A Real-Time Human Action Recognition System Using Depth and Inertial Sensor Fusion

Chen Chen, *Student Member, IEEE*, Roozbeh Jafari, *Senior Member, IEEE*, and Nasser Kehtarnavaz, *Fellow, IEEE*

***Abstract*— This paper presents a human action recognition system that runs in real time and simultaneously uses a depth camera and an inertial sensor based on a previously developed sensor fusion method. Computationally efficient depth image features and inertial signals features are fed into two computationally efficient collaborative representative classifiers. A decision-level fusion is then performed. The developed real-time system is evaluated using a publicly available multimodal human action recognition data set by considering a comprehensive set of human actions. The overall classification rate of the developed real-time system is shown to be >97%, which is at least 9% higher than when each sensing modality is used individually. The results from both offline and real-time experimentations demonstrate the effectiveness of the system and its real-time throughputs.**

***Index Terms*— Human action recognition, real-time human action recognition system, depth camera sensor, wearable inertial sensor, sensor fusion.**

I. Introduction

HUMAN action recognition is finding its way into commercial products and is of benefit to many human-computer interface applications. Example applications include hand gesture interaction, smart assistive living, and gaming. Different sensors have been used to perform human action recognition. These sensors include conventional RGB cameras, e.g. [1]–[3], depth cameras, in particular Kinect, e.g. [4]–[7], and inertial sensors, e.g. [8]–[10].

In our previous works [11]–[13], it was shown that improvements in recognition rates can be achieved by combining or fusing the information from a depth camera and an inertial sensor over the situations when each of these sensors is used individually due to the complementary aspect of the information provided by these two differing modality sensors. In [13], we reported a human action recognition method which involved the development of depth motion map features and the utilization of a collaborative representation classifier. However, the experimental analysis reported in [13] was conducted based on the data that were collected simultaneously from the sensors. In this paper, we have made modifications to the method reported in [13] in order to produce a human action recognition system which runs in real-time. These modifications include (i) adding a module to automatically detect the start and end of an action in real-time, (ii) modifying the fusion approach to reduce the computational complexity for real-time operation, (iii) carrying out extensive experimentations in offline and real-time manner for both subject-generic and subject-specific scenarios.

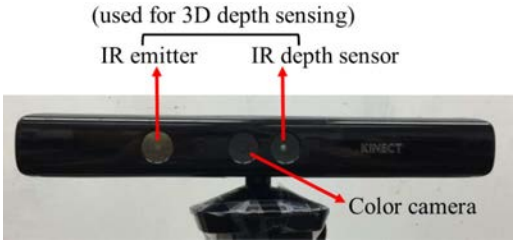

Fig. 1. Microsoft Kinect depth sensor.

The rest of the paper is organized as follows. In section II, an overview of the sensors and techniques used in our fusion method is provided. In section III, the modifications made in order to produce a real-time human action recognition system are presented. The experimental results for both offline and real-time recognition are included in section V. Finally, the conclusion appears in section VI.

II. Overview of Sensor Fusion Method

In this section, an overview of the sensors and techniques used in our fusion method in [13] is stated so that the stage is set for the modifications made in the next section towards enabling real-time operation.

A. Sensors

Kinect is a low-cost RGB-Depth camera sensor introduced by Microsoft for human-computer interface applications. It comprises a color camera, an infrared (IR) emitter, an IR depth sensor, a tilt motor, a microphone array, and an LED light. A picture of the Kinect sensor or depth camera is shown in Fig. 1. This sensor can capture 16-bit depth images with a resolution of 320×240 pixels. Two example depth images are depicted in Fig. 2. The frame rate is approximately 30 frames

Manuscript received June 30, 2015; revised September 9, 2015; accepted October 2, 2015. Date of publication October 5, 2015; date of current version January 12, 2016. This work was supported by the National Science Foundation under Grant CNS-1150079. The associate editor coordinating the review of this paper and approving it for publication was Prof. Danilo Demarchi.

C. Chen and N. Kehtarnavaz are with the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: chenchen870713@gmail.com; kehtar@utdallas.edu).

R. Jafari is with the Biomedical Engineering, Computer Science and Engineering, and Electrical and Computer Engineering Departments, Texas A&M University, College Station, TX 77843 USA (e-mail: rjafari@tamu.edu).

Digital Object Identifier 10.1109/JSEN.2015.2487358

Fig. 2. Example depth images from Kinect depth sensor.

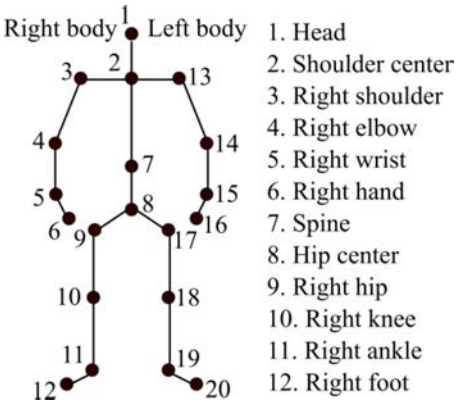

Fig. 3. Skeleton joints provided by Kinect depth sensor.

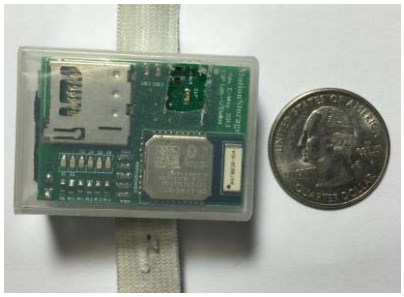

Fig. 4. Wearable inertial sensor developed in the ESP Lab.

per second. In addition, the Kinect SDK [14] is a publicly available software package which can be used to track 20 body skeleton joints (see Fig. 3) and their 3D spatial positions.

The wearable inertial sensor used in this work is a small size (1"×1.5") wireless inertial sensor built in the Embedded Signal Processing (ESP) Laboratory at Texas A&M University [15]. This sensor captures 3-axis acceleration, 3-axis angular velocity and 3-axis magnetic strength, which are transmitted wirelessly via a Bluetooth link to a laptop/PC. This wearable inertial sensor is shown in Fig. 4. The sampling rate of the sensor is 50 Hz and its measuring range is ±8g for acceleration and ±1000 degrees/second for rotation. It is worth mentioning that other commercially available inertial sensors can also be used in place of this inertial sensor. For practicality reasons or to avoid the intrusiveness associated with asking subjects to wear multiple inertial sensors, only one inertial sensor is considered in our work, either worn on the right wrist (similar to a watch) or the right thigh as depicted in Fig. 5 depending on the action of interest to be recognized in a particular application. More explanations about

Fig. 5. Inertial sensor placements: right wrist or right thigh.

