AUTO GUIDANCE OF GROUND CREW MEMBER ON AIRPLANES TAXIING MOVEMENTS

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Yifeng Shi

Aug 2020

ABSTRACT

Nowadays flight travel becomes a common way for people for work or leisure. With more frequent flight travels, lots of manual instructions by the ground tower and by ground crew members are needed. More resources and costs are accumulated, making up a big budget for the airplane managers to consider. Thus, a methodology of automatically guiding the airplanes to taxi based on the guidance gestures from the ground crew member is proposed. Many methodologies are used for human actions segmentation and change points detection of the human movements. Most of the algorithms focus on the single type of human action, which cannot generalize to the everyday human activities. This thesis presents a methodology that combine both the actions segmentation and recognition parts, and direct the airplanes based on the action recognition results.

BIOGRAPHICAL SKETCH

Yifeng Shi is a MS student in the Laboratory for Intelligent Systems and Controls (LISC) at Cornell University. He received his B.S. degree in Mechanical Engineering from the Pennsylvania State University, where he was an undergraduate researcher in the Intelligent Vehicle and System group (IVSG). His research interest includes machine learning, computer vision, with a focus on the human action recognition.

This document is dedicated to all Cornell graduate students.

ACKNOWLEDGEMENTS

I would like to sincerely thank Prof. Silvia Ferrari who has been providing me with great guidance and advice during my research at Cornell University. Her kindness and assistance help me the most throughout my research. I also want to thank Prof.Bharath Hariharan for his advice and suggestions during the defense. There are many others I would like to acknowledge. Qingze Huo helps me with the implementation of the algorithm and he gives me useful suggestions on the reports and the thesis, and Chang helps me to set up the Unreal airport environment before the creation of the synthetic dataset, and he provides lots of advice and helps on my reports. Lastly, I would like to thank all other members in LISC, Pingping, Jake, Taylor, Yucheng, Jane, Hengye, Shi, Rui, Zhihao, Shenghao, Xinyu, Dongheng, for their feedbacks in the lab meetings and my practice presentation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND BACKGROUND

Nowadays more people take flights, and airports need control towers and hundreds of ground crew members to help instructing the airplanes to taxi. These infrastructures cost lots of resources and human labor. The circumstance motivates me to design an algorithm to let the airplane capture taxi instructions from the ground crew's gestures and follow the gestures automatically. Here, the thesis proposes an algorithm to automatically instruct the airplanes to taxi with fewer human support. It may help the airport operators to save budget on human labor in the future.

The algorithm consists of human action recognition step and a control strategy to instruct the airplanes. The human recognition is first introduced, followed by a control strategy. Human actions recognition is important for intelligent systems and applications, such as city surveillance systems. Understanding the human actions becomes an popular research topic recently. Human actions consist of various types of actions, which include many sudden change points while the person is performing even an everyday activities. Another characteristic of the human activity is the uncertainty of the change point. The person changes his or her action at an unknown timestamp. These two characteristics make the recognition of the human actions difficult to conquer. Most of the existing approaches are focused on the video sequences with single motion type. Even though there is a significant progress in this direction, the assumption is limiting as the approaches assume the action sequence would only contain single type of actions. Due to the two characteristics of the human activity, this assumption cannot hold. Also, currently there is no available

dataset including the guidance gestures of the ground crew agents. The thesis also created a synthetic dataset in Unreal environment to demonstrate the guidance gestures, and the gestures will be used as the input data for the proposed algorithm.

To generalize the assumption and to extend it to general human movement, the thesis focuses on the continuous motion sequence, and the its change points are unknown. Different templates matching methods are proposed to identify and detect the different types of the actions from an original motion sequence. In [20], a sliding-window method is to find the intervals of actions from the video clips available in THUMOS-14 dataset. Later many approaches, such as [26], [21], [13], adopted ideas of finding the correct intervals from videos clips, and they improved the performance by applying Convolutional Neural Networks features. [26] combines the complete actions and the incomplete action fragments into a neural network model to detect and recognize the personnel actions. [21] applied new segment-based and aggregation module in their neural network model to model long-range motion sequence. [13] detects temporal boundaries using neural networks features. The three approaches only focus on removing the redundant frames which are unrelated with the actions. The goal in this thesis is to distinguish multiple actions from continuous videos.

For the algorithm, one important task is to temporally segment the motion sequence into various motion clusters. Many existing works proposes different segmentation approaches using motion capture data. For instance, an earlier work, [2], applies PCA-based method to partition the motion capture data into different actions segments. The work addresses the possibility of segmenting the motion capture data, but the proposed approach cannot distinguish similar

actions. Another segmentation approach, [27], introduced the aligned cluster analysis method to temporally group motion capture data into cycles of periodic motions, and then assigned to different motion classes. Recently, the paper [23] introduced a sparse subspace clustering method with geodesic exponential kernel to model the Riemannian manifold structure of human skeletons. These segmentation methods are primarily focus on 3D dataset. The method presented in this thesis will extends the data type as not only 3D data type but also other sensor modalities.

Based on various control signals, the control logic needs to handle the multiple scenarios under each signal. The paper [15] applied a hybrid ADP approach to a switching control system for determining the discrete and continuous control logic. In terms of target finding, Another paper [11] applied and updated a decision tree-based approach to adapt and to optimize the pursuit policies of the protagonist in the game. The optimization approach enabled the character to earn high scores, while preventing from being eaten by the enemies in the game. Instead of using decision-tree method, the paper [7] applied a connectivity graph to "hunt" for the targets in the broad game, *CLUE*. Regarding the human decision-making process, researchers at [17] addressed that human will adapt a "drop-the-worst" decision strategy while he or she is under time pressure and various conditions. This is an important insight on studying the human decision making against the machine decision making. Moreover, a study [18] applied Bayesian Inference to study the change of neural substrates due to the decision strategy under increased time pressure. Researchers at [14] applied an vision-guided control and planning method with the usage of convolutional neural networks (CNNs). To track the activities of the dynamic subject, the work [1] used a Bayesian approach to find the possible area of leak of methane in the

natural gas field, and information-theoretic approach to reduce the uncertainty of target source rate in the area of interest. In case the targets are moving, researchers in [22] adapted a geometric approach to improve the detect accuracy of the sensor towards the maneuvering targets, and the article formulated the kinodynamic constraints as an optimal control problem. Likewise, the work [16] applied a particle-filter information potential method to track the maneuvering targets. The paper [8] applied a cell-decomposition method to find the probability of the target and the cost of the operation of the sensors, and the article also proved the termination time is a function of the sensor parameters and of the numbers of detections. Finally, the article [24] summarized and compared various information-driven approaches of sensor planning and detection strategy in moving targets. For controlling the airplanes task, book [9] provided the basic information regarding the unmanned aerial vehicles and corresponding control strategies. The paper [10] introduced an optimal control method to track moving targets by using an omnidirectional sensor network. Finally, the article[25] proposed a probabilistic roadmap method to classify the fixed targets in an area of interest.

An efficient, human action recognition approach that automatically segment an untrimmed human motion sequence into disjoint groups and recognize each action group is proposed for this task. The input is an video sequence. The end results of the algorithm are the change points of the motion group. As the first step in the algorithm, the pose estimation of the skeleton structure is performed, and it will extract joint information of the personnel skeleton, which will act as the features for the segmentation step. The segmentation step is then proposed based on the multiple change point analysis to detect the transition intervals between distinct actions. Third, the temporal templates of each action clusters,

segmented from the previous step, are calculated and matched to the motion classes. Finally, a control logic is applied on the airplane system as to control the airplane based on the signals.

The outline of this thesis is given below. In chapter II, the problem will be formulated and presented. In chapter III, the recognition and segmentation framework will be shown. In chapter IV, experiments results of the proposed algorithm will be presented. In chapter V, the conclusion and the future developments will be presented.

CHAPTER 2

PROBLEM FORMULATIONS AND ASSUMPTIONS

This thesis considered the problem of automatic guidance to instruct the airplanes to taxi based on human actions segmentation and recognition from untrimmed human action videos. The action set is extracted from the action videos of the ground crew members. The action set is defined as $S = \{a_1, a_2, .., a_k\}$, where a_i is denoted as the i_{th} action label. The motion sequence with video frames f_i is denoted as $V = [f_1, f_2, \ldots, f_n]$, where n represents the total number of frames of a video. The proposed algorithm has two objectives to accomplish: the first objective is to segment and recognize the guidance gestures performed by a ground crew member in the airport, and the result of this objective is to output a control signal. The second objective is to control the airplane based on the signals captured from the guidance gestures performed by the ground crew member. One assumption in the thesis is that all actions are periodic actions. Regarding the human's actions, the thesis considers both typical taxiing activities and human exercises. For the motion sequences, each action is repeated and forms cycles for several times. The thesis also assumes that only a single agent performs the actions. Regarding the airplane taxiing movements based on the guidance gestures, the thesis assumes the airplane can have a view on the human subject, so the gestures of the ground crew member could be successfully captured by the airplane.

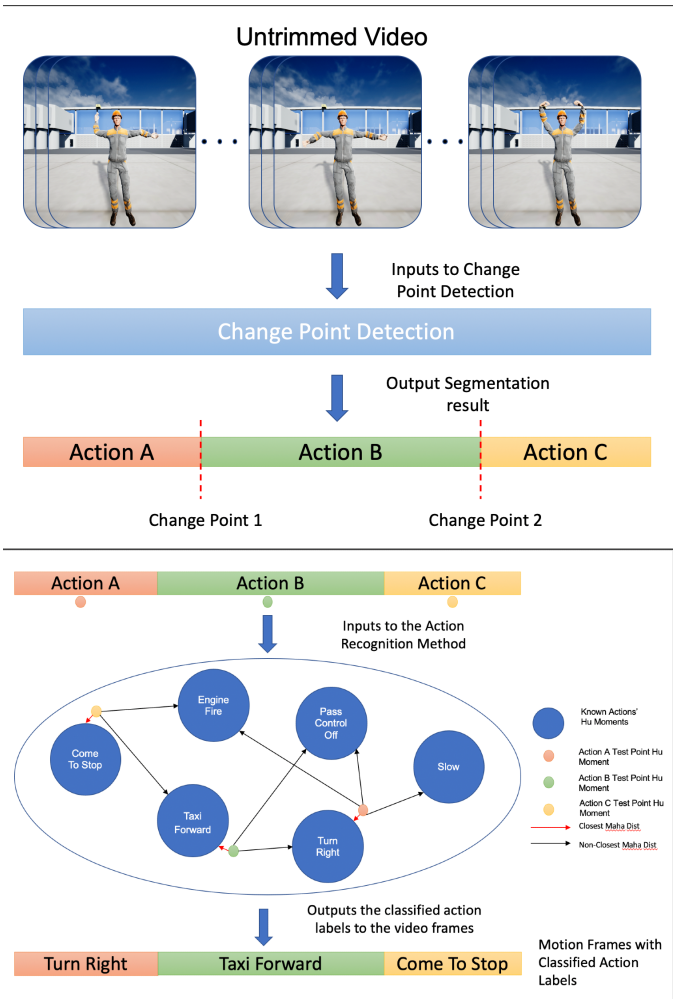

Figure 2.1: Algorithm Process

The figure introduces an overview of the human action segmentation and recognition algorithm.

