Recognition of Walking Motion Using Support Vector Machine

Takefumi Matsunaga
Kyushu Institute of Technology
680-4 Kawazu,
Iizuka, 820-0067 Japan.
Phone/Fax: +81-948-29-7738
matsunaga@cg.ces.kyutech.ac.jp

Masaki Oshita
Kyushu Institute of Technology
680-4 Kawazu,
Iizuka, 820-0067 Japan.
Phone/Fax: +81-948-29-7718
oshita@ces.kyutech.ac.jp

Abstract

This paper presents a motion recognition method which combines Support Vector Machine (SVM) and state machine. We applied our method to the recognition of walking motions. We divide walking motion into five states (right leg up, right leg down, left leg up, left leg down, not walking). Based on subject's posture that is acquired using a motion capture device, our method recognizes the subject's current walking state as well as the subject's walking speed. We use the velocity of primary body parts (hands, feet, and pelvis) as a feature vector. Based on a trained model, a SVM detects the subject's current state. However, it is difficult to consider the subject's previous states with SVM. Therefore, we introduced a state machine in which the subject's current state is determined based on the previous state and the recognized state by the SVM. The walking speed is also computed from the state transition speed on the state machine. This paper also describes the results from some experiments that were made to evaluate the accuracy of our system.

1. Introduction

There are many applications of motion recognition technique such as gaming interface, monitoring systems, and controlling an intelligent robot. To achieve recognition results that are close to human's recognition accuracy recognition is required. For example, in case of commanding a robot by user's gestures, it is necessary to distinguish an intended gesture motion from a lot of motions. It is also important to detect not only user's motion type but also user's motion speed for some applications. User's motion speed can be used to control the speed of resulting actions. For example, a character's motion speed can be controlled by user's motion speed in the case of using motion recognition for a gaming interface.

In this paper, we propose a motion recognition technique that detects a user's motion type and motion speed with high accuracy.

We have applied our motion recognition technique to detect a user's walk-in-position motion for a motion capture-based avatar control system which we have been developing [1] (Figure 1). On this system, an avatar is directly controlled based on the input motion from a motion capture system. However, since the space for motion capture is usually smaller than the virtual world. In order to solve the problem, when the user walks in position, our system makes the avatar walk instead of making the avatar walk in position. The avatar's walking speed is also controlled based on the user's motion speed. The motion recognition technique is used to detect the user's walk-in-position motion and the motion speed.

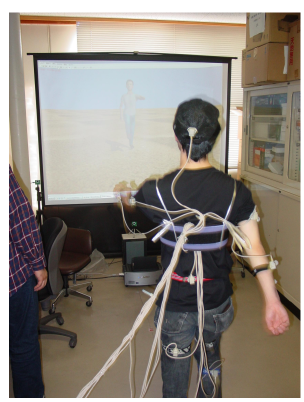

Figure 1. A motion capture-based avatar control system

Our method combines Support Vector Machine (SVM) and a state machine. Motion is recognized by using SVM, and the recognition result is changed according to the transition condition. SVM is a supervised learning method has been employed for various applications including motion recognition. We use SVM to detect each state of walking motion. The velocities of the user's primary body parts (hands, feet, and pelvis) are used as a feature vector. SVM detects only the current state of motion based on a feature vector. It is difficult to consider the user's previous states. We use a state machine with the SVM to solve these problems. We divide walking motion into five states (right leg up, right leg down, left leg up, left leg down, not walking) as shown in Figure 2. Based on a user's current state that is estimated by a SVM and the previous state, the state machine determines the user's current state. We introduce transition constraints to prevent undesired state transitions. The walking speed is also computed from the transition speed between states.

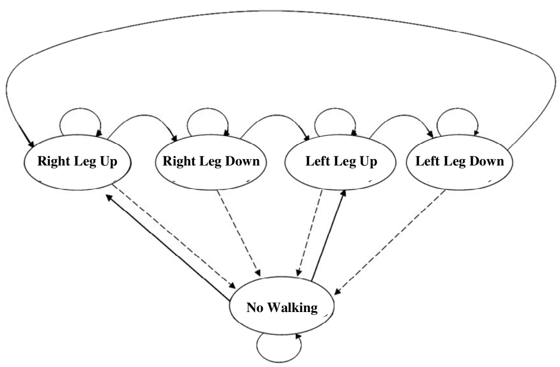

Figure 2. State machine

The rest of this paper is organized as follows. In Section 2 we review related works. In section 3 we explain the system overview and methods. The experimental system and results are explained in section 4. We discuss the future works in section 5. In section 6 we collect our research.