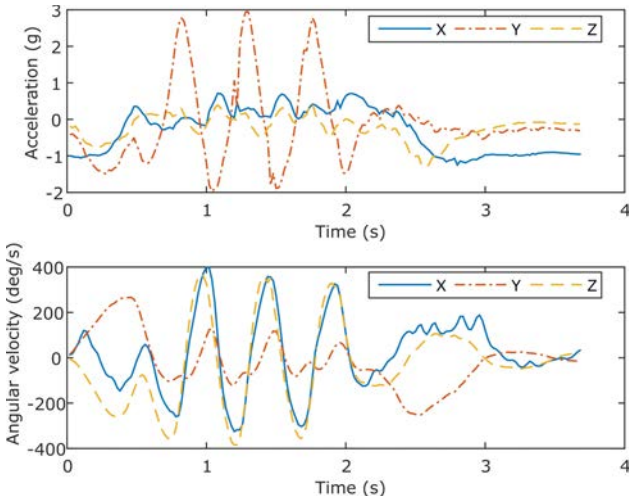

Fig. 6. Inertial sensor signals (3-axis accelerations and 3-axis angular velocities) for the action *right hand wave*.

the placement of the sensor for different actions are stated in Section IV. Fig. 6 shows the inertial sensor signals (3-axis accelerations and 3-axis angular velocities) for the action *right hand wave*.

B. Feature Extraction

To extract features from depth images, depth motion maps (DMMs) discussed in [7] are used due to their computational efficiency. More specifically, each 3D depth image in a depth video sequence is first projected onto three orthogonal Cartesian planes to generate three 2D projected maps corresponding to front, side, and top views, denoted by map_f, map_s, and map_t, respectively. For a depth video sequence with N frames, the DMMs are obtained as follows:

$$DMM_{\{f,s,t\}} = \sum_{i=1}^{N-1} \left| map^{i+1}_{\{f,s,t\}} - map^{i}_{\{f,s,t\}} \right|, \qquad (1)$$

where i represents frame index. A bounding box is considered to extract the non-zero region in each DMM and the foreground extracted DMMs are then used to serve as features. Since foreground DMMs of different video sequences may have different sizes, bicubic interpolation is applied to resize all such DMMs to a fixed size and thus to reduce the intra-class variability. An example set of DMMs for the action *one hand wave* is shown in Fig. 7. For the system developed

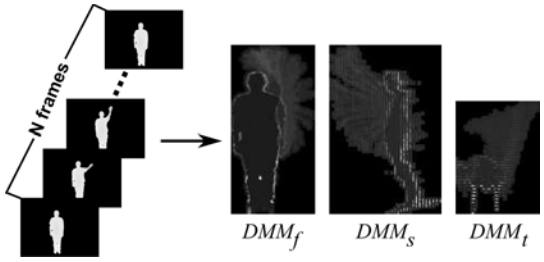

Fig. 7. DMMs generated from a sample video of the action *one hand wave*.

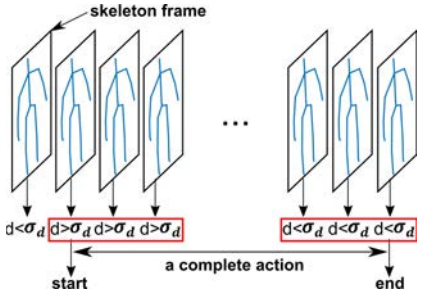

Fig. 8. Action segmentation illustration using skeleton joint positions.

in this paper, only the DMM generated from the front view, i.e. DMM_f, is processed in order to keep the computational complexity low towards achieving real-time throughputs.

For the inertial sensor, each acceleration and gyroscope signal sequence is partitioned into M temporal windows as reported in [16]. Three statistical features of *mean*, *variance*, and *standard deviation* are computed for each direction per temporal window. All the features from the temporal windows are concatenated to form a single combined feature vector. Thus, for M windows, the feature vector dimensionality is $3 \times M \times 3 \times 2 = 18M$.

C. Collaborative Representation Classifier

Collaborative representation classifier (CRC) [17] is a computationally efficient classifier that has been used in many image classification applications. Let C denote the number of classes and $\boldsymbol{X}_j \in \mathbb{R}^{D \times n_j}$ denote the training samples of class j (each column of $\boldsymbol{X}_j$ is a D-dimensional sample). Also, let $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_C] \in \mathbb{R}^{D \times n}$ denote the set of all the training samples, where $n = n_1 + \cdots + n_C$ is the total number of training samples. In this classifier, a test sample $\boldsymbol{y} \in \mathbb{R}^D$ is represented as a linear combination of all the training samples $\boldsymbol{X}$:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\alpha}, \tag{2}$$

where $\boldsymbol{\alpha}$ is an n-dimensional coefficients vector corresponding to all the training samples from C classes.

An l_2-norm is then considered to regularize $\boldsymbol{\alpha}$ based on this optimization formulation

$$\hat{\boldsymbol{\alpha}} = \arg \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_2^2, \tag{3}$$

where λ is a regularization parameter. The l_2-regularized minimization of (3) is in the form of the Tikhonov regularization [18] leading to the following closed form solution:

$$\hat{\boldsymbol{\alpha}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}. \tag{4}$$

Let $\boldsymbol{P} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T$. Given training sample set $\boldsymbol{X}$ and with λ determined via these samples, $\boldsymbol{P}$ is independent of a test sample $\boldsymbol{y}$. Therefore, $\boldsymbol{P}$ can be pre-computed as a projection matrix. Once a test sample arrives, the corresponding coefficient vector $\hat{\boldsymbol{\alpha}}$ can be simply found via $\boldsymbol{P}\boldsymbol{y}$, which is computationally efficient. According to the class labels of the training samples, $\hat{\boldsymbol{\alpha}}$ can be partitioned into C subsets $\hat{\boldsymbol{\alpha}} = [\hat{\boldsymbol{\alpha}}_1, \hat{\boldsymbol{\alpha}}_2, \ldots, \hat{\boldsymbol{\alpha}}_C]$ where $\hat{\boldsymbol{\alpha}}_j$ represents the coefficient vector associated with class j. The classification is made by

$$label\,(\boldsymbol{y}) = \arg\min_j \{e_j\}, \tag{5}$$

where $e_j = \|\boldsymbol{y} - \boldsymbol{X}_j\hat{\boldsymbol{\alpha}}_j\|_2$ denotes the residual error, and $\hat{\boldsymbol{y}}_j = \boldsymbol{X}_j\hat{\boldsymbol{\alpha}}_j$ indicates a class-specific representation of $\boldsymbol{y}$.