Figure 2.1 introduces two steps of the human actions recognition process: the first step is actions sequence segmentation step, followed by the action recognition step for each group segmented by the previous step. The inputs of the process are untrimmed video clips, and the output of the process is a vector containing classified action labels.

Firstly, motion features, denoted as V, are computed for the untrimmed

video. Secondly, the change points, denoted as *cp*, are discovered, and lastly the motion sequence will be temporally segmented based on the change points, and accuracy of the classification will be evaluated after the last step. A *change point* between two actions is defined, denoted as τ. The segmentation process is decomposed below: Given a vector of change points, $\boldsymbol{\tau} = [\tau_1, \tau_2, \ldots, \tau_k]^T$, where k represents the number of change points. The frames of the motion sequence are separately grouped into distinct time windows. A *time window* is defined as the range between two neighboring change point: $\mathbf{w} = [\tau_{i-1}, \tau_i]$. The ideal result would be time windows not sharing change points, meaning, all the segmented time windows have distinct boundaries, and all the frames within a time window will be classified and recognized by the segmentation step.

In this thesis, 2D videos will be used as the input for the proposed algorithm. For a video, $\iota = \{I_t | t = 1, , , , , n\}$ taken by a camera. The joint motion information, such as positions, velocities, of the agent's skeleton is extracted from each frame of the video. Positions of the joints are calculated by extracting and calculating the different displacements of the pixels corresponding to the joints between consecutive frames. Specifically, the velocity of i_{th} joints of the agent's skeleton, denoted as $\mathbf{u}$, is calculated as

$$\boldsymbol{u_{t,i}} = [x_{t,i} - x_{t-1,i}, y_{t,i} - y_{t-1,i}] \tag{2.1}$$

where $[x_{t,i}, y_{t,i}]$ is a tuple of positions of the i_{th} joint at the time t in the image. For each image, all the velocities of the joints are concatenated to form a joint velocity (ordered) set, as denoted $U_\iota = \{\boldsymbol{u_t} | t = 1, ..., n\}$, where n is the total number of image frames of the motion sequence.

The velocities of each joint serve as important features for the actions segmentation process. The segmentation process is essential for finding the distinct groups of the human actions and gathering the same action groups together. The action recognition process is then applied on the segment to discover the segmented actions. The output of the process will be used to guide the airplane.

The system of the airplane is assumed to have various movement modes $\xi = [1, \ldots, E]$, where E is a discrete integer. The discrete control ν selects the next system mode, such that $\xi, \nu \in \epsilon$. The switched dynamic system is selected:

$$x(k+1) = \boldsymbol{f}_{\xi}[x_{\xi}(k), \boldsymbol{u}_{\xi}(k)] \tag{2.2}$$

where the discrete control law is

$$\xi(k+1) = \nu(k) \tag{2.3}$$

where $\mathbf{x} \in X \subset \mathbb{R}^n$ is the continuous state. X is the state space, $\mathbf{u}_{\xi} \in \mathcal{U}_{\xi} \subset \mathbb{R}^m$ is the continuous control input, and $\mathcal{U}_{\xi}$ is the space of admissible control inputs for mode ξ. The initial state $x(0) = x_0$ and mode $\xi(0) = \xi_0$ are assumed given, and the final time N is known and finite. The system also obeys the following assumptions:

Assumption 1: Mode switching can occur at any time step k and is determined solely by ν with zero cost.

Assumption 2: The state X is fully observable and the measurement error is negligible

The continuous and discrete control laws are presented as below:

$$u_\xi = \mathbf{c}_\xi[x(k), k] \tag{2.4}$$

$$\nu(k) = a[x(k), \xi(k), k] \tag{2.5}$$

The system performance is represented by cost function:

$$J = \phi[\boldsymbol{x}(N)] + \sum_{j=0}^{N-1} \mathcal{L}_\xi[x(j), \boldsymbol{u}_\xi(j), \nu(j)] \tag{2.6}$$

where $\mathcal{L}_\xi[x(j), \boldsymbol{u}_\xi(j), \nu(j)]$ is the cost function of the system, the detail of the cost function will be introduced in the Chapter IV.

In this module, the problem is formulated, and the background and the assumptions are also introduced in the module. In the next modules, the algorithm background of the segmentation and recognition will be introduced below.

CHAPTER 3

ALGORITHM BACKGROUND

the Multiple Change Point Analysis [6] is studied to separate the actions sequence into distinct groups. The MCPA method is a robust and efficient offline change point detection algorithm, which outperforms widely-used binary change point detection methods in computation complexity and detection robustness. The method focuses on finding the change points of the 1-D time-series data as the input. The method models the data generating process as a segment-wise autoregression. It addresses the problem of detecting the change points in one-dimensional time series data with theoretical guarantee. It achieves fast segmentation by segment-wise autoregression process. The autoregression process consists of various segments, each is modeled by the autoregression model. Then, the autoregression model is transformed into multivariate time-series data, and a multi-windows method is proposed to discover the structure changes effectively. The MCPA also proves that a Bayesian Information Criterion (BIC) gives a strong consistent selection of the optimal number of change points.

Given the one-dimensional time-series data with n elements $\mathbf{x} = [x_1 x_2 \dots x_n]$, MCPA partitions the actions sequence into k distinct groups, each group has same length. Each partition is created using an autoregression model (AR):

$$x_i = \epsilon_{i,q} + [1 \;\; x_{i-1} \;\; \dots \;\; x_{i-m}]\boldsymbol{\phi_q} \tag{3.1}$$

In the equation, m means how many preceding terms are in the AR model. $i \in [s_{q-1} + 1, s_q]$, which is the range of segment. $\epsilon_{i,q}$ is independent Gaussian

noise. $\boldsymbol{\phi_q} = [c \;\; a_1 \;\; a_2 \;\; \ldots \;\; a_m]^T$ is a vector consists of parameters of AR model, which can be fitted using least square method:

$$\arg\min_{\boldsymbol{\phi_q}} \sum_{t=s_{q-1}+1}^{s_q} (x_i - [1 \;\; x_{i-1} \;\; \ldots \;\; x_{i-m}]\boldsymbol{\phi_q})^2 \tag{3.2}$$

The author also transformed the other parameters of AR model, which is in form $[\boldsymbol{\phi_1}, \boldsymbol{\phi_2}, \ldots, \boldsymbol{\phi_k}]$

A multiple time windows are used in MCPA method as to find the accurate estimations of the change points. The method detects the change points for each time window and merges all the information. For each window size, the similar partitions will cluster together. The change points are defined as the boundary points between different and distinct groups of actions. The change point will then be identified between different clusters, and the boundary will be scored by 1. Moreover, to avoid the tendency of choosing small clusters, a BIC-like penalty term β is introduced to penalize the small ranges. Finally all the scores are accumulated together as to form a score vector s, which represents the final score. A region with highest score will have the highest possibility of including a change point. The clustering is achieved by minimizing the within-group variance.

$$\arg\min_{\{L_1,\ldots,L_h\}} \sum_{l=1}^{h} \lambda(\boldsymbol{\phi_{L_{l-1}+1}}, \ldots, \boldsymbol{\phi_{L_l}}) \tag{3.3}$$

where

$$\lambda(\boldsymbol{\phi_{L_{l-1}+1}}, \ldots, \boldsymbol{\phi_{L_l}}) = \sum_{q=L_{l-1}+1}^{L_l} |\boldsymbol{\phi_q} - \overline{\boldsymbol{\phi}}_l|^2 \tag{3.4}$$

Finally, the scores from each window size are accumulated together to form a final score. A region associated with high score is likely to contain a change point. In the next module, the methodology of the proposed algorithm will be introduced.

CHAPTER 4

METHODOLOGY

The proposed algorithm framework is systematically developed in this section. First, features from human actions are extracted. Second, action change point detection is deployed. Thirdly, the temporal templates method is applied to classify the segments, and the output will be used as the control signals of the control logic of the airplane, and finally the control of the airplane is introduced to guide the airplane to move based on the control signals of the guidance.

