國立交通大學

多媒體工程研究所

碩士論文

角色動畫中腳步滑動之即時偵測與校正

Real-time Detection and Rectification of Footskate in Character Animation

研　究　生：徐銘聲
指導教授：林奕成　教授

中華民國九十八年二月

角色動畫中腳步滑動之即時偵測與校正
Real-time Detection and Rectification of Footskate in Character Animation

研究生：徐銘聲　　Student：Ming-Sheng Hsu

指導教授：林奕成　　Advisor：I-Chen Lin

國立交通大學
多媒體工程研究所
碩士論文

A Thesis
Submitted to Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十八年二月

角色動畫中腳步滑動 之 即時偵測與校正

研究生：徐銘聲　指導教授：林奕成　助理教授

國立交通大學
多媒體工程研究所

摘要

隨著電腦動畫的進步，在電影、電玩以及許多合成影像上加入電腦動畫已經是越來越頻繁的一種技術。當需要使用到人體的動作時，爲了取得真實的人體動作，經由動態捕捉技術來擷取動作在空間中的相對位置以及數據，已經是很普及的取得人體動作的方法。當設計動畫的人在使用這些人體動作時，由於取得的動作都是片段的方式呈現，爲了將動作及動作連接起來或者是產生新的動作，便會利用內插來產生所需要的動作。而直接內插的結果往往會產生異常的動作，像是在腳部的動作便會因爲內插的限制的關係而產生滑動或是不自然的移動，所以如何將這些情況解決便成爲一定要解決的問題。

要解決這個問題，首先便是要判斷在內插的結果之中是否有產生滑動或是不自然的移動，爲了要能夠自動化的判斷這些情況是否產生，可以把判斷分成兩個部分：首先便是判斷左右腳何時接觸地面，滑動以及所謂的不自然移動主要因素都是因爲人體腳掌在接觸地面之後，未離開地面卻產生雙腳移動的情況，因此可先藉由判斷雙腳的落地情況作更進一步的判斷。

我們利用支援向量機(Support Vector Machine) 來判斷腳部落地的時間可以更有效並且更快速的判斷連續的動作之間雙腳的落地情況，藉由不同的特徵來選出針對不同情況下的動作，可以讓整體的判斷更爲準確，經由正確的判斷腳部的落地之後，便可以對於腳部落地後是否產生移動作出更簡單的判斷。計算出合理的腳步落地的時間以及位置之後，便可以利用反向動力學(Inverse Kinematics)的方式將不自然的腳部運動調整到落地的位置，便可消除有腳部滑動的情況。

以往對於角色動畫中腳步滑動作處理的研究中，有研究對於消除滑步的產生，使用的方式是對於角色的骨骼構造提出一套調整的方式，利用幾個步驟的判斷及計算，並且對骨骼構造作調整後，便能消除滑步的產生；而有些研究則是利用機械學習的方式，提出一套機制並提供數據用來訓練之後，利用結果來判斷出角色在空間中的腳步情況，並且對腳步動作做類似的方式固定。

相較於之前的研究方式，我們所提出的方式，針對以上所提出的兩個部分做了改善。最爲主要的便是有關於判斷腳步接觸地面以及滑步產生的情況。加入了支持向量機的訓練方式，對於機械學習所需要提供的數據上，只需要使用較少並且減少繁複的計算的數據，便可以有更好的結果，同時也可以減少使用者互動測試的時間，而在角色動作的複雜度上，以前的研究成果中，僅對於較爲簡單的走路以及跑步作出正確的判斷，我們所提出的判斷方式，不僅對於原本的走路、跑步以及跳躍之外，也可以對武術動作作出判斷；而在校正動作的部分，我們簡化了判斷的部分，在判斷出滑步的時間後，便直接對腳步進行調整，利用逆向運動學的調整方式，將腳步滑動的情況降到最低，便可以跟以往有同樣的效果了。如此，在整體上的過程來看，便可以減少所需要的資料量，也可以減少計算的時間，便可以判斷出比以往好的結果。

利用這樣的方式，我們的系統可處理任何的人體動作在進行的過程，即使的判斷並且去除腳部滑動的問題，此技術可以快速且有效的讓動作看起來更爲自然順暢。

Real-time Detection and Rectification of Footskate in Character Animation

Student: Ming-Shang Hsu **Advisor: Dr. I-Chen Lin**
Institute of Multimedia Engineering
National Chiao Tung University

Abstracts

In this thesis, we present a real-time detection and adjustment method for human foot slides animation. To classify which frame foot slides happened, first of all we need to know when ground contacted happened. After ground contacted, we can determine that foot slides or not, then we can adjust the foot position and orientation.

In order to estimate the correct frame, we apply Support Vector Machine to each frame, and we use different features for different kinds of motions. Since SVM can estimate the result quickly and efficiently, we can identify when foot slide occur in real-time and we can adjust sliding foot to the right positions.

This approach can adjust the foot that slid to the right position easily and efficiently, and it could be used in different kind of human motion.

Keyword: Motion Interpolation, Footskate, Support Vector Machine (SVM), Center of Mass (CoM), Inverse Kinematic (IK).

致　　謝

在兩年半的碩士生活中，一路上得到了許多師長和同學的照顧以及幫助，再論文完成的同時，也要感謝他們：

1. 首先是要感謝父母的幫助，在這漫長的求學過程中，一路上都非常的支持以及照顧我，同時對於我的疑問也都盡力的協助，真的是非常的感謝他們。
2. 指導教授　林奕成博士，老師在這兩年半中非常辛苦的指導以及提供意見，同時對於我所遇到的瓶頸以及疑惑也都熱心的協助，讓我能夠順利的進行研究並且解決問題。
3. 實驗室的許多學長，任右學長、彥儒學長、國豪學長等三位學長對於我在個人研究上提供了相當大的幫助，同時對於研究時遇到的問題也都願意犧牲個人時間幫助我。
4. 藍英綬同學，在碩士的兩年多中，他熱心的教導並且改善我在程式撰寫上的一些錯誤習慣以及觀念，同時對於我的個人研究也提供了相當大的協助，是我在碩士研究中另一位非常重要的良師益友。
5. CAIG 的其餘同學，這些同學在日常生活中提供給我許多快樂的時光，同時在研究事情時也都一起提供了許多的觀念和新奇的想法，都是我在碩士生活中不可抹滅的回憶。

Contents

摘要..ii
Abstracts iv
致　　謝..v
Contents vi
1. **Introduction** 1
    1.2 **Introduction** 1
    1.3 **Overview** 2
    1.4 **Flowchart** 3
2. **Related Work** 4
    2.1 **Motion Transition** 4
    2.2 **Footskate Detection and Adjustment** 6
3. **Feature** 10
4. **Training** 14
5. **Footskate Rectification** 17
6. **Experiment and Result** 25
    6.1 **The Classification of Animation Data** 25
    6.2 **Experiment Comparison** 25
    6.3 **Animation Results** 27
7. **Conclusion & Future Work** 30
8. **References** 31

1. Introduction

In these years, there are more and more computer animation used in movies, cartoons and games. For a better quality, artists have to adjust data to make animation smooth and realistic. To automatically generate realistic characters animation, one way is using motion synthesis techniques. First of all, we acquire motion data from motion capture, which records subject motion through marker tracking. Artists could combine or reproduce new motions by motion blending or motion warping.

Unfortunately, data from motion capture are usually not perfect. Since motion capture data are recorded from the markers placed on the performers, the perturbation of the position of markers might degrade the motion capture data. For example, there are lots footplant in human motion, variable position of markers on the foot might cause an unexpected problem, called **footskate**, which means foot can't sticks on the ground and skates around. Footskate might also occur when handling motion interpolation and time alignment between motions, and this would also bring unnatural effect. Therefore, how to detect the foot slides and make the human motion looks naturally becomes an important task. So, we want to find out an approach to detect footskate quickly and correctly. After detection of footskate, we will do footskate adjustment in real-time.

1.2 Introduction

In this thesis, we focus on the footskate detection and footskate adjustment. There are lots algorithms for detecting footskate, such as using training machine or neural network algorithm to determine foot features and considering the ground contact event.

There are two main problems that cause footskate, variable position of foot markers and motion interpolation. Motion interpolation is used in motion editing, like motion blending and warping. One of the most common techniques is Motion

Graph. When handling motion interpolation, we can produce naturally upper body character animation but lower body would be discordant. Motion interpolation between footplants might casus footskate after character plant on the ground. Since footskate causes unnaturally human animation, we need to detect any rectify such situations.

Footskate handing has lots researches. Machine learning mechanism is one of efficiently way. System can collect user-specified training data. Using the data after cross comparison, system can estimate whether a foot contacts ground. There user can discover different feature or solution with different machine learning mechanism, like neural network or Support Vector Machine, SVM is a well-known solution for separate data with multi-dimension classification.

Another efficient solution is to detect according to the height of each foot position. Since the ground contact depends on the height of foot, we can avoid footskate by rectifying height of foot and use IK for residual joint pose adjustment

1.3 Overview

The goal of this paper is to design a training and editing system that can efficiently and accurately detect footplants in character animation. We propose using Support Vector Machine. In second chapter, we describe important papers about motion interpolation, footskate removal, physical footplant feature and machine learning mechanism. We would also introduce the correlation of these papers and our research. The third chapter is about the machine learning mechanism; we use Support Vector Machine (SVM) in our research to have a great power of discrimination on footplants. After the machine learning mechanism, we then present features provided to SVM for training and those to estimate result of footplants. Chapter 5 is about footskate rectification by using Inverse Kinematics, our system utilize separate handling methods for different situations, after footskate rectification, we would further smooth every footplant by Inverse Kinematics and Interpolation. Finally we would show our experiment result and data comparison between each feature and different constraints. At last, we would conclude this paper and mention our future work.

