A Collaborative Computing Framework of Cloud Network and WBSN Applied to Fall Detection and 3-D Motion Reconstruction

Chin-Feng Lai, *Member, IEEE*, Min Chen, *Senior Member, IEEE*, Jeng-Shyang Pan, Chan-Hyun Youn, *Member, IEEE*, and Han-Chieh Chao, *Senior Member, IEEE*

***Abstract*—As cloud computing and wireless body sensor network technologies become gradually developed, ubiquitous healthcare services prevent accidents instantly and effectively, as well as provides relevant information to reduce related processing time and cost. This study proposes a co-processing intermediary framework integrated cloud and wireless body sensor networks, which is mainly applied to fall detection and 3-D motion reconstruction. In this study, the main focuses includes distributed computing and resource allocation of processing sensing data over the computing architecture, network conditions and performance evaluation. Through this framework, the transmissions and computing time of sensing data are reduced to enhance overall performance for the services of fall events detection and 3-D motion reconstruction.**

***Index Terms*—3-D motion reconstruction, fall detection, Cloud computing, wireless body sensor network (WSBN).**

I. Introduction

FALLING detection is a mechanism that uses sensors to sense the actions and behaviors of the human body for judging whether there is an accidental fall. It is mainly used for the elderly or children that are likely to have accidental falls by providing a warning or protection mechanism to reduce the injuries caused by accidental falls. While the 3-D motion reconstruction is proposed to simulate the motions of the human body, it is able to intuitively provide the falling postures and cautionary messages for medical care personnel [1]. However, the action signals from various body parts are converted to reconstruct 3-D human body motions, whose computational complexity is high for mobile devices. In other words, the operations of the reconstruction service has high consumption of the mathematical capabilities and electric energies of mobile devices [2], [3].

As cloud computing is integrated with the concept of virtualization, users can define the operation diagram of work according to their needs. This operation diagram is usually used for the operation of massive data. Based on the framework, the management operations can automatically conduct of the massive data of wireless body sensor network (WBSN) decentralized and parallelized. Cloud computing effectively solve the difficulties of fall detection and 3-D motion reconstruction for arriving ubiquitous healthcare. However, there remain some challenges in the combination of the related technologies. First, previous researches found that when mobile devices upload sensed data, it is necessary to periodically collect sensor node information, then upload data to cloud systems [4], [5]. The frequent information collections and data transmissions will greatly increase the power consumption of mobile devices, and it also raises the heavy network bandwidth loading. In terms of mobile devices, hardware specifications and related techniques are more and more advanced gradually. Therefore, how to divide works distributively for collaborative computing between cloud network and WBSN according to the mobile computing capabilities and the sensing devices, will be an interesting and challenging topic.

Therefore, this study aims to develop a collaborative computing framework of cloud network and WBSN for ubiquitous fall detection and 3-D motion reconstruction shown as Fig. 1. When the framework identifies a fall, the information two seconds before and five seconds after the occurrence of the fall is retained, and this information is transferred to the cloud environment via wireless network. The personal health care and fall reconstruction method is constructed by means of the high-speed computing power and storage space of the cloud. According to this information, and using the proposed 3-D motion reconstruction method, including body constitution and skeleton construction, the course of human action when an accident occurs is simulated according to the recorded information. This framework can provide healthcare services with correct injury positions and situations for subsequent medical treatment.

Manuscript received June 30, 2013; revised October 20, 2013; accepted November 21, 2013. Date of publication January 9, 2014; date of current version March 3, 2014. This work was supported in part by the National Science Council and Science Park Administration of the Republic of China, Taiwan under Contract NSC 101-2628-E-194-003-MY3, 101-2221-E-197-008-MY3, and 102-2219-E-194-002, in part by the International Science and Technology Collaboration Program (2014DFT10070) funded by China Ministry of Science and Technology (MOST), the Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Plannig (2010-0020732), and in part by the ICT R&D Program 2013 in MSIP.

C.-F. Lai is with the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan (e-mail: cinfon@ieee.org).

M. Chen is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China (e-mail: minchen@ieee.org).

J.-S. Pan is with the Department of Computer Science, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China (e-mail: jengshyangpan@gmail.com).

C.-H. Youn is with the Department of Electrical Engineering, GRID Middleware Research Center, KAIST, Daejon, Korea (e-mail: chyoun@kaist.ac.kr).

H.-C. Chao is with the Department of Computer Science and Information Engineering, National ILan University, I-Lan, Taiwan (e-mail: hcc@niu.edu.tw).

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/JBHI.2014.2298467

2168-2194 © 2014 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.

The research contributions of this paper can be summarized as follows:

1) Collaborative framework of cloud network and WBSN;
2) Workload prediction and job scheduler over WBSN;
3) Cloud computing applied fall detection and motion reconstruction.

The remainder of this paper is organized as follows. Section II review related works on fall detection and motion reconstruction; Section III introduces the collaborative computing platform of cloud and WBSN sensor networks proposed in this study, including the overall architecture and module design; Section IV describes the implementation on an experiment platform, with network and falling event detection; Section V offers conclusions.

II. Related Works

This section describes the studies of fall detection and motion reconstruction, as well as the subjects of cloud computing and work dispersion.

A. Fall Detection and Motion Reconstruction on WBSN

With the development of sensor and WBSN technologies, the corresponding sensing range and applications become increasingly extensive. The ubiquitous network was first proposed by Mark Weiser (1991) as ubiquitous computing in "Computing in the 21st Century" [6]. At present, the subject of Ubiquitous is gradually paid attention to by all circles, and relevant theoretical bases are being developed. Anyone and anything can transmit messages conveniently anywhere and anytime. Ubiquitous healthcare is an extended application of the ubiquitous network. As some wearable or thin sensing devices are integrated with a WBSN, the medical care system has gradually evolved from home healthcare into ubiquitous healthcare [7]–[9], where nursing services can be provided at any time using a hand-held physiological system. In the area of fall detection research, there are three common implementation methods, which are fixed type, nonfixed type, and a hybrid type, which is integrated with the advantages of the former two types. The fixed fall detection method fixes or embeds a sensor into a specific environment, and uses the characteristics of the sensor to analyze environmental information [10]. However, the existing fall detection analysis modes can be approximately divided into two types, the threshold analysis method and an intelligent algorithm.