4.0.1 Human Action Features Representations

In this section, the thesis primarily focuses on the skeletal structure of human personnel. The skeletal structure is a good human action feature since it is easy to detect the changes in joints' positions of the human, which can be used for one of the features of the human actions [12]. Note that this thesis uses the joints around the personnel's hip, waist, shoulder, arms, legs, and neck since all changes of the actions this thesis includes occur around these joints. For this algorithm, the human skeleton joints' data is used to reflect the ground crew member's movements, specifically, their guidance gestures. The skeletal structure is beneficial for human action features since it is easy to detect the changes in joints' positions of the human, which can be used for one of the features of the human actions. The skeleton joint distribution is shown below

The blue dots in Figure 4.1 are the selected human joints points for movement information extraction. The joints velocities are used to characterize the personnel's actions. The joints positions and velocities are measured in both the

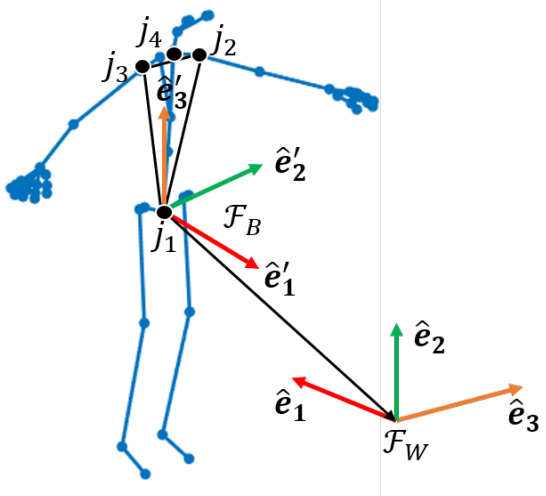

Figure 4.1: Human Skeleton Distribution

fixed frame $\mathcal{F}_{\mathcal{W}}$ and the body frame $\mathcal{F}_{\mathcal{B}}$. Velocities measured in both frames serve different but important roles. The velocities measured in $\mathcal{F}_{\mathcal{W}}$ are distinguishable features for various movements occurred, and the velocities measured in frame $\mathcal{F}_{\mathcal{B}}$ are effective features used to separate discriminate motions involving locomotion. Both origins of $\mathcal{F}_{\mathcal{W}}$ and $\mathcal{F}_{\mathcal{B}}$ are located in j_1, which is the hip joint of the human skeleton. The basis directions are fixed during the movement of the personnel. The basis vector, denoted as $\hat{e}'_1$, is perpendicular to the framed formed by the two shoulder joints, (j_2, j_3) and the hip joint, j_1.

$$\hat{\mathbf{e}}'_{\mathbf{1}} = \frac{\mathbf{p}_{\mathbf{j_1,j_2}} \times \mathbf{p}_{\mathbf{j_1,j_2}}}{||\mathbf{p}_{\mathbf{j_1,j_2}} \times \mathbf{p}_{\mathbf{j_1,j_2}}||} \tag{4.1}$$

and basis vector $\hat{e}'_3$ is the projection of relative position $\mathbf{p}_{\mathbf{j_1,j_4}}$ on the plane:

$$\hat{\mathbf{e}}'_{\mathbf{3}} = \frac{\mathbf{p}_{\mathbf{j_1,j_4}} - (\mathbf{p}_{\mathbf{j_1,j_4}} \cdot \hat{\mathbf{e}}'_{\mathbf{1}})\hat{\mathbf{e}}'_{\mathbf{1}}}{||\mathbf{p}_{\mathbf{j_1,j_4}} - (\mathbf{p}_{\mathbf{j_1,j_4}} \cdot \hat{\mathbf{e}}'_{\mathbf{1}})\hat{\mathbf{e}}'_{\mathbf{1}}||} \tag{4.2}$$

the basis vector $\hat{e}_2^{'}$ is the outer product between the $\hat{e}_1^{'}$ and $\hat{e}_1^{'}$. In the fixed frame, the definition of the movement sequence, $\mathcal{V} = \{\mathbf{v_t}|t = 1, \ldots, n\}$ at time t is shown below:

$$\mathbf{v}_t = [\mathbf{v}_{t,x}^T \quad \mathbf{v}_{t,y}^T]^T \tag{4.3}$$

where $\mathbf{v}_t \in \mathbb{R}^{2N\times 1}$, $\mathbf{v}_{t,x} \in \mathbb{R}^{N\times 1}$ and $\mathbf{v}_{t,y} \in \mathbb{R}^{N\times 1}$ and N is denoted as the number of joints of the skeleton. Specifically, the velocity component at x-axis, $\mathbf{v}_{t,x}$ includes all the velocity vectors of all the joints at x-axis:

$$\mathbf{v}_{t,x} = [v_{t,x_1} \;\; v_{t,x_2} \;\; \ldots \;\; v_{t,x_N}]^T \tag{4.4}$$

Similarly, the velocity set in the body frame $\mathcal{V}_\mathcal{B}$ is $\{\mathbf{v_{t,\mathcal{B}}}|t = 1, \ldots, n\}$ The form of the set $\mathcal{V}_\mathcal{B}$ is the same with the $\mathcal{V}$. The next step of the algorithm is to find the changes of actions of the skeleton, using the velocity sets as inputs. The change point detection is to find the different patterns between different groups of actions and group the similar ones together. In the change point detection step, the change points are the boundary points between different groups. The detail on the change point detection method will be introduced in the following section.

4.0.2 Action Detection and Segmentation

In this module, the segmentation step is developed on the MCPA method, but it extends it to adopt multi-dimensional time series data. The method models the

joints' velocities sets extracted from the personnel's skeleton. The model used in the algorithm is autoregression model. The module will first introduced the 1-dimensional time series data, then it introduces the multi-dimensional segmentation process.

Given the one-dimensional time-series data with n elements $\mathbf{x} = [x_1, \ x_2 \ \ldots \ x_n]$, the MCPA method partitions the actions sequence into k distinct groups, each group has same length. Each partition is created using an autoregression model (AR):

$$x_i = \epsilon_{i,q} + [1 \ x_{i-1} \ \ldots \ x_{i-m}]\boldsymbol{\phi_q} \tag{4.5}$$

In the equation, m means how many preceding terms are in the AR model. $i \in [s_{q-1} + 1, s_q]$, which is the range of segment. $\epsilon_{i,q}$ is independent Gaussian noise. $\boldsymbol{\phi_q} = [c \ \ a_1 \ a_2 \ \ldots \ a_m]^T$ is a vector consists of parameters of AR model, which can be fitted using least square method:

$$\arg\min_{\boldsymbol{\phi_q}} \sum_{t=s_{q-1}+1}^{s_q} (x_i - [1 \ x_{i-1} \ \ldots \ x_{i-m}]\boldsymbol{\phi_q})^2 \tag{4.6}$$

The author also transformed the other parameters of AR model, which is in form $[\boldsymbol{\phi_1}, \boldsymbol{\phi_1}, \ldots, \boldsymbol{\phi_k}]$

A multiple time windows are proposed in MCPA method as to find the accurate estimations of change points. The method detects the change points for each time window and merge all the information. For each window size, the similar partitions will cluster together. The change points are defined as the

boundary points between different and distinct groups of actions. The change point will then be identified between different clusters, and the boundary will be scored by 1. Moreover, to avoid the tendency of choosing small clusters, a BIC-like penalty term β is introduced to penalize the small ranges. Finally all the scores are accumulated together as to form a score vector s, which represent the final score. A region with highest score will have the highest possibility of including a change point. The clustering is achieved by minimizing the within-group variance.

$$\arg \min_{\{L_1,\ldots,L_h\}} \sum_{l=1}^{h} \lambda(\boldsymbol{\phi}_{\boldsymbol{L_{l-1}+1}}, \ldots, \boldsymbol{\phi}_{\boldsymbol{L_l}}) \tag{4.7}$$

where

$$\lambda(\boldsymbol{\phi}_{\boldsymbol{L_{l-1}+1}}, \ldots, \boldsymbol{\phi}_{\boldsymbol{L_l}}) = \sum_{q=L_{l-1}+1}^{L_l} |\boldsymbol{\phi_q} - \overline{\boldsymbol{\phi}}_l|^2 \tag{4.8}$$

where $\{L_1, \ldots, L_h\}$ are the ranges of clusters and h represents number of clusters. $\overline{\boldsymbol{\phi}}_l$ is the average AR parameter in cluster l.

From the 1-dimensional MCPA algorithm, the proposed algorithm extends it to multi-dimensional time series data. Similarly, given the joints velocities set, $\mathcal{V}$, the extension divided the set into k groups using a time window ω. Each segment q of the set, each dimension of the segment is generated by an autoregressive model (AR). The velocities set is represented as below

$$\mathcal{V} = [v_x, v_y] \tag{4.9}$$

where v_x represents the joints' velocities on x axis, and v_y represents the

joints' velocities on y axis. The proposed algorithm used multi-autoregression model (MAR) to extend the MCPA functionality to the v. For instance, a joint's velocities on x-axis, v_x, its AR model is given by:

$$v_{t,x_i} = \epsilon_{q,x_i} + [1 \;\; v_{t-1,x_i} \;\; \dots \;\; v_{t-m,x_i}]\boldsymbol{\phi_{q,x_i}} \tag{4.10}$$

where ϵ_{q,x_i} represents the Gaussian noise term, t belongs to the range of the segment, $t \in [s_{q-1}+1, s_q]$. m means how many preceding terms in the model. $\boldsymbol{\phi_{q,x_i}}$ is the AR parameters needed for estimation. These parameters can be fitted using least square method:

$$\arg\min_{\boldsymbol{\phi_{q,x_i}}} \sum_{t=s_{q-1}+1}^{s_q} (v_{t,x_i} - [1 \;\; v_{t-1,x_i} \;..\; v_{t-m,x_i}]\boldsymbol{\phi_{q,x_i}})^2 \tag{4.11}$$

Secondly, the estimated AR parameters at other dimension, x_i, are merged together. The resulting AR parameters form a vector, $[\boldsymbol{\phi^T_{1,x_i}} \;\; \boldsymbol{\phi^T_{2,x_i}} \;\; \dots \;\; \boldsymbol{\phi^T_{k,x_i}}]$

Repeating the process for other dimensions and concatenating all the$\boldsymbol{\phi^T_{1,x_i}}$, the parameters vectors are merged together to form a larger vector. The $\boldsymbol{\psi} = [\boldsymbol{\psi_1} \;\; \boldsymbol{\psi_2} \;\; \dots \;\; \boldsymbol{\psi_k}]$. For example, a single $\boldsymbol{\psi_{x,k}}$ is expressed as below:

$$\psi_{x,k} = [\boldsymbol{\phi^T_{x_1,k}} \;\; \boldsymbol{\phi^T_{x_2,k}} \;\; \dots \;\; \boldsymbol{\phi^T_{x_N,k}}] \tag{4.12}$$

where the ψ_k is the formed by ones from other dimensions.

$$\psi_{\mathbf{k}} = [\psi_{\mathbf{x},\mathbf{k}}, \psi_{\mathbf{y},\mathbf{k}}] \tag{4.13}$$

where $\psi_{k,x} \in \mathbb{R}^{1\times Nm}$, $\psi_k \in \mathbb{R}^{1\times 2Nm}$. Now, the algorithm will express the original velocity set $\mathcal{V}$ by transformed AR parameters ψ. The multiple time windows are used to detect the action change points on ψ and the score vector s, $\boldsymbol{s}$, is obtained. An additional penalty term, R, is introduced to avoid the small segmentation.

$$R = \frac{b}{1 + \exp \sigma(l - l_{min})} \tag{4.14}$$

where l_{min} is the minimum critical length of undesirable segment length. b and σ are two scalar parameters that can be user-defined based on the length of the segment.