1.4 Flowchart

Figure 1 is our framework flowchart. Our system can be separate into 2 parts: Footskate Detection, and Adjustment. First of all we load a BVH file and record the data to transform to the feature for our system. Then our system uses the features into Support Vector Machine and predicts whether contacts occur. After ground contacting, we determine whether the foot slides happened and adjust the position of each joint by Inverse Kinematics to clean up the footskate. Finally we smoothed whole motion to get a naturally human motion.

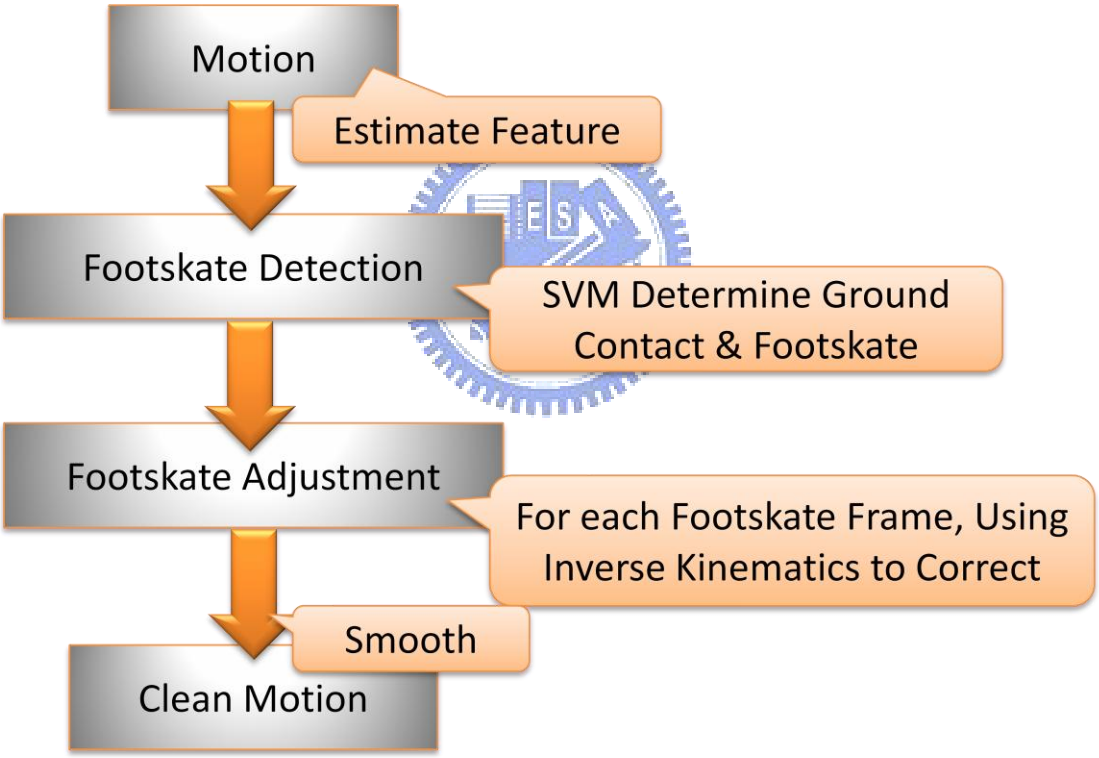


Figure (1): Our Flowchart.

2. Related Work

2.1 Motion Transition

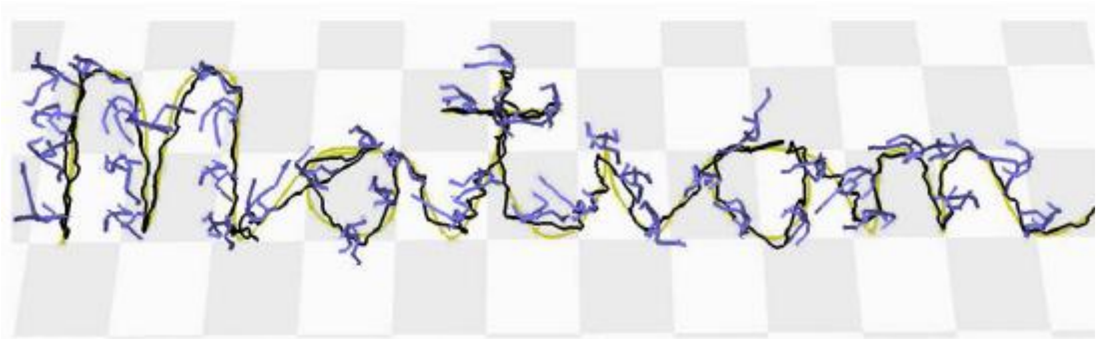


Figure (2): Motion Graph. Character motion applies to path fitting. This motion is shown the word "Motion". [Kovar et al. 2005]

The editing of motion capture data was used in many areas. Such kind of motion synthesis can be an effective way to construct impressive results. Users can construct novel character animations or increase variously of animation by combining different motion capture data. Besides these methods, motion interpolation is the most prevailing method to enrich motion capture data.

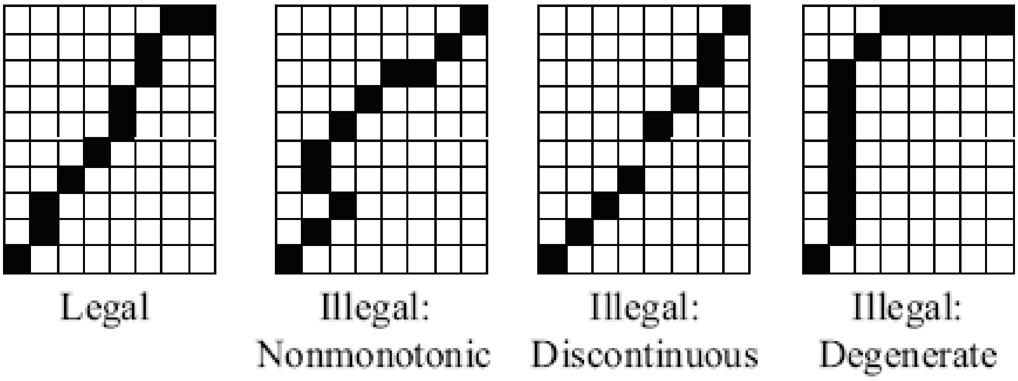


Figure (3): Time alignment restrictions. [Kovar et al. 2003]

L. Kovar & M. Gleicher [] brought up two criteria to determine numerical similarity for motion transition: **similar skeleton poses with corresponding frames** and, **the frame correspondences should be easy to identify**.

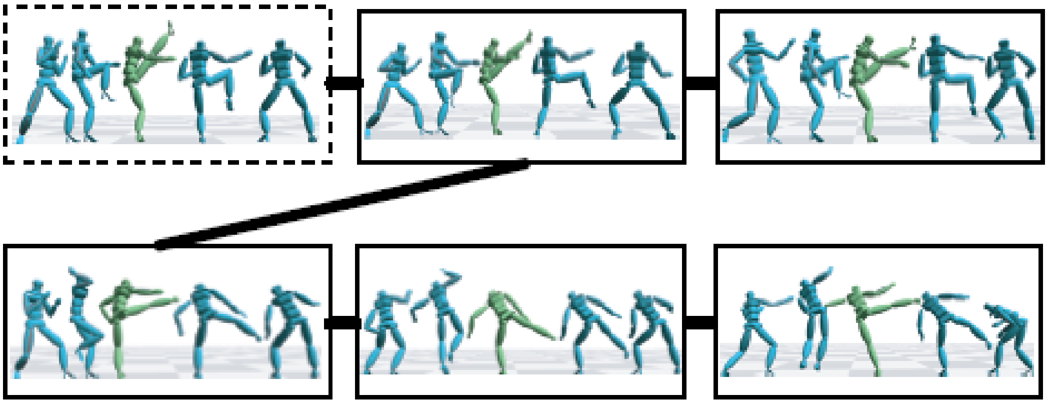


Figure (4): Motion Graph. [Kovar et al. 2003]

With the criteria, a motion graph system can find the similar motion segment with time alignment, and different character animation segments could be connecting smoothly.

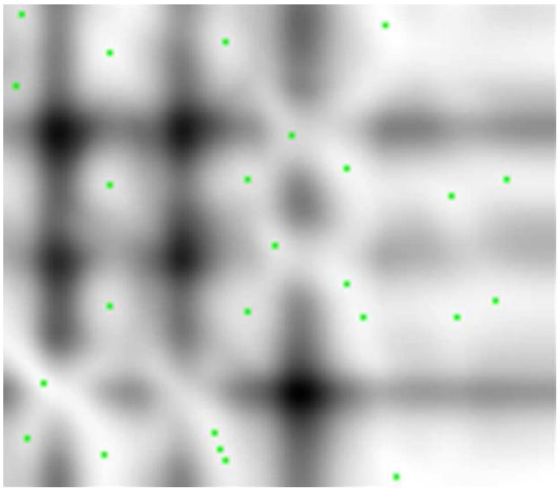


Figure (5): The window shows sum of distance of all joints between two random motions, black point is 1 and white is 0. The green points show local minimum. [Kovar et al. 2005]

L. Kovar & M. Gleicher [] created a two-dimensional map which shows two segments of motion capture data. By determining the sum of the differences between all joints with every two segments of motion, their system can build a two-dimensional map which shows the error difference.

After searching the optimal path of two different postures, their system can build a passage between the two postures and connect two postures by motion interpolation. With this approach, designers could create lots kinds of character animations. This system can also interpolate motions with different constrains like position of each joint, speed and footsteps constrains.

2.2 Footskate Detection and Adjustment

The quality of motion synthesis depends on how it looks natural or not. Due to inaccurately motion captured data or inappropriate interpolation in motion synthesis. Character’s foot might slide during ground contact, such unnatural effects called Footskate and it’s a serious problem in example-based motion synthesis.

