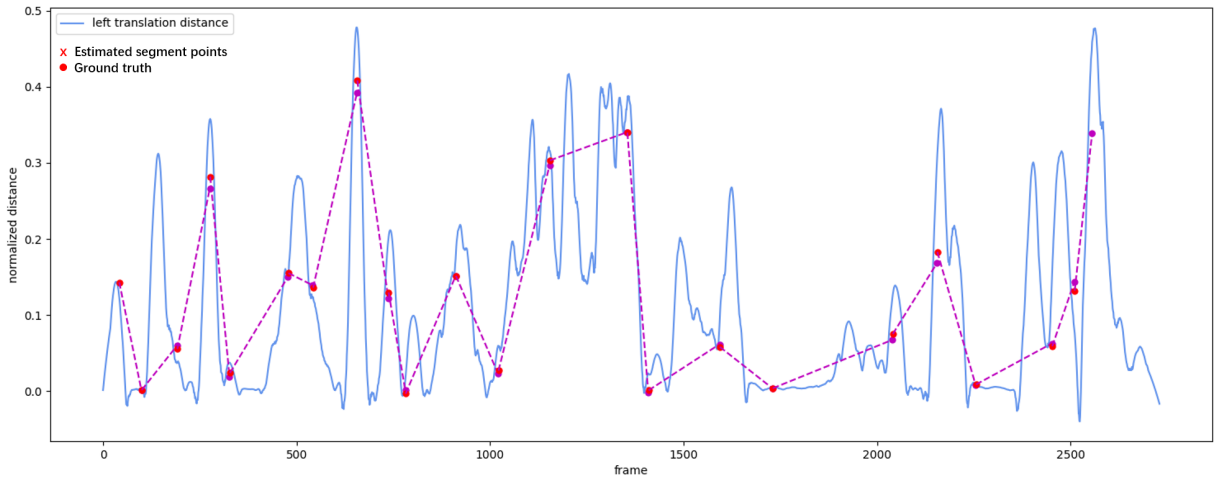

Figure 4.1: Fixed threshold method for euclidean distance (File D004)

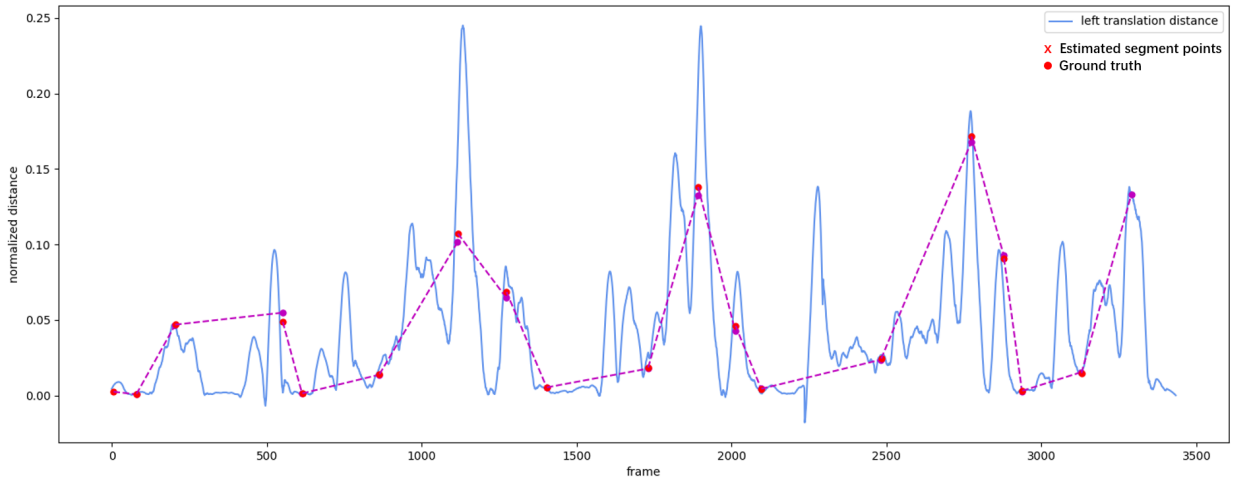

Figure 4.2: CWT method for euclidean distance (File B002)

4.2 Hyperparameters in visual features extraction model

There is a total of 39 Suturing demonstration files, which are separated into a train set, a validation set, and a test set in a 26:7:6 ratio. During the model training process of the Transformer network, all images are cropped to the size of 80x80 pixels due to CUDA memory and to accelerate the network training speed, and all images for validation have the dimensions of 120x120 pixels. To prevent overfitting, [6] uses random rescale, random crop, and random rotation between 15 and 15 degrees as data augmentation. This is implemented in this project as well.

Image analysis period uses the ResNet-18 architecture whose weights were previously learned

on ImageNet, while the rest of the architecture is trained from scratch[6]. Adam with weight decay(AdamW) is chosen as an optimizer which directly adds the gradient of the regular item into the formula of the reverse propagation and eliminates manually adding the regularization term in loss calculation, which accelerates calculation speed. The loss function is constructed by categorical cross-entropy. The learning rate is set to 0.001. As described in Section 3, a window length of 30, the step of 5 is chosen to construct the dataset for model training. The batch size is set as 30 and the total training epoch is set to 80.

To extract visual features, training models from epochs 25 and 80 (the final epoch) are used. This is to investigate the influence on the accuracy of the deep learning model.

As for the classification evaluation for the complete algorithm, all the demonstration files given the estimated segmentation points are used as the prediction dataset. For each section that is divided by two adjacent segment points, a sliding window of size 30 and step of 30 is used. Unlike for constructing a training dataset, the step s should be relatively small to create an overlap, this time, it is set as the same length as the sliding window which is 30. This is to reduce the size of prediction data to increase predicting speed. For each section, assume the length of a certain section is L, and the number of prediction labels is $L/30$. The final prediction label is refined by choosing the most frequent label that appears in a certain section.

4.2.1 Training data for DP-GMM

Two experiments are set up for DP-GMM methods. In the first experience, only visual features are used, while in the second experience, visual features are combined with raw kinematics data. This is to compare whether the pre-process kinematics data introduced in section 3.2.1 will influence segmentation points.

4.2.2 Target function for Bayesian optimizer

F1-score is chosen as the target function for Bayesian Optimizing. In comparison to the accuracy metric, the F1-score describes the harmonic mean of precision and recall, which provides a better assessment of the instances that were erroneously categorised. Especially when classes are unbalanced, F1-score achieves better statistic.

5 Results

In this section, all the experiments results described in Section 4 are presented to verify the purposed algorithm. All the figures are randomly selected.

5.1 Result of using kinematic data

Followed by the methodology in Section 3.1.1, pre-segmentation points can be generated using this unsupervised method.

