

Article

Depth-Sensor-Based Monitoring of Therapeutic Exercises

Mu-Chun Su ¹, Jhih-Jie Jhang ¹, Yi-Zeng Hsieh ^{1,2,*}, Shih-Ching Yeh ³, Shih-Chieh Lin ¹,
Shu-Fang Lee ⁴ and Kai-Ping Tseng ⁴

¹ Department of Computer Science and Information Engineering, National Central University, Taoyuan City 32001, Taiwan; E-Mails: muchun@csie.ncu.edu.tw (M.-C.S.); fieldddog@gmail.com (J.-J.J.); yakumolin@gmail.com (S.-C.L.)

² Department of Management and Information Technology, Southern Taiwan University of Science and Technology, Tainan City 71005, Taiwan

³ School of Information Science and Technology, Fudan University, Shanghai 200433, China; E-Mail: yeshiqing@fudan.edu.cn

⁴ Department of Rehabilitation, Landseed Hospital, Taoyuan City 324, Taiwan; E-Mails: leesf@landseed.com.tw (S.-F.L.), tsengkp@landseed.com.tw (K.-P.T.)

* Author to whom correspondence should be addressed; E-Mail: yizeng.hsieh@gmail.com; Tel.: +886-6-253-3131 (ext. 8122); Fax: +886-6-242-2029.

Academic Editor: Vittorio M. N. Passaro

Received: 21 July 2015 / Accepted: 30 September 2015 / Published: 9 October 2015

Abstract: In this paper, we propose a self-organizing feature map-based (SOM) monitoring system which is able to evaluate whether the physiotherapeutic exercise performed by a patient matches the corresponding assigned exercise. It allows patients to be able to perform their physiotherapeutic exercises on their own, but their progress during exercises can be monitored. The performance of the proposed the SOM-based monitoring system is tested on a database consisting of 12 different types of physiotherapeutic exercises. An average 98.8% correct rate was achieved.

Keywords: SOM; motion trajectory; spatial-temporal pattern recognition; therapeutic exercise

1. Introduction

At the rehabilitation departments of hospitals in Taiwan, it is not unusual that several patients are assigned to one exercise physiologist or physical therapist during their rehabilitation treatment in a clinical room. One-on-one clinical service is not possible, therefore an exercise physiologist or a physical therapist cannot know how well his or her patients are practicing the assigned exercises. In addition, patients themselves usually do not have the enough knowledge about whether they are practicing physiotherapeutic exercises correctly, leading to ineffective rehabilitation and even producing adverse compensation effects. Therefore, to alleviate the burdens of exercise physiologists or physical therapists, a physiotherapeutic exercise monitoring system which can automatically measure how well a patient is practicing the suggested exercises deserves to be developed [1–3]. A good physiotherapeutic exercise monitoring system should be able to provide a measurement report about how well a patient is practicing a suggested exercise, facilitate the effective execution of the assigned rehabilitation programs, issue a warning alarm signal when an incorrect exercise is detected, *etc.* For this to happen, it requires an effective motion trajectory recognition algorithm which is able to evaluate whether the physiotherapeutic exercise performed by a patient matches the corresponding assigned exercise.

Automatic motion trajectory recognition turns out to be very challenging because motion trajectories are spatio-temporal patterns. Speech recognition and gesture recognition are another two typical examples of spatio-temporal pattern recognition problems. Several reviews of the topics related to the domain of human motion analysis can be found in [4–10]. Basically, there are three kinds of approaches to spatio-temporal pattern recognition. The simplest way is first to turn a temporal sequence into a spatial pattern and then to employ a template matching technique to compute the degree of similarity between the test pattern and the template pattern. The dynamic time warping (DTW) algorithm provides the effect of a non-linear normalization process in order to make the similarity measure operate successfully [11]. The DTW algorithm operates by stretching the template pattern and measuring the amount of stretching required. The less stretching needed, the more similar the patterns are. Although the DTW algorithm is easy to implement, it requires substantial computation to reach an optimal DTW path. The other approach is to train recurrent neural networks [12]. The time delay neural network (TDNN) whose hidden nodes and output nodes are replicated across time is one of the popular recurrent neural networks [13]. Time-delayed input frames allow the TDNN to be able to respond to time-varying signals, however, one may also find that it usually takes a lot of time to train a recurrent neural network. Another approach is to employ Hidden Markov Models (HMMs) to recognize spatial-temporal patterns. The basic idea behind the HMM is that a spatio-temporal signal can be characterized by a parametric stochastic process. A HMM can be thought of as a finite-state machine where the transitions between the states are governed by probabilistic laws. The key parameters to be determined in an HMM are: (1) the number of states in the model; (2) the state-transition probability distribution; (3) the observation symbol probability distribution; and (4) the initial-state distribution [14–16]. The price paid for HMM is that we have to collect a great amount of data and a lot of time is required to estimate the corresponding parameters in HMMs. In addition to those three popular approaches, recently, several different approaches to solve motion capture data have also been proposed [17–27]. For example, the concept of motion templates (MTs) which can capture the spatio-temporal characteristics of an entire motion class in a compact matrix representation was introduced in [17,19]. The idea of incorporating spatio-temporal

invariance into the geometric features and adaptive segments was presented in [18]. Keogh *et al.* proposed a novel technique to speed up similarity search under uniform scaling based on bounding envelopes [20], and a novel fast method for global similarity searches was discussed in [21]. An efficient method for fully automatic temporal segmentation of human motion sequences was proposed in [22]. Baumann *et al.* presented a data-driven method for recognizing human full body actions from live motion data [23]. Morris *et al.* proposed REcoFit which is a system for automatically tracking repetitive exercises [24], while Beaudoin *et al.* presented a technique to automatically distill a motion-motif graph from an arbitrary collection of motion capture data [25], and Bernard *et al.* proposed an exploratory search and analysis system for sequences of human motion in large motion capture data collections [26]. Wilhelm *et al.* proposed a visual-interactive system for the exploration of horse motion data [27]. In the system, they employed the SOM algorithm as a non-linear projection technique to provide a data aggregation mechanism to facilitate visual overview.

Numerous reports have been written about the successful applications of the self-organizing feature map (SOM) algorithm [28]. These applications widely range from simulations used for the purpose of understanding and modeling of computational maps in the brain to subsystems for engineering applications [29]. Recently, several different approaches to the use of SOM in spatio-temporal pattern recognition have been proposed [30–34]. Gao *et al.* [30] and Fang *et al.* [31] used the SOM as an implicit feature extractor of different signers for continuous HMM. Corradini *et al.* regarded the SOM as a quantizer of 32 defined sub-gestures for discrete HMM [32]. Huang and Kuh proposed a neural network system which combines a SOM and a multi-layer perceptron (MLP) for the problem of isolated word speech recognition [33]. In one of our previous works [34], we proposed a SOMART system for the recognition of hand gestures. The proposed SOMART system integrates the SOM and the ART algorithm.

In this paper, we propose the SOM-based monitoring system of which core module is the SOM-based motion trajectory recognition algorithm. The algorithm starts with the generation of basic posture unit map. The sequence of a motion trajectory is transformed into a 2-D trajectory map based on the basic posture unit map. Then the problem of recognizing motion trajectories is transformed to the problem of recognizing 2-D trajectory maps. Finally, an unknown motion trajectory is classified to be the motion trajectory with the maximum similarity in the motion templates via a template matching technique.