III. Modifications Made for Real-Time System

A. Detection of Action Start and End

For real-time operation, it is necessary to identify the start and end of an action. Action segmentation is a challenging task. In our system, it is made a requirement that a subject performs an action naturally by completing the action without any pause in the middle of the action. Furthermore, it is required that an action begins with a static posture and ends with a static posture lasting for at least one second. For example, for the action *right hand wave*, a subject stands in front of the Kinect camera and wears the inertial sensor on his/her right wrist. The static posture is the stand still posture. These requirements allow the action segmentation to be performed in a computationally efficient manner.

In our real-time system, the initial skeleton frames from the Kinect and the initial samples from the accelerometer for a one second duration are used to verify a static posture. Let $J_s = (x_s, y_s, z_s)$ denote the average 3D position of a static posture and $A_s = a\_x_s, a\_y_s, a\_z_s)$ its average accelerations. Note that one can easily obtain these static posture data before the actual real-time operation. When a subject starts an action, the position of a corresponding joint will deviate from the position of the static posture. The following distance between the 3D joint position $J = (x, y, z)$ in a skeleton frame and J_s is then computed

$$d = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2}. \tag{6}$$

If for m_s consecutive skeleton frames, all the distances are greater than a specified sensitivity σ_d, the start of an action is triggered. If for m_s consecutive skeleton frames, all the distances are less than or equal to the specified sensitivity σ_d, the end of an action is triggered. Fig. 8 illustrates the procedure of using skeleton joint positions to indicate the start and end of an action. The use of m_s consecutive skeleton frames avoids responding to possible signal jitters.

An example of a subject performing the action *right hand wave* is shown in Fig. 9. Fig. 9(a) exhibits the 3D positions of the right wrist over time. Fig. 9(b) exhibits the distance d

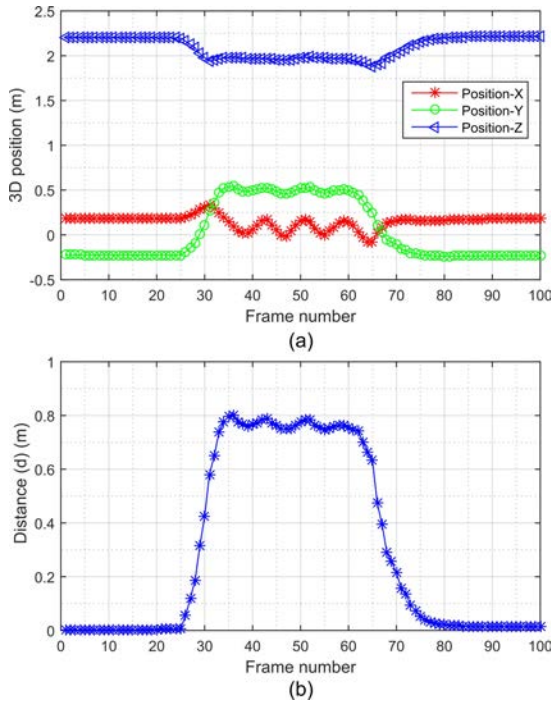

Fig. 9. (a) 3D positions of the right wrist joint for *right hand wave* action. (b) Corresponding joint position distance d versus skeleton frame number.

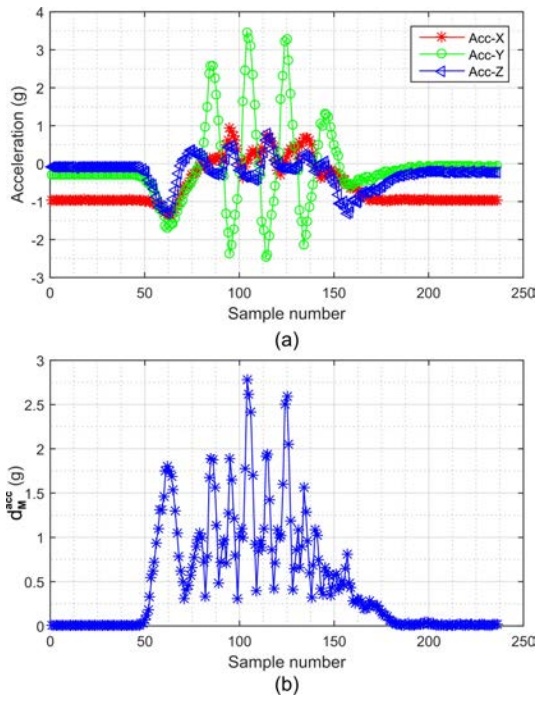

Fig. 10. (a) 3-axis acceleration signals from the inertial sensor placed on the right wrist for the action *right hand wave*. (b) Corresponding absolute acceleration magnitude difference d_M^{acc} versus sample number.

between the right wrist position in each skeleton frame and the static posture J_s. As seen from Fig. 9, the distance d reflects the change in the right wrist position. For this example, the starting point of the action occurs around the 25$^{\text{th}}$ frame and the ending point occurs around the 80$^{\text{th}}$ frame.

The sensitivity parameter σ_d essentially controls the desired level of detection sensitivity. If σ_d is set too high, the sensitivity to the start of an action is lowered resulting in a late start and an early end. If σ_d is set too low, the sensitivity to the end of an action is lowered resulting in an early start and a late end. In the results section, a subsection on parameter setting is provided giving guidelines as how to set this parameter.

For the inertial sensor, the acceleration signal is used as another clue to determine the start and end of an action. The accelerometer in the inertial sensor generates 3-axis acceleration signals. For an acceleration sample $A = (a\_x, a\_y, a\_z)$, the magnitude $M^{acc} = \sqrt{a_x^2 + a_y^2 + a_z^2}$ is computed and compared with the magnitude of the static posture $M_s^{acc} = \sqrt{a\_x_s^2 + a\_y_s^2 + a\_z_s^2}$. This absolute magnitude difference between A and A_s is then computed $d_M^{acc} = |M^{acc} - M_s^{acc}|$. The 3-axis acceleration signals of a sample *right hand wave* action are illustrated in Fig. 10(a) and the corresponding absolute magnitude difference in Fig. 10(b). Next, if all the distances d_M^{acc} for m_a consecutive accelerations are greater than a specified sensitivity σ_a, the start of an action is triggered. If all the distances d_M^{acc} for m_a consecutive accelerations are less than or equal to the sensitivity σ_a, the end of an action is triggered.