5.1.1 Filtering

Figure 5.2 shows the smoothed translation trajectory distance after implementing the Kalman filter, while 5.1 shows the raw trajectory distance. Ground truth segmentation points are shown in red and purple, indicating start and end points respectively. And the critical points are shown as the red 'x' mark. It is found that oscillation points are filtered out, such that the critical points are found more accurately. The selected figures are from demonstration C003, which is randomly selected.

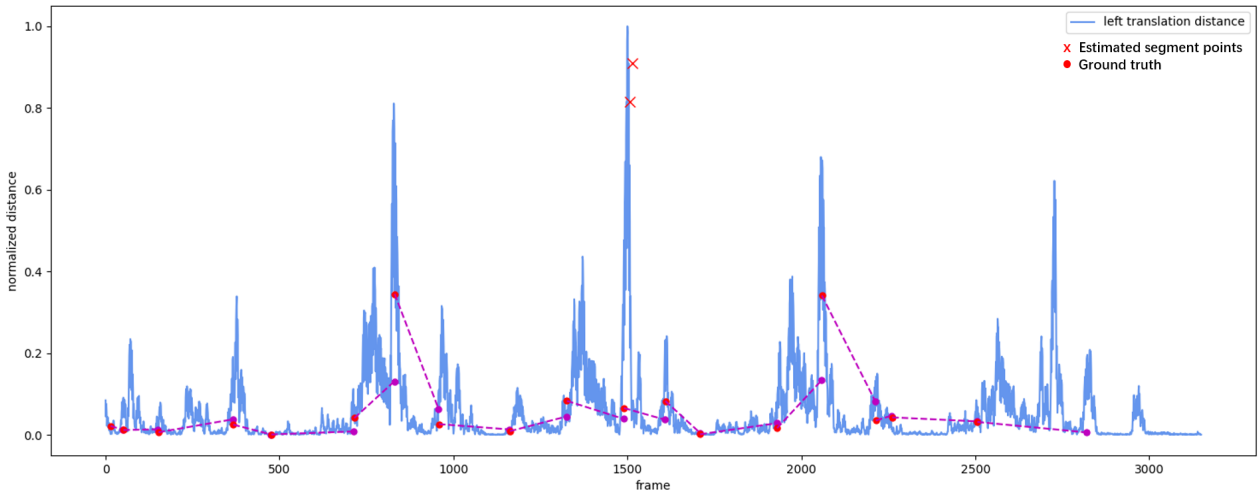

Figure 5.1: Raw translation trajectory distance (File C003).

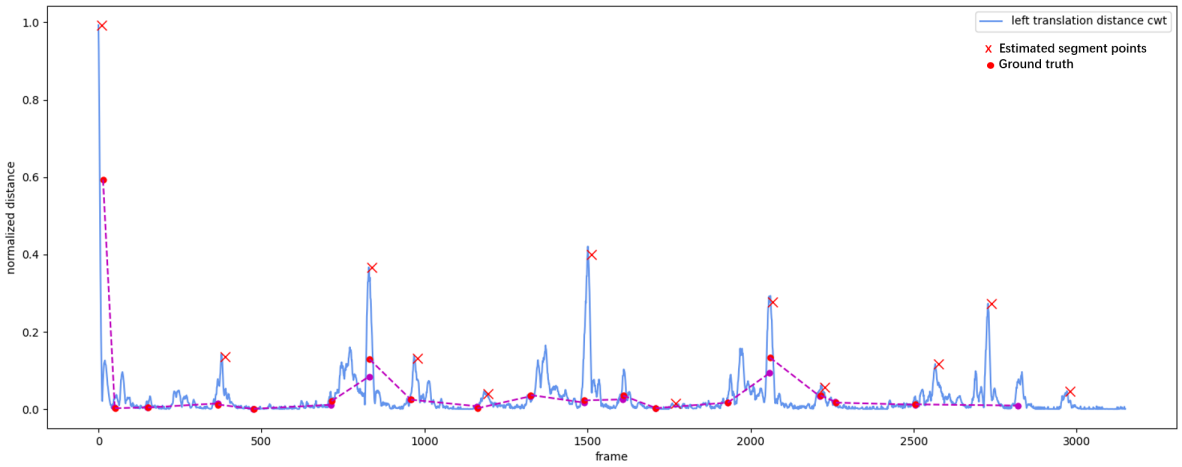

Figure 5.2: Smoothed translation trajectory distance with Kalman filter (File C003)

As in Figure 5.2, the segmentation points are not always happened at the peak of the translation distance profile. Thus, to increase accuracy, more strict rules should be used to avoid over-segment.

Figure 5.4 and 5.3 shows the rotation distance with spotted critical points with and without the Savitzky–Golay filter. Figures are from demonstration file C003. It is found that the critical points are more accurately founded after applying the Savitzky–Golay filter. As the ground truth points in the red dot show, gesture changes mostly happened at the peak and corner of the rotation distance profile.

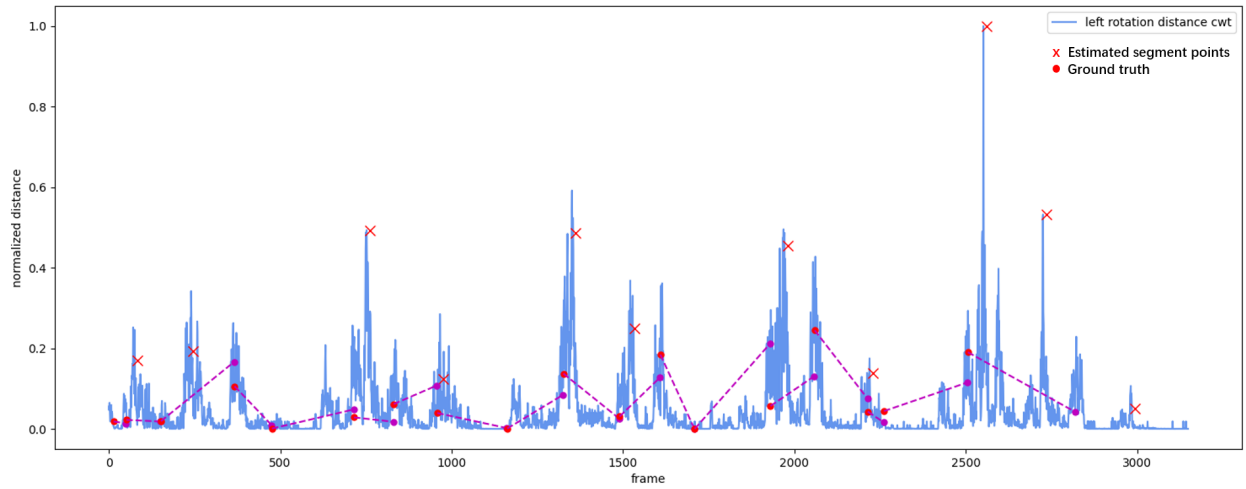

Figure 5.3: Without the Savitzky–Golay filter (File C003) Where: 1. Blue curve: normalized rotation distance; 2. Red dot: ground truth start and end points; 3. Red 'x': critical points

