Weighted Fusion of Depth and Inertial Data to Improve View Invariance for Human Action Recognition

Chen Chen[a], Huiyan Hao[a,b], Roozbeh Jafari[c], Nasser Kehtarnavaz[a]
[a]Center for Research in Computer Vision, University of Central Florida; [b]School of Information and Communication Engineering, North University of China; [c]Center for Remote Health Technologies and System, Texas A&M University

ABSTRACT

This paper presents an extension to our previously developed fusion framework [10] involving a depth camera and an inertial sensor in order to improve its view invariance aspect for human action recognition applications. A computationally efficient view estimation based on skeleton joints is considered in order to select the most relevant depth training data when recognizing test samples. Two collaborative representation classifiers, one for depth features and one for inertial features, are appropriately weighted to generate a decision making probability. The experimental results applied to a multi-view human action dataset show that this weighted extension improves the recognition performance by nearly 6% over equally weighted fusion deployed in our previous fusion framework.

Keywords: Fusion of depth and inertial data, view invariant human action recognition, weighted fusion framework

1. INTRODUCTION

Human action recognition based on depth cameras, in particular Microsoft Kinect, has been extensively studied in the literature for various applications including intelligent surveillance, human-computer interactions, and virtual reality, e.g. [1-4]. A depth camera provides depth images or a 3D structure of the human body in the scene. Due to differences in performing actions from subject to subject and variations in environmental conditions, there are still challenges in achieving robust human action recognition.

With the advancement of Micro-Electro-Mechanical Systems (MEMS), wearable inertial sensors such as accelerometers and gyroscopes are increasingly being utilized for action recognition, e.g. [5-7]. This sensor technology provides an alternative approach toward performing action recognition by utilizing 3D acceleration and rotation signals associated with an action. Considering the complementary aspect of the 3D action data provided by these two types of sensors, i.e. depth camera and inertial sensor, our research team has previously developed a number of action recognition solutions by utilizing both of these sensors at the same time [8-13]. As a result of fusing the data from these two differing sensor modalities, it has been shown that recognition rates are improved compared to situations when each sensor is used individually.

In this paper, an extension is made to our previous fusion framework to further cope with variations in depth images that are caused by the way a subject faces a depth camera. Similar to our previous works, a depth camera and an inertial sensor are used simultaneously irrespective of how the depth camera is placed in the scene as long as subjects appear in the camera field of view. In the literature when a depth camera is used for human action recognition, one sees that actions are often performed in a frontal view setting and there has been limited study of the effect of changing the subject orientation with respect to the camera on the recognition outcome. Thus, the focus of this paper is on studying view variations within the context of our fusion framework.

In the extension developed here, the classifiers are weighted not equally in order to gain more robustness when training data incorporate samples for subjects facing the camera at different viewing angles. To retain the computational efficiency aspect of our previous fusion framework, instead of using computationally intensive view-invariant features (e.g., [14]), a computationally simple view angle estimation is used here. The developed approach requires obtaining training data from different views. The view estimation allows using only the training samples of a specific view when examining test samples. The contributions made in this paper are two-fold: (1) the utilization of a computationally efficient view estimation based on the skeleton joint positions, and (2) a weighted fusion to assign different weights to

the two classifiers that are associated with depth features and inertial features. A block diagram of the developed weighted fusion framework is shown in Fig. 1.

The remainder of the paper is organized as follows. Section 2 describes the two sensors used in our fusion framework. Section 3 covers a computationally efficient and simple method to estimate different views by using skeleton joint positions. The weighted fusion framework is then presented in Section 4. The experimental results and discussion are stated in Section 5. Finally, the paper is concluded in Section 6.