After obtaining the final score vector, $\mathbf{s}$, the algorithm performs a refined search on the score vector, especially focus on the peak regions of the score vector. The precise change points are located at the minimums of the kinematic energy of the personnel. The reason is that the personnel is stationary when he/she is about to change the actions.

4.0.3 Action Recognition with temporal templates

After segmenting the actions sequence, each distinct action group will be recognized by using an action recognition step. The action recognition method is a fundamental method in the field of human action recognition [3]. This method applies a time window to select the correct action from the movements sequence, and this can be improved by the actions segmentation step in the previous section. The action recognition method uses temporal templates, a fea-

ture consist of Motion Energy image (MEI) and Motion History image (MHI), to identify the action from a motion sequence.

Each pair of MEI and MHI represents a movement of the subject in the motion sequence. The MHI is a scalar-valued image where the more recent pixels are brighter. MHIs are used for representing motions in a movement sequence. MEI is a cumulative binary motion image, which shows the locations of the motion in the image. Intuitively, the pixel value on MEI is 1 if the same pixel value on MHI is nonzero. The MEI and MHI equations are shown below:

$$E_\tau(x, y, t) = \sum_{i=0}^{\tau-1} D(x, y, t - i) \tag{4.15}$$

$$H_\tau(x, y, t) = \begin{cases} \tau, & \text{if } D(x, y, t) = 1 \\ \max(0, H_\tau(x, y, t-1) - 1), & \text{otherwise} \end{cases} \tag{4.16}$$

where $D(x, y, t)$ is the image difference at a timestamp t, τ is the duration of the movement.

The shapes of the MEI and MHI are effectively representing the human action. The algorithm then calculates the first 7 Hu moments. $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_7]^T$ These Hu moments are invariant to the rotations, positions, and scale, so they are good features to identify the pattern, which is the correct action group. The first 7 Hu moments' definitions are shown below, note that the subscripts in the right equation represent the $(p + q)_{th}$ order:

$$\mu_1 = \eta_{20} + \eta_{02} \tag{4.17}$$

$$\mu_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{4.18}$$

$$\mu_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{4.19}$$

$$\mu_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{4.20}$$

$$\begin{aligned}\mu_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned}$$

$$\begin{aligned}\mu_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ + 4\eta_{11}(\eta_3 + \eta_{12})(\eta_{21} + \eta_{03})\end{aligned}$$

$$\mu_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

where η_{pq} represents the normalized for scale from the central moments μ_{pq}

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^\gamma} \tag{4.21}$$

where $\gamma = (p+q)/2+1$ and $(p+q) \geq 2$ The Hu moments are thus independent of orientation, scale, and rotation.

After computing the Hu moments, the algorithm develops a recognition scheme matching the distance between the moments of the test points and the those from the known training set. The metric used here is Mahalanobis Distances D, which measures the distance from a point to a distribution. The Mahalonbis Distance between the unknown testing action and the combined distribution of the known training set is shown below:

$$D = \sqrt{(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}_{\boldsymbol{s}})\mathbf{K}_s^{-1}(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}_{\boldsymbol{s}})^T} \tag{4.22}$$

where $\bar{\boldsymbol{\mu}}_{\boldsymbol{s}}$ represents the mean of the Hu moments of the known movements, and $\mathbf{K}_s$ is the Covariance matrix of the stored (training) movements. $\boldsymbol{\mu}$ is the Hu moments vector for a testing action. The testing action with the minimal Mahalanobis distance is the matched with the closet training data.

To find the action from the sequence with a time window, the proposed algorithm applies an estimate of the minimum and maximum duration of the action: τ_{min} and τ_{max}. To compute the MHI for movements range from τ_{min} to τ_{max}, set $\tau = \tau_{max}$ and calculate its H_τ. The MHI for the rest in the range can be obtained by a threshold H_τ:

$$H_{\tau-\Delta\tau}(x,y,t) = \begin{cases} H_\tau(x,y,t) - \Delta\tau, & \text{if } H_\tau(x,y,t-1) > \Delta\tau \\ 0, & \text{otherwise} \end{cases} \tag{4.23}$$

where $\Delta\tau$ is the time step. After the Hu moments are computed, the matching scheme uses a distance metric to match the moments of an unknown test points against the ones from the known training set. The metric used here is the Mahalanobis distance, which measures the distance between a point and a distribution. It is used between the test point and the training data for each of the duration within the range. The best duration τ_{best} has the smallest distance. The testing action with the closest Mahalanobis distance is matched with the training data. Mathematically, the testing action label y_t is selected based on training label y_k with the closest Mahalanobis Distance, D^*:

$$y_t = y_k \quad \text{for } D^* = \min D \tag{4.24}$$

where y_k is the action labels from the known actions in the training set.

$$y_k \in \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} \tag{4.25}$$

where y_1 denotes the ”Come To Stop” action label. y_2 denotes the ”Engine Fire” action label. y_3 denotes the ”Slow” action label. y_4 denotes the ”Pass Control Off” action label. y_5 denotes the ”Taxi Forward” action label. y_6 denotes the ”Turn Right” action label. Using the metric D, the testing action can be matched with the training action which has the closest distance to the testing action.

The recognition algorithm will be repeated until all the action groups are identified.

The thesis tested the action recognition algorithm on the gesture actions against a small training set, whose actions are a small portions of the whole action histories.

The precision is the criterion the thesis used to evaluate based on the correct classified labels against the total number of frames of the action video.

$$Precision = \frac{n_{TP}}{n_{TP} + n_{FP}} \text{x} 100\% \tag{4.26}$$

where the n_{TP} denotes the numbers of true positive labels, $n_{TP}+n_{FP}$ denotes

the total numbers of labels of the action sequence.

The detection accuracy is measured numerically using standard Precision and Recall metric. In the metric, n_{TP} represents number of correctly detected change points, n_{FP} represents number of falsely detected change points and n_{FN} serves as the number of mistakenly undetected change points. Precision is defined as the ratio of number of correct detection versus the total number of detection. Recall is defined as the ratio of number of correct detection versus the total number of change points.

After the evaluation, the recognition scheme would output a control signal, which will be applied in the next module. In the following module, the airplane system and the control logic will be introduced.

4.0.4 Auto Guidance Demo Switching Control Logic

After each action groups are segmented and recognized in the previous step, the output of the previous step will act as the control signals, the system will response differently to various modes computed from the previous step. The system consists of three action modes: the "Turn Right" mode, the "Taxi Forward" mode, and the "Come To Stop" mode. Each mode can be represented by a linear time-invariant (LTI) dynamics with a continuous state vector $\mathbf{x} = [x, y, \theta]^T$, where $\mathbf{x} \in \mathbb{R}$. θ denotes the orientation of the airplane. $\mathbf{x}$ is fully observable and error free. The mode of the system is represented by a discrete state variable $\xi \in \varepsilon$, where $\varepsilon = [1, 2, 3]$ and $\mathbf{u}_\xi = [u_1, u_2, u_3]^T$. Each mode denotes the movements of the airplane based on the identified ground grew gesture. $\xi = 1$ denotes the "Come To Stop" mode, $\xi = 2$ denotes the "Taxi Forward" mode,

and $\xi = 3$ denotes the "Turn Right" mode. The system dynamics under each mode is represented below:

$$\mathbf{x}(k+1) = \begin{cases} \mathbf{A}_1\mathbf{x}(k) + \mathbf{B}_1 u_1(k), & \text{for } \nu(k) = 1 \\ \mathbf{A}_2\mathbf{x}(k) + \mathbf{B}_2 u_2(k), & \text{for } \nu(k) = 2 \\ \mathbf{A}_3\mathbf{x}(k) + \mathbf{B}_3 u u_3(k), & \text{for } \nu(k) = 3 \end{cases} \tag{4.27}$$

where

$$u_1(k) = v_0 \Delta k + \frac{a(\Delta^2)}{2} \tag{4.28}$$

$$u_2(k) = v_k \Delta k \tag{4.29}$$

$$u_3(k) = \begin{bmatrix} -r_C \sin(\theta_0) \\ r_C \cos(\theta_0) \end{bmatrix} \tag{4.30}$$

where Δk is the time step, a represent the acceleration of the airplane v_0 denotes as the initial speed of the airplane, and θ_0 is the initial orientation of the airplane, and r_C is the radius of the arc trajectory between the initial state and a terminal point.

At any time $k \in \{0, \ldots, (N-1)\}$, the system mode ξ can be fully controlled at no cost by a switching signal $\nu \in \varepsilon$ provided by the discrete controller.

