Cyberphysical System With Virtual Reality for Intelligent Motion Recognition and Training

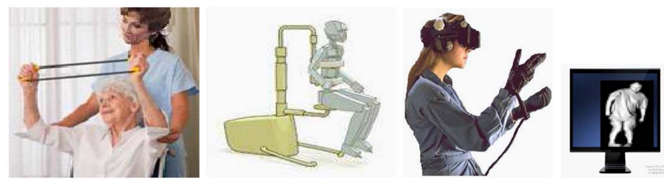
Fei Hu, Qi Hao, Qingquan Sun, Xiaojun Cao, Rui Ma, Ting Zhang, Yogendra Patil, and Jiang Lu, *Member, IEEE*

***Abstract*—In this paper, we propose to build a comprehensive cyberphysical system (CPS) with virtual reality (VR) and intelligent sensors for motion recognition and training. We use both wearable wireless sensors (such as electrocardiogram, motion sensors) and nonintrusive wireless sensors (such as gait sensors) to monitor the motion training status. We first provide our CPS architecture. Then we focus on motion training from three perspectives: 1) VR—first we introduce how we can use motion capture camera to trace the motions; 2) gait recognition—we have invented low-cost small wireless pyroelectric sensor, which can recognize different gaits through Bayesian pattern learning. It can automatically measure gait training effects; and 3) gesture recognition—to quickly tell what motions the subject is doing, we propose a low-cost, low-complexity motion recognition system with 3-axis accelerometers. We will provide hardware and software design. Our experimental results validate the efficiency and accuracy of our CPS design.**

***Index Terms*—Cyber-physical system (CPS), motion training, virtual reality (VR), wireless sensors.**

I. Introduction

The top three leading causes of death in many countries, especially in developing or developed countries, are heart disease, cancer, and stroke. Among those, stroke is one of the most typical reasons to cause permanent disability among adults. Fig. 1(a) shows conventional manual in-hospital post-stroke motion training method. It is labor intensive and expensive. A robot-aided training system [see Fig. 1(b)] could significantly reduce physical therapists (PTs)' involvement due to its intelligent patient motion assistance. However, it still requires the patient to visit the hospitals due to the high-cost, heavy-size robot system. A virtual reality (VR)-based training system [Fig. 1(c)] could be a simple computer video gaming system with some interactive units (such as digital glove). It has light weight and low cost.



(a) (b) (c)

Fig. 1. Illustration of motion training. (a) Manual training. (b) Robot-assisted training. Here we use KineAssist robot. (c) CPS-based training.

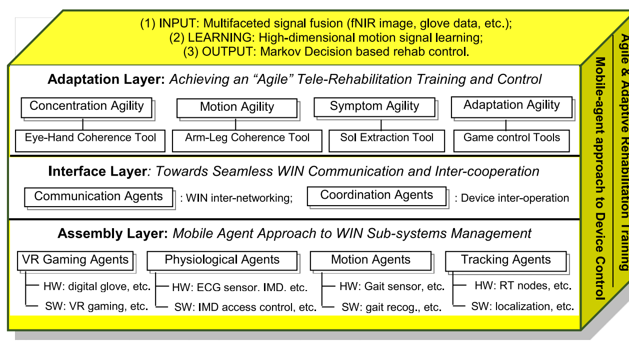


Fig. 2. Three-layer, multifaceted CPS architecture.

In this paper, we propose a cyberphysical system (CPS) with intelligent sensor data mining and motion analysis. It can be potentially used for next-generation rehabilitation due to its cyber controlled, automatic motion analysis. Such a CPS allows a patient to interact with a VR game. The motion sensors worn by the patient can automatically tell which gesture the patient is performing, and the worn medical sensors can tell the patient's health status.

Fig. 2 shows the big picture of the CPS. It consists of three layers.

1) The assembly layer uses the concept of mobile agents (MAs) [1] for sensing and VR devices control. We use four types of MAs: a) VR gaming; b) physiological monitoring; c) motion disorder detection; and d) patient tracking. Each MA is an independent hardware/software co-operation unit.

Manuscript received January 25, 2015; revised June 23, 2015; accepted April 3, 2016. Date of publication May 19, 2016; date of current version January 13, 2017. This work was supported in part by the National Science Foundation under Grant CNS 1335263, Grant IIS-0915862, and Grant CNS-1059212. *(Corresponding author: Qi Hao.)*

F. Hu, R. Ma, and Y. Patil are with the Department of Electrical and Computer Engineering, University of Alabama, Tuscaloosa, AL 35486 USA (e-mail: fei@eng.ua.edu).

Q. Hao is with the Department of Electrical Engineering, South University of Science and Technology of China, Shenzhen 518055, China (e-mail: hao.q@sustc.edu.cn).

Q. Sun is with the Department of Computer Science and Engineering, California State University at San Bernardino, San Bernardino, CA 92407 USA.

X. Cao is with the Department of Computer Science, Georgia State University, Atlanta, GA 30302 USA.

T. Zhang is with the Department of Computer Science, University of Houston–Downtown, Houston, TX 77002 USA (e-mail: mztbdf@gmail.com).

J. Lu is with the Department of Computer Engineering, University of Houston at Clear Lake, Houston, TX 77058 USA (e-mail: luj@uhcl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2016.2560127

2168-2216 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

TABLE I
COMPARING THREE MOTION TRAINING SCHEMES (USED IN OUR SYSTEM)

Motion recognition / training	How it works	Advantages	Shortcomings
(1) Kinect-based motion training (Section IV)	For body flexibility training; Use Kinect to capture depth image; can perform 3-D motion reconstruction	- Implement virtual reality: The reconstructed motion images may be used in a VR video; - The patient can clearly see his or her motions in videos; patient likes to interact with VR.	- Very coarse motion recognition; many body movement details cannot be captured or represented; - Need complex image processing algorithms
(2) Gait sensor based training (Section V)	For gait balancing training; Use surrounding wireless pyroelectric sensors to capture gait patterns	- Our invented RF pyroelectric sensors can discriminate gait changes; - Our Bayesian learning schemes can measure gait rehabilitation progress; - No complex image processing	- The gait sensors can only be deployed near the patient; thus they cannot capture far away gaits; - It can only tell the body gait balancing levels; but cannot tell limb motions
(3) Motion sensor based training (Section VI)	For gesture correctness training; wears accelerometers in limbs to capture motions	- The tiny, low-cost wireless motion sensors are directly attached to limbs; can accurately trace all limb motions; - No complex image processing	- Unlike Kinect, it cannot show 3D images; - It can only trace limb motions. Thus it cannot trace the entire body.

2) The interface layer aims to achieve a seamless interconnection of all devices for medical device interoperability and training information exchange. The operation interlocking among medical devices is important to patient safety. We use coordination agents to manage device interoperation. We also build communication agents to achieve device internetworking.
3) The adaptation layer uses open-source software tools for the analysis of the body flexibility and coherence level to achieve an agile and adaptive training.

This paper will present our CPS design, especially about the automatic gait/gesture recognition mechanisms. We will focus on motion recognition, which includes gait and gesture recognition. While gait recognition aims to extract the intrinsic walking/running patterns for body balancing analysis, gesture recognition aims to analyze the coherence level between all limbs and see if the patient can achieve certain standard limb motions. For gait recognition, we will explain how our system can automatically recognize different gait patterns during walking or running. Such patterns can tell the body balancing levels of a patient. For gesture recognition, we will use low-cost three-axis accelerometers to automatically recognize the patient's motions (such as raising right arm).

This paper has made three contributions as follows.

1) First, we propose a low-cost VR system with motion recognition. We use low-cost Microsoft Kinect (< \$200) as well as its software development kit to automatically recognize human motions. Such a system can help the patient to see clearly how well his or her body interacts with the VR scenes. This can significantly motivate the patient to finish each training phase via interesting VR games.
2) We have invented low-cost pyroelectric sensor, which uses wireless thermal sensing to capture different gait patterns. We have designed intelligent sensing data analysis algorithms which can extract the most important features from binary raw sensor data. We propose to use Bayesian non-negative matrix factorization (NMF) model to accurately recognize different gait patterns.
3) Third, we have designed an accurate limb motion (gesture) recognition system. By using multiple, low-cost, 3-axis, wireless accelerometers in the joints of limbs, we are able to automatically recognize dozens of limb gestures. Such a low-cost gesture recognition system is important to home-oriented motion training since it automatically tells the similarity between the patient's gesture and the standard ones (prestored in the gesture database).

Table I summarizes the pros and cons of the proposed three motion training approaches to be elaborated later. As we can see, they have their own application scenarios, and can complement with each other. For example, although Kinect-based 3-D reconstruction can help to visualize the body motion images, it involves complex image processing, and also does not have accurate gait/gesture capture.

The rest of this paper is organized as follows. In Section II, we will summarize related works. Section III then provides a systematic description of the entire CPS. The motion reconstruction is introduced in Section IV. We then move to the gait recognition in Section V. Here our new invented gait sensor and the corresponding machine learning algorithms will be discussed. The gesture recognition based on motion sensors will be explained in Section VI. Section VII has our experimental results. Section VIII concludes this paper.

II. RELATED WORK

Conventional rehabilitation systems are based on labor-intensive and therapist-assisted manual recovery [2]. Such high-cost, in-clinics treatment still dominates today's rehabilitation. Robot-aided rehabilitation platforms have been used in clinics to reduce therapy assistance [3]. An ankle robot was developed at MIT for neurorehabilitation of stoke patients [4]. VR-based platforms have been proposed recently. In [5], a system with a pneumatic glove and immersive VR environment was proposed for post-stroke rehabilitative training. However, very little work has designed an integrated low-cost sensing and VR system for automatic rehabilitation purpose. Our system will use low-cost wireless gait/motion sensors and low-complexity data processing algorithms to achieve motion training.

A. Gait Recognition and Training

There are dozens of neurodisorder related gait symptoms, which need distinct classifications between epileptic seizures and paroxysmal movement disorders [6]. The conventional gait assessment in clinical practice is typically performed manually through PTs' observations [7]. In many hospitals, the integration of videotaping and EEG sensing signal analysis (called video-EEG) [8], has been used for neurodisorder symptom diagnosis.

Most conventional approaches for automatic (labor-free) gait recognition are based on image sensors and digital image processing [9]. They typically involve intensive computation and also impose high hardware requirements such as real-time signal processing. Other wearable devices (such as gyroscopes, accelerometers, weight sensors, etc.) have also been used for gait measurement with low-fidelity motion recognition [10]. However, it is difficult for the conventional schemes to handle the high-dimensional, geometry-preserved gait signal pattern extraction. In this paper, we will use our new invented gait sensors as well as a Bayesian learning to analyze gait patterns from high-dimensional sensor data.

B. Gesture Recognition and Training

In gesture rehabilitation, our goal is to train the patient to achieve different limb motions often used in their lives. Since our CPS needs an automatic gesture recognition, it is important to use low-cost motion sensors as well as real-time pattern recognition software to detect different limb motions. By comparing the similarity level (such as using special distance definition) between the patient's gesture and the prestored standard ones, we can tell how well the patient can successfully achieve the desired gestures.