The remainder of the paper is organized as follows: we first briefly review the SOM algorithm in Section 2. Then the proposed SOM-based motion trajectory recognition algorithm is discussed in Section 3. In Section 4, we give the results obtained by applying the algorithm to a database consisting of 12 different types of physiotherapeutic exercises. Finally, Section 5 concludes the paper.

2. Brief Review of the SOM Algorithm

The training algorithm proposed by Kohonen for forming a feature map can be summarized as follows [28]:

- Step 1: Initialization: Choose random values for the initial weights $w_j(0)$.
- Step 2: Winner Finding: Find the winning neuron j^* at time k , using the minimum-distance Euclidean criterion:

$$j^* = \arg \min_j \|\underline{x}(k) - \underline{w}_j\|, \quad j = 1, \dots, M \times N \quad (1)$$

where $\underline{x}(k) = [x_1(k), \dots, x_n(k)]^T$ represents the k th input pattern, $M \times N$ is the total number of neurons, and $\|\cdot\|$ indicates the Euclidean norm. Depending on the application of interest, the response of the network could be either the index of the winning neuron or the weight vector [28].

- Step 3: Weight Updating: Adjust the weights of the winner and its neighbors, using the following rule:

$$\underline{w}_j(k+1) = \underline{w}_j(k) + \eta(k) \Lambda_{j^*}(k) [\underline{x}(k) - \underline{w}_j(k)] \quad (2)$$

where $\eta(k) = \eta_0 \left(1 - \frac{k}{N_K}\right)$, N_K is the number of iterations, $\Lambda_{j^*}(k)$ is the topological neighborhood function of the winner neuron j^* at time k . A typical choice of $\Lambda_{j^*}(k)$ is the Gaussian-type function:

$$\Lambda_{j^*}(k) = \exp\left(-\frac{d_{j^*,j}^2}{2\sigma(k)^2}\right) \quad (3)$$

where the parameter $\sigma(k)$ is the “effect width” of the topological neighborhood and $d_{j^*,j}^2$ is the lateral distance between neurons j^* and j in the discrete output space. The “effect width” measures the degree to which the excited neurons in the vicinity of the winning neuron [28]. It should be emphasized that the success of the map formation is critically dependent on how the values of the main parameters (*i.e.*, $\eta(k)$ and $\Lambda_{j^*}(k)$), initial values of weight vectors, and the number of iterations are predetermined. In our experiments discussed in Section 4, the initialization method proposed in [35] was utilized to initialize the weight vectors to quickly construct a good initial map. As for the values of the main parameters (*i.e.*, $\eta(k)$ and $\Lambda_{j^*}(k)$), we set $\eta_0 = 0.1$, $N_K = 100$, and $\sigma(k) = 1.0$. Depending on the application of interest, the response of the network could be either the index of the winning neuron or the weight vector [17].

- Step 4: Continuation. Increment time step k by one and go back to step 2 until some kind of termination criterion is met.

3. The SOM-Based Motion Trajectory Recognition Algorithm

Before we explain how the proposed SOM-based motion trajectory recognition algorithm works, we need to first introduce how we capture the information about the motion trajectories. The motion trajectories are captured via a Kinect sensor. To make the features insensitive to personal height, the standing distance from the user to the Kinect sensor, and the facing orientation with respect to the Kinect sensor, we need to adopt an effective coordinate transformation. In this paper, we propose the body-plane coordinate transformation to transform the original world coordinates to the new coordinate system with the origin of coordinate centered at the spine location of the user. Then the X-Y plane is transformed to the coronal plane, the Y-Z plane to the sagittal plane, and the X-Z plane to the transverse plane, as shown in Figure 1.

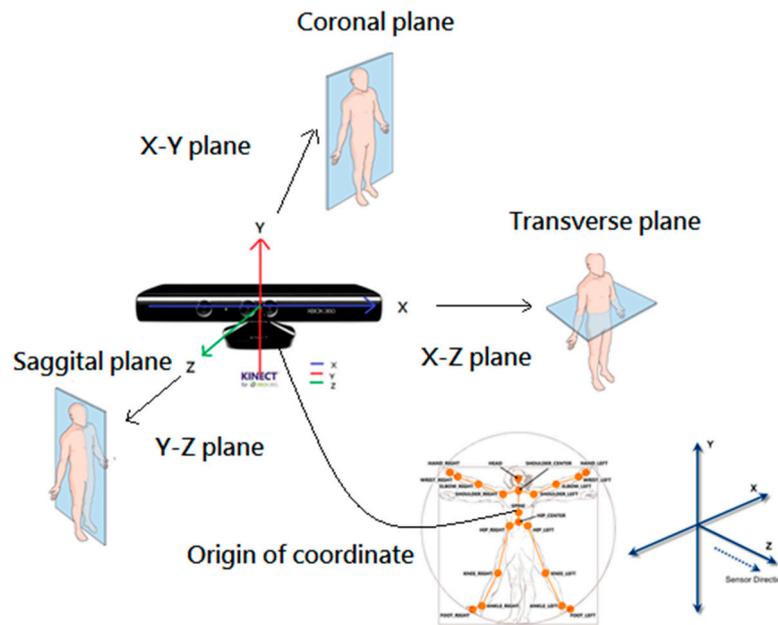
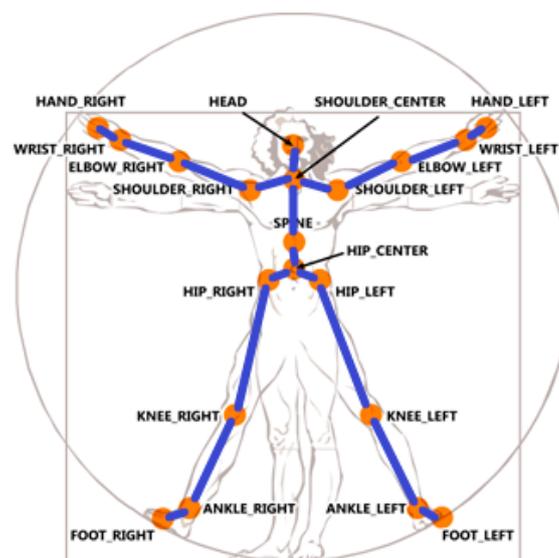


Figure 1. The body-plane coordinates transformation.

To implement the body-plane transformation, we need three orthogonal vectors to form the new coordinates located at the spine position of the user. These three orthogonal vectors consist of the vector from the spine position to the shoulder center position, the vector from the spine position to the shoulder left position, and the cross product vector of the former two vectors. Based on the skeleton information provided from the Kinect sensor, we decide to choose 19 unit vectors of the body segment vectors as shown in Figure 2a and 14 joint angles as shown in Figure 2b as the features for each image frame of a motion trajectory. The joint angle is the angle between two connected body segments. Therefore, the feature vector is a 71-dimensional vector because the total dimensions for the 19 unit vectors are 3×19 and the total dimensions for the 14 joint angles are 14.



(a)

Figure 2. Cont.