The paper [15] proposed a new ADP recurrence relationships and transver-

sality conditions for solving the switched optimal control problem. From Bellman's principle of optimality [4], the optimization of the objective function (2.6) can be embedded in the optimization of a switched system value function or *cost-to-go* which, at any time k, is defined as:

$$V[\boldsymbol{x}(k), \xi(k), \pi, k]\phi[\boldsymbol{x}(N)] + \sum_{j=k}^{N-1} \mathcal{L}_{\xi}[x(j), \boldsymbol{u}_{\xi}(j)] \tag{4.31}$$

From the definition above, the value function obeys the recurrence relationship

$$V[\boldsymbol{x}(k), \xi(k), \pi, k] = \mathcal{L}_{\xi}[\boldsymbol{x}(k), \boldsymbol{u}_{\xi}(k)] + V[x(k+1), \xi(k+1), \pi, k+1] \tag{4.32}$$

The system depends on both continuous and discrete state and control inputs. The cost function to be minimized is represented by:

$$J = \boldsymbol{x}^T(N)\boldsymbol{P}_f\boldsymbol{x}(N) + \sum_{j=0}^{N-1} x^T(j)\boldsymbol{Q}_{\xi}x(j) + u_{\xi}^T(j)\boldsymbol{R}_{\xi}u_{\xi}(j) \tag{4.33}$$

where $\boldsymbol{P}_f$ represents the terminal cost matrix, $\boldsymbol{Q}_{\xi}$ and $\boldsymbol{R}_{\xi}$ are the weighting matrices for each mode

From [19], the switched differential Riccati Equation is given by:

$$\boldsymbol{P}(k-1) - \boldsymbol{Q}_\xi = \boldsymbol{A}_\xi^T(\boldsymbol{P}(k) - \boldsymbol{P}(k)\boldsymbol{B}_\xi(R_\xi + \boldsymbol{B}_\xi^T\boldsymbol{P}(k)\boldsymbol{B}_\xi)^{-1}\boldsymbol{B}_\xi^T\boldsymbol{P}(k))\boldsymbol{A}_\xi \quad (4.34)$$

where the discrete control law is obtained by minimizing the Hamiltonian function, such that:

$$\nu(k) = \arg\min_{\nu}\{H[\boldsymbol{P}(k), x(k), \xi(k), u(k)]\} \quad (4.35)$$

where the Hamiltonian function H is defined as:

$$\begin{aligned} H = \mathcal{L}_\xi[\boldsymbol{x}(k), \boldsymbol{u}_\xi(k)] + \boldsymbol{\lambda}[x(k+1), \nu(k), k+1]\boldsymbol{f}_\xi[x(k), \boldsymbol{u}_\xi(k)] \\ = H[x, \boldsymbol{u}_\xi, \boldsymbol{\lambda}, \nu, k] \end{aligned}$$

where $\boldsymbol{\lambda}$ represents the gradient of the value function with respect to the state, such that

$$\boldsymbol{\lambda} = \frac{\delta V}{\delta x} \quad (4.36)$$

In this chapter, it introduces the skeleton feature representation, which is the input of the segmentation step. Additionally, the chapter also explains the three parts of the proposed algorithm, including the actions segmentation, action recognition, and aircraft control logic. Each step's equations are given in

details in this chapter. In the next chapter, the experimental results on each step will be present, and two demos are created to illustrate the practical application of the airplane taxiing based on the recognition results of the ground crew member.

CHAPTER 5

EXPERIMENTS ON UNREAL SYNTHETIC DATASET

5.0.1 Motion Segmentation using MCPA

In this chapter, experiments conducted on the Unreal Synthetic dataset, and the results of the MCPA and action recognition algorithm will be shown. The videos in this database are generated by using 12 motion-capture cameras around different people subjects, and the shooting angles for each action video range from 0 degree to 330 degree in counter-clockwise direction. The thesis simulated 7 subjects and 10 actions for each of them. In order to obtain joints' velocities of the animations, the proposed algorithm used a public toolbox [5, **?**] to compute the joint positions from the action videos. The results of the extracted velocities are shown below

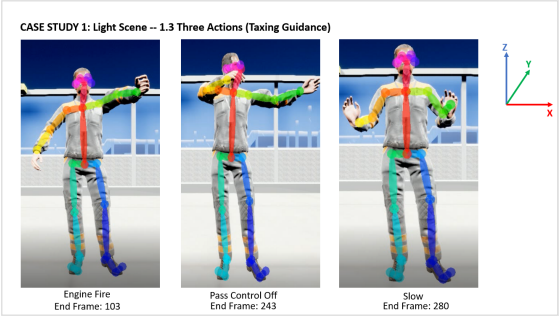

Figure 5.1: Three Actions from Ground Taxiing Guidance

Figure 5.1 shows the extracted joints of the ground crew member in the airport at daylight environment. The joints are the dots on the crew member, and the links connected by the dots in the crew member are the arms/legs of the personnel. Each joint is recording the its position and the velocities at each frame.

So the joints velocities history can be used as the input to the Motion Segmentation process.

The proposed algorithm applied the motions segmentation step on the video, and the segmentation result of three taxiing gestures sequence is shown below:

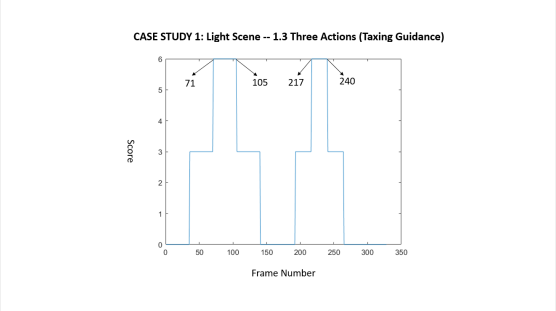

Figure 5.2: MCPA results from three actions of taxiing guidance

Fig 5.2 shows the MCPA result of the three actions of the taxiing guidance, and note that the two true change points, which are at 104_{th} and 243_{rd} frame, are covered inside the peaks of the score plots. The peaks are the limits separating the sequence into three distinct groups.

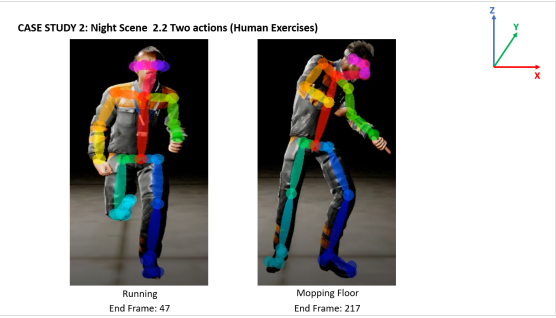

Figure 5.3: Joint Extraction Two Human Exercises

Figure 5.3 shows the extracted joints of the crew member in the airport at night. To visualize the motion better, a light is applied on the personnel. The

joints store the velocities history, which will be used as the inputs to the Motion Segmentation step.

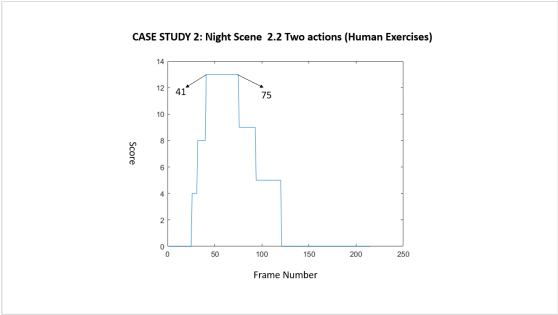

Figure 5.4: MCPA results from two actions of Human Exercises

Fig 5.4 shows the MCPA results on two human exercises under the light scene. Note that only one ground true change point, which is 42_{nd} frame, is included in the peaks of the score plot.

More test cases are used to test the effectiveness of the Motion Segmentation process. Here the three human exercises are used in the motion sequence.

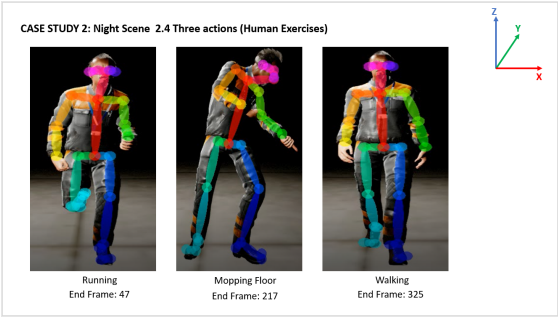

Figure 5.5: Joint Extraction of three actions of Human Exercises

Figure 5.5 shows the extracted joints of three human exercises of the ground crew member at night environment. The figure also shows the ground true change points, which are 48_{th} and 218_{th} frame.

The segmentation result is displayed below

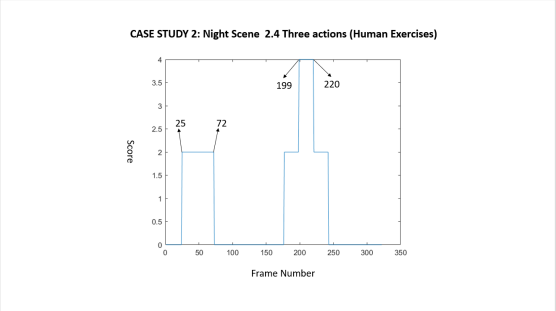

Figure 5.6: MCPA results from three human exercises at night

Figure 5.6 shows the segmentation result on three human exercises at night. The segmentation has two peaks, the first peak starts from the 25_{th} frame to 72_{nd} frame and the second one ranges from 199_{th} to 220_{th} frame. The peaks can effectively include the ground true change points for the sequence. The two peaks can also act as the separation criteria to segment the sequence.

For the Multi-dimensional motion segmentation step in the proposed algorithm, the algorithm can yield the correct result with small angle variations. The input data is the everyday human exercises actions (running, mopping floor, walking, and waving). Each action is repeated for five repetitions. The four actions are concatenated to form a continuous motion sequence. OpenPose toolbox[9] is used to compute the joint positions of the person subject. The outliers of the joint positions are removed by using Savitzky-Golay filter for the pose estimation.

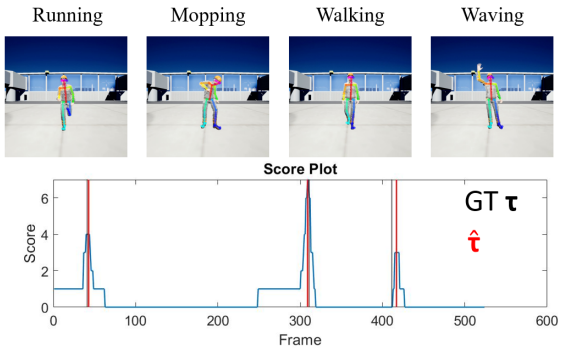

Figure 5.7: MCPA results Human Exercises at 0 degree

Fig 5.7 shows the segmentation results towards the personnel. The red lines are the predicted change points whose kinetic energy are at the peaks, and the black lines are the ground true change points. Two neighbors segments on two sides of a change point will be scored by one of each window sizes, and final score s is the accumulation for all of the windows. The three peak scores segment the actions into 4 groups, and each action group is identified by using the matching scheme with temporal templates. The GT τ represents the ground true change point, and the red $\hat{\tau}$ means the detected change points.