1) Threshold Analysis Method: This analysis mode is simple. First, the numerical values received by the sensor are observed, a reference value is defined, and then fall detection can be directly implemented. When the numerical value received is lower or higher than the defined reference value, it is determined as a fall event. For example, Bourke studied fall detection [11], and set the upper and lower thresholds as the signals of routine actions (sitting, lying, walking), where acceleration of the sensors were smooth. If the received signal is in the threshold range, it is identified as routine action; if the signal exceeds the upper or lower thresholds, it is identified as a fall event. In this experiment, a sensor was placed at the trunk and thigh in order to measure the recognition rate. The results showed that

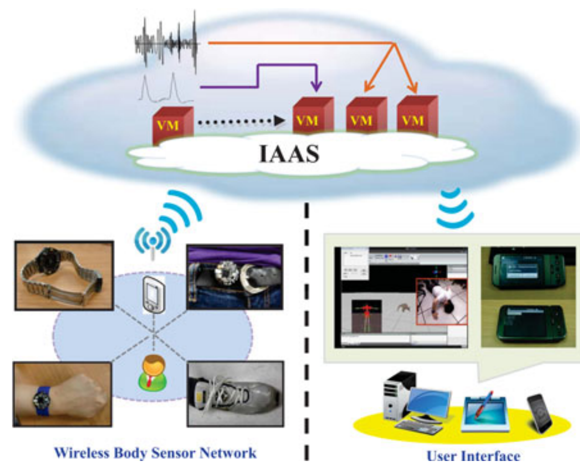

Fig. 1. Architecture of proposed real-time adaptive transrating system.

the overall recognition result was 100% when the sensor was placed on the trunk. In the study of fall detection by [12], a two-axis gyroscope was used as the sensing device, and was affixed to the chest, waist, and lower right part of the arm. First, the angular velocity information of the device in backward and sideways falls and day-to-day activities (e.g., standing, walking, bending, lying, sitting, etc.) was obtained, and the thresholds of actions were determined. The experimental results showed the sensitivity was 100%, and specificity was 84%. In the study of fall detection [13], six horizon sensors were placed inside the clothes (shoulders and both sides of chest and abdomen) as sensing devices. Multiple actions were sensed to obtain the angles and angular velocities of different actions, and thresholds were set to detect fall events. The forward and backward fall detection results showed that sensitivity was 98%, and specificity was 99%. [14] placed two accelerometers in hearing aid housing, which could recognize seven fall postures and five human behaviors. Afterwards, the three-axis accelerometer was gradually used, and falls could be identified more accurately by recording the accelerations of three directions. Yanget [15] placed the three-axis accelerometer at [16], and proposed placing the three-axis accelerometer at the head, thus, accidental falls were recognized according to the accelerations obtained by a three-axis accelerometer.

2) Intelligent Algorithm: In comparison to the threshold analysis method, this type of algorithm has higher complexity of computation, and is sometimes accompanied with the calculation of action simulation, thus, it has the higher recognition rate of fall detection. The information obtained by the sensor is analyzed using a specific method, where the eigenvalue is removed. The type of sorting module is created through learning characteristics, which can be used to classify the information obtained by the sensor to judge the classification result. Common sorting algorithms include support vector machine, linear discriminant analysis, principal component analysis, and artificial neural network. In the study of fall detection, multiple three-axis accelerators were placed at the neck, waist, and right and left wrists and thighs, for a total of six sensors. The human body was divided into the upper body and the lower body, and the inclinations of various sensors during different

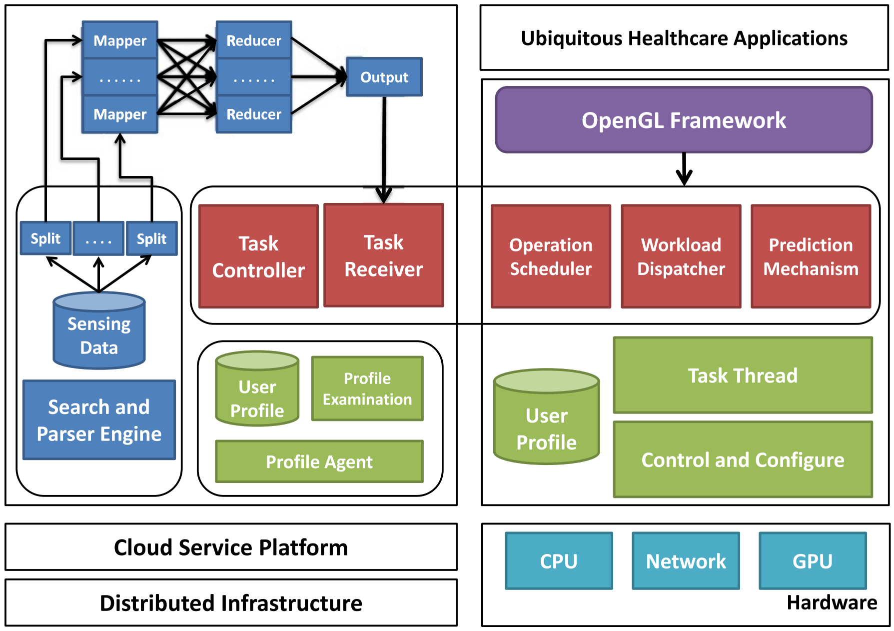

Fig. 2. Architecture of proposed framework.

actions were gathered for analysis. The analysis mode was the subtractive clustering method of the neural network. The center position of habitual inclination in each state was calculated using this method, and set as the basis of the posture determination threshold. The system flow first determined whether the body posture was dynamic or static, if it was static, the posture could be identified according to the inclination angle learned beforehand. If it was dynamic, whether signal strength is too high is checked first. If it was too high, it was regarded as a fall event. The accuracy rate of falls while standing and walking was 100%, and the accuracy of sitting and lying falls was 93.9% and 94.4%, respectively. Ge *et al.* [17] proposed a fall prediction and prevention system, which calculates the seconds of a fall collision while falling, according to the body height, and then the preventer is initiated, thus, providing a good prevention system. Jeon *et al.* [18] constructed a portable fall sensing system in a mobile device, which eliminated the indoor restriction of computer receiving, thus, the users could be protected by fall detection when outdoors.

B. Cloud Network

Cloud network are based on computation, software, data access, and storage technologies that are remotely provided to offer a degree of freedom in the computing infrastructure by abstracting network resources from the users [19]. Hence, users need only know their own application or content that is running or stored in the cloud services. Various cloud infrastructures have been studied prior to this research. Mostly Infrastructure as a Service (IaaS) cloud structures were studied since this research focuses on building its own scheduling algorithm within the system. It targets to solve the complexity of integrating various cloud service scheduling, which Li states the dynamic cloud scheduling issue remains largely unexplored. The research [20] presented a linear integer programming model that migrate virtual machine (VM) across clouds to provide flexibility. The model can be used to adapt to changes within infrastructure and service. The model is also evaluated against commercial cloud settings and it proves applicable in dynamic cloud scheduling cases in terms of performance. The research analyzes the demand for the number resources by setting a Trust degree for resources for users to choose highly reliable resources. The Trust degree mentioned in [21] is the number of failures since the virtual slot has been put into the resource pool. In the paper, the task requests are server access to the VM instances, where the scheduling monitoring mechanism distributes the requests by trust degree in a descending order. The scheduling monitor will timely collect the trust degree calculation of the slots. A record table is set and it saves the trust degree of the resources for the entire cloud. It studied provisioning for applications to solve the obstruction for smooth provisioning and delivery of application services. This paper presented a mechanism which detects the amount of VM requests within the system to create or destroy VM within the cluster. The analytical performance and workload information are provided by the cluster to meet quality of service (QoS) requirements of the service time and whether or not to reject the requests [22]. A survey for scheduling algorithms used in cloud environments by comparing various popular algorithms and the required parameters. Most of the algorithms that exist consider mostly throughput and cost effectiveness while disregarding reliability and availability [23].