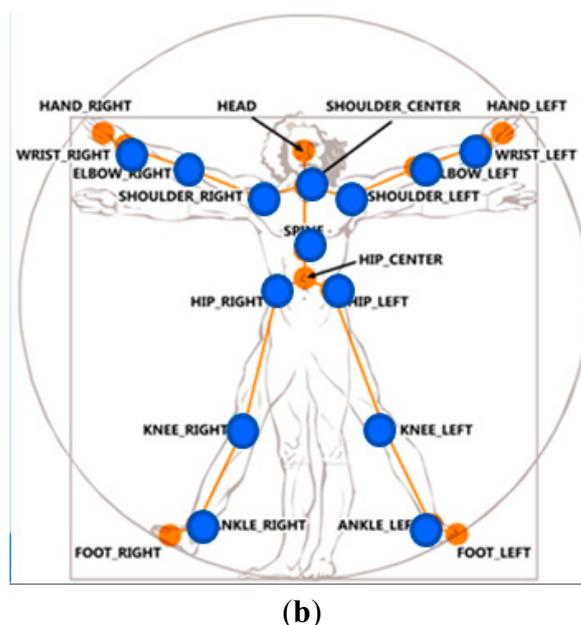


Figure 2. The 71 features for the motion trajectory extracted from the skeleton information provided from a Kinect sensor. (a) The 19 unit vectors; (b) The 14 joints.

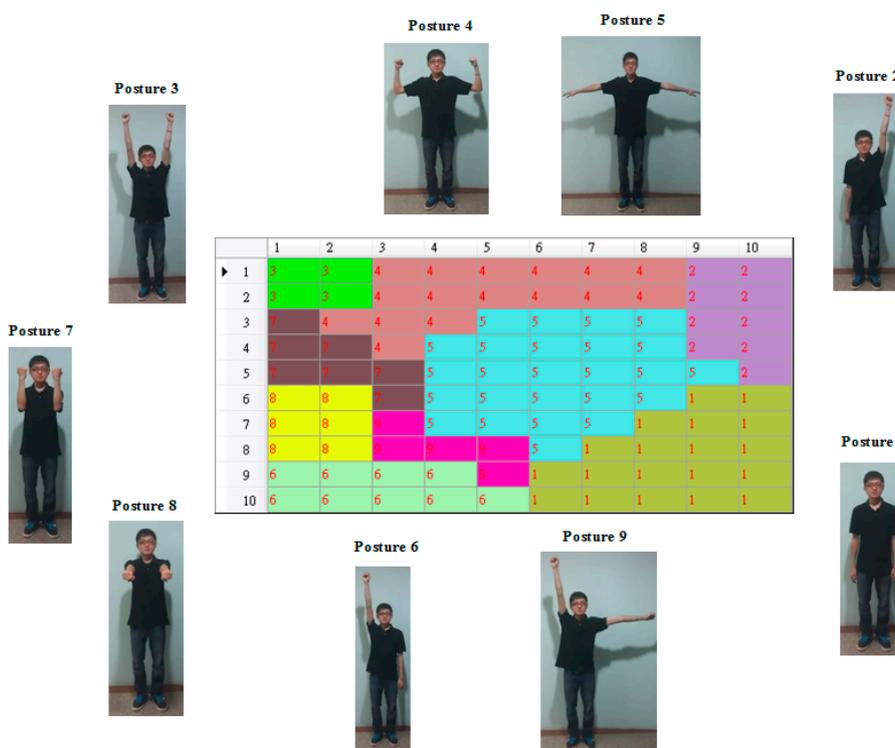
The proposed SOM-based motion trajectory recognition algorithm involves the following four steps:

Step 1. Generating a basic posture unit map

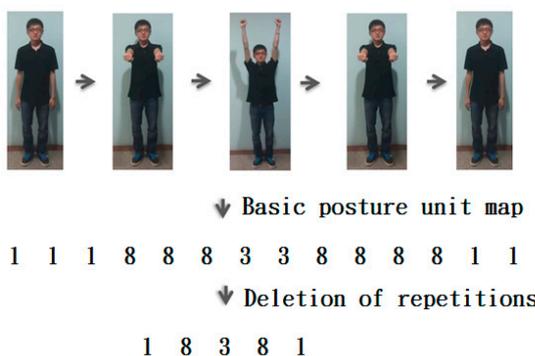
We use a data set consisting of many different motion trajectories practiced by different persons to run the SOM algorithm to construct a topologically preserved map with $M \times N$ neurons. The main mission of the SOM algorithm is to cluster motion trajectories into $M \times N$ small clusters. In our simulations, we found that the size 10×10 could produce a good recognition result. Basically, these 100 clusters can directly be used to serve as the 100 basic posture units for motion trajectories. Based on the 100 basic posture units, an individual motion trajectory can be represented as a sequence of a combination of 100 different basic postures with the sequence length equal to the number of the frames of the trajectory. To increase the tolerance to the person variability problem, we do not directly make each neuron to correspond to a basic posture unit. We fully utilize the topology-preserving characteristic of the SOM algorithm and then merge neighboring small clusters of the SOM map to form so-called “superclusters”. This merging process can reduce the number of basic posture units from 100 to the number of “superclusters”. Each supercluster then contains many small clusters. Via the use of superclusters, we allow each basic posture unit to have many templates so the sensitivity of the person variability can be reduced to some extent. We first use the U-matrix algorithm [36,37] to transform the trained map with 10×10 neurons into a digital image with 10×10 pixels. In the U-matrix algorithm, distances of each neuron to each of its immediate neighbors are calculated and visualized using gray shade. High values on the U-matrix mean large distance between neighboring map units, and thus indicate cluster borders. Clusters are typically uniform areas of low values and the number of clusters has to be determined by visualization. To more objectively determine the number of clusters, the watershed algorithm [38] is then adopted to segment the U-matrix image into several catchment basins. A simple merging scheme is adopted to merge shallow basins to their neighboring deep basins in order

to prevent over-segmentation. The posture located at a basin is the most representative posture for the neurons located at the corresponding basin.

The nine resultant basic postures are shown in Figure 3a. For example, while the first basic posture represents the postures located at the lower right corners, the second basic posture unit represents the postures located at the upper right corners. By viewing Figure 3a, we may find that these two basic postures are similar except that the left hand points up at the second basic posture. This observation matches the topological characteristic of the SOM algorithm. Based on the basic posture map, each motion trajectory can be transformed into a sequence of basic postures. For example, a motion trajectory shown in Figure 3b can be represented as a sequence, 1118883388811. Since repetitions contain no further information, we will delete repetitions in the sequence and then use the sequence 1,8,3,8,1 to represent the motion trajectory.



(a)



(b)

Figure 3. The basic posture unit map. (a) The basic posture map consisted of nine basic postures (b) A motion trajectory and its corresponding sequence.