Note that d for each skeleton frame and d_M^{acc} for each acceleration are computed, and the start and end of an action is detected when either one is triggered.

As part of our real-time human action recognition system, depth videos, skeleton joint positions, and inertial sensor signals are generated in two software threads. One thread is used for simultaneous capture of depth videos and skeleton positions, and the other thread for the inertial sensor signals (3-axis acceleration and 3-axis rotation signals). For data synchronization, a time stamp for each sample is recorded. Since the frame rate of the Kinect camera and the sampling rate of the wearable inertial sensor are different, the start and end of an action are synchronized by using such time stamps. More specifically, when using skeleton positions to identify the start and end of an action, let the time stamp of the starting depth/skeleton frame of an action sequence be t_D^s and the time stamp of the ending depth/skeleton frame of an action sequence be t_D^e. Then, the two time stamps (denoted by t_I^s and t_I^e) of the inertial sensor samples that are closest to t_D^s and t_D^e are used in order to identify the first and last sample of an action. The same procedure is applied when using acceleration signals to identify the start and end of an action.

B. Fusion Method

Two sets of features are generated and fused from depth images and inertial sensor signals. In our previous work [13], both the feature-level fusion and the decision-level fusion were examined. Although the feature-level fusion, i.e. concatenating two differing sets of features, was found simple and straightforward, it suffers from some shortcomings for real-time operation. First, the increase in the dimensionality of the fused feature vector increases the computational complexity for classification. Second, there exist incompatibilities with the two sets of features. For example, the dimensionality of the depth feature vector (i.e. DMM_f in vector form) is typically much higher than the inertial sensor signal feature vector.

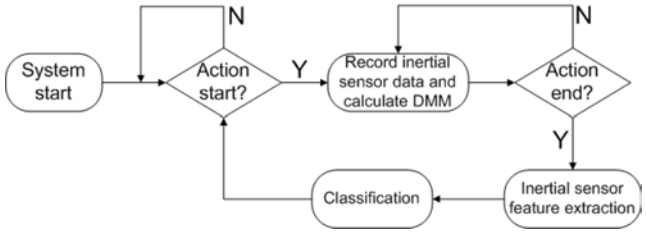

Fig. 11. Flowchart of the real-time human action recognition system.

Moreover, the numerical ranges of the two sets of features differ significantly. For the system to be able to operate in real-time, the decision-level fusion is implemented here.

For actual operation or testing of a sample $\boldsymbol{y}$, F_K and F_I are generated and used individually as inputs to two CRC classifiers as described in [13]. As a result, two error vectors $\boldsymbol{e}^K = [e_1^K, e_2^K, \ldots, e_C^K]$ and $\boldsymbol{e}^I = [e_1^I, e_2^I, \ldots, e_C^I]$ are generated, where $\boldsymbol{e}^K$ corresponds to the error vector of the CRC classifier using F_K and $\boldsymbol{e}^I$ to the CRC classifier using F_I. To merge the outcomes of the two classifiers, the logarithmic opinion pool (LOGP) [19] technique is employed. LOGP provides a so called soft fusion at the posterior-probability level. In LOGP, the individual posterior probability $p_q(\omega|\boldsymbol{y})$ of each classifier is used to estimate this global membership function

$$P(\omega|\boldsymbol{y}) = \prod_{q=1}^{Q} p_q(\omega|\boldsymbol{y})^{\alpha_q}, \tag{7}$$

where $\omega \in [1, \ldots, C]$ denotes a class label, Q the number of classifiers ($Q = 2$ in our case), and with α_q being uniformly distributed (that is, $\alpha_q = \frac{1}{Q}$). According to the residual output $\boldsymbol{e} = [e_1, e_2, \ldots, e_C]$, a Gaussian mass function

$$p_q(\omega|\boldsymbol{y}) = \exp(-\boldsymbol{e}), \tag{8}$$

is then employed which indicates a smaller residual error $e_j (j \in [1, \ldots, C])$ yields a higher probability $p_q(\omega|\boldsymbol{y})$. Therefore, in the implemented decision-level fusion, this fused probability from the two classifiers is considered

$$P(\omega|\boldsymbol{y}) = \exp(-\boldsymbol{e}^K)^{\frac{1}{2}} \times \exp(-\boldsymbol{e}^I)^{\frac{1}{2}}. \tag{9}$$

The final class label for $\boldsymbol{y}$ is then assigned to the class with the largest probability $P(\omega|\boldsymbol{y})$ with $\boldsymbol{e}^K$ and $\boldsymbol{e}^I$ normalized to [0, 1]. Note that the LOGP fusion employed here is computationally more efficient than the Dempster-Shafer theory fusion [20] that was used in [13].

The flowchart of the real-time operation of the system is shown in Fig. 11. The detection of an action start and end is continuously performed. After detecting an action start, the fusion classification method is activated while monitoring for the action end. Note that the DMM gets computed frame by frame. The DMM feature computation is completed when the end of an action is detected.

IV. Experimental Results and Discussion

A. Parameter Setting

For action segmentation, appropriate values for the parameter m_s, σ_d, m_a, and σ_a need to be set first. In [21], it was recommended making the start of an action within 4 depth/skeleton frames of its actual start. In experiments reported in [22], a latency of 4 frames resulted in a perceived latency of 0 to 1 frame. Hence, the parameter m_s corresponding to the number of consecutive frames was set to 4 in our experimentations. To set a proper value for σ_d, 60 skeleton frames in a static posture were used to calculate the mean μ_d of the distance d in Eq. (6) between the wrist joint positions and J_s. It was found that any sensitivity parameter σ_d in the range of $[2\mu_d, 5\mu_d]$ led to an acceptable level of visual latency of less than 30ms.