2. Related Works

Various methods for motion recognition have been studied.

Fuzzy rule based method is a common approach. We have applied this approach to the recognition of walking motion [1]. It is necessary to tune appropriate model parameters to make the Fuzzy rule based method work well. On the other hand, SVM has good generalization performance which means that it works well even on unknown data, it is not necessary to adjust the parameter manually. Our experiments showed that the SVM achieve a better recognition result compared to the Fuzzy rule based method.

Mori and Asada applied SVM for motion recognition [2]. They used the position and velocity of the primary body parts as a feature vector. They used SVM to detect one of three states (yes, no, neutral) for each motion such as walking and sitting down. They also used frequency features that are computed from a series of feature vectors in order consider the history of user input for motion recognition. On the other hand, we use a state machine for considering the history of user input. Using a state machine has an advantage that it can be used to compute user's motion speed.

Takahashi et al. used Hidden Markov Model (HMM) for recognition of person from their walking motions [3]. HMM is a probabilistic model which is widely applied to recognition of various temporal patterns including voice recognition. They used a pendulum model to represent the feet of the human body during walking motion. The length and angle of the pendulum model of feet are used as a feature vector. Compared to SVM, HMM has the possibility of falling into the local solution according to the setting of an initial value. The advantage of SVM can avoid the problem of trap in local solution. The recognition rate of SVM is higher than that of HMM because SVM doesn't fall into the local solution.

Ikehara et al. proposed a system for hand motion recognition that uses a Template Matching method [4]. The template matching is a simple classification method. The feature vector of an input is compared with feature vectors of many sample data (templates) and the closest sample is chosen. They used the positions and shape patterns of fingers as a feature vector. In their method, in order to make this method work well, the recognition environment must be limited or many templates must be prepared, because the shape of fingers varies by the direction of hands and the camera distance.

No matter what recognition method we use, the representation of feature vector is very important. We use the velocities of the primary body parts as a feature vector. We could improve the recognition result of our system by adding the position and the acceleration to a feature vector. However, adding irrelevant parameters to the feature vector could also make the recognition results worse. Therefore, we have to choose the appropriate parameters carefully.

3. Motion Recognition Method

3.1. System Overview

We have developed a walking motion recognition system as shown in Figure 3. The system consists of two processes: learning and recognition processes.

In the learning processing, a learning model (SVM) is generated based on a number of sets of a feature vector and a walking state that is given by hand. A feature vector is calculated from the posture on each frame of a sample motion as explained in Section 3.2. A corresponding walking state for each frame is given by a user manually. We defied walking motion into five states: "right leg up", "right leg down", "left leg up", "left leg down", and "not walking". This labeling process is explained in Section 6.1. Finally, a learning model is computed using the sets of a labeled state and a feature vector (Section 3.3).

In recognition processing, the user's current state is determined based on an input posture that is acquired from a motion capture system in real-time. First, a feature vector is calculated from the user's current posture in the same way that is used in the learning processing on each frame. Next, the user's walking state is recognized using the SVM learning model of SVM (Section 3.3). The output state is then determined using a state machine according to transition conditions based on the recognized state by SVM and the previous state (Section 3.4). Finally, the walking speed is also computed from the transition speed on the state machine (Section 3.5).

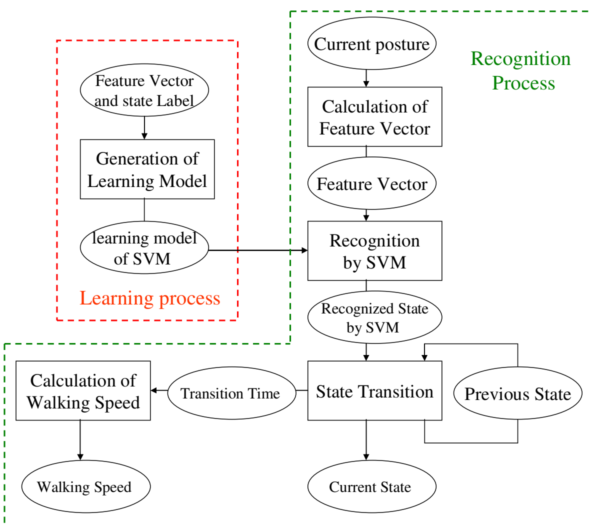

Figure 3. System flow

3.2. Feature vector

We use the velocities of primary body parts (hands, feet, and pelvis) as a feature vector. We use 6 velocities of each primary body parts in another direction: the lateral velocities of the hands, vertical velocities of the feet, and horizontal velocity of the pelvis. These velocities are computed in the pelvis's local coordinates as shown in Figure 4.