Some studies have recognized some simple gestures by using a single accelerometer [11]. uWave [12] provides an efficient recognition algorithm based on dynamic time warping (DTW) [13]. It is a user-dependent system only for personalized gesture recognition and thus limits its applications. In a practical application, the recognition of complex gestures by using multiple motion sensors is needed. In [14], a correlation method was proposed for gait recognition by capturing acceleration signals. Gafurov *et al.* [15] attached two bi-axial accelerometers to the participant's right leg to achieve recognition through histogram similarity and cycle length method. However, the dimension of feature set will be high when the number of sensors is large.

In this paper, we will solve the gesture recognition problem by efficiently processing high-dimensional, complex motion sensor streams. Multiple Wiimote sensors will be used to recognize complex gestures. A new gesture recognition scheme via NMF is used to reduce the complexity of gesture recognition. Our approach comprises two main phases: 1) a training phase and 2) a testing phase. During the training phase, the NMF algorithms are applied to create exemplars for training gestures. Then, in the testing phase, we project an unknown gesture trace onto a lower-dimensional subspace for effective gesture recognition.

This paper is a significant extension of our previous work in VR-based training [16]–[18], multiagent-based system management [19], [20], and gait recognition [21]–[25]. The dominant differences between this paper and our previous work include the following two aspects.

1) *New Gait Sensing System:* Compare to our previous work, this paper has made two new designs. First, regarding hardware design, we have designed a new sensor that is much more sensitive to human movements but with lower power consumption. Second, from software perspective, the previous system cannot maintain the temporal and spatial correlations between different gait sensors' signals. This new work uses NMF with smoothness and sparseness constraints to achieve such a goal.
2) *Complete CPS Interaction Scheme:* Those former works only provided the big picture and basic principle of the rehabilitation system without detailed description of hardware and software components. This paper will comprehensively describe the details of our CPS system, especially the sensor hardware as well as software design.

III. CPS Architecture

Our goal is to achieve the integration of VR and multifaceted [wearable, implantable, noninvasive (WIN)] sensing (Fig. 3 shows a logical architecture). The implantable medial devices and radio frequency identification (RFID) have been studied before and will be integrated into the system. The devices mainly include three types.

1) *Game Interaction Devices (Such as Digital Glove):* They directly reflect the patient's hand adaptation progress during VR game play.
2) *Medical Sensors:* They mainly include electrocardiography (ECG, to measure heart beat rhythm), electromyography (EMG, to measure muscle activities), accelerometers (to measure movement), SpO2 (to measure Oxygen saturation), and other wearable sensors.
3) *RFID Mini-Reader:* It can read the RFID tags attached to surrounding objects. This is helpful to vision impaired or elder patients who can hear the beep of RFID reader's speaker in dangerous situations (e.g., when wrong medicine bottle is grabbed). The noninvasive sensors include pyroelectric/photonic sensors for gait disorder detection.

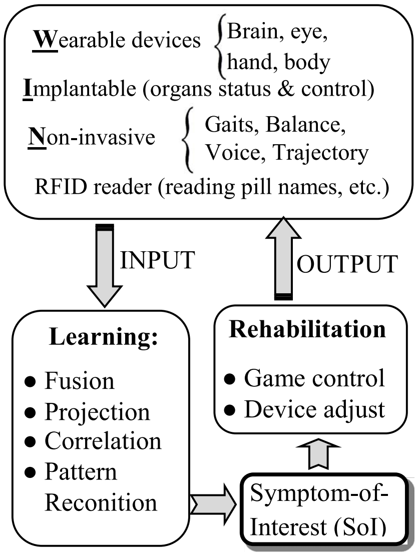


Fig. 3. CPS logical architecture.

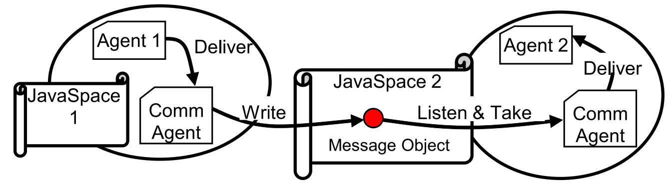


Fig. 4. Message exchange between MAs.

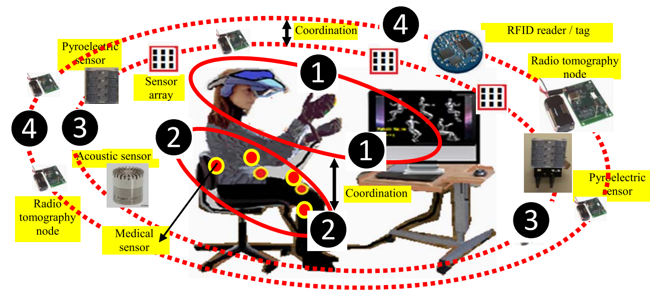


Fig. 5. CPS deployment (❶ VR agents, ❷ physiological agents, ❸ motion agents, and ❹ tracking agents).

A. Information Flow

The signals transferred among WIN devices include inputs (i.e., patient/device signal collections) and outputs (i.e., device/game control commands). As shown in Fig. 3, all device/patient status data are collected and sent to the server for symptom-of-interest (SoI) learning. Our system uses machine learning tools to achieve data fusion, signal projection (to visualize complex signals in low-dimensional subspace), signal correlation analysis (such as finding the patient's body coordination level), and pattern recognition (such as identifying heart attack events based on ECG processing). The learning results (for example, the patient shows satisfied cognitive capability for the current game level), can be used to achieve training adaptation, which includes two aspects.

1) *Game Control:* The game level should change based on the patient's training progress.
2) *Device Adjustment:* Some devices need to reconfigure themselves to adapt to the patient's status.

In the lowest layer of our platform (i.e., the assembly layer, see Fig. 2), we need to ensure that all hardware/software drivers work as what we expect. Moreover, devices of the same subsystem should seamlessly work together, and some software threads should be able to migrate from one device to another in the same subsystem. Here, thread migration means that we can use the same programming object with the function definition of action matching between the previous object and the new one. For example, the body sensors' signal pattern recognition thread needs to be migrated to the server's game control thread for game adjustment.

For the convenience of subsystem management and software thread migration, we propose to use an MA approach to manage all sensing and acting devices [1]. Without loss of generality, we define an agent as independent hardware/software co-operation unit (such as a digital glove with hand gesture recognition software). The message exchange and thread migration between MAs are implemented through message objects based on an agent interaction procedure of deliver-write-listen and take-deliver (Fig. 4).

Especially, in assembly layer we will manage four categories of MAs (Fig. 5 shows the device deployment architecture).

1) *VR Gaming Agents:* They control the basic functions of VR-based game interaction devices. They are responsible for the data collections including hand movement trajectory, brain activity patterns and game contents. They also call the communication and coordination agents (ComA) to analyze the coherence levels between hands and brain.
2) *Physiological Monitoring Agents:* They run in the body sensors (such as ECG, EMG, SpO2, etc.) to collect real-time physiological signals. Those agents could interact with VR gaming agents in order to control the game contents based on the patient's health status.
3) *Motion-Disorder Detection Agents:* They run in noninvasive sensors, especially in the gait recognition sensors, in order to monitor the patient's motion-disorder and body imbalance status. They provide direct indication of neurodisorder levels since motion disorder is strongly related to neurodisorder.
4) *Patient Tracking Agents:* They control the operations of RFID readers/tags. The RFID devices can detect the surrounding objects such as a medicine bottle.

B. VR Gaming Agents

They are used to control the operations of the following VR devices. As shown in Fig. 6(a), our platform uses the head mounted display (HMD) to provide the patient a 3-D virtual gaming interaction environment. The digital glove [27] [Fig. 6(b)] reflects the hand flexibility. VR MAs is designed with the following features. For the digital glove, the MA performs hand gesture recognition based on hidden Markov

(a) (b) (c)

Fig. 6. (a) HMD [26]. (b) Digital glove [27]. (c) Glove sensor model.

model (HMM). Such an HMM-based scheme can recognize more ranges of hand motions besides the 26-letter and other simple hand gestures provided by the manufacturer. Fig. 6(c) shows the digital glove model with accelerometers in the fingers' joints. Such a model will be used to build HMM state transition matrix.

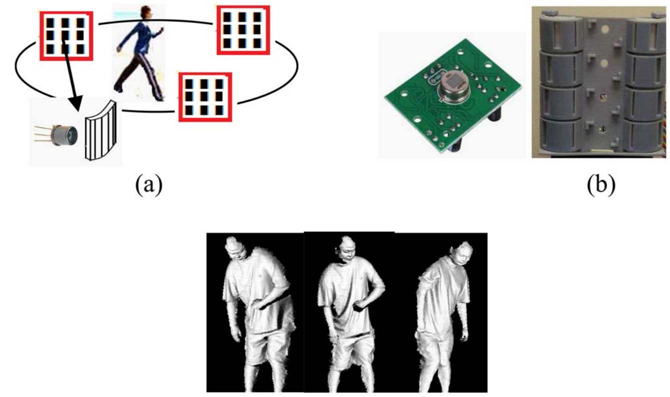
C. Physiological Monitoring Agents

These agents are used to collect the stroke patient's physiological data (such as heart beat rhythm) from wearable sensors.

1) The game exciting level can be adjusted based on the medical sensors' signals. For example, an action-intensive game should be stopped if the patient shows a heartbeat rate over 130.
2) Many rehabilitation tasks need to train the patient's body parts. The EMG sensors, combined with others (such as the accelerometers attached to the arms and legs), could be used to measure body motions.

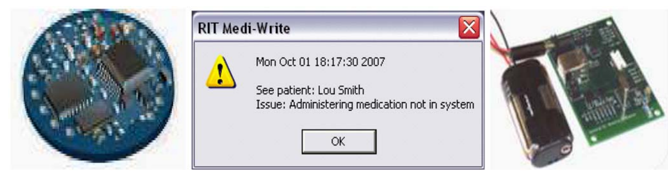
D. Motion-Disorder Detection Agents

This type of agents can control the functions of the gait sensors for motion disorder/body imbalance detection. The agents can also perform gait pattern recognition via machine learning algorithms. Because we target a home-oriented rehabilitation system, the gait sensors can provide important at-home patient behavior data for the doctor's reference. We have built a pyroelectric sensor network (PSN) [Fig. 7(a)] to recognize normal or abnormal gaits [Fig. 7(c)]. To generate rich visibility modes from thermal sources, we attach a Fresnel lens to each pyroelectric sensor [see Fig. 7(b)].



(a) (b) (c)

Fig. 7. (a) Gait sensor array. (b) Pyroelectric sensor and lens. (c) Disorder/imbalance.



(a) (b) (c)

Fig. 8. (a) RFID reader. (b) Medicine detection. (c) RF node (here RFID reader is Skyetek product; the RF node is developed by us).