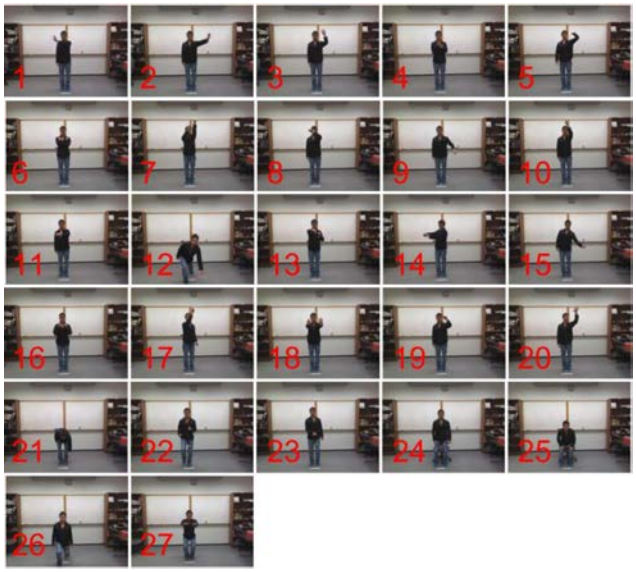

Fig. 12. Sample shots of the 27 actions in the UTD-MHAD database.

For acceleration signals, m_a was set to 8 since the sampling rate of the inertial sensor was about twice the frame rate of the Kinect sensor. 100 acceleration samples in a static posture were used to calculate the mean μ_a of the difference signal d_M^{acc}. Again, it was found that any σ_a in the range of $[2\mu_a, 5\mu_a]$ led to an acceptable level of visual latency of less than 30ms. A similar approach was considered in [23] for action segmentation using acceleration and gyroscope signals.

Furthermore, the same parameters reported in [24] were used here for the size of DMM_f (i.e., 150×75) and the number of temporal windows (i.e., $M = 6$). As a result, the dimensionality of the depth feature vector and the inertial sensor feature vector were 11250 and 108, respectively.

B. Offline Analysis

This section includes various experimentations that were conducted to test our developed real-time human action recognition system. This system was first tested on the publicly available database called University of Texas at Dallas Multimodal Human Action Dataset (UTD-MHAD) [24]. The dataset can be downloaded from http://www.utdallas.edu/~kehtar/UTD-MHAD.html.

UTD-MHAD consists of four temporally synchronized data modalities. These modalities include RGB videos, depth videos, skeleton positions from a Kinect camera sensor, and

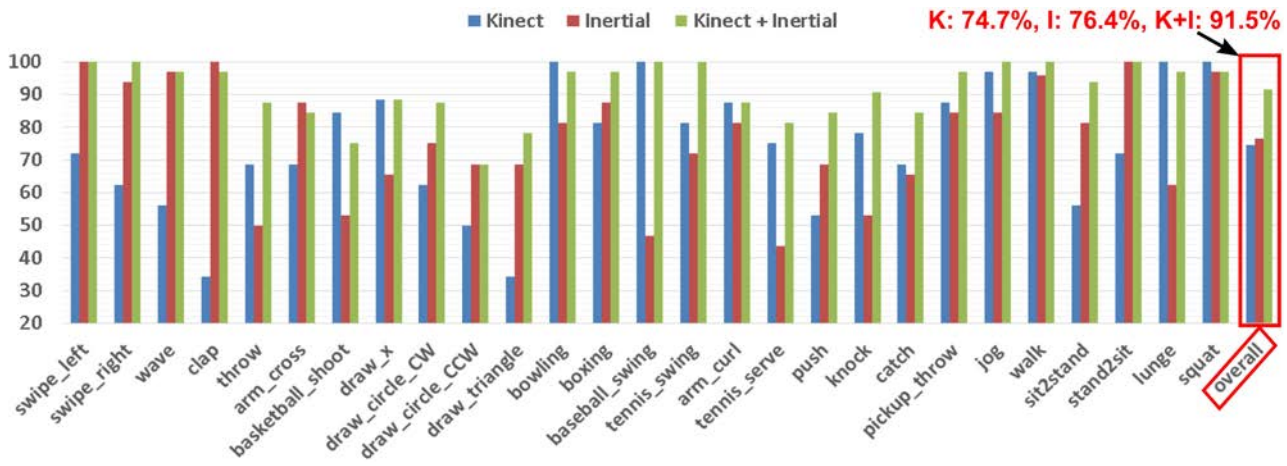

Fig. 13. Classification performance (recognition rates per action class and overall recognition rate) when using Kinect sensor only, inertial sensor only, and Kinect and inertial sensors fusion for the subject-generic experiment.

inertial signals from a wearable inertial sensor for a comprehensive set of 27 human actions encountered in the literature on human action recognition. The 27 actions are as follows: (1) *right arm swipe to the left*, (2) *right arm swipe to the right*, (3) *right hand wave*, (4) *two hand front clap*, (5) *right arm throw*, (6) *cross arms in the chest*, (7) *basketball shoot*, (8) *right hand draw X*, (9) *right hand draw circle (clockwise)*, (10) *right hand draw circle (counter clockwise)*, (11) *draw triangle*, (12) *bowling (right hand)*, (13) *front boxing*, (14) *baseball swing from right*, (15) *tennis right hand forehand swing*, (16) *arm curl (two arms)*, (17) *tennis serve*, (18) *two hand push*, (19) *right hand knock on door*, (20) *right hand catch an object*, (21) *right hand pick up and throw*, (22) *jogging in place*, (23) *walking in place*, (24) *sit to stand*, (25) *stand to sit*, (26) *forward lunge (left foot forward)*, (27) *squat (two arms stretch out)*. The 27 actions were performed by 8 subjects (4 females and 4 males). Each subject repeated each action 4 times. After removing three corrupted sequences, the dataset includes 861 data sequences. Sample shots of the 27 actions in the UTD-MHAD are shown in Fig. 12. The wearable inertial sensor was placed on the subjects' right wrists for actions (1) through (21) which were hand type movements, and on the subjects' right thigh for actions (22) through (27) which were leg type movements.

To demonstrate the advantages of sensor fusion or using depth and inertial sensors together for human action recognition, two types of experiments based on UTD-MHAD were performed. The first experiment is named subject-generic. More specifically, leave one subject out test was carried out. In other words, each time a subject was used as the testing subject (i.e., the action samples associated with this subject were regarded as testing samples) and the remaining seven subjects were used as the training subjects (i.e., the action samples associated with these seven subjects were regarded as training samples), which resulted in an 8-fold cross validation. The name subject-generic is used here as there was no training samples associated with the testing subject. Under this experimental setting, the classification performance was examined using the Kinect depth features only, the inertial sensor features only, and the combination of the depth and inertial sensor features (decision-level fusion). Fig. 13 displays the class-specific recognition rates and the overall recognition rate of these three situations. Note that the class-specific recognition rates and the overall recognition rate appearing in this figure are the averages over the 8 subjects (i.e., 8-fold cross validation). As evident from this figure, the fusion improved the classification performance by more than 15% over the situations when each sensor was used individually.