A feature vector is calculated from the user's posture at the current frame and the previous frame. The user's posture that is acquired from a motion capture device is represented by angles of all joints and the position and direction of the pelvis. The positions of the primary body parts are computed using a forward kinematics. The velocities of the body parts are then computed by computing the difference between the positions at the current frame and the positions at the previous frame.

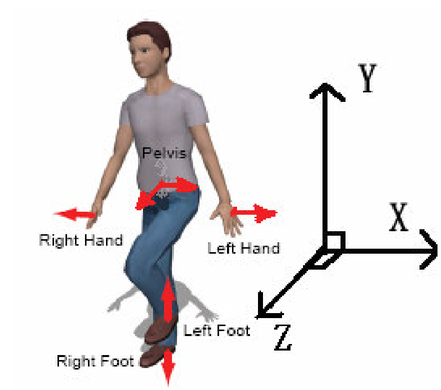

Figure 4. Feature vector

3.3. Support Vector Machine

SVM is a classification technique used in a wide field such as motion recognition [2], voice recognition [5] and person recognition [6]. SVM is a linear classification method that computes a hyper plane that divides the feature space into two classes [7]. The hyper plane is computed so that the distance between the data and the hyper plane (margin) is maximized. The SVM guarantees the convergence to the optimal solution. SVM has an advantage that it can achieve good recognition results even with a high dimension of the feature space.

In our implementation, we used a library for SVM, LIBSVM [8] that was developed by Lin. LIBSVM is an extension of SVM to a nonlinear multi-class classification.

Since LIBSVM supports multiple classes, we assign each walking state to a class and train a 5-class learned model from sample motion data as explained in Section 3.1. Given a feature vector, the learned model determines a walking state from the five walking states.

3.4. State Machine

In order to consider user's previous states, we introduce a state machine as shown in Figure 2. The state machine records the user's previous state. Based on the previous state and the recognized state by SVM, the state transition is determined according to the transition conditions shown in Table 1. The state transits based on the recognition result by SVM. When the user's state changes from leg up to leg down, the speeds of the hands and the feet become close to zero and the SVM recognizes this state as not walking state. In order to prevent the state transiting to not walking state in this case, we introduce a time condition on the transition from leg up state to not walking state and leg down state to not walking state. The state does not transit to not walking state until the SVM keeps recognizing the state as not walking during a fixed time. This ensures the smooth transition between leg up state to leg down state. We use 0.3 seconds for the transition condition.

Table 1. Transition condition table

Last state	Recognized state by SVM	Current state
Not Walking	Right Leg Up	Right Leg Up
	Left Leg Up	Left Leg Up
	Other States	Not Walking
Right (Left) Leg Up	Right (Left) Leg Down	Right (Left) Leg Down
	Not Walking (Passage of fixed time)	Not Walking
	Other States	Right (Left) Leg Up
Right (Left) Leg Down	Right (Left) Leg Up	Right (Left) Leg Up
	Not Walking (Passage of fixed time)	Not Walking
	Other States	Right (Left) Leg Down

3.5. Computation of walking speed

Our system computes the walking speed as the number of steps/sec. If necessary, we can estimate the walking speed m/s using an approximation of step distance [9].

The walking speed (steps/sec) is computed from the transition times. However, if we compute the walking speed from the transition times of one cycle, it rapidly changes. To avoid this, we use an average speed that is computed form a few previous steps. In our experiment, we used five or ten previous steps.

4. Experiments

4.1. Labeling

We developed a labeling interface for preparing sample data for training the SVM. Using our system, a user can specify the one of five walking states at any frame of a sample motion by pushing a corresponding key while watching the sample motion being played back. The labeling process depends on the user's subjectivity. The labeled sate is stored with the feature vector at the same time for training SVM. The labeling data is also used for evaluating the recognition results on our experiments. The learning model is generated from a number of sets of labeled state and feature vector.