III. A Collaborative Computing Framework of Cloud Network and WBSN

This study intends to build a collaborative computing framework of cloud network and WBSN for ubiquitous fall detection and 3-D motion reconstruction. Fig. 2 shows the overall framework architecture. The mobile terminal transmits the collected sensor data via wireless network to the cloud environment for personal classification, and relevant information is stored, thus, users can obtain cloud related physiological services from hand-held devices, and hospitals and relevant medical units can modify cloud services by observing the historical physiological information of patients, rendering the cloud system more suitable for each patient. The cloud provides services to implement a ubiquitous care system, thus, increasing the efficiency of medical resources.

Profile Agent is used to receive environmental parameters and create a user profile. The mobile device transmits its hardware specifications in the XML–Schemas format. However, a mobile device using the cloud service for the first time cannot provide such a profile; therefore, there is an additional profile examination function, which aims to provide the test effectiveness of the mobile device and sample relevant information. Based on this function, the mobile device can generate an XML–Schemas Profile and send it to the Profile Agent.

The intermediary collaborative working layer extracts effective sensor information, and allocates the operational data volume to both sides according to sensor information content, network bandwidth, and processing speed, transmits the partially sensed data volume to the cloud environment for calculation, and the result is fed back in order to reduce massive data transmissions and accelerate computing speeds.

The cloud computing mainly uses the MapReduce operating mode, where overall computing tasks are dispersed to various nodes. Finally, the computing result is fed back to the mobile device side.

A. Intermediary Collaborative Work Allocation Module

When the mobile device collects various sensor nodes, the intermediary collaborative working layer reaches the instancy and precision of application services for different kinds of sensor information, the sampling of sensor signals must be effective, namely, extracting signal retrieving from the starting point to end point of the user's action. Cloud works and self-computing services are effectively allocated according to the sensor data volume and mathematical capability of WBSN. The intermediary collaborative working layer extracts effective sensor information, and effectively allocates operational data volume to both sides according to sensor information content, network bandwidth, and processing speed. A part of the sensed data volume is transmitted to the cloud environment for calculation, and the result is fed back, thus, reducing massive data transmission and accelerating computing speed. This management layer supports the parallel processing of multiple application programs. In order to effectively exert the service of the intermediary management layer, it must consist of three major subfunction modules, Workload Splitter, Workload Dispatcher, and Operation Scheduler. The function design and purpose of this study are described, as follows:

1) Prediction Mechanism: In order to reduce the huge amount of computation and multistate changes, this framework proposes a prediction mechanism that adopts a sliding window principle in the current state, and considers avoiding excessive amounts of computations. Therefore, the sensor signals are observed in a fixed time cycle in order to judge whether there is action. The determination of occurrence is based on the X-component (A_x) of the three-axis accelerator built into the platform, and ADC samples 100 times per second and stores the data in the sampling buffer. For complete action detection, the starting time point must be determined, and then the end time point shall be found. First, the standard deviation D_{Ax} of A_x in the sampling buffer is calculated in a fixed 0.2 s cycle, and the D_T threshold is 0.03. If D_{Ax} is smaller than D_T, it means the user is in a steady state without action; and on the contrary if the signal begins to oscillate and move. If there is no action, the Sensor Observer records the D_{Ax} and average A_{Avg} of A_x in the sampling buffer, and then empties the sampling buffer and continues to seek the starting point. If an action is detected, the Sensor Observer stores all the information from the buffer into the Target Segments, marks it as a Start Segment, and enters the stage of searching for the end point. At this point, all of A_x in the sampling buffer are visited in turn and compared with the D_{Ax} and A_{Avg} before searching for the starting point, and with D'_{Ax} and D_T of the current sampling buffer, as shown in (1) and (2).

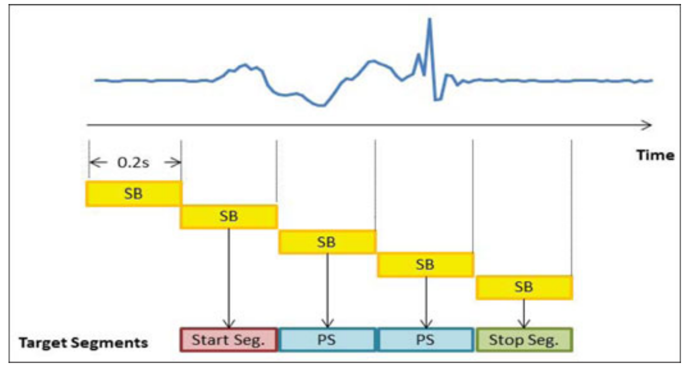

Fig. 3. Slide window and segment capacitor.

$$|A_x - A_{\mathrm{Avg}}| < D_{Ax} \tag{1}$$

$$D'_{Ax} < D_T. \tag{2}$$

If (1) and (2) are tenable, meaning the signals become steady again, the user's action ends. The Sensor Observer can mark the information in the buffer as a Stop Segment and store it in the Target Segments. On the contrary, if the action is still in progress, the Sensor Observer marks the information in the buffer as Progressing Segment and moves it to the Target Segments. When an action is successfully detected, the Target Segments can be delivered to another subsystem. The relationship between the sampling buffer, Target Segments, and the overall signal extraction processes, are as shown in Fig. 3.

2) Workload Dispatcher: This submodule calculates the operational data volume of a sensor and considers the mathematical capability of the mobile device and network bandwidth delay, and then distributes data groups for calculation. This study defines the sensor work data group $S = \{S1, S2Sn\}$, the mathematical capability of the computing node of WBSN is $C = \{C_1, \ldots, C_m\}$, and the predicted network bandwidth is calculated.

$$B'(t) = B(t-1) + D \tag{3}$$

where $B'(t)$ is the estimated bandwidth in No. t time interval, $B(t-1)$ is the bandwidth in No. $t-1$ time interval, and D is the estimation difference.

And the overall working time can be calculated.

$$T = S_w/C + S_c/B'(t) + T_{\text{cloud}} \tag{4}$$

where T_{cloud} is the cloud computing processing time required by the user, S_w is the processing data allocated to WBSN, and S_c is the data transmitted to the cloud for processing.

In this study, the wireless network transmission delay in WBSN is assumed to be very small, thus, the data volume of feedback result values after cloud computing is disregarded.

Meanwhile, the power consumption of the mathematical unit in the computing node of WBSN is assumed to be W_c, the power consumption of the wireless network transmission unit is W_t, and the power consumption of the computing node is approximately estimated at

$$P = W_c * S_w/C + W_t * S_c/B'(t). \tag{5}$$

According to (2) and (3), we can conduct optimal scheduling operations for T.