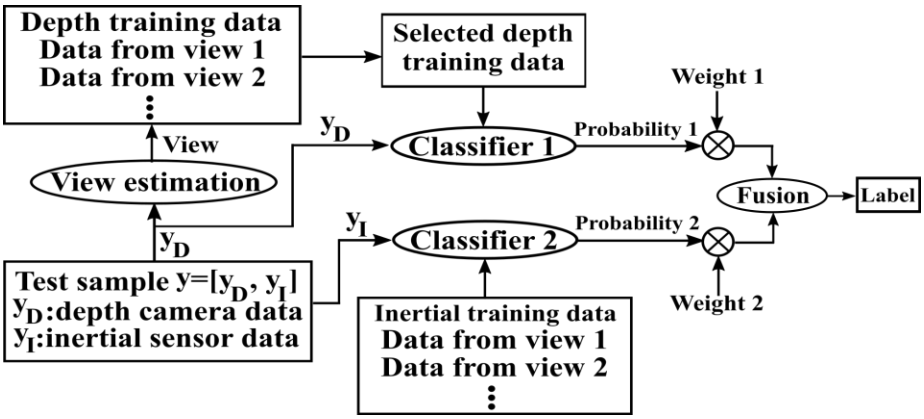


Figure 1. Weighted fusion framework.

2. DEPTH CAMERA AND INERTIAL SENSOR

The Microsoft Kinect v2 sensor (see Fig. 2(a)) comprises a color camera and an infrared depth camera. This sensor has a depth image resolution of 512 × 424 pixels. The frame rate is approximately 30 frames per second. The Kinect Windows SDK 2.0 software package [15] allows tracking 25 human skeleton joints as illustrated in Fig. 2(c).

The wearable inertial sensor used in this work is a small size (1"×1.5") wireless inertial sensor (see Fig. 2(b)) built in the Embedded Signal Processing (ESP) Laboratory at Texas A&M University [16]. This sensor captures 3-axis acceleration and 3-axis angular velocity which are transmitted wirelessly via a Bluetooth link to a laptop/PC. The sampling rate of the inertial sensor is 50Hz and its measuring range is ±8g for acceleration and ±1000 degrees/second for rotation. Although it is possible to use more than one sensor for actions that involve both arm and leg movements, only one inertial sensor is utilized in this work as the actions examined are all hand movements. The wearable inertial sensor provides view invariant data because it generates data based on its local coordinates. Once the sensor is placed on the body, the acceleration and rotation signals remain more or less the same irrespective of how the subject faces the depth camera.

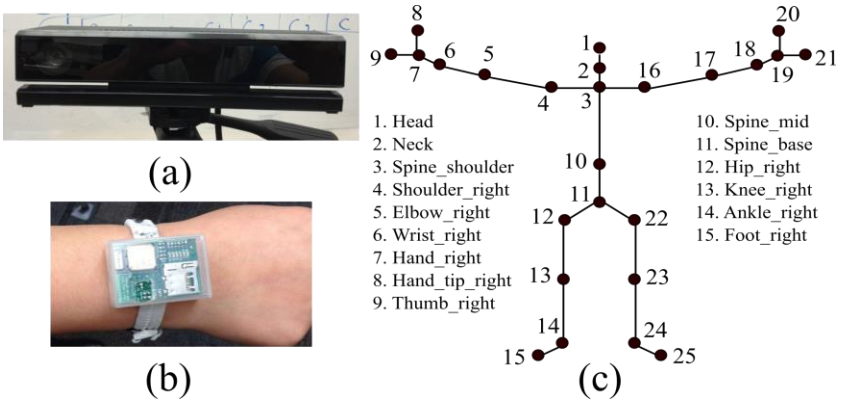


Figure 2. (a) Microsoft Kinect v2 camera. (b) Wearable inertial sensor. (c) 3D skeleton with 25 tracked skeleton joints.

3. COMPUTATIONALLY EFFICIENT VIEW ESTIMATION

As shown in Fig. 3, in practice, when performing an action, a subject's orientation facing the depth camera may be different. This figure illustrates five generic views: frontal view, left and right half-profile views, left and right full-profile views. If the subject's orientation with respect to the camera can be estimated, then one can simply use the training depth data of that view to perform action recognition. In order to maintain the real-time operation of our previously developed fusion framework [10], it is critical for such an estimation to be computationally efficient since the real-time operation of the entire recognition pipeline needs to be maintained. This is achieved by using the skeleton joint positions that are provided by the Kinect SDK 2.0. The SDK provides the 3D coordinates (x, y, z) of a joint, where the z value indicates the distance between the joint and the Kinect depth camera.