There are many researches focusing on Footskate clean-up. These tasks were concerning on the position constrains or Footskate problem. Usually, system considers the position of Knee, Heel and Toes. By defining an appropriate criterion, we can identify whether ground contact or footskate happened. L. Kovar and J. Schreiner [] handling footskate problem by considering the position and orientation of the joints data of lower body. Their system determined the position where foot should put on before contact occurred, and adjust the orientation of each joint recursively to fit foot position constrains. After every joint fit its position constrains, smoothness filter are applied to ensure entire character animation is clean.

With this footskate cleanup algorithm, we can solve the problem just in few steps. This can handle Footskate problem quickly and easily. On the other hand, training mechanism is also a useful solution for Footskate problem. Using motion data from motion captured, system can classify motion data with different feature.

Although the algorithm can handle footskate problem in few step, but there are some unnatural result might happen when adjusting lower body position and orientation in every footplant contact. And the mechanism might take few times for the footplant contact and footskate.

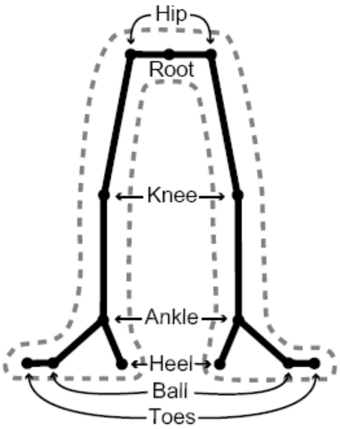


Figure (6): Lower body structure of character animation. [Kovar et al. 2002]

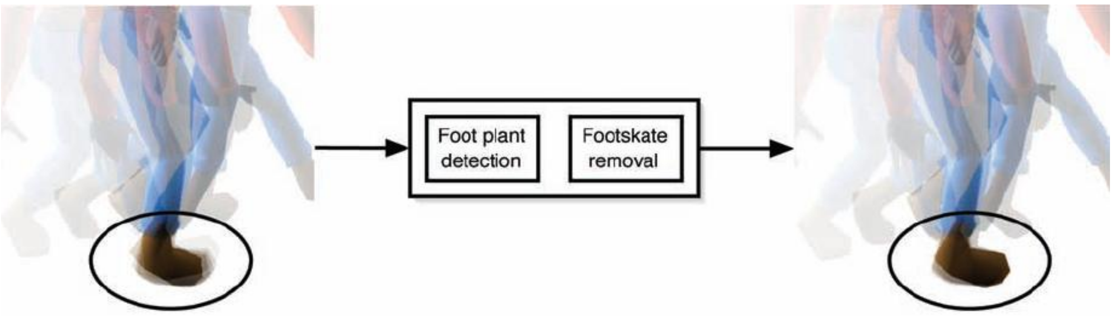


Figure (7): Footskate removal processing. [Ikemoto et al. 2006]

L. Ikemoto and O. Arikan [] presented a complete footplant detection and Footskate removal system. Their footplant detection system is interactively trained by joint position data and user labeled features. With the footplant data and feature, their training protocol used cross validation for footplant information, then the system applied k-nearest algorithm for all footplant data to get the final result.

Their Footskate removal system would use inverse kinematics to adjust motion data, changed heel position to fit the position. This footplant detection and removal system can execute in real-time and automatically. Although it’s effective in locomotion like walk and run, this footskate handling system can’t deal with complex animation data like dance or martial arts by u position of three joints.

Figure (8): joint position training system. [Ikemoto et al. 2006]

Instead, we propose using SVM to find a hyper plane in multi-dimension, support vector machine is a quick and efficient method to solve the problem. SVM algorithm can separates points into multiple groups. By using support vector machine algorithm, data need to be classified can be categorized with specific features. Since there are different kinds of character animation, our system can handle it with varied features for the best result.

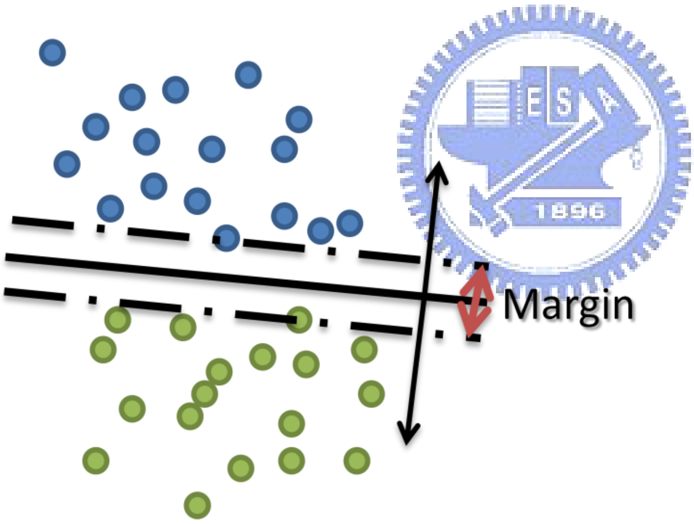


Figure (9): The SVM training machine classify objects according to its own features.

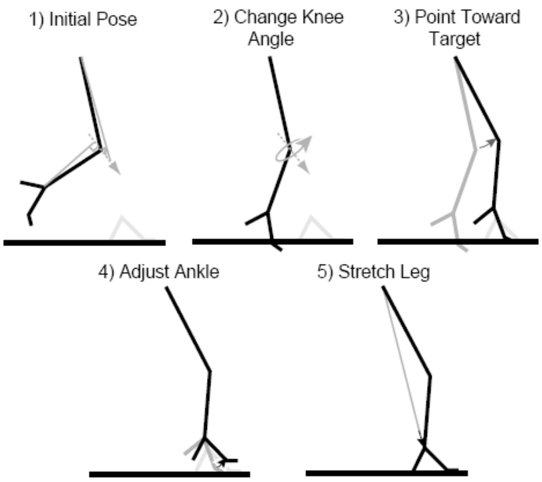


Figure (10): Footskate adjustment flowchart. [Kovar et al. 2002]

After figuring out the position that each foot should stop, feet should be adjusted to the right position by inverse kinematics. L. Kovar & J. Schreiner design an algorithm to adjust the foot position, they adjust character's root position, joint angle in the legs and modulate the lengths of skeleton structure of each leg to fit the constrains. However, there might bring discontinuities visual result with this algorithm.

Jen-Yu Peng et al. 2007 present a example-based motion synthesis technique. User can command character moved to the desire location and point to desire position with hand or lag. The algorithm can hit the correct position and modify character animation in real-time.

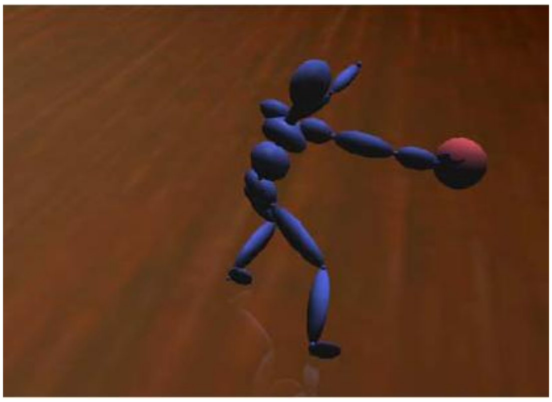


Figure (11): Character hit the position user assigned. [Alvin et al. 2007]

3. Feature

In this chapter, we would discuss the features used for our training system. The footskate detection problem could be separated into two parts: supporting foot contact to ground and footskate. Several features have been tried to test for the better training results. We would introduce these features and the advantage and drawbacks of these features, and the features used for our training system would be presented in the next chapter.

For more accurate estimation, we intend to find a suitable set of features for our system and the training system can predict motion label in real-time. Our features depend on the character skeletal configuration. We use both the joint position and orientation data to compare the result, and we also apply center of mass of each leg into our training data. We found that SVM can have better result with simple training data, and then we try to increase our data size from two to five times to compare the same feature. Since some error footskate results are unavoidable, we try to use SVM for footplant and footskate separately.

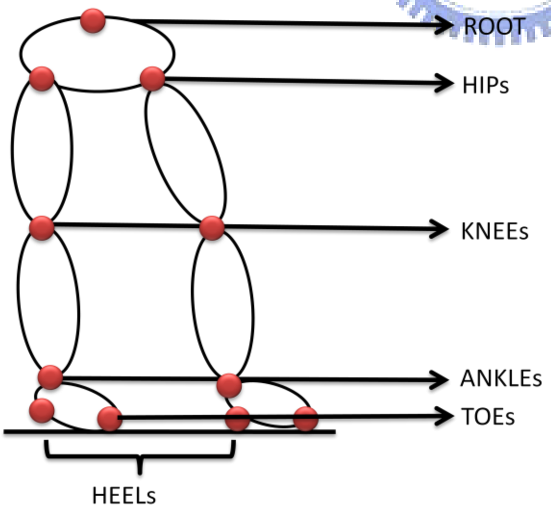


Figure (12): Character skeleton.

We introduce our features for classification in the following subsections:

A. Features in position domain

We use the joint position of both legs and root position, as our position features. For footplant contact detection, the most important joints are toe and heel, and knee or hip. Positions are also effective in our training. For joint positions in upper body, we found that they are unrelated to footplant detection, so we didn't test upper joint data in our training system. Our feature vector for position is

$$F_{\text{position}}\ (\text{Label}, \text{Frame}) = \begin{bmatrix} \text{Label} \\ \text{Pos}_{\text{Knee LR}}^{\text{Frame}} \\ \text{Pos}_{\text{Heel LR}}^{\text{Frame}} \\ \text{Pos}_{\text{Toe LR}}^{\text{Frame}} \\ \vdots \end{bmatrix}$$

Where Label is foot plants information.

B. Velocity & acceleration domain

After considering position domain, we use the position domain to estimate velocity and acceleration domain. Our test can separate to two situations: first of all we do calculate the velocity & acceleration with all dimension, thus we can get the global movement variation and use the data to train our motion data; otherwise, we can separate the y-dimension. We use X-Z dimension to estimate the plane variation for footskate and y-axis to estimate foot plants contact. Both situations can be accomplished with same algorithm in real-time.