3) Operation Scheduler: The submodule adopts three algorithms for scheduling. The round robin algorithm is quite simple. Using the nova-api to collect a list of nodes and the first node from the list is chosen. The list is received in IP order, then the requests are assigned to the nodes in order. The requests will be split into clips for transcoding. The clips will be assigned to each VM in round robin order as well. A list of VM instances are collected by order of creation. The Genetic algorithm uses the chromosome of node order for each request. It uses the number of instance on each node and the amount of segments that would be assigned to each node for the fitness function. This is more like a CPU usage combined with the execution of the round robin scheduling. The algorithm performs with the following steps. Requests coming in at a time point are analyzed. The request consists of the length of the video requested split into minor encode requests. The request will be ordered by time length. The CPU usage amounts of each node of the cluster are renewed every few seconds. The nodes are ordered by CPU usage. The requests are assigned by the longer requests given to the nodes with lower CPU usage up to the shorter requests given to the nodes with lower usage. If there are more requests than there are nodes, a round robin pattern will be used. For example, if there are five requests to four nodes, the first four requests are assigned as originally planned by the first three steps. Then, for the last request, it will be assigned to the node with the least CPU usage.

B. MapReduce Architecture in Cloud Computing

For framework operations, the allocation of work roles must be first discussed. In a MapReduce computing environment, the work roles include Master Node and Worker Node; generally speaking, there is one Master Node and multiple Worker Nodes. The Master Node plays the core role of coordinating and managing the entire computing framework. The Master Node can allocate appropriate workloads to various nodes for calculation according to the workload required for the application program, the mathematical capability of the existing Worker Node, the evaluation of network resources, and finally recovers the results. The Worker Node aims at calculations, and is in charge of processing the mathematical logic defined by the developer. For application program developers, all work can be executed in two stages, as mentioned in the Map and Reduce stages. In other words, the developers must design the mathematical functional equations of Map and Reduce, respectively, using the application program interface provided by the computing framework, and then define the Key and Value of each input data.

Mapper is mainly used in calculating the distance between center points of falling events to every sensing data point that is grouped to the nearest center point, and Reducer organizes the Mapper clustering falling event results and prepare for external processing. The pairing of Key and Value is usually called a Record. The Mapper functional equation is in charge of processing (k_1, v_1) all input data, and generating a series of intermediary data Lists (k_2, v_2). This intermediary data group contains multiple records, and each record consists of paired corresponding Key and Value.

In the Reducer procedure, the results of calculating cluster center points can be used to calculate the new center points of each clustered group. Then, the status of clustering will be checked whether it has been completed to compare the new center points and existing center points. If the new center points are equal to existing ones, the center points is considered to be convergent. After grouping all data into trained center points, each data entry is mapped to one of the center points for forming the groups. The Reducer functional equation processes the intermediary data group $(k_2, \text{List}(v_2))$, using the same Key, but different Values, to generate the final required List (k3, v3). In other words, all input data have corresponding Key and Value records, and the Worker Node in the computing environment uses the Mapper functional equation to process one into multiple records, thus, generating a series of intermediary data. This series of intermediary data have corresponding Key and Value intermediary records. Intermediary records can be calculated by the Reducer functional equation to generate the final result. For convenient calculation of the Reducer stage, the Shuffle stage or Group stage is conducted after the Mapper stage. The purpose of this stage is to rearrange the intermediary data group, i.e., intermediary records derived from the Mapper stage, into a sequence, or concentrate the intermediary data with the same Key for the Reducer operation in the final stage. The schematic of the framework operation is as shown in Fig. 4.

IV. System Implementation and Experiment

This section introduces the implementation of a system prototype and related experimental analysis. This case uses seven sets of wearable accelerometers to record the acceleration and angular acceleration values of seven positions of the human body, including the head, neck, both hands, waist, and both feet.

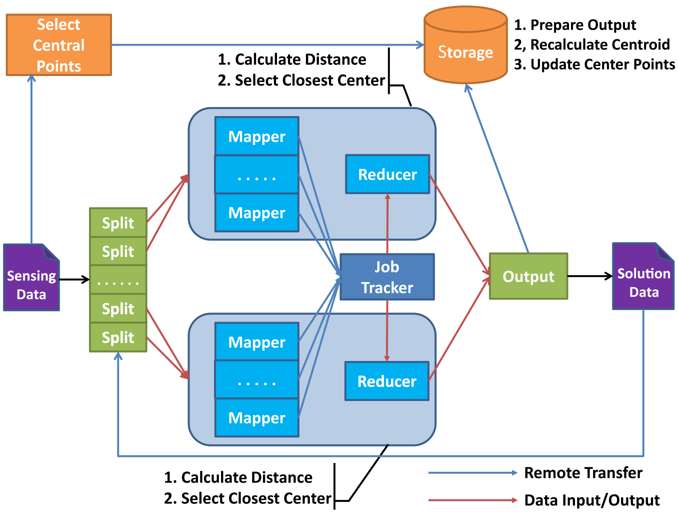

Fig. 4. Operating of MapReduce framework.

TABLE I
THE IMPLEMENTATION FUNCTION OF MAP OPERATING

Function	Description
AddMapInputRecord()	Input Key/Value
MapReduce()	Start Function
FinishMapReduce()	End Function
EmitInterCount()	Return size and number
EmitInter()	Return metadata Info
Emit()	Return Final data Info

The sensors sense 48 times per second and transfer this data to a mobile device. The system will calculate the implementation of fall detection and 3-D action reconstruction. The measured actions include walking, running, ascending stairs, descending stairs, falling, and jumping to test the action signals.

A. *Implement Fall Detection Application in Cloud Computing*

In order to enable the application program to run normally in the framework, the developers shall define the mathematical logic of Map and Reduce, and must use the application program interface provided by the framework to communicate with the program in the framework, including the corresponding Key and Value pairing records for each defined input problem data, the universal functional equations of result feedback in the Map and Reduce calculation completion stage, framework startup, and termination. The common framework application program interfaces in application programs are as shown in Table I.