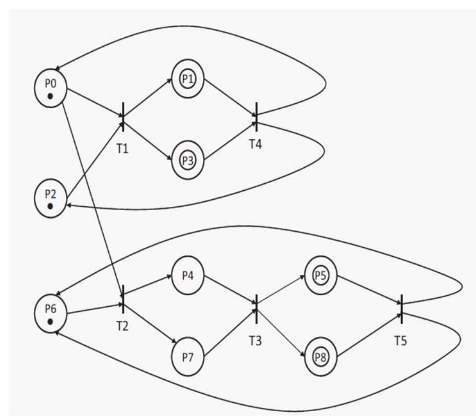


Fig. 9. PN model of agents.

E. Patient Tracking Agents

These agents control RFID readers/tags to keep track of the patient's indoor trajectory and to recognize the surrounding objects (such as a medicine bottle). We have used RFID mini-reader M1 [28] [Fig. 8(a)] to identify a medicine bottle [Fig. 8(b)]. The design of tracking agents are based on our customized wireless boards [Fig. 8(c)].

F. Communication/Coordination Agents

These agents are responsible for the internetworking of all devices. The VR devices could use wireless or wired interfaces to communicate with the server. A ComA aims to provide desired communication performance. The ComA also ensures seamless interface communications between RFID mini-readers and wireless motes. In order to resolve the possible conflicts in scheduling and resource allocation (such as wireless channel access), we propose to use Petri-net (PN) models to study the reachability and consistency issues during multiagent collaborations. As shown in Fig. 9, a PN consists of positions, transitions, and input and output functions. A PN is said to be safe for an initial state if all goal states are reachable. To check the security and reachability of a multiagent collaboration scheme, we define PN as a four-tuple combination P, T, IN, OUT, where $P = P_1, P_2, \ldots, P_n$ is a set of states, $T = T_1, T_2, \ldots, T_n$ is a set of transitions. IN is an input function that defines directed arcs from states to transitions, and OUT is an output function that defines directed arcs from transitions to states.

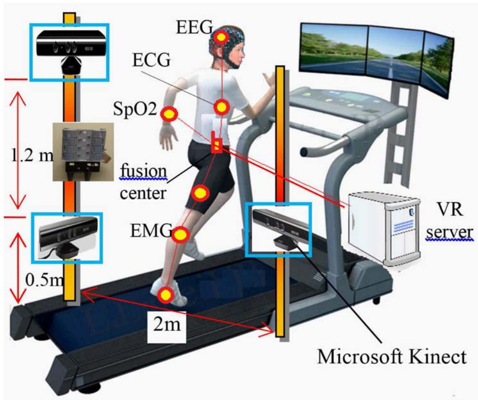


Fig. 10. Settings for VR.

Fig. 11. (Left) IR pattern from the IR emitter of the Kinect. (Right) Depth map generated by Kinect.

IV. Body Motion Reconstruction for Virtual Reality

A VR systems can give the patient a virtual training environment through video games. For example, cooking is a basic life activity in the patient's life. However, it is not convenient for a PT to go with the patient for cooking training. In our CPS, a patient can use digital glove to interact with a virtual scene. Our system adopts inexpensive depth cameras—Microsoft Kinects (< \$200), to capture 3-D human motions at video rate. We can avoid the interference between cameras through only three Kinects deployed as shown in Fig. 10: two Kinects capture the upper part and lower part of a human body, respectively, and the third Kinect is used to capture the middle body part. The Kinect IR emitter projects a pattern of a light throughout the room [see Fig. 11(left)]. The receiver then compares this data to a hard-coded image and generates a depth map [see Fig. 11(right)]. A skeleton is then applied to the body-like object in the depth map.

To utilize Kinect to automatically recognize the patient's body gesture and then generate the corresponding skeleton, an open source application called be the controller (BTC) was used to build the database of poses and gestures (Fig. 12).

V. Gait Rehabilitation

In many rehabilitation training tasks, we are interested in the capture of different abnormal gaits. By comparing them with a prestored gait pattern database (through a pattern distance model), we will get to know the body training progress. Although the above Kinect-based system can tell the basic body motions, it has a few shortcomings. First, the depth image processing involves complex image pixel analysis and huge memory overhead, which makes real-time motion recognition

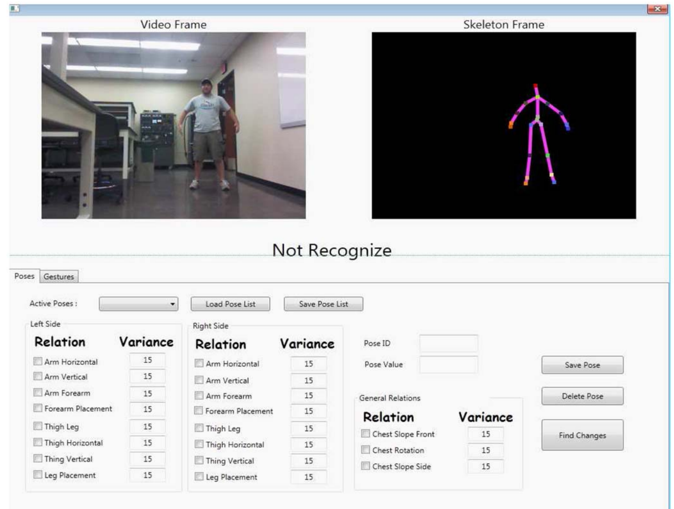
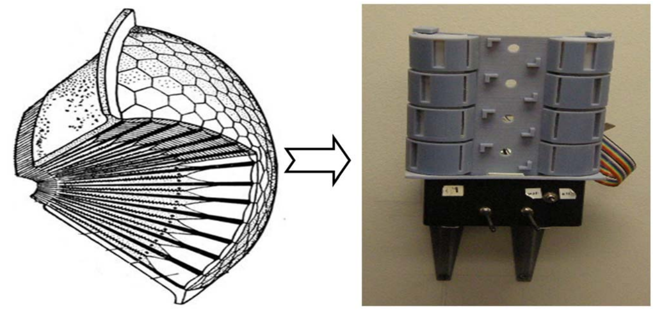


Fig. 12. BTC application used to create poses/gestures.

difficult to implement. Second, Kinect can only provides a coarse gait/gesture reconstruction. It cannot accurately capture the minor changes of body motions. Thus it cannot be used to analyze the quantitative rehabilitation progress levels for a gait training, which needs an accurate capture of body movement patterns during gait training. Our designed low-cost wireless pyroelectric sensors could detect the gait of an approaching walker through a highly sensitive thermal sensor that can discriminate among very minor thermal changes. Through the binary processing of the raw analog thermal signals, as well as using efficient machine learning algorithms, our system can recognize the normal gaits of different walkers.

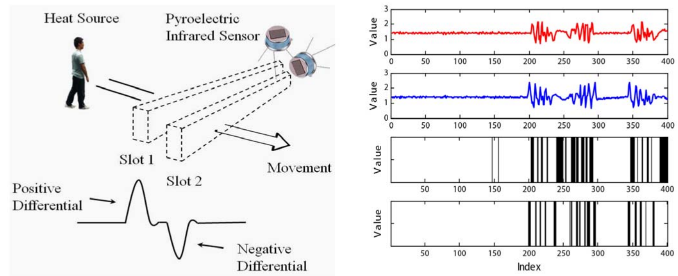
Our final goal is to extend our previous work to the recognition of different abnormal gaits (compared to healthy people's well-balanced gait patterns). Such an extension is not trivial due to two reasons.

1) *Higher Gait Recognition Accuracy Is Needed:* While it is relatively easier to detect different walkers' normal gaits due to people's obvious walking habit differences, it is difficult to distinguish among various abnormal gaits due to their minor differences. For example, the borderland between two types of motor disorders (epileptic seizures and paroxysmal movement disorders) is not obvious [6]. It needs high-fidelity gait recognition schemes to distinguish among the following motor disorders: episodic ataxia, stereotypies, drop attacks, paroxysmal kinesigenic dyskinesia, and other symptoms. In order to achieve higher resolution gait recognition, we have redesigned our pyroelectric sensors with a richer gait sensing mode. Inspired by the insect's compound eyes (Fig. 13), we have designed the Fresnal lens with interleaved, small thermal signal filters. Such a special lens architecture could segment the surrounding thermal detection space into many delicate regions, and thus improve the gait detection sensitivity by checking which regions have signals. We have invented low-cost, small-size, and wireless gait sensor.



(a) (b)

Fig. 13. (a) Insect compound eyes. (b) Fresnal lens.



(a) (b)

Fig. 15. (a) Principle of gait sensing via pyroelectric sensors. (b) Binary gait sensing signals.

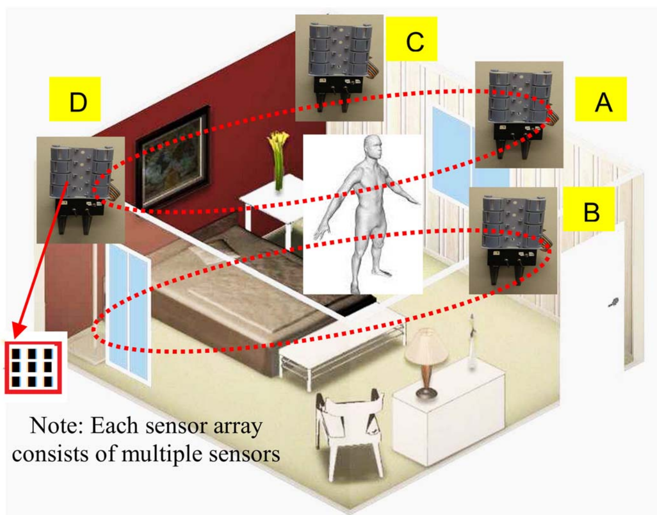


Fig. 14. Geometry-aware gait sensing.

2) *Temporal and Spatial Correlation Between Different Gait Sensors' Signals Needs to be Maintained During Gait Signal Processing:* In order to accurately record the gait features from different angles, we propose to use pyroelectric sensor array in each deployment position (Fig. 14). There exists data correlation between different array signals: one is intra-array geometry: in the same sensor array, their data have high similarity since they sense the same body part's gait. The other is interarray geometry: between different sensor arrays, A and B (Fig. 14) should have stronger spatial correlation than A and C since A and B capture gaits in the same vertical line. And A and C should have stronger spatial/temporal correlation than A and D due to two reasons. First, A and C are closer to the patient while D is far away. Second, A and C capture signals with closer temporal correlation when the patient walks from A to C. Later on we will explain how our proposed geometry-preserved NMF scheme could preserve the intrasensor and intersensor signal correlations.

A. Hardware—Wireless Gait Sensor

The basic principle of pyroelectric sensor-based gait detection is to utilize the thermal change detection when a human subject walks across a sensor [Fig. 15(a)]. We have invented Fresnel lens (Fig. 13) that use special filters to generate rich thermal detection patterns. Note that the layout of filters (see the vertical holes in Fig. 13) needs a careful design since human's thermal patterns are different from animals. We have compared over dozens of layouts and select some effective ones. Recently, we have invented a pyroelectric sensor which can automatically adjust its sensing power levels in order to capture a walker's thermal patterns. A field-programmable gate array-based control circuit can automatically reconfigure the gait sensor's sensitivity in order to detect gait patterns in different distances.