As can be seen from Fig. 3, the positions of shoulders are different for different orientations of the subject with respect to the camera and are more stable (less jitter) than other hand joints in depth images. Let z_{ls} denote the distance between the left shoulder (*ls*) and the camera, z_{sc} the distance between the shoulder center (*sc*) and the camera, and z_{rs} the distance between the right shoulder (*rs*) and the camera. It is easy to see that z_{ls}, z_{sc} and z_{rs} being close to each other indicate the subject is facing the camera. The condition $z_{ls} < z_{rs}$ indicates the subject is facing at a right angle towards the camera (e.g., right 45° in Fig. 4), and the condition $z_{ls} > z_{rs}$ indicates the subject is facing at a left angle towards the camera (e.g., left 45° in Fig. 3).

Fig. 4 shows the plots of the shoulder joints (*ls* , *sc* and *rs*) positions for the first 10 skeleton frames of an action (*catch*) performed by a subject at different angles or orientations with respect to the depth camera. The reason the first 10 skeleton frames of an action is considered here is that the subject is normally in a rest position for the first 1-2 seconds (approximately 30-60 skeleton frames) before performing an action. As noted from this figure, z_{ls} and z_{rs} are very close (see Fig. 4(a)) to z_{sc} when the subject stands facing straight towards the camera. The subject turning to the left or right is simply determined by comparing z_{ls} and z_{rs}. Moreover, the positions *ls* , *sc* and *rs* at the right/left 45° angle are quite close to those at the right/left 90° angle. As a result, here the views for 45° and 90° are combined and treated as side views compared to frontal views covering the view range of +/- 45°. Also, it is noted that the distance between *ls* and *sc* ($|z_{ls}-z_{sc}|$) and the distance between *rs* and *sc* ($|z_{rs}-z_{sc}|$) are greater than 50mm in side views. The pseudocode shown in Fig. 5 is used to determine the view or a subject's orientation facing the camera in a computationally efficient manner.

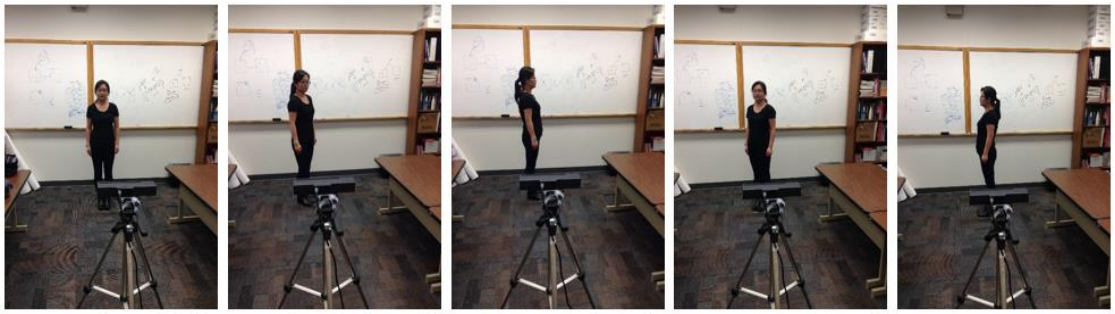
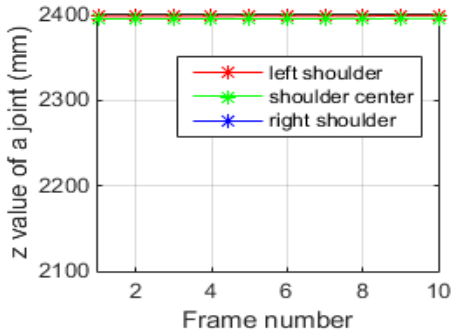
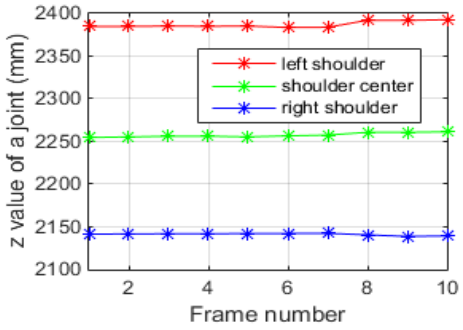