In terms of the algorithm for fall detection, this study uses K-Means and Bayesian inference for analysis and calculation. The K-Means clustering algorithm is a common data clustering algorithm used in the data mining domain, and is usually used for massive data clustering. The first purpose is to determine a suitable cluster center of the various data clusters, and to use the minimum distance between the data and various cluster centers as the basis of clustering. In other words, the clustering is expressed as (6)

$$\arg\min \sum_{i}^{k} \sum_{x_j \in S_i} \|X_j - U_i\|^2 \tag{6}$$

where k is the number of clusters to be clustered, represents clusters, and represents the average of the clusters. Each data can be a set of multidimensional observations. Before data clustering, the algorithm randomly selects a number of data equal to the number to be clustered from all the data as the initial cluster center of the various clusters. Afterwards, the distance between all data and each initial cluster center is calculated, and the cluster center at the minimum distance is designated as the cluster. Based on such calculation, each data has its cluster, and generates a new cluster. When the new cluster is generated, the average of all data in the cluster is used as the new cluster center of the cluster. The distances between all data and all new cluster centers are calculated, and the minimum distance is used for clustering. This step is repeated until all cluster members have completed changing, then all the data are clustered. According to the aforementioned algorithm, the allocation of clusters of data and the selection of new cluster center are expressed as (7) and (8).

$$S_i^{(t)} = \{x_j : \|x_j - u_i^{(t)}\| \le \|x_j - u_j^{(t)}\|\} \tag{7}$$

$$u_i^{(t+1)} = \frac{1}{|s_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j. \tag{8}$$

The K-Means clustering algorithm implemented in the framework takes each data as a record. The Key in each record contains fields for a serial number, dimensions, and cluster center number of the data. The Value contains the fields of data content and cluster center data. The algorithm runs in the MapReduce framework, the distances to all cluster centers are calculated in the Map stage, and the cluster center at the minimum distance is selected to update the cluster center number of the Key. Records with the same cluster center number in all Key values are sequenced in the Group stage. Finally, the new cluster centers of all the clusters are recalculated in the Reduce stage, and the cluster center data field bit Value in the corresponding record is updated. The clustering of all data can be completed by repeating the aforesaid steps.

The Bayesian Theorem is used to calculate the conditional probability of events, the prior probability, e.g., $P(B)$ and the conditional probability, e.g., $P(B|S_y)$ of relationship between variables S and B must be used. If, in an event B, the probable cases are turned into several repulsive parts, $S_1, S_2, \ldots\ldots, S_y$, the posterior probability can be obtained using the inference algorithm proposed by the Bayesian Theorem. The calculation method of Bayesian inference is expressed, as follows:

$$P(S_y|B) = \frac{P(B \cap S_y)}{P(B)} = \frac{P(B|S_y)P(S_y)}{\sum_{i=1}^{y} P(B|S_i)P(S_i)} \tag{9}$$

where B is the set of actions sensed by various sensors, S is the event of an accidental fall, and the relative urgency correlation can be worked out by Bayesian inference.

When the numerical value of the Bayesian inference is obtained, the relative urgency correlation can be converted into absolute urgency correlation using the correlation coefficient. At this point, the correlation coefficient theory for measuring the common changes of two variables can be used. It is usually used to analyze the relevance and similarities between files. Any file may be given its serial eigenvalues, and the serial eigenvalues are the eigenvectors of the file. According to the definition of the similarity integral correlation can be expressed as (10).

$$|\mathrm{corr}(X,Y)| = \frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2(Y-\bar{Y})^2}}. \tag{10}$$

The general correlation coefficient is expressed as $\mathrm{corr}(X,Y)$. In order to determine the urgent correlation between S_1 and S_2, the X and Y in the equation are substituted in the Bayesian inference probability values of S_1 and S_2, respectively, and are the average Bayesian inference probability values. In fact, the correlation coefficient measures the linear correlation between two random variables, namely, the closer the value of $|\mathrm{corr}(X,Y)|$ to 1, the higher the linear correlation between *X* and *Y*. On the contrary, the closer the value of $|\mathrm{corr}(X,Y)|$ to 0, the lower the linear correlation between them.

B. Implement Action Recurrence Application in Cloud Computing

This study uses the openGL architecture to simulate the motions of the body. OpenGL is a specification defining a cross-program language and cross-platform application programming interface. It is used to generate 2-D and 3-D images. This interface consists of approximately 350 function calls, for drawing complex 3-D scenes from simple graphics bits. OpenGL is used in cloud environments, and the mobile device is openGL Es. The overall architecture is displayed on the mobile device according to the Android Graphics Architecture. As the action data for action recurrence have dependence, this study divides the data into seven groups, as shown in the following table. The reconstruction of a fall process consists of two major parts. One is the construction of the body, which is divided into various body parts according to the parts that can be represented by sensors. The other one is process reconstruction, where the bones and joints of the skeleton are integrated with the data of sensors on the body to calculate the skeletal actions, thus, retrieving body fall postures.

The human body has many complex and specific bones; however, as the sensors fixed to the body are limited, in order to reach precise posture recurrence, the first step of posture recurrence is to simplify the skeleton. It is necessary to create a simple human skeleton that will not influence posture recurrence, and allow the retrieval of a fall process according to the sensor data. The degree of simulation is determined by the number of sensors. Based on the recurrence of most body actions in this study, the human skeleton can be reduced to 22 parts, as shown in Fig. 5.

In this section, we discuss how to use the data of six sensors to calculate the action tracks of the 22 parts of the body. The positive and reverse movements of the bones of various areas are calculated to obtain the effect of skeletal movement. As the calculation of positive movement and reverse movement is based on the positions of joints, the sensors shall be mounted at the joints. According to the skeletal partition, as sensors are mounted, the six nodes have detailed spatial location information, thus, it is unnecessary to use positive or reverse movement equations to calculate spatial location. Therefore, these points can be regarded as fixed points that are free from the effect of motion equations. In other words, the six points are the static joints of motion calculation, and their positions are influenced only by sensor information. The motions of static joints are determined by sensor information, while the motion information of nonstatic points is obtained by positive movement and reverse movement equations. According to the hierarchical structure of the skeleton, positive movement influences all the subnodes of the motion nodes. N_2 is the subnode of N_1, and N_1 is the subnode of N_0, P_2, P_1, and P_0 are joints of N_2, N_1, and N_0, respectively. If P_0 is a static node in space, it is observed that the motion of joint P_1 will influence joint P_2, and not joint P_0.

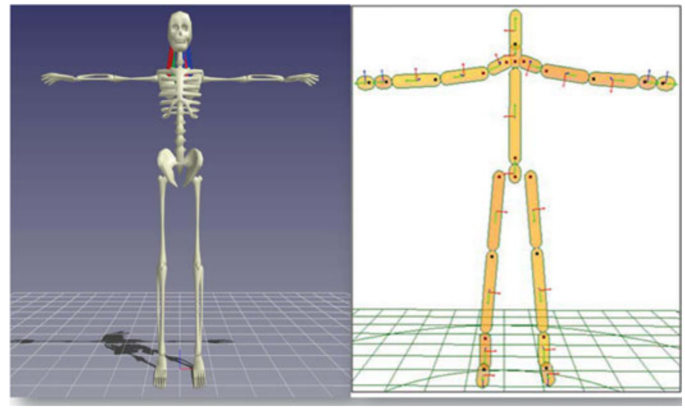

Fig. 5. Structure of human body skeleton.