Binary signal (0,1) is the simplest data representation. Fig. 15(b) shows the principle of binary gait sensing: we collect the raw analog thermal signals, and then use space encoding [25] to obtain the binary representation. Such binary data from multiple gait sensors in the same room place (we call those sensors as a sensor array), form a matrix for gait pattern extraction.

B. Software—Geometric Bayesian Learning of Gait Patterns

Our gait recognition scheme aims to seek a pattern extraction solution for high-dimensional array signals. Such a solution can maintain sensor array geometry structure (AGS) for delicate gait discrimination. For example, in Fig. 14, since sensor arrays A and B record body motions in the same location (but deployed in different heights to capture arms and legs' motions, respectively), by maintaining their AGS we can describe the entire body's gait features. By maintaining A and C and A and D pairs' AGSs, we guarantee the repeatability and consistency of all extracted gait patterns since we can analyze the signals along the patient's walking path (in this example, it is A-C-D). In order to retain such intraarray/interarray AGS information, we extend our previous general NMF-based gait recognition model [24] to a geometry-preserved NMF (denoted as gNMF) based on graph embedding models [29]. The gNMF could maintain the geometric structure among neighboring arrays' signals even after we map the high-dimensional sensor array signals to a low-dimensional gNMF feature subspace.

Because NMF uses iterative W (basis matrix) and H (feature matrix) updates [30], it is time-consuming to analyze a large observation time window. Therefore, before gNMF analysis we first use the on-line signal segmentation to limit our gNMF

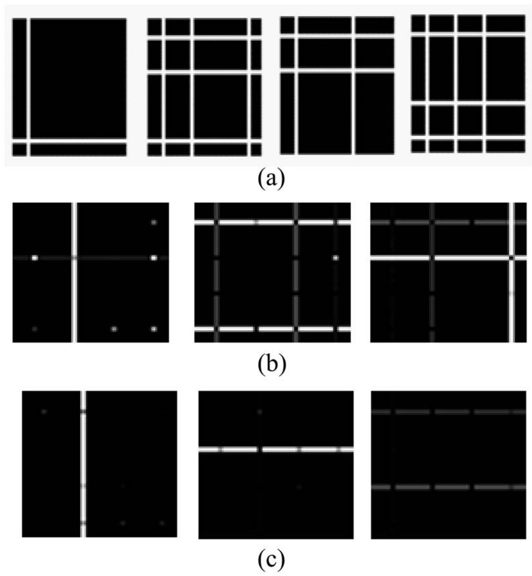


Fig. 16. (a) Fence data. (b) Feature basis obtained from general NMF. (c) Basis from gNMF.

analysis within a window with proper size. For a window of sensor array data, we can use a weighted graph $G = X, S$ to represent the geometric relation between all data points. Here $X = [x_1, x_2, \ldots, x_n]$ and n is window size (i.e., how many data points). $S = S_{ij}$ is the graph similarity matrix, which can be formed through a Gaussian kernel denoted as $\mathrm{Exp}(\|x_i - x_j\|^2/t)$, or using other kernel methods [31]. The diagonal matrix E of the graph is: $E = S_{ii}$, and Laplacian matrix $L = E - S$. For each original point x_i, we map it to gNMF low-dimensional subspace through $\tilde{x}_i = W^T x_i$. All mapped points form a data matrix: $\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n]$. In order to retain the AGS information of the original weighted graph, we could add a new constraint to the original NMF cost function as follows:

$$\mathrm{Cost}(X\|WH) = KL(X\|WH) + \zeta\left(\sum_{ij}\left\|\tilde{x}_i - \tilde{x}_j\right\|^2 S_{ij}\right). \quad (1)$$

Here $KL(\cdot\|\cdot)$ is *K-L* divergence. Obviously, the second item of the above function penalizes the graph distance of two data points that are far away from each other (thus with less geometric similarity).

Besides the reservation of AGS information between neighboring sensor array readings, another important advantage of gNMF is its capability of suppressing redundant dimensionality from high-dimensional signals. This is due to its matrix factorization nature and the use of kernel mapping. We have verified gNMF's signal suppression characteristics through the example of fence data analysis [Fig. 16(a)]. While general NMF generates basis vectors with detailed feature points [Fig. 16(b)], gNMF generates very sparse feature basis vectors [Fig. 16(c)].

The complete solution to gNMF (i.e., solving W and H through a gradient optimization) involves a stationarity-guaranteed conditional optimization problem. Conventional NMF uses simple multiplicative update through an auxiliary function. However, it cannot guarantee the algorithm to

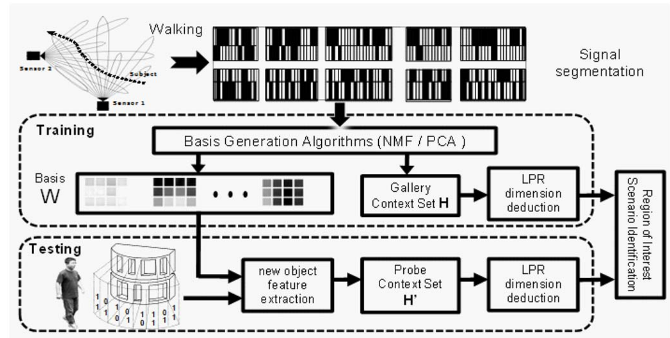


Fig. 17. Gait pattern detection scheme.

converge to a stationary point. We can form a conditional optimization problem in order to find such a stationary convergence point [32] (here we use the basis matrix W's update rule as an example)

$$W^{t+1} = \left[W^t - \beta^\gamma \cdot \nabla KL(X\|WH)^t\right]^+. \quad (2)$$

Here $[\cdot]^+$ means max $[\cdot, 0]$ (thus it keeps non-negative entries) and $KL(\cdot\|\cdot)$ is *K-L* divergence between two distributions. Here β can be set up to a constant based on some empirical analysis. The value of γ is the first non-negative integer that meets the following inequality:

$$KL(X\|WH)^{t+1} - KL(X\|WH)^t = c \cdot < \nabla KL(X\|WH)^t, W^t\beta^\gamma - W^t >. \quad (3)$$

Here c is a constant, and $<\cdot, \cdot>$ is the Frobenius inner product between two matrices.

Our gait training system uses a gait sensor cluster architecture (Fig. 14). Instead of evenly distributing sensors everywhere, we deploy sensors into "clusters." This can fully utilize the sensitive, wide-angle thermal detection capability of pyroelectric sensors and thus reduce repeated sensor measurements. Moreover, since a microcontroller has multiple analog-to-digital converter interfaces, by grouping multiple sensors in one cluster, we can use one RF communication board to send out multiple sensors' data, which reduces the hardware cost. Through careful control of each sensor's facing direction, we could well capture a 360 view of a neighborhood around a cluster.

Assume a cluster has N sensors. For such N-dimensional data stream, we will use the principle shown in Fig. 17 to identify a new gait context (called context extraction). As shown in Fig. 17, for such a N-dimensional data, first we need to segment it into different windows. The window size depends on how much data a sensor can handle in real-time. Here we use an 8×16 window size to form a binary matrix, called observation data X. Each value is either 1 (means "detected") or 0 (means "not detected").

The context extraction system includes two phases.

1) Training Phase: It is important to identify some common aspects to be compared among different scenarios. For instance, in traditional video systems, to identify human faces, we typically use eye size, nose length, distance between eyes, etc. to serve as comparison "bases." Likewise, we need to identify some gait bases in our system although each basis may not have physical interpretation of gaits that are as clear as human face does. Through NMF we could obtain a set of bases for all pyroelectric sensor data to be trained. We select the windows of binary data for almost all different contexts. For each of those contexts, we obtain the corresponding coefficients (called weights) for each basis. All contexts' bases and weights are stored in the context template database for testing purpose.

2) Testing Phase: When a new window of data comes, to extract the context information from this window, we project the data into the bases prestored into the context database and calculate the corresponding basis coefficients (weights). We then calculate the similarity level between the new calculated weights and the ones in the database. The closest match indicates a found "context." In Fig. 17, we use H to represent the context weights prestored in the database, and use H' to represent new tested context weights. In order to visualize the context features, we utilize the linear principal regression to project the multidimension vectors (H or H') to a 2-D space.

a) Improvement of NMF schemes through sparseness constraint: One of NMF advantages is its basis sparseness. For example, if we extract some common features (to form "basis") from humans' faces, such as eyes, nose, mouth, and so on, we could use different methods (NMF, principle component analysis (PCA), wavelet, etc.) to search for those bases. It was found that NMF gave us the sparsest bases [33]. The sparseness is important to the reduction of memory storage and calculation complexity. More importantly, it makes NMF more like "parts-based" feature extraction, i.e., we can easily recognize an object by looking at its few features.

The measurement of sparseness can be regarded as a mapping from $\Re^n$ to $\Re$ to quantify how much energy of a vector is packed into a few components. Without loss of generality, we adopt a sparseness definition used in [34] that considers an observation vector X with n elements (x_1, x_2, x_n)

$$\text{Sparseness}(X) = \frac{\sqrt{n} - \left(\sum |x_i|\right)/\sqrt{\sum x_i^2}}{\sqrt{n} - 1}. \tag{4}$$

For a PSN with K clusters and each cluster has N sensors, if we sense $N \times M$ data X, the goal of "sparse pyroelectric sensing" can be formulated into a matrix factorization procedure: we seek non-negative weight matrix W and context matrix H, such that the least square meets, that is, minimizing $\|X - WH\|^2$. In the meaning time, W and H should meet two sparseness constraints: 1) Sparseness $(w_i) = S_w$, for any ith row of W. Here S_w is the desired W sparseness (preset by user; range: [0, 1]) and 2) Sparseness $(h_i) = S_h$, for any ith column of H. Here S_h is the desired H sparseness (Range: [0, 1]).

b) Improvement of NMF schemes through smoothness constraint: Although sparseness can make NMF more like parts-based feature recognition, too sparse matrix representation could not accurately describe an object since most elements of the context matrix H will be zero (or very small values). Therefore in some applications, we could control the "richness" of context matrix H by adding smoothness constraints to H. Smoothness tries to reduce the big differences among elements to make values have "smooth" differences. Here we define a smoothing function S as follows [35]:

$$S = (1 - \theta)\mathbf{I} + \frac{\theta}{K}\mathbf{1}\mathbf{1}^T \tag{5}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a vector of ones $[111\cdots]$. The parameter θ is important. Its range is [0, 1]. Assume $H' = SH$, The larger θ is, the more smooth effect we can get. This can be seen from the following fact. If $\theta = 0$, $H' = H$ and no smoothing on H has occurred. However, when θ is approaching to 1, that is, $\theta \to 1$, H' tends to be a constant matrix with all elements almost equal to the average of the elements of H. Therefore, parameter θ determines the extent of smoothness.