Step 2. Transforming a motion trajectory into a trajectory map

As mentioned in the previous step, each motion trajectory can be represented as a sequence of nine symbols (*i.e.*, nine basic postures). We do not directly use a one-dimensional sequence to represent a motion trajectory because the length will vary with different motion trajectories. Therefore, we propose the use of a 2-dimensional trajectory map with a fixed size to represent a motion trajectory. The resultant sequence is projected on the basic posture map to generate a so-called “trajectory map” in the following way. Each frame of a motion trajectory is sequentially input to the basic posture map to locate the corresponding winner on the map. The basin corresponding to the winner will be rendered using a gray shade. Darker shades on the trajectory map mean earlier frames. Therefore, a motion trajectory will result in a trajectory map. Via this kind of rendering scheme, the temporal information will be retained in the trajectory map. We use an example shown in Figure 4 to illustrate how to generate a trajectory map. The motion trajectory shown in Figure 4a is consisted of basic posture units 1 and 5. Its corresponding sequence is 151. Therefore, its corresponding trajectory map is shown in Figure 4b where the gray levels for the basic posture units, 2, 3, 4, 6, 7, 8, and 9 are all white except the two regions corresponding to the basic postures 1 and 5. The binary string labeled at each cell of the basic posture unit map shown in Figure 4b represents the appearance orders of the corresponding basic postures. For example, the corresponding sequence is 151; therefore, there are three digits in the binary string. The first basic posture appears twice at the sequence and appears at the first time and the third time respectively; therefore, the binary string 101 is labeled at the region of the first basic posture located at the right bottom corner of the basic posture map.



(a)

	1	2	3	4	5	6	7	8	9	10
▶ 1	000	000	000	000	000	000	000	000	000	000
2	000	000	000	000	000	000	000	000	000	000
3	000	000	000	000	010	010	010	010	000	000
4	000	000	000	010	010	010	010	010	000	000
5	000	000	000	010	010	010	010	010	010	000
6	000	000	000	010	010	010	010	010	101	101
7	000	000	000	010	010	010	010	101	101	101
8	000	000	000	000	000	010	101	101	101	101
9	000	000	000	000	000	101	101	101	101	101
10	000	000	000	000	000	101	101	101	101	101

(b)

Figure 4. The trajectory map. (a) The motion trajectory and its corresponding sequence 151; (b) The resultant trajectory map.

Step 3. Generating trajectory map templates

Human variability causes the same type of a motion trajectory to be conducted differently each time. Even for the same person, different instances of a motion trajectory may not be identical. Therefore, a motion trajectory may result in more than one trajectory maps. There are two approaches to solve the variability problem. One approach is to store multiple templates for each motion trajectory. A problem associated with this kind of approach is the determination of the number of templates. The other approach is to generate one most representative template for each motion trajectory. We adopt the second approach to generate one template for each motion trajectory. We collect many instances for each motion trajectory and then the longest common subsequence (LCS) algorithm is adopted to search the longest common subsequence of multiple sequences [39]. An example is given in Table 1. In this example, there were three kinds of motion trajectories and four different users were asked to execute the motion trajectory for one time. Each motion trajectory is first converted into a sequence of basic postures via the basic posture unit map. Then resultant template for each kind of trajectory is shown at the last row of the table.

Table 1. The example of finding the template for each motion trajectory.

User	Trajectory 1	Trajectory 2	Trajectory 3
1	6, 1, 2, 1, 6	6, 11, 9, 11, 6	6, 1, 3, 6
2	6, 1, 2, 9, 2, 1, 3, 6	6, 10, 9, 10, 11, 6	6, 1, 5, 3, 6
3	6, 2, 1, 6	6, 11, 10, 9, 6, 11, 6	6, 5, 1, 5, 3, 6
4	6, 5, 4, 2, 1, 6	6, 10, 9, 11, 6	6, 1, 6
LCS	6, 2, 1, 6	6, 10, 9, 11, 6	6, 1, 6

Step 4. Decision making

To classify an unknown motion trajectory, we first project the sequence of the motion trajectory into the basic posture unit map to create the corresponding trajectory map. Then we match the trajectory map with the templates of the motion trajectories in the database. Finally, the unknown motion trajectory is classified to be the motion trajectory with the largest similarity in the database. One immediate problem is how to measure the similarity degree between two trajectory maps. The simplest way to measure the similarity degree between two trajectory maps is the use of the image correlation computation. The similarity degree S_k between the present trajectory map to be recognized, $p(x, y)$, and the template map for the k th motion trajectory, $t_k(x, y)$, can be computed as follows:

$$S_k = \frac{\sum_x \sum_y [p(x, y) - \bar{p}(x, y)][t_k(x, y) - \bar{t}_k(x, y)]}{\{\sum_x \sum_y [p(x, y) - \bar{p}(x, y)]^2 \sum_x \sum_y [t_k(x, y) - \bar{t}_k(x, y)]^2\}^{1/2}} \quad (4)$$

where $\bar{p}(x, y)$ and $\bar{t}_k(x, y)$ represent the average value of $p(x, y)$ and $t_k(x, y)$, respectively. If the computed similarity degree is below a pre-specified threshold, then the test motion trajectory is claimed to be not similar to any template in the database.

4. Experimental Results

To evaluate the performance of the proposed SOM-based motion trajectory recognition algorithm, four databases consisting of 12 physiotherapeutic exercises recommended for patients with Parkinson's disease were generated. These 12 physiotherapeutic exercises are shown in Figure 5. Ten subjects as shown in Table 2 were invited to generate the four databases shown in Table 3.

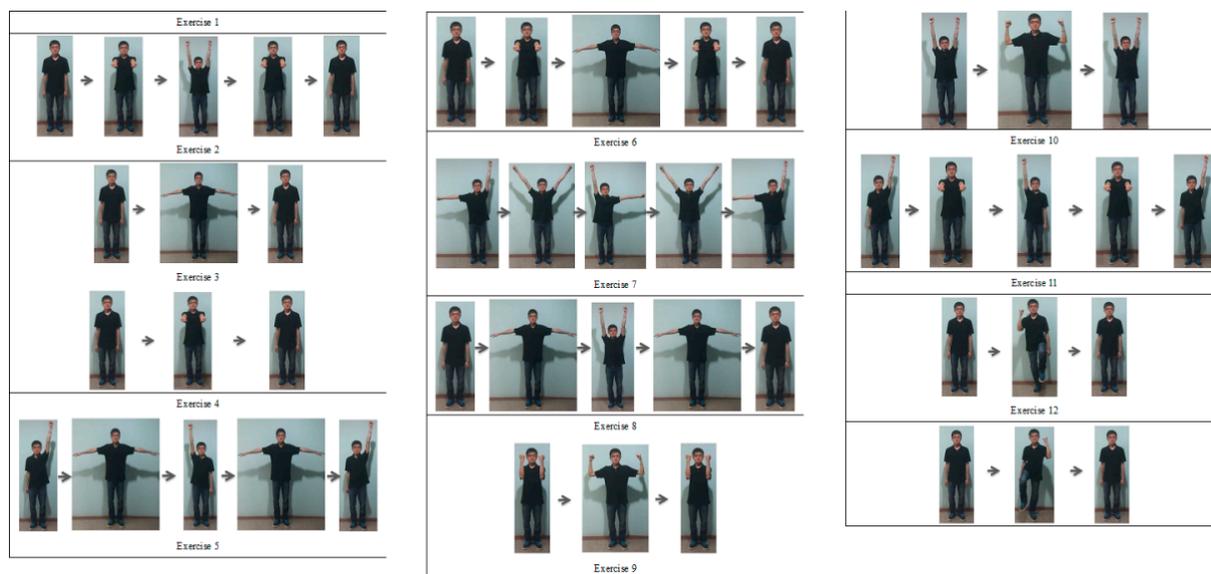


Figure 5. The 12 exercises recommended for patients with Parkinson's disease.