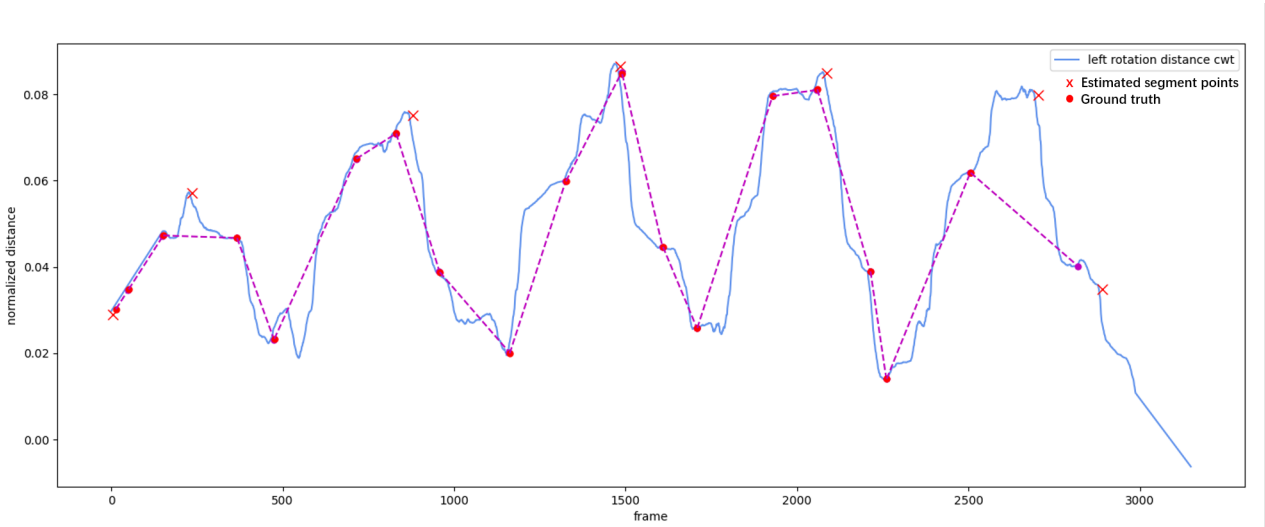

Figure 5.4: Rotation distance with the Savitzky–Golay filter
(File C003)

What's more, from Figure5.4, it is seen that with the cwt method, the correct corner points are not found. To do this, the variance value is reversed to find the critical points at a corner, as shown in Figure5.5.

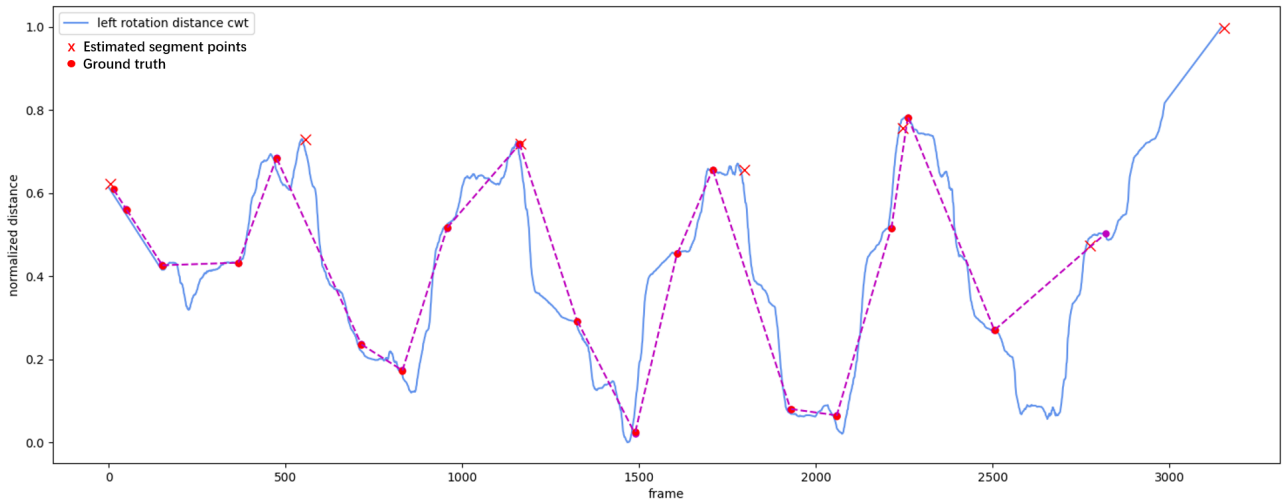

Figure 5.5: Reversed rotation distance with the Savitzky–Golay filter (File C003)

5.1.2 Spatial-temporal based

An example of extracted critical points is shown in Figure5.6 for left-hand properties, which is from the randomly selected demonstration file F004. Observing the estimated segmentation points

in the above and below figure, it indicated that the rotation distance profile provides a higher accuracy result than the translation distance profile. This might be because the suturing task requires more rotational motion than translational motion, thus causing more remarkable peaks and corners.