Solution 1:

$$F_{\text{Velocity}}\ (\text{Label}, \text{Frame}) = \begin{bmatrix} \text{Label} \\ \text{Vel}_{\text{Knee LR}}^{\text{Frame}} \\ \text{Vel}_{\text{Heel LR}}^{\text{Frame}} \\ \text{Vel}_{\text{Toe LR}}^{\text{Frame}} \\ \vdots \end{bmatrix} \quad \text{and}$$

$$F_{\text{Accelreation}}\ (\text{Label}, \text{Frame}) = \begin{bmatrix} \text{Label} \\ \text{Acc}_{\text{Knee LR}}^{\text{Frame}} \\ \text{Acc}_{\text{Heel LR}}^{\text{Frame}} \\ \text{Acc}_{\text{Toe LR}}^{\text{Frame}} \\ \vdots \end{bmatrix}$$

Solution 2:

$$\begin{cases} \mathrm{F}^{xz}_{\mathrm{Velocity}}\ (\mathrm{Label}, \mathrm{Frame}) = \left[\mathrm{Label} \quad {\mathrm{Vel}^{\mathrm{Frame}}_{\mathrm{Knee}}}_{xz} \quad {\mathrm{Vel}^{\mathrm{Frame}}_{\mathrm{Heel}}}_{xz} \quad {\mathrm{Vel}^{\mathrm{Frame}}_{\mathrm{Toe}}}_{xz} \quad \ldots\right] \\ \mathrm{F}^{y}_{\mathrm{Velocity}}\ (\mathrm{Label}, \mathrm{Frame}) = \left[\mathrm{Label} \quad {\mathrm{Vel}^{\mathrm{Frame}}_{\mathrm{Knee}}}_{y} \quad {\mathrm{Vel}^{\mathrm{Frame}}_{\mathrm{Heel}}}_{y} \quad {\mathrm{Vel}^{\mathrm{Frame}}_{\mathrm{Toe}}}_{y} \quad \ldots\right] \end{cases}$$

$$\begin{cases} \mathrm{F}^{xz}_{\mathrm{Accelreation}}\ (\mathrm{Label}, \mathrm{Frame}) = \left[\mathrm{Label} \quad {\mathrm{Acc}^{\mathrm{Frame}}_{\mathrm{Knee}}}_{xz} \quad {\mathrm{Acc}^{\mathrm{Frame}}_{\mathrm{Heel}}}_{xz} \quad {\mathrm{Acc}^{\mathrm{Frame}}_{\mathrm{Toe}}}_{xz} \quad \ldots\right] \\ \mathrm{F}^{y}_{\mathrm{Accelreation}}\ (\mathrm{Label}, \mathrm{Frame}) = \left[\mathrm{Label} \quad {\mathrm{Acc}^{\mathrm{Frame}}_{\mathrm{Knee}}}_{y} \quad {\mathrm{Acc}^{\mathrm{Frame}}_{\mathrm{Heel}}}_{y} \quad {\mathrm{Acc}^{\mathrm{Frame}}_{\mathrm{Toe}}}_{y} \quad \ldots\right] \end{cases}$$

C. Orientation domain

We get the orientation of each joint to determine the angular velocity and acceleration. We transform the Euler angle of each joint into one angle presented by Quaternion orientation. When character’s foot has contacted to ground, one foot which swing in the air should have local minimum angular velocity and acceleration. Therefore, orientation domain could be a kinesiology research on footplant contact detection and thus we can use for footskate detection.

$$\mathrm{F}_{\mathrm{Orientation}}\ (\mathrm{Label}, \mathrm{Frame}) = \begin{bmatrix} \mathrm{Label} \\ {\mathrm{Orien}^{\mathrm{Frame}}_{\mathrm{Hip}}}_{LR} \\ {\mathrm{Orien}^{\mathrm{Frame}}_{\mathrm{Knee}}}_{LR} \\ {\mathrm{Orien}^{\mathrm{Frame}}_{\mathrm{Ankle}}}_{LR} \\ \vdots \end{bmatrix}$$

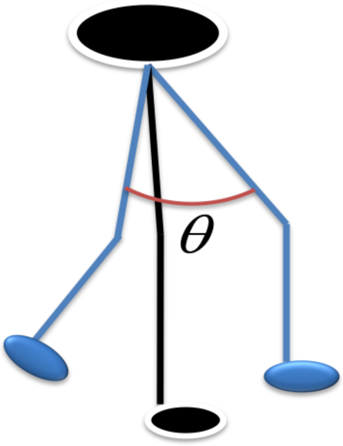


Figure (13): Blue line shows a swing leg and its orientation when character

D. Center of Mass domain

After the solution above, we refer the concept of momentum. We record joint position and combine skeleton position and weight, and then we can estimate the center of mass of each leg. Center of Mass presents a global movement of each leg, and we test our training data by COM position and apply toe and heel position

$$F(\mathrm{Label},\mathrm{Frame}) = \left[\mathrm{Label} \quad \mathrm{COM}_{\mathrm{foot}}^{\mathrm{Frame}} \quad \mathrm{Heel}_{\mathrm{foot}}^{\mathrm{Frame}} \quad \mathrm{Toe}_{\mathrm{foot}}^{\mathrm{Frame}} \quad \ldots\right]$$

Where the COM means center of mass.

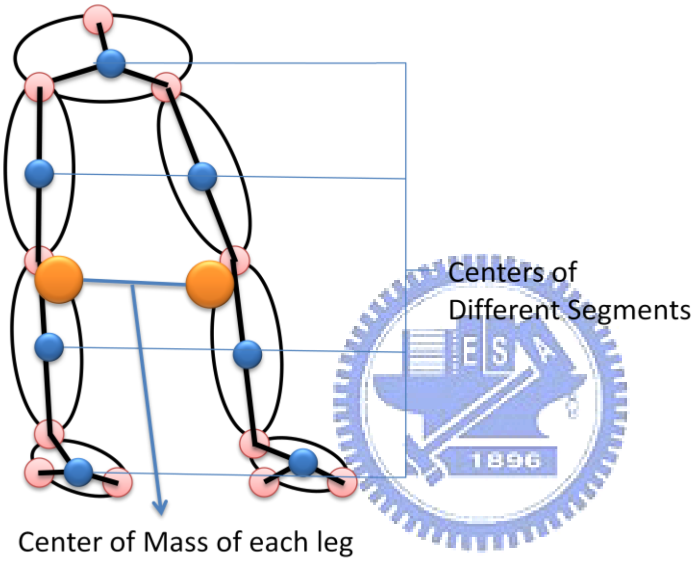


Figure (14): blue point is centre of each joint. We can use center position to calculate COM.

E. Other feature

We also test others features like straight line between any of two leg joints, and the correlation between the position and velocity of all joints. However, the training results are unsatisfactory. So we abandon those training features.

In the training processes, there are various joint configurations for joint registration. We compare both global and local position of each joint. And then we average the height of all root position to get an ordered joint position. With joint registration, the training process is more stable.

4. Training

We apply Support Vector Machine for data training. We train our system with 2200 motion clips with 5 labels. In this chapter, we would introduce SVM mechanism and how we handle our motion data with different features.

Our training system is trained from a group of human animation. We record human animation and apply the labels annotated by user, the classification of labels are **left foot plant**, **right foot plant**, **bi-foot plant**, **no foot plant**, and **footskate situation**.

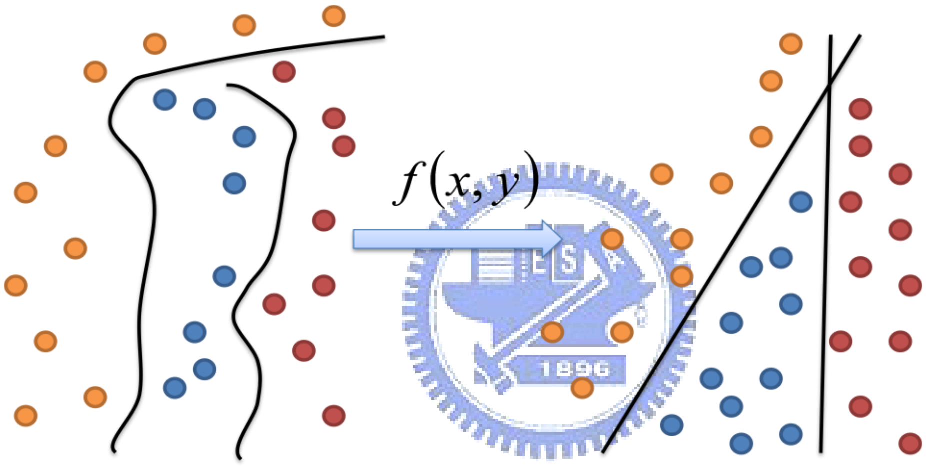


Figure (15): SVM can decrease multi-dimension data into lower level data.

We use SVM to train our motion features. First of all we input the feature with labels annotated by users, and then SVM would find the Optimal Separating Hyperplane (OSH) and Support Hyperplane. We can get the Support Hyperplane by the equation below, and we can use the equation to get the maximum Support Hyperplane.

$$w^{T}x - b \le -1 + \xi \quad \forall y_i = -1$$

$$w^{T}x - b \ge 1 - \xi \quad \forall y_i = 1 \quad \text{and} \ \ \xi \ge 0 \ \ \forall i$$

Where w is a vector vertical to OSH, b is displacement of the OSH, and ξ is used for measure the error of x.

After conclude the calculation, the SVM problem becomes:

$$\begin{cases} \min \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \\ y_i(w^T x_i - b) - 1 + \xi_i \geq 0 \quad \forall i \text{ and } \xi_i \geq 0 \ \forall i \end{cases}$$

Where C is the label of x.

Then we can use Lagrange Multiplier Method to transform the equation (3) to a quadratic equation, and we get equation (4). After we can apply the training data into the equation with Karush-Kuhn-Tucker condition, we can get the support vectors which are on the support hyperplane, and then we can classify our training data.