4.2. Experiments

We have done some experiments with some sample motion data. We used two walk-in-position motions and some non-walking motions for comparison. The motion data were captured using a magnetic motion capture device MotionStar from ascension tech [10]. Two walk-in-position motions are captured from the same person. They are labeled in an off-line process using our system as explained in Section 6.1 and used to train two different SVMs. Our recognition method is evaluated with the trained SVM and the labeled data. We applied our method to the captured motions and evaluated if the recognition result matched to the manually labeled state on each frame.

4.3. Recognition of Walking Motion

This section shows the experimental result when our method to applied to walk-in-position motions. We applied a SVM trained frame a motion to the same motion (Section 6.3.1) and to the other motion (Section 6.3.2). We also tested the recognition rate with or without the state machine. Without the state machine, the outputs from the SVM are compared with the labeled state. The recognition rates on the experiments are summarized in Table 2.

4.3.1. Recognition of the learned walking motion. On experiment 1, we applied each SVM to the same motion that is used to train the SVM. The recognition rate was approximately 85 % on motion A and 87 % on motion B. The recognition error was caused because of the delay of state change when motion state changes from lowering the foot to not walking (standing state).

4.3.2. Recognition of un-learned walking motion. On experiment 1, we also applied each SVM to the different motion with the motion that is used to train the SVM. The recognition rate was approximately 85% on application of SVM learned from motion A to motion B and 86% on application of SVM learned from motion B to motion A. This result indicates the walking motion not learned can be recognized in high accuracy.

Since we captured walk-in-position motions from only one person, the recognition rate of another user's motion is not evaluated. This is a future work.

4.3.3. Evaluation of state machine. On experiment 2, in order to evaluate the effectiveness of the state machine, the recognition rates of the outputs from the SVMs are also measured. The results showed that the recognition rates decreased without the sate machine. This is mainly caused because the recognition state changes to not walking state when the state changed from raising a foot state to lowering the foot state as explained in Section 3.4. Using the state machine, this problem was fixed and more accurate recognition rates were achieved.

4.3.4. Comparison with Fuzzy method. The recognition rates of the Fuzz rule based method [1] and our method are compared. On experiment 3, the recognition rates of the outputs from Fuzzy are measured. On experiment 4, the recognition rates of the outputs from the combines Fuzzy and a state machine are also measured. On experiment 3 and 4, the recognition rate was worse than SVM. With the Fuzzy rule based method, appropriate parameters for Fuzzy rules must be turned manually. On our experiment, it was difficult to tune the parameters so that it achieves good results. Even after manual tuning, the Fuzzy rule based method shows lower recognition rates compared to SVM.

Table2. Recognition rates on the experiments

Experiment no	Recognition method (sample motion)	Recognition result	
		Motion A	Motion B
1	SVM + State Machine (motion A)	85%	84.8%
	SVM + State Machine (motion B)	85.8%	87.1%
2	SVM (motion A)	82.1%	80.7%
	SVM (motion B)	83.4%	83.7%
3	Fuzzy	74.2%	76.6%
4	Fuzzy + State Machine	76.8%	76.6%

4.4. Distinction with other motions

We have applied a trained SVM to non-walking motions and evaluated if our method detects them as not walking state. We used the SVM trained with motion A. Some motions from a commercially available motion data package "rikiya" [11] was used for the experiments. The recognition rates are summarized in Table 3. The results indicate that our method works well especially with the state machine.

Table 3. Recognition rates of other motions

motion \ transition condition	With Conditions	No Conditions
Small jump	100%	94%
Sitting down	100%	100%
Striding	100%	90%
Big jump	100%	91%

4.5. Walking Speed

We also tested the computation of the walking speed with motion B as shown in Figure 5. Horizontal axis is the number of steps, and vertical axis is walking speed (steps per second). On this experiment, the subject was asked to keep a constant walking speed. The three curves shows the walking speeds computed form the average transition speed on one step, five steps, and the ten steps. By considering many steps, the walking speed becomes more stable while it also causes some delay. Based on our experiment, using the average transition speed of five steps seems to be appropriate.