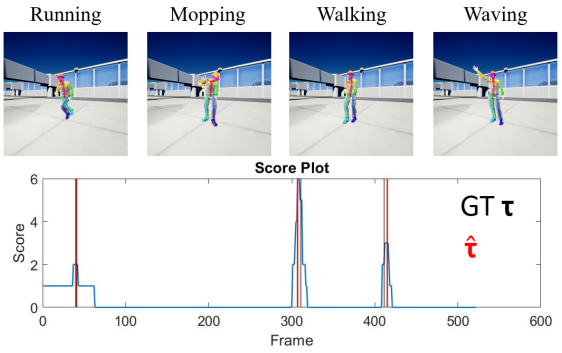

Figure 5.8: MCPA results Human Exercises at 30 degree

Fig 5.8 shows segmentation results of the crew personnel under under 30 degree view angle in counter clockwise direction. The red lines are the predicted change points whose kinetic energy are at the peaks, and the black lines are the

ground true change points. The sequence is separated by the segmentation step into four distinct groups. Then, each group is recognized by the action recognition scheme using temporal templates. The matched action class is shown above the score plot. The GT τ is the ground true time, and the red τ is the detected time. The figure shows the detected ones are matched with the ground true time.

the window size for MCPA algorithm is $w = [31, 6, 4, 3]$, and applied the algorithm to the video clips on other view angles. The segmentation accuracy is evaluated by using precision and recall equations, which are given below:

$$\text{precision} = \frac{n_{TP}}{n_{TP} + n_{FP}} \tag{5.1}$$

$$\text{recall} = \frac{n_{TP}}{n_{TP} + n_{FN}} \tag{5.2}$$

Where n_{TP} represents the number of correctly detected change points, n_{FP} represents number of falsely detected change points, and n_{FN} is the number of mistakenly undetected change points.

Table 5.1 shows the MCPA results on the four human exercises actions with multiple view angles. The precision and recall accuracy show three out total four view angles has segmented all the actions correctly.

Table 5.1: Precision and Recall Accuracy of MCPA with view angle variants

Angle (degree)	Precision (%)	Recall (%)
0	100	100
30	100	100
60	100	100

5.0.2 Demo: Airport Ground Crew Guidance

In the ground crew gestures demo, the actions sequence consist of three actions. The segmentation process segments the actions sequence into three groups. Each group represents an action.

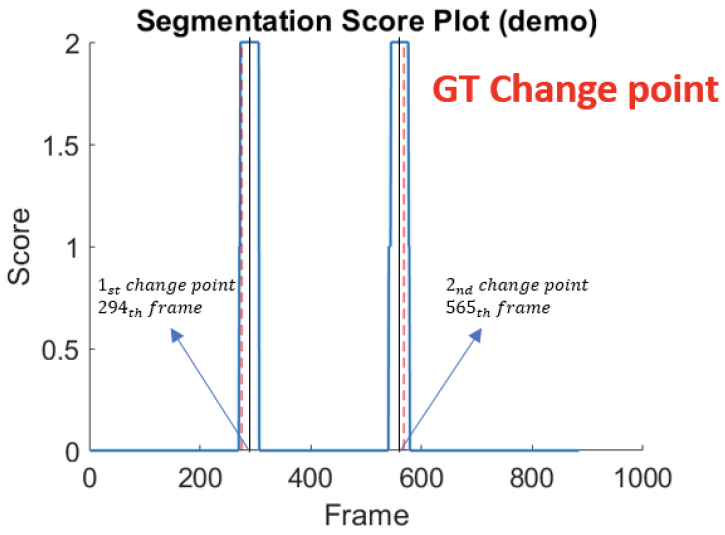

Figure 5.9: Segmentation Result of the actions for auto guidance demo

Fig 5.9 shows the segmentation result from the joints' velocities of the personnel in the video. There are two discovered change points, which are at 294_{th} and at 565_{th} frames respectively. The black dashed lines are the segmentation limit. The red dashed lines are the ground true change points. By this seg-

mentation process, two discovered change points separate the video into three segments, and each segment represents the action.

5.0.3 Action Recognition using Temporal Templates

For the experiment, the thesis used human exercises actions and all personnel subjects from Unreal Engine synthetic dataset and 9 view angles out of 12 view angles for each action of each personnel. The first step in the recognition frame work is to calculate the Motion-Energy and Motion-History images. There are six actions for the ground crew gestures: Come To Stop, Engine Fire, Pass Control Off, Slow, Taxi Forward, and Turn Right.

The MEIs of the ground crew gestures are listed below:

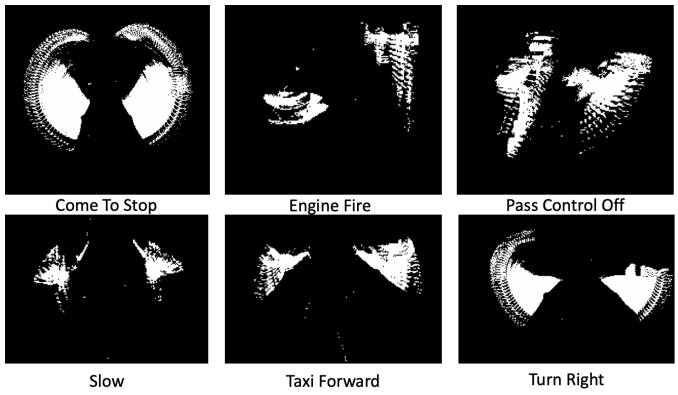

Figure 5.10: Motion Energy Images of the ground crew gestures

The MHIs of the ground crew gestures are presented below:

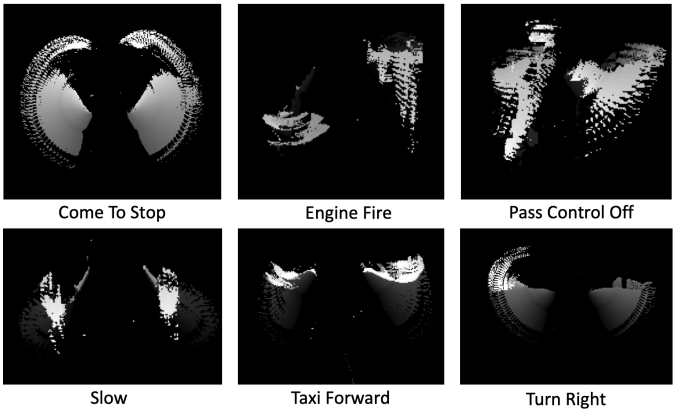

Figure 5.11: Motion History Images of the ground crew gestures

In the demo, the MEIs and the MHIs are computed, and their Hu moments are matched with the ones from the known actions in the training set. The distance metric is the Mahalanobis distance, and the lowest Mahalanobis distance is the matched action label. The action recognition is the second step for the AACD algorithm. The action sequence is separated into three groups. Each group is identified by the action recognition step of the algorithm. The Mahalanobis distances between the Hu moments of the segment and the ones from the training actions set are computed, and the action label with the closest Mahalanobis distance represents the matched action class.

Fig 5.12 shows the Mahalanobis distances of the Hu moments of the segment to the training set. The bar with the lowest distance is colored red, and it is the matched action label from the process. For the first segment, the classified action class is Turn Right.

Fig 5.13 For the second segment, the classified action class is Taxi Forward. The Mahalanobis distance is the smallest on the fifth action label. It shows that

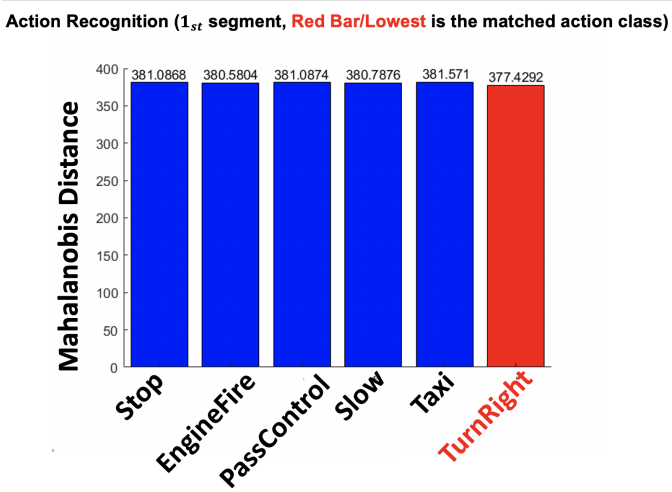

Figure 5.12: Action Recognition of first segment (Into Gate)

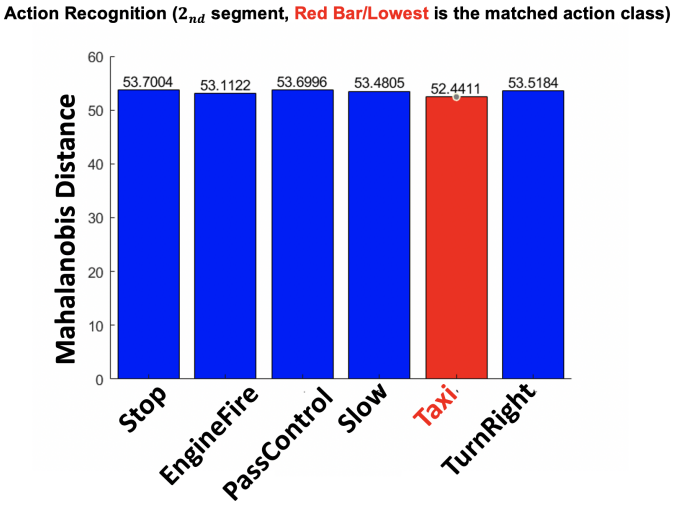

Figure 5.13: Action Recognition of second segment (Into Gate)

the recognized action label is matched with the ground true action label.

Figure 5.14 shows the action recognition result for the third group segmented by the motion segmentation step. The action class with the closest Mahalanobis distance is the first action class, which is ”Come To Stop”. The figure shows the detected action label is matched with the ground true action label.