Table 2. The ten persons invited to generate the four databases.

Subject	Gender	Height (cm)	Weight (kg)
1	Male	176	58
2	Female	162	47
3	Male	174	68
4	Male	177	72
5	Male	184	78
6	Female	160	43
7	Male	173	64
8	Female	163	45
9	Male	173	68
10	Male	170	65

The first five subjects including four males and one female were asked to perform the 12 exercises. Each subject practiced each exercise for one time to generate the first database called the template database. This database was used to generate the basic posture unit map as shown in Figure 3a. Then we used the LCS method to generate one template for each exercise from the trajectories practiced by five persons. The resultant motion trajectory maps were shown in Figure 6. Obviously, these 12 trajectory maps look different. This observation confirms that the idea about representing motion trajectories in terms of trajectory maps works. These 12 trajectory maps were then stored as the templates for the use of testing the following four experiments.

Table 3. The four databases generated for testing the performance of the proposed method.

Database	Name	Subjects	Exercises
1	The template database	S1, S2, S3, S4, S5	Each exercise for one time
2	The user dependent database	S1, S2, S3, S4, S5	Each exercise for ten times
3	The user robustness database	S1, S2, S3, S4, S5	Each exercise for ten times under three different conditions (<i>i.e.</i> , exercise speed, pause, and facing orientation)
4	The user independent database	S6, S7, S8, S9, S10	Each exercise for ten times

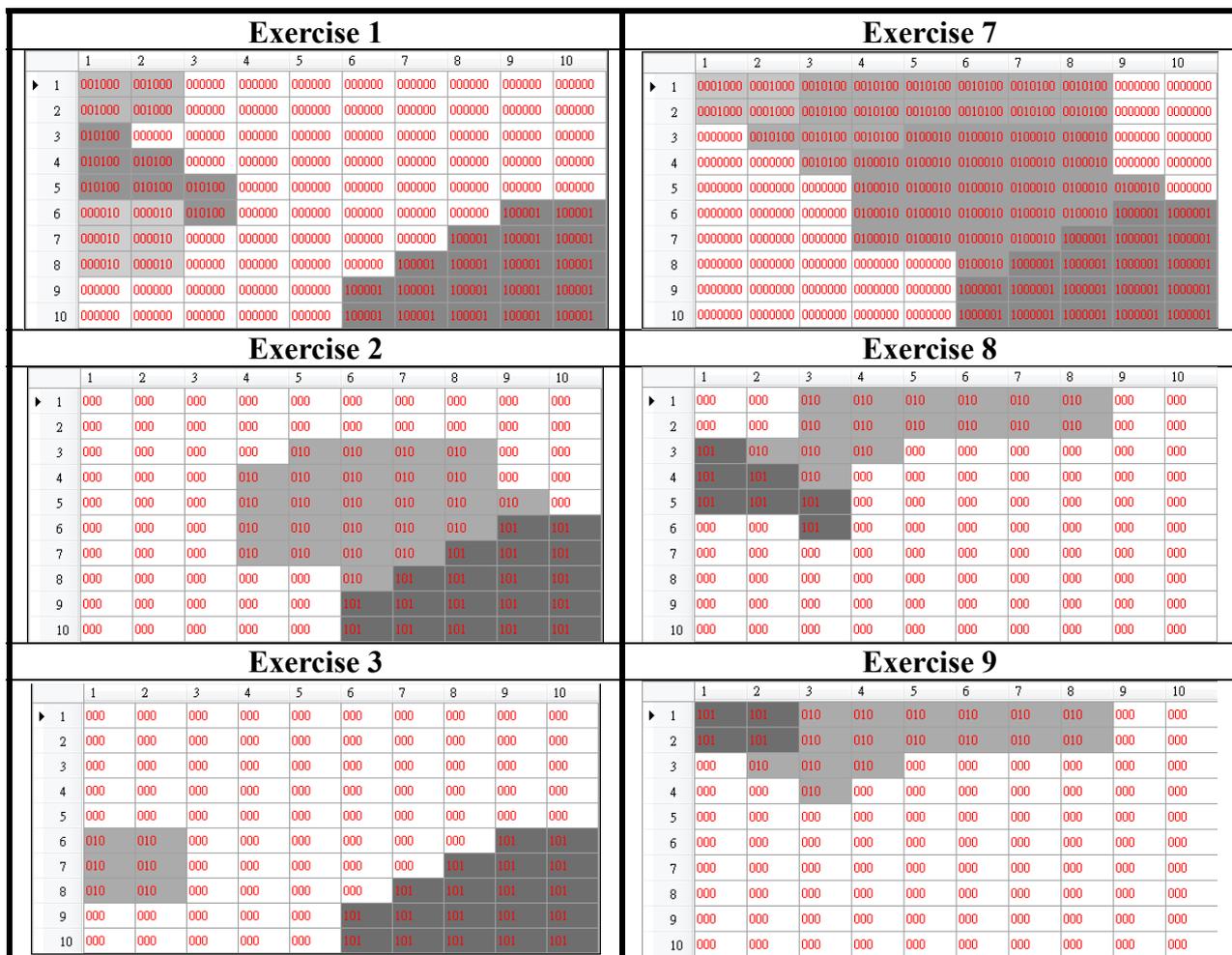


Figure 6. Cont.