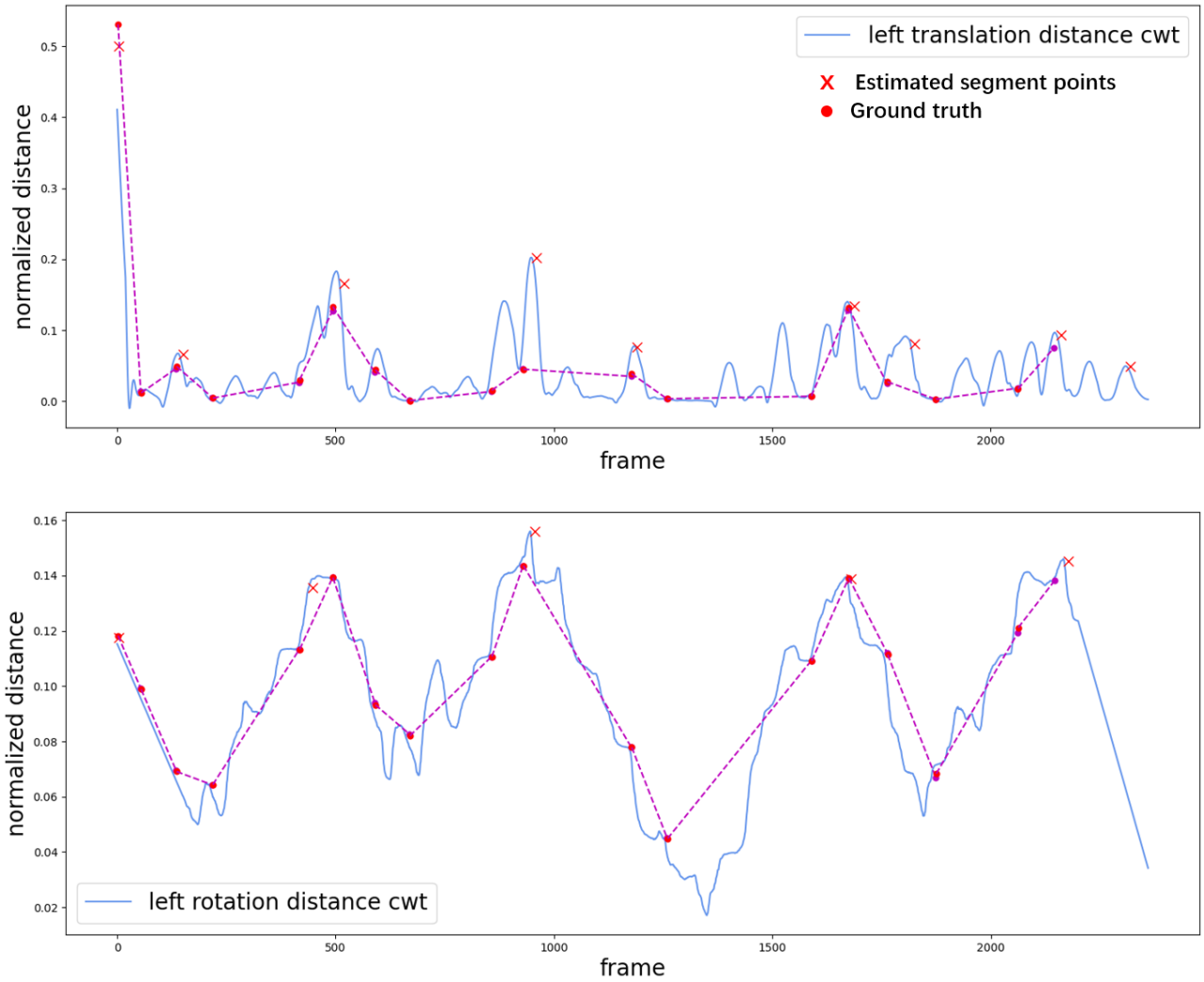

Figure 5.6: Critical points after Spatial-temporal method-left hand (File F004)

5.1.3 Variances based

Examples of critical points for rotation variances are shown in Figure5.7. The figure is from the randomly selected demonstration file B005. As the ground truth points in the red dot show, gesture changes mostly happened at the peak and corner.

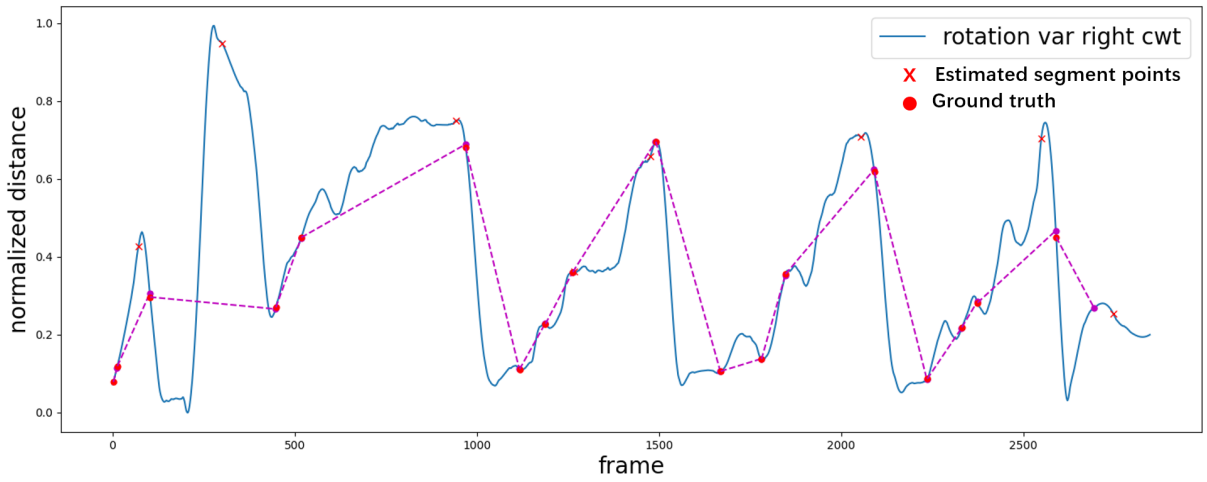

Figure 5.7: Rotation variance character-right hand (File B005)

One of the examples of the translation variance character for right-hand trajectories is present in Figure 5.8, which is selected from demonstration file H003. It shows that segment points in the translation variance profile are not as significant as the rotation variance profile, in which only several of them are located at the peak or corner. But both the critical points from two variance characters are kept because the purpose of critical points finding is to identify as much as possible in order to further cluster temporally nearby the point.

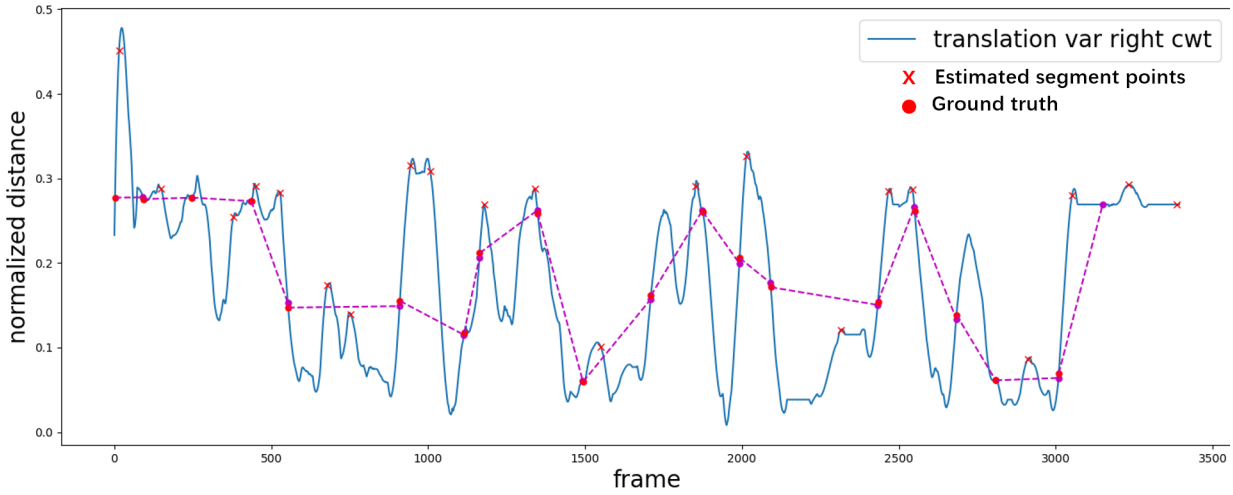

Figure 5.8: Translation variance character-right hand(File H003)

Combining all the critical points founded by distance-based and variance-based methods for each demonstration file, the estimated segmentation results by only using the kinematics data are

present in Figure5.9. This is the result for suturing demonstration file C003, with recall=0.467, and precision =0.368.