In terms of the six subareas of the skeleton, the parts using positive movements are in the bottommost static node, i.e., from the hands and feet to the bottom of the hierarchical structure of the skeleton; while the motion mode is a hollow that represents a nonstatic point, and solid represents a static point, i.e., the point with sensor information. In terms of the computing mode of positive movements, as based on the example, if there are three bone nodes, which are $Node_0$, $Node_1$, and $Node_2$, the joints are P_0, P_1, P_2, and P_3. The bone lengths are L_1,L_2, and L_3. When Node1 shifts and rotates, it influences $Node_2$ joints P_2 and P_3. If the $Node_1$ translation distance vector is d, and the rotation angle is θ, then (x_1, y_1, z_1) are the coordinates of P_1, the coordinates of joint P_2 are (x_2, y_2, z_2), where the components of d to x, y, and z axes are the rotational components of at Euler angle on x, y, and z axes.

Regarding the reverse movements of the skeleton, according to the hierarchical structure of the skeleton, the joints of the parent node are influenced. For example, the motion mode is as shown in Fig. 6. Where N_2 is the subnode of N_1, N_1 is the subnode of N_0, and P_2, P_1, and P_0 are the joints of N_2, N_1, and N_0, respectively. If P_0 is a static node in space, it is observed that the joint P_2 shifts left, directly influencing the position of

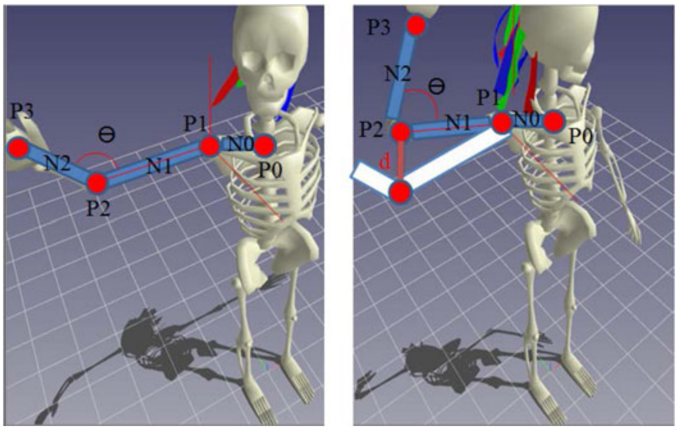

Fig. 6. Forward movement of human body skeleton.

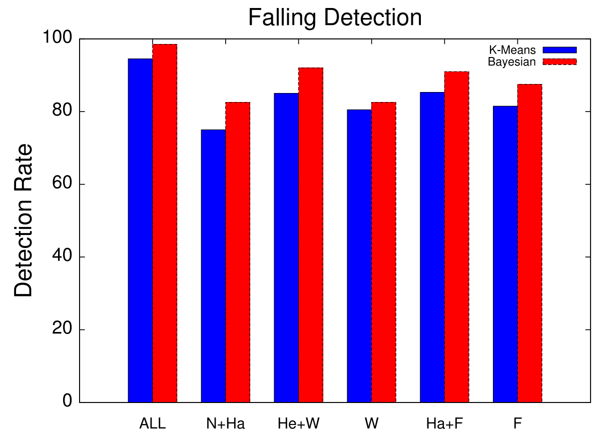

Fig. 7. Fall detection with different sensor set.

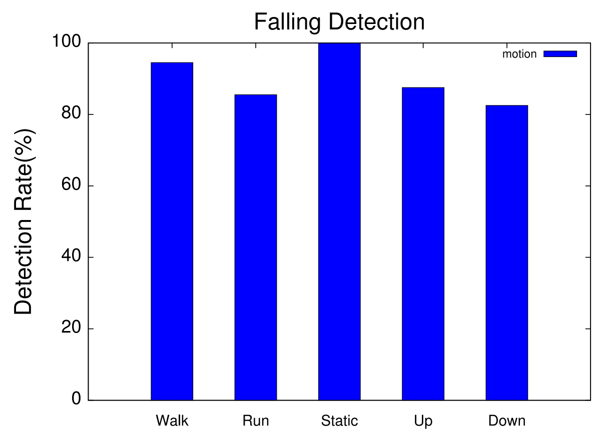

Fig. 8. Fall detection with different motion.

joint P_1 and the axial direction of the parent node. When the skeleton is divided into six areas to simplify the calculations of the processing motion, the parts requiring reverse movement calculation are reduced. The migration of the static endpoint is calculated by sensor. The bone that requires calculation by reverse movement is between the static nodes, and there must be more than one joint between two static nodes. According to the constructed skeleton, only the upper part of the arm meets this condition. It is found that the final position of the left clavicle nonstatic joint shall be calculated by reverse movement, due to the migration of the left humerus. The reverse movement of the clavicle shall be calculated as the left hand shifts left. This system determines the mathematical unit of theaforementioned seven major data groups according to the work allocation module, and calculates the position units of the human body. Finally, the results are transmitted to a mobile device, and the overall result is drawn according to this computed result. The overall computing result can be transmitted to the cloud environment for drawing, the action reconstruction can then be displayed on any device not supported by the openGL in streaming mode.

C. Experimental Data

Based on the research data, this fall detection application is experimented upon according to the quantity of sensors, and the effectiveness of cloud-based collaborative computing is analyzed.

1) Fall Detection in Different Motion States: This experiment aims at the influence of different sensor combinations on the fall recognition of two algorithms. The K-Means algorithm is a simple cluster center combination, and its recognition rate is not obvious. Bayesian inference has obvious accidental fall inference according to the combination of footstep and waist. This study uses seven sets of sensing devices for 3-D human body simulation.

Based on the aforementioned cluster states of various actions, in testing for fall detection, the fall states are divided into falls while walking, running, resting, and on stairs. Ten persons are tested, and ten groups of data in each state for each person are obtained. The detection results are as shown in Fig. 7, and fall detection results are discussed below. In the case of walking, as the body and feet move at a stable frequency, and the speed is low, the kinematic trajectory points are uniformly distributed. The population is extensive, the probability of misrecognition of an action change is low, and the body is hardly to have high acceleration and instant high acceleration variation. Therefore, when a fall or collision occurs, a high acceleration change can be obviously detected. It is a motion with a high success ratio in fall detection. The action trajectory population distribution of running inclines to a dense large population and some dispersive small populations, as a fall often occurs due to off balance resulted during stepping, e.g., slip or stumble. However, if a fall occurs when the feet are not stepping on the ground, namely, both feet are off the ground, the misrecognition probability of fall detection increases with the number of dispersive small populations. In a stationary state, the kinematic trajectory clustering result is almost a stably distributed population, and the body has no acceleration or acceleration changes. When an accidental fall occurs, the change in motion can be identified easily through populations or the high acceleration of the body. Therefore, there will be few misrecognitions in fall detection as shown in Fig. 8. The case of stairs is similar to a slope, as the regional distribution between the steps of a kinematic trajectory is more uniform. In addition, due to landform, stairs with steps at a fixed length force one to walk the height and length of each step each time, thus, the overall region of the kinematic trajectory converges in several regions, relatively increasing fall detection accuracy.