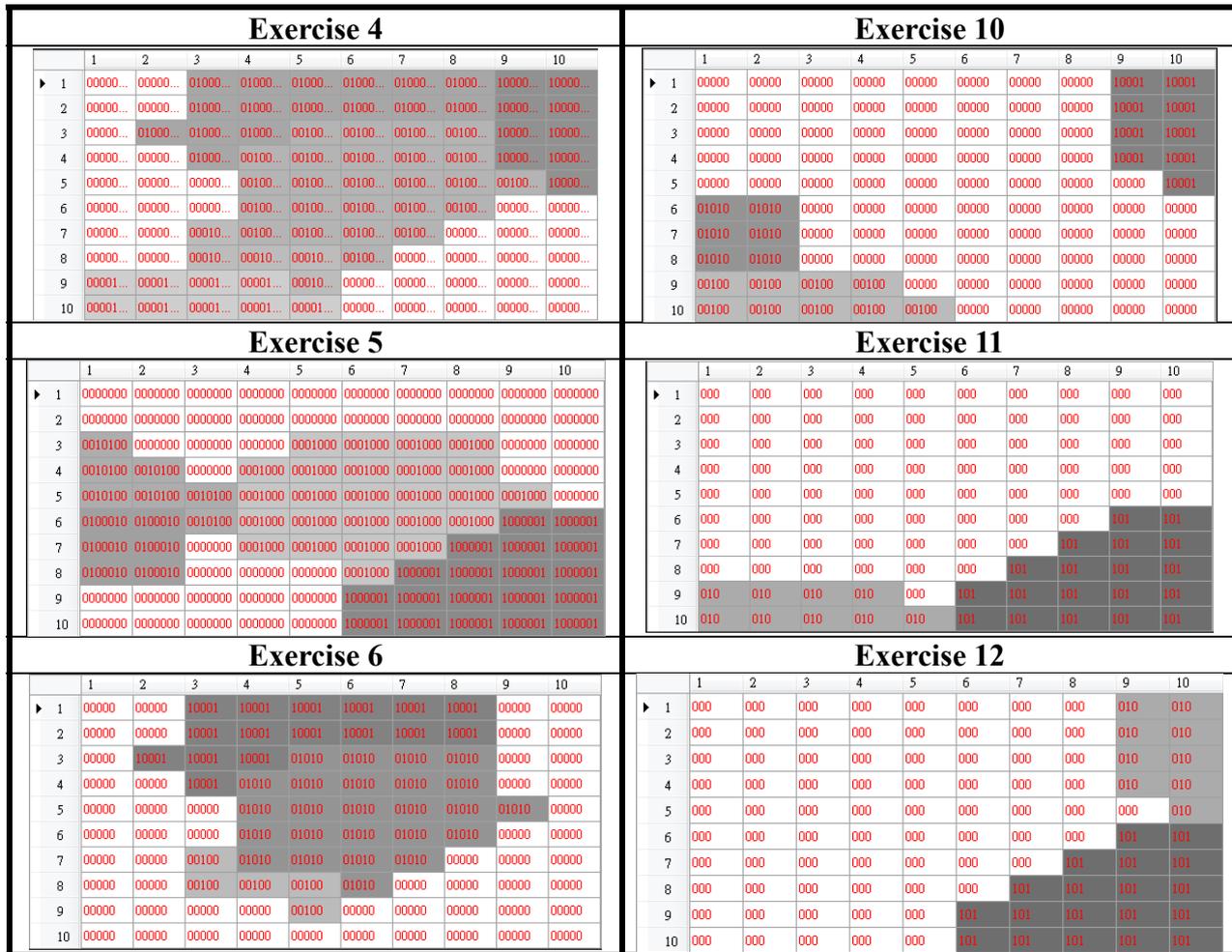


Figure 6. The 12 motion trajectory maps generated from the first database.

Experiment One: User Dependent Test

We asked the same five subjects to practice the 12 exercises again. Each exercise was repeated ten times for each person to generate the second database called the user dependent database. The user dependent database was used for the purpose of testing. The experimental results showed that the average recognition rates achieved by the proposed method were 100% correct as shown in Table 4. For the comparison purpose, we adopted the popular DTW method and Support Vector Machines (SVMs) to test the second database. The unknown motion trajectory is classified to be the motion trajectory with the smallest accumulated Euclidean distance in the database. Since there were five trajectories for each exercise, one randomly chosen trajectory was used for the template and the remaining four trajectories were used for testing. The recognition performance achieved by the DTW method was 78.3% correct. As for the training of the SVMs, we used the template database to train SVMs (*i.e.*, each exercise has only one training data point). The inputs presented to the SVMs were the trajectory map. Then the user dependent database was used to test the trained SVMs. The recognition performance achieved by the SVMs was 83.0% and 91.7% correct with the standard Gaussian radial basis function kernels and the polynomial kernels, respectively. Obviously, our proposed method outperformed the DTW method and the SVMs based on the comparison of correct recognition rate.

Table 5. Cont.

Subject		1	2	3	4	5	Average
Type							
4	Slow speed	100%	100%	100%	100%	100%	100%
	Pause	100%	100%	100%	100%	100%	100%
	45°	100%	100%	100%	100%	100%	100%
5	Slow speed	100%	100%	100%	100%	100%	100%
	Pause	100%	100%	100%	100%	100%	100%
	45°	100%	100%	100%	100%	100%	100%
6	Slow speed	100%	90%	100%	90%	100%	96%
	Pause	100%	100%	90%	100%	100%	98%
	45°	100%	100%	90%	90%	90%	94%
7	Slow speed	100%	100%	100%	100%	100%	100%
	Pause	100%	100%	100%	100%	100%	100%
	45°	100%	100%	100%	100%	100%	100%
8	Slow speed	100%	100%	100%	100%	100%	100%
	Pause	100%	100%	100%	100%	100%	100%
	45°	100%	100%	100%	100%	100%	100%
9	Slow speed	100%	90%	90%	100%	100%	96%
	Pause	100%	100%	100%	90%	90%	96%
	45°	90%	90%	100%	90%	100%	94%
10	Slow speed	100%	100%	100%	100%	100%	100%
	Pause	100%	100%	100%	100%	100%	100%
	45°	100%	100%	100%	100%	100%	100%
11	Slow speed	100%	100%	100%	100%	100%	100%
	Pause	100%	100%	100%	100%	100%	100%
	45°	100%	100%	100%	100%	100%	100%
12	Slow speed	100%	100%	100%	100%	100%	100%
	Pause	100%	100%	100%	100%	100%	100%
	45°	100%	100%	100%	100%	100%	100%
Average		100%	99%	99%	99%	99%	99.2%

Experiment Three: User Independent Test

To test whether the proposed recognition algorithm is user-independent, the remaining five subjects including three males and two females were asked to perform the 12 exercises. Each exercise was repeated for ten times for each subject to generate the fourth database called the user independent database. The fourth database was used for testing purpose. The experimental results showed that the average recognition rates achieved by the proposed method were 97.7% correct as shown in Table 6. Although the recognition performance degraded from 100% to 97.7%, the average recognition rate was still high. It means that the proposed recognition algorithm had good generalization ability.

Table 6. The recognition performance achieved by the proposed method for the user-independent test.

Subject Type	Subject					Average
	6	7	8	9	10	
1	100%	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%	100%
4	100%	100%	100%	100%	100%	100%
5	100%	100%	100%	100%	100%	100%
6	70%	90%	100%	100%	70%	86%
7	100%	100%	100%	100%	100%	100%
8	100%	100%	100%	100%	100%	100%
9	90%	100%	90%	90%	60%	86%
10	100%	100%	100%	100%	100%	100%
11	100%	100%	100%	100%	100%	100%
12	100%	100%	100%	100%	100%	100%
Average	96.7%	99.2%	99.2%	99.2%	94.2%	97.7%

Experiment Four: Grid Size Influence Test

In this experiment, we wanted to determine whether the grid size influences the recognition performance. Three different grid sizes, 6×6 , 8×8 , and 10×10 were compared here. The basic posture unit maps resulting from these three different grid sizes are shown in Figure 7.

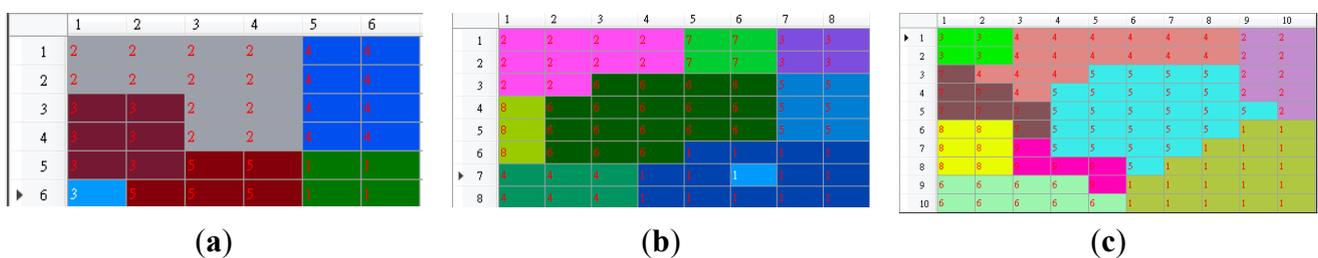


Figure 7. The basic posture unit maps resulted from grid size: (a) 6×6 , (b) 8×8 , and (c) 10×10 .