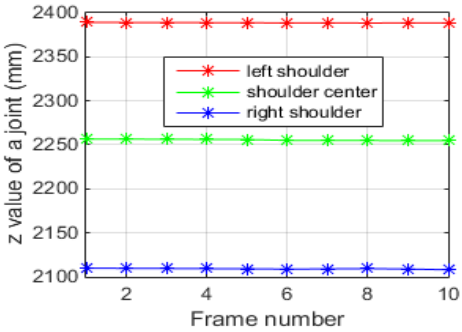
Figure 3. Five different standing positions of a subject with respect to the Kinect depth camera.



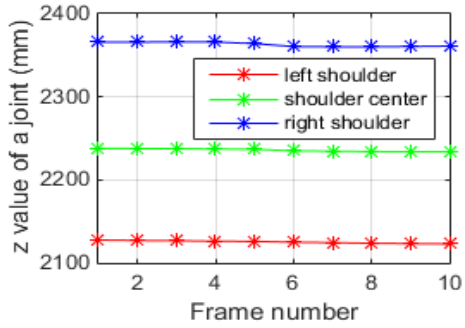
(a) View 1 (front)



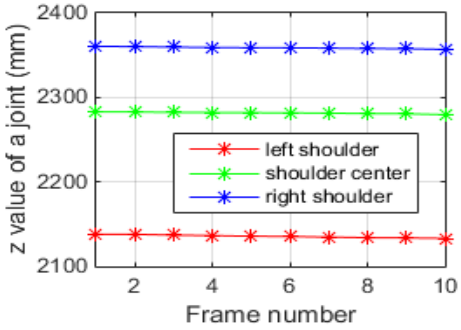
(b) View 2 (left 45°)



(c) View 3 (left 90°)



(d) View 4 (right 45°)



(e) View 5 (right 90°)

Figure 4. Positions of three shoulder joints for five different subject orientations with respect to the depth camera.

IF $|z_{ls} - z_{sc}| \leq 50\text{mm}$ and $|z_{rs} - z_{sc}| \leq 50\text{mm}$

    View = front;

ELSEIF $z_{ls} > z_{rs}$

    View = left side views (i.e., left 45° and 90°)

ELSE

    View = right side views (i.e., right 45° and 90°)

END

Figure 5. Pseudocode for view estimation.

4. WEIGHTED FUSION FRAMEWORK

For action recognition, features are extracted from depth images and inertial sensor signals. To extract features from depth images, the depth motion maps (DMMs) discussed in [1] are used due to their computational efficiency. Each 3D depth image in a depth 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 sequence with N frames, the DMMs are obtained as follows:

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

where i represents frame index. Here, the foreground (non-zero region) of each DMM is extracted and used as depth features. Since foreground DMMs of different video sequences may have different sizes, a bi-cubic interpolation is applied to resize all such DMMs to a fixed size and thus to reduce the intra-class variability. An example of DMMs of the action *catch* corresponding to three different views is illustrated in Fig. 6.

For the inertial sensor, the same feature extraction method reported in [8] is used where each acceleration and orientation signal sequence is partitioned into M temporally separated windows. Four statistical features (*mean*, *variance*, *standard deviation* and *root mean square*) are computed for each direction per temporal window. All the features from the temporal windows are then concatenated to form a single inertial feature vector.