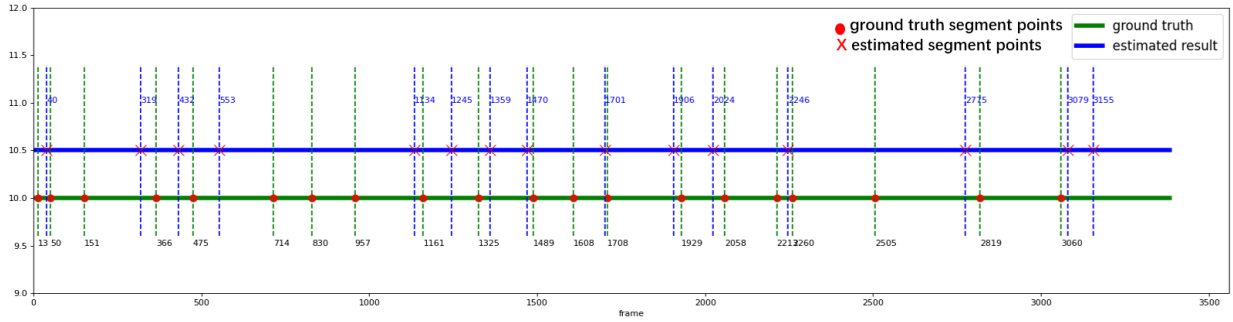

Figure 5.9: Estimated segmentation results by only using kinematics data(File C003)

The mean of recall, precision and f1 score for all the suturing demonstration files are 0.350, 0.264 and 0.301 respectively. The above figure shows that the points found are near the ground truth, however, still exists a big deviation.

5.2 Result of using visual data

The result of the 5 parameters as illustrated in Section 3.4 are listed in Table 5.1

Conditions	Target Function	Component No.1	Component No.2	$\alpha 1$	$\alpha 2$	eps
Visual only	F1-score	265.0	86.66	1.336	1.407	24.24
Visual + kinematics	F1-score	855	100	304	98	23

Table 5.1: Parameters choosing with Bayesian optimizer

Two training files are used to extract visual features. Figure 5.10 shows the segmentation result for demonstration file C003 with visual data only using a training model from epoch 25, which the training model from epoch 25 itself owns an accuracy of 0.73 on test datset. For the segmentation metric, the mean of recall, precision and f1 score for all the suturing demonstration files are 0.507, 0.409 and 0.453, respectively.

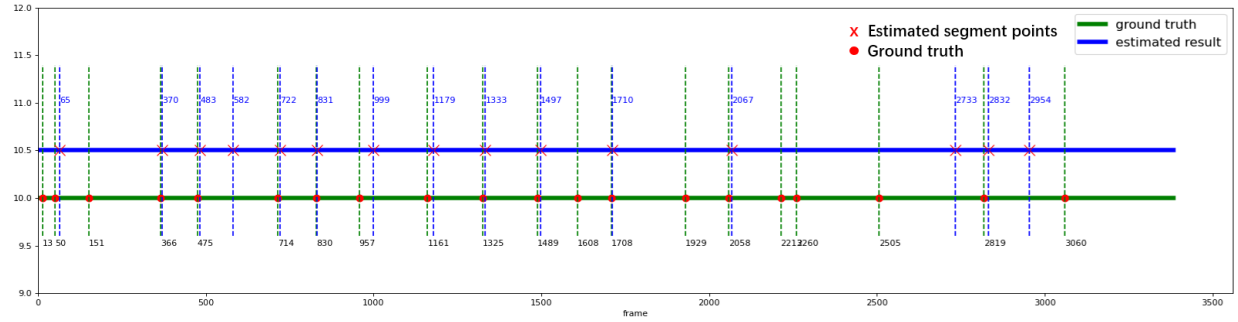

Figure 5.10: Segmentation result with visual data-epoch25(File C003)

The segmentation outcome for a training model from epoch 80 is shown in Figure 5.11, which the accuracy of the training model on the test dataset from epoch 80 is 0.93. The mean of recall, precision, and F1-score for segmentation metrics are 0.467, 0.359 and 0.401, respectively. The comparation of the segmentation score can be found in Table 3.1.

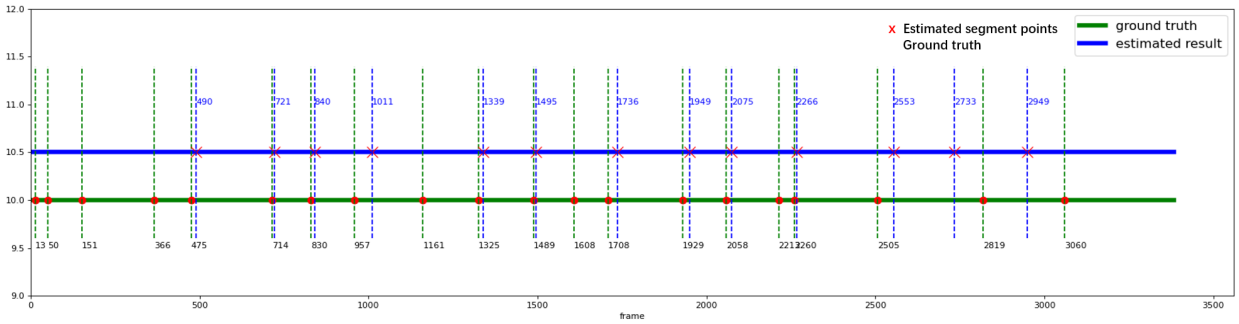

Figure 5.11: Segmentation result with visual data-epoch80(File C003)

Compared with the result only using kinematics data, the points are found more accurately. However, there are not many differences between the training model from epochs 27 and 80. Thus, for the segmentation task, there is no need to wait for the final training model. In the following experiments, the training model from epoch 80 is chosen,

5.3 Result of combining visual data and kinematics data

Figure 5.12 is the final result for demonstration file C003, with recall= 0.692, precision =0.947 and f1 score = 0.8. The result shows that more segment points are found successfully.