The numbers of basic posture units were 5, 8, and 9, for 6×6 , 8×8 , and 10×10 , respectively. The three resultant trajectory maps were shown in Figure 8. Then we used the user dependent database to test the recognition performance of the three different grid sizes. The performance comparison was tabulated in Table 7. Several interesting observations could be concluded as follows. First of all, the larger the grid size, the larger the number of the basic posture units. Secondly, the larger the grid size, the better the recognition performance. Thirdly, there existed more difference among the trajectory maps as the grid size increases; therefore, the recognition performance was improved as the grid size increased. Based on these observations, if we want to recognize more motion trajectories then we need a larger grid size to exhibit enough differences among trajectory maps.

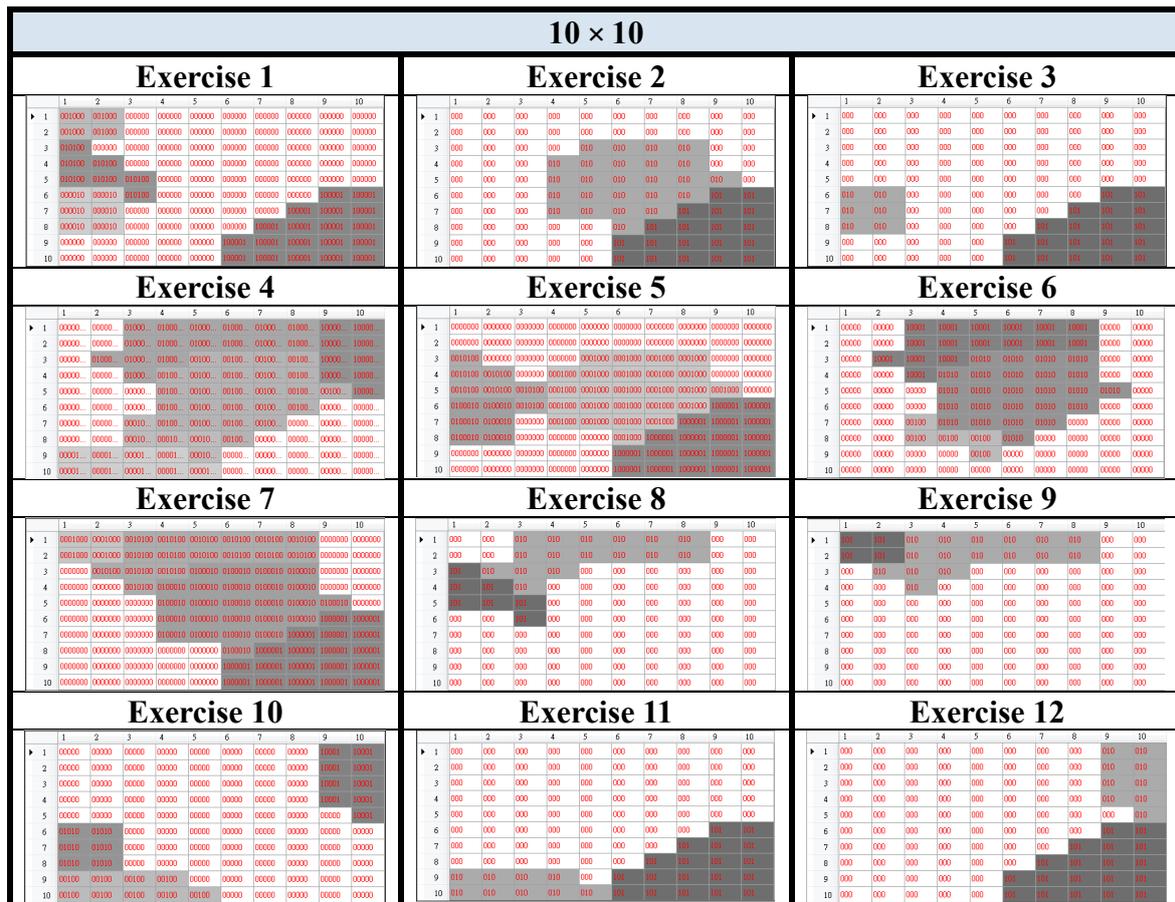


Figure 8. The trajectory maps resulted from grid size: 6 × 6, 8 × 8, and 10 × 10.

Table 7. The recognition performance achieved by the three different grid sizes.

Grid Size	6 × 6	8 × 8	10 × 10
Performance	33.67%	67%	100%

5. Conclusions

A method of recognizing spatio-temporal motion trajectories was proposed in this paper. Based on the proposed SOM-based motion trajectory recognition algorithm, we are able to implement a physiotherapeutic exercise monitoring system which can automatically measure how well a patient is practicing the suggested exercises. We used three databases to evaluate the effectiveness of the proposed SOM-based motion trajectory recognition algorithm. The average recognition rate that could be achieved was at least 97.7%. The robustness of the proposed algorithm was also verified by the second experiment.

Although the SOM algorithm has been applied in motion capture data for several years now, the major contribution of our method is the use of trajectory maps. Via a special kind of the proposed rendering scheme, the temporal information can be retained in the trajectory map. In addition, we adopt the image correlation computation to measure the similarity degree between two trajectory maps.

Since the Kinect depth sensor is more or less subject to the deployment environment (e.g., the distance between the sensor and the user, orientations, the lighting, etc.), our method is suitable for coarse-grained movements such as the twelve exercises shown in Figure 5. As for some more complex

or subtle rehabilitation exercises such as palmar pinch, wrist motion (e.g., flexion, extension, and deviation), hand-axis motion (e.g., pronation and supination), *etc.*, our present method may not work well. This can be partly attributed to the features of the Kinect sensor because the current Kinect sensor is inadequate to provide the fine-grained rotation information about the fingers and the wrists. Inexpensive tracking cameras can provide enough information about the fine-grained rotation exercises, so we will try to generalize our method to those complex or subtle rehabilitation exercises to meet the needs of physiotherapy patients in the future.

Acknowledgments

This paper was partly supported by Ministry of Science and Technology, Taiwan, under MOST 104-2221-E-008-074-MY2, MOST 103-2911-I-008-001 (support for the Center for Dynamical Biomarkers and Translational Medicine, National Central University, Taiwan), MOST 104-2622-E-218-006-CC3, MOST 104-2623-E-008-006-D and LSH-NCU Joint Research Foundation NCU-LSH-104-A-004.