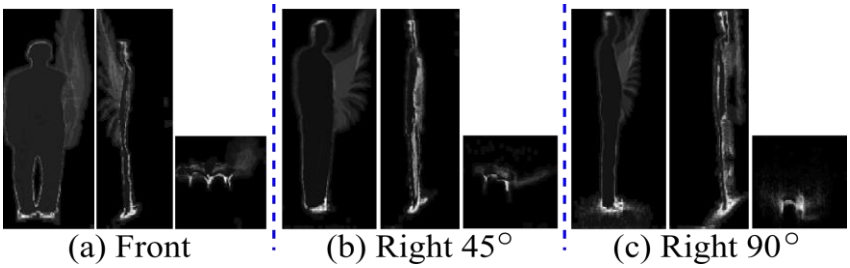


Figure 6. DMMs of action *catch* corresponding to three different views.

After extracting the depth and inertial features, the fusion method developed in [10] is applied. Specifically, the collaborative representation classifier (CRC) [17] is utilized due to its computational efficiency and high classification performance. For a test action y, its depth features denoted by $F_D(y)$ and its inertial features denoted by $F_I(y)$ are used separately as inputs to two CRCs. Each classifier generates a probability output by employing a Gaussian mass function on the residual errors as described in [10]. The logarithmic opinion pool (LOGP) [18] is then employed to estimate this global membership function:

$$\log P\left(\omega|y\right) = \sum_{q=1}^{Q} \alpha_q p_q\left(\omega|y\right), \tag{2}$$

where $\omega \in \{1,2,...,C\}$ indicates the class label from C action classes, Q indicates the number of classifiers (here $Q=2$), $p_q\left(\omega|y\right)$ denotes the probability output of the q^{th} classifier, and α_q represents appropriate weights assigned to the classifiers. In our previous fusion framework [10], equal weights (i.e., $\alpha_q = 1/Q$) were considered.

However, it is deemed more effective to assign unequal weights to different sensor features depending on their importance in the decision making process. A larger weight indicates a greater importance or role played by one of the sensors. Therefore, by assigning unequal weights, one gains more flexibility in the fusion framework as far as view invariance is concerned. For the weighted fusion, the non-negativity and sum-to-one constraints are imposed. As a result, the fused probability output can be rewritten as:

$$\log P\left(\omega|y\right) = \beta p_1\left(\omega|F_D(y)\right) + (1-\beta) p_2\left(\omega|F_I(y)\right), \tag{3}$$

where $\beta(\beta \geq 0)$ and 1-β are the weights assigned to the two CRCs using depth $F_D(y)$ and inertial $F_I(y)$ features, respectively. The final class label ω^* is then determined as follows:

$$\omega^* = \arg\max_{\omega=1,\ldots,C} P\left(\omega \middle| y\right). \tag{4}$$

5. EXPERIMENTAL RESULTS AND DISCUSSION

This section covers the results of our experimentations based on the weighted fusion framework and a comparison with the original fusion framework in [10]. First, a multi-view action dataset was collected by using a Kinect depth camera and an inertial sensor simultaneously. The data synchronization from the two sensors was achieved by using the method described in [9]. The inertial sensor was placed on the right wrist of subjects. The dataset included the following six actions: *catch*, *draw circle*, *draw tick*, *draw triangle*, *knock* and *throw*. These six actions were chosen from the UTD-MHAD dataset [9] due to their similarity to make the action recognition problem more challenging. Five subjects were asked to perform each action with five different subject orientations or views as shown in Fig. 3. For each view, a subject repeated an action 6 times (i.e., 6 trials). Therefore, in total 900 action samples were generated. This dataset is made available for public use at this link: http://www.utdallas.edu/~kehtar/UTD-MHAD.html.

For each subject, half of the samples (or 3 samples) per action from all fives views were used for training, resulting in 450 training samples in total from the five subjects. The remaining 450 samples were used for testing. For the 450 testing samples, they were separated by views in order to examine the view invariance of the developed weighted fusion framework. Each set of 90 samples from a view were tested and the recognition rate was found for each view.