Before we train our animation data, we classify the animation data according to their variety for a better recognition. We separate the animation data into “Locomotion” and “Comprehensive Animation”. The “Locomotion” includes “WALK”, “RUN”, and “JUMP”, and the “Complex Animation” includes “PUNCH”, “KICK”, and other kind complex motion data. After classify our animation data, we can train each kind animation with different feature which can acquire better effect.

A. LOCOMOTION:

Since locomotion has less variation, so when we choose features, we can select easier ones to use for reducing the estimation time. Finally we choose the position of heel, toe and the orientation of ankle. And we though the orientation of ankle might affect ground contact directly. So the position of heel and toe with orientation of ankle might be a good feature for estimating ground contact.

B. COMPREHENSIVE ANIMATION:

For comprehensive animation, we use the position of heel and toe to estimate the distance between foot and ground. Because there is larger variation of complex animation, so we apply the Center of Mass of each leg into our training feature. We apply the weight into each lower body skeleton, and we evaluate the center of mass for each leg. Finally, these features can provide great training effect for our system.

We compare the results between different sizes of training data, and increase the frame size to get the neighbor frame data. The training system can predict label in the same speed with different size of training data. Finally, we apply the predict label to its frame data every frame, and then we would use the predict labels to implement our motion adjustment algorithm with different constrain.

5. Footskate Rectification

Once a footskate situation is detected, system them adjusts and rectify the detection by Inverse Kinematics. We use IK on the supporting leg. While directly applying IK, we found that it can also generate unnatural motion during switch of supporting foots. Therefore, we further divide the footskate into three situations:

1. Supporting leg changed from one leg to another.
2. Supporting leg changed from one leg to both leg vice versa.
3. Supporting leg fixed.

Our algorithm is applied to system as below:

Supporting leg		Solution
Before Footskate	After Footskate	
LEFT	RIGHT	**Switch Supporting Foot**
RIGHT	*LEFT*	
LEFT	*LEFT*	**Steady Supporting Foot**
RIGHT	RIGHT	
BOTH	BOTH	
BOTH	*LEFT*	**Switching Supporting Foot from or to two feet**
	RIGHT	
LEFT	BOTH	
RIGHT		

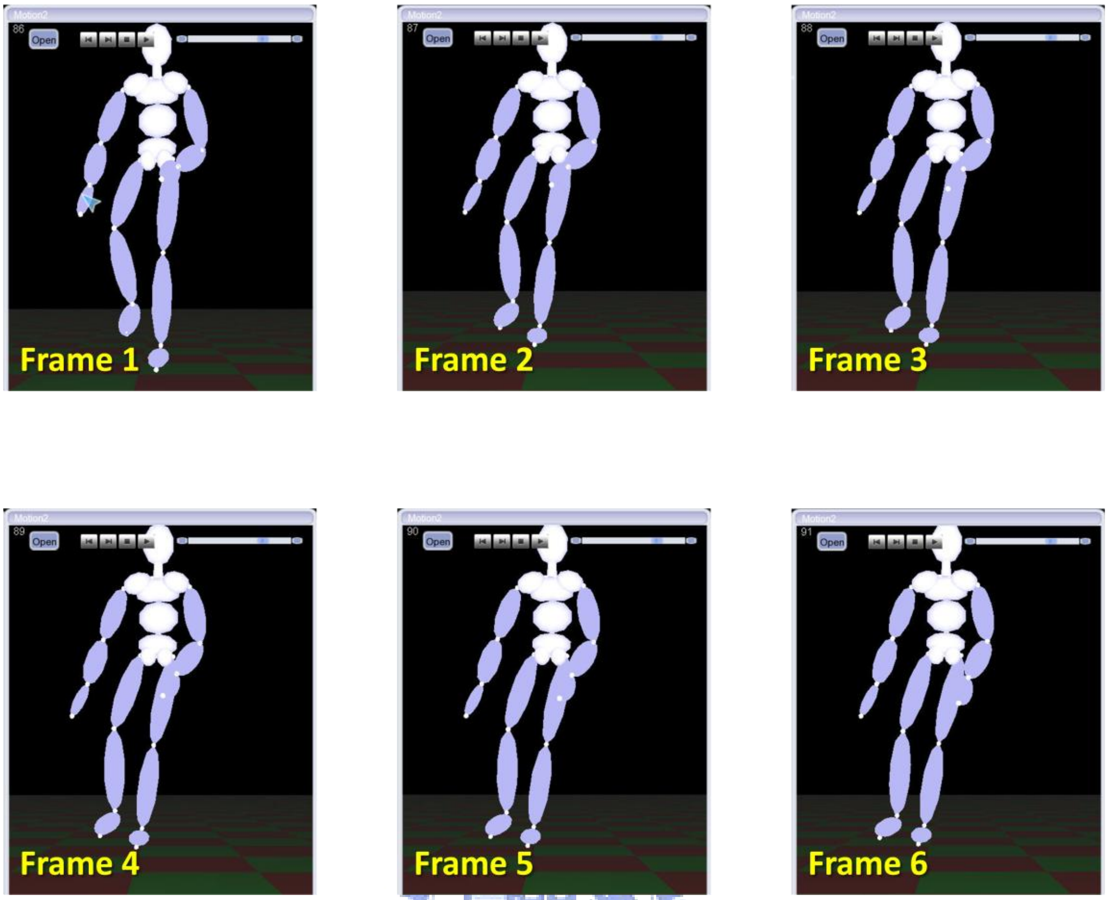


Figure (16): Supporting foot steady (Left foot is supporting foot and both feet slide).

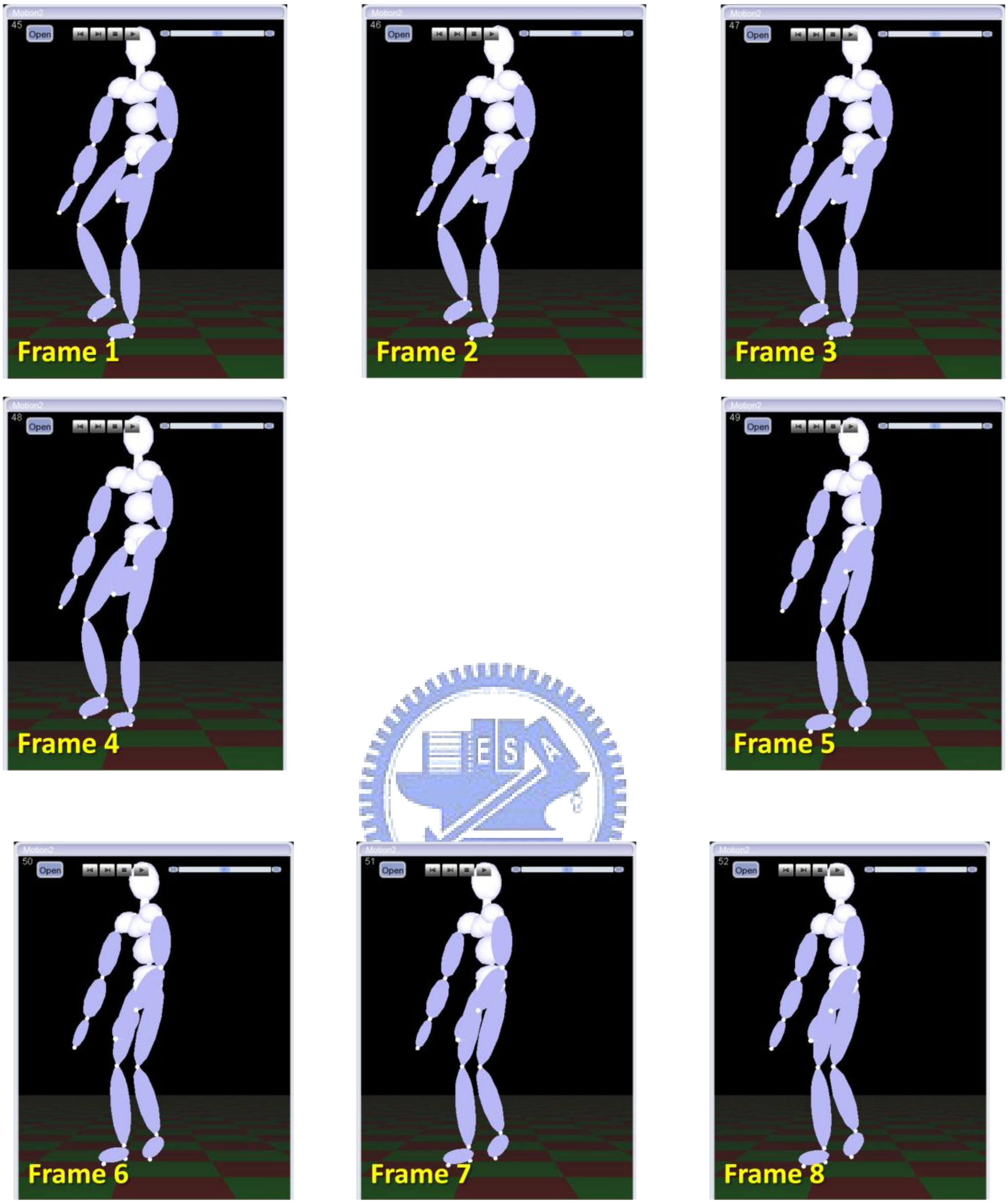


Figure (17): Supporting foot switch (supporting foot is changed from left to right).