Author Contributions

Mu-Chun Su, Jhih-Jie Jhang, Yi-Zeng Hsieh, Shih-Ching Yeh, and Shih-Chieh Lin designed the algorithm, performed experiments and prepared the manuscript. Shu-Fang Lee and Kai-Ping Tseng made the research plan, provided the guidance of the research and reviewed the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Niazmand, K.; Tonn, K.; Kalaras, A. A measurement device for motion analysis of patients with Parkinson's disease using sensor based smart clothes. In Proceedings of the 5th International Conference on Pervasive Computing Technologies for Healthcare, Dublin, Ireland, 23–26 May 2011.
2. Huang, J.D. Kinerehab: A Kinect-based system for physical rehabilitation—A pilot study for young adults with motor disabilities. *Res. Dev. Disabil.* **2011**, doi:10.1016/j.ridd.2011.07.002.
3. Yeager, R. *An Automated Physiotherapy Exercise Generator*; Technical Report for Electrical Engineering and Computer Sciences: Berkeley, CA, USA, May 2013.
4. Moeslund, T.B.; Granum, E. A survey of computer vision-based human motion capture. *Comput. Vis. Imag. Underst.* **2001**, 231–268.
5. Moeslund, T.B.; Hilton, A.; Kruger, V. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Imag. Underst.* **2006**, 104, 90–126.
6. Wang, J.J.; Singh, S. Video analysis of human dynamics: A survey. *Real-Time Imag. J.* **2003**, 9, 320–345.
7. Wang, L.; Hu, W.; Tan, T. Recent developments in human motion analysis. *Pattern Recognit.* **2003**, 36, 85–601.

8. Poppe, R. Vision-based human motion analysis: An overview. *Comput. Vis. Imag. Underst.* **2007**, *108*, 4–18.
9. Weinland, D.; Ronfard, R.; Boyer, E. A survey of vision-based methods for action representation, segmentation and recognition. *Comput. Vis. Imag. Underst.* **2010**, *15*, 224–241.
10. Aggarwal, J.K.; Ryoo, M.S. Human activity analysis: A review. *ACM Comput. Surv.* **2011**, *43*, 1–43.
11. Rabiner, L.R.; Juang, B.H. *Fundamentals of Speech Recognition*; Prentice Hall: Englewood Cliffs, NJ, USA, 1993.
12. Haykin, S. *Neural Networks—A Comprehensive Foundation*; Macmillan College Publishing Company: New York, NY, USA, 1994.
13. Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Lang, K.J. Phone recognition using time-delay neural networks. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 328–339.
14. Levinson, S.E.; Rabiner, L.R.; Sondhi, M.M. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell Syst. Tech. J.* **1983**, *62*, 1035–1073.
15. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE Proc.* **1989**, *77*, 257–285.
16. Lanchantin, P.; Pieczynski, W. Unsupervised restoration of hidden non stationary Markov chain using evidential priors. *IEEE Trans. Signal Proc.* **2005**, *53*, 3091–3098.
17. Müller, M.; Röder, T.; Clausen, M. Efficient content-based retrieval of motion capture data. *ACM Trans. Graph.* **2005**, *24*, 677–685.
18. Müller, M.; Röder, T. Motion templates for automatic classification and retrieval of motion capture data. In Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), Vienna, Austria, 2–4 September 2006.
19. Müller, M.; Baak, A.; Seidel, H. Efficient and robust annotation of motion capture data. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), New Orleans, LA, USA, 1–2 August 2009.
20. Keogh, E.; Palpanas, T.; Zordan, V.; Gunopulos, D.; Cardle, M. Indexing large human-motion databases. In Proceedings of the 30th International Conference on Very Large Data Bases, Toronto, ON, Canada, 31 August–3 September 2004.
21. Krüger, B.; Tautges, J.; Weber, A.; Zinke, A. Fast local and global similarity searches in large motion capture databases. In Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Madrid, Spain, 2–4 July 2010.
22. Vögele, A.; Krüger, B.; Klein, R. Efficient unsupervised temporal segmentation of human motion. In Proceedings of the 2014 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Copenhagen, Denmark, 21–23 July 2014.
23. Baumann, J.; Wessel, R.; Krüger, B.; Weber, A. Action graph: A versatile data structure for action recognition. In Proceedings of the GRAPP 2014-International Conference on Computer Graphics Theory and Applications, Lisbon, Portugal, 5–8 January 2014.
24. Morris, D.; Saponas, T.S.; Guillory, A.; Kelner, I. RecoFit: Using a wearable sensor to find, recognize, and count repetitive exercises. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14), Toronto, ON, Canada, 26 April–1 May 2014.

25. Beaudoin, P.; van de Panne, M.; Poulin, P.; Coros, S. Motion-motif graphs. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Dublin, Ireland, 7–9 July 2008.
26. Bernard, J.; Wilhelm, N.; Krüger, B.; May, T.; Schreck, T.; Kohlhammer, J. Motion explorer: Exploratory search in human motion capture data based on hierarchical aggregation. *Vis. Comput. Graphics*. **2013**, doi:10.1109/TVCG.2013.178.
27. Wilhelm, N.; Vögele, A.; Zsoldos, R.; Licka, T.; Krüger, B.; Bernard, J. Fury explorer: Visual-interactive exploration of horse motion capture data. In Proceedings of the Visualization and Data Analysis (VDA 2015), San Francisco, CA, USA, 8–12 February 2015.
28. Kohonen, T. *Self-Organization and Associative Memory*, 3rd ed.; Springer Berlin Heidelberg: Berlin, Germany, 1989.
29. Kohonen, T.; Oja, E.; Simula, O.; Visa, A.; Kangas, J. Engineering applications of the self-organizing map. *IEEE Proc.* **1996**, *84*, 1358–1383.
30. Gao, W.; Fang, G.; Zhao, D.; Chen, Y. A Chinese sign language recognition system based on SOFM/SRN/HMM. *Pattern Recognit.* **2004**, *37*, 2389–2402.
31. Fang, G.; Gao, W.; Zhao, D. Large vocabulary sign language recognition based on fuzzy decision trees. *IEEE Trans. Syst. Man Cybern. Part A* **2004**, *34*, 305–314.
32. Corradini, A.; Böhme, H.J.; Gross, H.M. A hybrid stochastic connectionist approach to gesture recognition. *Int. J. Artif. Intell. Tools* **2000**, *9*, 177–204.
33. Huang, Z.; Kuh, A. A combined self-organizing feature map and multilayer perceptron for isolated word recognition. *IEEE Trans. Signal Process.* **1992**, *40*, 2651–2657.
34. Su, M.C.; Hung, C.H.; Zhao, X.Y. A SOMART system for gesture recognition. *WSEAS Trans. Comput.* **2006**, *5*, 2764–2771.
35. Su, M.C.; Chang, H.C. Fast self-organizing feature map algorithm. *IEEE Trans. Neural Netw.* **2000**, *13*, 721–733.
36. Kraaijveld, M.A.; Mao, J.; Jain, A.K. A nonlinear projection method based on Kohonen's topology preserving maps. *IEEE Trans. Neural Netw.* **1995**, *6*, 548–559.
37. Mao, J.; Jain, A.K. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Trans. Neural Netw.* **1995**, *6*, 296–317.
38. Vincent, L.; Soille, P. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 583–598.
39. Neapolitan, R.; Naimipour, K. Longest Common Subsequence. In *Foundations of Algorithms Using C++ Pseudocode*, 3rd ed.; Jones & Bartlett Learning: Burlington, MA, USA, 2003.