The sizes of the DMMs were determined by the average sizes of the DMMs associated with the training samples as noted in [1]. As per the approach in [10], the number of temporal windows M for the inertial feature extraction was set to 6. The weight β was determined by carrying out a 5-fold cross validation. The weight β was varied from 0 to 1 in 0.1 steps. As an example, the outcome of the 90 testing samples associated with view 1 (i.e., test set for view 1) is reported here to see the effect of different values of β. Figure 7 shows the recognition outcome when using different weights. As can be seen from this figure, the cross validation revealed β=0.3 generated the highest recognition outcome. This cross-validation approach needs to be repeated when considering a different training set. The weight obtained in this manner indicated that it was more effective to attach a higher weight to the inertial sensor features towards gaining a more robust view invariance.

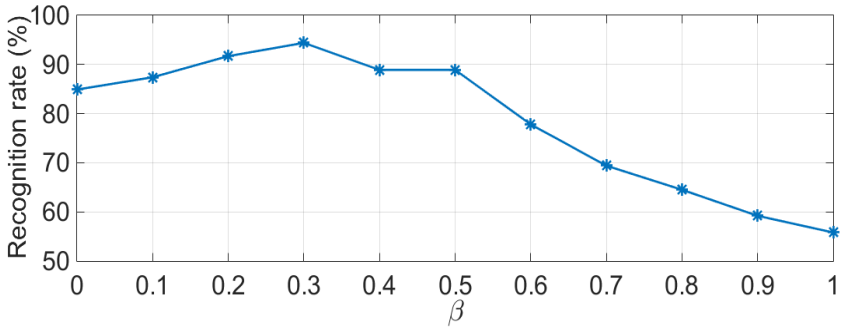
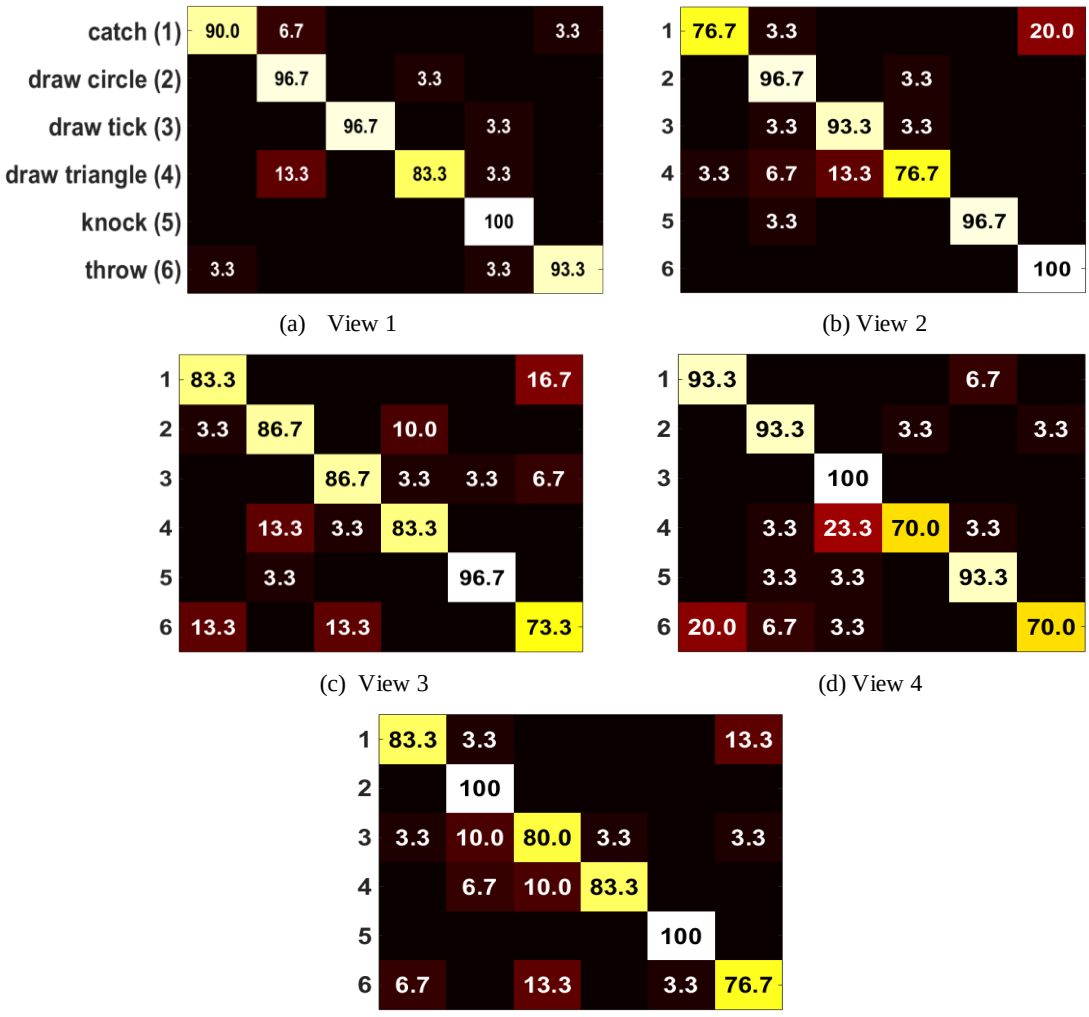