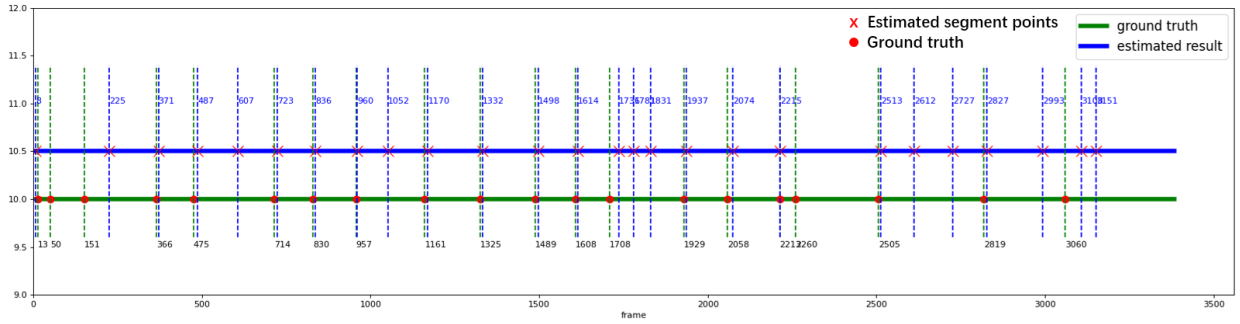

Figure 5.12: Final segmentation result with visual data + processed kinematics data (File C003)

The mean recall, precision and f1 score for all the suturing demonstration files are 0.555, 0.738 and 0.634, respectively.

5.3.1 Summary of segmentation results

All segmentation scores were summarised in the table 5.2. The findings demonstrate that the suggested approach produces the best results for locating segmentation sites, with both improvements in recall and precision.

Data used	Recall	Precision	F1 score
Processed kinematics	0.350	0.264	0.301
Visual-epoch25	0.507	0.409	0.453
Visual-epoch80	0.467	0.359	0.401
Visual + raw kinematics-epoch25	0.523	0.562	0.542
Visual + raw kinematics-epoch80	0.543	0.634	0.580
Processed kinematics + Visual	**0.555**	**0.738**	**0.634**

Table 5.2: Summary of segmentation results

5.4 Classification results with estimated segmentation points

Figures 5.13 and 5.14 provide examples of classification results with estimated segmentation points. Figures are from the demonstration file E005 and G005, Figures are taken from the demonstration file E005 and G005, which were randomly chosen.

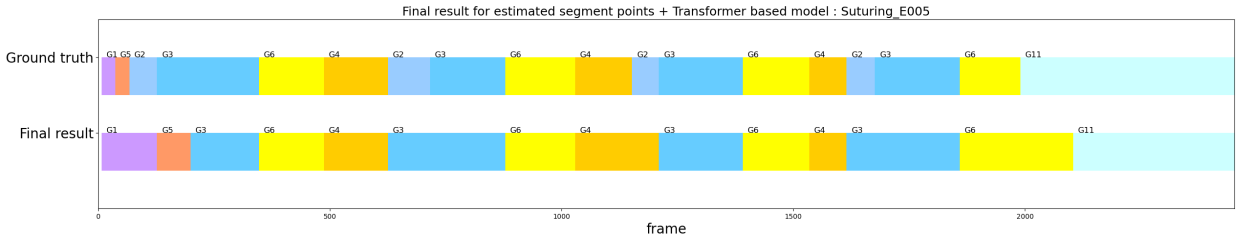

Figure 5.13: Final segmentation result with visual data+kinematics data(File E005)

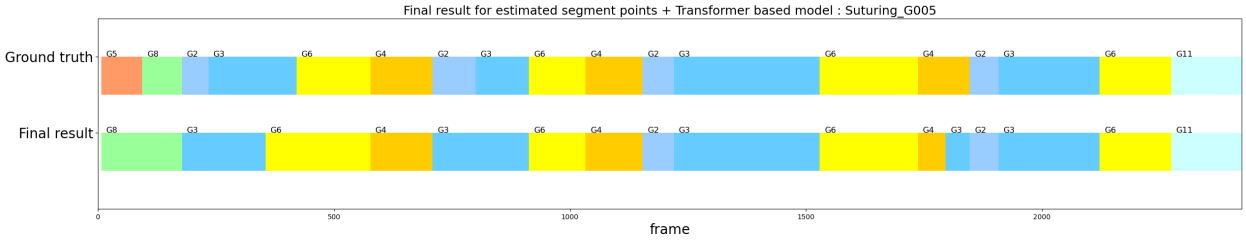

Figure 5.14: Final segmentation result with visual data+kinematics data(File G005)

Scores for classification of the prediction dataset are shown in Table 5.16, which is calculated depending on the confusion matrix as shown in Figure 5.15. The result in Table 5.16 suggests that the transformer-based model can be extended to identify segmentation points by the given approaches.

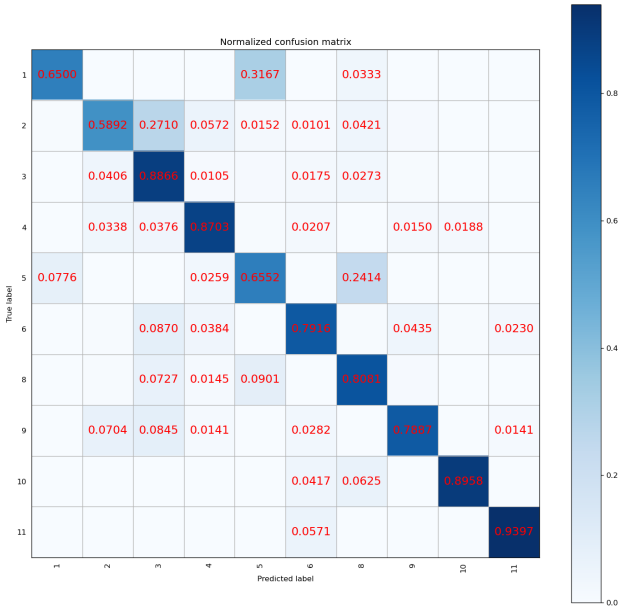

Figure 5.15: Confusion matrix for prediction dataset

Recall_micro	0.812
Precision_micro	0.812
Recall_macro	0.743
Precision_macro	0.752
Recall_weighted	0.823
Precision_weighted	0.812
Accuracy	0.813

Figure 5.16: Summary of segmentation results

6 Discussion and Conclusion

In this study, a novel approach was devised to learn gesture segmentation and recognition. A hierarchical model for clustering change points and identifying pre-segmentation points is developed. The model first investigates the normalized euclidean distance and variance characters on kinematics data and identifies critical points on corners and peaks. The visual features are extracted after the encoding layer of the transformer network, and the pre-segmentation points are identified by DP-GMM. Making use of implicit information from both kinematic and video data, the proposed methods could successfully identify segmentation points and recognize gesture labels. The proposed method is verified on suturing demonstration in JIGSAWS dataset, achieving an accuracy of 0.813 for classification and an F1 score of 0.634 for the segmentation metrics proposed in this paper.