In the second experiment, the samples from only one subject were divided into a training and a testing set. Since each subject had performed an action 4 times in the database, the first two samples of each action was used to form the training set and the remaining samples to form the testing set. In this case, all the training and testing samples were associated with the same subject. This is named the subject-specific experiment. This experimental setting was repeated for all the 8 subjects. Again, the three situations of Kinect sensor only, inertial sensor only, and Kinect and inertial sensor fusion were examined. The results obtained are shown in Fig. 14. As evident from Fig. 14, the fusion led to a superior classification performance compared to the situations when using the Kinect sensor only or the inertial sensor only. The overall recognition rate of the fusion reached 97.2% which, as expected, was higher than the recognition rate in the subject-generic experiment. This is because since the training and testing samples were from the same subject, the intra-class variation was less compared to the subject-generic experiment. The larger intra-class variation in the subject-generic experiment occurred due to different body sizes (e.g., heights) of the subjects in the depth images and different subjects performing the same actions differently.

From Fig. 14, it can be seen that the fusion approach improved the classification rates for the great majority of the actions. For some actions, e.g., *draw X*, *boxing and tennis serve*, the fusion resulted in the same classification rates as when using the Kinect sensor only or inertial sensor only. However, for the three actions of *clap*, *draw circle counter clockwise*, and *jogging*, the classification rates of the fusion

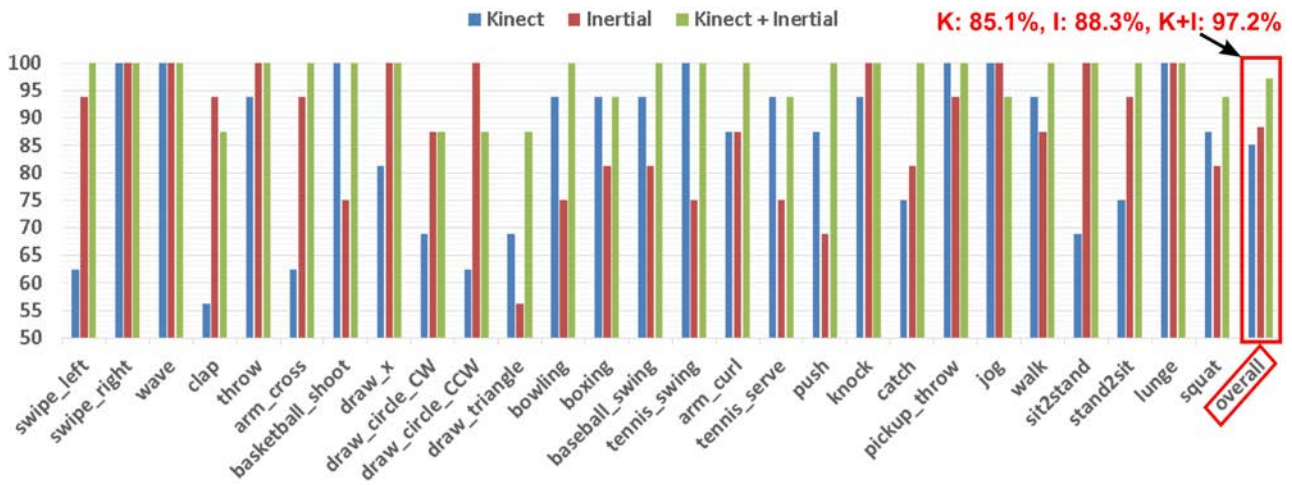

Fig. 14. Classification performance (recognition rates per action class and overall recognition rate) when using Kinect sensor only, inertial sensor only, and Kinect and inertial sensors fusion for the subject-specific experiment.

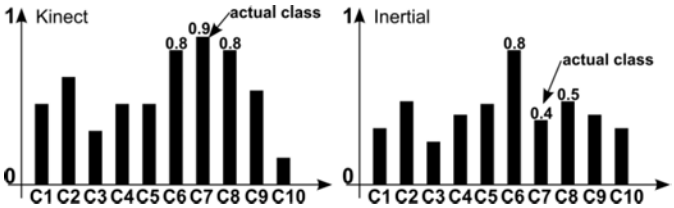

Fig. 15. A misclassification case using Kinect and inertial sensor fusion.

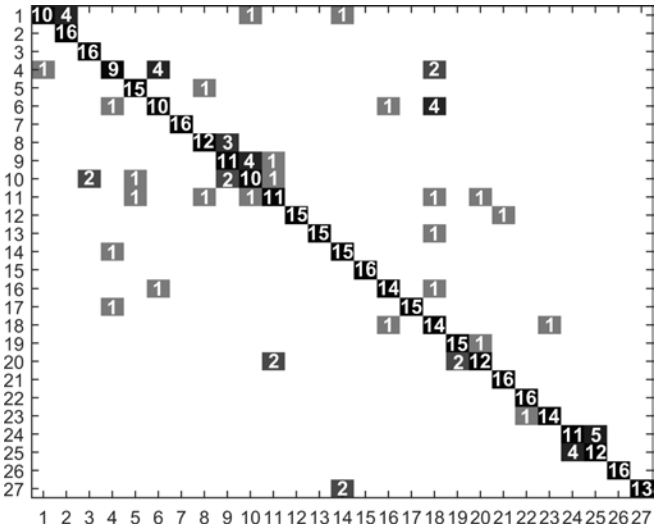

Fig. 16. Recognition confusion matrix when using Kinect sensor only.

became lower than when using the Kinect sensor only or the inertial sensor only. This was caused by the fused probability of these three actions not favoring the actual class. An example of this case is shown in Fig. 15. In this example, an action C7 was correctly classified by using the Kinect sensor features since the class C7 had the highest probability of 0.9. The probability of the class C7 however was 0.4 when using the inertial sensor features. By fusing the probability outputs from the Kinect and the inertial sensor, the probability of the class C6 was made the highest ($0.8\times0.8=0.64$) which in turn was higher than the probability of the class C7 ($0.9\times0.4=0.36$), leading to the selection of the wrong class when using the fused probability output. Although for these three actions, the fusion did not improve the classification rates, it is important to note that only one or two misclassified samples for these actions occurred for the total number of 16 testing samples in these classes.