VI. Gesture Rehabilitation

The above gait sensing system can discriminate the minor gait pattern changes. However, it is still a nonintrusive sensing system that cannot well capture gaits if deployed far away from the target. Moreover, the gait sensing aims to extract the walking/running related patterns without knowing exactly what gestures the patient is doing in the upper or lower limbs. Therefore, wearable motion sensors attached to limbs can be used to trace the exact gestures such as far reaching, flat arm, kicking, etc. By using some pattern distance functions (such as K-L divergence) between the patient's motion signals and the prestored standard gestures, we know how close to the standard gestures a patient can achieve. Especially in home-oriented rehabilitation, it is critical to have such a low-cost gesture recognition system for self-training purpose. Note that today the light-weight motion sensors can be seamlessly embedded into e-textile [36]. Thus a patient can just wear such "smart clothing" during rehabilitation.

Unlike traditional single-accelerometer schemes, we adopt multisensor system to recognize more complex gestures. By deploying multiple, tiny, wireless accelerometers in the joints of limbs, we can obtain a more accurate gesture understanding. In order to recognize gestures, we again need to compare a gesture trace to a prestored template. Such a gesture distance can be defined by multisensor DTW as follows (assume we have total k motion sensors):

$$\text{DTW}\left(T_i, T_j\right) = \sqrt{\sum\nolimits_{l=1}^{k} D_{n,m}^2(x_l) + D_{n,m}^2(y_l) + D_{n,m}^2(z_l)} \tag{6}$$

where $D_{n,m}(x), D_{n,m}(y)$, and $D_{n,m}(z)$ are the DTW costs between the gesture traces T_i of size $n \times 3$ and T_j of size $m \times 3$ in the x-, y-, and z-axis, respectively (we use 3-axis accelerometers). The general DTW is calculated as follows. Assume two time sequences, $p = [p_1, \ldots, p_n]$ and $q = [q_1, \ldots, q_m]$, with differential phases and lengths (n and m, respectively). We can then compute the matching cost DTW$[p, q]$ based on dynamic programming $D_{i,j}$

$$\text{DTW}[p, q] = D_{n,m} \tag{7}$$

here

$$D_{i,j} = d\left(p_i, q_j\right) + \min D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1} \tag{8}$$

where $d(p_i, q_j)$ is defined as: $d(p_i, q_j) = (p_i - q_j)^2$.

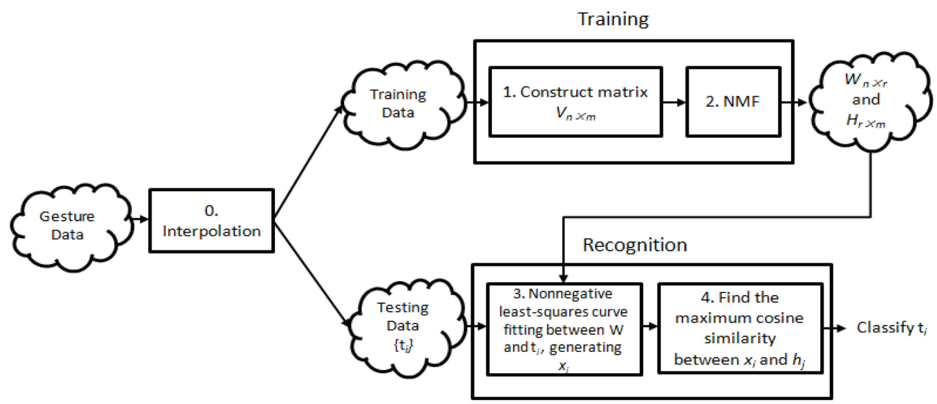


Fig. 18. NMF-based gesture recognition system.

We use an NMF-based gesture recognition method. The overview of this system is depicted as Fig. 18. It also incorporates a training stage and a testing stage.

In "step 0—interpolation," we uniformly interpolate all the shorter gesture traces to the same length as the longest trace. Thus all traces have the same size. In "step 1—matrix construction," conventional ways just simply align all sensors' data into a matrix $V_{n\times m}$, where n is the length of the trace and m is the number of traces

$$V_{n\times m} = \begin{bmatrix} x_{1,1} & y_{1,1} & z_{1,1} & \cdots & x_{r,1} & y_{r,1} & z_{r,1} \\ x_{1,2} & y_{1,2} & z_{1,2} & \cdots & x_{r,2} & y_{r,2} & z_{r,2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{1,n} & y_{1,n} & z_{1,n} & \cdots & x_{r,n} & y_{r,n} & z_{r,n} \end{bmatrix}_{n\times 3r}. \quad (9)$$

Here $x_{j,l}$, $y_{j,l}$, and $z_{j,l}$ are the lth $(1 \le l \le n)$ acceleration data point of the jth $(1 \le j \le k)$ accelerometer in x-, y-, and z-axis, respectively.

However, our experiments show that the above simple multisensor data alignment causes very low gesture recognition rate (less than 70%). Therefore, we propose a new way to construct the V matrix. The observation on the curves of different gesture traces inspires us to generate the idea of transforming gesture traces into images. As a matter of fact, when NMF algorithm was invented in the beginning [37], it was successfully used for face image decomposition and recognition.

Fig. 19 shows the basic principle of the trace-to-image transformation. Each trace image has $s \times n$ pixels, where s is the range of the acceleration value and n is the length of the trace. The value of any pixel is

$$\begin{cases} 0 & \text{if there is no curve in the pixel} \\ \sum_{i\in[1,3\times k]} 2^i \times 16 & \text{if there is curve in the pixel} \end{cases} \quad (10)$$

where i is the indexes of curves across this pixel and k is the number of accelerometers. The value of a pixel can be thought of the color index in the image of Fig. 19. The image is sparse, i.e., the value of most pixels is 0. The pixel value is not 0 when there is a curve across the pixel. Each curve has a unique value to fill in the pixels it crosses. Moreover, if there is more than one curve across the same pixel, the value of that pixel is the sum of the values of those curves. In this way, a gesture trace is transformed into an image.

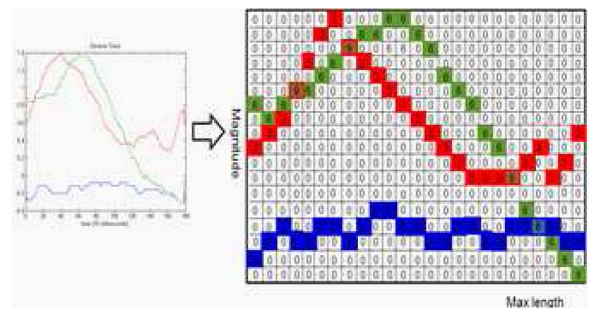


Fig. 19. V matrix on the curves of gesture traces.

In "step 2—NMF on V matrix," V is factorized into two smaller matrixes, W and H. W is the basis matrix and H is the coefficient matrix. Different varieties of NMF algorithms rely on the choice of cost functions or the regulation of W and/or H [37]. In this paper, two different implementations of NMF algorithm are employed. One is proposed in [38] called graph regularized NMF (gNMF). The other one is proposed in [39] as a block principal pivot NMF (BPNMF). The results of these two NMF algorithms will be compared later.

In "step 3—non-negative least squares curve fitting between W and t, for a test trace t, our target is to find the coefficient x that makes Wx closest to the t. This is non-negative least-squares curve fitting problem

$$\min_x \|Wx - t\|_x^2, \text{ where } x \ge 0. \quad (11)$$

In "step 4—find the closest coefficient column in H," we will find the coefficient column in the coefficient matrix H that is closest to the coefficient vector x calculated in step 3. We can find out the candidate column by calculating the cosine similarity between vector x and each column vector of H. The cosine similarity [40] between vector x and vector h_j is calculated as

$$\cos(\theta_j) = \frac{x' \cdot h_j}{\|x\| \times \|h_j\|}. \quad (12)$$

Two vectors are more similar to each other when their cosine similarity is closer to 1 than other vector pairs. Therefore, the corresponding gesture of trace j is the most probable gesture

TABLE II
GESTURE DICTIONARY I

Index	Gestures	Description
1	Forward reach	Start with the elbow at the side of the body, arm bent, and extend arm straight out slightly lower than shoulder height, return to starting position
2	Upward reach	Start with the elbow at the side of the body with the wrist near the shoulder and extend arm straight up, return to starting position.
3	Wiping motion	With the elbow near the side of the body and the arm bent, rotate forearm from roughly 45 inside the frame of the body to 45 pointing away from the body, return to starting position.
4	Sawing motion	Start with the elbow bent and near the side of the body, move arm forward & back roughly 6 inches.
5	Touching forehead	Start with the elbow near the side of the body, arm bent, palm up, then touch the palm to the forehead, return to starting position.
6	Lifting using forearm,(Dumbbell curl)	Keeping the elbow stationary on a table, lift the forearm from parallel to the table to a 90 angle, return to starting position.
7	Rotating elbow	With the arm bent and keeping the wrist at a stationary point, the elbow is rotated from straight down to 90 pointing away from the body, return to starting position
8	Lifting barbell	Start with the arm straight down near the side of the body, bend arm upward to the front of the breast, like lifting a barbell
9	Stretching to level forward	Start with the arm straight down near the side of the body, stretch the arm forward to level in front of body
10	Stretching to level side	Start with the arm straight down near the side of the body, stretch arm out to level at the side of body
11	Covering mouth	Start with the arm straight down near the side of the body, bend arm upward to 4-6 inches in front of the mouth
12	Stretching upward 1	Start with the arm straight down near the side of the body, stretch arm upward to the highest point while keeping arm in front of the body
13	Stretching upward 2	Start with the arm straight down near the side of the body, stretch arm upward to the highest point while keeping arm in the side of the body
14	Dumbbell flying	Start with the arm bent in front of the breast, move the arm levelly to the back of the body while keeping bent.

TABLE III
GESTURE DICTIONARY II

Index	Gestures	Description
1	Touching forehead	Start with the arm straight down near the side of the body, and then touch the palm to the forehead, return to starting position.
2	Pick up a cell phone	Start with the arm straight down near the side of the body, and then touch the palm to the right ear, return to starting position.
3	Waving whole arm	Start with the arm upward 45^o pointing away from the body, wave arm to 45^o inside the frame of the body but over head, return to starting position.
4	Waving forearm	Start with the forearm upward 45^o pointing away from the body, wave forearm to 45^o inside the frame of the body with elbow fixed, return to starting position.
5	Wiping motion	With the elbow near the side of the body and the arm bent, rotate forearm from roughly 45^o inside the frame of the body to 45^o pointing away from the body, return to starting position.
6	Waving arm levelly	Start with the arm reaching forward in level, wave arm 90^o levelly to the side of the body, return to starting position.

of test trace x if the cosine similarity between h_j and x is the largest.

A dictionary of 14 gestures is defined as shown in Table II. The first 7 gestures are from [41]. The other seven gestures are also typical motions. Another gesture dictionary containing six gestures is defined in Table III. Those gestures are more complex than the ones in dictionary I and multiple sensors are needed.