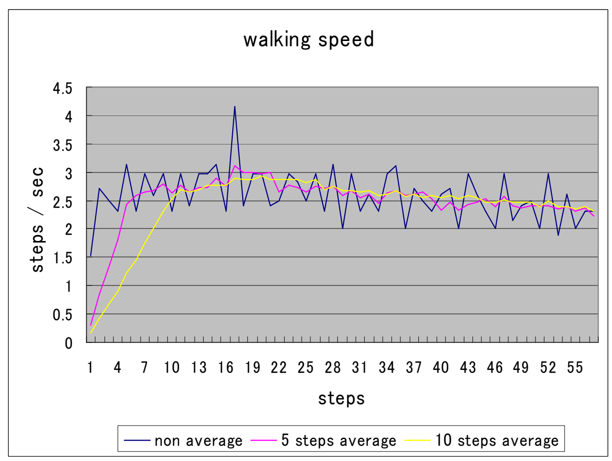

Figure 5. Walking speed on the experiments

5. Future Work

One of our future works is to revise the feature vector. The accuracy can be improved by adding other parameters such as the positions of feet to the feature vector. Currently the SVM cannot distinguish not walking state and the state transiting from leg up state to leg down state. These states could be distinguishable by introducing the vertical position of the feet to the feature vector. However, adding irrelevant parameters to the feature vector could also make the recognition results worse. Therefore, we have to choose the appropriate parameters carefully.

We are also trying to apply our method to other kinds of motions. Although the framework of our method is applied to any motion, other features may be needed to be introduced depending on the target motions. For instance, the velocities and positions of the foot and the waist are essential for the recognition of jump motions. Moreover, since motion states depend on motion types, state machines must be designed for each type of motion.

The recognition of two or multiple motions simultaneously (for example, raising hands and walking) [2] is not considered on this work. We consider that this also can be handled by our framework by preparing SVMs and a state machine for each motion and making them work in parallel.

6. Conclusions

In this paper, we presented a motion recognition method that combines SVM and state machine. We applied our method to the recognition of walk-in-position motion. The experimental results showed a high accuracy. Future work includes to revise the feature vector and to apply our method to other kinds of motions.

References

[1] Masaki Oshita, "Motion-Capture-Based Avatar Control Framework in Third-Person View Virtual Environments", ACM SIGCHI International Conference on Advance in Computer Entertainment Technology 2006 (ACE 2006). 9 pages, June 2006.

[2] Taketoshi Mori, Haruhiko H. Asada, "Design and Architecture of an Agent-based Physiological Human Model", BME: Bio Medical Engineering, Vol.16, No.8, pp. 46-51, 2002.

[3] Hideaki Takasaki, Shinkichi Inagaki, and Tatsuya Suzuki, "Modeling and Analysis of Walking Motion based on Stochastic Switched Link Model", JSME Conference on Robotics and Mechatronics 2006, 2P1-A13, 2006.

[4] Kiyoshi Ikehara, Atsushi Maebara, Takayuki Kitasaka, Kensaku Mori, Yasuhito Suegawa, "A Study on Hand Motion Recognition Using Tracking with Active Cameras and Template Matching", Technical report of IEICE PRMU, Vol. 102, No. 652, PRMU2002-212, pp. 7-12, 2002.

[5] Kenichi Noma, Mitsuru Nakai, Hiroshi Shimodaira, Shigeki Sagayama, " Sequential-Pattern Recognition by Support Vector Machines using Dynamic Time-Alignment Kernels" , Technical report of IEICE PRMU, Vol.100, No.507, PRMU2000-128, pp. 63-68, 2000.

[6] Yuji Kasuya, Hayato Yamana, "Recognition of Similar Character Pairs with Two Types of SVMs for Online Mathematical Expression Recognition" , Technical report of IEICE Thought and language, Vol.105, No.612, TL2005-58, pp. 55-60,2005.

[7] Nello Cristianini, Jhon Shawe-Taylor, An Introduction to Support Vector Machines (Tsuyoshi Okita, Trans), kyoritsu, 2000.

[8] Chih-Chung Chang and Chih-Jen Lin, LIBSVM:a library for support vector machines, 2001. Software available http://www.csie.ntu.edu.tw/~cjlin/libsv

[9] Kaoru Kobayashi, Hitoshi Sato, "Improvement of Gait Ability by Resistive Component Movement", Rigakuryoho Kagaku, Vol.21, No.4, pp. 411-415, 2006.

[10] Ascension Technology, MotionStar. http://www.ascension-tech.com/

[11] Viewworks Company, Motion Capture Data Base RIKIYA, 1999.