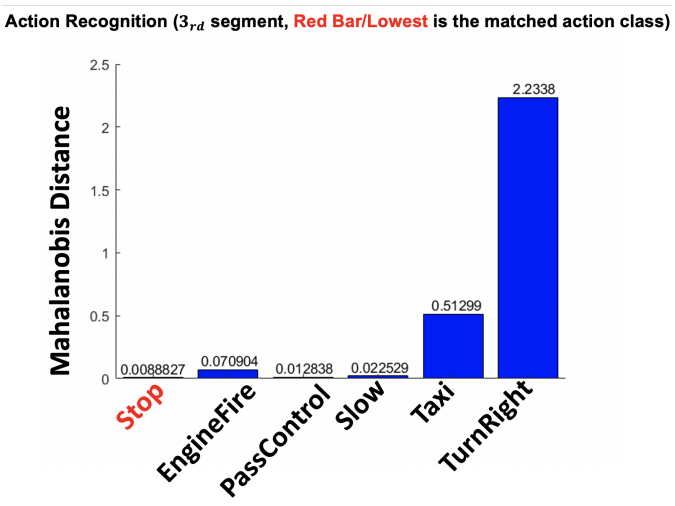

Figure 5.14: Action Recognition of third segment (Into Gate)

For the Out to Taxiway demo, the thesis applied the action recognition step for each action of the ground crew, the results are shown below:

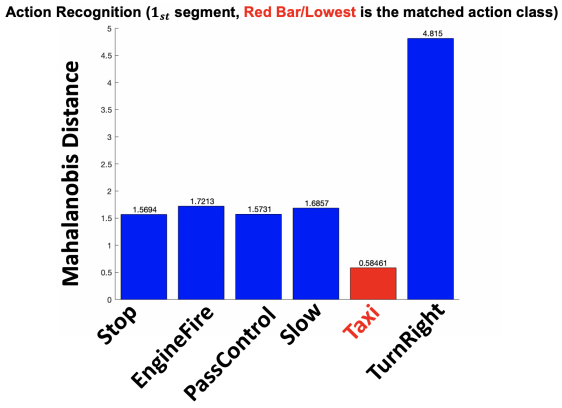

Figure 5.15: First Action Recognition Result (Out to Taxiway)

Figure 5.15 shows the action recognition result for the first action group. The action class with the smallest Mahalanobis distance is the Taxi action class, which is "Taxi Forward". The figure shows the detected action label is matched with the ground true action label.

Figure 5.16 shows the action recognition result for the second action group.

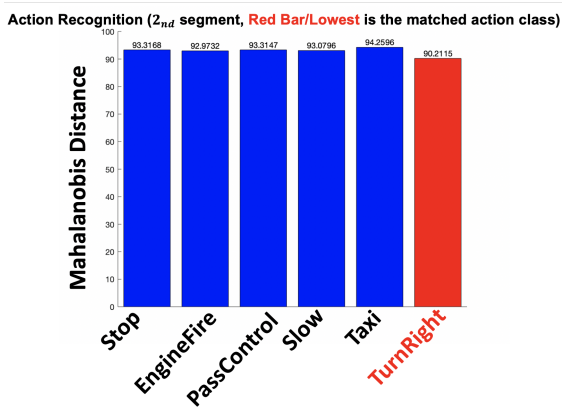

Figure 5.16: Second Action Recognition Result (Out to Taxiway)

The action class with smallest Mahalanobis distance is the ”Turn Right” action class. The figure shows the detected action label is matched with the ground true action label.

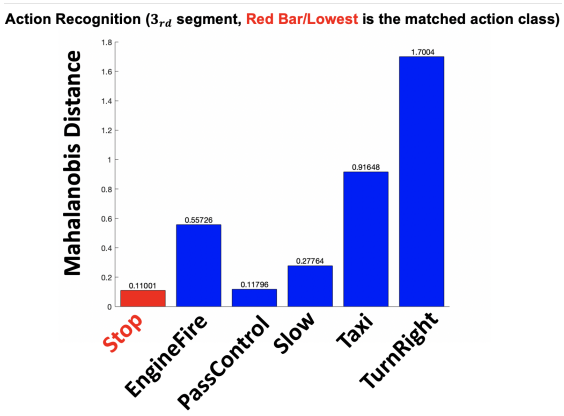

Figure 5.17: Third Action Recognition Result (Out to Taxiway)

Figure 5.17 shows the action recognition result for the third action group. The action class with smallest Mahalanobis distance is the ”Come To Stop” action class. The figure shows the detected action label is matched with the ground true action label.

Table 5.2 below summarize the classification result and the the ground true

result.

Table 5.2: Classification and the Ground true results of the segments

Segment	Classification Label	Ground True Label
1	Turn Right	Turn Right
2	Taxi Forward	Taxi Forward
3	Come To Stop	Come To Stop

5.0.4 Auto Guidance Demo

To demonstrate the control logic, the screenshots of the Into Gate demo video are shown below:

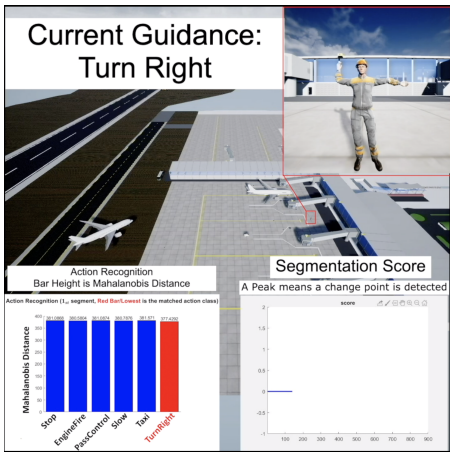

Figure 5.18: First Action in the Into Gate Demo

Figure 5.18 shows the first action in the sequence, it demonstrates the airplane is moving right while the ground crew member is doing the "Turn Right" Action, and the action recognition result gives the correct classification result.

Figure 5.19 shows the second group of the sequence. The airplane is mov-

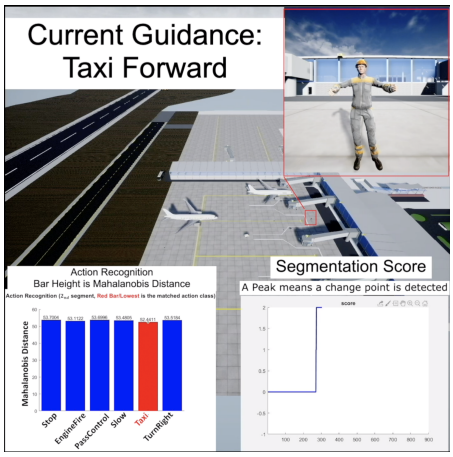

Figure 5.19: Second Action in the Into Gate Demo

ing towards the ground crew member with a constant speed. The ground crew member is also doing "Taxi Forward". The action recognition also gives a correct result.

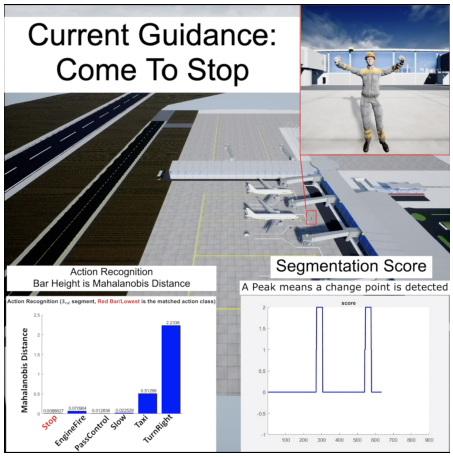

Figure 5.20: Third Action in the Into Gate Demo

Figure 5.20 is the last action segment of the sequence. The airplane is coming to the ground crew member with a decreasing speed, followed by the ground crew member's gesture. The action recognition method is showing the correct classification action.

The solutions of the system in this demo are obtained from the SDRE numerically with approach [19] is shown below:

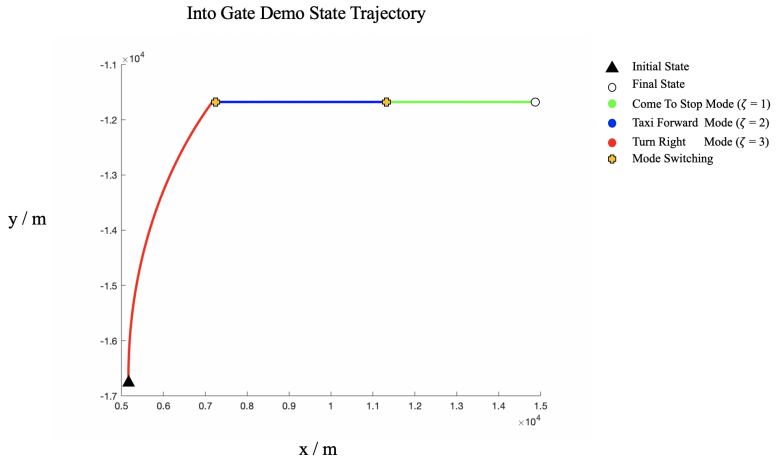

Figure 5.21: Demo Solutions from SDRE (Into Gate)

Figure 5.21 shows the switched dynamical Ricartti equation solutions of the into gate demo. The black solid triangle represents the initial state, the hollow black circle denotes the final state, the green dots line represents the first optimal mode, which is "Come To Stop" mode, the blue dots line represents the second optimal mode, which is "Engine Fire" mode, and the red dots line represents the third optimal mode, which is "Turn Right" mode. The orange crosses are the time when the control modes switched. The process starts with the "Turn Right" mode, followed by the "Taxi Forward" mode, and it is finished by the "Come To Stop" mode. The order of the movements matched with the correct order of the ground true sequence.

The thesis also introduced another demo that instruct the airplanes going out from the terminal gate to the taxiway for take-off. The actions sequence is ”Taxi Forward” – ”Turn Right” – ”Come To Stop”. The recognition and segmentation results of the sequence is shown in the figures below

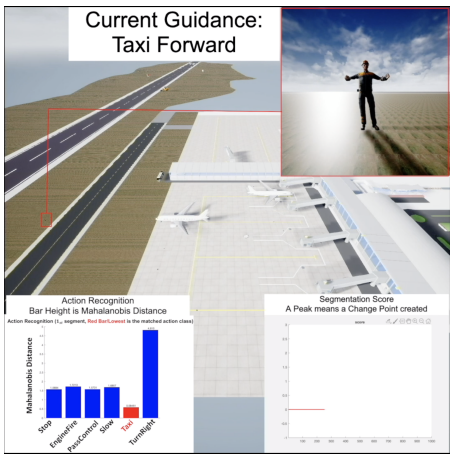

Figure 5.22: First action of the Out To Taxiway demo

Figure 5.22 shows the first action that the ground crew member performed, and the airplane captured his action as ”Taxi Forward” action, the red bar in the action recognition shows the action matched with the ground true action.