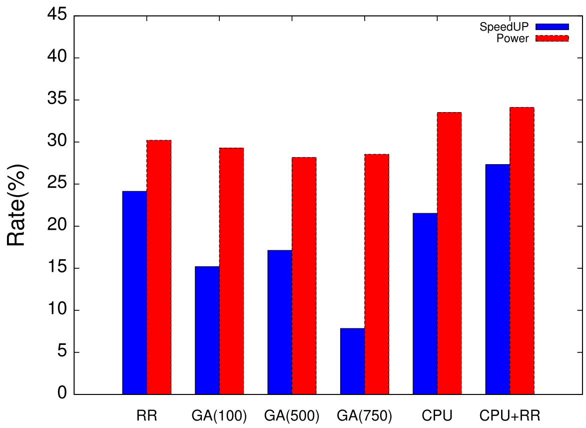

Fig. 9. Speed and power consumption with different algorithms.

2) Influence of Distributed Collaborative Computing on Computing Effectiveness: The influence of different work allocation models on overall effectiveness is experimentized, and the overall experimental data are shown in Fig. 9. According to the results, the RR algorithm has the minimum influence on the increase in overall effectiveness. The CPU usage algorithm and RR can effectively increase the overall speed. The GA aims at the influence of effective generation, while overlong generation prolongs the overall computing time, thus, reducing effectiveness. In terms of the effect of different allocation algorithms and recognition algorithms on the power consumption of mobile devices, the results are as shown in Fig. 9. It is observed that although various algorithm result in different power consumptions, the collaborative computing work platform can effectively reduce the power consumption of mobile devices for switching on a wireless network and frequent data transmission. The cloud environment can bear partial work units to reduce the requirements for the operations of overall mobile devices.

V. Conclusion

This study proposed an intermediary collaborative framework of cloud and WBSN, and applies it to fall detection and 3-D action reconstruction environments. The module is easily worn by users in order to measure the body motion state, which is transmitted to mobile devices via wireless transmission module, thus, users can watch the accidental fall sensing and simulated actions on their mobile devices. The intermediary collaborative framework implements collaborative computing of sensed data and cloud computing according to the number of sensors and the mathematical capability of WBSN, thus, avoiding the transmission of mass data, resulting in the power consumption of mobile devices and loss of network bandwidth. This study implements several work allocation models and the application of cloud Mapreduce works to fall detection and body reconstruction simulation operations. This design can effectively increase overall computing effectiveness and reduce the power consumption of mobile devices, and therefore, can implement ubiquitous sensing services.

References

[1] C. F. Lai, R. Zhu, B. F. Chen, and Y. Lee, “A 3D Falling reconstruction system using sensor awareness for ubiquitous healthcare,” *Sens. Lett.*, vol. 11, no. 5, pp. 828–835, 2013.

[2] M. Krunz, A. Muqattash, and S.-J. Lee, “Transmission power control in wireless ad hoc networks: Challenges, solutions and open issues,” *IEEE Netw.*, vol. 18, no. 5, pp. 8–14, Sep./Oct. 2004.

[3] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic, “ATPC: Adaptive transmission power control for wireless sensor networks,” in *Proc. 4th Int. Conf. Embedded Netw. Sens. Syst.*, New York, NY, USA, 2006, pp. 223–236.

[4] H. Wang, D. Peng, W. Wang, H. Sharif, H. H. Chen, and A. Khoynezhad, “Resource-aware secure ECG healthcare monitoring through body sensor networks,” *IEEE Wireless Commun. Mag.*, vol. 17, no. 1, pp. 12–19, Feb. 2010.

[5] W. Wang, H. Wang, M. Hempel, D. Peng, H. Sharif, and H. H. Chen, “Study of stochastic ECG signal security via Gaussian Mixture model in wireless healthcare,” *IEEE Syst. J.*, vol. 5, no. 4, pp. 564–573, Dec. 2011.

[6] M. Weiser, “The computer for the twenty-first century,” *Sci. Amer.*, pp. 94–10, Sep. 1991.

[7] C. He, X. Fan, and Y. Li, “Toward ubiquitous healthcare services with a novel efficient cloud platform,” *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 230–234, Jan. 2013.

[8] H. Viswanathan, B. Chen, and D. Pompili, “Research challenges in computation, communication, and context awareness for ubiquitous healthcare,” *IEEE Commun. Mag.*, vol. 50, no. 5, pp. 92–99, May 2012.

[9] Z. Zhang, H. G. Wang, A.V. Vasilakos, and H. Fang, “ECG-Cryptography and authentication in body area networks,” *IEEE Trans. Inform. Technol. Biomed.*, vol. 16, no. 6, pp. 1070–1078, Nov. 2012.

[10] F. Chiti, R. Fantacci, F. Archett, E. Messina, and D. Toscani, “An integrated communications framework for context aware continuous monitoring with body sensor networks,” *IEEE J. Select. Areas Commun.*, vol. 27, no. 4, May 2009.

[11] A. K. Bourke, J. V. O’brien, and G. M. Lyons, “Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm,” *Gait Posture*, vol. 26, no. 2, pp. 194–199, 2007.

[12] M. N. Nyan, F. E. H. Tay, and E. Murugasu, “A wearable system for pre-impact fall detection,” *J. Biomechanics*, vol. 41, no. 16, 3475481, 2008.

[13] T. Degen, H. Jaeckel, M. Rufer, and S. Wyss, “SPEEDY: A fall detector in a wrist watch,” in *Proc. 7th IEEE Int. Symp. Wearable Comput.*, 2003, pp. 184–187.

[14] U. Lindemann, A. Hock, and M. Stuber, “Evaluation of a fall detector based on accelerometers: A pilot study,” *Med. Biol. Eng. Comput.*, vol. 43, p. 1146154, 2005.

[15] C. C. Yang and Y. L Hsu, "Development of a portable system for physical activity assessment in a home environment, in *Proc. Int. Comput. Symp.*, Taipei, Taiwan, Dec. 2006, pp. 1339–1344.

[16] C. C. Wang, C. Y. Chiang, P. Y. Lin, Y. C. Chou, I. T. Kuo, C. N. Huang, and C. T. Chan, “Development of a fall detecting system for the elderly residents,” in *Proc. 2nd Int. Conf. Bioinformatics Biomed. Eng.*, 2008, pp. 1359–1362.

[17] G. Wu and S. Xue, “Portable preimpact fall detector with inertial sensors,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 2, pp. 178–183, Apr. 2008.

[18] Z. Zhang, H. Wang, C. Wang, and H. Fang, “Interference mitigation for cyber-physical wireless body area network system using social networks,” *IEEE Trans. Emerging Topics Comput.*, vol. 1, no. 1, pp. 121–132, Jun. 2013.

[19] R. N. alheiros, R. Ranjan, and R. Buyya “Virtual machine provisioning based on analytical performance and QoS in cloud computing environments,” in *Proc. Int. Conf. Parallel Process*, Sep. 13–16, 2011, pp. 295–304.