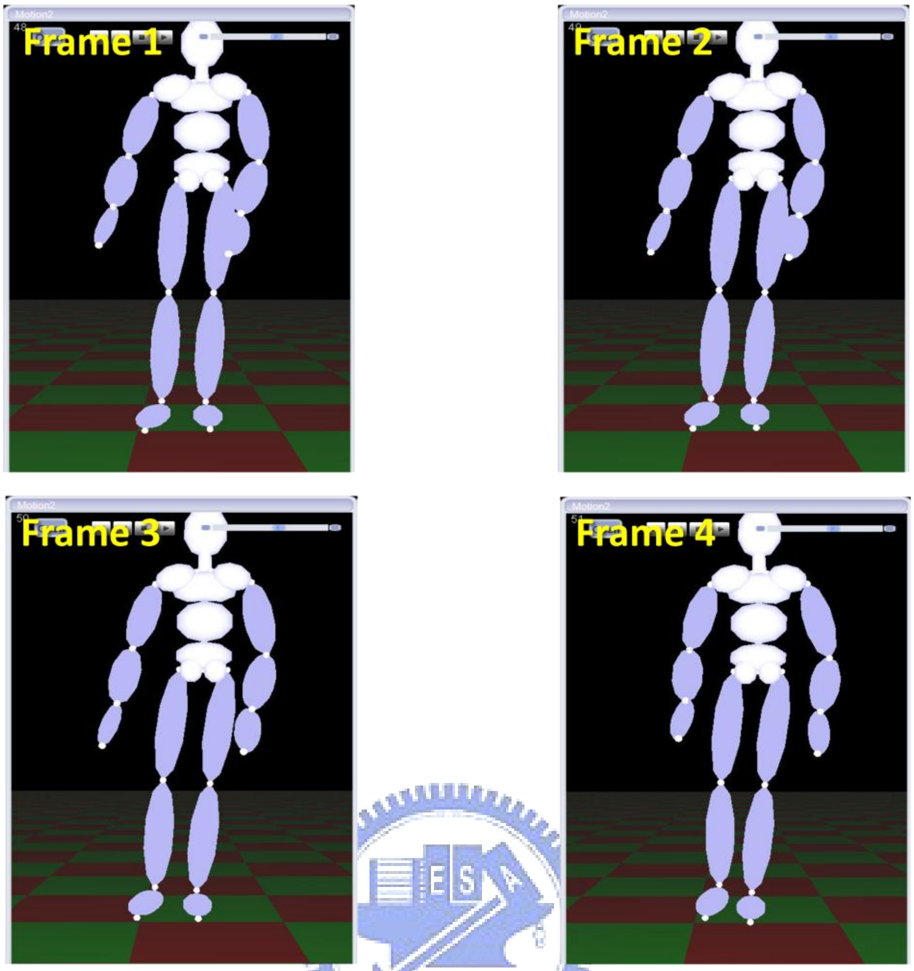


Figure (18): Switching Supporting Foot from or to two foot (two feet slide).

Algorithm shows below:

Switch Supporting Foot $\mathbf{FOOT_S^R \gg FOOT_T^L}$ Where R & L mean Right and Left foot and S & T mean frame num. The notation above means Supporting foot switch from Right foot in frame S to Left foot in frame T.	1. Determine which frame has the minimum distance of height of each heel from frame A to frame B. suppose the frame K has the minimum distance. 2. From frame S to K: Adjust the position of left foot to the position in frame T by Inverse Kinematic every frame. The trace of the left foot is determined by Bezier Curve. And we set the position of right foot in the same position to eliminate footskate. $$\mathrm{pos}_I^L\|_{I=S+1}^{K} = \left(1-\frac{I-S}{K-S}\right)\times\left(\left(1-\frac{I-S}{K-S}\right)\times \mathrm{pos}_S^L+\left(\frac{I-S}{K-S}\right)\times \mathrm{pos}_K^L\right)+\left(\frac{I-S}{K-S}\right)\times\left(1\times \mathrm{pos}_K^L\right)$$ 3. From frame K to T: Adjust the position of right foot to the position to the position in frame T, and set the position of left foot in the same position. $$\mathrm{pos}_I^R\|_{I=K+1}^{T} = \left(1-\frac{I-K}{T-K}\right)\times\left(1\times \mathrm{pos}_K^R\right)+\left(\frac{I-K}{T-K}\right)\times\left(\left(1-\frac{I-K}{T-K}\right)\times \mathrm{pos}_K^R+\left(\frac{I-K}{T-K}\right)\times \mathrm{pos}_T^R\right)$$
Steady Supporting Foot $\mathbf{FOOT_S^R \gg FOOT_T^R}$	1. Set the supporting foot in the same position to eliminate footskate. And adjust another foot from position in frame S to the position in frame T by IK every frame.
Switching Supporting Foot from or to two foot	1. Determine which frame has the minimum cost of height of each heel from frame A to frame B. suppose the frame K has the minimum cost.

$\mathbf{FOOT_S^R} \gg \mathbf{FOOT_T^B}$	2. From frame S to K: Adjust the position of left foot to the position in frame T by Inverse Kinematic every frame. The trace of the left foot is determined by Bezier Curve. And we set the position of right foot in the same position to eliminate footskate. $\mathrm{pos}_{\mathrm{I}}^{\mathrm{L}}\|_{\mathrm{I}=\mathrm{S}+1}^{\mathrm{K}} = \left(1-\frac{\mathrm{I}-\mathrm{S}}{\mathrm{K}-\mathrm{S}}\right) \times \left(\left(1-\frac{\mathrm{I}-\mathrm{S}}{\mathrm{K}-\mathrm{S}}\right) \times \mathrm{pos}_{\mathrm{S}}^{\mathrm{L}} + \left(\frac{\mathrm{I}-\mathrm{S}}{\mathrm{K}-\mathrm{S}}\right) \times \mathrm{pos}_{\mathrm{K}}^{\mathrm{L}}\right) + \left(\frac{\mathrm{I}-\mathrm{S}}{\mathrm{K}-\mathrm{S}}\right) \times \left(1 \times \mathrm{pos}_{\mathrm{K}}^{\mathrm{L}}\right)$ 3. From frame K to T: Set the both feet in the same position.

Figure (19): Table IK handler for footskate.

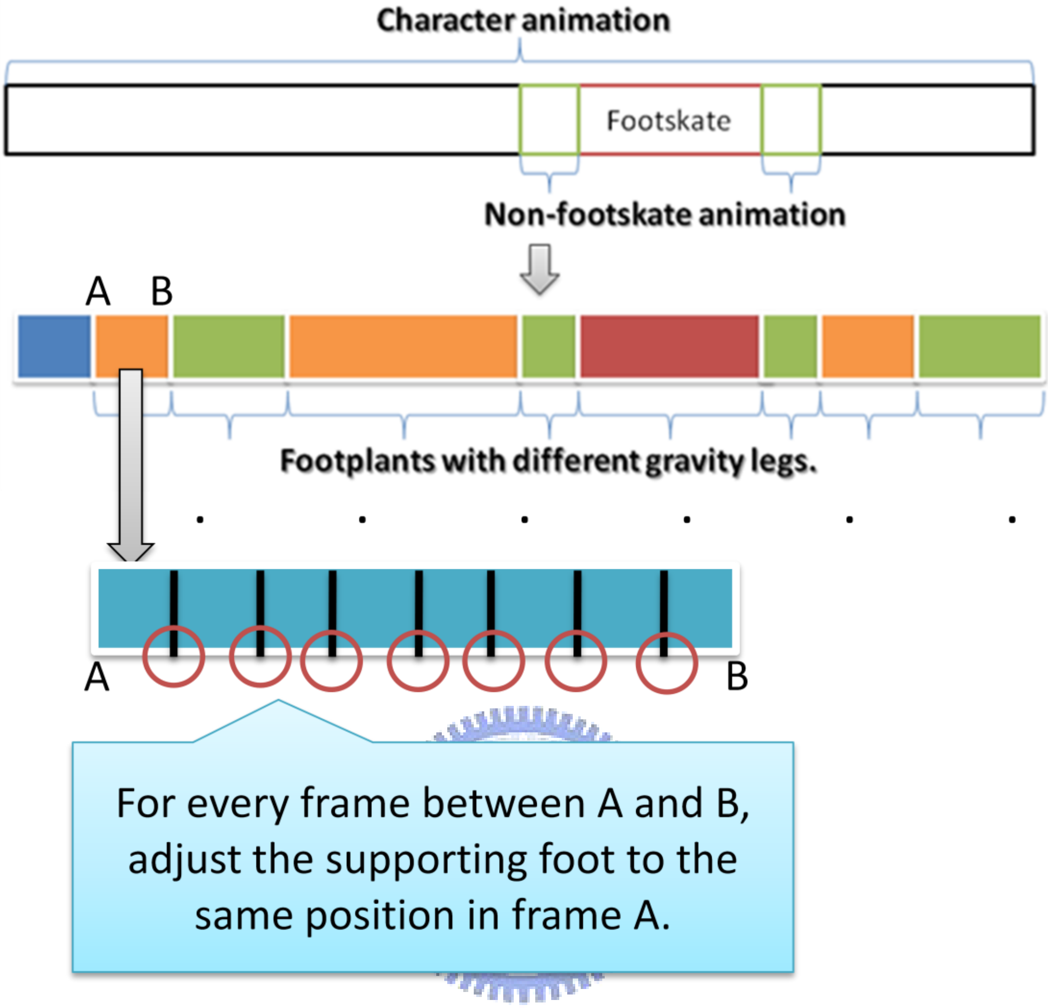


Figure (20): IK flowchart.

After supporting leg Inverse Kinematics rectification, we focus on every footplant in our on-line animation. During time segment that legs stay on the ground, we adjust every footplants after the first frame that contacts. Let foot holds in the same position and adjust other joints by Inverse kinematics. Finally, we eliminate footskate made of interpolation or data flaw and smooth the on-line character animation.