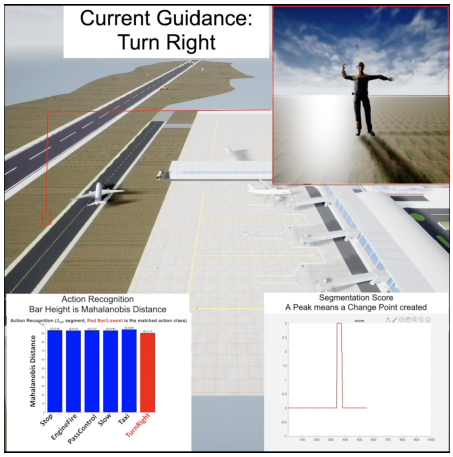

Figure 5.23: Second action of the Out To Taxiway demo

Figure 5.23 shows the second action, and the airplane captured the ground

crew member's action as "Turn Right", the red bar in the action recognition shows the action matched with the ground true action, and the segmentation step detects a structural change occurs in the ground crew member.

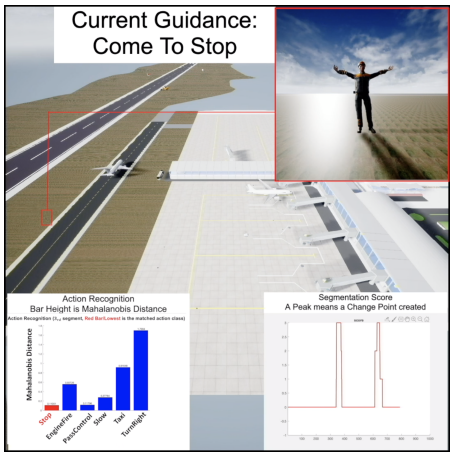

Figure 5.24: Third action of the Out To Taxiway demo

Figure 5.24 shows the last action that the ground crew member performed, and the airplane captured his action command as "Come to Stop", the red bar in the action recognition shows the action matched with the ground true action, and the segmentation step detects another structural change occurs in the ground crew member's guidance, meaning the ground crew member changed his action from one to a new action.

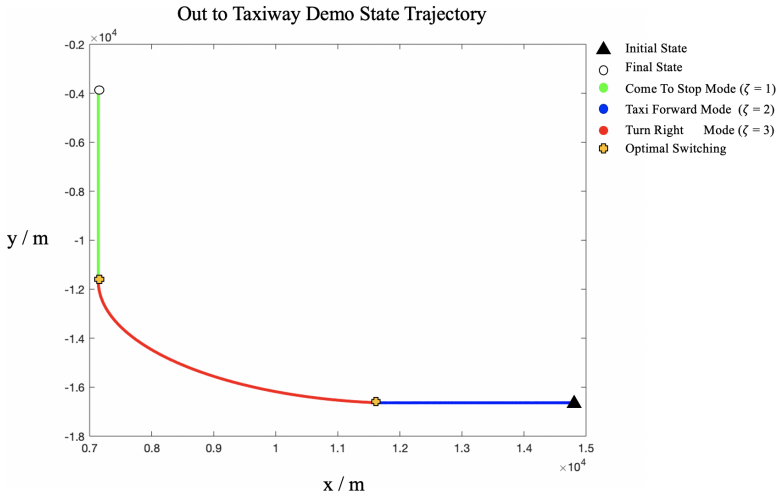

Figure 5.25: Demo Out To Taxiway Solutions from SDRE

The solutions of the system of the airplane from SRDE using approach [19] is shown above:

Figure 5.25 shows the switched dynamical Ricartti solutions of the second demo of out to taxi way. The black solid triangle represents the initial state, the hollow black circle denotes the final state, the green dots line represents the first optimal mode, which is ”Come To Stop” mode, the blue dots line represents the second optimal mode, which is ”Engine Fire” mode, and the red dots line represents the third optimal mode, which is”Turn Right” mode. It shows that the process starts with the third mode, which is the ”Turning Right” mode, and once it met with the first orange cross, which indicates that the system switches from the ”Turn Right” mode to the ”Taxi Forward” mode, and the system switched again when it meets with the second orange cross, and the system changed its movement from ”Taxi Forward” to ”Come To Stop” mode. The order of the movements of the airplane matches with the correct order from the ground true sequence.

CHAPTER 6

CONCLUSIONS AND FURTHER DEVELOPMENT

In this thesis, a framework of action segmentation and an action recognition with airplane control logic is proposed to control the airplane's taxiing movements based on the automatic recognition of the ground crew guidance gestures. The multi-dimensional change point detection method is developed to find the transition stage between the action groups in a motion sequence, and an action recognition method using temporal templates is employed in order to identify the action of the current group. Then, a methodology combining the two components is introduced in this thesis for automatic change points and action recognition task to segment the guidance sequence and identify the action of the segmented group. Then, a switching control logic is introduced in order to instruct the airplane based on the guidance gestures identified by the method. The proposed algorithm is evaluated on 3D motion sequences produced under Unreal Engine simulation environment. The results show that the framework can effectively segment and recognize actions within a small angle variation. Moreover, The airplane switching control logic is proposed to control the airplane's motion based on the framework results. Finally, Two demos of auto guidance of an airplane based on the segmentation and recognition results of a ground crew member are created in order to demonstrate the effectiveness of the algorithm. A Unreal Synthetic dataset of the ground crew member guidance gestures are created for future development.

One of the future extensions of the framework is to adapt to real-world action videos to guide the airplanes. Additionally, many real-world action videos are taken into various angles. Another future extension of the proposed algo-

rithm is to yield correct results for videos taken under different view angles. Another future work would focus on the interaction between multi-ground crew agents and multiple airplanes movements scenarios. Regarding the control logic, an analysis of the robustness and the complexity would be a future development for the framework, and various multi-dynamical systems would be suitable for the multi-agent scenario. Conducting some small-scaled real-world experiments would be a potential development. The future work on the framework could have a broad prospective and applications.

BIBLIOGRAPHY

[1] John. D. Albertson, Tierney Harvey, Greg Foderaro, Pingping Zhu, Xiaochi Zhou, Silvia Ferrari, M. Shahrooz Amin, Mark Modrak, Halley Brantley, and Eben D. Thoma. A mobile sensing approach for regional surveillance of fugitive methane emissions in oil and gas production. *Environmental Science & Technology*, 50(5):2487–2497, 2016. PMID: 26807713.

[2] Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K Hodgins, and Nancy S Pollard. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface 2004*, pages 185–194. Citeseer, 2004.

[3] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, 23(3):257–267, 2001.

[4] Michael Branicky, V.s Borkar, and Sanjoy Mitter. A unified framework for hybrid control: Model and optimal control theory. *Automatic Control, IEEE Transactions on*, 43:31–45, 01 1998.

[5] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*, 2018.

[6] Jie Ding, Yu Xiang, Lu Shen, and Vahid Tarokh. Multiple change point analysis: Fast implementation and strong consistency. *IEEE Transactions on Signal Processing*, 65(17):4495–4510, 2017.

[7] S. Ferrari and C. Cai. Information-driven search strategies in the board game of clue ^{r}. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(3):607–625, 2009.

[8] S. Ferrari, C. Cai, R. Fierro, and B. Perteet. A geometric optimization approach to detecting and intercepting dynamic targets. In *2007 American Control Conference*, pages 5316–5321, 2007.

[9] Silvia Ferrari and Thomas Allen Wettergren. *Information-driven planning and control: adaptive management of sensor networks*. CRC Press, 2017.

[10] G. Foderaro, P. Zhu, H. Wei, T. A. Wettergren, and S. Ferrari. Distributed

optimal control of sensor networks for dynamic target tracking. *IEEE Transactions on Control of Network Systems*, 5(1):142–153, 2018.

[11] Greg Foderaro, Ashleigh Swingler, and Silvia Ferrari. A model-based approach to optimizing ms . pac-man game strategies in real time 1 2.

[12] Fei Han, Brian Reily, William Hoff, and Hao Zhang. Space-time representation of people based on 3d skeletal data. *Comput. Vis. Image Underst.*, 158(C):85–105, May 2017.

[13] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898, 2019.

[14] Chang Liu and Silvia Ferrari. *Vision-guided Planning and Control for Autonomous Taxiing via Convolutional Neural Networks*.

[15] W. Lu, P. Zhu, and S. Ferrari. A hybrid-adaptive dynamic programming approach for the model-free control of nonlinear switched systems. *IEEE Transactions on Automatic Control*, 61(10):3203–3208, 2016.

[16] Wenjie lu, G. Zhang, S. Ferrari, M. Anderson, and R. Fierro. A particle-filter information potential method for tracking and monitoring maneuvering targets using a mobile sensor agent. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 11:47–58, 01 2014.

[17] Hanna Oh, Jeffrey M. Beck, Pingping Zhu, Marc A. Sommer, Silvia Ferrari, and Tobias Egner. Satisficing in split-second decision making is characterized by strategic cue discounting. *Journal of experimental psychology. Learning, memory, and cognition*, 42 12:1937–1956, 2016.

[18] Hanna Oh-Descher, Jeffrey Beck, Silvia Ferrari, Marc Sommer, and Tobias Egner. Probabilistic inference under time pressure leads to a cortical-to-subcortical shift in decision evidence integration. *NeuroImage*, 162, 09 2017.

[19] P. Riedinger, F. Kratz, C. Iung, and C. Zanne. Linear quadratic optimization for hybrid systems. In *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, volume 3, pages 3059–3064 vol.3, 1999.

[20] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine

grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1194–1201, 2012.

[21] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018.

[22] H. Wei and S. Ferrari. A geometric transversals approach to sensor motion planning for tracking maneuvering targets. *IEEE Transactions on Automatic Control*, 60(10):2773–2778, 2015.

[23] G. Xia, H. Sun, L. Feng, G. Zhang, and Y. Liu. Human motion segmentation via robust kernel sparse subspace clustering. *IEEE Transactions on Image Processing*, 27(1):135–150, 2018.

[24] Guoxian Zhang, Silvia Ferrari, and Chenghui Cai. A comparison of information functions and search strategies for sensor planning in target classification. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 42:2–16, 10 2011.

[25] Guoxin Zhang, Silvia Ferrari, and M. Qian. An information roadmap method for robotic sensor path planning. *Journal of Intelligent and Robotic Systems*, 56:69–98, 2009.

[26] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[27] Feng Zhou, Fernando De la Torre, and Jessica K Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):582–596, 2012.