To further show the classification performance, the confusion matrices corresponding to the three scenarios of the subject-specific experiment are shown in Figures 16 through 18. In these figures, the 27 actions are numbered for a compact display of the matrices. The diagonal elements in the matrices indicate the correctly classified number of samples for the actions. The sum of each row indicates the total number of samples for the corresponding action. By comparing these three confusion matrices, it can clearly be seen that the fusion achieved higher overall classification performance compared to using each sensor individually.

C. Real-Time Operation Results

In this subsection, the results of our developed human action recognition system for a typical real-time run are provided. Five subjects participated in the results reported here while running the system in real-time. Each subject was first asked to perform the actions twice for the system training purposes or the generation of training samples. A typical training time of the system was less than a minute once the training session was over. The testing or actual operation of the system occurred in real-time. The same subject performed the actions twice during the testing. As it was mentioned earlier, in general, the subject-specific scenario leads to higher classification performance and it is deemed more appropriate to use in commercial products. It is worth emphasizing here that each subject performed the actions naturally and followed the action segmentation requirements.

The confusion matrix of a real-time run of the system for the five subjects is shown in Fig. 19. As can be seen from this figure, the real-time results were similar to the offline analysis results based on the UTD-MHAD dataset.

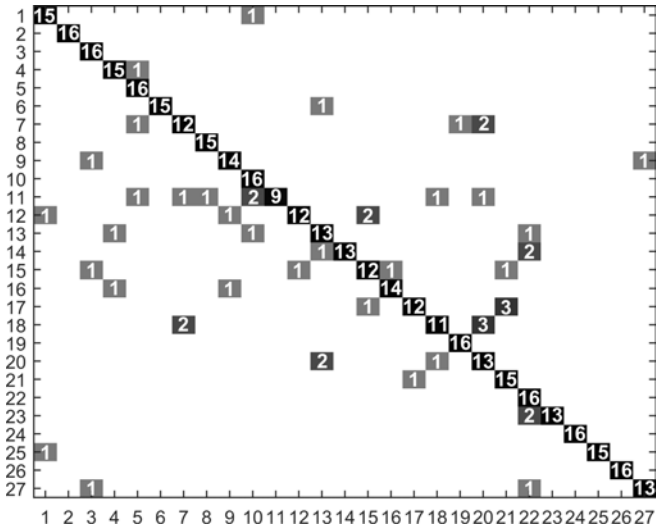

Fig. 17. Recognition confusion matrix when using inertial sensor only.

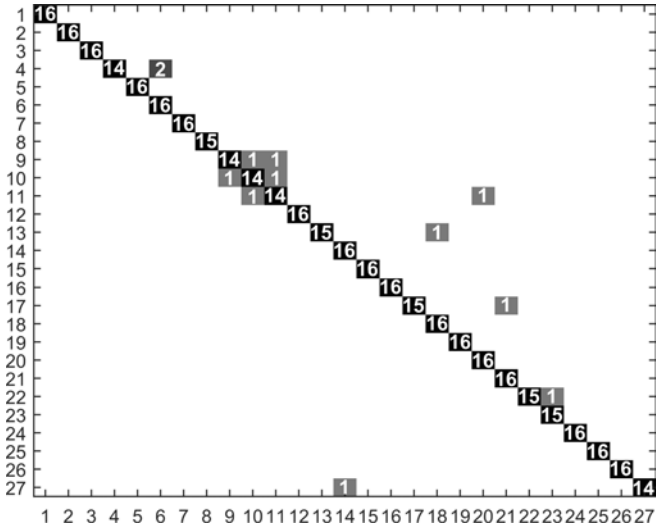

Fig. 18. Recognition confusion matrix when using Kinect and inertial sensors fusion.

TABLE I
PROCESSING TIMES TAKEN BY THE MAJOR COMPONENTS OF THE REAL-TIME SYSTEM

System components	Average processing time (ms)
DMM computation	0.48 / frame
Inertial feature extraction	0.61 / sample
Fusion classification	2.8 / sample

The real-time human action recognition system was programmed using Microsoft Visual Studio 2012 (C/C++). The system runs in real-time on a typical modern desktop computer. The computer used had a 3.4 GHz Intel Core i7 CPU with 8 GB RAM. The processing time of the major components of the program is listed in Table I, indicating achieving real-time throughputs. A video clip of the system running in real-time can be viewed at http://www.utdallas.edu/∼kehtar/FusionDemo.wmv.

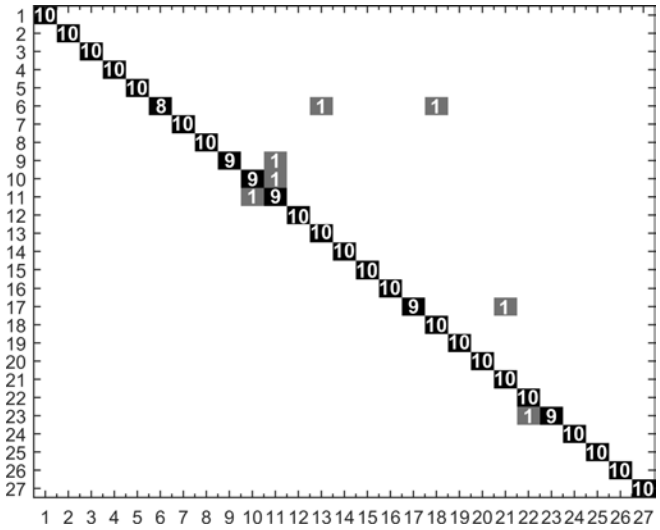

Fig. 19. Real-time recognition confusion matrix.

V. CONCLUSION

In this paper, a real-time fusion system for human action recognition has been developed that uses data from two differing modality sensors: vision depth and inertial. The system merges the probability outputs of the features from these two differing modality sensors in real-time via a decision-based fusion method involving collaborative representation classifiers. The extensive experimental results reported have indicated the effectiveness of the system towards recognizing human actions in real-time compared to the situations when using each sensor individually. In our future work, we plan to examine specific applications of the fusion framework presented in this paper by using depth cameras and wearable inertial sensors that have recently become commercially available including the second generation Kinect depth camera [25], Texas Instruments time-of-flight depth camera [26], Google Tango miniaturized depth camera [27], Samsung Gear [28], and Apple Watch [29].