[20] W. Li, J. Tordsson, and E. Elmroth, “Modeling for dynamic cloud scheduling via nigration of virtual machines,” in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2011, pp. 163–171.

[21] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. H. J. Epema, “Performance analysis of cloud computing services for many-tasks scientific computing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 6, pp. 931–945, Jun. 2011.

[22] S. Chaisiri, B.-S. Lee, and D. Niyato, “Optimization of resource provisioning cost in cloud computing,” *IEEE Trans. Serv. Comput.*, vol. 5, no. 2, pp. 164–177, Apr.–Jun. 2012.

[23] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 847–859, May 2011.

Chin-Feng Lai (M'07) was born in Taiwan in 1980. He received the Ph.D. degree from the Department of Engineering Science, National Cheng Kung University, Taiwan, in 2008.

He is currently an Assistant Professor with the Department of Computer Science and Information Engineering, National Chung Cheng University since 2013. His research interests include multimedia communications, sensor-based healthcare, and embedded systems. After receiving the Ph.D. degree, he has authored or co-authored more than 60 refereed papers in journals, conference, and workshop proceedings about his research areas within two years. He was selected as an Honorary Member of the Phi Tau Phi Scholastic Honor Society of the Republic of China and as a TC Member of Multimedia Systems & Applications Technical Committee IEEE Circuits and Systems Society, both in 2009. He is also a Project Leader of several international, industrially funded multimedia projects. Now, he is making efforts to publish his latest research in theIEEE TRANSACTIONS ON MULTIMEDIA and the IEEE TRANSACTIONS ON CIRCUIT AND SYSTEM ON VIDEO TECHNOLOGY. He is also a Member of the IEEE Circuits and Systems Society and the IEEE Communication Society.

Min Chen (SM'09) is currently a Professor in the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He was an Assistant Professor in School of Computer Science and Engineering, Seoul National University (SNU), Seoul, Korea, from Sep. 2009 to Feb. 2012. He was R&D director at Confederal Network Inc. from 2008 to 2009. He worked as a Postdoctoral Fellow in Department of Electrical and Computer Engineering at University of British Columbia (UBC), Vancouver, BC, Canada, for three years. Before joining UBC, he was a Postdoctoral Fellow at SNU for one and half years. He has more than 170 paper publications. He serves as an Editor or Associate Editor for *Information Sciences*, *Wireless Communications and Mobile Computing*, *IET Communications, IET Networks, Wiley I. J. of Security and Communication Networks, Journal of Internet Technology, KSII Transactions Internet and Information Systems, International Journal of Sensor Network*s. He is Managing Editor for IJAACS and IJART. He is a Guest Editor for IEEE Network, IEEE Wireless Communications Magazine, etc. He is Co-Chair of IEEE ICC 2012-Communications Theory Symposium, and Co-Chair of IEEE ICC 2013-Wireless Networks Symposium. He is General Co-Chair for the 12th IEEE International Conference on Computer and Information Technology (IEEE CIT-2012).

Mr. Chen received the Best Paper Award from IEEE ICC 2012, and the Best Paper Runner-up Award from QShine 2008.

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology, Taiwan, in 1986, the M.S. degree in communication engineering from the National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, Edinburgh, U.K., in 1996.

Currently, he is a Professor with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan. He is also invited to be the Doctoral Advisor both at the University of South Australia, Adelaide, Australia, and Harbin Institute of Technology, Harbin, China. He joined the editorial board of *International Journal of Innovative Computing, Information and Control*, *LNCS Transactions on Data Hiding and Multimedia Security*, *International Journal of Hybrid Intelligent System*, *Journal of Information Assurance and Security*, *International Journal of Computer Sciences and Engineering System*, *Journal of Computers*, *International Journal of Digital Crime and Forensics*, *ICIC Express Letters*, and *Computational Intelligence and Its Applications Book Series (IGI Publishing)*. His current research interests include soft computing, information security, and signal processing.

Chan-Hyun Youn (M'87) received the B.Sc. and M.Sc. degrees in electronics engineering from Kyungpook National University, Daegu, Korea, in 1981 and 1985, respectively. He also received the Ph.D. degree in electrical and communications engineering from Tohoku University, Japan, in 1994.

He served in the Korean Army as a Communications Officer, First Lieutenant, from 1981 to 1983. Before joining the University, from 1986 to 1997, he was the Leader of high-speed networking team at KT Telecommunications Network Research Laboratories, where he had been involved in the research and developments of centralized switching maintenance system, maintenance and operation system for various ESS's system, high-speed networking, and HAN/B-ISDN network testbed. Especially, he was a Principal Investigator of high-speed networking projects including ATM technical trial between KT and KDD, Japan, Asia-Pacific Information Infrastructure testbed, Korea Research and Education Network, and Asia-Pacific Advanced Network, respectively. Since 2009, he has been a Professor with the Department of Electrical Engineering in KAIST, Daejeon, Korea. He also was a Dean of Office of Planning Affairs and a Director of Research and Industrial Cooperation Group at former Information and Communications University, in 2006 and 2007. Currently, he is serving as an Editor-in-Chief in KIPS (Korea Information Processing Society), and Editor of Journal of Healthcare Engineering (U.K.), and served as Head of Korea branch (computer section) of IEICE, Japan (2009, 2010). He is a Member the KICS and IEICE.

Han-Chieh Chao (SM'92) received the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1989 and 1993, respectively.

He is currently a joint appointed Full Professor with the Department of Electronic Engineering and the Department of Computer Science & Information Engineering where he also serves as the President of National Ilan University, I-Lan, Taiwan. He has been appointed as the Director of the Computer Center for Ministry of Education starting from September 2008 to July 2010. His research interests include high speed networks, wireless networks, IPv6 based networks, digital creative arts and digital divide. He has authored or co-authored four books and has published about 280 refereed professional research papers. He has completed 100 MSEE thesis students and three Ph.D. students. He has also received many funded research grants from NSC, Ministry of Education (MOE), RDEC, Industrial Technology of Research Institute, Institute of Information Industry, and FarEasTone Telecommunications Lab. He has been invited frequently to give talks at national and international conferences and research organizations. He is the Editor-in-Chief for I*ET Communications, Journal of Internet Technology, International Journal of Internet Protocol Technology,* and *International Journal of Ad Hoc and Ubiquitous Computing*. He has served as the Guest Editor for Mobile Networking and Applications (ACM MONET), IEEE JSAC, IEEE Communications Magazine, Computer Communications, IEE Proceedings Communications, the Computer Journal, Telecommunication Systems, Wireless Personal Communications, and Wireless Communications & Mobile Computing. He is a Fellow of the IET (IEE) and a Chartered Fellow of the British Computer Society.

Dr. Chao has received many research awards, including Purdue University SRC awards, and NSC research awards (National Science Council of Taiwan).