6.1 Compared with the state-of-art

Using JIGSAWS, the suturing task has been tested using various deep-learning and unsupervised methods for gesture classification tasks, LSTM[53], TCN[54] are some of the state-of-art deep learning methods for classification and Bimanual space and variance[47], Soft-UGS[45], and Zero-shot[55] are some the unsupervised methods nowadays. All of these techniques can be used as benchmarks for measuring how well the surgical gesture recognition for suturing task is performed. Their average scores on the suturing demonstration of JIGSAW are presented in Table 6.1:

Method	Metrics	Scores
LSTM	Accuracy	0.805
TCN	Accuracy	0.796
Bimanual space and variance	Recall/Precision/F1	0.652 / 0.92 / 0.754
Soft-UGS	Recall/Precision/F1	0.74 / 0.71 / 0.72
Zero-shot	Accuracy	0.56
Unsupervised + Transformer	**Accuracy**	**0.813**

Table 6.1: Summary of segmentation results

Table 6.1 displays the findings of comparisons between the suggested technique and cutting-edge algorithms in terms of surgical gesture recognition accuracy. According to the results, the suggested technique in this paper performs well.

6.2 Future work

The proposed algorithm is only tested on JIGSAW suturing data, thus, to check the transfer ability of the algorithm, additional experimental validation based on other surgical activities, such as knot-tying and needle-passing tasks, can be carried out. In addition, as mentioned in Section 5.2, the transformer network itself can achieve 0.93 accuracy for gesture recognition. Thus, keeping tuning parameters for the unsupervised part will assuring better accuracy for both segmentation and recognition metrics. Future directions also include extending the proposed algorithm to a semi-supervised or self-supervised method, to avoid repeated training for each surgical task.

References

[1] K. R. Cleary, D. S. Stoianovici, N. D. Glossop, *et al.*, "Ct-directed robotic biopsy testbed: Motivation and concept," in *Medical Imaging 2001: Visualization, Display, and Image-Guided Procedures*, International Society for Optics and Photonics, vol. 4319, 2001, pp. 231–236.

[2] R. H. Taylor, B. D. Mittelstadt, H. A. Paul, *et al.*, An image-directed robotic system for precise orthopaedic surgery, *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3 1994, pp. 261–275, 1994.

[3] D.-S. Kwon, J.-J. Lee, Y.-S. Yoon, *et al.*, "The mechanism and registration method of a surgical robot for hip arthroplasty," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, IEEE, vol. 2, 2002, pp. 1889–1894.

[4] G. P. Moustris, S. C. Hiridis, K. M. Deliparaschos, and K. M. Konstantinidis, Evolution of autonomous and semi-autonomous robotic surgical systems: A review of the literature, *The international journal of medical robotics and computer assisted surgery*, vol. 7, no. 4 2011, pp. 375–392, 2011.

[5] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, Continuous role adaptation for human–robot shared control, *IEEE Transactions on Robotics*, vol. 31, no. 3 2015, pp. 672–681, 2015.

[6] A. D'Eusanio, A. Simoni, S. Pini, G. Borghi, R. Vezzani, and R. Cucchiara, "A transformer-based network for dynamic hand gesture recognition," in *2020 International Conference on 3D Vision (3DV)*, IEEE, 2020, pp. 623–632.

[7] T. Czempiel, M. Paschali, D. Ostler, S. T. Kim, B. Busam, and N. Navab, "Opera: Attention-regularized transformers for surgical phase recognition," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2021, pp. 604–614.

[8] B. van Amsterdam, M. J. Clarkson, and D. Stoyanov, Gesture recognition in robotic surgery: A review, *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 6 2021, 2021.

[9] D. Katić, J. Schuck, A.-L. Wekerle, *et al.*, Bridging the gap between formal and experience-based knowledge for context-aware laparoscopy, *International journal of computer assisted radiology and surgery*, vol. 11, no. 6 2016, pp. 881–888, 2016.

[10] N. Schoch, F. Kißler, M. Stoll, *et al.*, Comprehensive patient-specific information preprocessing for cardiac surgery simulations, *International journal of computer assisted radiology and surgery*, vol. 11, no. 6 2016, pp. 1051–1059, 2016.

[11] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. Kim, Supervised autonomous robotic soft tissue surgery, *Science translational medicine*, vol. 8, no. 337 2016, 337ra64–337ra64, 2016.

[12] H. C. Lin, *Structure in surgical motion*. The Johns Hopkins University, 2010.

[13] C. E. Reiley and G. D. Hager, "Decomposition of robotic surgical tasks: An analysis of subtasks and their correlation to skill," in *M2CAI workshop. MICCAI, London*, vol. 36, 2009, pp. 40–43.

[14] L. Al-Hakim, M. Wang, J. Xiao, D. Gyomber, and S. Sengupta, Hierarchical task analysis for identification of interrelationships between ergonomic, external disruption, and internal disruption in complex laparoscopic procedures, *Surgical Endoscopy* 2018, pp. 1–15, 2018.

[15] B. K. Chakraborty, D. Sarma, M. K. Bhuyan, and K. F. MacDorman, Review of constraints on vision-based gesture recognition for human–computer interaction, *IET Computer Vision*, vol. 12, no. 1 2018, pp. 3–15, 2018.

[16] G. Fang, W. Gao, and D. Zhao, Large-vocabulary continuous sign language recognition based on transition-movement models, *IEEE transactions on systems, man, and cybernetics-part a: systems and humans*, vol. 37, no. 1 2006, pp. 1–9, 2006.

[17] C. Lea, G. D. Hager, and R. Vidal, "An improved model for segmentation and recognition of fine-grained activities with application to surgical training tasks," in *2015 IEEE Winter Conference on Applications of Computer Vision*, 2015, pp. 1123–1129. DOI: 10.1109/WACV.2015.154.

[18] H.-K. Lee and J.-H. Kim, An hmm-based threshold model approach for gesture recognition, *IEEE Transactions on pattern analysis and machine intelligence*, vol. 21, no. 10 1999, pp. 961–973, 1999.

[19] S. Sefati, N. J. Cowan, and R. Vidal, "Learning shared, discriminative dictionaries for surgical gesture segmentation and classification," in *MICCAI Workshop: M2CAI*, vol. 4, 2015.