VII. EXPERIMENTAL RESULTS

A. Multiagent-Based CPS Management

We have implemented multiagent-based model in Java development environment (JADE). A group of devices can use JADE to implement sharing of data and control. Fig. 20 shows our agent control architecture. We have programmed the agent behaviors and built four communication components:

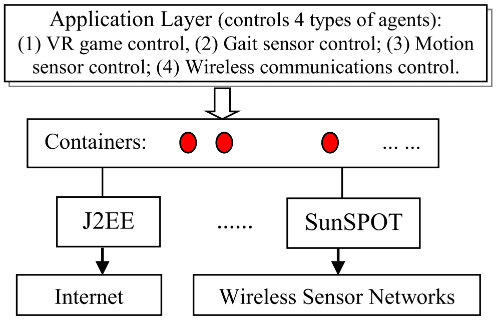


Fig. 20. Agent control model.

1) message format and content definitions; 2) searching agents for communication purpose; 3) the sending/receiving mechanism between agents; and 4) message filtering function.

We have also developed a belief, desire, and intention (BDI)-based agent behavior control scheme. The beliefs are derived from motion training status and results (such as gait/gesture training progress levels). We then formulate the possible desires based on the available beliefs. A new set of intentions and decisions can be generated based on previous BDIs. Generally, a simple agent behavior (such as sending out a sensing data sample) has low complexity intentions. However, a complicated agent behavior (such as multiagent interaction behavior) has high-complexity intentions.

For JADE programming convenience, we further classify the agents into four types.

1) *Sensing Agents:* They detect medical, gait, or motion signals.
2) *Decision Agents:* They are abstract software units for device control purpose. For example, the VR games can be controlled based on the training progress. A better body flexibility can indicate that it is the time to upgrade the game difficulty level.
3) *Action Agents:* They are devices that can be reprogrammed in order to change their actions. For example, a reprogrammable treadmill can change its running speed.
4) *Database Agents:* They can update the gait/gesture databases.

In our experiments, we have investigated three types of agent behaviors: 1) simple—such as generating a depth image; 2) high—such as NMF-based gesture recognition; and 3) medium—such as calculating 3-D body reconstruction. They generate three types of beliefs: 1) belief 1; 2) belief 2; and 3) belief 3. Fig. 21 shows the intention complexity result. Here p is the complexity coefficient, and n is the total number of states for an agent. As we can see, a more complex behavior generates a higher intention complexity. Here sensing and decision agents have higher intention complexity due to the complex sensor data mining for motion pattern extraction. Because belief 1 agent behavior has the low belief, it has the highest intention complexity. Belief 2 has the high belief and thus it has the lowest intention complexity.

B. Kinect-Based 3-D Motion Reconstruction

As discussed in Section III, we have used Microsoft Kinect [Fig. 22(a)] to perform low-cost, fast 3-D motion

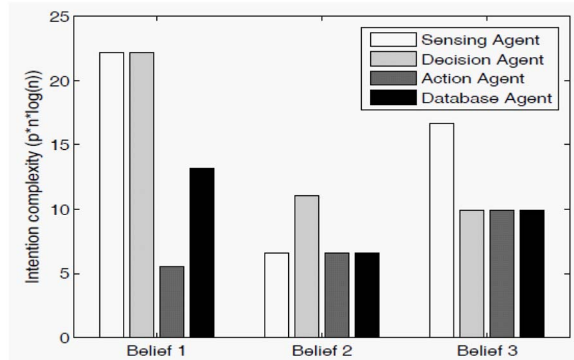
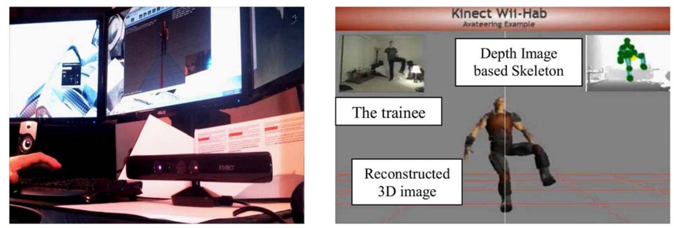
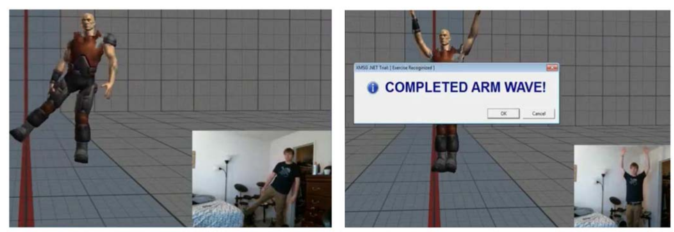


Fig. 21. BDI models for four types of agents.



(a) (b)

Fig. 22. (a) Kinect-based image capture system. (b) Three-dimensional reconstruction GUI.



(a) (b)

Fig. 23. (a) Three-dimensional reconstruction of "right leg rise." (b) Three-dimensional reconstruction of "arm wave."

reconstruction. Fig. 22(b) shows the graphical user interface (GUI) for motion recognition and reconstruction. The skeleton is generated from the depth image. It is then used for 3-D reconstruction.

Fig. 23 shows two motion reconstruction examples. Currently, we have implemented the recognition and reconstruction of 18 poses/gestures. More motions can be easily added to the database. As discussed before, it is used for coarse body motion recognition, and cannot accurately tell the gait/gesture changes. But it is good enough for general VR-based motion recognition applications.

C. Gait Rehabilitation

We have used our new invented pyroelectric sensors to measure the trainee's walking gait patterns. Fig. 24 shows our gait sensor data and gait classification results. As shown in Fig. 24(a), a certain gait shows noisy, however, periodical thermal sensing patterns. We have used a sensor array (with four sensors) to detect human gaits. Here we use binary format to represent the space encoding results: the dark parts means

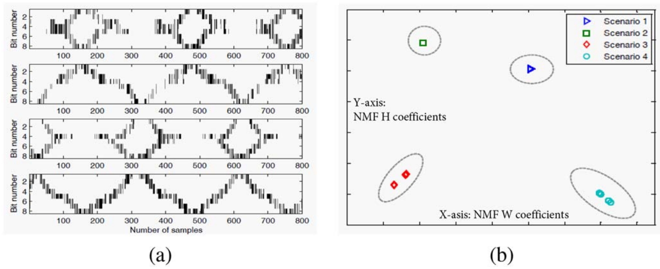


Fig. 24. (a) Gait sensor array's signals. (b) Gait classification/clustering results.

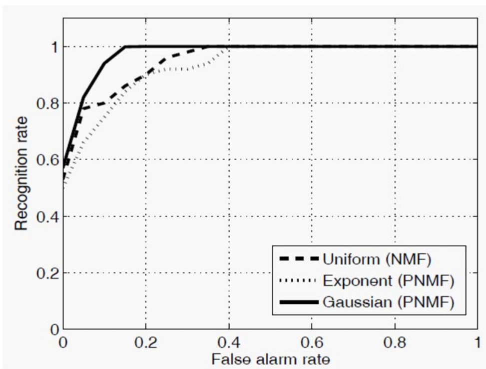


Fig. 25. Gait recognition rate with different NMFs.

binary "1," which means the sensor detects the walker in its detection angle. The blank parts mean binary "0."

We have further used NMF-based pattern recognition schemes to discriminate among four different gait scenarios. Those four scenarios have similar gaits if using human eye's observations. But our wireless gait sensors could successfully cluster those scenarios into different groups [see Fig. 24(b)]. This indicates that our gait training system can be used to distinguish among similar gaits. By comparing the NMF feature matrix to a template database, we get to know the gait training progress levels.

We have created different variants of NMF algorithms for gait recognition purpose: one is general NMF algorithm under the assumption of uniform distribution of gait sensor detection events in a room; the second one is probabilistic NMF (PNMF) with the exponential assumption on the gain detection events; the third one is PNMF with Gaussian assumption. The received operating characteristics (ROC) curves are shown in Fig. 25. As we can see, the Gaussian assumption with PNMF has the highest ROC. This could be explained as follows.

1) The thermal detection events during the patient's walking is a random event, and Gaussian distribution is the suitable assumption.
2) PNMF is a better scheme than NMF since the thermal sensing results have certain noise, and probabilistic model has better noise effect capture.

We then tested the similarity scores in NMF-based gait recognition schemes. The similarity score is defined as follows. During gait context understanding (testing phase) we can use K-means cluster and vector distance to compute the

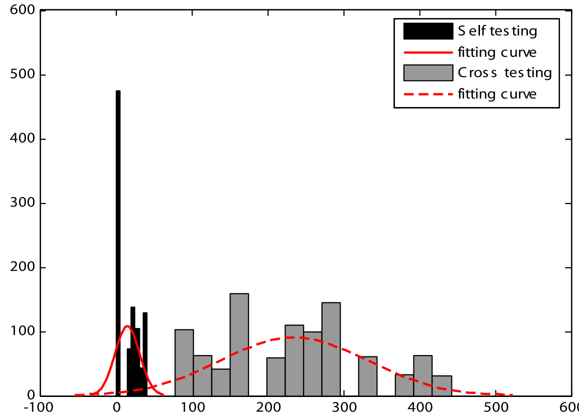


Fig. 26. Similarity score distributions.

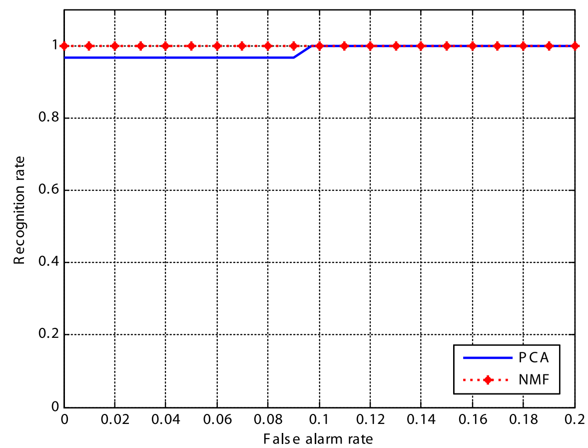


Fig. 27. ROC of PCA and NMF.

similarity score between H and H'

$$\arg\min \sum_{i=1}^{k} \sum_{h_j \in C_i} \left\| h_j - \mu_i \right\|^2. \tag{13}$$

Here h_j is a context feature vector and μ_i is the mean of cluster i. $C_1, C_2, \ldots, C_k$ are k clusters.

As shown in Fig. 26, the two distributions (self-testing and cross-testing) can be completely separated from each other. The self-testing scores occupy only a small region and cross-testing scores cross a larger region. These results show that NMF can detect context more accurately.

Then we compare the ROC graphs between a conventional pattern recognition scheme—PCA and NMF. As we can see in Fig. 27, NMF is always 1 no matter what value the false alarm rate is. PCA is lower than 1 in certain rates. This further illustrates the advantage of using NMF for gait pattern recognition.