ACKNOWLEDGMENT

Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

REFERENCES

[1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. IEEE 17th Int. Conf. Pattern Recognit.*, vol. 3. Cambridge, U.K., Aug. 2004, pp. 32–36.

[2] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 2004–2011.

[3] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[4] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, San Francisco, CA, USA, Jun. 2010, pp. 9–14.

[5] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 872–885.

[6] C. Chen, R. Jafari, and N. Kehtarnavaz, "Action recognition from depth sequences using depth motion maps-based local binary patterns," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Waikoloa Beach, HI, USA, Jan. 2015, pp. 1092–1099.
[7] C. Chen, K. Liu, and N. Kehtarnavaz, "Real-time human action recognition based on depth motion maps," *J. Real-Time Image Process.*, pp. 1–9, Aug. 2013. [Online]. Available: http://link.springer.com/article/10.1007%2Fs11554-013-0370-1
[8] P. Gupta and T. Dallas, "Feature selection and activity recognition system using a single triaxial accelerometer," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 6, pp. 1780–1786, Jun. 2014.
[9] A. Y. Yang, R. Jafari, S. S. Sastry, and R. Bajcsy, "Distributed recognition of human actions using wearable motion sensor networks," *J. Ambient Intell. Smart Environ.*, vol. 1, no. 2, pp. 103–115, 2009.
[10] C. Chen, N. Kehtarnavaz, and R. Jafari, "A medication adherence monitoring system for pill bottles based on a wearable inertial sensor," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Chicago, IL, USA, Aug. 2014, pp. 4983–4986.
[11] B. Delachaux, J. Rebetez, A. Perez-Uribe, and H. F. S. Mejia, "Indoor activity recognition by combining one-vs.-all neural network classifiers exploiting wearable and depth sensors," in *Proc. 12th Int. Work-Conf. Artif. Neural Netw.*, Puerto de la Cruz, Spain, Jun. 2013, pp. 216–223.
[12] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz, "Fusion of inertial and depth sensor data for robust hand gesture recognition," *IEEE Sensors J.*, vol. 14, no. 6, pp. 1898–1903, Jun. 2014.
[13] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 1, pp. 51–61, Feb. 2015.
[14] [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/, accessed Jun. 17, 2015.
[15] C. Chen, K. Liu, R. Jafari, and N. Kehtarnavaz, "Home-based senior fitness test measurement system using collaborative inertial and depth sensors," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Chicago, IL, USA, Aug. 2014, pp. 4135–4138.
[16] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley MHAD: A comprehensive multimodal human action database," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Clearwater, FL, USA, Jan. 2013, pp. 53–60.
[17] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 471–478.
[18] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. Washington, DC, USA: Winston, 1977.
[19] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.
[20] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ, USA, Princeton Univ. Press, 1976.
[21] V. Bloom, D. Makris, and V. Argyriou, "G3D: A gaming action dataset and real time action recognition evaluation framework," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Providence, RI, USA, Jun. 2012, pp. 7–12.
[22] C. Marais. (2011). *Gesture Detection Using Machine Learning*. [Online]. Available: http://www.microsoft.com/en-us/download/confirmation.aspx?id=28066, accessed Jun. 17, 2015.
[23] S. Patil, H. R. Chintalapalli, D. Kim, and Y. Chai, "Inertial sensor-based touch and shake metaphor for expressive control of 3D virtual avatars," *Sensors*, vol. 15, no. 6, pp. 14435–14457, Jun. 2015.
[24] C. Chen, R. Jafari, and N. Kehtarnavaz, "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *Proc. IEEE Int. Conf. Image Process.*, Quebec City, QC, Canada, Sep. 2015, pp. 1–5.
[25] [Online]. Available: https://www.microsoft.com/en-us/kinectforwindows/meetkinect/default.aspx, accessed Jun. 23, 2015.
[26] [Online]. Available: http://www.ti.com/ww/en/analog/3dtof/, accessed Jun. 23, 2015.
[27] [Online]. Available: https://www.google.com/atap/project-tango/, accessed Jun. 23, 2015.
[28] [Online]. Available: http://www.samsung.com/us/mobile/wearable-tech, accessed Jun. 23, 2015.
[29] [Online]. Available: http://www.apple.com/watch/?cid=wwa-us-kwg-watch-com, accessed Jun. 23, 2015.

Chen Chen (S'10) received the B.E. degree in automation from Beijing Forestry University, Beijing, China, in 2009, and the M.S. degree in electrical engineering from Mississippi State University, Starkville, in 2012. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX. His research interests include compressed sensing, signal and image processing, pattern recognition, and computer vision.

Roozbeh Jafari (SM'12) received the Ph.D. degree in computer science from UCLA. He completed a post-doctoral fellowship at UC-Berkeley. He is currently an Associate Professor of Biomedical Engineering, Computer Science and Engineering, and Electrical and Computer Engineering with Texas A&M University. His research interest lies in the area of wearable computer design and signal processing. His research has been funded by the NSF, NIH, DoD (TATRC), AFRL, AFOSR, DARPA, SRC, and industry (Texas Instruments, Tektronix, Samsung, and Telecom Italia). He has authored over 100 papers in refereed journals and conferences. He was a recipient of the NSF CAREER Award in 2012, the IEEE Real-Time and Embedded Technology and Applications Symposium Best Paper Award in 2011, and the Andrew P. Sage Best Transactions Paper Award from the IEEE Systems, Man and Cybernetics Society in 2014. He is an Associate Editor of the IEEE SENSORS JOURNAL, the IEEE INTERNET OF THINGS JOURNAL, and the IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS.

Nasser Kehtarnavaz (F'12) received the Ph.D. degree in electrical and computer engineering from Rice University, in 1987. He is currently a Professor of Electrical Engineering and the Director of the Signal and Image Processing Laboratory, University of Texas at Dallas. His research areas include signal and image processing, real-time processing on embedded processors, biomedical image analysis, and pattern recognition. He has authored or co-authored nine books and more than 300 publications in these areas. He has had industrial experience in various capacities with Texas Instruments, AT&T Bell Labs, the U.S. Army TACOM Research Lab, and the Houston Health Science Center. He is currently the Editor-in-Chief of the *Journal of Real-Time Image Processing*, and the Chair of the SPIE Conference on Real-Time Image and Video Processing. He is a Fellow of SPIE, and a licensed Professional Engineer.