[20] E. Mavroudi, D. Bhaskara, S. Sefati, H. Ali, and R. Vidal, "End-to-end fine-grained action segmentation and recognition using conditional random field models and discriminative sparse coding," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2018, pp. 1558–1567.

[21] S. Krishnan, A. Garg, S. Patil, *et al.*, Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning, *The International Journal of Robotics Research*, vol. 36, no. 13-14 2017, pp. 1595–1618, 2017.

[22] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez, View-independent action recognition from temporal self-similarities, *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1 2010, pp. 172–185, 2010.

[23] B. Krüger, A. Vögele, T. Willig, A. Yao, R. Klein, and A. Weber, Efficient unsupervised temporal segmentation of motion data, *IEEE Transactions on Multimedia*, vol. 19, no. 4 2016, pp. 797–812, 2016.

[24] F. Lalys, L. Riffaud, D. Bouget, and P. Jannin, A framework for the recognition of high-level surgical tasks from video images for cataract surgeries, *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4 2011, pp. 966–976, 2011.

[25] L. Tao, E. Elhamifar, S. Khudanpur, G. D. Hager, and R. Vidal, "Sparse hidden markov models for surgical gesture classification and skill evaluation," in *International conference on information processing in computer-assisted interventions*, Springer, 2012, pp. 167–177.

[26] L. Zappella, B. Béjar, G. Hager, and R. Vidal, Surgical gesture classification from video and kinematic data, *Medical image analysis*, vol. 17, no. 7 2013, pp. 732–745, 2013.

[27] L. Tao, L. Zappella, G. D. Hager, and R. Vidal, "Surgical gesture segmentation and recognition," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2013, pp. 339–346.

[28] C. Rupprecht, C. Lea, F. Tombari, N. Navab, and G. D. Hager, "Sensor substitution for video-based action recognition," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 5230–5237.

[29] B. Varadarajan, *Learning and inference algorithms for dynamical system models of dextrous motion*. The Johns Hopkins University, 2011.

[30] M. K. Bhuyan, D. Ajay Kumar, K. F. MacDorman, and Y. Iwahori, A novel set of features for continuous hand gesture recognition, *Journal on Multimodal User Interfaces*, vol. 8, no. 4 2014, pp. 333–343, 2014.

[31] R. DiPietro, N. Ahmidi, A. Malpani, *et al.*, Segmenting and classifying activities in robot-assisted surgery with recurrent neural networks, *International journal of computer assisted radiology and surgery*, vol. 14, no. 11 2019, pp. 2005–2020, 2019.

[32] I. Gurcan and H. Van Nguyen, "Surgical activities recognition using multi-scale recurrent networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 2887–2891.

[33] L. Ding and C. Xu, Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation, *arXiv preprint arXiv:1705.07818* 2017, 2017.

[34] D. Liu and T. Jiang, "Deep reinforcement learning for surgical gesture segmentation and classification," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2018, pp. 247–255.

[35] X. Gao, Y. Jin, Q. Dou, and P.-A. Heng, "Automatic gesture recognition in robot-assisted surgery with reinforcement learning and tree search," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8440–8446. DOI: 10.1109/ICRA40945.2020.9196674.

[36] Y. Gao, S. S. Vedula, G. I. Lee, M. R. Lee, S. Khudanpur, and G. D. Hager, "Unsupervised surgical data alignment with application to automatic activity annotation," in *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2016, pp. 4158–4163.

[37] R. DiPietro and G. D. Hager, "Automated surgical activity recognition with one labeled sequence," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2019, pp. 458–466.

[38] Y.-Y. Tsai, B. Huang, Y. Guo, and G.-Z. Yang, "Transfer learning for surgical task segmentation," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 9166–9172.

[39] A. Murali, A. Garg, S. Krishnan, *et al.*, "Tsc-dl: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with deep learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 4150–4157.

[40] Z. Shao, H. Zhao, J. Xie, Y. Qu, Y. Guan, and J. Tan, "Unsupervised trajectory segmentation and promoting of multi-modal surgical demonstrations," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 777–782.

[41] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International conference on artificial neural networks*, Springer, 2011, pp. 52–59.

[42] H. Zhao, J. Xie, Z. Shao, Y. Qu, Y. Guan, and J. Tan, A fast unsupervised approach for multi-modality surgical trajectory segmentation, *IEEE Access*, vol. 6 2018, pp. 56 411–56 422, 2018.

[43] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2472–2481.

[44] K. Goel and E. Brunskill, "Learning procedural abstractions and evaluating discrete latent temporal structure," in *International Conference on Learning Representations*, 2018.

[45] M. J. Fard, S. Ameri, R. B. Chinnam, and R. D. Ellis, Soft boundary approach for unsupervised gesture segmentation in robotic-assisted surgery, *IEEE Robotics and Automation Letters*, vol. 2, no. 1 2016, pp. 171–178, 2016.

[46] F. Despinoy, D. Bouget, G. Forestier, *et al.*, Unsupervised trajectory segmentation for surgical gesture recognition in robotic training, *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6 2015, pp. 1280–1291, 2015.

[47] Y.-Y. Tsai, Y. Guo, and G.-Z. Yang, "Unsupervised task segmentation approach for bimanual surgical tasks using spatiotemporal and variance properties," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 1–7.

[48] Y. Gao, S. S. Vedula, C. E. Reiley, *et al.*, "Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling," in *MICCAI workshop: M2cai*, vol. 3, 2014, p. 3.

[49] Y. Guo, Y. Li, and Z. Shao, Rrv: A spatiotemporal descriptor for rigid body motion recognition, *IEEE transactions on cybernetics*, vol. 48, no. 5 2017, pp. 1513–1525, 2017.

[50] J. Ryu, Y. Moon, J. Choi, and H. C. Kim, A kalman-filter-based common algorithm approach for object detection in surgery scene to assist surgeon's situation awareness in robot-assisted laparoscopic surgery, *Journal of Healthcare Engineering* [online], vol. 2018 2018, pp. 1–11, 2018, ISSN: 2040-2295. DOI: 10.1155/2018/8079713.

[51] M. Ashikuzzaman, N. Jafarpisheh, S. Rottoo, P. Brisson, and H. Rivaz, *Fast and robust localization of surgical array using kalman filter*, Dec. 2020.

[52] P. I. Frazier, A tutorial on bayesian optimization, *arXiv preprint arXiv:1807.02811* 2018, 2018.

[53] R. DiPietro, C. Lea, A. Malpani, *et al.*, "Recognizing surgical activities with recurrent neural networks," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2016, pp. 551–558.

[54] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *European conference on computer vision*, Springer, 2016, pp. 47–54.

[55] T. S. Kim, J. Jones, M. Peven, *et al.*, "Daszl: Dynamic action signatures for zero-shot learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 1817–1826.