Fig. 28 shows the NMF schemes under some constraints assumptions. It shows that NMF under smoothness constraints have the best performance. This is mainly because that scenario-dependent context prefers smoothness constraints. Total two kinds of hidden context patterns can be extracted via NMF algorithm. One is scenario-dependent context (i.e., gaits under different activities—running/walking); the other

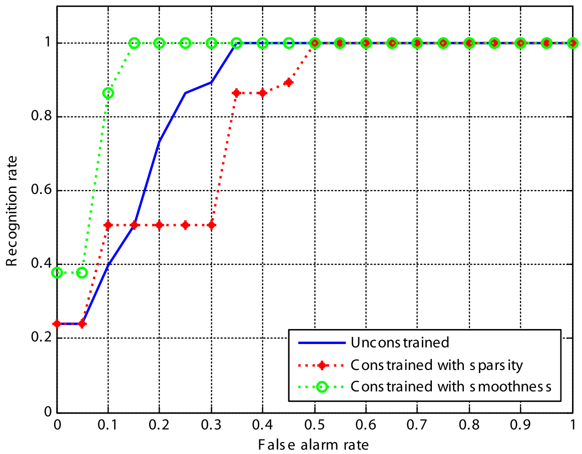


Fig. 28. ROC of NMF with constraints.

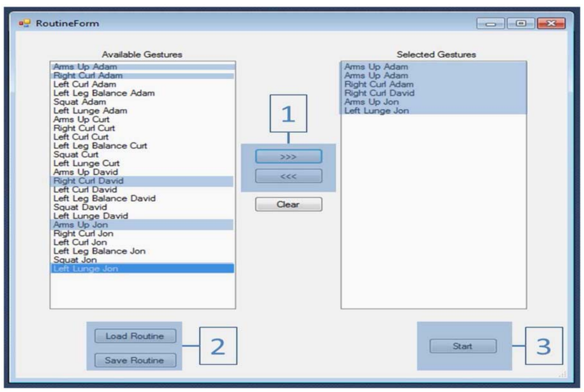


Fig. 29. GUI of gesture training software.

is path-dependent context (i.e., patient changes paths each time he/she walks through the sensors). The former (scenario-dependent) generates holistic context patterns (i.e., all NMF weights tend to be more evenly distributed), while the latter (path-dependent) generates local context characteristics (i.e., NMF weights tend to be distributed in two extremities). Our experiments have chosen the former (scenario-dependent) as the context identification objective. Therefore, adding smoothness constraint makes the NMF weights look more holistic (i.e., all weights become more evenly distributed), which makes context extraction more convenient.

D. Gesture Rehabilitation

We have used C# to build a GUI for gesture training. Fig. 29 shows the gesture type management. The patient can select any gesture for current rehabilitation task.

For the gesture dictionary I (Table II), two datasets have been generated. Dataset A is made by one subject with one Wiimote in hand performing 25 repetitions of each of the 14 gestures. Dataset B is made by one subject with two Wiimotes (one is in hand, the other is bound with the upper arm) performing 25 repetitions of each of the 14 gestures. For the gesture dictionary II (Table III), three datasets have been generated. The summary of datasets A to E is listed in Table IV.

TABLE IV
GESTURE DATASETS

Dataset	Dictionary ID	Subject numbers	Gesture numbers	Repetition numbers	Wiimote numbers
A	I	1	14	25	1
B	I	1	14	25	2
C	II	1	6	25	1
D	II	1	6	25	2
E	II	1	6	25	3

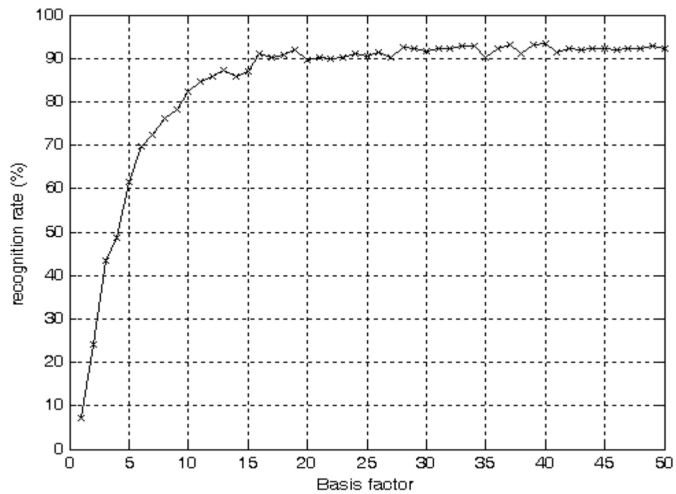
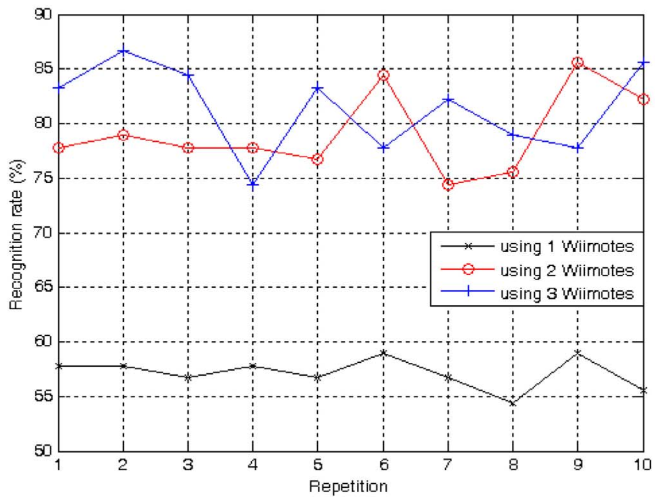


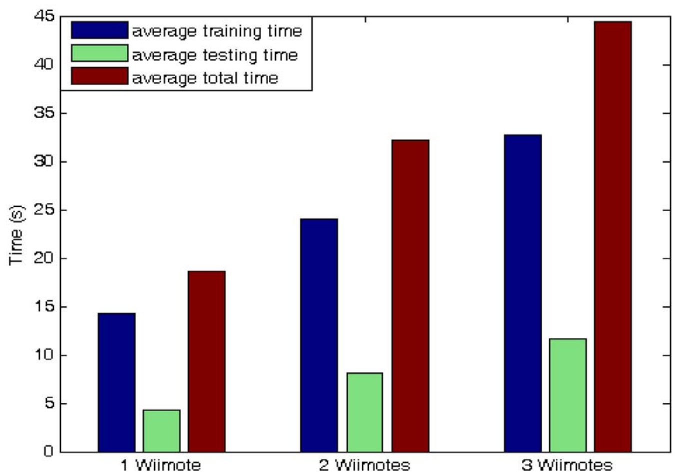
Fig. 30. Effect of basis factor selection.

An optimal NMF basis factor r (i.e., how many NMF basis we should set up) is theoretically undecided and depends on the input data statistics. Therefore, we have tried some possible values to find a proper choice of r. Fig. 30 shows the recognition rate of different basis factors from 1 to 50. The experiment is carried on the dataset A with binary NMF algorithm. From the result, we can see that the recognition rate keeps in a relative high level after the basis factor is larger than 15, but the trend is not stable. Therefore, we select 25 as the universal basis factor for all NMF algorithms.

Fig. 31(a) shows the recognition rate comparison using different number of Wiimotes. Fig. 31(b) shows the average time elapse accordingly. Datasets C, D, and E are based on the gesture dictionary II, which is specifically designed to show the different effects when using different number of Wiimotes. Among the six gestures, three pairs of gestures (1 and 2, 3 and 4, 5 and 6) are supposed to be difficult to recognize if using a single Wiimote. Therefore, the recognition rate using a single Wiimote should be low. The result confirms our expectation. The average recognition rate using a single Wiimote is between 45% and 58%. The recognition rate using two Wiimotes is much higher than that using a single Wiimote and reaches nearly 80%. The second Wiimote provides more information to make gesture recognition perform much better. After using three Wiimotes, the average recognition rate is over 81%. On the other hand, the time elapse using two Wiimotes is a little longer than the case of using



(a)



(b)

Fig. 31. (a) Recognition rate. (b) Time elapse.

a single Wiimote, but much less than the case of using three Wiimotes.

VIII. Conclusion

This paper described our CPS design for body motion training. Especially we proposed three effective ways for motion recognition and training.

1) In order to perform a coarse 3-D image reconstruction for the patient's motion, we proposed to use Kinect hardware to reconstruct the body shape. Such a shape can be embedded into a VR game in order to implement interactive virtual scene experience.
2) In order to accurately capture gait changes, we used our new invented pyroelectric sensors and a set of Bayesian learning algorithms to analyze the patient's gait patterns.
3) We designed a low-cost, accelerometer-based motion tracking system to automatically tell what limb gestures the patient is performing.

One of the novelties is that we have proposed the multiagent-based integration of wireless sensing and VR devices. The other novelty is that we proposed a geometry-preserved NMF algorithm to recognize the interarray sensing patterns. The third novelty is that we have designed a motion recognition model for more accurate limb motion capture.

Future works include some important aspects. We are investigating the neuroimaging to locomotor mapping model, which can find the quantitative relationship between EEG signals and motion patterns. This can give us a deep understanding of brain structure change during rehabilitation. The other field we are conducting now is multisensor signal fusion for high-dimensional data mining. It aims to use the complementary characteristics between different sensors (such as gait sensors, pressure sensors, medical sensors, etc.) to generate a more comprehensive picture on the rehabilitation training progress.

Acknowledgment

The authors would like to thank A. Akl from the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, for his help with the MATLAB codes on accelerometer-based motion recognition. They would also like to thank the Kinect/Wiimote programming and testing from the following undergraduate students (they were in Dr. F. Hu's Capstone Design classes), T. Anderson, D. Blalock, A. Brown, J. Foster, T. Grier, W. Hudgens, C. Buckallew, J. Won, S. Duncan, M. Smith, and B. Onyeukwu. All ideas presented here do not necessarily represent National Science Foundation's opinions.

References

[1] P. Braun and W. Rossak, *Mobile Agents: Basic Concepts, Mobility Models, and the Tracy Toolkit*. San Francisco, CA, USA: Morgan Kaufmann, 2004.
[2] G. P. Prigatano, *Principles of Neuropsychological Rehabilitation*, 1st ed. New York, NY, USA: Oxford Univ. Press, Jul. 1999.
[3] H. I. Krebs, N. Hogan, M. L. Aisen, and B. T. Volpe, "Robot-aided neurorehabilitation," *IEEE Trans. Rehabil. Eng.*, vol. 6, no. 1, pp. 75–87, Mar. 1998.
[4] A. Roy *et al.*, "Robot-aided neurorehabilitation: A novel robot for ankle rehabilitation," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 569–582, Jun. 2009.
[5] L. Connelly *et al.*, "A pneumatic glove and immersive virtual reality environment for hand rehabilitative training after stroke," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 5, pp. 551–559, Oct. 2010.
[6] C. W. Huang and W. J. Hwang, "The borderland between epilepsy and movement disorders," *Acta Neurol Taiwan*, vol. 18, no. 1, pp. 42–55, Mar. 2009.
[7] R. B. Davis, "Reflections on clinical gait analysis," *J. Electromyograph. Kinesiol.*, vol. 7, no. 4, pp. 251–257, 1997.
[8] A. B. Rose *et al.*, "Occurrence of seizure clusters and status epilepticus during inpatient video-EEG monitoring," *Neurology*, vol. 60, no. 6, pp. 975–978, 2003.
[9] N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis, "Gait recognition: A challenging signal processing technology for biometric identification," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 78–90, Nov. 2005.
[10] J. Ghika *et al.*, "Portable system for quantifying motor abnormalities in Parkinson's disease," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 3, pp. 276–283, Mar. 1993.
[11] H. Sawada and S. Hashimoto, "Gesture recognition using an acceleration sensor and its application to musical performance control," *Electron. Commun. Japan*, vol. 80, no. 5, pp. 9–17, 1997.