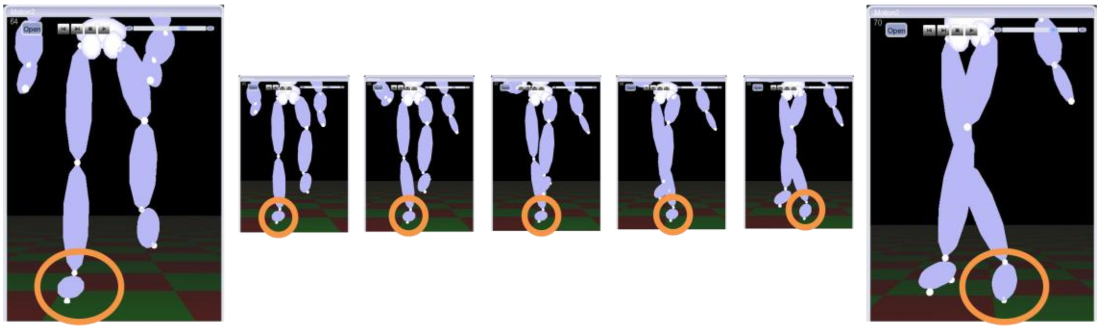


Figure (21): System would set the supporting foot in the same position after ground contact,

6. Experiment and Result

In this chapter, we will show our experiment data and the final result images. First, our motion data are classified according to its complexity. Then we will show the experiment result. The result data includes the compared labels between users' decision and the estimation by our training system. Character animations estimated and rectified by our method are also provided.

6.1 The Classification of Animation Data

Animation Num.	**127**	**Total Animation size**	**8486**
Animation Classification (file num / frame size)			
Locomotion		**Comprehensive Animation**	
Walk	**9 / 692**	**Punch**	**18 / 639**
Run	**9 / 549**	**Kick**	**18 / 920**
Jump	**18 / 923**		**-/-**
Motion Synthesis	**15 / 1542**	**Motion Synthesis**	**40 / 3221**

Table (1): Numbers and Classifications of Animation Data.

6.2 Experiment Comparison

In our experiment, we compare the user-specified labels and those estimated by our system. We separate the comparison with different features and it helps us to analyze the test result with different features. And we list three different result features which show the better predicted result.

$[\mathrm{Pos}_{\mathrm{Knee}}\quad \mathrm{Pos}_{\mathrm{Heel}}\quad \mathrm{Pos}_{\mathrm{Toe}}]$			
Locomotion		**Comprehensive Animation**	
Correct % / Total	**91.2% / 3706**	**Correct% / Total**	**90.1% / 3468**

$[\mathrm{Orien}_{\mathrm{Knee}} \quad \mathrm{Pos}_{\mathrm{Heel}} \quad \mathrm{Pos}_{\mathrm{Toe}}]$ (error / total)			
Locomotion		**Comprehensive Animation**	
Correct% / Total	**94.2% / 3706**	**Correct% / Total**	**92.9% / 3468**

$\lfloor\mathrm{COM}_{\mathrm{RL}} \quad \mathrm{Pos}_{\mathrm{Heel}} \quad \mathrm{Pos}_{\mathrm{Toe}}\rfloor$ (error / total)			
Locomotion		**Comprehensive Animation**	
Correct% / Total	**94.6% / 3706**	**Correct% / Total**	**92.8%/3468**

Table (2): The Correctness with Different Testing Features.

We compare our experiment data with different numbers of the size of neighbors. We test our animation with the position of the joint of lower body and apply Center of Mass of each leg, and the result shows that change of the neighbor frames doesn't affect the predicted result significantly.

Neighbors Frame number (Correct % / num. of frames)			
Locomotion		**Comprehensive Animation**	
1 frame	**94.1% / 1502**	**1 frame**	**95.1% / 1146**
3 frame	**93.3% / 1502**	**3 frame**	**94.4% / 1146**
5 frame	**94.5% / 1502**	**5 frame**	**95.1% / 1146**
10 frame	**93.6% / 1502**	**10 frame**	**92.3% / 1146**

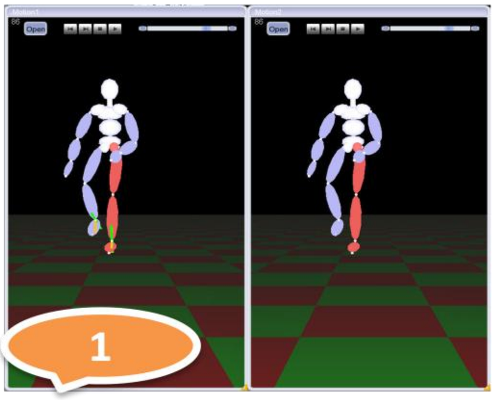
Table (3): The Correctness between Different Sizes of Neighboring frames.

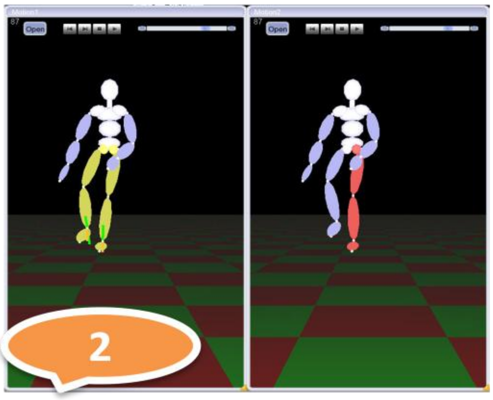
Racial between Footplant & Footskate	
Footplant frame num / correct%	**Footskate frame num / correct%**
5580 / 95.7%	**876 / 90.8%**

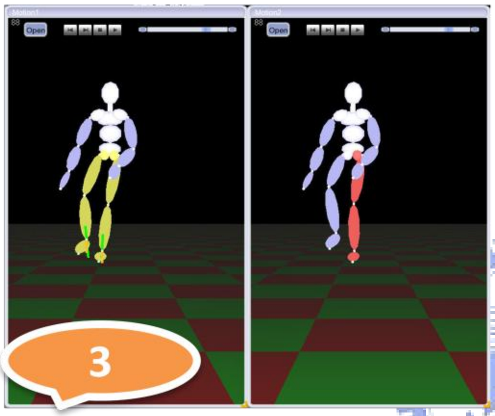
Table (4): .

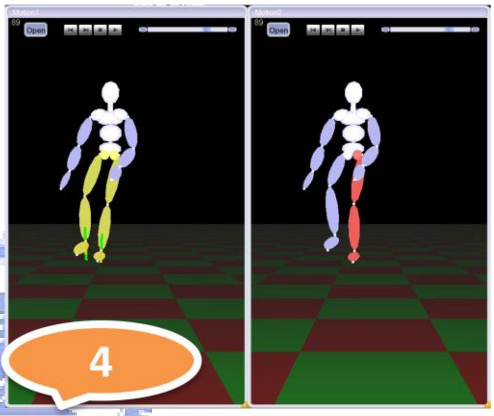
6.3 Animation Results

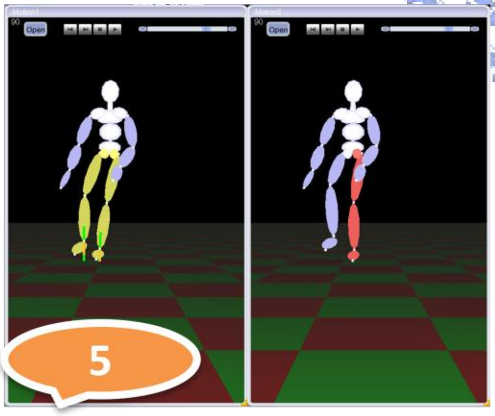
Our animation data show below:

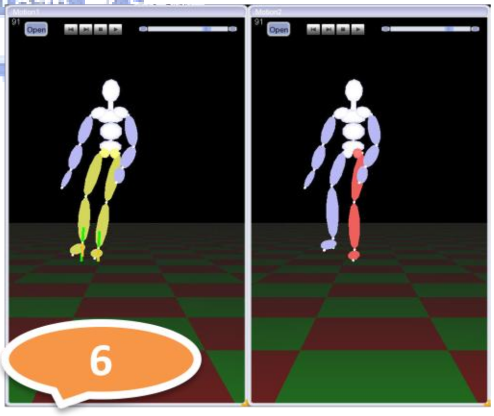


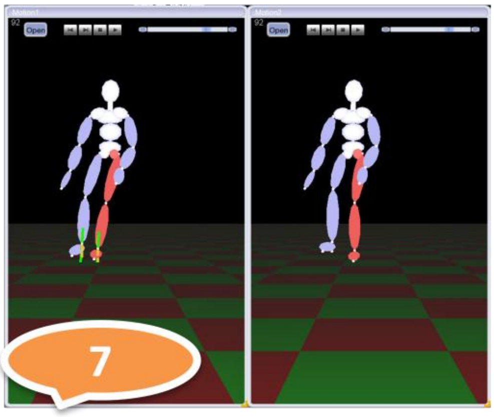


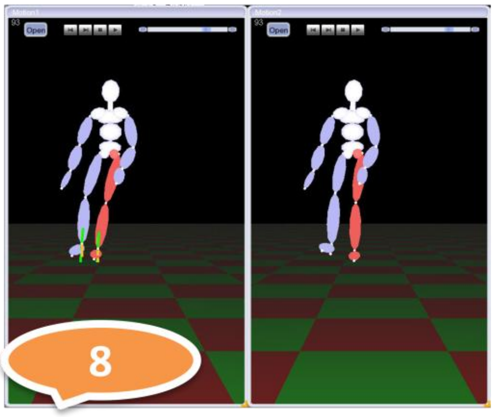


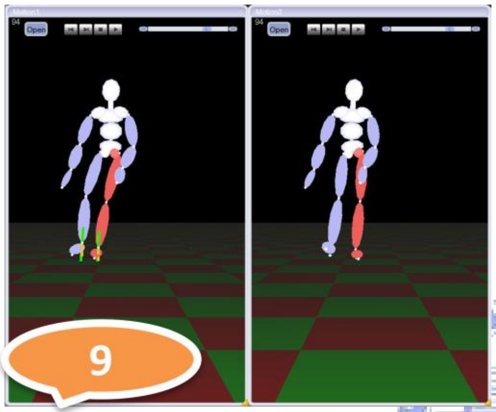


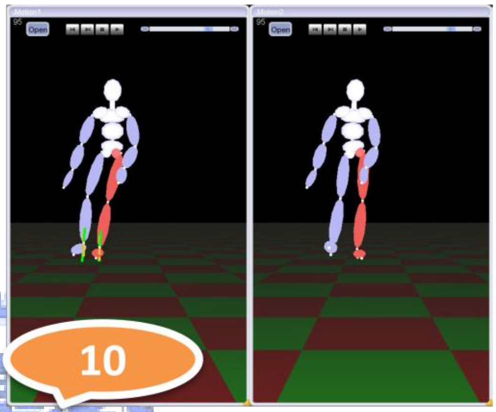


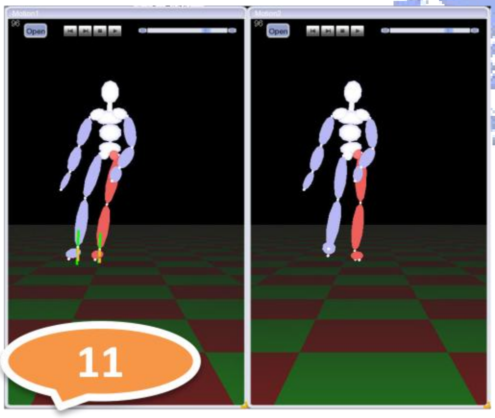


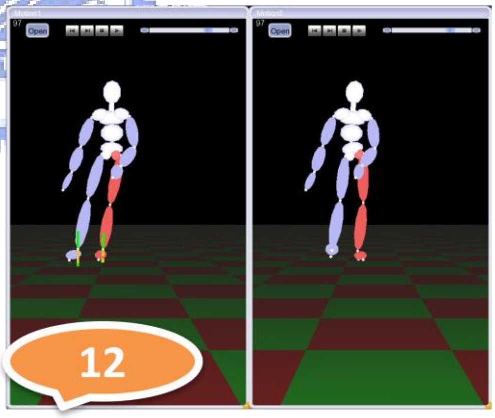

7

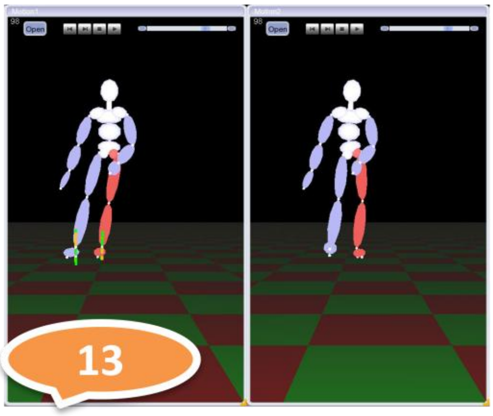

8

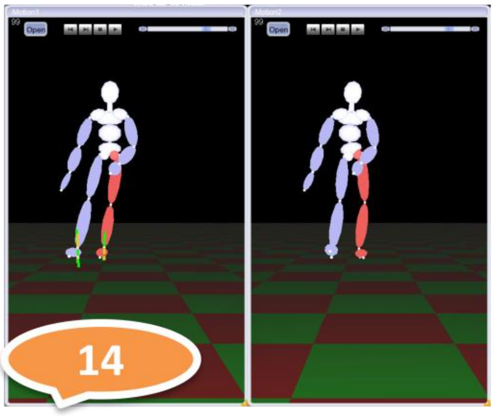

9


10


11


12


13


14

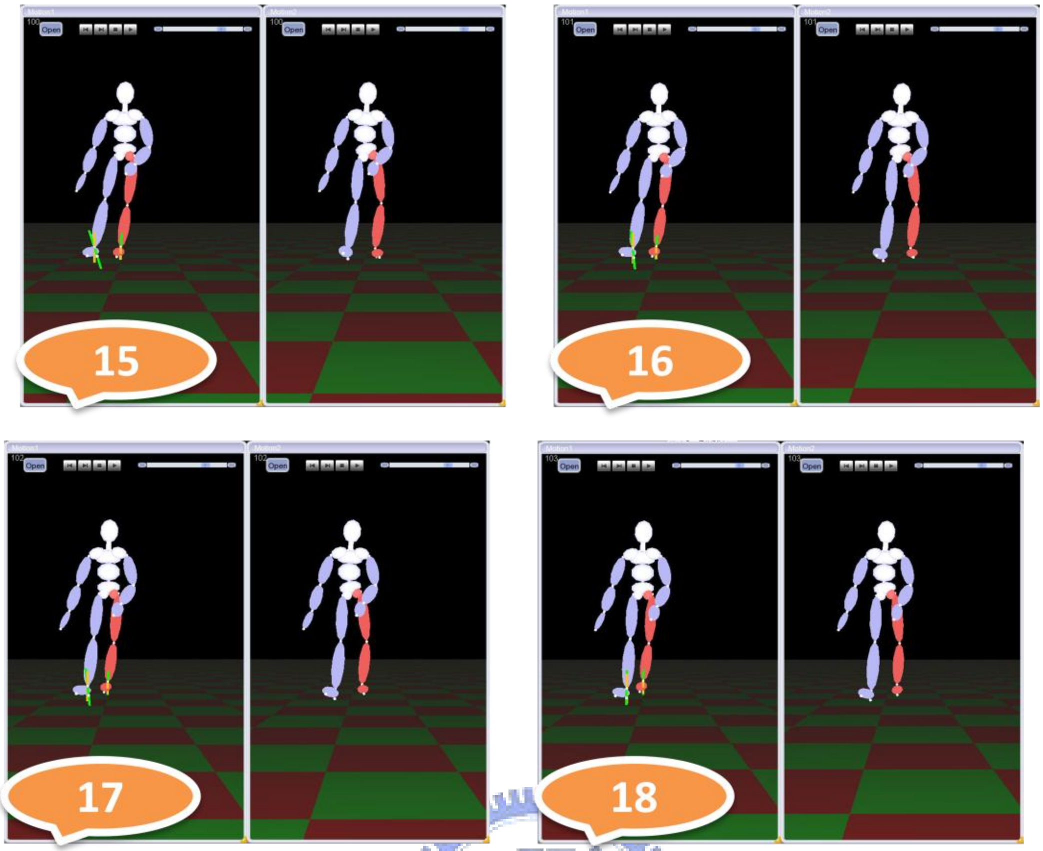


Figure (22): Animation Result. Left side is original animation and right is after detecting and rectifying.

7. Conclusion & Future Work

In this thesis, we present a system that can detects and adjusts footskate drawbacks of character animation in real-time. Our system can separate to two parts: Training and editing. The training system is trained by Support Vector Machine, and the training data is based on the information from legs and center of mass of lower body with different character physics states. We trained our system with various motion data and different features which selected depend on the complexity of human animation.

We use our motion editing algorithm to rectify character's leg position and orientation to fit the target position. To ensure the supporting foot sticks on the ground, we apply motion interpolation and Inverse kinematics to where foot plants happen. All the process can achieve in real-time, and our Inverse Kinematics algorithm runs ten iterations for a frame.

In our future work, several tasks could be included. First we plan to increase our character animation database for more complete training result, and we can try more features for better results.

8. References

[1] Anthony C. Fang, Nancy S. Pollard. 2003. Efficient synthesis of Physically Valid Human Motion. In Proceedings of SIGGRAPH '03.

[2] Arikan, O., Forsyth, D., and O'Brien, J. 2003. Motion synthesis from annotations. SIGGRAPH.

[3] Bindiganavale, R., and Badler, N. I. 1998. Motion abstraction and mapping with spatial constraints. Lecture Notes in Computer Science 1537.

[4] Bodenheimer, B, Rose, C., Rosenthal, S., and Pella, J. 1997. The process of motion capture: Dealing with the data. In Computer Animation and Simulation '97. Proceedings of the Eurographics Workshop.

[5] Jen-Yu Peng, I-Chen Lin, Jui-Hsiang Chao, Yan-Ju Chen and Gwo-Hao Juang. 2007. Interactive and flexible motion transition. In ...

[6] Gleicher, M. 1997. Motion editing with spacetime constraints. In proceedings of the 1997 Symposium on Interactive 3D Graphics.

[7] Gleicher, M. 1998. Retargeting motion to new characters. In Proceedings of SIGGRAPH 1998, 33-42.

[8] Ikemoto, L., Arikan, O., and Forsyth David. 2006. Knowing When to Put Your Foot Down. Proceedings of the 2006 symposium on Interactive 3D graphics and games.

[9] Indyk. P., and Motwani, R. 1998 Approximate nearest neighbors: towards removing the curse of dimensionality. In Proc. Of 30th STOC, 604-613.

[10] Kovar L., Gleicher M. 2003. Flexible Automatic Motion Blending with

Registration Curves. In *Proceedings Of ACM SIGGRAPH/Eurographics Symposium on Computer Animation,* 214-224.

[11] Kovar, L., Gleicher, M., and Pighin, F. 2005. Motion Graphs. In Proceedings of the 29th annual conference on Computer graphics and interactive techniques, ACM press 473-482.

[12] Kovar, L., Schreiner, J., and Gleicher, M. 2002. Footskate cleanup for motion capture editing. In Proceedings of the 2002 ACM symposium on Computer Animation(SCA).

[13] Lee, J., and Shin, S. Y. 1999. A hierarchical approach to interactive motion editing for human-like figures. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., 39-48.

[14] Liu, C.K., and Popovic, Z. 2002. Synthesis of complex dynamic character motion from simple animations. ACM Transactions on Graphics 21, 3(July), 408-416.

[15] Rose, C., Cohen, M., and Bodenheimer, B. 1998. Verbs and adverbs: Multidimensional motion interpolation. IEEE Computer Graphics and Applications 18, 5, 32-41.

[16] Witkin, A., and Popovic, Z. 1995. Motion warping. In Proceedings of SIGGRAPH 1995, 105-108.