Figure 7. Recognition rate (%) *vs*. weight β.

The original fusion with equal weights, denoted by D+I, the original fusion with view estimation denoted by D+I+VE, and the weighted fusion with view estimation, denoted by Weighted D+I+VE, were compared by using the leave-one-subject-out testing approach. For each view, the recognition rates for the five subjects were averaged and the results are reported in Table 1. This table shows that the view estimation for selecting the training samples of a specific view, that is D+I+VE, improved the recognition performance for each view as compared with the original fusion framework which used all the training samples from all the views. This is because the view estimation allows the training samples of the view that is more similar to a test sample to be used for the probabilistic CRC decision making. Moreover, when utilizing the weighted fusion, the recognition performance was further improved. The Weighted D+I+VE framework

achieved the highest recognition rates in every view leading to an improvement of 5.6% on average over the original fusion framework. For a more detailed examination of the recognition rates, the confusion matrices of the weighted fusion are displayed in Fig. 8 for the five different views. As indicated in this figure, the greatest amount of confusion occurred when the depth and inertial features were not discriminatory enough to distinguish similar actions with subtle differences.



(a) View 1 (b) View 2

(c) View 3 (d) View 4

(e) View 5

Figure 8. Confusion matrices corresponding to six actions for five views utilizing the weighted fusion framework with view estimation.

Table 1. Comparison of recognition rates (%) with original fusion framework. (D=depth, I=inertial, VE=view estimation)

	View1	View2	View3	View4	View5	Average
D+I	87.2	86.1	76.7	79.4	84.4	82.8
D+I+VE	92.2	87.8	81.7	83.9	85.6	86.2
Weighted D+I+VE	**93.3**	**90.0**	**85.0**	**86.7**	**87.2**	**88.4**

Here it is worth stating that although our developed weighted fusion framework to improve view invariance requires training samples from different views, it is important to note that such training samples can be easily obtained in applications such as gaming, rehabilitation, etc.

Finally, the computational efficiency of the proposed fusion framework is presented. The program is implemented in MATLAB and run on a laptop with a 2.6 GHz Intel Core i7 CPU with 12 GB of RAM. The processing time of the major components of the program is listed in Table 2, indicating achieving real-time throughputs.

Table 2. Processing times (mean ± std) associated with the components of our method.