[12] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive Mobile Comput.*, vol. 5, no. 6, pp. 657–675, 2009.

[13] C. S. Myers and L. R. Rabiner, "A comparative study of several dynamic time-warping algorithms for connected-word recognition," *Bell Syst. Tech. J.*, vol. 60, no. 7, pp. 1389–1409, Sep. 1981.

[14] H. J. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, and S.-M. Makela, "Identifying people from gait pattern with accelerometers," in *Proc. SPIE Biometr. Technol. Human Identification II*, vol. 5779. Orlando, FL, USA, 2005, pp. 7–14.

[15] D. Gafurov, K. Helkala, and T. Soendrol, "Gait recognition using acceleration from MEMS," in *Proc. 1st Int. Conf. Availability Reliab. Security*, Vienna, Austria, Apr. 2006, pp. 432–439.

[16] F. Hu *et al.*, "Tele-rehabilitation computing: From a cyber-physical perspective," in *Tele-Healthcare Computing and Engineering: Principles and Design*. Boca Raton, FL, USA: CRC Press, Apr. 2013.

[17] F. Hu, "Functional near-infrared (fNIR) spectroscopy for auto-rehabilitation cyber-physical systems: Towards intelligent signal processing for training progress indication," in *Cyber-Physical Systems: Integrated Computing and Engineering Design*. Boca Raton, FL, USA: CRC Press, Sep. 2013.

[18] T. Zhang, J. Lu, R. Ma, K. M. Koushik, and X. Li, "Low-cost, home-oriented neuro-patient monitoring: Towards an information geometry based," in *Virtual Reality Enhanced Robotic Systems for Disability Rehabilitation*. Hershey, PA, USA: IGI Glob., Jan. 2016.

[19] Q. Sun, W. Yu, N. Kochurov, Q. Hao, and F. Hu, "A multi-agent-based intelligent sensor and actuator network design for smart house and home automation," *J. Sensor Actuator Netw.*, vol. 2, no. 3, pp. 557–588, 2013.

[20] J. Lu, J. Gong, Q. Hao, and F. Hu, "Multi-agent based wireless pyroelectric infrared sensor networks for multi-human tracking and self-calibration," in *Proc. IEEE Conf. Sensors*, Baltimore, MD, USA, Nov. 2013, pp. 1–4.

[21] Q. Hao, F. Hu, and J. Lu, "Distributed multiple human tracking with wireless binary pyroelectric infrared (PIR) sensor networks," in *Proc. IEEE Conf. Sensors*, Kailua, HI, USA, 2010, pp. 946–950.

[22] Q. Hao, F. Hu, and Y. Xiao, "Multiple human tracking and identification with wireless distributed pyroelectric sensor systems," *IEEE Syst. J.*, vol. 3, no. 4, pp. 428–439, Dec. 2009.

[23] J. Lu, T. Zhang, F. Hu, and Q. Hao, "Preprocessing design in pyroelectric infrared sensor-based human-tracking system: On sensor selection and calibration," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2016.2523914.

[24] F. Hu, Q. Sun, and Q. Hao, "Mobile targets region-of-interest via distributed pyroelectric sensor network: Towards a robust, real-time context reasoning," in *Proc. IEEE Conf. Sensors*, Kailua, HI, USA, 2010, pp. 1832–1836.

[25] J. Lu, J. Gong, Q. Hao, and F. Hu, "Space encoding based compressive multiple human tracking with distributed binary pyroelectric infrared sensor networks," in *Proc. IEEE Conf. Multisensor Fusion Integr. Intell. Syst.*, Hamburg, Germany, Sep. 2012, pp. 180–185.

[26] (Jan. 2016). *HDT (eMagin Z800 3DVisor)*. [Online]. Available: http://www.emagin.com/

[27] (Jan. 2016). *Re-Programmable Open Source Data Glove*. [Online]. Available: https://www.technologyreview.com/s/414021/open-source-data-glove/

[28] (Jan. 2016). *RFID Mini-Reader From Skyetek*. [Online]. Available: http://www.skyetek.com/products/rfid/skyemodule-m1-mini

[29] X. Liu, S. Yan, and H. Jin, "Projective nonnegative graph embedding," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1126–1137, May 2010.

[30] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, Denver, CO, USA, 2001, pp. 556–562.

[31] S. V. N. Vishwanathan, N. N. Schraudolph, I. R. Kondor, and K. M. Borgwardt, "Graph kernels," *J. Mach. Learn. Res.*, vol. 11, pp. 1201–1242, Apr. 2010.

[32] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, 2007.

[33] R. Schachtner, G. Pöppel, and E. W. Lang, "Binary nonnegative matrix factorization applied to semi-conductor wafer test sets," in *Proc. 8th Int. Conf. Independent Component Anal. Signal Separation*, Paraty, Brazil, Mar. 2009, pp. 710–717.

[34] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Dec. 2004.

[35] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsNMF)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 403–415, Mar. 2006.

[36] M. T. Jones, T. L. Martin, and B. Sawyer, "An architecture for electronic textiles," in *Proc. ICST 3rd Int. Conf. Body Area Netw. (BodyNets)*, Tempe, AZ, USA, 2008, Art. no. 24.

[37] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[38] D. Cai, X. F. He, J. W. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

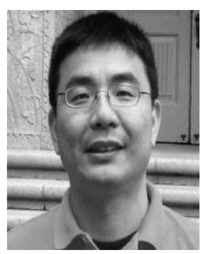
[39] J. G. Kim and H. S. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, Pisa, Italy, 2008, pp. 353–362.

[40] S. Tata and J. M. Patel, "Estimating the selectivity of *tf-idf* based cosine similarity predicates," *SIGMOD Rec.*, vol. 36, no. 2, pp. 7–12, Jun. 2007.

[41] A. L. Hall, "Method for the acquisition of arm movement data using accelerometers," S.B. thesis, Dept. Mech. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2005. [Online]. Available: http://hdl.handle.net/1721.1/32893

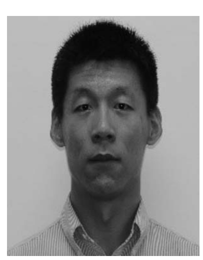
Fei Hu received the Ph.D. degree in signal processing from Tongji University, Shanghai, China, in 1999, and the Ph.D. degree in electrical and computer engineering from Clarkson University, Potsdam, NY, USA, in 2002.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of Alabama, Tuscaloosa, AL, USA. He has published over 200 journal/conference papers, books, and book chapters. His research has been supported by the U.S. National Science Foundation, the U.S. Department of Defense, Cisco, Sprint, and other sources. His current research interests include 3S—security, signals, sensors.



Qi Hao received the B.E. and M.E. degrees from Shanghai Jiao Tong University, Shanghai, China, in 1994 and 1997, respectively, and the Ph.D. degree from Duke University, Durham, NC, USA, in 2006, all in electrical engineering.

He is currently an Associate Professor with the South University of Science and Technology of China, Shenzhen, China. His postdoctoral training in the Center for Visualization and Virtual Environment, University of Kentucky, Lexington, KY, USA, was focused on 3-D computer vision for human tracking and identification. From 2007 to 2014, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Alabama, Tuscaloosa, AL, USA. His research has been supported by the U.S. National Science Foundation and other sources. His current research interests include smart sensors, intelligent wireless sensor networks, and distributed information processing.



Qingquan Sun received the master's degree from the Chinese Academy of Sciences, Beijing, China, in 2009, and the Ph.D. degree from the University of Alabama, Tuscaloosa, AL, USA, in 2013.

He is currently an Assistant Professor with the School of Computer Science and Compute Engineering, California State University at San Bernardino, San Bernardino, CA, USA. His research interests include machine learning and sensor networks.

Xiaojun Cao received the B.S. degree from Tsinghua University, Beijing, China, and the M.S. degree from the Chinese Academy of Sciences, Beijing, and the Ph.D. degree in computer science and engineering from the State University of New York at Buffalo, Buffalo, NY, USA, in 2004.

He is currently an Associate Professor with the Department of Computer Science, Georgia State University, Atlanta, GA, USA. His research has been supported by the U.S. National Science Foundation, IBM, and Cisco's University Research Program.

Dr. Cao was a recipient of NSF CAREER Award from 2006 to 2011.

Rui Ma received the B.S. degree in physics from Beijing Normal University, Beijing, China, in 2008, and the M.S. degree in microelectronics from the Institute of Semiconductors, Chinese Academy of Sciences, Beijing, in 2011. He is currently pursuing the Ph.D. degree in electrical engineering with the University of Alabama, Tuscaloosa, AL, USA.

His research interests include wireless sensor network, intelligent sensing system, machine learning for human activity recognition, and time series analysis.

Ting Zhang received the B.S. degree in computer science from Tianjing University, Tianjing, China, in 2008, the M.S. degree in computer science from the University of Texas at Brownsville, Brownsville, TX, USA, in 2011, and the Ph.D. degree from the University of Alabama, Tuscaloosa, AL, USA, in 2016.

She is currently an Assistant Professor with the Department of Computer Science, University of Houston–Downtown, Houston, TX, USA. Her research interests include data mining and classification algorithms, computer vision and computer graphics, tele-healthcare, and mobile computing.

Yogendra Patil received the B.S. (Engineering) degree in electronics and telecommunications, and the master's degree in electrical engineering from Wright State University, Dayton, OH, USA, in 2007. He is currently pursuing the Ph.D. degree from the University of Alabama, Tuscaloosa, AL, USA.

He is also a Research Assistant with the Pattern Recognition and Machine Learning, University of Alabama.

Jiang Lu (M'15) received the B.Sc. degree from Shanghai Maritime University, Shanghai, China, in 2007, the M.Sc. degree from the University of Florida, Gainesville, FL, USA, in 2007, and the Ph.D. degree from the University of Alabama, Tuscaloosa, AL, USA, in 2015, all in electrical and computer engineering.

He is currently an Assistant Professor with the Department of Computer Engineering, University of Houston–Clear Lake, Houston, TX, USA. His research interests include intelligent sensor systems, biomedical devices, tele-healthcare, wireless communication, distributed information processing, machine learning, and embedded system designs.