System components	Average processing time (ms)
View estimation	0.05 ± 0.01 / action
DMM computation	6.4 ± 3.1 / depth frame
Inertial feature extraction	1.4 ± 0.5 / action
Fusion classification	3.2 ± 1.2 / action

6. CONCLUSION

This paper has presented an extension of our previously developed fusion framework in [10] to gain more robust view invariance for human action recognition applications. A depth camera and an inertial sensor have been used simultaneously in this framework. This extension includes a computationally efficient view estimation and a weighted fusion of probabilities for the two collaborative representation classifiers in the fusion framework. The results obtained indicate that this extension increases the recognition tolerance to different orientation angles that subjects may be facing the depth camera.

REFERENCES

[1] Chen, C., Liu, K. and Kehtarnavaz, N., "Real-time human action recognition based depth motion maps," Journal of Real-Time Image Processing, 12(1), 155-163 (2016).

[2] Li, W., Zhang, Z. and Liu, Z., "Action recognition based on a bag of 3D points," Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops, 9-14 (2010).

[3] Chen, C., Hou, Z., Zhang, B., Jiang, J. and Yang, Y., "Gradient local auto-correlations and extreme learning machine for depth-based activity recognition," Proc. the 11th International Symposium on Visual Computing, 613-623 (2015).

[4] Oreifej, O. and Liu, Z., "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," Proc. IEEE Conference on Computer Vision and Pattern Recognition, 716-723 (2013).

[5] Jovanov, E., Milenkovic, A., Otto, C. and de Groen, P. C., "A wireless bodyarea network of intelligent motion sensors for computer assisted physical rehabilitation," Journal of NeuroEngineering and Rehabilitation, 2(6), 1-10 (2005).

[6] Yang, A., Jafari, R., Sastry, S. S. and Bajcsy, R., "Distributed recognition of human actions using wearable motion sensor networks," Journal of Ambient Intelligence and Smart Environments, 1(2), 103-115 (2009).

[7] Lockhart, J., Pulickal, T. and Weiss, G., "Applications of mobile activity recognition," Proc. of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, 1054-1058 (2012).

[8] Chen, C., Jafari, R. and Kehtarnavaz, N., "Improving human action recognition using fusion of depth camera and inertial sensors," IEEE Transactions on Human-Machine Systems, 45(1), 51-61 (2015).

[9] Chen, C., Jafari, R. and Kehtarnavaz, N., "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," Proc. of IEEE International Conference on Image Processing, 168-172 (2015).

[10] Chen, C., Jafari, R. and Kehtarnavaz, N., "A real-time human action recognition system using depth and inertial sensor fusion," IEEE Sensors Journal, 16(3), 773-781 (2016).
[11] Liu, K., Chen, C., Jafari, R. and Kehtarnavaz, N., "Fusion of inertial and depth sensor data for robust hand gesture recognition," IEEE Sensors Journal, 14(6), 1898-1903 (2014).
[12] Chen, C., Liu, K., Jafari, R. and Kehtarnavaz, N., "Home-based senior fitness test measurement system using collaborative inertial and depth sensors," Proc. of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 4135-4138 (2014).
[13] Chen, C., Jafari, R. and Kehtarnavaz, N., "Fusion of depth, skeleton, and inertial data for human action recognition," Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, 2712-2716 (2016).
[14] Gong, D. and Medioni. G., "Dynamic manifold warping for view invariant action recognition," Proc. of the 13th International Conference on Computer Vision, 571-578 (2011).
[15] https://www.microsoft.com/en-us/kinectforwindows/develop/
[16] Chen, C., Kehtarnavaz, N. and Jafari, R., "A medication adherence monitoring system for pill bottles based on a wearable inertial sensor," Proc. of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 4983-4986 (2014).
[17] Zhang, L., Yang, M. and Feng, X., "Sparse representation or collaborative representation: Which helps face recognition?" Proc. of IEEE International Conference on Computer Vision, 471-478 (2011).
[18] Chen, C., Jafari, R. and Kehtarnavaz, N., "Action recognition from depth sequences using depth motion maps-based local binary patterns," Proc. of the IEEE Winter Conference on Applications of Computer Vision, 1092-1099 (2015).